Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Adaptive multi-layer deployment for a digital-twin-empowered satellite-terrestrial integrated network*

Yihong TAO[1], Bo LEI[2], Haoyang SHI[1], Jingkai CHEN[1], Xing ZHANG[†‡1]

[1]*Wireless Signal Processing and Network Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China*

[2]*Research Institute of China Telecom Co., Ltd., Beijing 102209, China*

[†]E-mail: hszhang@bupt.edu.cn

Received Apr. 25, 2024; Revision accepted July 24, 2024; Crosschecked Nov. 14, 2024

Abstract: With the development of satellite communication technology, satellite-terrestrial integrated networks (STINs), which integrate satellite networks and ground networks, can realize global seamless coverage of communication services. Confronting the intricacies of network dynamics, the resource heterogeneity, and the unpredictability of user mobility, dynamic resource allocation within networks faces formidable challenges. Digital twin (DT), as a new technique, can reflect a physical network to a virtual network to monitor, analyze, and optimize the physical networks. Nevertheless, in the process of constructing a DT model, the deployment location and resource allocation of DTs may adversely affect its performance. Therefore, we propose a STIN model, which alleviates the problem of insufficient single-layer deployment flexibility of the traditional edge network by deploying DTs in multi-layer nodes in a STIN. To address the challenge of deploying DTs in the network, we propose a multi-layer DT deployment problem in the STIN to reduce system delay. Then we adopt a multi-agent reinforcement learning (MARL) scheme to explore the optimal strategy of the DT multi-layer deployment problem. The implemented scheme demonstrates a notable reduction in system delay, as evidenced by simulation outcomes.

Key words: Digital twin; Satellite-terrestrial integrated network; Deployment; Multi-agent reinforcement learning
https://doi.org/10.1631/FITEE.2400327 **CLC number:** TN91

1 Introduction

With the development of networks, satellite-terrestrial integrated networks (STINs) are a new network type that can make full use of ground networks and satellite networks to meet user service requirements (Kato et al., 2019; Zhang ZJ et al., 2019). According to the network environment and service requirements, STINs provide global seamless coverage and diversified mobile access services for various types of mobile terminals, breaking through geographical and environmental limitations. In remote areas, where aircraft and oceangoing ships cannot be covered by ground networks, satellites can provide network services. Systems specializing in commercial mobile satellite communication, exemplified by Iridium and dedicated maritime satellites, offer proficient solutions tailored to maritime operations, emergency response, and personal mobile communication needs. The emergence of substantial commercial low-orbit satellite constellations, notably OneWeb, StarLink, and Space Norway, underscores the pivotal role that low-orbit satellites are poised to play as a robust augmentation to the anticipated sixth-generation (6G) ground networks of the future (Liu JJ et al., 2018b; Zhang ZQ et al., 2019; Tang QQ et al., 2021).

Satellite networks transmit data and

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 62271062)
ORCID: Yihong TAO, https://orcid.org/0009-0007-8805-7763

information through satellite signals and possess the capability to encompass diverse geographical territories extensively, including remote areas and places such as oceans that cannot be covered. Terrestrial networks provide more stable and high-speed communication services in cities and densely populated areas, and are also used for data transmission and connectivity with satellite networks. Therefore, academia and industry are promoting the combination of the two, which can achieve communication coverage and connectivity on a global scale, meeting the communication requirements of different regions (Zhu et al., 2019; Fu et al., 2020; Wang GC et al., 2020). For different application scenarios, STINs need to meet the differentiated requirements of large-scale connection of users. However, the existing independent control of multi-dimensional satellite ground resources prevents many physical resources in heterogeneous nodes from being effectively used, which affects the system ability to provide reliable services for users (Wang P et al., 2020; Guo HZ et al., 2022; Zhang JX et al., 2023). A pressing need arises for a revolutionary technology that can significantly bolster network flexibility, streamline decision-making processes, and elevate the proficiency of data management practices. The emerging digital twin (DT) intelligent technology can monitor, control, and optimize the physical entities, and is expected to solve the above problems well (Bellavista et al., 2021; Wu et al., 2021).

DT technology can create accurate virtual twin copies of physical objects through digital technology, and the created DT can communicate with the object in the real world in real time, and dynamically reflect its state and change. On one hand, DT technology enables the physical system to be monitored and managed in real time, including the monitoring of system status, fault detection, and early warning. On the other hand, DT technology can help predict the behavior and performance of real systems and optimize them accordingly. By simulating and analyzing the DT data, problems in the actual system can be found and solved, and the efficiency and the system performance can be improved (Barricelli et al., 2019; Tao et al., 2019). Some works have proposed using DTs to solve the problems existing in STINs. Zhao et al. (2022) proposed using DTs to assist in solving the problem of possible loops in the satellite handover process. Zhou et al. (2023) established a DT copy for the satellite network to alleviate the challenges of overly complex satellite network design, simulation, deployment, and maintenance. Yin et al. (2023) ensured the security of wireless communication between the satellite and ground through DT-assisted multi-dimensional domain cooperative precoding. Fan et al. (2023) proposed a novel approach using dynamically adapted micro-cloud architecture, grounded in DTs and multi-agent system technology, which was introduced to tackle the challenge of optimizing satellite liaison window task scheduling priorities. Jiang et al. (2023) established a handover prediction and congestion prediction model for inter-satellite link in the DT network to ensure the quality of satellite communication services. Ji et al. (2023) established a DT-driven satellite-terrestrial integrated edge computing network to optimize resource allocation.

The premise of the above research is that DTs have been properly deployed in the server; however, the deployment of DTs in the network affects their performance, and building DT models for users requires significant data communication capacity and computing capacity to process data, so DT deployment is a basic problem that needs further study. As stated in Tang FX et al. (2022), when considering DT deployment solutions, we should consider the unique requirements of each application, which vary significantly. These requirements encompass latency sensitivity, the desired level of service experience quality, the specific allocation of computing resources, and the system reliability. On one hand, the dynamic state in the network may affect DT performance and the quality of services. On the other hand, the computing resources required to maintain DTs in the server and ensure low-latency data interaction between physical entities and DTs need to be considered, which makes the DT deployment problem different from traditional placement problems (Cao et al., 2018; Liu JJ et al., 2018a). There are studies proposing DT deployment in the edge network to form digital twin edge networks (DITENs). Lu et al. (2021b) formulated the DT placement challenge as an edge association task and proposed a scheme based on federated learning to reduce communication and computing latency. Sun et al. (2020) proposed the use of DTs in the entire mobile edge computing (MEC) network to provide an offloading scheme in DITENs to minimize system latency.

However, with the further development and application of DT technology, there are higher requirements for the deployment of DT models, such as providing remote DT services for users at sea; in this challenging situation, considering the cost and user service experience, it is not feasible to use only a traditional ground network to transmit and process data. Introducing DT technology into STINs has become a promising solution to achieve global coverage, improve network reliability, and ensure user service quality. Therefore, it is necessary to further study the application of DTs in STINs, and how to ensure the interaction between physical space and DT space in the highly dynamic and complex heterogeneous network environment. To our knowledge, no work has investigated DT deployment in DT-assisted STINs.

We first describe DT technology in STINs and propose a DT STIN model. By placing DTs in different nodes in the network, we propose a DT multi-layer deployment problem. Finally, we propose a solution to this problem using the multi-agent reinforcement learning (MARL) algorithm. The primary contributions of this research are outlined as follows:

1. We propose a DT architecture for a STIN. DTs can capture real-time user status and monitor and optimize users. We propose deploying DTs in multi-layer nodes in the network, which alleviates the problem of insufficient flexibility in single-layer deployment of traditional DITENs.

