SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2025, Vol. 68, Iss. 2, 122303:1–122303:21
https://doi.org/10.1007/s11432-024-4258-x

Satellite-assisted 6G wide-area edge intelligence: dynamics-aware task offloading and resource allocation for remote IoT services

Di ZHAO[1,2], Rui DING[3] & Bin SONG[1,2*]

[1]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China*
[2]*Hangzhou Institute of Technology, Xidian University, Hangzhou 311231, China*
[3]*China Satellite Network Group Co., Ltd., Beijing 100029, China*

Received 9 August 2024/Revised 2 November 2024/Accepted 16 December 2024/Published online 16 January 2025

Abstract Future 6G networks will integrate non-terrestrial communication with mobile edge computing to enable wide-area edge intelligence, providing ubiquitous communication and computation services for everyone and everything. In this paper, we introduce the intelligent cloud-edge-device architecture into low Earth orbit (LEO) satellite-supported remote Internet of Things (IoT) networks to facilitate latency-sensitive and compute-intensive IoT applications. However, the dynamic spatio-temporal characteristics inherent in LEO satellite networks, such as the high volatility of task traffic and the high mobility of space nodes, pose severe challenges. For efficient task execution, we investigate the joint optimization of cross-region task offloading and cross-domain resource allocation, and innovatively propose a spatio-temporal attention-based proximal policy optimization (STA-PPO) algorithm. Specifically, the temporal attention-based actor network analyzes the timing dependencies of task arrival and makes time-variant decisions. Simultaneously, the spatial attention-based critic network captures the spatial variations of satellite motion and evaluates the topology-related value. We verify the effectiveness of the cloud-edge-device collaboration through extensive simulations. Numerical results demonstrate that the STA-PPO algorithm outperforms benchmarks, showing the lowest system delay and the highest link throughput.

Keywords 6G wide-area edge intelligence, cross-region task offloading, cross-domain resource allocation, spatio-temporal attention, proximal policy optimization

Citation Zhao D, Ding R, Song B. Satellite-assisted 6G wide-area edge intelligence: dynamics-aware task offloading and resource allocation for remote IoT services. Sci China Inf Sci, 2025, 68(2): 122303, https://doi.org/10.1007/s11432-024-4258-x

1 Introduction

The 6th generation (6G) mobile communication network will undergo a transformation from the traditional architecture of the Internet of Things (IoT) to the new paradigm of intelligent Internet of Everything (IoE) with global connectivity and pervasive artificial intelligence (AI) capabilities [1]. However, terrestrial networks cannot support global connections because they face limitations in fully spanning the intricate terrain of remote areas [2]. The recent advancements in non-terrestrial networks (NTN) technology provide a solution by enhancing satellite communications to deliver global coverage, ubiquitous connectivity and seamless communication services [3]. Consequently, the future 6G network will inevitably leverage satellite communications to fill coverage gaps and provide everyone-centric and everywhere-available services [4], such as emergency response, environment monitoring, and intelligent agriculture [5].

The emergence of wide-area IoT applications also brings new challenges, particularly in processing large volumes of compute-intensive tasks on remote IoT devices. Traditionally, these tasks are forwarded by the satellite network [6] and undertaken by distant data centers or cloud servers deployed in terrestrial networks. However, this approach is significantly impacted by the height of the satellites and leads to increased round-trip delay [7], which often fails to meet the real-time requirements of urgent tasks. As a solution, the integration of multi-access edge computing (MEC) technology [8] with low Earth orbit (LEO) satellite networks can reduce latency and improve the real-time performance. By sinking the computing capability to the edge side, the tasks that cannot be afforded by IoT devices will be offloaded

* Corresponding author (email: bsong@mail.xidian.edu.cn)

to LEO satellites. Then, LEO satellites equipped with edge servers have the capability to perform on-orbit computing [9], thereby reducing round-trip transmission cost and end-to-end service delay.

Although the benefits of integrating LEO satellites and MEC technology are evident, other critical considerations need to be taken into account. On the one hand, the high mobility of LEO satellites introduces uncertainties to task offloading. On the other hand, the available resources of LEO satellites are restricted by payload design, specifically in terms of weight and volume limitations. Considering these aspects, the joint optimization of task offloading and resource allocation in LEO satellite-enabled IoT networks is still an open issue for academic research and industrial practice [10]. The critical technical challenges are listed as follows.

• Dynamic spatio-temporal characteristics of LEO satellite networks. The temporal dynamics are reflected in traffic fluctuations [11], characterized by stochastic task arrivals and unknown task volumes. Without prior knowledge or information, it is challenging to promptly respond to time-varying task traffic. Moreover, the spatial dynamics manifest as the topological changes induced by LEO satellite orbital movements, e.g., moving space nodes and intermittent communication links [12]. The space-varying network topology also poses significant challenges to task offloading and resource allocation.

• Tightly-coupled nature between task offloading and resource allocation. This joint optimization problem has been proven to be NP-hard [13] due to the combinatorial property of task offloading and its intimate connection with resource allocation. The collaborative task offloading of massive IoT devices involves a vast number of possible combinations, leading to a dramatic increase in the search space and even a combinatorial explosion dilemma. Furthermore, the tight coupling between task offloading and resource allocation introduces additional intricacy, making it difficult to find a globally optimal solution.

To cope with the above challenges, deep reinforcement learning (DRL) has been developed as a typical tool for exploring pervasive artificial intelligence (AI) paradigms in the upcoming 6G networks [14, 15] due to its capabilities of self-learning and self-adapting in complex dynamic systems. In particular, the successful application of the proximal policy optimization (PPO) algorithm in wireless communication networks, including task offloading [16], bandwidth allocation [17], and energy optimization [18], has sparked great research interest. In this paper, we have chosen the PPO algorithm as the backbone for optimizing task offloading and resource allocation. The reason is that the PPO algorithm not only supports continuous action spaces, making it suitable for complex decision-making problems, but also ensures stability and efficiency in the update process through the Kullback-Leibler divergence and clipping mechanism. Moreover, the adaptability and scalability of the PPO algorithm enable it to perform well with limited data, which is crucial for dynamic LEO satellite networks that require real-time policy adjustments. However, traditional DRL-based methods often use fully connected networks (FCNs) as their main structure, making them less suitable for real-time strategic adjustments in LEO satellite networks. On the one hand, these approaches essentially mine the temporal correlations in task arrival at the expense of longer exploration time, which results in a still weak responsiveness to task bursts. On the other hand, these methods lack insights into network topology due to the inability to effectively represent and process structured data, rendering them ineffective for direct application in highly dynamic LEO satellite networks. Recently, the advancement of attention mechanisms has ushered in a new turning point for the above difficulty. Powerful attention-based learning models, such as Transformer, Bert, and ChatGPT [19,20], have been proposed to imitate human vision and cognition. It naturally comes to mind that by incorporating attention mechanisms into DRL methods [21], the learning model is expected to focus on dynamic or critical information to achieve fast convergence and better generalization. Therefore, we investigate how to exploit the focus capability of attention mechanisms and the exploration ability of DRL methods to design a dynamics-aware scheme, making adaptive decisions for everyone and everything in AI-empowered 6G networks.

• We explore an LEO-enabled edge intelligence scheme to support task offloading and computing for remote IoT applications, where IoT tasks and LEO satellites are characterized by temporal and spatial dynamics, respectively. Furthermore, we study the joint optimization of task offloading and resource allocation and transform it into a Markov decision process (MDP) to minimize the system delay, subject to coverage, communication, and computation capabilities of LEO satellites.

• We propose a dynamics-aware DRL algorithm that incorporates the attention mechanism into the actor-critic network to focus on and adapt to the inherent dynamic spatio-temporal variations in LEO satellite networks. The actor network utilizes temporal attention to mine historical regularities of task arrival and makes time-variant decisions. The critic network leverages spatial attention to extract topological relationships of satellite motion and evaluates topology-related state values and action advantages.

• In the proposed algorithm, we customize a multi-actor network to achieve satellite coverage, task offloading, as well as communication and computation resource allocation. Furthermore, we delve into the interrelationships of multi-dimensional actions and design tailored masking schemes to decouple them. First, the mask guided by satellite coverage guarantees the seamless execution of cross-region task offloading. Second, the mask provided by task offloading ensures the rationality and efficiency of cross-domain resource allocation.

• We conduct comprehensive simulations to verify the superiority of the proposed algorithm and its robust adaptability to ongoing spatial-temporal variations. Numerical results demonstrate that our dynamics-aware algorithm consistently outperforms other learning-based methods, achieving the lowest system delay and the highest link throughput.

The remainder of this paper is organized as follows. Section 2 introduces the related work on task offloading and resource allocation in satellite-enabled networks. Section 3 describes the system model. Then, the problem formulation and transformation are elaborated in Section 4. Then, we propose the dynamics-aware DRL algorithm in Section 5, and the performance of the proposed algorithm is evaluated in Section 6. Section 7 concludes this paper.

2 Related work

In this section, we first investigate task offloading schemes in satellite-supported networks, and then review how to schedule resources to enhance system performance.

2.1 Task offloading in satellite-enabled wide-area networks

Numerous studies have been proposed to explore the performance of diverse frameworks for task offloading and computing. Categorized by the offload destination, these frameworks can be further classified into cloud-local [22, 23], edge-local [24, 25], and cloud-edge-local [26] cooperative computing.

In [22], Mao et al. proposed a space-aerial-assisted computing framework, where LEO satellites provide ubiquitous cloud computing services. Similarly, Cheng et al. [23] presented a mixed edge-cloud computing idea for space-air-ground integrated networks (SAGINs). Here, LEO satellites serve as cloud servers to support away-user computing services, and flying unmanned aerial vehicles (UAVs) are deployed with edge servers to offer near-user computing abilities. Despite being cost-effective and easy to deploy, the endurance of UAVs is limited to a few hundred meters due to constraints in battery energy or fuel cells [27]. Consequently, there is a growing interest in integrating LEO satellites and edge computing to extend access range and endurance time.

Wang et al. [24] introduced a typical scene involving LEO satellites for edge computing, where tasks from remote devices can be performed locally or transmitted to LEO satellites via satellite-terrestrial communication. Wang et al. [25] further explored the integration of LEO satellite and edge computing for lower-latency applications. In particular, the study in [25] took into account the challenges posed by intermittent communication resulting from the orbital dynamics of LEO satellites. In [28], terrestrial-satellite terminals (TSTs) were introduced to divide the task offloading from IoT devices to LEO satellites into a two-stage process, i.e., the ground and space segments. Here, TSTs act as access points for IoT devices to collect offloading tasks and then upload them simultaneously to multiple LEO satellites to access edge computing services. However, constrained by the payload size of satellites and the transmission power of devices, it may be impractical to fully offload and serve a substantial number of tasks with limited edge computing capabilities and satellite-to-terrestrial transmission rates.

Recently, researchers have shifted their focus towards hybrid edge and cloud computing to build a three-tier computation architecture [26, 29, 30], aiming to provide heterogeneous computation services for global users. Chen et al. [31] studied the multi-layer parallel computation architecture in the civil aircraft-augmented SAGIN. In this architecture, ground devices perform end-to-side computing; civil aircraft and LEO satellites provide edge computing services; and ground stations within the satellite network are responsible for cloud computing. In [32], Jia et al. investigated the multi-task offloading problem by placing edge servers on LEO satellites and accessing cloud servers through geostationary orbit (GEO) satellite relays. Despite these advancements, the existing literature overlooks the highly dynamic characteristics introduced by LEO satellite orbiting, e.g., the frequent switching of communication links and dynamic changes in communication time. Therefore, our work focuses on developing dynamics-

aware task offloading schemes to answer when and where task computations should be executed to meet real-time requirements.

In addition, satellite edge intelligence plays a crucial role in areas such as direct-to-smartphone communication [33], environmental monitoring [34], and disaster detection in satellite-assisted wide-area IoT networks. For massive concurrent access scenarios [35], Zhang et al. [36] designed a satellite peer offloading scheme based on an online distributed algorithm, where task offloading is performed along multi-hop paths to explore collaborative computing capabilities. Unlike existing studies that focus on general computational tasks, Zhu et al. [37] proposed a dual-layer collaborative processing framework designed for important and unique hyperspectral image processing tasks. In [34], Leyva-Mayorga et al. formulated a satellite mobile edge computing framework for real-time and very-high resolution Earth observation, optimizing image distribution and compression parameters to minimize energy consumption. Jung et al. [38] established a hybrid LEO and UAV edge computing framework by jointly optimizing the bit allocation of communication and computation along with the UAV path planning, thereby assisting environment protection, military reconnaissance, and sea transportation.

2.2 Multi-dimensional resource allocation for task offloading

In satellite-enabled networks, efficient resource allocation, such as communication and computation resources, is crucial for tasks to achieve the lowest computing latency and minimum energy consumption.

