Proceedings of Machine Learning Research vol 278:1–15, 2025 2025 2nd International Conference on Machine Learning and Intelligent Computing

Digital Twin-Assisted Satellite-Ground Cooperative Edge Network Resource Optimization Method

Shuopeng Li* LISHUOPENG@BJUT.EDU.CN and **Ran Han** HANRAN@EMAILS.BJUT.EDU.CN
College of Computer Science, Beijing University of Technology, Beijing, China

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

To address the challenge of efficiently processing data on resource-constrained IoT devices, this paper proposes a digital twin architecture for satellite-terrestrial collaborative edge networks and introduces Coordinated Constrained DDPG (CC_DDPG), a deep reinforcement learning (DRL)-based task offloading and resource allocation algorithm tailored for model training tasks. First, digital twin models for UAVs and satellites are constructed to enable real-time network state monitoring and decision support. Second, the joint optimization of task offloading, communication, and computing resources is formulated as a Markov Decision Process (MDP). By enhancing the actor network in the conventional DDPG method, the proposed algorithm dynamically balances training latency and energy consumption. Simulation results demonstrate that CC_DDPG significantly outperforms benchmark heuristic algorithms in convergence stability and multi-objective optimization performance.

Keywords: digital twin, satellite edge calculation, computing offloading, resource allocation, deep reinforcement learning.

1. Introduction

With the rapid development of 6G networks, the Internet of Things (IoT), and edge computing technologies, the Digital Twin Network (DTN), as a network architecture integrating virtual and physical elements, has shown tremendous potential in dynamic resource optimization, network performance improvement, and intelligent operation and maintenance. Currently, many studies have applied DTN technology to tasks such as computation offloading (Liu et al., 2022) and resource management (Hui et al., 2023). Sun et al. (2020) in the digital twin edge network utilized the digital twin (DT) model of the entire MEC system to provide training data for offloading decisions and proposed a DRL-based mobile offloading scheme to minimize offloading latency. Guo et al. (2023) introduced DT technology into airborne computing networks, investigating the deployment and resource allocation issues of intelligent UAVs, and proposed two DT-assisted hybrid task offloading schemes. Hazarika et al. (2023) proposed a DT-driven framework for UAV-assisted IoT networks and introduced a multi-network deep reinforcement learning-based resource allocation algorithm in the DT-assisted network, optimizing task offloading strategies while maximizing the utility of vehicular networks. Peng et al. (2023) proposed the use of DT-enabled edge networks, where resource allocation strategies are determined by the interactions between DTs, achieving efficient resource distribution and computation offloading.

The aforementioned studies focus on the application of DTN in terrestrial networks. However, with the development of satellite communication technology, the application of DTN in satellite networks has gradually gained attention. Combined with artificial intelligence algorithms, DTNs can analyze historical data and data generated by DT networks to predict future behavior, optimize performance, and improve reliability. To address the challenges of deploying DTs in networks, Tao et al. (2025) proposed the deployment of multi-layer DTs in satellite-ground integrated networks to reduce system latency and explored the optimal strategy for multi-layer DT deployment using a multi-agent reinforcement learning approach. Zhou et al. (2023) designed a hierarchical digital twin network for satellite communication, defining different types of DT models and proposing three methods to improve model synchronization efficiency. Ji et al. (2023) designed a digital twin satellite-ground cooperative edge computing framework and, addressing DT layer issues, proposed a centralized multi-agent DRL algorithm to optimize resource sharing between adjacent satellites. Bui et al. (2024) built a virtual digital twin network based on micro-cloud architecture for mapping data from physical satellite networks, enabling dynamic acquisition and updating of satellite network resource states.

Currently, research on satellite-ground cooperative edge digital twin networks is limited, and there is a lack of efficient resource allocation algorithms specifically designed for model training tasks in this context. Therefore, this paper proposes a digital twin-assisted satellite-ground cooperative edge network resource allocation scheme.

The main research contributions are as follows:

- A digital twin architecture for satellite-ground cooperative edge networks is proposed. Based on this architecture, for satellite-ground cooperative model training tasks, blockchain technology is used to achieve secure and efficient model aggregation. The computational and communication resource allocation problem in this process is modeled as a constrained collaborative optimization problem for latency and energy consumption.

- A task offloading and resource allocation algorithm CC_DDPG based on DRL is proposed. By improving the actor network in the DDPG method, the algorithm achieves collaborative handling of different constraints, reducing latency and energy consumption during the model training process.

- The convergence and stability of the proposed algorithm are evaluated through simulations, demonstrating its superior performance compared to heuristic algorithms and other benchmark methods.

2. System Model

As shown in Figure 1, this paper proposes a digital twin architecture for satellite-ground cooperative edge networks. Based on this architecture, for satellite-ground cooperative model training tasks, blockchain technology is used to achieve model training and aggregation.

2.1. Physical Space

In the physical entity space, the edge layer consists of satellites with the same computational capabilities, forming set $J = \{1, 2, \cdots, j, \cdots, M\}$. These satellites cover N UAV devices in the terminal layer, forming set $I = \{1, 2, \cdots, i, \cdots, N\}$. For a specific task, UAVs collect different data sets D, and based on the real-time association relationship $\varphi_{i,j}$ with the satellites, upload data set D to the associated satellites. Here, $\varphi_{i,j}$ is a binary decision variable that indicates whether the task data of UAV i is offloaded to satellite j. If the task data of UAV i is offloaded to satellite j, the value is 1; otherwise, it is 0.