2. To address the challenges of deploying DTs in the network, we innovatively formulate the DT multi-layer deployment problem in a STIN to reduce system delay, ensure the interaction between users and DTs, and improve the user experience of DT services.

3. We propose an algorithm based on MARL that formulaically solves the proposed DT deployment problem by considering the DT deployment strategy and system delay. Ultimately, the simulation results substantiate the efficacy of the proposed algorithm.

2 Related works

DTs can virtually model physical entities, and deploying DTs in the network can then monitor, analyze, predict, and optimize the network. Dong et al. (2019) proposed the use of DTs to optimize user association, resource allocation, and offloading probability in MEC to reduce system energy consumption. Zhang K et al. (2022) integrated artificial intelligence and DT technology principles into the architecture of an edge computing network, showcasing an innovative approach that combines these advanced technologies for enhanced performance and functionality, and focused on matching the potential edge services. Liu T et al. (2022) introduced a strategy where mobile users can intelligently delegate computational tasks to servers, facilitated by the utilization of DTs. Guo Q et al. (2023) devised a novel unmanned aerial vehicle-augmented mobile network infrastructure, tailored to ensure seamless and efficient communication services for all mobile users operating within congested, high-density traffic environments, and introduced an online training based DT authorization mechanism for dynamic resource allocation. For the relevant satellite network schemes, to realize an intelligent sky and ground integrated vehicle network, Hui et al. (2023) designed an architecture that supports DT technology. Mao et al. (2024) used DT technology to create a virtual network for a physical satellite network, optimize network performance, and solve the limitation of low reliability of traditional satellite simulation platforms. Duong et al. (2023) supported the optimal multi-beam design of a 6G satellite–ground integrated network according to the requirements of critical tasks and combined it with the DT technology. Guo Q et al. (2024) presented the challenges of meeting the requirements of DTs in the 6G network, and analyzed the technologies of implementing DT systems.

For the problem of deploying DTs in a network to improve the performance of a physical system, Lu et al. (2021a) proposed a wireless DITEN to effectively build and maintain DTs in a wireless DT network, and the DT placement problem was proposed for dynamic network status and changing network topology. Zhang H et al. (2023) designed a dynamic edge network DT model that captures the dynamic and service requirements of a real-time edge network and realizes the efficient deployment of DT servers. Chukhno et al. (2022) considered the deployment details of the edge network and device types and their social characteristics, and solved the dynamic layout problem of DTs with social ability at the edge. To enhance the property of the DT model, Zhang YD et al. (2024) proposed a DT architecture with wireless computing capability, and the placement and migration problems of DTs were solved by considering synchronization delay and DT error. Vaezi et al. (2023) considered the placement of DTs to meet the requirements to provide response delay minimization to all users.

The above studies provide reference solutions for the placement and deployment of DTs. Distinct from the aforementioned research efforts, this study considers how to deploy DTs in a STIN. To optimize the utilization of available resources in the STIN, we consider a multi-layer deployment method. Our goal is to minimize the delay between users and DTs.

3 System model

We introduce the architecture of the DT-empowered STIN model, and then study the communication and computing models in the system to formulate the DT multi-layer deployment problem.

3.1 Architecture of the DT-empowered STIN system

We propose an architecture that introduces DT technology in a STIN, as shown in Fig. 1. The architecture includes two spaces: a physical space and a DT space. With the help of DT nodes, terminal devices along with satellites and cloud servers comprise the diverse entities of the physical layer, which can be systematically mapped and modeled to generate DT models. This process enables the creation of DT models that accurately mirror the composition and operations of these physical devices. Satellites can provide seamless coverage of network services for the terminal devices, and cloud servers have relatively sufficient computing resources. DT nodes are selected from nodes with computing power for deploying DT models.

In the STIN, the association between terminal devices and DT nodes can be established to meet the needs of communication and computing resources. DTs realize accurate mapping of physical entities through data transmission, DT modeling,

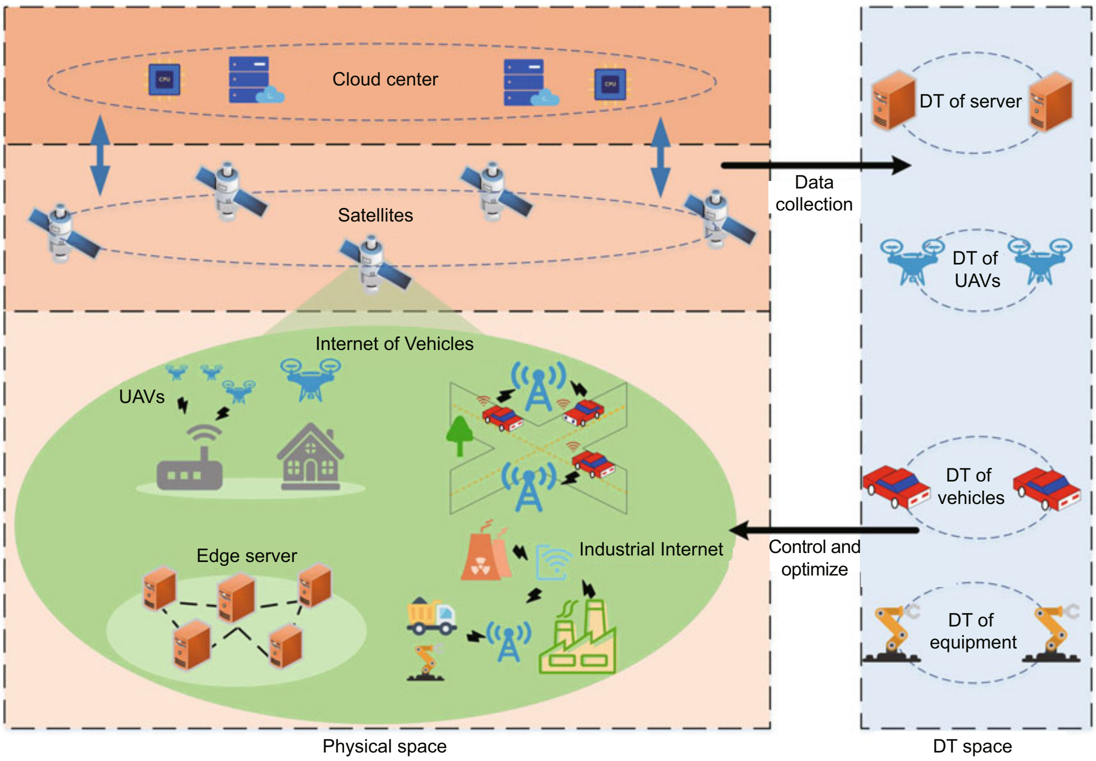

Fig. 1 DT network architecture of a satellite-terrestrial integrated network (UAVs: unmanned aerial vehicles; DT: digital twin)

and feedback optimization. Through DTs, the historical state information and real-time operation status of the terminal devices can be collected to monitor, predict, and analyze these devices. Each DT model in the DT layer can map the real-time operation status of the terminal devices. In addition, we assume that the DT corresponding to the terminal devices is deployed on the resource-rich compute nodes. In this architecture, we select an optimized compute node as the deployment location of the DT model for different terminal devices to build the DT model. Let $\mathcal{I} = \{1, 2, \ldots, I\}$ and $\mathcal{J} = \{1, 2, \ldots, J\}$ represent the terminal devices that are needed to build a DT and the nodes with computing resources respectively, and there is some overlap between $\mathcal{I}$ and $\mathcal{J}$. This is because some terminal devices need the DT services and have free computing resources to provide for other terminal devices to maximize the utilization of computing resources.