Considering both the inner-coupling and inter-coupling relationships of multi-dimensional resources, numerous resource allocation methods based on optimization theory [26, 30] have been proposed. Chen et al. [31] considered the hybrid strategy optimization of user access, task offloading, and resource management. Here, parallel successive convex approximation (SCA) theory is employed to convert the multivariable NP-hard problem into a tractable problem, aiming to achieve a reasonable trade-off between service latency and energy cost. In a similar vein, Wei et al. [39] addressed that joint optimization problem using the linear reconstruction technique. In [40], the hybrid optimization problem of communication, computation and cache resource allocation was studied. To tackle this, the authors proposed a Lagrange dual decomposition (LDD)-based approach to derive the closed-form optimal value. However, existing studies have involved extensive relaxations and time-consuming repetitions, making the solutions vulnerable to the dynamic characteristics of LEO satellite networks.

With the advancement of AI technology, DRL emerges as a promising method for multi-dimensional resource allocation in response to the dynamic satellite communication environment. He et al. [41] developed a time-frequency joint resource allocation model based on the double deep Q-network algorithm to accommodate the dynamic changes in the number of users and their timeliness requirements. Apart from scalability in user size, several research efforts have focused on addressing the spatio-temporal dynamics in satellite communication networks. Acknowledging the dynamics in task arrival, Liao et al. [13] proposed a queue-aware actor-critic network to solve task offloading and resource scheduling. Zhou et al. [42] devised a novel risk-sensitive DRL algorithm for making real-time resource allocation decisions to keep pace with dynamic task arrivals. In addition, the high mobility of space and air nodes, coupled with the high dynamics of link conditions, poses significant challenges to resource allocation. Tang et al. [43] leveraged live topological information to develop a double Q-learning algorithm improved by delay-sensitive replay memory. Asheralieva et al. [44] innovatively proposed a hierarchical DRL algorithm to explore resource allocation and pricing. This algorithm combines deep Q-learning with Bayesian reinforcement learning to adapt to unstable and dynamic environments.

Recently, the integration of DRL with other emerging technologies, such as federated learning (FL), meta learning, digital twins, and multi-agent collaboration, has fostered the rapid development of intelligent edge AI services in satellite communication scenarios. In [45], a leader FL architecture for distributed satellite edge intelligence was developed, employing a distributed PPO approach to optimize the computing capability and transmit power of satellites. Zhou et al. [46] formulated a multi-objective optimization for FL based on decomposition and meta-DRL, aimed at adapting to the dynamic satellite-terrestrial environments and achieving efficient uploading and aggregation. Empowered by digital twin technology, a multi-agent deep deterministic policy gradient (DDPG) approach [47] was designed in [48] to optimize user association, cache placement and power control for maximum energy efficiency. Despite these commendable efforts, research on efficient resource allocation that can follow dynamic environments is still weak. The remarkable success of attention-based models, e.g., Transformer [49] and InstructGPT [50], has sparked a surge in research interest and brought new opportunities for addressing the two aforementioned

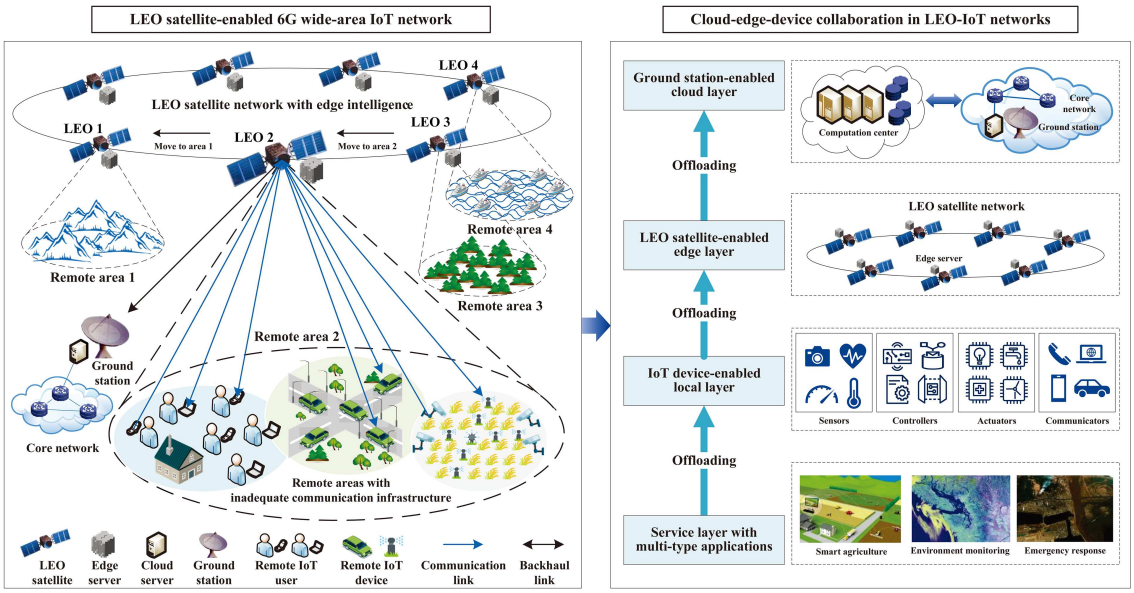

Figure 1 (Color online) Illustration of the 6G wide-area edge intelligence scenario.

technical challenges. The attention mechanism selectively follows crucial or dynamic information and automatically learns relevance. Inspired by this, our research focuses on attention-assisted DRL methods to optimize resource allocation decisions in alignment with the dynamic properties of task arrival and satellite motion.

3 System model

3.1 Network architecture

As shown in Figure 1, we consider a 6G wide-area edge intelligence scenario to support remote IoT services. In this scenario, the LEO satellite constellation serves remote natural areas to enable seamless coverage, large-capacity communication and low-latency computation services. Furthermore, we introduce a "cloud-edge-device" architecture for task offloading and computing. This architecture consists of an IoT device-enabled layer, an LEO satellite-enabled edge layer, a ground station-enabled cloud layer and a service layer with multi-type applications, as detailed below.

• **IoT device-enabled local layer.** This layer is composed of numerous IoT devices such as sensors, controllers, actuators, and transceivers. These devices perform compute-intensive and latency-sensitive task, such as data analytics, image processing, and real-time monitoring.

• **LEO satellite-enabled edge layer.** This layer is comprised of LEO satellites equipped with flexible payloads and edge servers, providing both network access and task computing capabilities.

• **Ground station-enabled cloud layer.** The ground stations of LEO satellites are connected to the core network via wired links, serving as the cloud center to provide powerful data processing capabilities.

• **Service layer with multi-type applications.** The "cloud-edge-device" architecture can be implemented to support a variety of services, such as intelligent agriculture and environment monitoring.

In this paper, we focus on a dynamic LEO satellite-supported remote IoT network (LEO-IoT) with M visible LEO satellites and N IoT devices, denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{N} = \{1, 2, \ldots, N\}$, respectively. The environment characteristics of the LEO-IoT network, e.g., channel condition and network topology, are assumed to be constant within the same time slot but dynamically change across different time slots [13]. Consequently, the system is based on a time-slotted model $\mathcal{T} = \{1, 2, \ldots, T\}$ with equal duration. Furthermore, we use a tuple $(K_{n,i}, T_{n,i}^{\max})$ to characterize the arriving task i on device n, where $K_{n,i}$ is the data size and $T_{n,i}^{\max}$ represents the maximum tolerable delay. We assume that the dynamic task arrival of IoT devices follows a Poisson process with rate λ. Therefore, the probability that task i

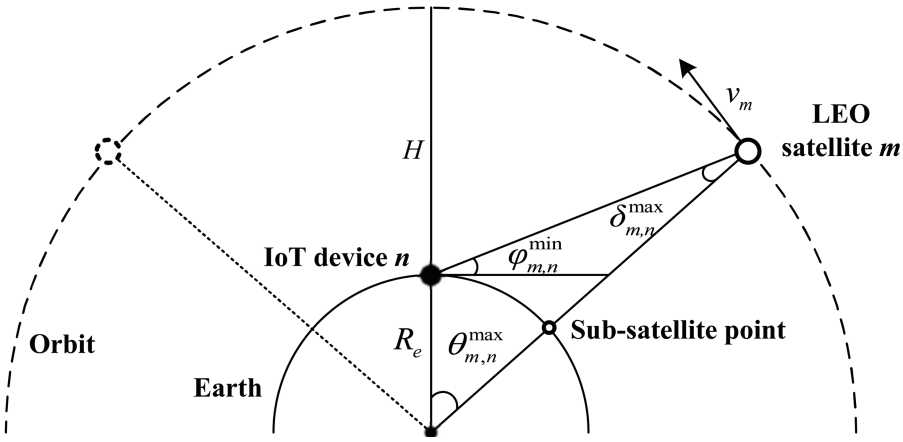

Figure 2 Diagram of the geometric relationship between the LEO satellite and the IoT device.

with data size $K_{n,i}$ arrives at IoT device n during time slot t is given by

$$P\left\{\left\lfloor\frac{K_{n,i}(t)}{k}\right\rfloor\right\}=\frac{\mathrm{e}^{-\lambda}\cdot(\lambda)^{\lfloor\frac{K_{n,i}(t)}{k}\rfloor}}{(\lfloor\frac{K_{n,i}(t)}{k}\rfloor)!}. \tag{1}$$

Here, k denotes the data size of each data packet. Furthermore, the queue backlog on the device at time t can be represented as

$$Q_n^{t+1}=\max\left\{Q_n^t-Z_n^t,0\right\}+\sum_i K_{n,i}^t, \tag{2}$$

where Q_n^t and Q_n^{t+1} represent the queue backlogs of device n at times t and $t+1$, respectively. Z_n^t is the amount of task data processed within time t, which is related to the task offloading and resource allocation strategies.

3.2 Coverage model

Figure 2 illustrates the geometric relationship [51] between an IoT device and the LEO satellite. It is crucial to account for the impact of orbital dynamics because the coverage model of LEO satellites is inherently constrained by both limited coverage times and changing coverage areas.

Coverage time. The coverage time for an IoT device by an LEO satellite is closely related to the elevation angle. As the elevation angle decreases, the link attenuation gradually increases to the critical point where the satellite cannot provide adequate signal strength to that device, resulting in the failure of data transmission. Referred to [25], the minimum elevation angle $\varphi_{m,n}^{\min}$ between device n and satellite m can be calculated based on the geometric relationship. Then, the maximum available communication time can be expressed as [26]

$$T_{m,n}^{\max}=\frac{2(R_e+H)\left[\arccos\left(\frac{R_e}{R_e+H}\cos\varphi_{m,n}^{\min}\right)-\varphi_{m,n}^{\min}\right]}{v_m}. \tag{3}$$

Here, R_e represents the radius of the Earth, and H is the altitude of the LEO satellite orbit. v_m represents the orbital velocity of LEO satellite m.

Coverage area. As stipulated in [51], the satellite coverage is constrained by the minimum elevation angle $\varphi_{m,n}^{\min}$. According to the law of sines, the maximum radius of the coverage area is given as

$$E_{m,n}^{\max}=R_e\left\{\frac{\pi}{2}-\varphi_{m,n}^{\min}-\arcsin\left[\frac{R_e\sin\left(\frac{\pi}{2}+\varphi_{m,n}^{\min}\right)}{R_e+H}\right]\right\}. \tag{4}$$

However, the coverage radius is not an effective measure for assessing the coverage relationship between ground terminals and LEO satellites. In this paper, we utilize another pivotal parameter from the coverage model, namely the maximum coverage angle, also known as the half-depression angle, to characterize the

visibility of satellites to users. The maximum coverage angle between LEO satellite m and IoT device n is represented in (5). If $\delta_{m,n} < \delta_{m,n}^{\max}$ is satisfied, then the satellite is considered visible to the device.

$$\delta_{m,n}^{\max} = \frac{\pi}{2} - \varphi_{m,n}^{\min} - \theta_{m,n}^{\max}. \tag{5}$$

3.3 Communication model

In the LEO satellite-supported remote IoT network, tasks from the device can be uploaded to the LEO satellite through the uplink, or forwarded by the LEO satellite and then transmitted to the remote cloud via the downlink. Here, the Rician fading model is adopted to characterize the time-varying satellite-terrestrial links. Then, we can derive the achievable transmission rates of the uplink and downlink based on the analysis of interference issues. The transmission rate affects the data size transmitted to the edge and cloud servers, and modeling it will facilitate the analysis of transmission delay.

Channel gain. Referred to [31], the channel gain can be represented as

$$g_{m,n} = G_T G_R L_{m,n} |h_{m,n}|^2, \tag{6}$$

where G_T and G_R represent the transmitting antenna gain and the receiving antenna gain, respectively. $L_{m,n} = (4\pi f d_{m,n}/\mathrm{c})^{-2}$ denotes the free space loss, where c is the velocity of light, f is the carrier frequency, and $d_{m,n}$ represents the distance between satellite m and device n. In addition, $h_{m,n}$ is the Rician fading coefficient, and $|h_{m,n}|^2$ follows non-central chi-square distribution with factor κ.

