sensors

MDPI

Article

Robust Offloading for Edge Computing-Assisted Sensing and Communication Systems: A Deep Reinforcement Learning Approach

Li Shen [1], Bin Li [1,*] and Xiaojie Zhu [2]

[1] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202183290493@nuist.edu.cn
[2] Division of Computer Science, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia; xiaojie.zhu@kaust.edu.sa
* Correspondence: bin.li@nuist.edu.cn

Abstract: In this paper, we consider an integrated sensing, communication, and computation (ISCC) system to alleviate the spectrum congestion and computation burden problem. Specifically, while serving communication users, a base station (BS) actively engages in sensing targets and collaborates seamlessly with the edge server to concurrently process the acquired sensing data for efficient target recognition. A significant challenge in edge computing systems arises from the inherent uncertainty in computations, mainly stemming from the unpredictable complexity of tasks. With this consideration, we address the computation uncertainty by formulating a robust communication and computing resource allocation problem in ISCC systems. The primary goal of the system is to minimize total energy consumption while adhering to perception and delay constraints. This is achieved through the optimization of transmit beamforming, offloading ratio, and computing resource allocation, effectively managing the trade-offs between local execution and edge computing. To overcome this challenge, we employ a Markov decision process (MDP) in conjunction with the proximal policy optimization (PPO) algorithm, establishing an adaptive learning strategy. The proposed algorithm stands out for its rapid training speed, ensuring compliance with latency requirements for perception and computation in applications. Simulation results highlight its robustness and effectiveness within ISCC systems compared to baseline approaches.

Keywords: integrated communication and sensing; mobile edge computing; deep reinforcement learning; robust design; computation uncertainty

check for updates

Citation: Shen, L.; Li, B.; Zhu, X. Robust Offloading for Edge Computing-Assisted Sensing and Communication Systems: A Deep Reinforcement Learning Approach. *Sensors* **2024**, *24*, 2489. https://doi.org/10.3390/s24082489

Academic Editor: Charith Perera

Received: 12 March 2024
Revised: 30 March 2024
Accepted: 11 April 2024
Published: 12 April 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recent years have witnessed a rise in intelligence applications and services. Integrated sensing and communication (ISAC) has been suggested as a pivotal concept in next-generation wireless communication systems [1]. Conventional methodologies separate sensing and communication, giving rise to challenges such as intricate design, bandwidth interference, and resource inefficiencies. Nevertheless, wireless sensing shares significant similarities with wireless communication technology in aspects such as hardware infrastructure and signal processing, making the mentioned integration possible. In view of the shared spectrum resources and hardware in ISAC systems, efficient resource utilization as well as mutual reciprocity and benefit between sensing and communication functions can be enabled [2].

However, with the implementation of these advanced functionalities, especially during the rapid growth of the internet of things (IoT), network edge nodes have begun to generate substantial amounts of data, thereby escalating the demand for effective data processing capabilities. Confronted with such voluminous data, devising a strategy for its rapid and efficient processing has emerged as an urgent and critical challenge [3]. In

addressing this challenge, mobile edge computing (MEC) emerges as an innovative computing paradigm [4]. In this framework, proximally located servers are designated as edge servers, endowing user devices (UEs) with edge computing capabilities. This strategy markedly enhances data processing capabilities [5].

The network nodes in next-generation wireless communication systems will execute a variety of functions in an integrated manner, including high-precision, multi-objective environmental sensing and low-latency computing. This motivates the seamless integration of the ISAC network architecture with the MEC architecture, referred to as integrated sensing, communication, and computation (ISCC) [6], which is expected to support communication, sensing, and computation functionalities using the same signals and wireless infrastructure. Within this integrated framework, not only are the network nodes capable of simultaneously performing sensing and communication functions but also the system infrastructure is equipped to carry out efficient edge data processing [2]. To this end, ISCC can remarkably simplify equipment complexity and lower both production and usage costs significantly, which is essential for the advancement of wireless communication technology [7].