The DT multi-layer deployment problem is established according to the DT model construction method in this study. DT deployment is the first step in constructing a DT, and includes finding the best location for deploying the DT model for each terminal device from multi-layer compute nodes, to ensure low-latency interaction between physical entities and the DT model.

3.2 Communication and computing model

Fig. 2 shows the complete procedure of deploying the DT of user i in different nodes. In our system, DT can be deployed locally, by other terminal devices, or by satellite transmitting to the cloud server (CS). The delay in deploying a DT includes mainly three issues: transmission delay, computation delay, and feedback delay. The data of user i are transferred to the corresponding compute node responsible for deploying the DT, and then the compute node uses the computing power to efficiently process and interpret the acquired data, subsequently constructing a personalized DT model specifically for user i, as shown in Fig. 2. Ultimately, the feedback results are transmitted back to the user; the feedback results meet the service requirements of the user or have certain optimization effects. The specific communication and computing models are as follows:

1. Deployment in local nodes. If the local terminal has certain computing resources, the DT can be deployed on the local terminal without communica-

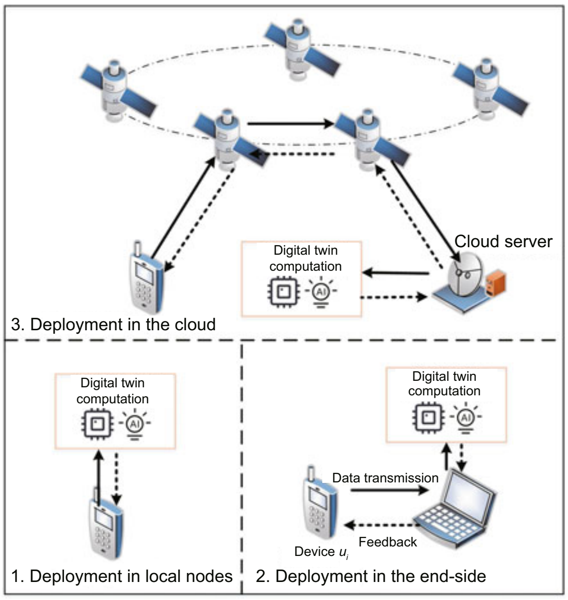

Fig. 2 Process of constructing a digital twin model

tion delay. The delay calculation is

$$L_i = \frac{\mu D_i}{f_i}, \tag{1}$$

where D_i is the data size of the DT of user i and f_i is the computing resource of user i. μ is the computing resource required for processing each bit of data.

2. Deployment in the end-side. DT can be deployed on other terminal devices with rich computing resources. Communication between terminal devices is via Bluetooth or Wi-Fi. The communication between the terminal and the end-side includes uplink communication for transmitting data to the DT node and downlink communication for transmitting feedback results from the end-side node back to the user. The size of the feedback results returned to the user is much smaller than that of the DT data that the user needs to transmit, so only the uplink communication delay is considered in the communication model (Yao et al., 2023). Wireless communication is adopted between the user and the end-side devices, and the achievable data rate is

$$r_{ij} = W\log_2\left(1 + \frac{P_i G_{ij}}{WN_0}\right), \tag{2}$$

where P_i represents the transmission power, G_{ij} represents the channel power gain between user i and the compute node j, W is the channel bandwidth, N_0 is the noise power spectral density, and the transmission delay from user i to the end-side device j is

$$L_{ij}^{\mathrm{com}} = \frac{D_i}{r_{ij}}. \tag{3}$$

The total computing resources of end-side device j are expressed as F_j. Computing resources can be allocated from end-side devices with sufficient computing resources to multiple users to maintain their DT deployed on end-side device j. Define f_{ij} as the computing resources allocated by terminal device j to user i. The delay required for user i to process the DT data can then be expressed as

$$L_{ij}^{\mathrm{cmp}} = \frac{\mu D_i}{f_{ij}}, \tag{4}$$

where $\sum\limits_{i=1}^{|\mathcal{U}|} \lambda_{ij} f_{ij} \leq F_i$, $\lambda_{ij}=1$ if $f_{ij}>0$, and $\lambda_{ij}=0$ otherwise. Based on the preceding formula, the delay for deploying the DT at the end-side is

$$L_{ij} = \frac{D_i}{r_{ij}} + \frac{\mu D_i}{f_{ij}}. \tag{5}$$

3. Deployment in the cloud. DTs can be deployed on the CS. When users cannot connect to the CS through the ground network, DT data can be transmitted to the CS through the STIN by the backhaul link. In our system model, user i can connect to the satellite directly through wireless communication, and then the satellite will return to the ground gateway station, which is connected to the CS. DT data processing and modeling are performed in the CS. The computing resources in the CS are sufficient, but there is a large communication delay.

For wireless communications between users and satellites, the channel capacity of the connection between them is limited by the distance between them, transmission power, signal-to-noise ratio, and other parameters. The transmission bandwidth for wireless data transmission from user i to the covered satellite is equally distributed among all users, expressed as $W_{i\mathrm{s}}$. Our framework employs a quasi-static approach, where the channel conditions and network topology are assumed to be stable within a given time slot, facilitating the analysis and modeling process (Li et al., 2023), $G_{i\mathrm{s}}$ represents the channel gain, $P_{i\mathrm{s}}$ represents the transmission power, and the achievable data rate from user i to the satellites is

$$r_{i\mathrm{s}} = W_{i\mathrm{s}} \log_2 \left(1 + \frac{P_{i\mathrm{s}} G_{i\mathrm{s}}}{W_{i\mathrm{s}} N_0}\right). \tag{6}$$

After the satellite receives the user data, it transmits the data to the CS through the ground gateway station. Similar to Eq. (6), the achievable data rate from the satellite to the ground gateway station is

$$r_{\mathrm{sc}} = W_{\mathrm{sc}} \log_2 \left(1 + \frac{P_{\mathrm{sc}} G_{\mathrm{sc}}}{W_{\mathrm{sc}} N_0}\right), \tag{7}$$

where W_{sc} represents the bandwidth assigned to the satellite by the ground gateway station, P_{sc} is the transmission power of the satellite, and G_{sc} is the channel gain between them.

The transmission delay for transferring D_i is

$$L_{i\mathrm{c}}^{\mathrm{com}} = \frac{D_i}{r_{i\mathrm{s}}} + \frac{D_i}{r_{\mathrm{sc}}} + \frac{d_{i\mathrm{s}} + d_{\mathrm{sc}}}{c}, \tag{8}$$

where c is the speed of light, and $d_{i\mathrm{s}}$ and d_{sc} are the distance between user i and the satellite and the distance between the satellite and the CS, respectively. The calculation delay in the CS is expressed as

$$L_{i\mathrm{c}}^{\mathrm{cmp}} = \frac{\mu D_i}{f_{i\mathrm{c}}}, \tag{9}$$

where $f_{i\mathrm{c}}$ is the computing resource allocated by the CS to user i.