Date rate. Regardless of the uplink or downlink, orthogonal frequency division multiple access (OFDMA) is employed to manage frequency resources, with all IoT devices sharing these resources to transmit task data. Given this full-frequency reuse scheme, the issue of co-channel interference cannot be ignored. According to the Shannon-Hartley theorem, the achievable data rate can be calculated by

$$R_{m,n} = B_{m,n}^{\mathrm{Tot}} \log_2 \left(1 + \frac{p_{m,n} g_{m,n}}{\sum p_{m,n} g_{m',n'} + N_0 B_{m,n}^{\mathrm{Tot}}}\right). \tag{7}$$

$B_{m,n}^{\mathrm{Tot}}$ denotes the total allocated bandwidth, which can be further expressed as $B_{m,n}^{\mathrm{Tot}} = \sum_{c\in\mathcal{C}} w_{n,c} B_c^{\mathrm{Sub}}$. Here, the sub-channel set is defined as $\mathcal{C} = \{1, 2, \ldots, C\}$. Moreover, $w_{n,c}$ indicates the sub-channel allocation variable, while B_c^{Sub} represents the sub-channel bandwidth. Additionally, $p_{m,n}$ is the transmit power, and N_0 is the power spectral density of the additive white Gaussian noise (AWGN). For clarity, let $R_{n,m}^{\mathrm{UL}}$ denote the uplink rate and $R_{m,c}^{\mathrm{DL}}$ denote the downlink rate in the following analysis.

3.4 Computation model

Referred to [26], we also assume that all IoT tasks can only be offloaded to a single destination, either in whole or in part. Here, we define the offloading vector of task i from device n to both the edge and cloud servers as $\boldsymbol{\alpha}_{n,i} = \left[\alpha_{n,i}^E[1], \alpha_{n,i}^E[2], \ldots, \alpha_{n,i}^E[M], \alpha_{n,i}^C\right]$. Moreover, the offloading variables are binary, where 1 indicates that offloading is confirmed, and 0 indicates otherwise. In this paper, we consider the following three computation offloading schemes.

Case 1: local computing. The computation task is executed on the local device, i.e., $\forall m,\ \alpha_{n,i}^E[m] = 0,\ \alpha_{n,i}^C = 0$. If the computing capability of device n is denoted as $F_l(n,i)$ (in CPU cycles/s), the local computation latency is

$$T_l(n,i) = \frac{\varpi K_{n,i}}{F_l(n,i)}, \tag{8}$$

where ϖ is the computational density, indicating the number of CPU cycles required to process one bit.

Case 2: local computing and edge computing. In this mode, aside from local computing, part of the task will be offloaded to an edge server for computation, i.e., $\exists m,\ \alpha_{n,i}^E[m] = 1,\ \alpha_{n,i}^C = 0$. As a result, both the transmission and computation processes need to be considered in edge computing, so the overall latency can be expressed as

$$T_e(n,i) = \sum_{m\in\mathcal{M}} \left(\frac{\varpi \varepsilon_{n,i}^E[m] K_{n,i}}{F_m(n,i)} + \frac{\varepsilon_{n,i}^E[m] K_{n,i}}{R_{n,m}^{\mathrm{UL}}} + \varrho_{n,m}\right). \tag{9}$$

Here, $\varepsilon_{n,i}^{E}[m]$ represents the offloading ratio of task i from device n to visible satellite m. Additionally, $F_m(n,i)$ is the computing capability of satellite m, and $\varrho_{n,m}$ is the propagation delay.

Case 3: local computing and cloud computing. In this case, the task can be split into a local computing portion and a cloud computing portion, i.e., $\forall m,\ \alpha_{n,i}^{E}[m]=0,\ \alpha_{n,i}^{C}=1$. Considering both the uplink and downlink communication processes through the LEO satellite relay, the total latency for cloud processing is given by

$$T_c(n,i)=\frac{\varpi\varepsilon_{n,i}^{C}K_{n,i}}{F_c(n,i)}+\frac{\varepsilon_{n,i}^{C}K_{n,i}}{R_{n,m}^{\mathrm{UL}}}+\frac{\varepsilon_{n,i}^{C}K_{n,i}}{R_{m,c}^{\mathrm{DL}}}+\varrho_{n,m}+\varrho_{m,c}. \tag{10}$$

Here, $\varepsilon_{n,i}^{C}$ is the cloud offloading ratio, and $F_c(n,i)$ is the cloud computing capability allocated to device n to perform task i. The subscript m in $R_{n,m}^{\mathrm{UL}}$, $R_{m,c}^{\mathrm{DL}}$ $\varrho_{n,m}$, and $\varrho_{m,c}$ indicates the relay satellite.

To enhance computational efficiency, parallel processing is employed for tasks [31]. Therefore, the ultimate delay for device n to compute task i can be determined by

$$T_{n,i}=\max\left\{T_l(n,i),\ T_e(n,i),\ T_c(n,i)\right\}. \tag{11}$$

4 Problem formulation and transformation

In this section, we first introduce the problem constraints and formulate the joint optimization problem of cross-region task offloading and cross-domain resource allocation as a mixed integer non-linear programming (MINLP) problem. Then, we elaborate on the constraints of the optimization problem and transform it into an MDP-based decision problem.

4.1 Problem constraints

Here is a detailed explanation of delay constraints in coverage, communication, and computation models.

(1) Delay constraints of coverage. As explained in the coverage model, the coverage time is defined as $\mathcal{T}_{m,n}^{\mathrm{cov}}=\left\{t|T_{m,n}^{\mathrm{com}}\leqslant t\leqslant T_{m,n}^{\mathrm{com}}+T_{m,n}^{\max}\right\}$, where $T_{m,n}^{\mathrm{com}}$ represents the start time when device n is just able to communicate with satellite m. Let $\mathcal{T}_{n,i}^{\mathrm{tol}}=\left\{t|T_{n,i}^{\mathrm{arr}}\leqslant t\leqslant T_{n,i}^{\mathrm{arr}}+T_{n,i}^{\max}\right\}$ denote the tolerance time, where $T_{n,i}^{\mathrm{arr}}$ is the time that task i arrives at device n. According to the relationship between tolerance time $\mathcal{T}_{n,i}^{\mathrm{tol}}$ and coverage time $\mathcal{T}_{m,n}^{\mathrm{cov}}$, the coverage cases can be described as follows.

(i) Always visible. If the tolerance time and coverage time follow $\mathcal{T}_{n,i}^{\mathrm{tol}}\in\mathcal{T}_{m,n}^{\mathrm{cov}}$, then the delay constraint in this case is expressed as

$$\mathbb{I}\left\{\sum_{m\in\mathcal{M}}\alpha_{n,i}^{E}[m]\leqslant 1\right\}\mathbb{I}\left\{\alpha_{n,i}^{C}\leqslant 1\right\}T_{n,i}\leqslant T_{n,i}^{\max}. \tag{12}$$

(ii) Always invisible. This case is defined by $\mathcal{T}_{n,i}^{\mathrm{tol}}\cap\mathcal{T}_{m,n}^{\mathrm{cov}}=\emptyset$, then the delay constraint is written as

$$\mathbb{I}\left\{\sum_{m\in\mathcal{M}}\alpha_{n,i}^{E}[m]=0\right\}\mathbb{I}\left\{\alpha_{n,i}^{C}=0\right\}T_l^{\mathrm{L}}(n,i)\leqslant T_{n,i}^{\max}. \tag{13}$$

(iii) Intermittently visible. This coverage scenario is formulated as $\mathcal{T}_{n,i}^{\mathrm{tol}}\cap\mathcal{T}_{m,n}^{\mathrm{cov}}\neq\emptyset,\mathcal{T}_{n,i}^{\mathrm{tol}}-\mathcal{T}_{m,n}^{\mathrm{cov}}\neq\emptyset,\mathcal{T}_{m,n}^{\mathrm{cov}}-\mathcal{T}_{n,i}^{\mathrm{tol}}\neq\emptyset$, and the delay is constrained by

$$\mathbb{I}\left\{\sum_{m\in\mathcal{M}}\alpha_{n,i}^{E}[m]\leqslant 1\right\}\mathbb{I}\left\{\alpha_{n,i}^{C}\leqslant 1\right\}\left\{\mathcal{T}_{n,i}^{\mathrm{tol}}\cap\mathcal{T}_{m,n}^{\mathrm{cov}}\right\}+\mathbb{I}\left\{\sum_{m\in\mathcal{M}}\alpha_{n,i}^{E}[m]=0\right\}\mathbb{I}\left\{\alpha_{n,i}^{C}=0\right\}T_l^{\mathrm{L}}(n,i)\leqslant T_{n,i}^{\max}. \tag{14}$$

Note that $\mathbb{I}\{x\}$ is a characteristic function, where $\mathbb{I}\{x\}=1$ if event x is true and $\mathbb{I}\{x\}=0$ otherwise.

(2) Delay constraints of communication. Little's law[1)] states that the average time a task spends in the communication system is equal to the long-term average amount of task data divided by the average

1) Little's law is a theorem by John Little which states that the long-term average number of customers in a stationary system is equal to the long-term average effective arrival rate multiplied by the average time that a customer spends in the system.

effective arrival rate. Here, the task arrival rate refers to the uplink throughput $R_{m,n}^{\mathrm{UL}}$ and downlink throughput $R_{m,n}^{\mathrm{DL}}$ of egress to computing destinations. According to Little's law, the transmission delay is constrained by

$$\frac{1}{NI}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}\left\{\sum_{m\in\mathcal{M}}\left(\frac{\alpha_{n,i}^{E}[m]\varepsilon_{n,i}^{E}[m]\,K_{n,i}}{R_{m,n}^{\mathrm{UL}}}\right)\right\}+\frac{1}{NI}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}\left\{\frac{\alpha_{n,i}^{C}\varepsilon_{n,i}^{C}K_{n,i}}{R_{m,n}^{\mathrm{UL}}}+\frac{\alpha_{n,i}^{C}\varepsilon_{n,i}^{C}K_{n,i}}{R_{m,n}^{\mathrm{DL}}}\right\}\leqslant T_{r}^{\max}, \tag{15}$$

where $\sum_{m\in\mathcal{M}}\alpha_{n,i}^{E}[m]+\alpha_{n,i}^{C}=1$ is defined. In addition, $T_{r}^{\max}$ represents the bound on transmission delay, implying that the task will be discarded if this bound is exceeded.

(3) Delay constraints of computation. Following Little's law, the average computation delay is equal to the ratio of the average data size to the average computation rate. If $T_{f}^{\max}$ denotes the upper bound of the computation delay, then the constraint on computation delay is

$$\frac{1}{NI}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}\left\{\frac{\left(1-\varepsilon_{n,i}^{E}-\varepsilon_{n,i}^{C}\right)K_{n,i}\varpi}{F_{l}(n,i)}\right\}+\frac{1}{NI}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}\left\{\sum_{m\in\mathcal{M}}\frac{\alpha_{n,i}^{E}[m]\varepsilon_{n,i}^{E}[m]\,K_{n,i}\varpi}{F_{m}(n,i)}+\frac{\alpha_{n,i}^{C}\varepsilon_{n,i}^{C}K_{n,i}\varpi}{F_{c}(n,i)}\right\}\leqslant T_{f}^{\max}. \tag{16}$$

Here, $\sum_{m\in\mathcal{M}}\alpha_{n,i}^{E}[m]+\alpha_{n,i}^{C}=1$ is also defined.