However, due to the unpredictability of computation types and the complexity of tasks, there is an inherent uncertainty in the computation processes of edge networks [8,9]. To address these challenges, a robust design scheme has been developed for the MEC network, which specifically accounts for the uncertainties associated with task complexity. To date, there has been limited research attention dedicated to the co-design of sensing, communication, and computation, especially when addressing the practical challenge of computation uncertainty. The main focus of this paper is to bolster a system's robustness while simultaneously minimizing system energy consumption. The primary technical contributions of this paper can be outlined as follows:

- The computation uncertainty of a UE's task within the framework of ISCC is investigated in this paper. To tackle the uncertainty of computation, a robust optimization problem aimed at minimizing system energy consumption is formulated. This is achieved by simultaneously optimizing communication and computation resources, beamforming, and offloading ratio.
- To address the outlined optimization challenges, this paper introduces a method that incorporates the proximal policy optimization (PPO) algorithm into deep reinforcement learning (DRL). This approach is designed to meet multiple constraints, including radar estimation information rate, computational offloading delay, and resource allocation. Utilizing a DRL training framework, this method allows for the efficient exploration and resolution of this intricate optimization problem. The system not only addresses practical constraints but also elevates decision making through the integration of intelligent learning algorithms.
- Through a series of simulation experiments, we assess the performance of this method, confirming the effectiveness of the computational robustness design and the PPO method in enhancing system efficiency and reducing energy expenditure. The robustness design demonstrates the improved performance in scenarios with uncertain task complexities. The simulations further reveal that the system's weighted energy consumption could be significantly lowered when using the PPO algorithm with robustness.

The remainder of this paper is structured in the following manner. In Section 2, related work is detailed. In Section 3, the system model is proposed. Then, the training framework is advanced, and its complexity is analyzed in Section 4. Section 5 delivers detailed simulations to confirm the effectiveness and robustness of the algorithm. Finally, Section 6 draws conclusions.

2. Related Works

There have been many works focused on enhancing the performance of MEC systems. For instance, the authors of [10] suggested a resource allocation strategy using DRL, which adaptively allocates computation and network resources. This method aimed to decrease

the mean service duration and balance resource usage under dynamic MEC conditions. Reference [11] investigated the minimization of system overhead in an MEC environment by jointly optimizing sampling, sensing, and computation offloading processes, effectively updating the status information of IoT systems. The work in [12] focused on optimizing resource allocation in ambient intelligence to maximize the convergence speed of federated edge learning. Further, ref. [13] presented the robust offloading policy and the joint allocation of communication and computation resources in an MEC system.

In the research on ISCC networks, efforts mainly place emphasis on the resource scheduling and optimization of beamforming techniques, ensuring the stability of communication links and enhancing the efficiency of computing task processing. In [14], the optimization of wireless spectrum resource strategies was performed in ISCC for dealing with high data transmission demands and complex computing tasks. The authors of [15] introduced adaptive digital twin technology in ISCC networks to improve network performance, especially in dynamic environments, in terms of application efficiency and reliability. While interesting, the works in [16,17] emphasized the significant role of beamforming in enhancing spectrum efficiency and reducing communication delay. The authors of [18] addressed the interference between radar sensing and MEC by jointly optimizing the sensing of beam pattern and task offloading with the aid of intelligent reflective surfaces. Considering previous work, there has been little research on robustness issues within ISCC networks. In this context, we study the computation uncertainty present in ISCC networks.

3. System Model and Problem Formulation

This section describes the system model for a robust ISCC system first and then formulates the energy consumption minimization problem.

3.1. System Model

This paper focuses on a beamforming and resource optimization issue in networks integrating communication, sensing, and computation, which are augmented by MEC. As shown in Figure 1, the network comprises one base station (BS) outfitted with M antennas, which serves K UEs with a single antenna each. The BS is not only equipped with an MEC server to enable computational offloading but also integrated with a radar sensing system, designed for the real-time detection and precise localization of potential targets. Concurrently, it ensures the provision of stable and reliable communication services to users. In practical applications, user devices can be regarded as smart wearable devices or AR devices, etc.