The total delay for user i is expressed as

$$L_{i\mathrm{c}} = L_{i\mathrm{c}}^{\mathrm{com}} + L_{i\mathrm{c}}^{\mathrm{cmp}}. \tag{10}$$

The main notations used in this paper are summarized in Table 1.

4 Problem formulation

We formulate the DT multi-layer deployment problem by considering the different characteristics of compute nodes located at different layers. DT deployment needs to consider the delay caused by data transmission and the establishment of DT models. In our system model, the dynamic network state changes are considered, and the optimal DT deployment location is selected for each user to maximize the system performance.

The constraints of the optimization problem are described below. We use $G(\mathcal{U}, \mathcal{N}, \mathcal{M}, \mathcal{C})$ to represent our DT STIN, where $\mathcal{U}$ is the set of users who need to build the DT, $\mathcal{N}$ is the set of nodes with computing power, $\mathcal{M}$ is the set of nodes deploying the DT, and $\mathcal{C}$ is the set of communication links in the network. Capacity N_j of the DT node indicates the maximum

Table 1 Summary of main notations

Notation	Description
$\mathcal{I}$	Set of users
$\mathcal{J}$	Set of compute nodes
L_i	Latency of user i deployed in local nodes
μ	Computing resource for processing 1-bit data
D_i	Data size of user i
f_i	Computing resource of user i
r_{ij}	Achievable data rate from user i to node j
W	Bandwidth of the transmission channel
P_i	Transmission power of user i
G_{ij}	Channel power gain between user i and node j
L_{ij}^{com}	Communication latency from user i to node j
L_{ij}^{cmp}	Computation latency of tasks for user i in node j
$r_{i\text{s}}$	Achievable data rate between user i and the satellite
$W_{i\text{s}}$	Transmission bandwidth between user i and the satellite
$P_{i\text{s}}$	Transmission power of user i to the satellite
$G_{i\text{s}}$	Channel power gain between user i and the satellite
r_{sc}	Achievable data rate between the satellite and the cloud
W_{sc}	Transmission bandwidth between the satellite and the cloud
P_{sc}	Transmission power of the satellite to the cloud
G_{sc}	Channel power gain between the satellite and the cloud
L_{ic}	Communication latency between user i and the cloud
S^t	MARL state at time slot t
A^t	MARL action at time slot t
R^t	MARL reward at time slot t

number of DTs that the node can maintain. We use the matrix

$$\boldsymbol{\lambda}=\begin{bmatrix}\lambda_{11} & \lambda_{12} & \cdots & \lambda_{1J}\\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2J}\\ \vdots & \vdots & & \vdots\\ \lambda_{I1} & \lambda_{I2} & \cdots & \lambda_{IJ}\end{bmatrix} \tag{11}$$

to represent the correspondence between the user and the deployed DT, where $\lambda_{ij}=1$ if the DT of user i is deployed on node j, and $\lambda_{ij}=0$ otherwise. The total system delay is an indicator to measure the DT deployment policy and represents the operating performance of the system. The total deployment delay is

$$L_{\text{sum}}=\sum_{i\in\mathcal{I},\ j\in\mathcal{J}}\lambda_{ij}L_{ij}. \tag{12}$$

In the above formula, the system delay is determined by the DT deployment strategy and the deployment delay between users and DT nodes, and the deployment delay is affected by compute nodes located at different layers. The goal of the DT multilayer deployment problem is to find the right deployment location to minimize system latency, and deployment at different layers directly affects system latency performance. Therefore, the optimization problem is expressed as follows:

$$\min \quad L_{\text{sum}} \tag{13a}$$

$$\text{s.t.}\ \lambda_{ij}\in\{0,1\}, \tag{13b}$$

$$\sum_{j\in\mathcal{J}}\lambda_{ij}=1,\forall i\in\mathcal{I}, \tag{13c}$$

$$\sum_{i\in\mathcal{I}}\lambda_{ij}\leq N_j,\forall j\in\mathcal{J}, \tag{13d}$$

$$0\leq\sum_{i\in\mathcal{I}}\lambda_{ij}f_i\leq f_j^{\max},\forall j\in\mathcal{J}. \tag{13e}$$

Constraint (13b) indicates that variable λ_{ij} has only two states, 0 and 1, which represent undeployed and deployed, respectively. Constraint (13c) indicates that user's DT can be deployed in only one DT compute node. Constraint (13d) indicates that the number of DTs deployed in compute nodes cannot exceed the capacity. Constraint (13e) means that the computing capacity allocated to each compute node cannot exceed its maximum computing capacity. To obtain a satisfactory deployment strategy, it is necessary to search it in a space consisting of J^I possible deployment decisions. Traditional optimization methods, such as game theory and convex optimization, have difficulty in guaranteeing the long-term performance of the resulting decisions, have high complexity, and are not suitable for the network with rapidly changing environments considered in this study. To address Problem (13a), it is essential to find the optimal decision in each time slot. The model-free reinforcement learning method adopted in this study is a very promising approach for learning the optimal strategy through direct interaction with dynamic and decentralized environments when dealing with tasks with unknown environments. Specifically, through an algorithm based on MARL, for Constraint (13c), each agent will select only the action space where the only one action is 1. For Constraint (13d), when the number of DTs deployed in compute node j reaches the maximum, the agent will not choose to continue to deploy on this node. For Constraint (13e), computing resources in a compute node are evenly distributed among all agents.

5 MARL-based algorithm for the multi-layer deployment problem in the STIN

We propose an MARL-based algorithm to solve the proposed DT multi-layer deployment problem. The MARL framework will be introduced in the STIN considered in this study to maximize system performance. We treat Problem (13a) as a distributed MARL task and set each user as one agent. This is explained in detail below.

5.1 Problem transformation

Reinforcement learning does not require datasets, but rather learns through interaction with the environment. It follows a trial-and-error approach, optimizing its decision-making strategies by constantly exploring the environment and learning from experience. In this kind of communication network, with the increase of user number, the application of traditional single-agent reinforcement learning methodologies in complex and ever-evolving environments poses a significant challenge, as it may cause the agent to unduly focus its attention solely on the immediate behavior of its competitors, while ignoring other potential behavior patterns or strategies. The MARL algorithm in this paper can help the agent consider the existence and behavior of other agents in the learning process, make more intelligent decisions, and achieve better system performance.

Problem (13a) can be reframed as a decentralized partially observable Markov decision process (Dec-POMDP) model specifically tailored for DT users, each corresponding to an agent, each of which can observe only a local space at each time slot (Guo ZY et al., 2022). Each agent independently selects actions based on the observed local spatial states to maximize system rewards. This problem can be formulated as $\langle S, \{O_i\}_{i\in\mathcal{I}}, \{A_i\}_{i\in\mathcal{I}}, R, \gamma\rangle$, where S is the system state, O_i is the state space observed locally by agent i, A_i is the action space of agent i, R is the reward function, and γ is the discount factor. Each agent first performs an action $a_i^t \in A_i$ based on the locally observed state $o_i^t \in O_i$, and then the joint action can be obtained, expressed as $a^t \in A = A_1 \times A_2 \times \cdots \times A_I$. Following the execution of a collective action at time slot t, the environment computes a global reward $r^t = R(s^t, a^t)$ and proceeds to update the system state to s^{t+1}. In the subsequent sections, we introduce the constructs of the system state space S, the individual agent's observation spaces $\{O_i\}_{i\in\mathcal{I}}$, the action spaces $\{A_i\}_{i\in\mathcal{I}}$ each tailored to a user, and the reward function R that evaluates the effectiveness of actions.