4.2 Problem formulation

This paper aims to provide adaptive task offloading and efficient resource allocation schemes to support delay-sensitive and compute-intensive services in the dynamic LEO-IoT network. The objective is to minimize the task delay of all IoT devices by jointly optimizing multi-layer task offloading and multi-dimensional resource scheduling. Mathematically, the optimization problem is formulated as

$$\mathbf{P0:}\quad \min_{\left\{\boldsymbol{\alpha}_{n,i},\boldsymbol{\varepsilon}_{n,i},\boldsymbol{w}_{n,i}^{\mathrm{UL}},\boldsymbol{w}_{n,i}^{\mathrm{DL}},\boldsymbol{F}_{n,i},G_{m}^{T},p_{n}^{\mathrm{Tx}}\right\}}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}T_{n,i}, \tag{17a}$$

$$\text{s.t.}\quad \text{C1: } \alpha_{n,i}^{E}[m]\in\{0,1\},\alpha_{n,i}^{C}\in\{0,1\},\ \forall i,m,n, \tag{17b}$$

$$\text{C2: } \varepsilon_{n,i}^{E}[m]\in[0,1],\varepsilon_{n,i}^{C}\in[0,1],\ \forall i,m,n, \tag{17c}$$

$$\text{C3: } \sum_{m\in\mathcal{M}}\alpha_{n,i}^{E}[m]+\alpha_{n,i}^{C}=1,\ \forall i,m,n, \tag{17d}$$

$$\text{C4: } \sum_{m\in\mathcal{M}}\varepsilon_{n,i}^{E}[m]+\varepsilon_{n,i}^{C}\leqslant 1,\ \forall i,m,n, \tag{17e}$$

$$\text{C5: } \sum_{i\in\mathcal{I}}F_{l}(n,i)\leqslant F_{l}^{\max},\ \forall n, \tag{17f}$$

$$\text{C6: } \sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}F_{m}(n,i)\leqslant F_{m}^{\max},\ \forall m, \tag{17g}$$

$$\text{C7: } \sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}F_{c}(n,i)\leqslant F_{c}^{\max}, \tag{17h}$$

$$\text{C8: } 0\leqslant\sum_{c\in\mathcal{C}}w_{n,c}^{\mathrm{UL}}B_{c}^{\mathrm{UL}}\leqslant B_{\mathrm{tot}}^{\mathrm{UL}},\ \forall n, \tag{17i}$$

$$\text{C9: } 0\leqslant\sum_{c\in\mathcal{C}}w_{n,c}^{\mathrm{DL}}B_{c}^{\mathrm{DL}}\leqslant B_{\mathrm{tot}}^{\mathrm{DL}},\ \forall n, \tag{17j}$$

$$\text{C10: } 0\leqslant p_{n}^{\mathrm{Tx}}\leqslant P_{n}^{\max},\ \forall i,n, \tag{17k}$$

$$\text{C11: } \left[G_{m}^{T}\right]+\left[p_{m}^{\mathrm{Tx}}\right]\leqslant\left[\mathrm{EIRP}_{\mathrm{up}}\right],\ \forall m, \tag{17l}$$

$$\text{C12: } (18),(19),\text{and }(20),\ \forall i,m,n. \tag{17m}$$

Here, vectors $\boldsymbol{\alpha}_{n,i}=[\alpha_{n,i}^{E}[1],\ldots,\alpha_{n,i}^{E}[m],\ldots,\alpha_{n,i}^{E}[M],\alpha_{n,i}^{C}]$ and $\boldsymbol{\varepsilon}_{n,i}=[\varepsilon_{n,i}^{E}[1],\ldots,\varepsilon_{n,i}^{E}[m],\ldots,\varepsilon_{n,i}^{E}[M],\varepsilon_{n,i}^{C}]$ denote the offloading destination and offloading ratio, respectively. Vector $\boldsymbol{w}_{n,i}^{\mathrm{UL}}=[w_{n,1}^{\mathrm{UL}},\ldots,w_{n,c}^{\mathrm{UL}},\ldots,w_{n,C}^{\mathrm{UL}}]$ and vector $\boldsymbol{w}_{n,i}^{\mathrm{DL}}=[w_{n,1}^{\mathrm{DL}},\ldots,w_{n,c}^{\mathrm{DL}},\ldots,w_{n,C}^{\mathrm{DL}}]$ represent the sub-channel allocation for the uplink and

downlink, respectively. Vector $\boldsymbol{F}_{n,i} = [F_l(n,i), F_m(n,i), F_c(n,i)]$ denotes the device-side, edge-side and cloud-side computing resources allocated to task i of device n. Constraints C1 and C3 indicate that each IoT device can only choose one server for task offloading and computing. Constraints C2 and C4 imply that both binary offloading and partial offloading are supported. Constraint C5 ensures that the total amount of local computing resources allocated to tasks on the same device does not exceed the maximum computing capacity of that device. Constraint C6 specifies that the edge computing resources assigned to tasks across all devices must not surpass the maximum computing capacity of the satellite server. Also, the constraint on the cloud computing server is defined in C7. The communication resource allocations of uplink and downlink are constrained in C8 and C9, guaranteeing that the resources assigned to IoT devices will not exceed the total available bandwidth. Constraint C10 stipulates that the transmit power of the IoT device must be non-negative and less than the maximum value. Constraint C11 ensures that the sum of the decibel values of the satellite transmit power and transmit antenna gain cannot be allowed to exceed the upper limit of the equivalent isotropic radiated power (EIRP). Consistent with the aforementioned, C12 defines the delay constraints of coverage, communication and computation.

4.3 Problem transformation with MDP

As mentioned above, the formulated problem **P0** is an MINLP problem, characterized by the NP-hard nature due to the intricate couplings among multiple optimization variables. Consequently, it is nontrivial to directly solve the optimal solution in highly dynamic LEO-IoT networks. To address this challenge, RL has emerged as an effective approach. This method is based on MDP theory to accommodate dynamic environments and tackle long-term stochastic decision-making problems in a slot-by-slot manner. In general, the MDP model is characterized as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ with state $\mathcal{S}$, action $\mathcal{A}$, transition probability $\mathcal{P}$, reward $\mathcal{R}$ and discount factor γ. The detailed definitions are described as follows.

State. The state $\boldsymbol{s}_t \in \mathcal{S}$ is comprised of device-related information for all IoT devices and task-related information for all tasks on each device. Specifically, the collection of device-related information is denoted as $\mathcal{S}_t^D = \{\boldsymbol{s}_t^1, \ldots, \boldsymbol{s}_t^N\}$, and the information for device n is represented as

$$\boldsymbol{s}_t^n = \left[n, m, \phi_n^{\mathrm{Lon}}, \phi_n^{\mathrm{Lat}}, d_{m,n}, Q_n\right], \ \forall n \in \mathcal{N}, \tag{18}$$

which includes device index n, satellite index m, longitude ϕ_n^{Lon}, latitude ϕ_n^{Lat}, the distance $d_{m,n}$ between device n and satellite m, and the total queue length Q_n of all tasks on the device.

In addition, the collection of task-related information is denoted as $\mathcal{S}_t^T = \{\boldsymbol{s}_t^{1,1}, \ldots, \boldsymbol{s}_t^{n,i}, \ldots, \boldsymbol{s}_t^{N,I}\}$, and the information about task i of device n is

$$\boldsymbol{s}_t^{n,i} = \left[K_{n,i}, T_{n,i}^{\mathrm{arr}}, T_{n,i}^{\max}\right], \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \tag{19}$$

which contains the data size $K_{n,i}$, arrival time $T_{n,i}^{\mathrm{arr}}$ and maximum tolerable delay $T_{n,i}^{\max}$.

Action. The action $\boldsymbol{a}_t \in \mathcal{A}$ is defined as $\boldsymbol{a}_t = \{\boldsymbol{a}_t^{1,1}, \ldots, \boldsymbol{a}_t^{n,i}, \ldots, \boldsymbol{a}_t^{N,I}\}$. Specifically, $\boldsymbol{a}_t^{n,i} = \{\boldsymbol{o}_t^{n,i}, \boldsymbol{z}_t^{n,i}\}$ encompasses the multi-layer task offloading decision and the multi-dimensional resource allocation scheme for each task. The action of multi-layer task offloading is expressed as

$$\boldsymbol{o}_t^{n,i} = [\boldsymbol{\alpha}_{n,i}, \boldsymbol{\varepsilon}_{n,i}], \ \forall i \in \mathcal{I}, \ \forall n \in \mathcal{N}, \tag{20}$$

which contains the offloading destination $\boldsymbol{\alpha}_{n,i}$ and offloading ratio $\boldsymbol{\varepsilon}_{n,i}$. Moreover, the action for multi-dimensional resource allocation is given by

$$\boldsymbol{z}_t^{n,i} = \left[\boldsymbol{w}_{n,i}^{\mathrm{UL}}, \boldsymbol{w}_{n,i}^{\mathrm{DL}}, G_m^T, p_n^{\mathrm{Tx}}, \boldsymbol{F}_{n,i}\right], \forall i \in \mathcal{I}, \forall n \in \mathcal{N}. \tag{21}$$

Here, $\left\{\boldsymbol{w}_{n,i}^{\mathrm{UL}}, \boldsymbol{w}_{n,i}^{\mathrm{DL}}\right\}$ represents the communication resource allocation for the uplink and downlink. Additionally, $\left\{G_m^T, p_n^{\mathrm{Tx}}\right\}$ defines the adjustment for satellite coverage, which is achieved through transmit gain and transmit power. $\boldsymbol{F}_{n,i} = \{F_l(n,i), F_m(n,i), F_c(n,i)\}$ corresponds to the computational resource allocation for IoT devices, edge satellites and cloud servers.

Transition probability. Following the execution of action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$, the network environment will undergo a transition to the next state with probability $p\left(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t\right)$. Moreover, the transition probability can be decoupled into the action-induced transition $p\left(\hat{\boldsymbol{s}}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t\right)$ and the dynamics-induced transition $p\left(\boldsymbol{s}_{t+1}|\hat{\boldsymbol{s}}_{t+1}\right)$. On the one side, the state transition resulting from environmental dynamics

(e.g., satellite orbiting and device mobility) is manifested as changes in $\left\{m, \phi_n^{\mathrm{Lon}}, \phi_n^{\mathrm{Lat}}, d_{m,n}\right\}$. On the other side, the state transition caused by action execution is represented by

$$\begin{aligned} p\left(\hat{\boldsymbol{s}}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t\right) &= \prod_{n\in\mathcal{N}} p\left\{Q_n(t+1)|Q_n(t), \boldsymbol{a}_t\right\} \\ &= \prod_{n\in\mathcal{N}}\prod_{i\in\mathcal{I}} p\left\{K_{n,i}(t+1)|K_{n,i}(t), \boldsymbol{a}_t\right\}. \end{aligned} \tag{22}$$

Reward. An immediate reward, denoted as $r_t \in \mathcal{R}$, will be fed back after the execution of action $\boldsymbol{a}_t$. This reward serves as a means to assess the advantages or disadvantages associated with the performed action. Given that **P0** is a minimization problem, we characterize the immediate reward as the negation of the delay cost incurred at time slot t, expressed as

$$r_t = -\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}} T_{n,i}(t). \tag{23}$$

Furthermore, the expected cumulative reward is defined as

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{j=0}^{\infty} \gamma^k r_{t+j+1}, \tag{24}$$

where $\gamma \in [0, 1]$ is the discount factor representing the contribution to the long-term goal. In accordance with the origin problem **P0**, the optimization objective is reformulated to maximize the expected cumulative reward by optimizing the joint task offloading and resource scheduling strategy. The reformulated problem can be expressed as

$$\mathbf{P1:} \quad \max_{\boldsymbol{\pi}} \quad \mathbb{E}[R_t|\boldsymbol{\pi}], \tag{25a}$$

$$\text{s.t.} \quad \text{(21b)–(21m)}, \tag{25b}$$

where $\boldsymbol{\pi} \sim p(\boldsymbol{a}_t|\boldsymbol{s}_t)$ represents the policy. Problem **P0** is transformed into problem **P1** to find the optimal policy π^* that maximizes the expected cumulative reward, i.e., minimizes the system delay.

5 Dynamics-aware STA-PPO algorithm

The original problem **P0** has been transformed into **P1**, and an MDP-based RL algorithm, called the spatio-temporal attention-based proximal policy optimization (STA-PPO), is detailed in this section. The proposed algorithm is designed with the architecture of multi-actor and mixed-critic. The multi-actor network is coupled with temporal attention to make time-varying decisions, and the mixed-critic network utilizes spatial attention to evaluate topology-related values.

5.1 Temporal attention-based multi-actor network

As depicted in Figure 3, the multi-actor network is a parallel structure, with each network specializing in actions on satellite coverage, task offloading, data transmission and task computation. For each actor network, we employ a hybrid neural network architecture that combines the temporal attention mechanism (TAM) with an FCN. This design is customized to extract features from sequential data and discover historical regularities in task arrivals and movement trajectories of devices, facilitating the algorithm in making time-sensitive decisions for task offloading and resource allocation.