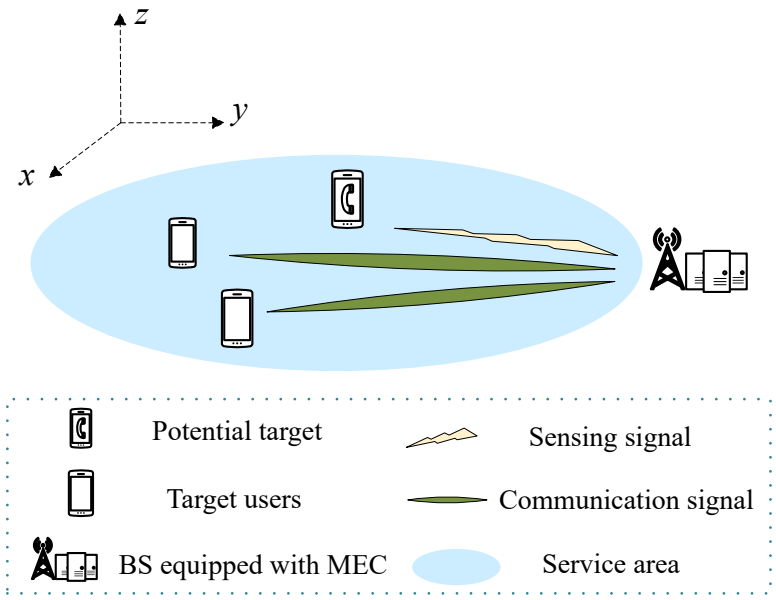


Figure 1. System model.

Taking into account the operational cycle and real-time requirements of the system, this paper posits that the task cycle is T, which is further subdivided into N time slots to facilitate meticulous resource scheduling, management, and computation. Consequently, the duration of each individual time slot is $\delta_n = T/N$, and it is assumed that the intervals between time slots are sufficiently short to satisfy the requirements for real-time processing. To simplify the problem, it is further assumed that within each time slot, the UE's computing tasks, data transmission, and other operations can be completed within that same time slot.

Within this framework, the primary entities and time units of the network are defined. In this context, the sets of UEs are denoted as $\forall k \in \mathcal{K} \triangleq \{1, \ldots, K\}$, and time slots are denoted as $\forall n \in \mathcal{N} \triangleq \{1, \ldots, N\}$. Without the loss of generality, this paper adopts the Cartesian coordinate system. Each UE is associated with a specific two-dimensional coordinate (x_k, y_k), and the BS is assigned a fixed two-dimensional coordinate $(x_{\mathrm{BS}}, y_{\mathrm{BS}})$ and has a height of 100 meters. It should be noted that all UEs are located on the ground; therefore, only their two-dimensional coordinates are taken into consideration.

3.2. Signal Model

In time slot n, for BS and UE k, the communication channel is characterized as a Gaussian channel.

(1) Received Signal.

During time slot n, the signal $\boldsymbol{x}[n]$, captured by the BS, is the superposition of the UE transmission signal $\boldsymbol{x}_{\mathrm{com}}[n]$, the radar sensing signal $\boldsymbol{x}_{\mathrm{sen}}[n]$, and the noise $\boldsymbol{n}[n]$, denoted as

$$\boldsymbol{x}[n] = \boldsymbol{x}_{\mathrm{com}}[n] + \boldsymbol{x}_{\mathrm{sen}}[n] + \boldsymbol{n}[n] \tag{1}$$

where $\boldsymbol{n}[n] \in \mathbb{C}^{U \times 1}$ is an independent and identically distributed Gaussian random noise vector with a mean value of zero and a variance of σ^2.

The transmission signal from the UE and the radar sensing signal are the essential components of the signal received at the BS. The ensuing sections will delve into the detailed composition and properties of these transmission signals.

The transmission signal $\boldsymbol{x}_{\mathrm{com}}[n]$ is characterized within the same time slot n, and the BS receives superimposed transmission signals from K distinct UEs, which can be expressed as

$$\boldsymbol{x}_{\mathrm{com}}[n] = \sum_{k \in \mathcal{K}} \boldsymbol{x}_{\mathrm{com}}^{k}[n] \tag{2}$$

where $\boldsymbol{x}_{\mathrm{com}}^{k}[n]$ is used as the transmission signal of the UE k, expressed as

$$\boldsymbol{x}_{\mathrm{com}}^{k}[n] = p_k[n] s_k[n] h_k[n] \tag{3}$$

here, $p_k[n]$ is the transmission power allocated to UE k during time slot n, and $s_k[n]$ is the data symbol.