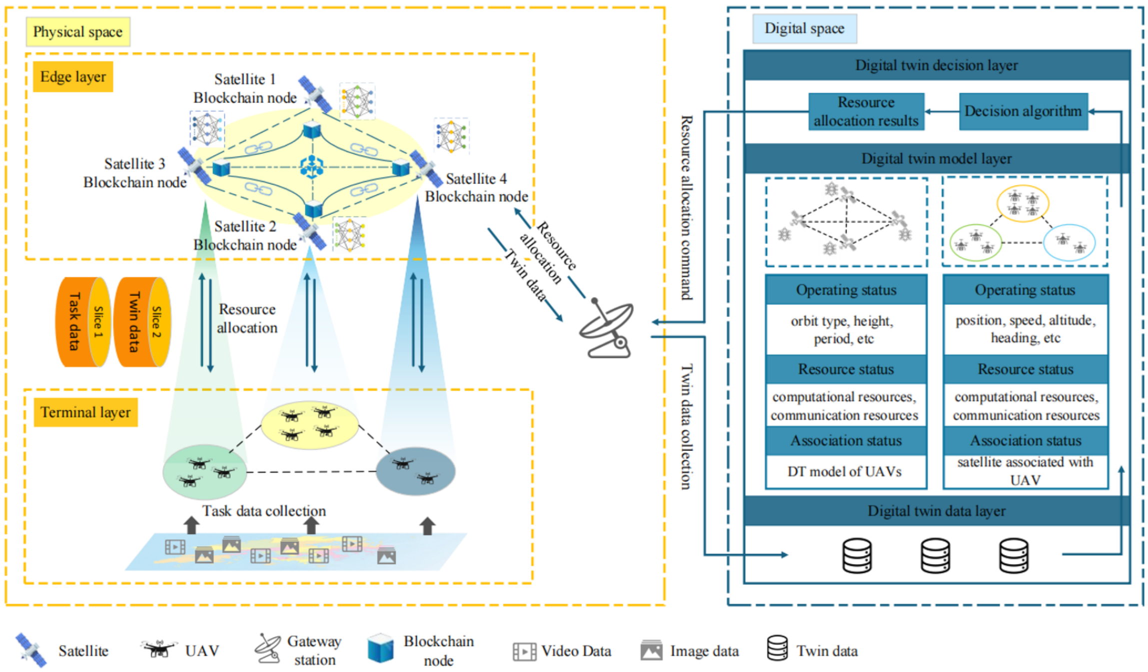

Figure 1: Digital twin architecture of satellite-ground cooperative edge network.

2.2. Digital Twin Space

In the digital twin space, this paper constructs DT models for UAVs and satellites by leveraging their resource status and relational states.

2.2.1. DT MODEL OF UAVs

The DT model of UAV i is represented as follows:

$$DT_i^u = (D_i, \xi_i, P_u, \varphi_{i,j}), \forall i \in I, \forall j \in J \tag{1}$$

Where D_i is the data set size preprocessed by UAV i, ξ_i is the number of channels allocated to UAV i, P_u is the fixed transmission power of UAV, and $\varphi_{i,j}$ is the association relationship between UAV i and satellite j.

2.2.2. DT MODEL OF SATELLITE

For satellite j, it can be similarly represented as:

$$DT_j^s = (f_j, DT_{i,j}, N_j), \forall j \in J, \forall i \in I \tag{2}$$

Where f_j is the computational capacity of satellite j, $DT_{i,j}$ is the set of DT models of UAV i mapped to satellite j, and N_j is the number of UAV DT models mapped to satellite j.

2.3. Communication Model

In the satellite-ground user link, UAVs use the Orthogonal Frequency Division Multiple Access (OFDMA) technology to transmit data to satellites, without giving much consideration to the impact of other factors on communication. The number of channels allocated to UAV i, denoted as ξ_i, satisfies $0 < \xi_i \leq \xi_{total}$, where ξ_{total} is the total number of channels. The transmission rate of UAV i can be denoted as:

$$r_i = \xi_i B_{up} \log_2 \left(1 + \frac{h_i P_u}{N_0} \right) \tag{3}$$

Where B_{up} is the transmission bandwidth of a single uplink channel, N_0 is the noise power, h_i is the channel gain, and P_u is the transmission power of UAV i.

The inter-satellite link uses laser communication technology, with a fixed rate $r_{\text{int}\,er}$.

2.4. Delay Model

The total delay of UAVs during the global model training process includes communication delay and computation delay.

2.4.1. COMMUNICATION DELAY

The communication delay consists of the uplink data transmission delay, inter-satellite link model propagation delay, blockchain propagation delay, and downlink model transmission delay.

The uplink transmission delay of UAV i is represented as:

$$T_{up}^i = \frac{D_i}{r_i} \tag{4}$$

The preprocessed dataset is processed by the satellite edge neural network to generate local model parameters, which are then propagated and cross-validated through blockchain nodes (satellites). According to the Gossip protocol (Jiang et al., 2020), the model parameter propagation delay $T_{\text{int}\,er}$ is determined by the number of satellite nodes, inter-satellite transmission rate $r_{\text{int}\,er}$, and parameter size ω. $T_{\text{int}\,er}$ can be expressed as:

$$T_{\text{int}\,er} = (\log_2 M) \frac{\omega \max\limits_j (N_j)}{r_{\text{int}\,er}} + (\log_2 M_g) \frac{D_B}{r_{\text{int}\,er}} \tag{5}$$

Where M_g is the number of block producers, and D_B is the size of each block.