Define $\boldsymbol{H}_t \in \mathbb{R}^{d\times\tilde{t}}$ as the hidden representation of the past $\tilde{t}$ slots, expressed as

$$\boldsymbol{H}_t = \left[\boldsymbol{h}_t, \boldsymbol{h}_{t-1}, \ldots, \boldsymbol{h}_{t-\tilde{t}+1}\right]. \tag{26}$$

Since tasks at different times in the backlog queue are not equally important, the TAM module is designed to assign temporal weights for aggregating the hidden representations of the recent $\tilde{t}$ slots. The temporal attention weights $\boldsymbol{\eta}_t \in \mathbb{R}^{d\times\tilde{t}}$ are calculated by

$$\boldsymbol{\eta}_t = \mathrm{SoftMax}(\boldsymbol{W}_{\mathrm{TAM}}^{\mathrm{T}}\mathrm{Tanh}(\boldsymbol{H}_t)), \tag{27}$$

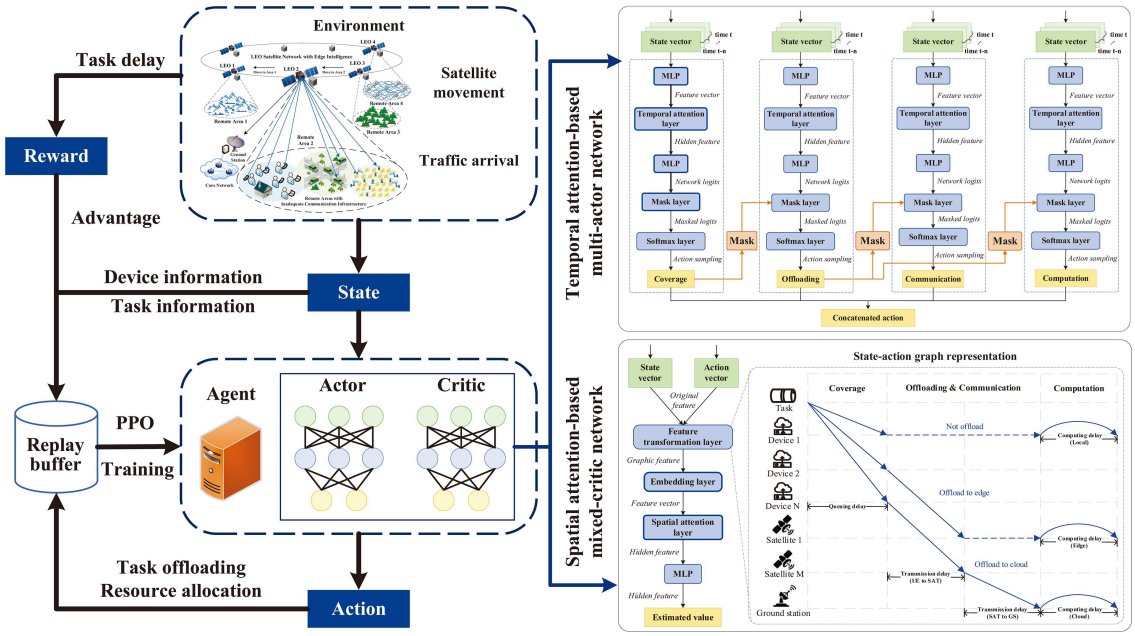

Figure 3 (Color online) Overview of the STA-PPO algorithm.

where $\boldsymbol{W}_{\text{TAM}} \in \mathbb{R}^{d\times d}$ represents the learnable parameters of the TAM module. The SoftMax operator is denoted as SoftMax$(\cdot)$, and the Tanh activation function is written as Tanh$(\cdot)$. Thus, the hidden representation with weighted aggregation is

$$\boldsymbol{h}_t' = \langle \boldsymbol{\eta}_t, \boldsymbol{H}_t \rangle, \tag{28}$$

where $\langle \boldsymbol{\eta}_t, \boldsymbol{H}_t \rangle$ is the inner product. Then, the action probability distribution after the fully connected layer and SoftMax operator can be expressed as

$$\boldsymbol{\pi}(\boldsymbol{a}_t|\boldsymbol{s}_t) = \text{SoftMax}(\text{FCN}(\boldsymbol{h}_t'; \boldsymbol{W}_{\text{FCN}}, \boldsymbol{B}_{\text{FCN}})), \tag{29}$$

and $\boldsymbol{W}_{\text{FCN}}$ and $\boldsymbol{B}_{\text{FCN}}$ are the weight and bias, respectively.

Furthermore, the optimization process involves multiple actions, including (i) $\boldsymbol{a}_t^c = \{G_m^T, p_n^{\text{Tx}}\}$: adjusting beam gain and transmit power for seamless satellite coverage; (ii) $\boldsymbol{a}_t^o = \{\boldsymbol{\alpha}_{n,i}, \boldsymbol{\varepsilon}_{n,i}\}$: optimizing offloading destination and offloading ratio for efficient task offloading; (iii) $\boldsymbol{a}_t^w = \{\boldsymbol{w}_{n,i}^{\text{UL}}, \boldsymbol{w}_{n,i}^{\text{DL}}\}$: optimizing uplink and downlink bandwidth for high-capacity task transmission; and (iv) $\boldsymbol{a}_t^e = \{F_l(n,i), F_m(n,i), F_c(n,i)\}$: optimizing local-side, edge-side and cloud-side computing resource allocation for low-delay task computation. Consequently, the action strategy of the multi-actor network can be decomposed into

$$\boldsymbol{\pi}(\boldsymbol{a}_t|\boldsymbol{s}_t) = \boldsymbol{\pi}(\boldsymbol{a}_t^c|\boldsymbol{s}_t)\boldsymbol{\pi}(\boldsymbol{a}_t^o|\boldsymbol{s}_t)\boldsymbol{\pi}(\boldsymbol{a}_t^w|\boldsymbol{s}_t)\boldsymbol{\pi}(\boldsymbol{a}_t^e|\boldsymbol{s}_t). \tag{30}$$

However, these actions are interrelated. It is imperative to ensure that the task offloading process occurs within the coverage area of LEO satellites. Once the task offloading decisions are determined, it also imposes constraints on the strategies for communication and computation resource allocation. Violating the established relationship will result in invalid actions. For instance, allocating extra communication resources to a locally executed task is considered invalid. To address this, we mask out invalid actions, achieved by

$$p_i = \frac{v_i e^{x_i}}{\sum_i v_i e^{x_i}}, \tag{31}$$

where v_i is a binary variable with values 1 for valid actions and 0 for invalid actions. The softmax computation with masks ensures that the probabilities computed by the model will be non-zero for valid actions. On this basis, the sequential decision-making process for multi-actions with respect to coverage, offloading, communication and computation can be given by

$$\{G_m^T(t), p_n^{\text{Tx}}(t)\} \sim \boldsymbol{\pi}(\boldsymbol{a}_t^c|\boldsymbol{s}_t), \tag{32a}$$

$$\{\boldsymbol{\alpha}_{n,i}(t), \boldsymbol{\varepsilon}_{n,i}(t)\} \sim \ \boldsymbol{\pi}(\boldsymbol{a}_t^o | \boldsymbol{s}_t, \boldsymbol{\chi}_t^c), \tag{32b}$$

$$\left\{\boldsymbol{w}_{n,i}^{\mathrm{UL}}(t), \boldsymbol{w}_{n,i}^{\mathrm{DL}}(t)\right\} \sim \ \boldsymbol{\pi}(\boldsymbol{a}_t^w | \boldsymbol{s}_t, \boldsymbol{\chi}_t^o), \tag{32c}$$

$$\left\{F_{n,i}^l(t), F_{n,i}^m(t), F_{n,i}^c(t)\right\} \sim \ \boldsymbol{\pi}(\boldsymbol{a}_t^e | \boldsymbol{s}_t, \boldsymbol{\chi}_t^o). \tag{32d}$$

Here, $\boldsymbol{\chi}_t^c$ is the mask for task offloading determined by the coverage action, and $\boldsymbol{\chi}_t^o$ is the mask for communication and computation resource allocation generated by the offloading action.

5.2 Spatial attention-based mixed-critic network

As shown in Figure 3, the actions generated by the multi-actor network, combined with the state, are transformed into a graph structure known as the state-action graph (SAG), which describes the connectivity between satellites and devices, as well as the computing schemes and offloading paths for tasks. The SAG is characterized by

$$\mathrm{SAG} = \{\{V_n, V_m, V_c\}, \{E(V_n, V_m), E(V_m, V_c)\}\}. \tag{33}$$

Here, $\boldsymbol{V} = \{V_n, V_m, V_c\}$ is the set of devices, satellites, and cloud nodes. Moreover, the set of edges includes the uplink from the device to the satellite $E\langle V_n, V_m\rangle$ and the downlink from the satellite to the ground station $E\langle V_m, V_c\rangle$. The adjacency matrix of SAG describes the connectivity between device, satellite, and cloud nodes determined by the task offloading strategies, which is expressed as

$$a_{x,y}(t) = \begin{cases} 1, & \text{if } \alpha_{x,i}^y(t) = 1, \ \exists i \in \mathcal{I}, \\ 0, & \text{if } \alpha_{x,i}^y(t) = 0, \ \forall i \in \mathcal{I}. \end{cases} \tag{34}$$

The SAG provides a structured view of the connectivity between satellites and devices, as well as task offloading strategies, enabling the use of sophisticated graph-based techniques to effectively address the challenges posed by the highly dynamic network topology in LEO-IoT networks.

For the mixed-critic network, we similarly design a hybrid neural network that integrates the spatial attention mechanism (SAM) with an FCN. This design enables the algorithm to extract features from structural data and identify topological relationships in satellite movement and task offloading, thereby conducting topology-related assessments regarding task offloading and resource allocation in dynamic LEO satellite networks. We define $\boldsymbol{G}_t \in \mathbb{R}^{d\times(N+M+1)}$ as the hidden representation of nodes,

$$\boldsymbol{G}_t = \left[\boldsymbol{g}_t^1, \ldots, \boldsymbol{g}_t^N, \boldsymbol{g}_t^1, \ldots, \boldsymbol{g}_t^M, \boldsymbol{g}_t^c\right], \tag{35}$$

where $\boldsymbol{g}_t^n, n \in \mathcal{N}$, $\boldsymbol{g}_t^m, m \in \mathcal{M}$, and $\boldsymbol{g}_t^c$ are the embedding vectors of device node n, satellite node m and cloud node c, respectively. Then, we utilize the SAM module to calculate spatial weights for aggregating the embedding representations of neighboring nodes. To illustrate, for example, the spatial attention weight $\boldsymbol{\zeta}_t^{n,m} \in \mathbb{R}$ for device node n and satellite node m is given by

$$\boldsymbol{\zeta}_t^{n,m} = \mathrm{SoftMax}(\mathrm{LeakyReLU}(\boldsymbol{W}_{\mathrm{SAM}}^{\mathrm{T}}[\boldsymbol{W}_{\mathrm{FA}}\boldsymbol{g}_t^n || \boldsymbol{W}_{\mathrm{FA}}\boldsymbol{g}_t^m])). \tag{36}$$

Here, $||$ denotes the concatenation operator, and LeakyReLU$(\cdot)$ is the LeakyReLU activation function. Besides, $\boldsymbol{W}_{\mathrm{FA}} \in \mathbb{R}^{d'\times d}$ represents the shared parameters for feature argument, and $\boldsymbol{W}_{\mathrm{SAM}}^{\mathrm{T}} \in \mathbb{R}^{2d'}$ is the learnable parameters of the SAM module. As an example, the hidden representation with weighted aggregation of the device node n is

$$\boldsymbol{g}_t'[n] = \mathrm{Sigmoid}\left(\sum_{m\in\mathcal{M}} \boldsymbol{\zeta}_t^{n,m}\boldsymbol{W}_{\mathrm{FA}}\boldsymbol{g}_t^m + \boldsymbol{\zeta}_t^{n,c}\boldsymbol{W}_{\mathrm{FA}}\boldsymbol{g}_t^c\right). \tag{37}$$

Sigmoid$(\cdot)$ is the Sigmoid activation function. To enhance the stability of the learning process, a multi-head attention mechanism is incorporated. Then, the estimated value after the fully connected layer can be obtained by

$$\mathcal{V}^{\boldsymbol{\pi}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathrm{FCN}\left(\mathop{\Big\|}_{o=1}^{O} \boldsymbol{g}_t'[o]; \boldsymbol{W}_{\mathrm{FCN}}, \boldsymbol{B}_{\mathrm{FCN}}\right), \tag{38}$$

where $g_t'[o]$ is the node representation calculated by the oth attention head.

Algorithm 1 STA-PPO algorithm for task offloading and resource allocation.

