Low-Energy Resource Optimization and Task Assignment for Satellite Edge Computing Networks

Xiaoteng Yang, *Graduate Student Member, IEEE*, Jie Feng, *Member, IEEE*, Lei Liu, *Member, IEEE*, Qingqi Pei, *Senior Member, IEEE*, Mianxiong Dong, *Senior Member, IEEE*, Keqin Li, *Fellow, IEEE*, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*—Satellite edge computing (SEC) has emerged as a promising paradigm to enhance in-orbit data processing capabilities and reduce transmission latency. However, satellite image processing tasks in SEC environments face critical challenges in efficient data handling, resource coordination, and transmission scheduling. The dynamic network topology and time-varying resource availability in satellite constellations further degrade the quality and stability of SEC services. To address these challenges, we propose a deep learning-based Collaborative Image Feature-extraction Task Optimization (CIFTO) framework. CIFTO dynamically distributes image processing workloads across multiple Low Earth Orbit (LEO) satellites, enabling continuous temporal updates for task allocation while significantly accelerating convergence and reducing computational overhead. By integrating temporal modeling and iterative optimization, CIFTO effectively mitigates the NP-hard nature of satellite task allocation. Furthermore, a lightweight satellite image processing model is designed to meet the strict constraints of on-orbit computation, achieving efficient image inference with minimal parameters. Extensive experimental evaluations demonstrate that the proposed framework ensures timely task completion, substantially lowers system-wide energy consumption, and enhances the adaptability and training efficiency of SEC services.**

***Index Terms*—Satellite edge computing, resource collaboration, task assignment, energy consumption.**

Received 5 September 2024; revised 11 March 2026; accepted 6 July 2026. Date of publication 15 July 2026; date of current version 11 August 2026. This work was supported in part by the National Natural Science Foundation of China under Grant U24A20241, Grant 62132013, and Grant 62272365; in part by the Fundamental Research Funds for the Central Universities under Grant QTZX26142 QTZX25085; in part by the Key Research and Development Programs of Shaanxi under Grant 2025SYS-SYSZD-080; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515012741; and in part by the Industrial Internet Innovation and Development Project under Grant ZC25T320059. Recommended for acceptance by J. Cao. *(Corresponding author: Jie Feng.)*

Xiaoteng Yang and Jie Feng are with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: xtengyang@stu.xidian.edu.cn; jiefengcl@163.com).

Lei Liu is with Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China, and also with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Qilu University of Technology Shandong Academy of Sciences, Jinan 250014, China (e-mail: leiliu@xidian.edu.cn).

Qingqi Pei is with the State Key Laboratory of Integrated Services Network, Shaanxi Key Laboratory of Blockchain and Secure Computing, Xidian University, Xi'an 710071, China (e-mail: qqpei@mail.xidian.edu.cn).

Mianxiong Dong is with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran 050-8585, Japan (e-mail: mx.dong@csse.muroran-it.ac.jp).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Schahram Dustdar is with the Distributed Systems Group, TU Wien, 1040 Vienna, Austria (e-mail: dustdar@dsg.tuwien.ac.at).

Digital Object Identifier 10.1109/TC.2026.3713664

I. Introduction

DRIVEN by the rapid development of space technology, modern satellites are increasingly equipped with advanced sensors and onboard computing devices, which significantly enhance their computational capabilities and broaden their application scope [1]. The rapid development of mobile edge computing and 6G technologies has created new opportunities to advance satellite edge computing (SEC) [2], [3], [4]. Notably, Low-Earth Orbit (LEO) satellites, due to their miniaturization and cost-effectiveness, can be deployed as large-scale, fully functional constellations to provide critical services in applications such as precision agriculture, remote sensing, and environmental monitoring. Moreover, LEO satellite constellations can offer scalable computing resources and on-demand services, enabling dynamic allocation and efficient utilization of satellite resources to satisfy diverse application demands while delivering user-specific, high-quality services [5], [6], [7].

However, with the widespread deployment of LEO satellite services, the volume of data generated by onboard sensors has increased dramatically, exceeding the processing capacity of satellite-ground links for timely data transmission. Even with the use of high-frequency Ka/Ku bands for data downlink, satellites can only transmit a portion of the collected data to ground receiving centers within the limited communication window [8], [9]. Meanwhile, the rapid expansion in the number of LEO satellites has outpaced the deployment of corresponding ground receiving stations. As a result, a large amount of sensor data cannot be downlinked in time due to the limited capacity of satellite-ground links and must be discarded, leading to severe resource waste. To mitigate this issue and reduce the overhead of satellite-ground link resources, SEC has emerged as a promising solution [10]. As shown in Fig. 1, the SEC network architecture offloads computing tasks to satellite edge nodes near the data source, thereby reducing transmission latency and bandwidth demands. This approach improves network resource utilization and enables users to achieve better quality of service and faster response times [6].

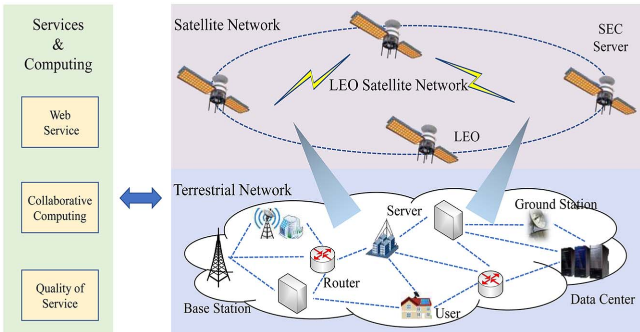


Fig. 1. SEC services network architecture.

In recent years, researchers have conducted in-depth and extensive studies to address the multifaceted challenges faced by satellite edge computing and processing tasks. Firstly, in [11], the concept of satellite edge intelligent computing was proposed, which captures satellite images and then performs intelligent processing directly on the satellite, transmitting the computational results to the ground to reduce communication overheads and improve the quality of service in the star-ground network. Secondly, [6] integrated satellite on-orbit computing with distributed cloud computing by considering each LEO satellite as a distributed node in the network, executing on-orbit computation via inter-satellite links, and transmitting the results to ground users, thereby providing an efficient task processing method. Considering the variability of satellite links, [12] proposed a processing algorithm based on satellite resources, which successfully improves the utilization of satellite computing resources. In order to solve the problem of complex operation and management of LEO satellites performing tasks on-orbit, [13] proposed a task decomposition method, which uses the IP model to generate the upper bound of the satellite scheduling problem and, for the first time, applies an optimization algorithm to the satellite broadcast scheduling problem, thereby effectively solving the management problem of LEO satellites performing tasks. Given the increasing computational demand of satellite native data for on-orbit processing, [14] proposed using cloud-native technology to improve the computational capability of on-orbit processing tasks. Driven by the large number of satellite remote sensing images, [15] proposed a working model for computing nanosatellite pipelines to alleviate resource constraints when processing tasks on a single satellite. [16] proposed an SDN-based network architecture that successfully reduces end-to-end latency and improves the efficiency of network resource utilization by converging satellite and terrestrial networks and adopting an on-demand and hierarchical approach to managing and scheduling network resources. Finally, [17] introduced a satellite IoT framework to fully utilize the LEO satellite task processing capability and proposed the E-CORA resource assignment algorithm, successfully minimizing the energy consumption of processing tasks. These studies offer substantial theoretical and technical support for LEO satellite processing tasks on-orbit and provide a solid foundation for the development of future satellite applications.

Meanwhile, image processing algorithms are crucial in SEC [18]. In this regard, [19] proposed using saliency information to assist ship detection in complex maritime environments, thereby reducing false alarms. In addition, tropical cyclone prediction can be performed by analyzing satellite images in real time. To this end, [20] employed deep convolutional neural networks to classify features of satellite cloud images in real time, thereby improving the accuracy of cyclone condition estimation. Furthermore, [6] introduced a new satellite framework that integrates on-orbit computing and artificial intelligence services, with the LEO satellite constellation covering the entire Earth's surface. However, the proposed system is not specifically designed to address on-orbit satellite image processing tasks. The limitations of satellite computational resources and the system complexity caused by multiple LEO satellites executing concurrent tasks are often ignored. As a result, energy-efficient task allocation strategies for resource collaboration in SEC environments remain underexplored.

To overcome these limitations, we combine artificial intelligence (AI) with satellite computing to enhance the on-orbit task processing capabilities of SEC. However, due to the complexity of SEC systems, achieving efficient joint management of resources such as on-orbit image processing and inter-satellite communication remains challenging. In this context, we investigate a low-energy image processing task allocation optimization framework within the satellite service computing paradigm to improve the energy efficiency of on-orbit computational services in SEC systems. Specifically, we propose a Collaborative Image Feature-extraction Task Optimization (CIFTO) framework that integrates computing resource allocation and lightweight image processing model optimization to maximize image detection accuracy, minimize system energy consumption, and ensure timely task completion. The main contributions of this paper are summarized below:

1) We propose a deep learning-based task allocation framework for satellite edge computing, named CIFTO. The framework jointly optimizes task allocation among multiple LEO satellites and image feature extraction through an adaptive temporal model.
2) We design a lightweight satellite image processing model tailored for in-orbit computation. Experimental results demonstrate that this model achieves superior classification accuracy with a minimal number of parameters, ensuring efficient and stable operation under the constrained resources of LEO satellites.
3) To ensure continuous optimization and low-energy operation, we introduce the Dynamic Iterative Preserving-sequential Optimization (DIPO) unit into CIFTO. This unit iteratively refines the assignment scheme, guiding the system toward energy-efficient task execution while preserving temporal consistency across iterations.
4) Simulation results validate that CIFTO significantly reduces the overall system energy consumption and guarantees timely task completion under dynamic satellite network conditions. Moreover, the proposed framework exhibits faster convergence and higher training stability compared with baseline approaches.