1. System state. In our proposed scheme, the system state includes the physical state of all users φ^t, the network resource state β^t, and deployment policy decisions λ^t. The physical status of user i includes the distance, the data to be transferred of user i, and the data transmission rate. We represent the system state as

$$S^t = \{\varphi^t,\ \beta^t,\ \lambda^t\}. \tag{14}$$

2. Action space. Agents are users in our system. Each agent selects its action, including the DT deployment policy. The deployment policy of user i is $\{\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{ij}\}$, where $\lambda_{ij} = 1$ if the DT of end user i is deployed to compute node j; otherwise, $\lambda_{ij} = 0$. The action space across all agents can be collectively represented as

$$A^t = \{a_i | a_{ij} = \lambda_{ij}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}\}, \tag{15}$$

where a_{ij} is the deployment decision made by the agents.

3. Reward function. In this MARL, all agents share the reward. When a joint action is taken, the environment returns a reward. The primary objective for each agent is to expedite the deployment of its DT by minimizing associated delays. The overall reward function encompasses two key components: a delay-based reward function R_{L} and a deployment cost function R_{C}. Here, we elaborate that the delay-based reward function is

$$R_{\mathrm{L}} = -L_{\mathrm{sum}}. \tag{16}$$

The cost incentive function for deploying DT services is

$$R_{\mathrm{C}} = \phi m, \tag{17}$$

where ϕ is defined as the unit expense coefficient for deploying the DT services, and m is the number of DTs. The total reward function is

$$R^t = \alpha R_{\mathrm{L}} - \beta R_{\mathrm{C}}, \tag{18}$$

where α and β are the balancing coefficients for reward and cost functions, respectively. We introduce the concept of applying MARL as a means to optimize the search for the most effective DT deployment strategy, where all agents collaborate to make deployment decisions. The goal is to maximize the global system reward, which is described as follows:

$$R = \sum_{t=1}^{T} \gamma^{t-1} R^t, \tag{19}$$

where $\gamma \in (0,1]$ signifies the reward discount factor, and each agent i acts according to the local observation state and its local strategy in the partially observable environment. We use $\pi = \{\pi_1, \pi_2, \ldots, \pi_i\}$ to represent the joint strategy of the agents. The joint action function is

$$Q^\pi(s^t, a^t) = E[R^t \mid s^t, a^t], \tag{20}$$

where $E[\cdot]$ signifies the expected value operation, whereas the joint action function, denoted as $Q^\pi(s^t, a^t)$, quantifies the anticipated global reward when commencing from s^t, a^t is followed by the strategy π, and the optimal union is π^* when $Q^\pi(s^t, a^t)$ is the largest.

5.2 MARL for DT deployment

The agent in a traditional centralized reinforcement learning algorithm updates its policy independently as learning progresses, which makes the convergence of the algorithm difficult to achieve (Guo ZY et al., 2022). Therefore, we propose using an MARL-based approach to solve our DT deployment problem.

This algorithm adopts a centralized training decentralized execution approach, which can use system's global environment information to train agents centrally, and each agent chooses to deploy actions in a decentralized manner according to the local observation environment (Tan et al., 2022). However, the local observed state (o_i, a_i) cannot describe the environment state, and historical observations can be stored to help the function Q. The historical observation results of the agent joint actions are expressed as $\tau^t = \{\tau_1^t, \tau_2^t, \cdots, \tau_I^t\}$. The joint actions of all agents are represented as $a^t = \{a_1^t, a_2^t, \cdots, a_I^t\}$. Hence, the primary objective of this algorithmic approach is to acquire the cumulative action-value function $Q_{\text{sum}}(\tau^t, a^t)$ that integrates global information during the centralized training phase, enabling a comprehensive evaluation of joint actions. By combining action-value functions, distributed agents can choose actions that are closer to the global optimal solution. The architecture of the presented algorithm, as depicted in Fig. 3, comprises the agent network module and the integrated hybrid network module.

1. Agent Q-network. Each agent incorporates a gate recurrent unit (GRU) alongside fully connected (FC) layers. The agent processes its local observation o_i^t and the previous action a_i^{t-1} as inputs, subsequently generating the local action-value function $Q_i(\tau_i^t, a_i^t)$ as its output. In the Q-network of each agent, h_i^{t-1} plays a crucial role. As the input of the GRU, h_i^{t-1} not only contains the internal state information of agent i at time step $t-1$, but also integrates all the historical observation and action information from the initial time step to $t-1$. This ability allows the agent to learn and make decisions effectively in a partially observable environment, even when the observed value o_i^t of the current time step may not be sufficient to fully reveal the state of the environment.

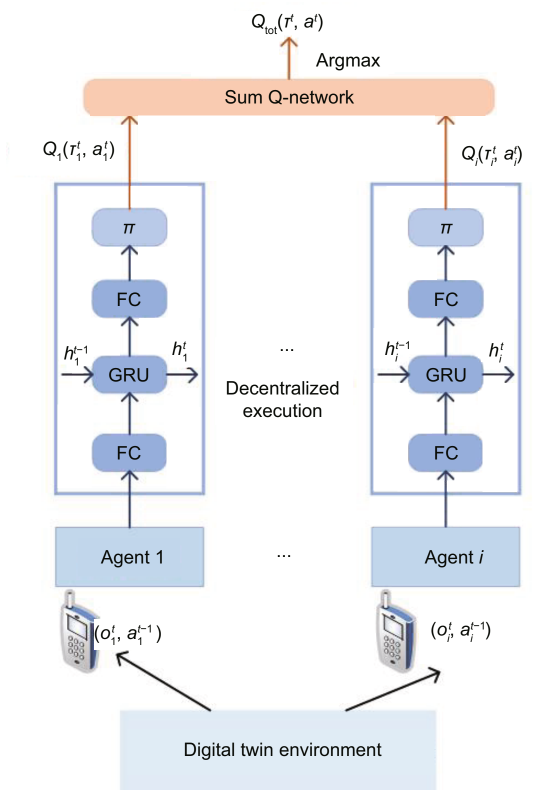

Fig. 3 The MARL method for the DT deployment structure (MARL: multi-agent reinforcement learning; DT: digital twin; GRU: gate recurrent unit; FC: fully connected)

In a partially observable network, historical observations can be stored in a network, which contributes to agent's ability to learn over longer timescales (Greff et al., 2017).

2. Mixing network. The MARL algorithm based on value decomposition dissects the cumulative action-value function $Q_{\text{tot}}(\tau^t, a^t)$ into individual local action-value functions $Q_i(\tau_i^t, a_i^t)$ to guide the strategy of each agent separately (Sunehag et al., 2018). The joint action-value function is defined as follows:

$$Q_{\text{tot}}(\tau^t, a^t) \approx \sum_{i \in \mathcal{I}} Q_i(\tau_i^t, a_i^t; \theta_i^t), \tag{21}$$

where θ_i^t is the Q-network parameter. The constraints between Q_{tot} and any single action-value function Q_i need to be satisfied as follows:

$$\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0,\ \forall i \in \mathcal{I}. \tag{22}$$

We express the target Q-value as Q_i^-, thus $Q_{\text{tot}}^-(\tau^t, a^t) \approx \sum\limits_{i \in \mathcal{I}} Q_i^-(\tau_i^t, a_i^t; \theta_i^t)$. The time difference target is represented by $y_{\text{tot}} = r + \gamma \max_{a'} Q_{\text{tot}}(\tau', a'; \theta^-)$, where θ^- is the target network parameter. The overall loss function is

$$L(\theta) = \frac{1}{X} \sum_{x=1}^{X} \left(y_{\text{tot}}^x - Q_{\text{tot}}^x(\tau^x, a^x)\right)^2, \tag{23}$$

where X denotes the quantity of small batch training data that are randomly drawn from the replay buffer, and x signifies the specific instance within this collection, representing the x^{th} sampled batch. The pseudo-code for the MARL-based DT deployment algorithm is shown in Algorithm 1.