The model transmission delay T_{down} for the downlink can be denoted as ω / r_{down}. Therefore, the total communication delay T_{trans} can be expressed as:

$$T_{trans} = \max_i (T_{up}^i) + T_{\text{int}\,er} + T_{down} \tag{6}$$

2.4.2. COMPUTATION DELAY

The computation delay includes the local model training delay and the block processing delay.

After the satellites receive the dataset uploaded by the associated UAVs, local model training is performed, and its processing delay can be modeled as:

$$T_{train} = \max_{j} \left\{ \sum_{i=1}^{N} \varphi_{i,j} \frac{D_i C_j}{(\rho_{j,i} + \varepsilon) f_j} \right\} \tag{7}$$

Where $\rho_{j,i}$ is the proportion of computing resources allocated by satellite j to DT_i, $\rho_j = \sum_{i=1}^{N} \rho_{j,i}$ is the total proportion of computing resources allocated by satellite j to the DTs, satisfying $\rho_j \leq 1$, and C_j is determined by satellite j's CPU chip, representing the number of CPU cycles required to train a unit data sample. is a very small constant to prevent division by zero.

The block processing delay refers to the time spent by a node to validate and add the received block to the blockchain. During this phase, the node verifies the integrity of the block, checks the validity of the transactions, and executes a consensus algorithm to ensure network consensus. The complexity of the validation process and the computational power of the node both impact the processing delay. The block processing delay can be simplified as:

$$T_{block} = \max_{j} \frac{D_B f_{block}}{f_j} \tag{8}$$

Where f_{block} is the number of CPU cycles required to process a unit block.

Therefore, in the global model training process, the total computation delay can be expressed as:

$$T_{com} = T_{train} + T_{block} \tag{9}$$

In summary, the total delay can be expressed as:

$$T_{total} = T_{trans} + T_{com} \tag{10}$$

2.5. Energy Consumpyion Model

The total energy consumption considers the transmission energy consumption from UAVs to the satellites, the hovering energy consumption throughout the process, and the computation energy consumption for satellite model training.

2.5.1. Transmission energy consumption of UAVs

The data transmission energy consumption of UAV i can be expressed as:

$$E_i^{com} = \frac{D_i P_u}{r_i} \tag{11}$$

2.5.2. Hover energy consumption of UAVs

The hovering energy consumption of UAV i exhibits a linear relationship with its hovering duration. The total hovering energy consumption E_i^{hover} can be expressed as:

$$E_i^{hover} = P_{hover} T_{total} \tag{12}$$

Where P_{hover} is the power of the UAV.

In summary, the total energy consumption of all UAVs can be expressed as:

$$E_{UAV} = \sum_{i=1}^{N} \left(E_i^{com} + E_i^{hover} \right) \tag{13}$$

2.5.3. THE COMPUTATIONAL ENERGY CONSUMPTION OF SATELLITES

This paper models the CPU power consumption of satellite j as $P_j = \alpha_j \left(\sum_{i=1}^{N} \rho_{j,i} f_j \right)^3$, where α_j represents the effective capacitance coefficient of satellite j. The edge computing energy consumption of satellite j is:

$$E_j^{cmp} = P_j T_j = \alpha_j C_j \left(\sum_{i=1}^{N} D_i (\rho_{j,i} f_j)^2 \right) \tag{14}$$

$$T_j = \frac{C_j \sum_{i=1}^{N} D_i}{\sum_{i=1}^{N} \rho_{j,i} f_j} \tag{15}$$

Where T_j is the time required by satellite j to process the datasets of all associated UAVs.

In summary, the total energy consumption in the global model training process can be expressed as:

$$E_{total} = \sum_{i=1}^{N} \left(E_i^{com} + E_i^{hover} \right) + \sum_{j=1}^{M} E_j^{cmp} \tag{16}$$

2.6. System Joint Optimization Problem Model

$$\begin{aligned}
&\min_{\xi,\rho,\varphi} \, (1-\mu) \frac{T_{total}}{T_{\max}} + \mu \frac{E_{total}}{E_{\max}} \\
&\text{s.t. } \mathrm{C}_1 : \mu \in [0,1] \\
&\quad \mathrm{C}_2 : \sum_{i=1}^{N} \rho_{j,i} \leq 1, \forall j \in J \\
&\quad \mathrm{C}_3 : \sum_{j=1}^{M} N_j = N, j \in J \\
&\quad \mathrm{C}_4 : \sum_{j=1}^{M} \varphi_{i,j} = 1, \forall i \in I \\
&\quad \mathrm{C}_5 : \sum_{i=1}^{N} \xi_i \leq \xi_{total} \\
&\quad \mathrm{C}_6 : T_{total} \leq T_{th} \\
&\quad \mathrm{C}_7 : E_{UAV} \leq E_{th}
\end{aligned} \tag{17}$$

The research goal of this paper is to minimize the delay and energy consumption during the global model training process by reasonably allocating channel and computational resources. The optimization objective is defined as (17).

Where ξ, ρ, and φ are the decision variables, representing the channel allocation decision, computational resource allocation decision, and the association relationship between UAVs and satellites, respectively. $T_{\max}$ and $E_{\max}$ represent the maximum values of total delay and total energy consumption, and constraint C_1 is the constraint on weight coefficients. μ is the weight coefficient for energy consumption. C_2 indicates that the total computational resources allocated to the associated data for any satellite should be less than its total computational resources. C_3 ensures that the total number of DTs across all satellites is equal to the number of UAV devices. C_4 ensures that the data from any UAV is uploaded to only one satellite. C_5 restricts the allocation of channels, meaning that the total number of channels allocated to UAVs should be less than the total available channels. C_6 and C_7 ensure that the total delay and energy consumption of UAVs during the global model training process are better than the set thresholds.