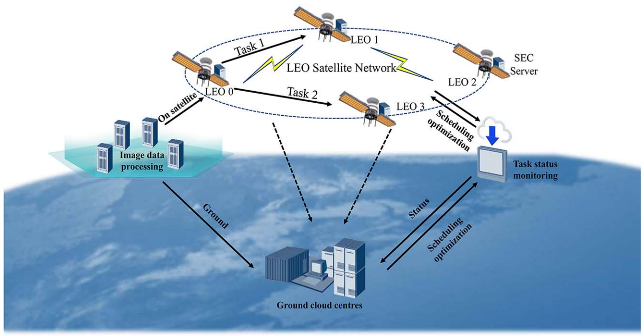


Fig. 2. The proposed SEC system.

The rest of this paper is structured as follows. Section II introduces the system model and formulates the optimization problem aimed at minimizing the overall energy consumption. Section III presents the proposed on-orbit task assignment framework, CIFTO, and the lightweight satellite image processing model. Section IV details the experimental setup, while Section V evaluates and discusses the performance of the proposed scheme. Finally, Section VI concludes the paper and outlines future research directions.

II. System Model and Problem Formulation

In this section, we first present an overall overview of the system and then elaborate on the computational models within the system. Subsequently, we explore the system's communication model, clarify the problem definition, and propose an optimization objective.

A. System Overview

Fig. 2 presents the proposed multi-LEO multi-task SEC system. The system consists of a local LEO (denoted as LEO 0), surrounded by L neighboring satellites, denoted as $\mathcal{L} = \{1, 2, \ldots, L\}$. The imaging camera on the local satellite captures and collects image data, and then transmits the corresponding data mission to the neighboring satellites. The SEC server on each neighboring satellite receives and processes the data using its onboard computing resources. The processing results are subsequently transmitted to the ground cloud center through the downlink. The ground cloud center is responsible for monitoring the satellite status and dynamically adjusting and assigning real-time tasks according to the mission's completion and system feedback.

The system operates as follows: a lightweight satellite image feature selector is initially trained on the ground and deployed onboard each neighboring satellite. During the time sequence $\mathcal{N} = \{1, 2, \ldots, N\}$, the local satellite captures images and segments them into several independent sets, each representing a separate task, with Q tasks denoted as $\mathcal{Q} = \{1, 2, \ldots, Q\}$. The ground cloud center collects state information from each satellite and inputs this critical data into a task assignment model (discussed in detail in the next section). This model predicts an assignment matrix $\delta_n \in \{0, 1\}^{Q \times L}$, where Q denotes the number of tasks and L denotes the number of neighboring satellites. For a task q, $\delta_{q,l}(n) = 1$ indicates that task q has been assigned to satellite l ($l \leq L$). Each task is assigned to only one satellite, represented by $\sum_{l=1}^{L} \delta_{q,l}(n) = 1$. The local satellite then distributes the stored images to neighboring satellites for feature extraction based on the assignment scheme. The features extracted by different tasks are denoted as $\boldsymbol{x_q}(n)$. After task execution, each neighboring satellite transmits the selected features to the ground cloud center via the downlink for further analysis and global decision-making.

B. Computing Model

In this section, we describe the computational model of the system. The computing system in each neighboring satellite is equipped with an image feature selector, which selects appropriate features and transmits them back to the ground. We use a lightweight architecture for feature extraction across different image tasks. The performance of the selected features is described by the metric A. In this paper, we use $A_{q,l}(n)$ to represent the overall accuracy of the feature $\boldsymbol{x_q}(n)$ that is processed on satellite l at different times for task q, i.e.,

$$A_{q,l}(n) = A(f(\boldsymbol{x_q}(n), q), y_q(n)), \tag{1}$$

where $y_q(n)$ denotes the ground-truth label or expected output of the image in task q, and f represents the corresponding feature selection processing model. Thus, the model can achieve a target detection accuracy of A_{target} for a specific number of parameters, i.e., $A_{q,l}(n) \geq A_{target}$.

After the satellite receives the task command, the various components of the system operate. Within the period n, we focus on different tasks q, which is denoted by $\phi_q(n) = \{D_q(n), C_q(n), T_q(n)\}$, where $D_q(n)$ is the size of the data, $C_q(n)$ is the number of CPU cycles required to process the task and $T_q(n)$ is the tolerable latency.

Within the system time frame, the time required to process task q on satellite l is

$$t_{q,l}^{com}(n) = \frac{C_q(n)}{S_l}, \tag{2}$$

where S_l represents the CPU clock speed of the satellite l. Here, $t_{q,l}^{com}(n)$ denotes the computation time required for executing task q on satellite l during time slot n. Therefore, the energy consumed in performing task q on the satellite is defined as

$$E_q^{com}(n) = \sum_{l \in \mathcal{L}} p_l \cdot t_{q,l}^{com}(n) \cdot \delta_{q,l}(n), \tag{3}$$

where $\delta_{q,l}(n) \in \{0, 1\}$ is the task assignment decision variable, indicating whether task q is executed on satellite l at time slot n. p_l denotes the power consumption of satellite l per unit time.

C. Communication Model

In this section, we detail the communication model of the SEC system, which is responsible for the efficient transmission of information between the local and neighboring satellites, as well as between the ground cloud center and neighboring satellites, to ensure that critical information can be transferred smoothly within the system.

According to the command, the primary communication process consists of the local satellite transmitting the image task to the neighboring satellites. Then, the neighboring satellites perform feature extraction and transmit the processed data back to the ground through the downlink. In this paper, it is assumed that all satellites operate in near-circular orbits within the same orbital plane. The distance between the local satellite 0 and neighboring satellite l is obtained as follows:

$$d_{0,l}(n) = (R+h)\sqrt{2(1-\cos(\Delta\theta_{0,l}(n)))}, \tag{4}$$

where R is the radius of the Earth, h is the orbital altitude, and $\Delta\theta_{0,l}(n)$ is the angular difference between the local satellite 0 and neighboring satellite l.

The Inter-Satellite Link (ISL) between satellites uses the Ka-band [9], and the channel gain $h_{0,l}(n)$ for data transmission from the local satellite 0 to neighboring satellite l is given by

$$h_{0,l}(n) = \frac{c}{4\pi f d_{0,l}(n)}, \tag{5}$$

where c is the speed of light and f is the carrier frequency. Therefore, the data rate of transmission between the local satellite 0 and neighboring satellite l is expressed as [21]

$$r_{0,l}(n) = b_{0,l}(n)\log_2\left(1+\frac{P_0G_0\left(h_{0,l}(n)\right)^2}{N_{0,l}}\right), \tag{6}$$

where $b_{0,l}(n)$ is the channel bandwidth, P_0 is the local satellite transmit power, G_0 is the local satellite antenna gain, and $N_{0,l}$ is the noise power between the channels of the local satellite 0 and neighboring satellite l. The antenna gain of the local satellite is approximated as $G_0 \approx (4D_0 f/c)^2$, where D_0 is the diameter of the transmitting antenna on the local satellite. Therefore, the transmission time $t_{0,l}^{tran}(n)$ for data transmission between satellites is [22]

$$t_{0,l}^{tran}(n) = \frac{D_q(n)}{\sum_{l\in\mathcal{L}} r_{0,l}(n)}\cdot\delta_{q,l}(n). \tag{7}$$

The energy consumed at the same time $E_{0,l}^{tran}(n)$ is

$$E_{0,l}^{tran}(n) = P_0\cdot t_{0,l}^{tran}(n). \tag{8}$$

Another part of the energy consumption is the neighboring satellite l after processing task q, which transmits the data to the ground cloud center. The distance from the nearby satellite l to the center of the ground cloud is

$$d_{l,g}(n) = \sqrt{(R+h)^2+R^2-2(R+h)r\cos(\phi_{l,g}(n))}, \tag{9}$$

where $\phi_{l,g}(n)$ is the geocentric Angle of satellite l, that is, the center Angle of the satellite concerning the center of the ground cloud. Its calculation formula is as follows

$$\phi_{l,g}(n) = \arccos\frac{R\cos\phi_{l,g}^{\min}(n)}{R+h} - \phi_{l,g}^{\min}(n), \tag{10}$$

where $\phi_{l,g}^{\min}(n)$ is the minimum elevation Angle of the ground cloud center to the satellite l.

After satellite l completes task q, the data rate $r_{l,g}(n)$ transmitted to the ground cloud center is

$$r_{l,g}(n) = b_{l,g}(n)\log_2\left(1+\frac{P_lG_l\left(h_{l,g}(n)\right)^2}{N_{l,g}}\right), \tag{11}$$

where $b_{l,g}(n)$ is the channel bandwidth transmitted by satellite l to the ground, P_l is the transtask power of satellite l, G_l is the antenna gain of satellite l, and $N_{l,g}$ is the noise power transmitted by satellite l to the ground, $h_{l,g}(n) = 1/\left(4\pi d_{l,g}(n)f/c\right)^2$ is the channel gain of satellite l transmitted data to the ground.