Input: Information on tasks, devices, and satellites.
Output: Optimal offloading and computing strategy.
1: Initialize the parameters θ of the multi-actor network;
2: Initialize the parameters ξ of the mixed-critic network;
3: **for** epoch $i \leftarrow 0$ **to** I **do**
4: Initialize the experience replay buffer;
5: Reset the gradients of the networks;
6: **for** episode $n \leftarrow 0$ **to** N **do**
7: Initialize the states and accumulated rewards;
8: **for** step $t \leftarrow 0$ **to** T **do**
9: Observe state $\boldsymbol{s}_t$ in (18) and (19);
10: **for** multi-actor network **do**
11: Select $G_m^T(t)$ for coverage;
12: Select $\boldsymbol{\alpha}_{n,i}(t)$, $\boldsymbol{\varepsilon}_{n,i}(t)$ for offloading;
13: Select $\boldsymbol{w}_{n,i}^{\mathrm{UL}}(t)$, $\boldsymbol{w}_{n,i}^{\mathrm{DL}}(t)$ and $p_n^{\mathrm{Tx}}(t)$ for task transmission;
14: Select $F_{n,i}^l(t)$, $F_{n,i}^m(t)$ and $F_{n,i}^c(t)$ for task computation;
15: **end for**
16: **for** mixed-critic network **do**
17: Convert the vectors of state $\boldsymbol{s}_t$ and action $\boldsymbol{a}_t$ into graph representations;
18: Estimate the value V_t for selecting action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$;
19: **end for**
20: Concatenate the outputs of the multi-actor network as action $\boldsymbol{a}_t$ and execute it;
21: Get reward r_t in (23) and next state $\boldsymbol{s}_{t+1}$ from the environment;
22: Store $\{\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1}, V_t\}$ in the experience replay buffer;
23: **end for**
24: **end for**
25: Sample a mini-batch data from the buffer;
26: **for** iteration $l \leftarrow 0$ to L **do**
27: **if** $\mathrm{KL}[\boldsymbol{\pi}_{\theta_{\mathrm{old}}}, \boldsymbol{\pi}_\theta] \leqslant \mathrm{KL}_{\mathrm{target}}$ **then**
28: Calculate the multi-actor network loss in (39) and update the parameter θ;
29: **end if**
30: Calculate the mixed-critic network loss in (40) and update the parameter ξ;
31: **end for**
32: **end for**

5.3 Algorithm pseudocode

The proposed STA-PPO algorithm for task offloading and resource allocation in dynamic LEO-IoT networks is summarized in Algorithm 1. The algorithm begins by initializing network parameters and the experience replay buffer (lines 1 to 5). Then, experience data, including states, actions, rewards, next states, and estimated values, is stored in the replay buffer by operating over multiple episodes with multiple steps (lines 6 to 24). At each step, the TAM-based multi-actor network selects actions for satellite coverage, task offloading, transmission and computation (lines 10 to 15). Simultaneously, the SAM-based mixed-critic network estimates the value of chosen actions (lines 16 to 19). Finally, the algorithm iteratively optimizes the strategies for task offloading and resource allocation through parameter updates guided by Kullback-Leibler divergence thresholds (lines 25 to 31). Specifically, the multi-actor network incorporates a clip function to restrict the scope of policy updates, and the objective function is

$$\mathcal{L}_{\theta_{\mathrm{old}}}^{\mathrm{Actor}}(\theta) = \mathop{\mathbb{E}}_{\boldsymbol{\tau}\sim\boldsymbol{\pi}_{\theta_{\mathrm{old}}}} \left[\sum_{t=0}^{T} \min \left[\rho_t(\theta) A_t^{\boldsymbol{\pi}_{\theta_{\mathrm{old}}}}, \mathrm{clip}\left(\rho_t(\theta), 1 \pm \epsilon\right) A_t^{\boldsymbol{\pi}_{\theta_{\mathrm{old}}}} \right] \right]. \tag{39}$$

Here, θ represents the parameter of the multi-actor network, and $\rho_t(\theta) = \boldsymbol{\pi}_\theta(a_t|s_t)/\boldsymbol{\pi}_{\theta_{\mathrm{old}}}(a_t|s_t)$ denotes importance sampling. The samples will be reweighted based on the relative probabilities of the old and new policies, eliminating bias in the update process. The clip function $\mathrm{clip}\left(\rho_{\mathrm{t}}(\theta), 1-\epsilon, 1+\epsilon\right)$ ensures that the new policy will not deviate too far from the old policy, preventing the algorithm from falling into a poor policy area due to overshooting. $A_t^{\boldsymbol{\pi}_\theta}$ is the advantage function, providing information about the current action relative to the average performance of the policy. It is estimated by a generalized advantage estimation (GAE) method. Additionally, the mixed-critic network aims to minimize the error of the value function $\mathcal{V}^{\boldsymbol{\pi}_\theta}$, and the objective function is given by

$$\mathcal{L}_{\boldsymbol{\pi}_\theta}^{\mathrm{Critic}}(\xi) = \frac{1}{2}\left[r_t + \mathcal{V}_\xi^{\boldsymbol{\pi}_\theta}(\boldsymbol{s}_{t+1}) - \mathcal{V}_\xi^{\boldsymbol{\pi}_\theta}(\boldsymbol{s}_t) \right]^2, \tag{40}$$

where ξ is the parameter of the mixed-critic network. It is worth noting that $r_t + \mathcal{V}_\xi^{\boldsymbol{\pi}_\theta}(\boldsymbol{s}_{t+1})$ is the target value, while $\mathcal{V}_\xi^{\boldsymbol{\pi}_\theta}(\boldsymbol{s}_t)$ is the estimated value. In summary, the proposed STA-PPO algorithm is a

Table 1 Satellite TLE data at 00:00:00 on March 1, 2023.

STARLINK-2291								
1	47862U	21021C	23080.27443524	0.00019376	00000+0	13155−2	0	9999
2	47862	53.0513	89.7447	0001758	81.7401	278.3787	15.06415959112402	–
STARLINK-3737								
1	52114U	22029AC	23079.80535556	0.00000789	00000+0	67661−4	0	9999
2	52114	53.2133	269.3593	0001062	82.6153	277.4959	15.08840264	57017
STARLINK-5359								
1	54864U	22177AW	23079.74356850	−0.00000635	00000+0	−31933−4	0	9993
2	54864	43.0023	255.4031	0001577	266.2526	93.8136	15.02540217	13900

sophisticated approach that takes into account temporal dependencies and spatial correlations, making it a potent tool for cross-region task offloading and cross-domain resource allocation in highly dynamic LEO-IoT networks.

6 Simulation results and performance analysis

6.1 Simulation setup

Our simulations are conducted in a remote scenario where the Starlink constellation provides task offloading and computing services for IoT devices. The simulated time spans from March 1, 2023, 00:00:00 to March 1, 2023, 00:05:00. During the simulation period, all IoT devices are uniformly distributed within a circular area with a 380 km radius, centered at (1.5°N, 0°E). In addition, the Starlink constellation comprises 1584 LEO satellites with an inclination of 53°, distributed across 72 orbital planes at 550 km [52]. This configuration ensures seamless coverage of the target area with at least 3 satellites [17]. For instance, at 00:00:00 on March 1, 2023, the satellites covering the target area include "STARLINK-2291", "STARLINK-3737" and "STARLINK-5359". The two-line element (TLE)[2)] sets for these satellites are detailed in Table 1. Within the coverage region, task arrivals on IoT devices follow a Poisson distribution with $\lambda = 5$, and the task size is set at 0.5 Mb. The computation capabilities of IoT devices, LEO satellites, and cloud servers are 0.4, 10, and 50 Gcycles/s, respectively. For the uplink of device-satellite transmission, the transmit power of IoT devices is in the range of [0.2, 3.0] W. Moreover, the available bandwidth for uplink communication is set as 20 MHz, equal to the downlink. For the downlink of satellite-cloud transmission, the transmit power of LEO satellites is fixed at 10 W, and the transmit gain is adjustable within the range of [34.5, 44.5] dBi. The other key simulation parameters are summarized in Table 2.

6.2 Performance comparison with benchmark algorithms

To demonstrate the effectiveness of our proposed STA-PPO algorithm, we compare it with benchmark algorithms for task offloading and resource allocation (TORA). The benchmark algorithms are described as follows.

• **RAND-based TORA algorithm.** Random sampling is performed within the feasible decision space [8], and the TORA policy remains static with no updates.

• **PG-based TORA algorithm.** The FCN-based policy network parameterizes the TORA policy and provides feasible decisions, while the policy gradient (PG) method [22] is applied for policy optimization.

• **PPO-based TORA algorithm.** Compared to the proposed approach, the PPO algorithm [53] is based on the FCNs and does not incorporate the attention mechanism.

(a) System delay. Figure 4(a) compares the system delay of all TORA algorithms. Except for the RAND-based TORA algorithm, the curves of other algorithms decrease with the increase in the number of training epochs and gradually saturate at their corresponding optimal values. The RAND-based TORA algorithm maintains an average delay of 755.78 ms without any noticeable downward trend, indicating the worst performance. The PG-based and PPO-based TORA algorithms exhibit similar performance, both with an average delay of approximately 330 ms. However, the curve of the PG algorithm shows a larger fluctuation, particularly in the interval of 150–170 epochs. In contrast, the curve of the PPO

2) The orbital data for Starlink is derived from CelesTrak.

Table 2 Simulation parameters.

Parameters	Values
Number of IoT devices (N), visible edge satellites (M), and cloud servers	20, 3, 1
Task size and computational density ($K_{n,i}$, ϖ)	0.5 Mb, 1000 cycles/bit
Maximum delay constraints of tasks ($T_{n,i}^{\max}$, $T_r^{\max}$, $T_f^{\max}$)	50, 30, 20 ms
Minimum elevation angle and maximum geocentric angle ($\varphi_{m,n}^{\min}$, $\theta_{m,n}^{\max}$)	10°, 25°
Satellite orbital velocity and maximum communication time (v_m, $T_{m,n}^{\max}$)	7.5 km/s, 810 s
Maximum computational capabilities ($F_l^{\max}$, $F_m^{\max}$, $F_c^{\max}$)	0.4, 10, 50 Gcycles/s
Maximum available bandwidths for uplink and downlink ($B_{\text{tot}}^{\text{UL}}$, $B_{\text{tot}}^{\text{DL}}$)	20, 20 MHz
Number of sub-channels for uplink and downlink (C)	12, 12
Carrier frequencies for uplink and downlink (f^{UL}, f^{DL})	30, 20 GHz
Transmit powers of satellite and device (p_m^{Tx}, p_n^{Tx})	10, [0.2, 3.0] W
Transmitter gains of satellite and device (G_m^T, G_n^T)	[34.5, 44.5], 30 dBi
Receiver gains of satellite and device (G_m^R, G_n^R)	43, 30 dBi
Power spectral density of AWGN (N_0)	−199.6 dBw/MHz
Training epochs and episode length	300, 100
Buffer size and discount factor ($\mathcal{D}$, γ)	100, 0.99
Clip ratio and target KL ($\rho(\theta)$, $\text{KL}_{\text{target}}$)	0.2, 0.01
Hidden layer size and activation function	[128, 256, 512, 256, 128], Tanh function
Learning rates of the actor and critic networks ($\varsigma_\theta^{\text{Actor}}$, $\varsigma_\xi^{\text{Critic}}$)	3e−3, 1e−3
Optimizer and update frequency	Adam optimizer, 50 iterations/epoch

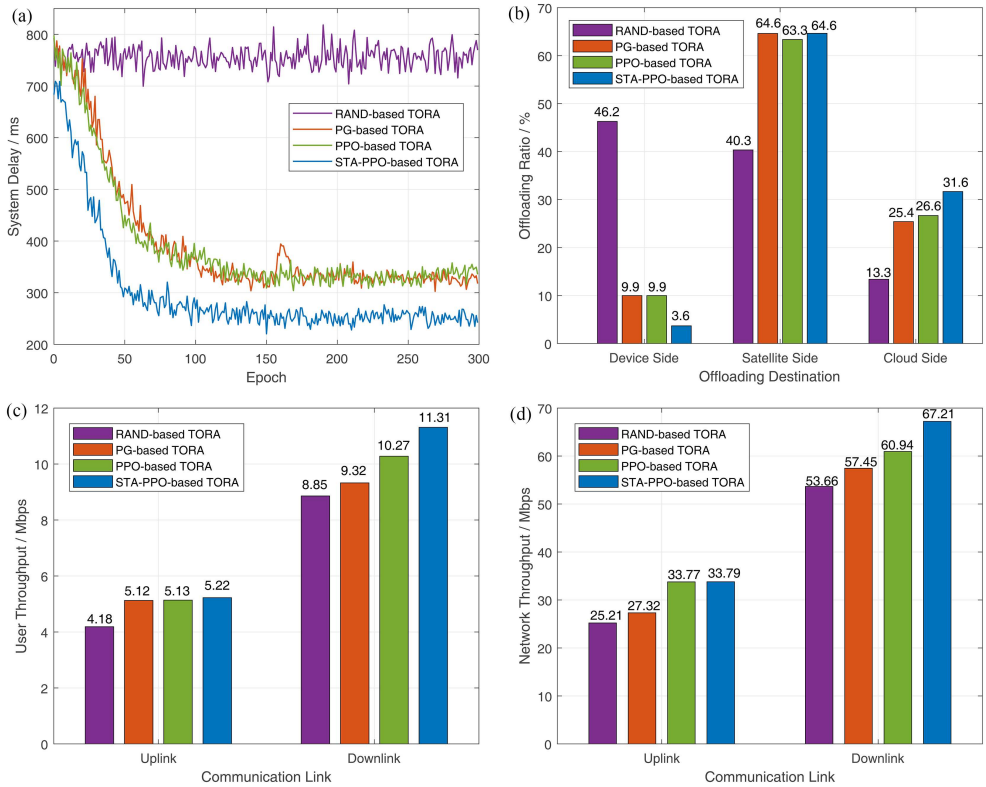

Figure 4 (Color online) Performance comparison of different TORA algorithms. (a) System delay; (b) offloading ratio; (c) link throughput; (d) network throughput.

algorithm is more stable, as it eliminates the variances introduced by the policy model through the critic model. Also, the STA-PPO-based TORA algorithm utilizes the actor-critic architecture, combined with spatio-temporal dynamics analysis, to achieve the lowest delay. Compared with RAND, PG and PPO algorithms, the system delay of our proposed algorithm is reduced by 66.38%, 23.55% and 23.39%, respectively. This superiority demonstrates that the design of the STA mechanism has a beneficial effect on reducing system delay.