3. Algorithm Design

3.1. MDP Model

The optimization problem (17) is an NP-hard optimization problem, which is difficult to solve using classical convex optimization algorithms. In this paper, the joint objective optimization problem in the satellite-ground network is modeled as a MDP, and a DRL-based task offloading and resource allocation algorithm, called CC_DDPG, is proposed. By reasonably allocating the number of channels and computational resources, the algorithm reduces the delay and energy consumption during the global model training process.

The MDP model of this optimization problem can be represented as a tuple (S, A, R, γ), where S is the state space of the environment, A is the action space, R is the reward function, γ is the discount factor of the reward function. The actions taken by the agent include the allocation of the number of channels ξ, the allocation of computational resources ρ, and the offloading decision φ.

3.1.1. STATE SPACE

The state space S of the agent describes the data size of all UAVs within the satellite coverage. The environmental state at time slot t can be represented as:

$$S_t = \left\{D_1^t, D_i^t, ..., D_N^t\right\} \tag{18}$$

Where D_i^t is the data size of UAV i at time slot t.

3.1.2. ACTION SPACE

The action space contains all possible actions taken by the agent. In the scenario proposed in this paper, for a specific task, each UAV is assigned a corresponding number of channels and computing resource ratio, and the offloading decision depends on the distribution of computing resources. The global action taken by the agent at time slot t can be represented as:

$$A_t = \left\{\xi_t, \rho_1^t, \rho_j^t, ..., \rho_M^t, \varphi_1^t, \varphi_i^t, ..., \varphi_N^t\right\} \tag{19}$$

Where $\xi_t = \left(\xi_1^t, \xi_2^t, ...\xi_N^t\right)$ is the channel allocation for each UAV at time slot t, $\rho_j^t = \left(\rho_{j,1}^t, \rho_{j,2}^t, ...\rho_{j,N}^t\right)$ is the computation resource allocation ratio for each UAV by satellite j at time slot t, and $\varphi_i^t = \left(\varphi_{i,1}^t, \varphi_{i,j}^t, ...\varphi_{i,M}^t\right)$ is the association relationship between UAV i and the satellite j at time slot t.

3.1.3. REWARD

Resource allocation and offloading decisions aim to minimize system delay and energy consumption in the satellite-ground cooperative edge network. The agent dynamically adjusts its strategy based on the environmental state, calculates the total delay and energy consumption, and generates a reward value through weighted fusion. Ultimately, the delay-energy joint optimization reward function is designed as shown in equation (17):

$$R_t = -(1-\mu)\frac{T_{total}}{T_{\max}} - \mu\frac{E_{total}}{E_{\max}} \tag{20}$$

3.2. CC_DDPG Algorithm Framework

The action space includes discrete decision variables ξ and φ, as well as continuous variable ρ. In this paper, the DDPG algorithm, which supports continuous actions, is adopted. This algorithm combines the experience replay and target network mechanisms of deep Q-networks to enhance convergence speed and stability. However, its offline nature leads to insufficient sampling efficiency. Traditional penalty constraint methods tend to generate a large number of invalid samples, further decreasing sampling efficiency. To address this, the paper proposes the CC_DDPG algorithm (Figure 2), which innovatively designs the actor network structure within the actor-critic framework of DDPG. The constraint generation mechanism ensures sample validity, and learnable parameters are introduced to cover the feasible solution space, enabling precise handling of the constraints in equation (17) and effectively improving algorithm performance in complex tasks.

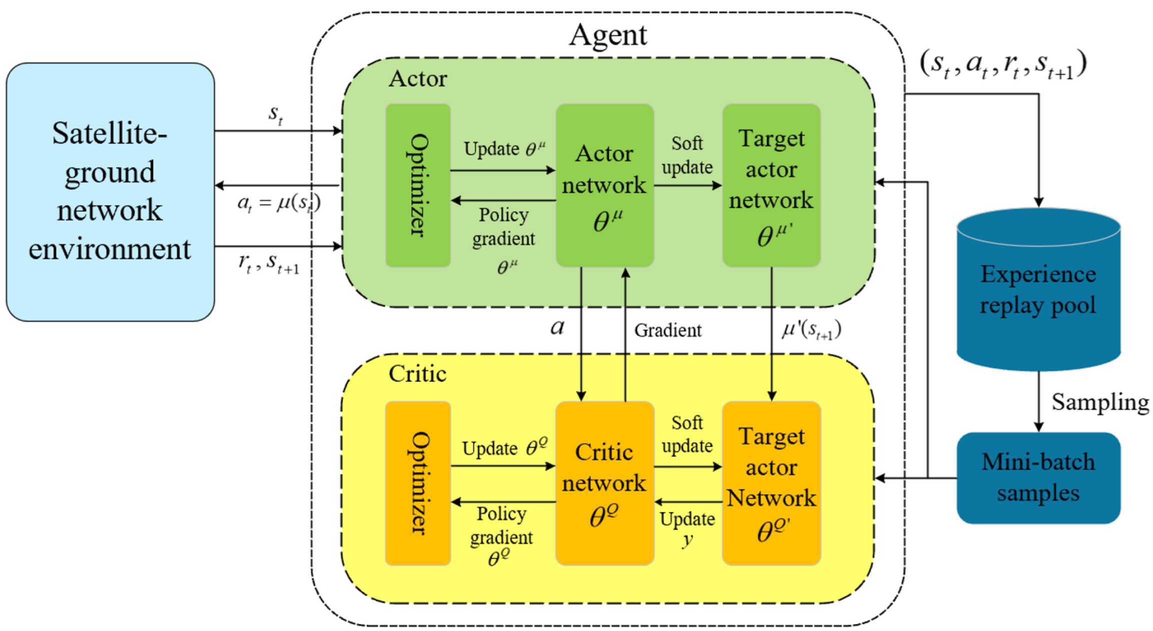

Figure 2: CC_DDPG algorithm model.