Therefore, after satellite l has processed task q, the time required to transmit the processed data back to the ground station through the downlink is

$$t_{l,g}^{tran}(n) = \frac{b_q(n)}{r_{l,g}(n)}\cdot\delta_{q,l}(n), \tag{12}$$

where $b_q(n)$ represents the size of the feature extracted by the processing task q, and $r_{l,g}(n)$ denotes the downlink transmission rate between satellite l and the ground station at time slot n. The amount of energy expended at the same time is

$$E_{l,g}^{tran}(n) = P_l\cdot t_{l,g}^{tran}(n). \tag{13}$$

D. Problem Formulation

In this section, we first define the energy consumption in a satellite computing system. The energy consumption within the system consists of three components: the energy $E_{0,l}^{tran}(n)$ consumed by the local satellite 0 for transmitting data to the neighboring satellite l, the power $E_q^{com}(n)$ consumed for feature extraction of task q on satellite l, and the energy $E_{l,g}^{tran}(n)$ consumed for transmitting the processed task q to the ground-based cloud center, defined as

$$E(n) = E_{0,l}^{tran}(n) + E_q^{com}(n) + E_{l,g}^{tran}(n). \tag{14}$$

Next, we formally define the problem as an optimization problem aiming to minimize the total energy consumption in a satellite computing system. Our goal is to find a resource assignment strategy to minimize the system's total energy consumption while satisfying various constraints. Specifically, our optimization objective can be expressed as

$$\begin{aligned}
&\mathcal{P}1: \min_{\delta}\frac{1}{N}\sum_{n}\sum_{q\in\mathcal{Q}}E(n)\\
&\text{s.t. (C1)}: \delta_{q,l}(n)\in\{0,1\}, \forall q\in\mathcal{Q}, \forall l\in\mathcal{L},\\
&\quad\text{(C2)}: \sum_{l=1}^{L}\delta_{q,l}(n) = 1,\\
&\quad\text{(C3)}: A_{q,l}(n)\geq A_{target}, \forall q\in\mathcal{Q}, \forall l\in\mathcal{L},\\
&\quad\text{(C4)}: \sum_{l=1}^{L}\delta_{q,l}(n)b_{0,l}(n)\leq B, \forall l\in\mathcal{L},\\
&\quad\text{(C5)}: t_{0,l}^{tran}(n)+t_{q,l}^{com}(n)+t_{l,g}^{tran}(n)\leq T_q(n), \forall q\in\mathcal{Q}.
\end{aligned} \tag{15}$$

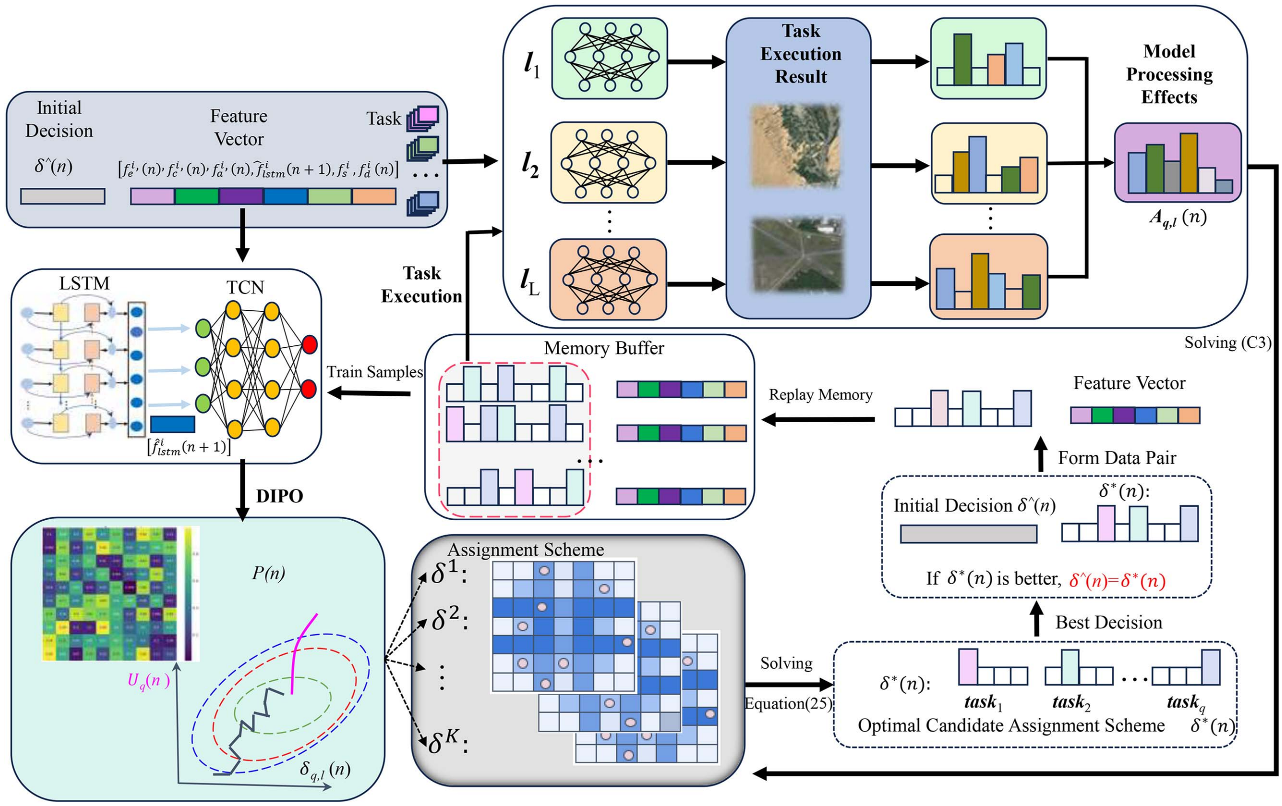


Fig. 3. The structure of the proposed CIFTO framework.

$\mathcal{P}1$ represents the minimum total system energy consumption value for a given task assignment. We need to find an assignment scheme $\delta(n)$ that minimizes the system's total energy consumption for a given task. The optimal assignment scheme is denoted as $\delta^*(n)$, with the constraints (C1) and (C2) ensuring that each task is assigned exactly once to one of the available LEO satellites in time frame n. The constraint (C3) ensures that each on-orbit computation task achieves the predefined target accuracy A_{target}, and the constraint (C4) limits the total communication bandwidth of all satellites performing the tasks so that it does not exceed the maximum available bandwidth B supported by the system. The final constraint (C5) guarantees that each task q is completed within its acceptable delay threshold $T_q(n)$.

$\mathcal{P}1$ is an NP-hard mixed-integer nonlinear programming (MINLP) problem. To solve this problem efficiently, we propose an approach consisting of three key steps. First, we introduce a deep learning-based task assignment model, whose computational complexity and structure will be analyzed in detail in the following section. Next, we perform image feature extraction using a lightweight processing model deployed onboard the LEO satellite's computational module to ensure that the expected performance can be achieved during on-orbit task execution. Finally, we optimize the energy consumption of the entire system under the above constraints to achieve minimal energy usage while ensuring timely task allocation and effective processing. Through this three-stage optimization strategy, the proposed framework achieves efficient task assignment and processing while significantly reducing overall system energy consumption.

III. The CIFTO Framework

A. CIFTO Framework Overview

As illustrated in Fig. 3, the proposed CIFTO framework consists of four tightly coupled modules: dynamic network condition prediction, temporal convolution-based task assignment decision generation, policy update, and execution. To effectively adapt to the time-varying and heterogeneous nature of terrestrial-satellite communication environments, CIFTO integrates a Long Short-Term Memory (LSTM) network for temporal prediction and a Temporal Convolutional Network (TCN) for sequential decision learning. This hybrid LSTM–TCN structure enables CIFTO to capture both long-term dependencies and local temporal correlations in network dynamics while maintaining computational efficiency for onboard inference.

B. Satellite Feature Embedding and Temporal Decision Generation

To enhance decision robustness under dynamic communication environments, the CIFTO framework employs an LSTM-based temporal predictor to model the sequential evolution of satellite-ground communication quality, represented by the signal-to-noise ratio (SNR). At each system time step $n \in N$, the LSTM receives a sequence of historical SNR observations

$(\psi(n-T+1), \ldots, \psi(n))$ and outputs the predicted next-step network state:

$$\hat{f}_{lstm}(n+1) = F_\phi(\psi(n-T+1), \ldots, \psi(n)), \tag{16}$$

where F_ϕ denotes the LSTM model parameterized by ϕ.

The predicted link quality $\hat{f}_{lstm}(n+1)$ is then combined with other satellite-related features to construct a comprehensive embedding vector for each satellite node. The input feature of satellite v_i at time step n is defined as

$$x_i(n) = [f_e^i(n), f_c^i(n), f_a^i(n), \hat{f}_{lstm}^i(n+1), f_s^i, f_d^i(n)], \tag{17}$$

where $f_e^i(n)$, $f_c^i(n)$, and $f_a^i(n)$ represent the energy, computational, and task accuracy features, respectively; $\hat{f}_{lstm}^i(n+1)$ corresponds to the predicted SNR obtained from the LSTM; f_s^i encodes the intrinsic attributes of the satellite; and $f_d^i(n) = [L_i(n), P_i(n), Q_i(n)]$ denotes dynamic operational attributes including computational load, residual power, and task queue length.