Algorithm 1 MARL-based DT deployment algorithm

Input: The user set $\mathcal{I}$ and the compute node set $\mathcal{J}$
Output: The DT server deployment strategy
for all agents $i \in \mathcal{I}$ **do**
  Initialize the network parameters and replay buffer $\mathcal{D}_i$
end for
for episode $e = 1$ to E **do**
  Initialize the system state S^t
  for time step t **do**
    for all agents $i \in \mathcal{I}$ **do**
      Determine actions $a_i^t = \max_a Q(o_i^t, a; \theta_i)$ based on the observation o^t
    end for
    Execute joint action $a^t = (a_1^t, a_2^t, \ldots, a_I^t)$ and receive joint reward r^t and new observations o^{t+1}
    for all agents $i \in \mathcal{I}$ **do**
      Store transition $(o_i^t, a_i^t, r^t, o_i^{t+1})$ in $\mathcal{D}_i$
    end for
    for all agents $i \in \mathcal{I}$ **do**
      Update θ_i using gradient descent based on transitions sampled from $\mathcal{D}_i$
      Update θ_i^- towards θ_i
    end for
  end for
  return the optimized DT server deployment strategy based on the learned policies
end for

5.3 Computational complexity analysis

The computational complexity (CC) of the proposed DT deployment algorithm is analyzed. First, the initialized network parameters are inputted into the agent network to generate the initial network environment, which requires $O(1)$. The centralized hybrid training process can be implemented in the main server with sufficient computing power. Thus, we consider mainly the CC of the algorithmic distributed process. For the multi-agent distributed execution process, each action and local observation are evaluated by the neural network to generate the corresponding local action-value function Q. We employ N_1, N_2, N_3, and N_4 to respectively represent the number of neurons of neural layers 1–4 in the neural network, and the number of hidden neural layers is represented by H. Hence, the CC for each agent is $O(N_1N_2 + (H-1)N_2N_3 + N_3N_4)$. For each neuron, an activation function needs to be applied before the output. Therefore, the total computational cost of the activation function can be approximated as $O(N_2H)$, because each neuron must perform an activation function calculation. By summing up the computational cost of matrix multiplication and the computational cost of the activation function, the total CC of the neural network is $O(N_1N_2 + (H-1)N_2N_3 + N_3N_4 + N_2H)$. Based on the aforementioned analysis, the CC of our algorithm can be approximated as $O(N_2(N_1 + HN_3) + N_3N_4)$.

6 Simulation results

This section provides several numerical simulations to verify the performance of the MARL-based DT deployment algorithm. We compare this algorithm to the following three benchmarks:

1. Independent Q-learning (IQL) scheme. In the IQL algorithm, each agent trains and makes

decisions based on local observations (Tang M and Wong, 2022).

2. QMIX scheme. In the QMIX algorithm, the individual action-value functions are summed to derive the combined joint action-value function by using hypernetwork weighted summation (Tan et al., 2022).

3. Random (RD) scheme. In this algorithm, each user randomly selects the deployment node.

6.1 Simulation settings

We consider simulating a network scenario with 20 users, 3 end-side devices, and 1 satellite that can provide backhaul links for transmitting user data to a ground cloud server. Users and terminal devices are randomly distributed in the 500 m × 500 m area. In our simulation, user status and network channel status change over time. The maximum transmission power of the user devices is 200 mW, the number of training rounds in our method is 1000, and the discount factor γ is set to 0.9. We set the weights of the reward function α and β to 0.7 and 0.3 respectively, and the weights of the reward function will affect the decision of the agent. This study focuses on reducing the system delay. A comprehensive listing of the simulation parameters can be found in Table 2 for detailed reference.

6.2 Results and analysis

Fig. 4 shows the convergence curves of the three algorithms as the number of training rounds increases. The proposed algorithm obtains the highest reward value, followed by the IQL and QMIX algorithms. The complexity of the IQL and QMIX algorithms is approximately equal to that of the algorithm in this study. Most operations involve matrix multiplication and addition, and the specific algorithm complexity is related to the structure of the neural network in the agent network (Tan et al., 2022; Tang M and Wong, 2022). The reason is that agents in the IQL scheme explore and learn independently in a shared environment, and as a result, agents lack the ability to discern whether alterations in the system environment stem from their individual actions or are a consequence of the behaviors executed by other agents within the system. This strategy of multiple agents learning independently in a non-stationary environment will lead to a decline in decision-making quality. Therefore, the efficiency of this algorithm is inferior to that of the proposed algorithm. The hybrid network can learn the weight of the local action-value function from the global action-value function. In our system, the Q-value weights of each agent are similar; therefore, the QMIX algorithm is not suitable for our system and the reward value is the lowest.

Table 2 Simulation parameters

Parameter	Value
Number of users	20
Number of end-side devices	3
Data size of user i (D_i)	[0.5, 2] Mb
Workload of one-bit data μ	[50, 150] cycles/bit
Computing capacity f_i	[0.5, 20] GHz
Noise power function N_0	−174 dBm/Hz
Number of training iterations	1000
Discount factor γ	0.9
Learning rate	0.0001
Batch size	64

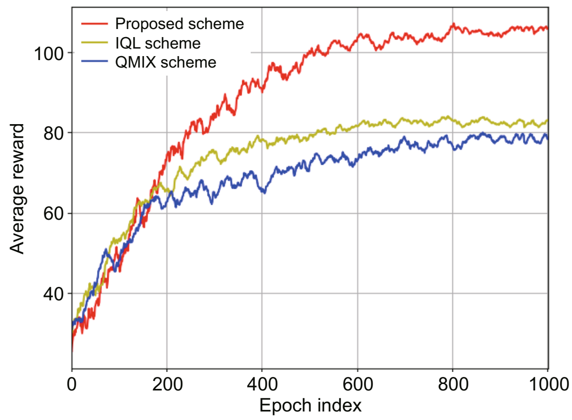

Fig. 4 Average reward

Fig. 5 shows that when different schemes are adopted, the average system delay changes with increasing number of training rounds. Fig. 5 shows that the average system delay is the lowest when the proposed scheme is adopted. The system delay is about 2.75 s when the IQL scheme is adopted, and about 2.8 s when the QMIX scheme is adopted, both higher than that of the proposed scheme. In the RD scheme, each user randomly selects the location of DT deployment, and the system delay generates certain jitter because of the random selection. Compared with the three benchmarks, our method can select the optimal deployment strategy based on the multi-agent network framework we trained, according to different users and network states.