For the sensing signal $\boldsymbol{x}_{\mathrm{sen}}[n]$, it is noted that the BS receives echoes from the target during time slot n. To effectively capture such echoes, the BS must first predict the radar's emission signal based on prior knowledge of the target. Nonetheless, the signal that the radar emits may encounter various interferences on its return. To mitigate these interferences and more accurately extract information about the target, this paper adopts the approach of subtracting the radar's transmission signal from the received signal to isolate the radar signal post-interference elimination, which can be described as $\tilde{\boldsymbol{s}}_{\mathrm{sen}}[n]$. The beamforming vector $\boldsymbol{w}_{\mathrm{sen}}[n]$ is then applied to $\tilde{\boldsymbol{s}}_{\mathrm{sen}}[n]$ to process and obtain the resultant sensing received signal as follows:

$$\boldsymbol{x}_{\mathrm{sen}}[n] = \boldsymbol{w}_{\mathrm{sen}}[n] \boldsymbol{H}_{\mathrm{sen}}[n] \tilde{\boldsymbol{s}}_{\mathrm{sen}}[n] \tag{4}$$

where $\boldsymbol{w}_{\mathrm{sen}}[n] \in \mathbb{C}^{U \times 1}$ and $\boldsymbol{H}_{\mathrm{sen}}[n] \in \mathbb{C}^{U \times U}$ describe the target response matrix of the radar.

Furthermore, upon receiving the signal $\boldsymbol{x}[n]$, the BS employs the beamforming vector $\boldsymbol{w}_k[n]$ to retrieve the signal. The retrieved signal is presented as

$$\begin{aligned}\hat{\boldsymbol{x}}_k[n] &= \boldsymbol{w}_k^{\mathrm{H}}[n]\boldsymbol{x}[n] \\ &= \boldsymbol{w}_k^{\mathrm{H}}[n](\boldsymbol{x}_{\mathrm{com}}[n] + \boldsymbol{x}_{\mathrm{sen}}[n] + \boldsymbol{n}[n]) \\ &= \boldsymbol{w}_k^{\mathrm{H}}[n](\sum_{k=1}^{K}\sqrt{p_k[n]}\boldsymbol{h}_k[n]s_k[n] \\ &+ \boldsymbol{H}_{\mathrm{sen}}[n]\boldsymbol{w}_{\mathrm{sen}}[n]\tilde{\boldsymbol{s}}_{\mathrm{sen}}[n] + \boldsymbol{n}[n])\end{aligned} \tag{5}$$

(2) Offloading Rate

Through beamforming, the directivity of the signal is optimized, enhancing the reception of the intended signal and concurrently attenuating the influence of unrelated signals and noise. This optimization is vital for the performance of communication systems, particularly for signal recovery and the data rates of UEs. Accordingly, in time slot n, the Shannon formula is employed to compute the offloading rate for the UEs, which can be articulated as follows:

$$R_k[n] = B \cdot \log_2(1 + P_{s/n}[n]) \tag{6}$$

where B denotes the channel bandwidth, $P_{s/n}[n]$ represents the signal–noise power ratio in time slot n, which is the ratio of the signal power $s_k[n]$ to the noise power $n_k[n]$ during this interval. The signal power $s_k[n]$ and the noise power $n_k[n]$ are described as follows, respectively:

$$s_k[n] = p_k[n]\left|\boldsymbol{w}_k^{\mathrm{H}}[n]\boldsymbol{h}_k[n]\right|^2 \tag{7}$$

$$\begin{aligned}n_k[n] = {} & B^2\psi^2\sigma_{\mathrm{sen}}^2\left|\boldsymbol{w}_k^{\mathrm{H}}[n]\boldsymbol{H}_{\mathrm{sen}}[n]w_{\mathrm{sen}}[n]\right|^2 \\ & + \sigma^2\boldsymbol{w}_k^{\mathrm{H}}[n]\boldsymbol{w}_k[n] + \sum_{i=1,i\neq k}^{K} p_i[n]\left|\boldsymbol{w}_k^{\mathrm{H}}[n]\boldsymbol{h}_i[n]\right|^2\end{aligned} \tag{8}$$

where, specifically, ψ represents the constant of the power amplifier, and σ_{sen}^2 is the variance of the radar's received signal noise.

3.3. Sensing Model

In the Signal Model section, a detailed description of the signal transmission process between the UEs and the BS is provided, as well as the specific expressions for each signal component. Nevertheless, it is also necessary to conduct the thorough modeling and analysis of the radar sensing aspect. The efficacy of radar sensing is intrinsically connected to the system's capacity for the precise and efficient execution of tasks related to target detection and localization. Consequently, the key metric of radar estimation information rate is introduced as a solution for quantifying the signal's sensing abilities [19].