The concatenated feature representations are processed by the TCN to capture short-term dependencies and inter-feature correlations through dilated causal convolutions. For the l-th convolutional layer, the temporal feature transformation is expressed as:

$$h_i^{(l)}(t) = \sigma(W^{(l)} *_{d_l} h_i^{(l-1)}(t) + b^{(l)}), \tag{18}$$

where $*_{d_l}$ denotes a one-dimensional convolution with dilation factor d_l, $W^{(l)}$ and $b^{(l)}$ are trainable parameters, and $\sigma(\cdot)$ represents the ELU activation function. After multiple temporal convolutional layers, the TCN outputs the probabilistic task assignment distribution

$$P(n) = \mathrm{Softmax}(W_o[h_1^{(L)}, h_2^{(L)}, \ldots, h_K^{(L)}]), \tag{19}$$

where $P(n) = [p_{q,l}(n)] \in [0,1]^{Q \times L}$, and $p_{q,l}(n)$ denotes the probability that task q is assigned to satellite l at time step n.

After obtaining the probabilistic task assignment matrix $P(n)$ from the LSTM–TCN module, the next step is to determine the optimal executable task assignment matrix $\delta(n)$. To achieve this, the probability outputs are fed into the Dynamic Iterative Preserving-sequential Optimization (DIPO) unit, which performs a refined optimization process that jointly considers task accuracy and system energy consumption.

In the DIPO unit, the feature extraction performance of each task–satellite pair during execution, denoted as $A_{q,l}(n)$, is integrated into the optimization process. The objective is to minimize total energy consumption while maintaining high feature extraction accuracy. To balance these two factors, a Lagrange-based utility function is defined as

$$U_q(n) = \delta_{q,l}(n)\left(A_{q,l}(n) - \lambda E_{q,l}(n)\right), \tag{20}$$

where λ is a weighting factor that regulates the trade-off between task accuracy and energy efficiency. The overall optimization problem can thus be formulated as

$$\mathcal{P}2: \quad \max_{\delta} \sum_{q=1}^{Q} \sum_{l=1}^{L} U_q(n),$$
$$\text{s.t. (C1), (C2).} \tag{21}$$

Algorithm 1: Dynamic Iterative Preserving-sequential Optimization (DIPO) unit

Input: Probability matrix $P(n)$, number of candidate solutions K, initial values: $\lambda = 1.0$, $\alpha = 0.01$, adjustment factor $\eta = 0.05$, convergence threshold $\epsilon = 1e-3$.
Output: Task assignment plans $\{\delta^1, \delta^2, \ldots, \delta^k\}$.
Initialize Lagrange multipliers $\mu_q \leftarrow 0, \forall q$.
Initialize task assignment matrix $\delta_{q,l}(n) \leftarrow 0, \forall q, l$
while $k \le K$ and not converged **do**
  Initialize converged $\leftarrow$ True
  for $q = 1$ to Q **do**
    Compute $U_q(n)$ with equation (20).
    Generate candidate solutions based on equation (23).
    Store the previous value of μ_q: $\mu_q^{\text{old}} \leftarrow \mu_q$.
    Update μ_q with equation (24).
    if $|\mu_q - \mu_q^{\text{old}}| > \epsilon$ **then**
      converged $\leftarrow$ False
    end if
  end for
  Adjust the weighting factor λ:
  $\lambda \leftarrow \max(0, \min(1, \lambda \times (1-\eta)))$.
end while
return $\{\delta^1, \delta^2, \ldots, \delta^k\}$

To solve $\mathcal{P}2$, we define the Lagrangian function as follows:

$$\mathcal{L}(\delta, \mu) = \sum_{q=1}^{Q} \sum_{l=1}^{L} U + \sum_{q=1}^{Q} \mu_q \left(1 - \sum_{l=1}^{L} \delta_{q,l}(n)\right), \tag{22}$$

where μ is the Lagrange multiplier. Next, we initialize the $\delta(n)$ matrix to be an all-zero matrix, i.e., $\delta_{q,l}(n) = 0 \quad \forall q, l$. then, for each task q, compute its utility function $U_q(n)$. At different moments, generate K candidate solutions based on the following rule

$$\delta^k(n) = \begin{cases} 1 & l = \arg\max_k \left(U_q^{(k)}(n) + \mu_q P^{(k)}(n)\right). \\ 0 & otherwise. \end{cases} \tag{23}$$

The Lagrange multiplier μ_q is then updated, i.e.

$$\mu_q \leftarrow \mu_q + \alpha \left(1 - \sum_{l=1}^{L} \delta_{q,l}(n)\right), \tag{24}$$

where α is the learning rate. We iteratively optimize until μ_q converges. After completing the DIPO, we obtain a set of K allocation schemes, denoted as $\{\delta^1(n), \delta^2(n), \ldots, \delta^K(n)\}$. The pseudo-code of DIPO is presented in Algorithm 1. Finally, the best allocation scheme $\delta^*(n)$, which corresponds to the minimum energy consumption of the whole system, is selected from the resulting k candidates, denoted as

$$\delta^*(n) = \mathop{\arg\min}_{\delta^k \in \{\delta^1, \delta^2, \ldots, \delta^K\}} E(n). \tag{25}$$

Importantly, as the optimization process progresses, CIFTO can dynamically generate K candidate assignment schemes based on the temporal representations obtained from the

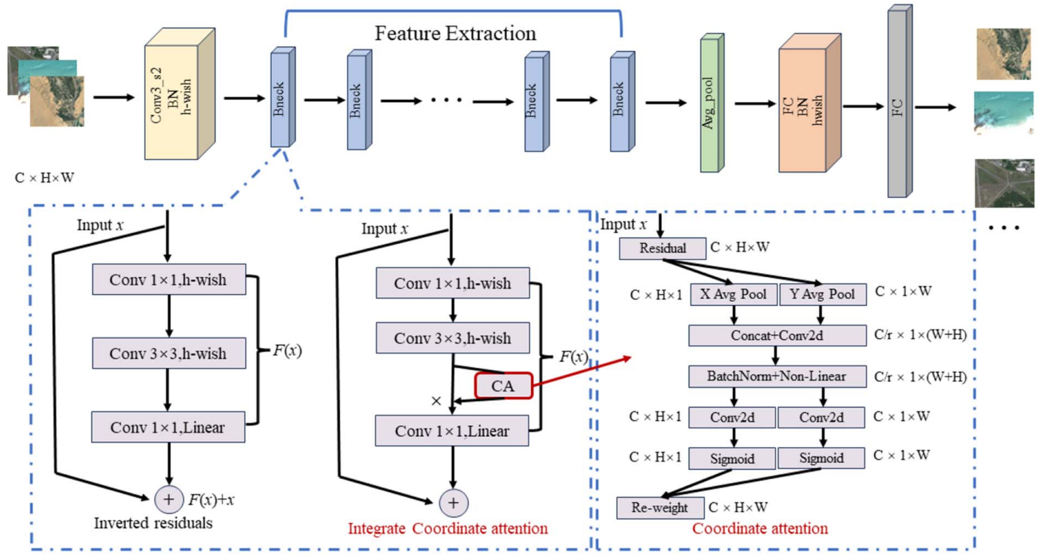


Fig. 4. The architecture of lightweight satellite image processing model.

LSTM–TCN hybrid network. This mechanism provides an efficient solution, as it enables the framework to continuously explore task assignment schemes with lower energy consumption. During the iterative optimization, CIFTO updates its internal state representations to further refine the temporal decision process, thereby enhancing its ability to identify energy-efficient allocation strategies under varying satellite conditions.

To ensure that the predicted assignment scheme remains consistent with the optimal solution of $\mathcal{P}1$, CIFTO continuously updates the temporal decision parameters through iterative learning. At the early stage of task execution, however, the model may face a cold-start problem, as the LSTM–TCN parameters are not yet well trained, potentially leading to suboptimal assignment schemes. To mitigate this issue, we introduce a high-quality initial assignment scheme generated by a greedy algorithm, denoted as $\delta^\wedge(n)$, which serves as the initialization reference.

As CIFTO executes and learns from temporal data, the quality of the predicted assignment scheme $\delta^*(n)$ gradually improves and may surpass the initial scheme $\delta^\wedge(n)$. To ensure continuous improvement, the two schemes are compared at each iteration. If the energy consumption of $\delta^*(n)$ is lower, the system updates $\delta^\wedge(n) = \delta^*(n)$, ensuring that CIFTO iteratively evolves toward high-quality solutions. At the n-th system time point, new temporal decision samples $(\delta^\wedge(n), x(n))$ are stored in a limited-capacity memory buffer. When the buffer is full, older samples are replaced by the most recent ones to maintain the adaptability and timeliness of the iterative optimization process.

C. Task Assignment Execution

After completing the optimal task assignment through the CIFTO framework, the ground cloud center issues execution commands to satellites via the satellite–ground communication link. As illustrated in Fig. 4, we enhance the MobileNetV3 architecture by integrating the Coordinate Attention (CA) mechanism, resulting in a lightweight model termed LCA-MobileNet [25], [26]. MobileNetV3 employs depthwise separable convolution, which reduces the computational cost to approximately one-ninth of that of standard convolution operations, and adopts an inverted residual structure to alleviate gradient vanishing issues during training. By replacing the original Squeeze-and-Excitation (SE) attention with the CA mechanism, the model achieves a more effective fusion of positional and channel feature information, which is particularly beneficial for capturing spatially distributed characteristics in satellite imagery [27].

Coordinate information embedding is achieved by decomposing global average pooling into two spatially scaled pooling kernels, $(H, 1)$ and $(1, W)$, encoding each channel with horizontal and vertical coordinates for feature aggregation. This enables the attention mechanism to capture remote dependencies in one spatial direction while preserving location information in the other, enhancing the model's focus on regions of interest. The outputs for the c-th channel in the horizontal and vertical directions are given by

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h, i), \quad Z_c^w(w) = \frac{1}{H} \sum_{0 \le j \le H} x_c(j, w), \tag{26}$$

where H and W are the height and width of the input feature map.