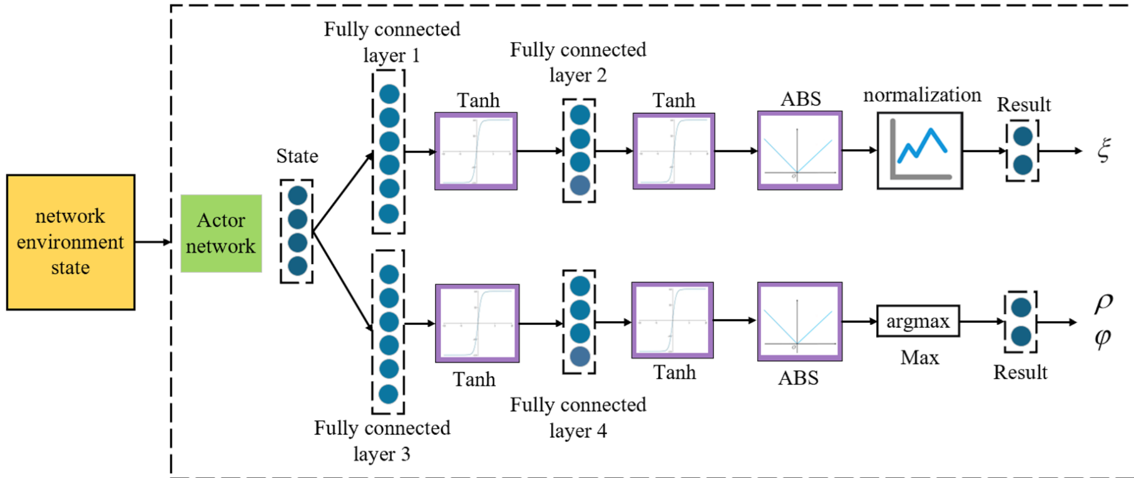

Figure 3: Actor neural network architecture.

3.3. Design of the Actor Neural Network Architecture

The actor neural network architecture of the CC_DDPG algorithm is shown in Figure 3. It consists of two parts:

3.3.1. NEURAL NETWORK DESIGN FOR ξ

First, feature extraction is performed through two fully connected layers and two tanh activation functions:

$$\xi_{\text{tanh}} = \text{Tanh}\left(W_1 h_1 + b_1\right) \tag{21}$$

Where ξ_{tanh} is the output of the second hidden layer, W_1 is the weight matrix, and b_1 is the bias term.

Next, the absolute value of ξ is taken and normalized:

$$\xi_{norm} = \frac{|\xi_{\text{tanh}}|}{\sum\limits_{i=1}^{N} \xi_{raw,i}} \tag{22}$$

Finally, it is multiplied by a learnable parameter α_0 to control the overall allocation intensity of ξ, in order to satisfy condition C_5 in equation (17).

$$\xi = \alpha_0 \xi_{norm} \tag{23}$$

3.3.2. NEURAL NETWORK DESIGN FOR ρ AND φ

First, the raw state data is processed through two fully connected layers and two tanh activation functions to obtain the feature representation ρ_{tanh}, and then the absolute value of ρ_{tanh} is taken to obtain ρ_{res}:

$$\rho_{\text{tanh}} = \text{Tanh}\left(W_2 h_2 + b_2\right) \tag{24}$$

$$
\begin{aligned}
\rho_{res} &= |\rho_{\tanh}| \\
&= [\rho_{res,1}, \rho_{res,2}, \ldots, \rho_{res,N}] \\
&= \begin{pmatrix} \rho_{11} & \cdots & \rho_{1N} \\ \vdots & \ddots & \vdots \\ \rho_{M1}, & \cdots & a_{MN} \end{pmatrix}
\end{aligned} \tag{25}
$$

Where $\rho_{\tanh}$ is the output of the second hidden layer, W_2 is the weight matrix, and b_2 is the bias term.

To determine the value of φ, the algorithm performs a column-wise maximum index operation to extract and store the optimal computation resource allocation for each UAV device. While determining the φ matrix, the operation processes ρ_{res} to obtain $\rho_{\max}$, as shown in equations (26) and (27). This operation ensures that the constraints C_3 and C_4 in equation (17) are satisfied.

$$
\rho_{\max}[x,y] = \begin{cases} \rho_{xy}, & \text{if } \rho_{xy} = \max\left(\rho_{\text{res}}[:,y]\right) \\ 0, & \text{otherwise} \end{cases}, \forall x \in [1,M], \forall y \in [1,N] \tag{26}
$$

$$
\varphi[x,y] = \begin{cases} 1, & \text{if } \rho_{\max}[x,y] \neq 0 \\ 0, & \text{otherwise} \end{cases}, \forall x \in [1,M], \forall y \in [1,N] \tag{27}
$$

Where x is the row index and y is the column index.