The radar estimation information rate serves as the metric for target information acquired by radar. It is essentially the information shared between radar and the target through mutual interaction, quantifying the efficacy with which ISAC devices discern target information from the received echoes. This rate is employed to measure the volume of target information extractable from the sensing echoes.

In the context of radar estimation information rate, the signal–noise ratio of the radar echo signal is a key concept. It is denoted using a specific symbol $P_r[n]$ to denote the signal–noise ratio of the radar echo signal suppressed by ISAC devices in time slot n. The ratio of the radar duty cycle factor δ to the radar pulse duration μ, denoted by B_r, characterizes the

proportion of time where the radar is actively transmitting versus the duration of a single pulse. These are described as follows:

$$P_r[n] = \frac{B^2\psi^2\sigma_{\text{sen}}^2\left|\boldsymbol{c}[n]^{\text{H}}\boldsymbol{H}_{\text{sen}}[n]\boldsymbol{w}_{\text{sen}}[n]\right|^2}{\sigma^2\boldsymbol{c}[n]^{\text{H}}\boldsymbol{c}[n]} \tag{9}$$

$$B_r = \frac{\delta}{\mu} \tag{10}$$

where $\boldsymbol{c}[n] \in \mathbb{C}^{U\times 1}$ denotes the finite impulse response filter (FIR). Thus, during time slot n, the radar estimation information rate can be formulated as

$$R_{\text{r}}[n] = \frac{1}{2}B_r\log_2(1 + 2B\mu P_r[n]) \tag{11}$$

3.4. Computation Model

In real-world application scenarios, task complexities often vary across different types. This paper has developed a multi-task model comprising a set of diverse task types, denoted as $\forall z \in \mathcal{Z} \triangleq \{1, \ldots, Z\}$. The term $d_k[n]$ is used to quantify the data volume generated by UE k in time slot n, while c_z indicates the intricacy linked to task z, reflecting the computational power required for processing. Given that the exact complexity c_z may be unknown, computational uncertainty is introduced, reflecting the unpredictability of the real world. This can render the task scale measurable yet leave the completion time indeterminate. Through the analysis of historical data on multiple tasks, this research estimates the complexity of c_z, relating it to the error bound $\Delta\delta_z$. This error bound $\Delta\delta_z$ is constrained within a predefined threshold ε_z to bolster robustness, which is represented as

$$c_z = \hat{c}_z + \Delta\delta_z, |\Delta\delta_z| \leq \varepsilon_z \tag{12}$$

Furthermore, to facilitate the representation of task scheduling in time slot n, it is imperative to match the tasks and their respective types to each user independently, thereby ensuring that the task types assigned to each user within the same time slot are non-interfering. This is described as follows:

$$\begin{aligned} &\zeta_{z,k}[n] = \zeta_z[n]\zeta_k[n] \\ &\forall z \in \mathcal{Z}, \zeta_z[n] \in \{0, 1\} \\ &\forall k \in \mathcal{K}, \zeta_k[n] \in \{0, 1\} \end{aligned} \tag{13}$$

If $\zeta_z[n] = 1$, then the assigned task type is z; conversely, if $\zeta_z[n] = 0$, the task type is not z. In the same way, $\zeta_k[n] = 1$ denotes that the task comes from UE k; otherwise, it does not. It is established that a task is attributed to UE k and classified as type z if and only if $\zeta_z[n] = 1$ and $\zeta_k[n] = 1$, a condition that can be succinctly described as $\zeta_{z,k}[n] = 1$.

Due to the computational resource and energy constraints of UEs, it is infeasible to complete tasks locally within an expected timeframe. Consequently, this paper employs a partial offloading model. This means that each computational task is segmented into two components based on the offloading ratio $\rho_k[n]$. One proposed local computation is $d_k^{\text{loc}}[n]$, and the other one is transferred to BS $d_k^{\text{off}}[n]$, represented as follows:

$$d_k^{\text{loc}}[n] = (1 - \rho_k[n])d_k[n] \tag{14}$$

$$d_k^{\text{off}}[n] = \rho_k[n]d_k[n] \tag{15}$$

(1) Delay

In the time slot n, the delay due to local computation for UE k is expressed as

$$t_k^{\text{loc}}[n] = \frac{\sum_{z=1}^{Z} d_k^{\text{loc}}[n]\zeta_{z,k}[n]c_z}{f_k^{\text{loc}}[n]} \tag{16}$$

where $f_k^{\text{loc}}[n]$ denotes the processing rate of UE k in time slot n.