The coordinate attention generation process first compresses the feature map using 1×1 convolution, reduces channels by a factor of r, applies ReLU activation, and segments the feature map into horizontal and vertical tensors. The number of channels is restored using 1×1 convolution, followed by Sigmoid activation and weighted fusion. The final output is

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j), \tag{27}$$

where $g_c^h(i)$ and $g_c^w(j)$ are the attentional weights in the horizontal and vertical directions, respectively.

D. Assignment Policy Update

To obtain the optimal task assignment strategy with minimal energy consumption, the CIFTO framework continuously refines the assignment policy generated by the TCN-based probabilistic model. At each iteration, temporal decision samples are randomly drawn from a memory buffer to form a batch B, where $d \in B$ indicates that the data pair was generated in the d-th period. Each data pair in the batch consists of the reference assignment $\delta^\wedge(d)$ and its corresponding feature vector $x(d)$. The TCN processes $x(d)$ and outputs a probabilistic task assignment distribution $P(d) = [p_{q,l}(d)] \in [0,1]^{Q \times L}$, where $p_{q,l}(d)$ denotes the probability of assigning task q to satellite l. The single-sample objective is to minimize the expected energy consumption

$$\ell(d) = \sum_{q=1}^{Q} \sum_{l=1}^{L} p_{q,l}(d) \cdot E_{q,l}(d), \tag{28}$$

where $E_{q,l}(d)$ represents the estimated energy cost of assigning task q to satellite l under the system state at period d.

Based on the single-sample objective, the average batch loss function is defined as

$$\mathcal{L}(B) = \frac{1}{|B|} \sum_{d \in B} \ell(d), \tag{29}$$

where $|B|$ denotes the size of the training batch. The model parameters are updated using the Adam optimizer with learning rate η, which enables stable convergence and efficient adaptation under dynamic satellite communication environments [28]. By minimizing $\mathcal{L}(B)$, the CIFTO framework adaptively updates its task assignment policy, ensuring energy-efficient and reliable satellite collaboration. The complete optimization process is summarized in Algorithm 2.

Algorithm 2: Execution Procedure of the CIFTO Framework

Input: Feature vector $x_i(n)$ for each satellite v_i; system parameters.

Initialization: Initialize reference assignment $\delta^\wedge(n)$; initialize model parameters θ of TCN; initialize memory buffer $B = \emptyset$.

for each time step $n \in [1, N]$ **do**
  Predict future link quality $\hat{f}_{lstm}(n+1)$ for all satellites using LSTM.
  Construct feature embeddings $x_i(n)$ and input them into the TCN.
  Obtain probabilistic task assignment distribution $P(n) = [p_{q,l}(n)]$ from TCN output.
  Estimate the energy cost $E_{q,l}(n)$ for assigning task q to satellite l.
  Determine the optimal assignment $\delta^*(n)$ that minimizes $\ell(n) = \sum_{q,l} p_{q,l}(n) \cdot E_{q,l}(n)$.
  if $\delta^*(n)$ yields lower energy consumption or higher reliability than $\delta^\wedge(n)$ **then**
    Update reference assignment: $\delta^\wedge(n) \leftarrow \delta^*(n)$.
  end if
  Execute tasks according to $\delta^\wedge(n)$.
  Store the current temporal sample $(\delta^\wedge(n), x(n))$ into buffer B.
  if $n \bmod I = 0$ **then**
    Randomly sample a mini-batch $B_{\text{train}} \subset B$.
    Compute batch loss: $\mathcal{L}(B_{\text{train}}) = \frac{1}{|B_{\text{train}}|} \sum_{d \in B_{\text{train}}} \ell(d)$.
    Update TCN parameters θ using the Adam optimizer:

$$\theta \leftarrow \text{Adam}(\nabla_\theta \mathcal{L}(B_{\text{train}}), \eta)$$

    where η denotes the learning rate.
  end if
end for

Output: Final optimized task assignment $\delta^*(n)$.

IV. Experimental Settings

A. Simulation Setup

To systematically evaluate the proposed CIFTO framework, we first establish the overall experimental configuration, which includes both the LEO satellite system parameters and the task assignment framework execution settings. The LEO satellite configuration includes an orbital altitude $h = 780$ km, an Earth radius $R = 6371$ km, speed of light $c = 3 \times 10^8$ m/s, and carrier frequency $f = 30$ GHz. The phase difference between satellites $\theta_{0,l}(n)$ varies from 5° to 15°, and the minimum elevation angle of the ground center to the satellite $\phi_{l,g}^{\min}(n)$ is 16°. Interchannel noise powers $N_{0,l}$ and $N_{l,g}$ are set to -203 dBm/Hz, while satellite antenna gains G_0 and G_l are 43.3 dBi. Transtask power P_0 and P_l ranges from 50 W to 60 W, and computational power p_l ranges from 50 W to 100 W. The CPU clock speed S_l varies from 0.8 GHz to 4 GHz. The channel bandwidth $b_{0,l}(n)$ and $b_{l,g}(n)$ vary from 80 MHz to 100 MHz. Task sizes $D_q(n)$ vary between 64 MB and 100 MB [9], [21], [22].

The CIFTO framework was executed for $N = 6000$ system time steps. The task assignment model f_θ is trained under supervision at the ground-based cloud center. Initial assignment schemes are generated using a greedy selection method. To avoid overfitting, the batch size is 256, dropout rate is 0.1, and training parameters are updated every 10 CIFTO steps (i.e., training interval $I = 10$). The memory buffer size is set to 1.5 times the batch size. Adam's optimizer is used with a learning rate of 0.001. In CIFTO execution, the assignment scheme generated by DIPO uses $K = 30$ candidates. A timeout penalty mechanism is introduced, consuming 100 units of energy per timeout task to ensure that tasks exceeding deadlines are deprioritized.

B. Task Assignment Methods

To evaluate the effectiveness of CIFTO, we compare it against several representative baseline task assignment schemes. Energy consumption over the last 1000 system time steps is used as the primary metric:

1) *Random*: Tasks are randomly assigned to available satellites. Ten independent schemes are generated, and the one achieving the lowest total energy consumption is selected.
2) *Greedy*: Tasks with the longest estimated processing times are assigned to satellites with the highest available computational resources, updating satellite states dynamically.
3) *PPO* [34]: Task offloading is formulated as a Markov decision process. PPO learns an adaptive strategy minimizing energy and delay in a dynamic satellite network.
4) *JTO-CCRO* [35]: Task offloading and resource allocation are decoupled. The task offloading subproblem is solved via a game-theoretic approach, and resource allocation is optimized via Lagrange multipliers.
5) *ATO-SLA* [36]: Spatiotemporal satellite workloads are considered for adaptive task allocation to prevent overload while optimizing system performance.

C. Lightweight Model Evaluation

In addition, the proposed lightweight image processing model LCA-MobileNet is evaluated on single-label and multi-label datasets. The single-labeled dataset is merged from AID [29], NWPU RESISC45 [30], and PatternNet [31], selecting 35 common categories. For each category, 350 images are randomly selected for training, and 140 images are reserved for validation. The multi-labeled dataset merges MLRSNet [32] and AID-multi [33], containing 76 categories; 20% of data from each category is used for training, and the remaining for validation.

V. Experimental Results

A. Comparison of Task Processing Models

By comparing evaluation metrics on satellite image datasets, the effectiveness of the LCA-MobileNet model in image processing tasks is verified. When deploying the model to LEO satellites, accuracy (satisfying constraint C3), number of parameters, and computational requirements must be balanced to avoid reduced processing speed due to limited resources.

LCA-MobileNet demonstrates consistently high processing accuracy across both single-label and multi-label datasets. As shown in Figs. 5 and 6, the solid curves represent the average test accuracy obtained over ten independent experiments, indicating strong stability and robustness. The quantitative results in Table I show that LCA-MobileNet outperforms MobileNetV3 by 1.58$\%$ and 2.00$\%$ on the respective datasets, while maintaining a compact architecture with 6.24M parameters. This balance between accuracy and efficiency highlights the model's suitability for deployment in resource-constrained LEO satellite environments. Subsequently, the CIFTO framework will be evaluated under various operational scenarios to further verify its adaptability and performance.

B. CIFTO Performance Given Different Numbers of Tasks

In this experiment, the number of LEO satellites is fixed at three, and the performance of the proposed CIFTO model is evaluated under different task volumes. Fig. 7 illustrates the predicted energy cost corresponding to various task assignment scenarios, where each subplot represents a configuration with a distinct number of LEO satellites. The notation (x, y) indicates a system with x LEO satellites and y tasks. The expected assignment scheme may include tasks that fail to meet the acceptable delay thresholds, i.e., tasks violating constraints (C3) and (C4). The ideal assignment aims to minimize the overall energy cost while ensuring that all delay constraints are satisfied. To incorporate this trade-off, a timeout penalty mechanism is introduced, where each delayed task incurs an additional penalty of 100 energy units added to the system's total energy cost. This allows the evaluation of assignment schemes to jointly consider both energy efficiency and delay performance. As shown in Fig. 7, the proposed CIFTO model consistently achieves the lowest total energy cost, including timeout penalties, across all system configurations. These results demonstrate that CIFTO effectively generates task assignment strategies that minimize energy consumption while ensuring timely task completion under varying network conditions.