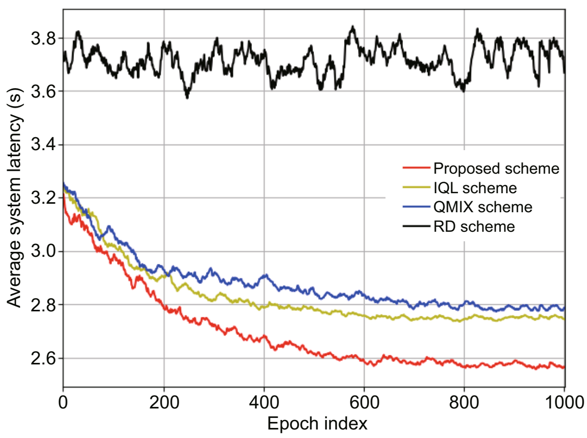

Fig. 5 Number of epoch rounds vs average system latency for different algorithms

Fig. 6 shows the comparative average system delays across four distinct schemes, each assessed under varying user counts. In this simulation, we specially configure user number from 18 to 24, the step size is 2, and the other parameters are based on the default configuration mentioned above. With the increasing user number, the average system delay increases in all schemes. This is because more users will make more service requests, thus occupying more resources and reducing the average quality of services for users. With different numbers of users, the proposed scheme always achieves the lowest average system delay compared with the other three benchmarks.

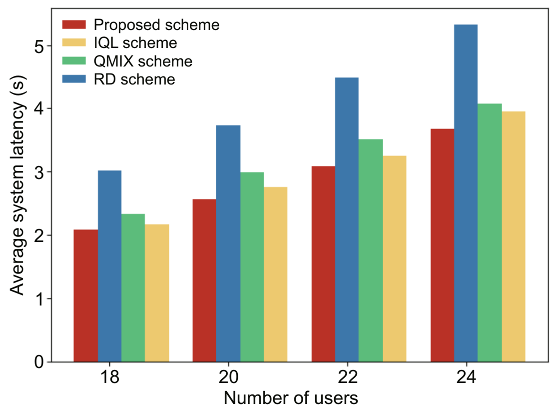

Fig. 6 Number of users vs average system latency for different algorithms

In Fig. 7, we compare the average system delay of different schemes with different parameters. We gradually increase the user data size from 0.5 to 2.0 Mb. As the size of user data increases, the average system latency increases for all schemes. The reason is that larger user data leads to the increased time required to transmit and process data. With different user data sizes, the proposed scheme achieves the best effect. The scheme we adopted has good robustness, and can adapt to different network environments to give the optimal strategy.

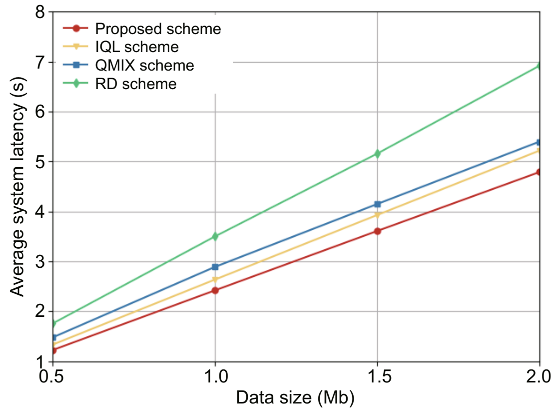

Fig. 7 Data size vs average system latency for different algorithms

7 Conclusions

We propose a new DT network model based on a STIN. We first propose a DT-driven STIN system model, including users, compute nodes, satellites, and cloud servers. Then we propose the multilayer deployment of DTs, aiming to reduce system latency, meet user service requirements, and alleviate the lack of flexibility of a traditional DITEN. In the case of dynamic network changes, we use an MARL algorithm to solve the proposed multi-layer deployment problem. Through comprehensive simulation encompassing diverse network conditions, it has been validated that the proposed approach significantly reduces system latency.

Contributors

Yihong TAO designed the research. Yihong TAO and Haoyang SHI processed the data. Yihong TAO and Jingkai CHEN drafted the paper. Haoyang SHI helped organize the paper. Bo LEI and Xing ZHANG revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Barricelli BR, Casiraghi E, Fogli D, 2019. A survey on digital twin: definitions, characteristics, applications, and design implications. *IEEE Access*, 7:167653-167671. https://doi.org/10.1109/ACCESS.2019.2953499

Bellavista P, Giannelli C, Mamei M, et al., 2021. Application-driven network-aware digital twin management in industrial edge environments. *IEEE Trans Ind Inform*, 17(11):7791-7801. https://doi.org/10.1109/TII.2021.3067447

Cao YR, Guo HZ, Liu JJ, et al., 2018. Optimal satellite gateway placement in space-ground integrated networks. *IEEE Netw*, 32(5):32-37. https://doi.org/10.1109/MNET.2018.1800016

Chukhno O, Chukhno N, Araniti G, et al., 2022. Placement of social digital twins at the edge for beyond 5G IoT networks. *IEEE Int Things J*, 9(23):23927-23940. https://doi.org/10.1109/JIOT.2022.3190737

Dong R, She C, Hardjawana W, et al., 2019. Deep learning for hybrid 5G services in mobile edge computing systems: learn from a digital twin. *IEEE Trans Wirel Commun*, 18(10):4692-4707. https://doi.org/10.1109/TWC.2019.2927312

Duong TQ, Nguyen LD, Bui TT, et al., 2023. Machine learning-aided real-time optimized multibeam for 6G integrated satellite-terrestrial networks: global coverage for mobile services. *IEEE Netw*, 37(2):86-93. https://doi.org/10.1109/MNET.003.2200275

Fan HL, Long J, Liu LM, et al., 2023. Dynamic digital twin and online scheduling for contact window resources in satellite network. *IEEE Trans Ind Inform*, 19(5):7217-7227. https://doi.org/10.1109/TII.2022.3228682

Fu S, Gao J, Zhao L, 2020. Integrated resource management for terrestrial-satellite systems. *IEEE Trans Veh Technol*, 69(3):3256-3266. https://doi.org/10.1109/TVT.2020.2964659

Greff K, Srivastava RK, Koutník J, et al., 2017. LSTM: a search space odyssey. *IEEE Trans Neur Netw Learn Syst*, 28(10):2222-2232. https://doi.org/10.1109/TNNLS.2016.2582924

Guo HZ, Li JY, Liu JJ, et al., 2022. A survey on space-air-ground-sea integrated network security in 6G. *IEEE Commun Surv Tutor*, 24(1):53-87. https://doi.org/10.1109/COMST.2021.3131332

Guo Q, Tang FX, Kato N, 2023. Resource allocation for aerial assisted digital twin edge mobile network. *IEEE J Sel Areas Commun*, 41(10):3070-3079. https://doi.org/10.1109/JSAC.2023.3310065

Guo Q, Tang FX, Rodrigues TK, et al., 2024. Five disruptive technologies in 6G to support digital twin networks. *IEEE Wirel Commun*, 31(1):149-155. https://doi.org/10.1109/MWC.013.2200296

Guo ZY, Chen ZY, Liu P, et al., 2022. Multi-agent reinforcement learning-based distributed channel access for next generation wireless networks. *IEEE J Sel Areas Commun*, 40(5):1587-1599. https://doi.org/10.1109/JSAC.2022.3143251

Hui YL, Qiu Y, Su Z, et al., 2023. Digital twins for intelligent space-air-ground integrated vehicular network: challenges and solutions. *IEEE Int Things Mag*, 6(3):70-76. https://doi.org/10.1109/IOTM.001.2300067

Ji Z, Wu S, Jiang CX, 2023. Cooperative multi-agent deep reinforcement learning for computation offloading in digital twin satellite edge networks. *IEEE J Sel Areas Commun*, 41(11):3414-3429. https://doi.org/10.1109/JSAC.2023.3313595