(b) Offloading ratio. Figure 4(b) provides a comparison of offloading ratios to the devise side, edge side, and cloud side. The RAND-based TORA algorithm exhibits a significant gap in the local computing ratio, being 36.2986%, 36.2995% and 42.5988% higher than other algorithms. Combined with the poorest results of the RAND algorithm, it can be speculated that an excessively high proportion of local computation is not advisable. This is attributed to the fact that the computing capacity of the device is significantly smaller than that of the edge and cloud servers. Regarding the edge offloading ratio, our proposed STA-PPO algorithm performs comparably to the PG and PPO algorithms, surpassing the RAND algorithm by over 20%. Furthermore, it is crucial to highlight that the success of our algorithm lies in less local offloading and more cloud offloading. This suggests that in scenarios with a limited number of IoT devices (e.g., $K = 20$), the delay incurred by long-distance communication in the cloud computing scheme is relatively lower than that of long-time computation in the local computing scheme.

(c) Link throughput. In Figure 4(c), we compare the uplink throughput and downlink throughput of the above TORA algorithms. We have noticed that the sum of cloud and edge offloading ratios in our proposed scheme surpasses that of the RAND, PG and PPO algorithms by 79.48%, 6.89%, and 7.01%, respectively. This result indicates that our algorithm may have more preemption and even multiplexing of transmission resources to support large-volume task offloading. Contrary to the expectation, the proposed algorithm achieves the highest uplink throughput, suggesting that the spatial-temporal awareness embedded in our STA-PPO algorithm effectively guides the TORA strategy. Additionally, the downlink throughput of the STA-PPO algorithm outperforms RAND, PG and PPO algorithms by 27.79%, 21.35% and 10.12%, respectively. This superiority demonstrates that our proposed algorithm maximizes the usage of available resources through effective resource management and allocation.

(d) Network throughput. In Figure 4(d), we compare the network throughput of the four TORA algorithms. Overall, the STA-PPO algorithm outperforms the other algorithms in both uplink and downlink throughput. For the uplink, although the network throughput of the STA-PPO algorithm is similar to that of the PPO algorithm, this is primarily because all algorithms have a high offloading ratio to the satellite side, resulting in minimal differences. However, the STA-PPO algorithm still shows a noticeable gap in performance compared to the PG algorithm and the random strategy. This is because the STA-PPO algorithm demonstrates more effective spatio-temporal awareness in resource management, providing an advantage in resource contention. For the downlink, the network throughput of the STA-PPO algorithm significantly surpasses that of the RAND, PG, and PPO algorithms by 27.79%, 21.35%, and 10.12%, respectively. This indicates that the STA-PPO algorithm maximizes resource utilization through efficient resource allocation strategies, further enhancing support for large-volume task transmissions.

Overall, the STA-PPO algorithm demonstrates clear advantages in reducing system delay, optimizing offloading ratios, and maximizing link throughput and network throughput, making it the most efficient and effective approach among the evaluated TORA algorithms.

6.3 Performance comparison with spatio-temporal variations

In this subsection, we compare the impact of task arrival and satellite coverage to verify the adaptability to spatio-temporal changes.

(1) Task arrival. We present the distribution of task sizes with task arrival rates $\lambda = 1$ to $\lambda = 8$ to illustrate the temporal dynamics in the LEO-IoT network. As depicted in Figure 5, the depth of the color indicates the task size of the device. The distribution of task sizes exhibits large-scale variations for different task arrival rates, while small-scale variations for the same task arrival rate. In Table 3, we provide a performance comparison of four TORA algorithms concerning system delay, uplink throughput, and downlink throughput under different task arrival rates. As the task arrival rate increases, the system delay of all TORA algorithms shows an upward trend. However, the STA-PPO algorithm consistently demonstrates the lowest system delay across various task arrival rates. Compared with RAND, PG and PPO algorithms, the system delay of our proposed algorithm is reduced by 63.85%, 18.41% and 14.98%, respectively. In addition, the STA-PPO algorithm surpasses other algorithms in uplink and downlink throughput due to efficient resource utilization. In particular, the throughput of our proposed algorithm is improved by 16.45%–19.37% in the case of $\lambda = 7$ and $\lambda = 8$, showcasing its ability to meet high-load task demands. To sum up, the STA-PPO-based TORA method outperforms other methods for all task arrival rates, indicating that the TAM effectively adapts to the temporal-dynamic characteristics in LEO-IoT networks.

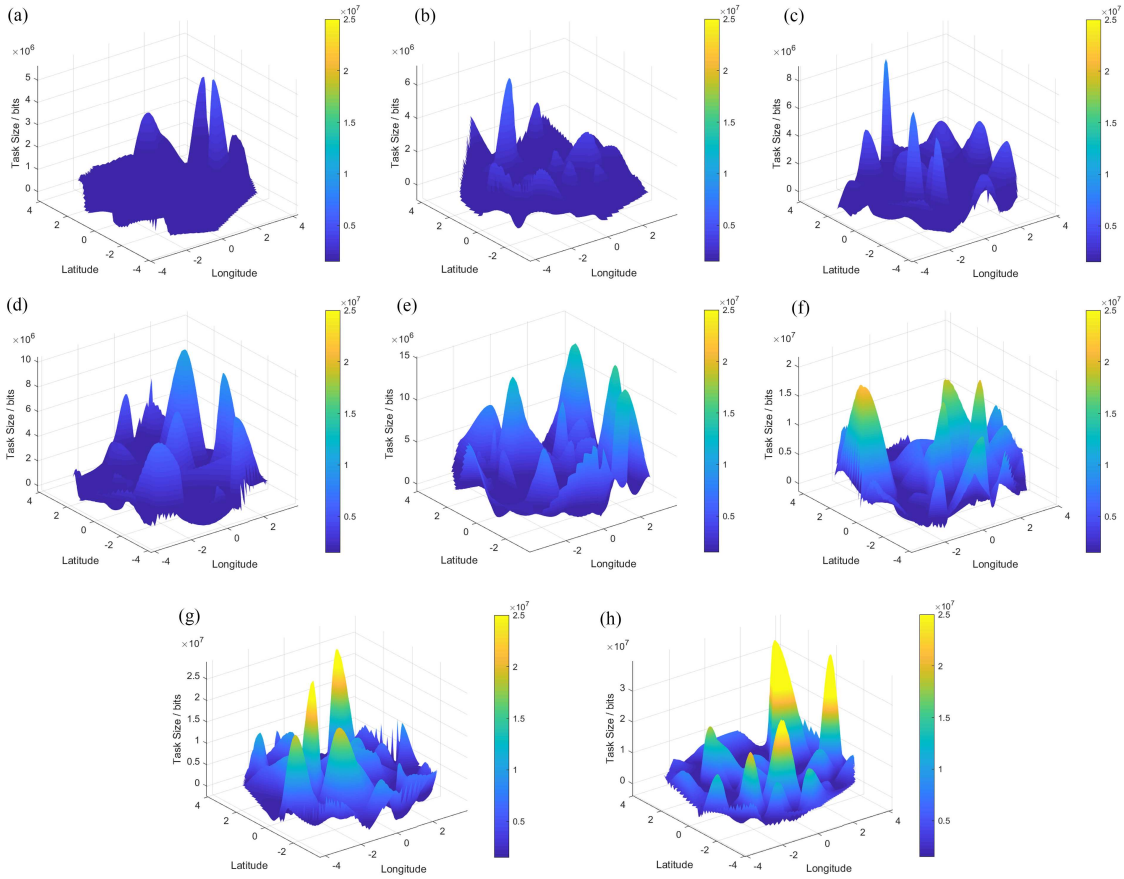

Figure 5 (Color online) Distribution of task sizes with different task arrival rates. (a) $\lambda = 1$; (b) $\lambda = 2$; (c) $\lambda = 3$; (d) $\lambda = 4$; (e) $\lambda = 5$; (f) $\lambda = 6$; (g) $\lambda = 7$; (h) $\lambda = 8$.

Table 3 Performance comparison with different task arrival rates.

Performance	Algorithm	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 4$	$\lambda = 5$	$\lambda = 6$	$\lambda = 7$	$\lambda = 8$
System delay (ms)	RAND	522.91	649.84	705.24	736.20	755.78	769.31	779.27	865.62
	PG	256.69	266.25	301.88	312.90	332.59	342.60	351.99	395.35
	PPO	246.23	262.22	281.63	288.34	331.66	340.95	346.21	359.66
	STA-PPO	193.68	234.77	238.98	245.85	254.06	294.24	300.08	331.72
Uplink throughput (Mbps/device)	RAND	4.187	4.187	4.187	4.187	4.187	4.187	4.187	4.187
	PG	5.399	5.384	5.273	5.207	5.124	4.620	4.347	4.291
	PPO	5.390	5.378	5.277	5.197	5.137	5.092	4.851	4.745
	STA-PPO	5.979	5.976	5.651	5.595	5.224	5.156	5.069	4.849
Downlink throughput (Mbps/device)	RAND	8.858	8.858	8.858	8.858	8.858	8.858	8.858	8.858
	PG	10.475	10.288	9.724	9.535	9.326	9.225	9.086	8.914
	PPO	10.900	10.468	10.381	10.325	10.279	9.937	9.202	9.076
	STA-PPO	12.039	11.973	11.895	11.886	11.310	11.274	11.268	10.968

(2) Satellite coverage. We demonstrate the beam coverage of visible satellites at different times to elucidate the spatial dynamics in the LEO-IoT network. As shown in Figure 6, the antenna radiation patterns corresponding to multiple beam units are depicted as colored areas, and the colored areas visually convey the beam gains at different amplitudes. Here, we emphasize that both the coverage area and beam gain of visible satellites exhibit sustained variations over time. Table 4 provides a comparison of four TORA algorithms regarding system delay, uplink throughput, and downlink throughput under different satellite coverage situations. In terms of system delay, the STA-PPO algorithm outperforms other methods in all satellite coverage areas, achieving an average reduction of 62.03%, 16.19% and 1.55%, respectively. Worst of all, the RAND-based TORA method shows the maximum system delays, suggesting that its stochastic nature leads to poor performance. Moreover, it is obvious that the system

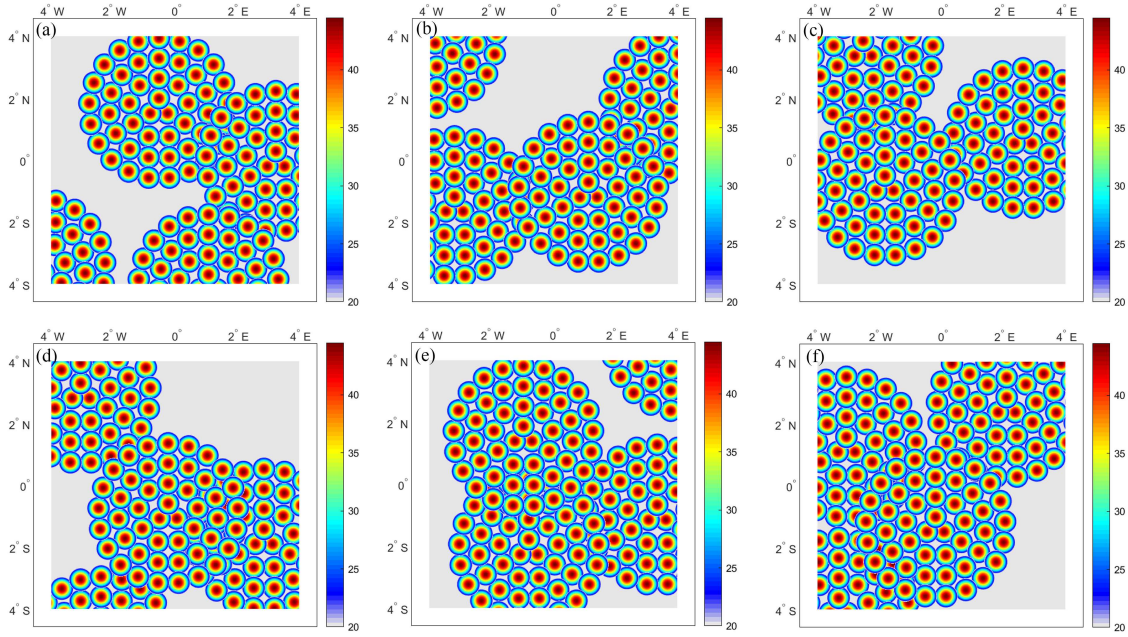

Figure 6 (Color online) Beam coverage of visible satellites at different time points. (a) March 1, 2023, 00:00:00; (b) March 15, 2023, 00:00:00; (c) March 30, 2023, 00:00:00; (d) April 1, 2023, 00:00:00; (e) April 15, 2023, 00:00:00; (f) April 30, 2023, 00:00:00.