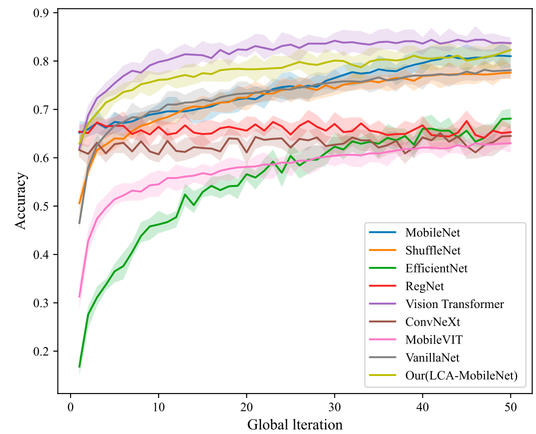


Fig. 5. Test accuracy of the proposed model with other models on single-label dataset [29], [30], [31].

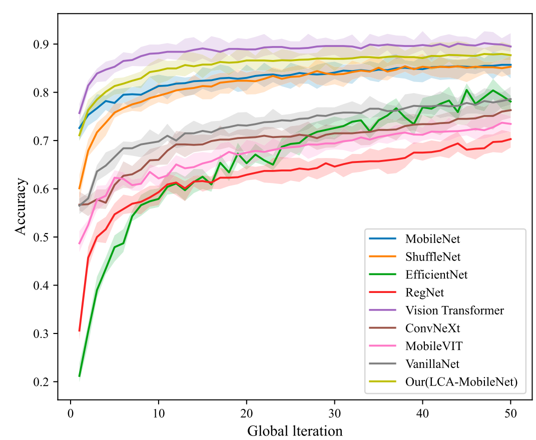


Fig. 6. Test accuracy of the proposed model with other models on multi-label dataset [32], [33].

TABLE I
COMPARISON ON SINGLE-LABEL, MULTI-LABEL DATASETS. LATENCY IS TESTED ON NVIDIA 4070TI GPU WITH BATCH SIZE OF 1

Model	Parameters (M)	FLOPs(B)	Latency (ms)	Acc(single-label)	Acc(multi-label)
MobileNet_v3_large [25]	5.48	0.22	4.19	81.50%	85.70%
ShuffleNetv2_x2 [37]	7.39	0.59	2.91	80.93%	85.41%
EfficientNet_b0 [38]	5.29	0.39	5.06	73.66%	80.46%
RegNet [39]	2.68	0.20	2.99	68.09%	70.30%
Vision Transform [40]	86.57	17.57	3.38	85.27%	90.25%
ConvNeXt [41]	28.59	4.46	4.46	66.98%	75.10%
MobileViT [42]	1.27	0.31	5.51	67.06%	71.81%
VanillaNet [43]	15.52	5.16	1.15	78.41%	78.06%
LCA-MobileNet	6.24	0.22	5.03	83.08%	87.70%

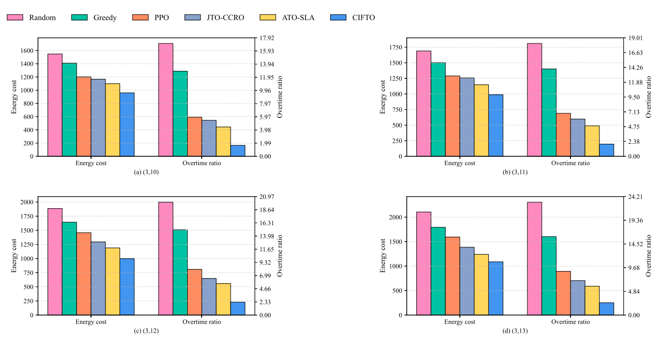


Fig. 7. Comparison of different methods for different numbers of tasks given in the system. The energy cost represents the energy consumed to perform the tasks and the overtime ratio represents the percentage of overtime in the task satisfaction scenario.

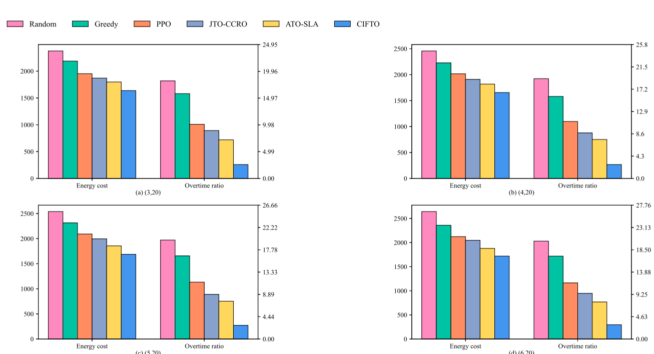


Fig. 8. Comparison of different methods of performing tasks given different numbers of satellites in the system. The energy cost represents the energy consumed to perform the tasks and the overtime ratio represents the percentage of overtime in the task satisfaction scenario.

C. CIFTO Performance Given Different Numbers of LEO Satellite

In this experiment, we evaluate the performance of the proposed CIFTO model in systems with different numbers of LEO satellite. The number of assignments in the system is fixed at 20. Fig. 8 shows the predicted energy cost and time delay for different assignment scenarios. The results show that CIFTO consistently outperforms the baseline approach and produces the best energy-efficient assignment scheme for different system configurations.

D. Ablation Experiment

This experiment aims to investigate the effect of the proposed DIPO unit on the performance of the CIFTO framework. We compare the energy cost of task execution under different configurations to verify the contribution of each module. Table II summarizes the average performance of the methods over the last 1000 system time steps. Specifically, CIFTO–E denotes the model variant without the initialization reference assignment scheme $\delta^\wedge(n)$, while CIFTO–B represents the baseline configuration where the probabilistic task assignment matrix $P(n) \in \mathbb{R}^{Q\times L}$ generated by the LSTM–TCN module is directly used for decision-making without further DIPO refinement.

At each system time step n, CIFTO–B selects the satellite with the maximum assignment probability for each task $q \in \mathcal{Q}$:

$$\delta_{q,l}(n) = \begin{cases} 1, & l = \arg\max_{l} p_{q,l}(n), \\ 0, & \text{otherwise}, \end{cases} \tag{30}$$

thereby satisfying constraints (C1) and (C2).

As shown in Table II, introducing the DIPO unit significantly reduces the system's overall energy cost compared to CIFTO–B. This improvement arises because DIPO iteratively refines the task assignment matrix by considering both feature extraction accuracy and energy efficiency, rather than relying solely on probabilistic predictions. Throughout the CIFTO execution, DIPO continuously explores improved assignment strategies and outperforms the baseline probabilistic approach. In summary, these results demonstrate that the integration of DIPO enables CIFTO to generate more energy-efficient and dynamically adaptive task allocation schemes.

E. Cold-Start Problem Analysis

This experiment investigates the effectiveness of introducing the initial reference assignment scheme $\delta^\wedge(n)$ in mitigating the cold-start problem during the early phase of CIFTO operation. The cold-start problem arises when the LSTM–TCN and DIPO units lack prior knowledge of high-quality assignment schemes, leading to suboptimal initialization and slower convergence to the optimal policy.

In the proposed framework, the initial reference assignment $\delta^\wedge(n)$ provides a high-quality starting point generated by a greedy algorithm, which guides the learning process of both

TABLE II

A Comparison of Different CIFTO Configurations for Task Execution Under Various Scenarios is Presented. CIFTOB Denotes the Model Without the DIPO Optimization Unit, and CIFTOE Refers to the Model Without the Initial Reference Assignment. Fig. 9 Shows the Energy Cost, Representing the Energy Consumed for Task Execution, While Fig. 10 Illustrates the Overtime Ratio, Indicating the Proportion of Time-Outs in Each Scenario. The Results Report the Average Values Across 1000 System Time Steps

Method	(3,15)		(3,20)		(4,15)		(4,20)	
	Energy Cost	Overtime Ratio	Energy Cost	Overtime Ratio	Energy Cost	Overtime Ratio	Energy Cost	Overtime Ratio
CIFTO–B	1518.342	3.74%	1983.277	8.12%	2068.452	5.88%	2381.911	10.32%
CIFTO–E	1476.829	1.62%	1910.564	3.81%	2009.732	5.92%	2294.648	6.84%
CIFTO	1382.217	1.58%	1678.503	2.96%	1835.967	3.97%	2071.482	6.35%

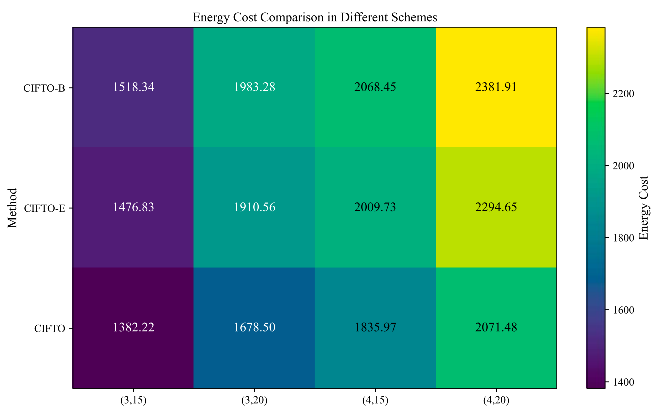


Fig. 9. Energy cost comparison of different CIFTO configurations under multiple task–satellite settings.