Jiang XY, Zhang T, Liu L, 2023. Research on satellite QoS routing algorithm based on digital twin. Proc 11th Int Conf on Intelligent Computing and Wireless Optical Communications, p.118-122. https://doi.org/10.1109/ICWOC57905.2023.10200404

Kato N, Fadlullah ZM, Tang FX, et al., 2019. Optimizing space-air-ground integrated networks by artificial intelligence. *IEEE Wirel Commun*, 26(4):140-147. https://doi.org/10.1109/MWC.2018.1800365

Li XN, Zhang HJ, Zhou H, et al., 2023. Multi-agent DRL for resource allocation and cache design in terrestrial-satellite networks. *IEEE Trans Wirel Commun*, 22(8):5031-5042. https://doi.org/10.1109/TWC.2022.3231379

Liu JJ, Shi YP, Zhao L, et al., 2018a. Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network. *IEEE J Sel Areas Commun*, 36(2):221-232. https://doi.org/10.1109/JSAC.2018.2804019

Liu JJ, Shi YP, Fadlullah ZM, et al., 2018b. Space-air-ground integrated network: a survey. *IEEE Commun Surv Tutor*, 20(4):2714-2741. https://doi.org/10.1109/COMST.2018.2841996

Liu T, Tang L, Wang WL, et al., 2022. Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network. *IEEE Int Things J*, 9(2):1427-1444. https://doi.org/10.1109/JIOT.2021.3086961

Lu YL, Maharjan S, Zhang Y, 2021a. Adaptive edge association for wireless digital twin networks in 6G. *IEEE Int Things J*, 8(22):16219-16230. https://doi.org/10.1109/JIOT.2021.3098508

Lu YL, Huang XH, Zhang K, et al., 2021b. Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks. *IEEE Trans Ind Inform*, 17(7):5098-5107. https://doi.org/10.1109/TII.2020.3017668

Mao BM, Zhou XM, Liu JJ, et al., 2024. Digital twin satellite networks toward 6G: motivations, challenges, and future perspectives. *IEEE Netw*, 38(1):54-60. https://doi.org/10.1109/MNET.2023.3332895

Sun W, Zhang HB, Wang R, et al., 2020. Reducing offloading latency for digital twin edge networks in 6G. *IEEE Trans Veh Technol*, 69(10):12240-12251. https://doi.org/10.1109/TVT.2020.3018817

Sunehag P, Lever G, Gruslys A, et al., 2018. Value-decomposition networks for cooperative multi-agent learning based on team reward. Proc 17th Int Conf on Autonomous Agents and Multiagent Systems, p.2085-2087.

Tan X, Zhou L, Wang HJ, et al., 2022. Cooperative multi-agent reinforcement-learning-based distributed dynamic spectrum access in cognitive radio networks. *IEEE Int Things J*, 9(19):19477-19488. https://doi.org/10.1109/JIOT.2022.3168296

Tang FX, Chen XH, Rodrigues TK, et al., 2022. Survey on digital twin edge networks (DITEN) toward 6G. *IEEE Open J Commun Soc*, 3:1360-1381. https://doi.org/10.1109/OJCOMS.2022.3197811

Tang M, Wong VWS, 2022. Deep reinforcement learning for task offloading in mobile edge computing systems. *IEEE Trans Mob Comput*, 21(6):1985-1997. https://doi.org/10.1109/TMC.2020.3036871

Tang QQ, Fei ZS, Li B, et al., 2021. Computation offloading in LEO satellite networks with hybrid cloud and edge computing. *IEEE Int Things J*, 8(11):9164-9176. https://doi.org/10.1109/JIOT.2021.3056569

Tao F, Zhang H, Liu A, et al., 2019. Digital twin in industry: state-of-the-art. *IEEE Trans Ind Inform*, 15(4):2405-2415. https://doi.org/10.1109/TII.2018.2873186

Vaezi M, Noroozi K, Todd TD, et al., 2023. Digital twin placement for minimum application request delay with data age targets. *IEEE Int Things J*, 10(13):11547-11557. https://doi.org/10.1109/JIOT.2023.3244424

Wang GC, Zhou S, Zhang S, et al., 2020. SFC-based service provisioning for reconfigurable space-air-ground integrated networks. *IEEE J Sel Areas Commun*, 38(7):1478-1489. https://doi.org/10.1109/JSAC.2020.2986851

Wang P, Zhang JX, Zhang X, et al., 2020. Convergence of satellite and terrestrial networks: a comprehensive survey. *IEEE Access*, 8:5550-5588. https://doi.org/10.1109/ACCESS.2019.2963223

Wu YW, Zhang K, Zhang Y, 2021. Digital twin networks: a survey. *IEEE Int Things J*, 8(18):13789-13804. https://doi.org/10.1109/JIOT.2021.3079510

Yao ZX, Xia SC, Li Y, et al., 2023. Cooperative task offloading and service caching for digital twin edge networks: a graph attention multi-agent reinforcement learning approach. *IEEE J Sel Areas Commun*, 41(11):3401-3413. https://doi.org/10.1109/JSAC.2023.3310080

Yin ZS, Cheng N, Luan TH, et al., 2023. DT-assisted multi-point symbiotic security in space-air-ground integrated networks. *IEEE Trans Inform Forens Secur*, 18:5721-5734. https://doi.org/10.1109/TIFS.2023.3313326

Zhang H, Luo TX, Wang QQ, 2023. Adaptive digital twin server deployment for dynamic edge networks in IoT system. IEEE/CIC Int Conf on Communications in China, p.1-6. https://doi.org/10.1109/ICCC57788.2023.10233465

Zhang JX, Wang KW, Li R, et al., 2023. MaCro: mega-constellations routing systems with multi-edge cross-domain features. *IEEE Wirel Commun*, 30(6):69-76. https://doi.org/10.1109/MWC.006.2300216

Zhang K, Cao JY, Zhang Y, 2022. Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks. *IEEE Trans Ind Inform*, 18(2):1405-1413. https://doi.org/10.1109/TII.2021.3088407

Zhang YD, Zhang HB, Lu YL, et al., 2024. Adaptive digital twin placement and transfer in wireless computing power network. *IEEE Int Things J*, 11(6):10924-10936. https://doi.org/10.1109/JIOT.2023.3328380

Zhang ZJ, Zhang WY, Tseng FH, 2019. Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques. *IEEE Netw*, 33(1):70-76. https://doi.org/10.1109/MNET.2018.1800172

Zhang ZQ, Xiao Y, Ma Z, et al., 2019. 6G wireless networks: vision, requirements, architecture, and key technologies. *IEEE Veh Technol Mag*, 14(3):28-41. https://doi.org/10.1109/MVT.2019.2921208

Zhao L, Wang CC, Zhao KL, et al., 2022. INTERLINK: a digital twin-assisted storage strategy for satellite-terrestrial networks. *IEEE Trans Aerosp Electron Syst*, 58(5):3746-3759. https://doi.org/10.1109/TAES.2022.3169130

Zhou YK, Zhang R, Liu J, et al., 2023. A hierarchical digital twin network for satellite communication networks. *IEEE Commun Mag*, 61(11):104-110. https://doi.org/10.1109/MCOM.001.2200864

Zhu XM, Jiang CX, Kuang LL, et al., 2019. Cooperative transmission in integrated terrestrial-satellite networks. *IEEE Netw*, 33(3):204-210. https://doi.org/10.1109/MNET.2018.1800164