Table 4 Performance comparison with different satellite coverage.

Performance	Algorithm	2023/3/1	2023/3/15	2023/3/30	2023/4/1	2023/4/15	2023/4/30
System delay (ms)	RAND	755.78	747.83	724.89	756.15	732.57	728.73
	PG	332.59	311.68	299.57	336.54	410.11	333.32
	PPO	331.66	274.61	282.75	291.68	386.77	329.19
	STA-PPO	254.06	252.12	280.92	281.89	304.48	312.52
Uplink throughput (Mbps/device)	RAND	4.187	5.076	4.517	4.956	4.431	5.314
	PG	5.124	5.137	5.301	4.916	4.689	5.453
	PPO	5.224	5.333	5.369	5.495	4.702	5.808
	STA-PPO	5.557	5.585	5.579	5.848	5.122	6.578
Downlink throughput (Mbps/device)	RAND	8.858	11.052	9.811	10.498	9.525	11.786
	PG	9.326	11.376	11.121	11.047	9.715	12.043
	PPO	10.279	12.166	11.203	11.321	9.899	11.754
	STA-PPO	11.310	11.949	11.425	11.756	11.098	12.394

delays dated 2023/4/15 and 2023/4/30 are relatively higher. Combining the beam coverage in Figures 6(c) and (e), we speculate that the larger beam overlap area may cause severe co-channel interference, resulting in a decline in offloading efficiency. Concerning link throughput, the proposed algorithm also performs the best with an average improvement of 17.46%, 10.32% and 6.27%. In summary, the STA-PPO algorithm outperforms other TORA methods for all satellite coverage patterns, which indicates that the SAM embedded in our proposed algorithm can better accommodate the spatial-dynamic characteristics in LEO-IoT networks.

In summary, the STA-PPO algorithm demonstrates superior performance across a variety of task arrival rates and satellite coverage scenarios. Through intelligent task offloading and efficient resource allocation, the STA-PPO algorithm not only minimizes task latency but also maximizes link throughput. This clearly proves that the spatio-temporal attention mechanism of the STA-PPO algorithm effectively adapts to the dynamic characteristics of LEO-IoT networks. These capabilities make it an ideal solution for optimizing performance in LEO-IoT environments, showcasing its advanced adaptability and efficiency in handling the complexities of satellite-based IoT systems.

7 Conclusion

In this paper, we consider the cloud-edge-device collaboration in 6G wide-area IoT networks, which leverages the capabilities of LEO satellites to deliver near-user edge computing and facilitate access to far-user cloud computing. In 6G wide-area edge intelligence scenarios supported by LEO satellites, we study the joint optimization of cross-region task offloading and cross-domain resource allocation, and transform it into a sequential decision-making problem. Considering the highly dynamic environment, we proposed an STA-PPO algorithm as the solution to accommodate the temporal-dynamic characteristics of task arrival and the spatial-dynamic characteristics of network topology. The proposed STA-PPO algorithm has been validated to consistently outperform other TORA methods, exhibiting the lowest system delay and the highest link throughput. For future research, the authors will consider the seamless integration of space, aerial, and ground segments, and the collaborative "cloud-edge-device" framework will also be extended with other advanced techniques, e.g., UAV trajectory optimization and HAP placement optimization. In addition, innovative algorithms based on AI will be further studied to enable 6G networks with intelligent air interface and network architecture.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2023YFB2904700) and National Natural Science Foundation of China (Grant Nos. 62372357, 62201419).

References

1 Babbar H, Rani S, Bouachir O, et al. From massive IoT toward IoE: evolution of energy efficient autonomous wireless networks. IEEE Comm Stand Mag, 2023, 7: 32–39
2 Xu W, Huang Y, Wang W, et al. Toward ubiquitous and intelligent 6G networks: from architecture to technology. Sci China Inf Sci, 2023, 66: 130300
3 Al Homssi B, Al-Hourani A, Wang K, et al. Next generation mega satellite networks for access equality: opportunities, challenges, and performance. IEEE Commun Mag, 2022, 60: 18–24
4 Centenaro M, Costa C E, Granelli F, et al. A survey on technologies, standards and open challenges in satellite IoT. IEEE Commun Surv Tut, 2021, 23: 1693–1720
5 You X H, Wang C X, Huang J, et al. Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts. Sci China Inf Sci, 2021, 64: 110301
6 Ni S, Liu J, Sheng M, et al. Joint optimization of user association and resource allocation in cache-enabled terrestrial-satellite integrating network. Sci China Inf Sci, 2021, 64: 182306
7 Sheng M, Zhou D, Bai W, et al. Coverage enhancement for 6G satellite-terrestrial integrated networks: performance metrics, constellation configuration and resource allocation. Sci China Inf Sci, 2023, 66: 130303
8 Wang D, Bai Y, Huang G, et al. Cache-aided MEC for IoT: resource allocation using deep graph reinforcement learning. IEEE Int Things J, 2023, 10: 11486–11496
9 Wang S, Li Q. Satellite computing: vision and challenges. IEEE Int Things J, 2023, 10: 22514–22529
10 Zhou D, Sheng M, Li J, et al. Aerospace integrated networks innovation for empowering 6G: a survey and future challenges. IEEE Commun Surv Tut, 2023, 25: 975–1019
11 Wang D, Li B, Song B, et al. Dual-driven resource management for sustainable computing in the blockchain-supported digital twin IoT. IEEE Int Things J, 2023, 10: 6549–6560
12 Zhai Z, Wu Q, Yu S, et al. FedLEO: an offloading-assisted decentralized federated learning framework for low earth orbit satellite networks. IEEE Trans Mobile Comput, 2024, 23: 5260–5279
13 Liao H, Zhou Z, Zhao X, et al. Learning-based queue-aware task offloading and resource allocation for space-air-ground-integrated power IoT. IEEE Int Things J, 2021, 8: 5250–5263
14 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 1998
15 Lu K, Liu H, Zeng L, et al. Applications and prospects of artificial intelligence in covert satellite communication: a review. Sci China Inf Sci, 2023, 66: 121301
16 Chen J H, Kuo W C, Liao W. SpaceEdge: optimizing service latency and sustainability for space-centric task offloading in LEO satellite networks. IEEE Trans Wireless Commun, 2024, 23: 15435–15446
17 Han D, Ye Q, Peng H, et al. Two-timescale learning-based task offloading for remote IoT in integrated satellite-terrestrial networks. IEEE Int Things J, 2023, 10: 10131–10145
18 Zhang R, Xiong K, Lu Y, et al. Energy efficiency maximization in RIS-assisted SWIPT networks with RSMA: a PPO-based approach. IEEE J Sel Areas Commun, 2023, 41: 1413–1430
19 Devlin J, Chang M, Lee K, et al. BERT: pre-training of deep bidirectional Transformers for language understanding. 2018. ArXiv:1810.04805
20 Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020. 1877–1901
21 He N, Yang S, Li F, et al. Leveraging deep reinforcement learning with attention mechanism for virtual network function placement and routing. IEEE Trans Parallel Distrib Syst, 2023, 34: 1186–1201
22 Mao S, He S, Wu J. Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing. IEEE Syst J, 2021, 15: 3992–4002
23 Cheng X, Lyu F, Quan W, et al. Space/aerial-assisted computing offloading for IoT applications: a learning-based approach. IEEE J Sel Areas Commun, 2019, 37: 1117–1129
24 Wang B, Feng T, Huang D. A joint computation offloading and resource allocation strategy for LEO satellite edge computing system. In: Proceedings of the IEEE 20th International Conference on Communication Technology (ICCT), Nanning, 2020. 649–655
25 Wang Y, Yang J, Guo X, et al. A game-theoretic approach to computation offloading in satellite edge computing. IEEE Access, 2020, 8: 12510–12520
26 Tang Q, Fei Z, Li B, et al. Computation offloading in LEO satellite networks with hybrid cloud and edge computing. IEEE Int Things J, 2021, 8: 9164–9176
27 Ding C, Wang J B, Zhang H, et al. Joint optimization of transmission and computation resources for satellite and high altitude platform assisted edge computing. IEEE Trans Wireless Commun, 2022, 21: 1362–1377
28 Song Z, Hao Y, Liu Y, et al. Energy-efficient multiaccess edge computing for terrestrial-satellite Internet of Things. IEEE Int Things J, 2021, 8: 14202–14218

29 Zhang H, Liu R, Kaushik A, et al. Satellite edge computing with collaborative computation offloading: an intelligent deep deterministic policy gradient approach. IEEE Int Things J, 2023, 10: 9092–9107
30 Chen Y, Ai B, Niu Y, et al. Energy-constrained computation offloading in space-air-ground integrated networks using distributionally robust optimization. IEEE Trans Veh Technol, 2021, 70: 12113–12125
31 Chen Q, Meng W, Quek T Q S, et al. Multi-tier hybrid offloading for computation-aware IoT applications in civil aircraft-augmented SAGIN. IEEE J Sel Areas Commun, 2023, 41: 399–417
32 Jia M, Zhang L, Wu J, et al. Joint computing and communication resource allocation for edge computing towards huge LEO networks. China Commun, 2022, 19: 73–84
33 He Y, Xiao Y, Zhang S, et al. Direct-to-smartphone for 6G NTN: technical routes, challenges, and key technologies. IEEE Netw, 2024, 38: 128–135
34 Leyva-Mayorga I, Martinez-Gost M, Moretti M, et al. Satellite edge computing for real-time and very-high resolution Earth observation. IEEE Trans Commun, 2023, 23: 2241–2253
35 He Y, Liu Y, Jiang C, et al. Multiobjective anti-collision for massive access ranging in MF-TDMA satellite communication system. IEEE Int Things J, 2022, 9: 14655–14666
36 Zhang X, Liu J, Zhang R, et al. Energy-efficient computation peer offloading in satellite edge computing networks. IEEE Trans Mobile Comput, 2024, 23: 3077–3091
37 Zhu B, Lin S, Zhu Y, et al. Collaborative hyperspectral image processing using satellite edge computing. IEEE Trans Mobile Comput, 2024, 23: 2241–2253
38 Jung S, Jeong S, Kang J, et al. Marine IoT systems with space-air-sea integrated networks: hybrid LEO and UAV edge computing. IEEE Int Things J, 2023, 10: 20498–20510
39 Wei K, Tang Q, Guo J, et al. Resource scheduling and offloading strategy based on LEO satellite edge computing. In: Proceedings of the IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, 2021. 1–6
40 Hao Y, Song Z, Zheng Z, et al. Joint communication, computing, and caching resource allocation in LEO satellite MEC networks. IEEE Access, 2023, 11: 6708–6716
41 He Y, Sheng B, Yin H, et al. Multi-objective deep reinforcement learning based time-frequency resource allocation for multi-beam satellite communications. China Commun, 2022, 19: 77–91
42 Zhou C, Wu W, He H, et al. Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN. IEEE Trans Wireless Commun, 2020, 20: 911–925
43 Tang F, Hofner H, Kato N, et al. A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN). IEEE J Sel Areas Commun, 2021, 40: 276–289
44 Asheralieva A, Niyato D. Distributed dynamic resource management and pricing in the IoT systems with blockchain-as-a-service and UAV-enabled mobile edge computing. IEEE Int Things J, 2020, 7: 1974–1993
45 Zhang H, Zhao H, Liu R, et al. Leader federated learning optimization using deep reinforcement learning for distributed satellite edge intelligence. IEEE Trans Serv Comput, 2024, 17: 2544–2557
46 Zhou Y, Lei L, Zhao X, et al. Decomposition and meta-DRL based multi-objective optimization for asynchronous federated learning in 6G-satellite systems. IEEE J Sel Areas Commun, 2024, 42: 1115–1129
47 Ji Z, Wu S, Jiang C. Cooperative multi-agent deep reinforcement learning for computation offloading in digital twin satellite edge networks. IEEE J Sel Areas Commun, 2023, 41: 3414–3429
48 Li X, Zhang H, Zhou H, et al. Multi-agent DRL for resource allocation and cache design in terrestrial-satellite networks. IEEE Trans Wireless Commun, 2023, 22: 5031–5042
49 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. 2017. ArXiv:1706.03762
50 Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022. 27730–27744
51 Elbert B R. Introduction to Satellite Communication. Norwood: Artech House, 2008
52 SpaceX. SpaceX Non-geostationary Satellite System: Technical Information to Supplement Schedule S. Federal Communications Commission, SAT-MOD-20200417-00037, 2021
53 Chai F, Zhang Q, Yao H, et al. Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT. IEEE Trans Veh Technol, 2023, 72: 7783–7795