Next, perform row-wise normalization on matrix $\rho_{\max}$ to ensure $\sum\limits_{y=1}^{N} \rho_{xy} = 1, \forall x \in [1M]$, which guarantees that the satellite's allocation of computational resources does not exceed its own available resources:

$$
\rho_{\text{norm}}[x,y] = \frac{\rho_{\max}[x,y]}{\sum_{y=1}^{N} \rho_{\max}[x,y]}, \forall x \in [1,M], \forall y \in [1,N] \tag{28}
$$

Finally, multiply each row of matrix ρ_{norm} by a different learnable parameter to generate feasible solutions that satisfy the C_2 constraint:

$$
\rho = \alpha \rho_{norm}[x,y], \forall x \in [1,M], \forall y \in [1,N] \tag{29}
$$

Where $\alpha = [\alpha_1, \alpha_2, ..., \alpha_M]$, α_j are the learnable parameters associated with satellite j.

4. Simulation Analysis

4.1. Simulation Setup

In this paper, we consider a scenario where 3 satellites act as edge servers within a coverage radius of 5 km, providing data processing and model aggregation services for 6 to 11 UAVs on the ground. The system parameters are set as $B_{up} = 2\text{MHz}$, $\xi_{total} = 32$, $D_i = [100, 1000]\,\text{MB}$, $p_i = 1\text{W}$, $P_{hover} = 100\text{W}$, $f_j = 4\text{GHz}$, $D_B = 15\text{MB}$, $r_{\text{int}\,er} = 1\text{Gbps}$, $C_j = 10^2$, $\alpha_j = 10^{-25}$.

This paper compares the CC_DDPG algorithm with the Genetic Algorithm (GA) and the Random Algorithm (RA), where the RA randomly selects feasible solutions within the constraint range and calculates their objective values, while the GA iteratively optimizes solutions through selection, crossover, and mutation based on the RA. Three experiments are conducted to evaluate the effectiveness of the CC_DDPG algorithm: (1) convergence analysis, (2) evaluating the effect of varying the number of UAVs, and (3) adjusting the energy consumption weight coefficient. The algorithm is evaluated using three metrics: average task reward, average task delay, and average task energy consumption.

4.2. Convergence Analysis of CC_DDPG

Figure 4 illustrates the convergence of the CC_DDPG algorithm during the training phase. The CC_DDPG algorithm converges quickly and maintains stability in a short training time, demonstrating its ability to effectively solve resource allocation problems with complex constraints.

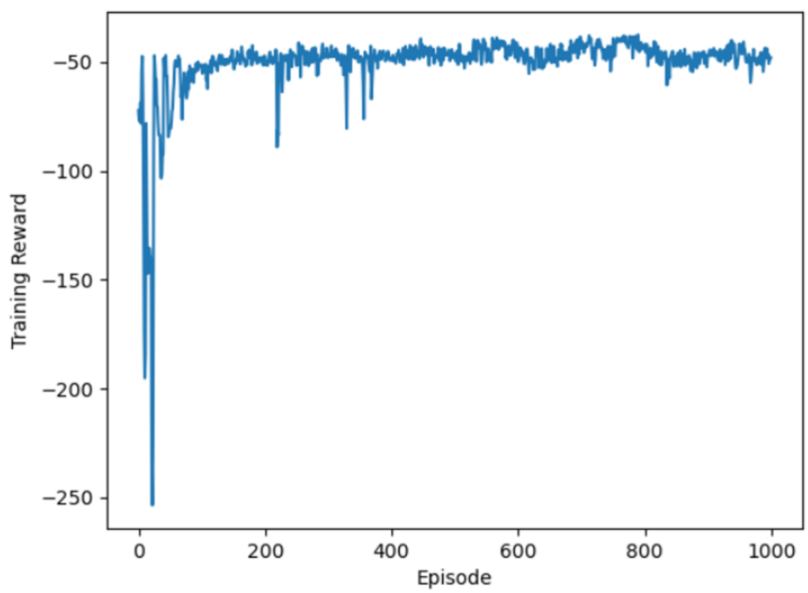

Figure 4: The cumulative reward of each episode during the training phase.

4.3. The Impact of the Number of UAVs on Algorithm Performance

Figure 5 shows that as the number of UAVs increases, both the average task delay and average energy consumption increase, while the average task reward decreases. The CC_DDPG algorithm outperforms the GA and RA algorithms in terms of average task reward, delay, and energy consumption. When the number of UAVs reaches its maximum, the average task reward of the CC_DDPG algorithm is 33.7% higher than that of GA and 71.0% higher than that of RA. Additionally, the average task delay is 25.1% lower than GA and 60.5% lower than RA, while the average task energy consumption is 57.0% lower than GA and 75.1% lower than RA. These results demonstrate that the CC_DDPG algorithm is more effective in optimizing resource allocation and task offloading strategies, offering superior performance, particularly in the complex satellite-ground collaborative edge network resource optimization scenario.