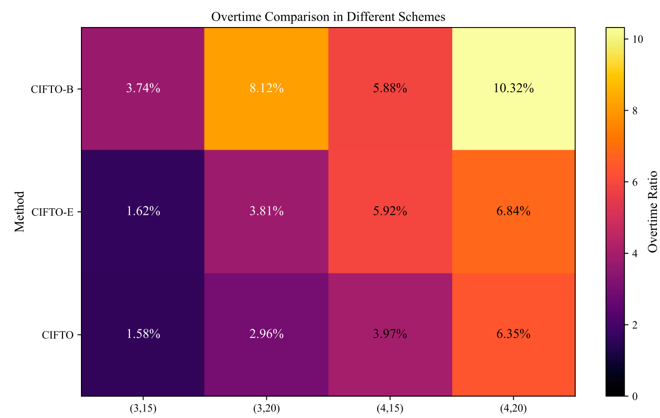


Fig. 10. Overtime ratio comparison of different CIFTO configurations across multiple scenarios.

the LSTM–TCN module and the DIPO optimization unit. In contrast, the CIFTO–E variant excludes this initialization and relies on random assignments at the beginning of training.

As illustrated in Fig. 9, CIFTO–E suffers from higher energy consumption and slower convergence compared to the complete CIFTO model. Without the initial reference assignment, the system struggles to efficiently explore the decision space, resulting in unstable task execution during the early learning stages. Conversely, the initialization strategy accelerates convergence by providing effective guidance for the collaborative optimization among the LSTM-based temporal predictor, TCN-based feature extractor, and DIPO policy optimizer.

Furthermore, Fig. 10 demonstrates that CIFTO achieves a consistently lower overtime ratio, reflecting its ability to maintain stable real-time task execution across varying task–satellite configurations. These results, together with the data summarized in Table II, confirm that incorporating the initial reference assignment effectively alleviates the cold-start issue, ensuring more stable and energy-efficient convergence in dynamic satellite communication environments.

F. Model Complexity Affects System Performance

This experiment investigates the impact of the model complexity within the CIFTO framework on overall system performance. As shown in Fig. 11, we evaluate three variants of CIFTO with different model configurations: (1) a lightweight model (*Simplified*), (2) the proposed balanced model (*Proposed*), and (3) a high-complexity model (*Complex*). The main differences lie in the number of layers and hidden units within the LSTM–TCN modules. The *Simplified* model employs a single LSTM layer with reduced temporal depth and a smaller TCN kernel size, while the *Complex* model incorporates multiple stacked LSTM layers and an expanded TCN structure with larger receptive fields. The *Proposed* model maintains a moderate configuration designed to balance representation ability and computational efficiency.

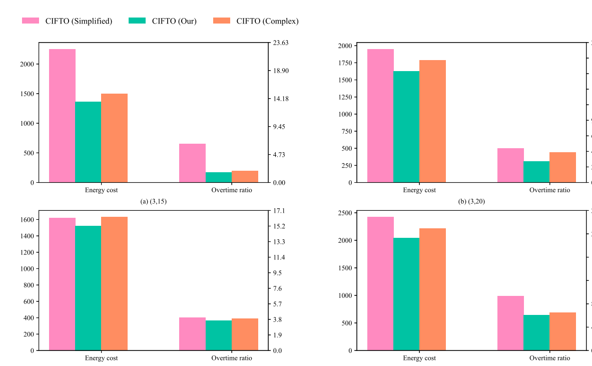


Fig. 11. Performance comparison of CIFTO under different model complexities.

Experimental results indicate that excessive model simplification leads to degraded decision quality and higher energy costs, primarily due to limited temporal feature extraction capacity. Although the complex variant achieves slightly better

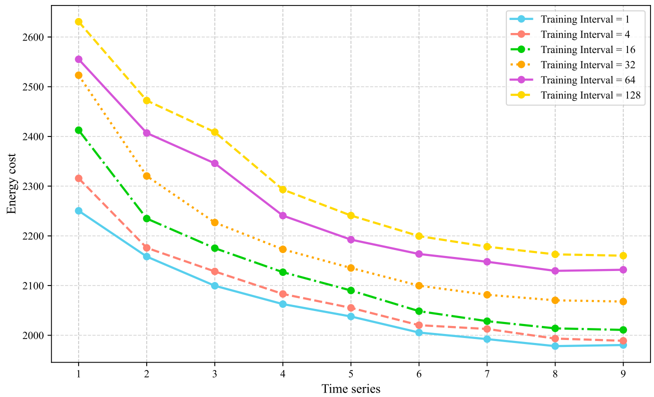


Fig. 12. Impact of parameter update interval on system energy cost. Each period represents the average energy cost of 10,000 CIFTO execution steps.

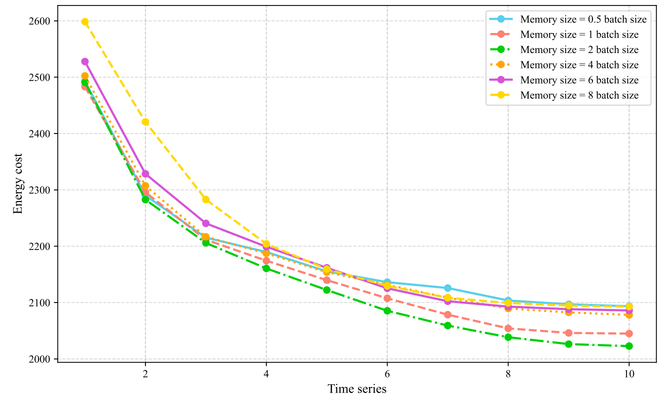


Fig. 13. Impact of memory buffer size on system energy cost. Each period represents the average energy cost of 10,000 CIFTO execution steps.

prediction accuracy, its higher computational burden results in increased system energy consumption and slower adaptation to dynamic environments. In contrast, the proposed balanced model achieves the optimal trade-off between computational overhead and task performance, ensuring efficient and stable operation of the CIFTO framework.

G. Model Analysis: Training Intervals

In this experiment, we investigate the effect of the temporal decision update interval I on the performance of CIFTO. The system is configured with 20 tasks and 5 LEO satellites. As shown in Fig. 12, the system energy cost under different update intervals demonstrates clear variation. CIFTO maintains relatively low and stable energy consumption when the update interval I is small, while higher energy costs emerge as I increases to larger values such as 64 or 128.

A smaller update interval enables more frequent parameter updates in the TCN-based temporal decision model, enhancing adaptability to dynamic communication environments but increasing computational overhead and slowing system execution. In contrast, an excessively large interval causes the model to lag behind environmental changes, resulting in suboptimal task assignment. Therefore, setting I to 4 or 16 provides a balanced trade-off between adaptive capability and computational efficiency. In our system, an update interval of 16 achieves near-optimal performance while maintaining a fast response speed for decision updates.

H. Model Analysis: Experience Memory Buffer Size

We further evaluate the influence of the experience memory buffer size on CIFTO performance, with the system set to 20 tasks and 5 LEO satellites. As illustrated in Fig. 13, CIFTO gradually produces task assignment schemes with lower energy costs across different buffer sizes due to the iterative refinement mechanism of the DIPO unit.

The system achieves the lowest energy consumption when the memory buffer size is approximately twice the batch size, ensuring that recent temporal samples play a dominant role in optimization. When the buffer size becomes excessively large, CIFTO exhibits slightly higher energy consumption and an increased probability of delayed task execution. This occurs because a large buffer retains outdated temporal samples, reducing the adaptability of the TCN-based decision model to the latest communication dynamics. A moderate buffer capacity therefore provides an optimal balance between adaptability and stability in the iterative optimization process of CIFTO.

VI. Conclusion and Future Work

In this paper, we proposed a deep learning-based task allocation framework named CIFTO, along with a lightweight satellite image processing model for SEC. The proposed CIFTO framework minimizes the energy consumption of on-orbit computation by jointly optimizing computational resource allocation, image feature extraction accuracy, and task processing delay. Furthermore, the embedded policy optimization unit ensures that task execution consistently progresses toward the direction of minimal energy consumption. Experimental results demonstrate that the proposed lightweight model significantly enhances image processing accuracy while maintaining a compact parameter scale. Simulation results further verify that CIFTO outperforms baseline schemes in both energy efficiency and execution latency. In future work, we will extend our study to on-orbit target detection tasks and focus on the joint optimization of system energy consumption and task completion delay.

References

[1] X. Zhang et al., "Energy-efficient computation peer offloading in SEC networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3077–3091, Apr. 2024.

[2] J. Wang, J. Hu, G. Min, W. Zhan, A. Y. Zomaya, and N. Georgalas, "Dependent task offloading for edge computing based on deep reinforcement learning," *IEEE Trans. Comput.*, vol. 71, no. 10, pp. 2449–2461, Oct. 2022.

[3] J. Feng, W. Zhang, Q. Pei, J. Wu, and X. Lin, "Heterogeneous computation and resource assignment for wireless powered federated edge learning systems," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3220–3233, May 2022.

[4] F. Al-Doghman, N. Moustafa, I. Khalil, N. Sohrabi, Z. Tari, and A. Y. Zomaya, "AI-enabled secure microservices in edge computing: Opportunities and challenges," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1485–1504, Mar./Apr. 2023.