4.4. The Impact of weight coefficients on Algorithm Performance

Figure 6 illustrates the significant performance advantages of the CC_DDPG algorithm under various energy weight coefficients. The average task reward improves by 21.5%-81.6% compared to

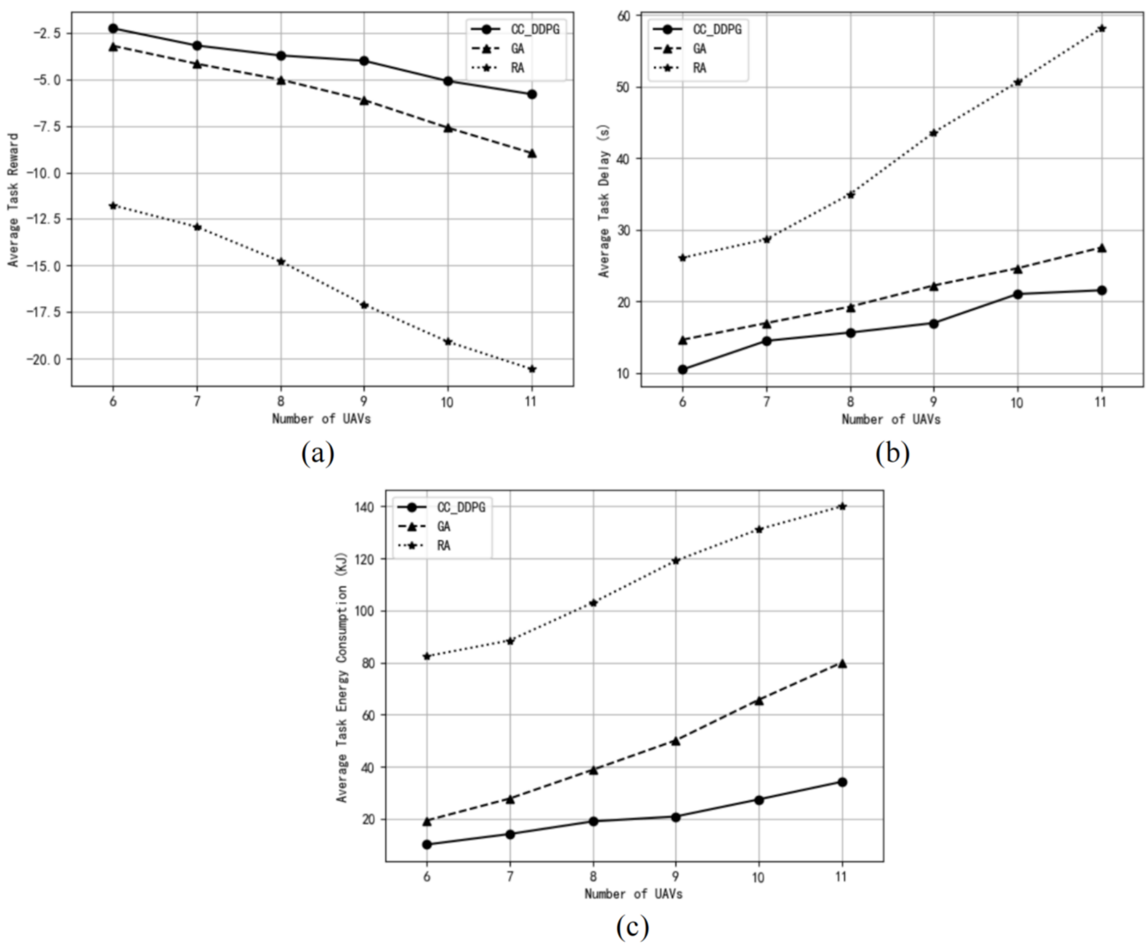

Figure 5: The impact of the number of UAVs on the performance of different algorithms: (a) average task reward, (b) average task delay, (c) average task energy consumption.

benchmark algorithms, with the reward decrease rate being the least steep (only one-third of that of the RA algorithm) as the energy weight increases. The CC_DDPG algorithm reduces the average task delay by 15.3%-25.0% compared to the GA algorithm and by 45.4%-70.0% compared to the RA algorithm. Additionally, the average task energy consumption decreases by 46.2%-52.6% compared to GA and by 72.9%-81.7% compared to RA. Particularly when the energy weight reaches 0.99, the algorithm maintains its delay optimization capability (45.4% lower than RA), indicating the proposed algorithm's strong adaptability in multi-objective optimization. It effectively balances the trade-off between energy consumption and delay, while achieving optimal resource scheduling under dynamic weights.