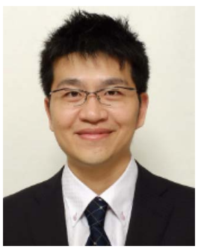
[5] J. Li et al., “Mobility-aware utility maximization in digital twin-enabled serverless edge computing,” *IEEE Trans. Comput.*, vol. 73, no. 7, pp. 1837–1851, Jul. 2024.
[6] Y. Zhang, C. Chen, L. Liu, D. Lan, H. Jiang, and S. Wan, “Aerial edge computing on orbit: A task offloading and allocation scheme,” *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 1, pp. 275–285, Jan./Feb. 2023.
[7] X. Yang et al., “IRS-assisted hyperspectral image processing in satellite edge computing services,” *IEEE Trans. Services Comput.*, vol. 18, no. 5, pp. 2946–2959, Sep.–Oct. 2025.
[8] Z. Xu, G. Xu, H. Wang, W. Liang, Q. Xia, and S. Wang, “Enabling streaming analytics in SEC via timely evaluation of big data queries,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 1, pp. 105–122, Jan. 2024.
[9] Y. Lee and J. P. Choi, “Connectivity analysis of mega-constellation satellite networks with optical intersatellite links,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 6, pp. 4213–4226, Dec. 2021.
[10] S. Chu, C. Gao, M. Xu, K. Ye, Z. Xiao, and C. Xu, “Efficient multi-task computation offloading game for mobile edge computing,” *IEEE Trans. Services Comput.*, vol. 17, no. 1, pp. 30–46, Jan.–Feb. 2024.
[11] Z. Zhang, W. Zhang, and F. H. Tseng, “Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques,” *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan./Feb. 2019.
[12] X. Chen, Z. Chen, X. Chang, T. Ji, Z. Wu, and C. Li, “Fast reroute algorithms for satellite network with segment routing,” *IEEE Access*, vol. 11, pp. 133509–133520, 2023.
[13] S. Kilic and O. Ozkan, “Modeling and optimization approaches for satellite broadcast scheduling problem,” *IEEE Trans. Eng. Manag.*, vol. 69, no. 5, pp. 1848–1864, Oct. 2022.
[14] C. Wang, Y. Zhang, Q. Li, A. Zhou, and S. Wang, “Satellite computing: A case study of cloud-native satellites,” in *Proc. IEEE Int. Conf. Edge Comput. Commun. (EDGE)*, Chicago, IL, USA, 2023, pp. 262–270.
[15] B. Denby and B. Lucia, “Orbital edge computing: Nanosatellite constellations as a new class of computer system,” in *Proc. 25th Int. Conf. Archit. Support Program. Lang. Oper. Syst.* 2020, pp. 939–954.
[16] F. Tang, “Dynamically adaptive cooperation transtask task among satellite-ground integrated networks,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, 2020, pp. 1559–1568.
[17] Z. Song, Y. Hao, Y. Liu, and X. Sun, “Energy-efficient multiaccess edge computing for terrestrial-satellite Internet of Things,” *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14202–14218, Sep. 2021.
[18] B. Zhu, S. Lin, Y. Zhu, and X. Wang, “Collaborative hyperspectral image processing using SEC,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 3, pp. 2241–2253, Mar. 2024.
[19] Z. Ren, Y. Tang, Z. He et al., “Ship detection in high-resolution optical remote sensing images aided by saliency information,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
[20] K. K. Jena, S. K. Bhoi, S. R. Nayak, R. Panigrahi, and A. K. Bhoi, “Deep convolutional network based machine intelligence model for satellite cloud image processing,” *Big Data Mining Analytics*, vol. 6, no. 1, pp. 32–43, Mar. 2023.
[21] R. Deng, B. Di, H. Zhang, L. Kuang, and L. Song, “Ultra-dense LEO satellite constellations: how many LEO satellites do we need?,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4843–4857, Aug. 2021.
[22] J. Wu et al., “A fast task offloading optimization framework for IRS-assisted multi-access edge computing system,” 2023, *arXiv:2307.08474*.
[23] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” 2015, *arXiv:1511.07289*.
[24] N. Srivastava, G. Hinton, A. Krizhevsky et al., “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[25] A. Howard et al., “Searching for MobileNetV3,” in *Proc. IEEE/CVF Int. Conf. Comput. Vision*, 2019, pp. 1314–1324.
[26] Q. Hou, D. Zhou, and J. Feng, “Coordinate attention for efficient mobile network design,” in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2021, pp. 13713–13722.
[27] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 7132–7141.
[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.
[29] G. S. Xia et al., “AID: A benchmark data set for performance evaluation of aerial scene processing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.
[30] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene processing: Benchmark and state of the art,” *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.
[31] W. Zhou, S. Newsam, C. Li, and Z. Shao, “PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval,” *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 197–209, Nov. 2018.
[32] X. Qi et al., “MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding,” *ISPRS J. Photogramm. Remote Sens.*, vol. 169, pp. 337–350, Nov. 2020.
[33] Y. Hua, L. Mou, and X. X. Zhu, “Relation network for multilabel aerial image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4558–4572, Jul. 2020.
[34] J. Schulman et al., “Proximal policy optimization algorithms,” 2017, *arXiv:1707.06347*.
[35] M. Jia, L. Zhang, J. Wu, Q. Guo, and X. Gu, “Joint computing and communication resource allocation for edge computing towards Huge LEO networks,” *China Commun.*, vol. 19, no. 8, pp. 73–84, Aug. 2022.
[36] J. Zhou, Y. Zhao, L. Zhao, H. Cai, and F. Xiao, “Adaptive task offloading with spatiotemporal load awareness in satellite edge computing,” *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 6, pp. 5311–5322, Nov./Dec. 2024.
[37] N. Ma et al., “ShuffleNet V2: Practical guidelines for efficient CNN architecture design,” in *Proc. Eur. Conf. Comput. Vision (ECCV)*, 2018, pp. 116–131.
[38] M. L. Q. Tan, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[39] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, and Z. Xu, “RegNet: Self-regulated network for image classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9562–9567, Nov. 2023.
[40] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, *arXiv:2010.11929*.
[41] Z. Liu et al., “A convnet for the 2020s,” in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2022, pp. 11976–11986.
[42] S. Mehta and M. Rastegari, “MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer,” 2021, *arXiv:2110.02178.*
[43] H. Chen et al., “VanillaNet: The power of minimalism in deep learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, p. 36.

Xiaoteng Yang (Graduate Student Member, IEEE) is currently working toward the Ph.D. degree in communication and information systems with Xidian University, Xi’an, China. His research interests include edge computing, resource assignment, satellite networks, and stochastic network optimization.

Jie Feng (Member, IEEE) received the Ph.D. degree in communication and information systems from Xidian University, China, in 2020. She is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Xidian University, Xian, China. From 2018 to 2019, she was with Carleton University, Ottawa, ON, Canada, as a visiting Ph.D. student. Her research interests include mobile-edge computing, blockchain, deep reinforcement learning, the device to device communication, resource assignment, and convex optimization, and stochastic network optimization.

Lei Liu (Member, IEEE) received the B.Eng. degree from Zhengzhou University, Zhengzhou, China, in 2010, and the M.Sc. and Ph.D. degrees from Xidian University, Xi’an, China, in 2013 and 2019, respectively, all in communication engineering. He is currently a Lecturer with the Department of Electrical Engineering and Computer Science, Xidian University. From 2013 to 2015, he was with a technology company. From 2018 to 2019, he was a visiting Ph.D. student with the University of Oslo, Oslo, Norway. His research interests include vehicular ad hoc networks, intelligent transportation, mobile edge computing, and the Internet of Things.

Qingqi Pei (Senior Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in computer science and cryptography from Xidian University, Xi'an, China, in 1998, 2005, and 2008, respectively. He is currently a Professor and member with the State Key Laboratory of Integrated Services Networks, Xidian University, where he is also the Director of Shaanxi Key Laboratory of Blockchain and Secure Computing. His research interests include digital contents protection and wireless networks and security. He is a Professional Member of the Association for Computing Machinery. He is a Senior Member of the Chinese Institute of Electronics and the China Computer Federation.



Mianxiong Dong (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and engineering from The University of Aizu, Aizuwakamatsu, Japan, in 2006, 2008, and 2013, respectively. He was a recipient of the 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, the Funai Research Award in 2018, the NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology, the Young Scientists' Award from MEXT in 2021, the SUEMATSU-Yasuharu Award from IEIEC in 2021, and the IEEE TCSC Middle Career Award in 2021. He is Clarivate Analytics 2019, a 2021 highly cited researcher (Web of Science), and a Foreign Fellow of Engineering Academy of Japan (EAJ).

Keqin Li (Fellow, IEEE) is a SUNY Distinguished Professor of computer science with the State University of New York and a National Distinguished Professor with Hunan University, China. His research interests include cloud computing, fog computing, mobile edge computing, and energy-efficient computing. He has authored over 870 journal articles, book chapters, and conference papers. Ranked among the top five most influential scientists in parallel and distributed computing based on Scopus citations, he is an Associate Editor of *ACM Computing Surveys* and *CCF Transactions on High Performance Computing*. He has also served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, and others. He is a fellow of AAIA, and a member of Academia Europaea.

Schahram Dustdar (Fellow, IEEE) received the Ph.D. degree in business informatics from the University of Linz, Linz, Austria, in 1992. He is currently a Full Professor of Computer Science (Informatics) with a focus on internet technologies heading the Distributed Systems Group, TU Wien, Wein, Austria. He has been the Chairman of the Informatics Section of the Academia Europaea, since 2016. He has been a member of the IEEE Conference Activities Committee (CAC), since 2016, the Section Committee of Informatics of the Academia Europaea, since 2015, and the Academia Europaea: The Academy of Europe, Informatics Section, since 2013. He was a recipient of the ACM Distinguished Scientist Award in 2009 and the IBM Faculty Award in 2012. He is an Associate Editor for IEEE Transactions on Services Computing, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*. He is on the editorial board of IEEE.