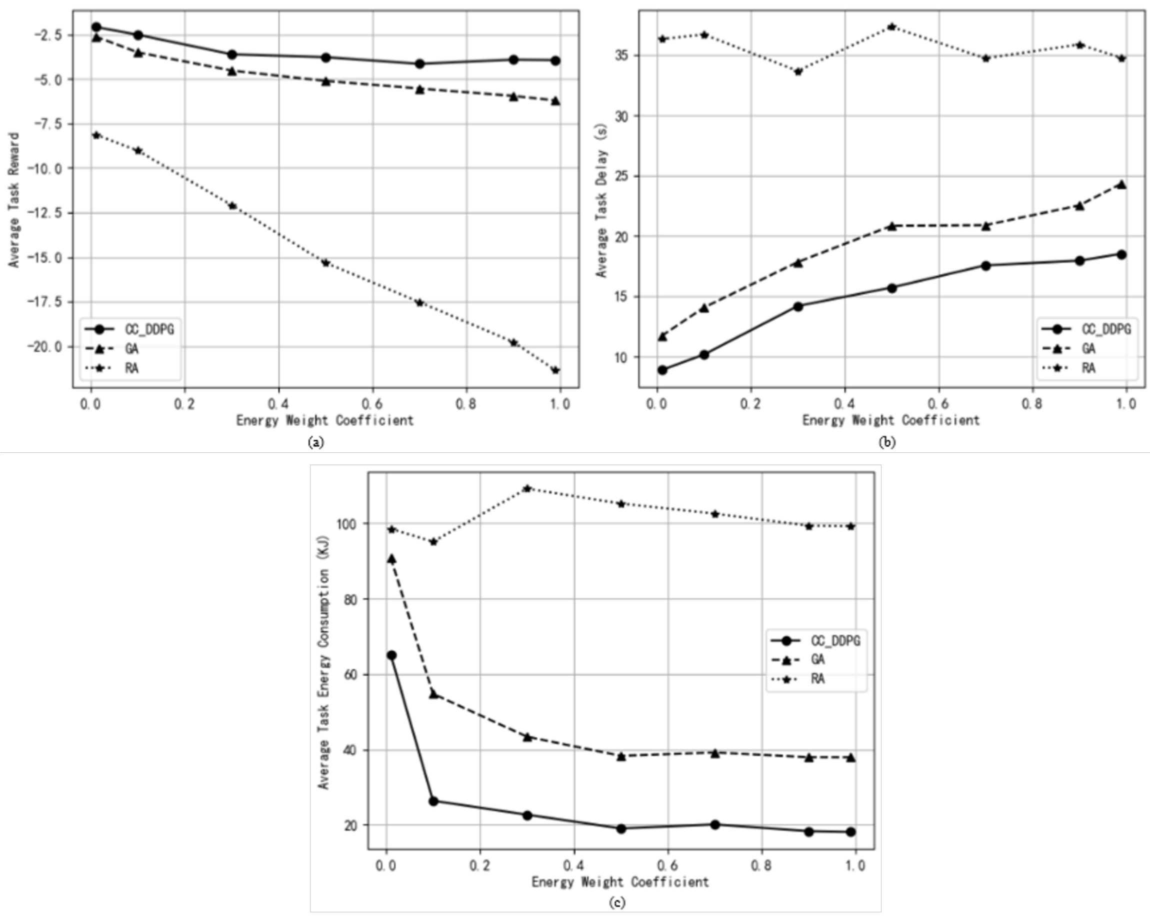

Figure 6: The impact of weight coefficients on the performance of different algorithms: (a) average task reward, (b) average task delay, (c) average task energy consumption.

5. Conclusion

In this paper, we propose a digital twin architecture for satellite-ground collaborative edge networks, focusing on model training tasks, and introduce a DRL-based task offloading and resource allocation algorithm, CC_DDPG. The algorithm effectively reduces system delay and energy consumption by optimizing offloading decisions, bandwidth allocation, and computational resource distribution. Simulation results show that the proposed algorithm outperforms benchmark algorithms in performance.

Acknowledgments

Foundation Items: Hainan Provincial Key Research and Development Program (No.ZDYF2024GXJS034)

References

Tinh T. Bui, Long D. Nguyen, Berk Canberk, Vishal Sharma, Octavia A. Dobre, Hyundong Shin, and Trung Q. Duong. Digital twin-empowered integrated satellite-terrestrial networks toward 6g internet of things. *IEEE Communications Magazine*, 62(12):74–81, 2024.

Hongzhi Guo, Xiaoyi Zhou, Jiadai Wang, Jiajia Liu, and Abderrahim Benslimane. Intelligent task offloading and resource allocation in digital twin based aerial computing networks. *IEEE Journal on Selected Areas in Communications*, 41(10):3095–3110, 2023.

Bishmita Hazarika, Keshav Singh, Chih-Peng Li, Anke Schmeink, and Kim Fung Tsang. Radit: Resource allocation in digital twin-driven uav-aided internet of vehicle networks. *IEEE Journal on Selected Areas in Communications*, 41(11):3369–3385, 2023.

Yilong Hui, Yi Qiu, Nan Cheng, Zhisheng Yin, Rui Chen, Kai Liang, and Tom H. Luan. Digital-twin-enabled on-demand content delivery in hetvnets. *IEEE Internet of Things Journal*, 10(16): 14028–14041, 2023.

Zhe Ji, Sheng Wu, and Chunxiao Jiang. Cooperative multi-agent deep reinforcement learning for computation offloading in digital twin satellite edge networks. *IEEE Journal on Selected Areas in Communications*, 41(11):3414–3429, 2023.

Yuna Jiang, Xiaohu Ge, Yang Yang, Chengxiang Wang, and Jie Li. 6g ori-ented blockchain based internet of things data sharing and storage mechanism. *Journal on Communications*, 41(10): 48–58, 10 2020.

Tong Liu, Lun Tang, Weili Wang, Qianbin Chen, and Xiaoping Zeng. Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network. *IEEE Internet of Things Journal*, 9(2):1427–1444, 2022.

Kai Peng, Hualong Huang, Muhammad Bilal, and Xiaolong Xu. Distributed incentives for intelligent offloading and resource allocation in digital twin driven smart industry. *IEEE Transactions on Industrial Informatics*, 19(3):3133–3143, 2023.

Wen Sun, Haibin Zhang, Rong Wang, and Yan Zhang. Reducing offloading latency for digital twin edge networks in 6g. *IEEE Transactions on Vehicular Technology*, 69(10):12240–12251, 2020.

Yihong Tao, Bo Lei, Haoyang Shi, Jingkai Chen, and Xing Zhang. Adaptive multi-layer deployment for a digital-twin-empowered satellite-terrestrial integrated network. *Frontiers of Information Technology & Electronic Engineering*, 26(2):246–259, 2025. ISSN 2095-9230.

Yuke Zhou, Ran Zhang, Jiang Liu, Tao Huang, Qinqin Tang, and F. Richard Yu. A hierarchical digital twin network for satellite communication networks. *IEEE Communications Magazine*, 61 (11):104–110, 2023.