The offloading delay of a task offloaded from the user device to BS is described as

$$t_k^{\text{off}}[n] = \frac{d_k^{\text{off}}[n]}{R_k[n]} \tag{17}$$

When the task from UE k is uploaded to the BS, since the BS is furnished with one MEC server, the MEC server processes the tasks submitted by the UE k, and the computational delay incurred by the MEC is denoted as

$$t_k^{\text{mec}}[n] = \frac{\sum_{z=1}^{Z} d_k^{\text{off}}[n]\zeta_{z,k}[n]c_z}{f_k^{\text{mec}}[n]} \tag{18}$$

where $f_k^{\text{mec}}[n]$ denotes the processing rate by MEC for the task from UE k.

Given that the data volume processed by the MEC server is typically minimal, the delay associated with the return transmission is considered negligible in comparison to offloading and computational delays. Therefore, it is postulated that the return transmission occurs instantaneously. Consequently, the overall service delay for UE k is described as

$$t_k^{fin}[n] = \max\left\{t_k^{\text{off}}[n] + t_k^{\text{mec}}[n], t_k^{\text{loc}}[n]\right\} \tag{19}$$

(2) Energy Consumption

For the energy consumption attributable to computation, given that the energy resources of the BS can be considered unlimited, it is only necessary to account for energy consumption associated with the computation of UE k, which can be articulated as follows:

$$E_k^{\text{loc}}[n] = \sum_{z=1}^{Z} \varepsilon (f_k^{\text{loc}}[n])^2 d_k^{\text{loc}}[n]\zeta_{z,k}[n]c_z \tag{20}$$

where ε is defined as the effective capacitance coefficients that depend on the chip architecture of the local computing device.

Beyond the energy consumed for computation, the energy expenditure for offloading transmissions also should be taken into consideration. As the return transmission is assumed to be instantaneous and the data volume is approximated to zero, the energy cost associated with the return transmission is deemed negligible. Therefore, only the energy consumed during offloading is considered. The offloading energy from UE k is articulated as follows:

$$E_k^{\text{off}}[n] = \frac{d_k^{\text{off}}[n]}{R_k[n]} p_k[n] \tag{21}$$

Therefore, in time slot n, UE k's energy consumption can be obtained:

$$E_k[n] = E_k^{\text{off}}[n] + E_k^{\text{loc}}[n] \tag{22}$$

3.5. Problem Formulation

This study endeavors to minimize the aggregate system energy consumption throughout the entire cycle by jointly optimizing the offloading ratio $\boldsymbol{\rho} \triangleq \{\rho_k[n], \forall k \in \mathcal{K}, n \in \mathcal{N}\}$, the computational resource allocation of the MEC to each UE $\boldsymbol{f_e} \triangleq \{f_k^{\text{mec}}[n], \forall k \in \mathcal{K}, n \in$

$\mathcal{N}\}$, the local computing resource allocation for UEs $\boldsymbol{f}_l \triangleq \{f_k[n], \forall k \in \mathcal{K}, n \in \mathcal{N}\}$, and the beamforming $\boldsymbol{W} \triangleq \{w_k[n], \forall k \in \mathcal{K}, n \in \mathcal{N}\}$. The optimization problem is denoted as follows:

$$\begin{aligned}
\text{C0:} & \min_{\{\rho, \mathbf{f}_e, \mathbf{f}_l, \mathbf{W}\}} \sum_{n \in \mathcal{N}} \sum_{k \in \mathcal{K}} E_k[n] \\
\text{s.t. C1:} & \ 0 \le \rho_k[n] \le 1, \forall k \in \mathcal{K}, n \in \mathcal{N} \\
\text{C2:} & \ \zeta_{z,k}[n] \in \{0,1\}, \forall z \in \mathcal{Z}, k \in \mathcal{K}, n \in \mathcal{N} \\
\text{C3:} & \ \sum_{z \in Z} \zeta_{z,k}[n] = 1, \forall k \in \mathcal{K} \\
\text{C4:} & \ 0 \le f_k^{\text{mec}}[n] \le f_{\text{mec}}^{\max}, \forall k \in \mathcal{K}, n \in \mathcal{N} \\
& \ \sum_{k=1}^{K} f_k^{\text{mec}}[n] \le f_{\text{mec}}^{\max}, \forall k \in \mathcal{K}, n \in \mathcal{N} \\
\text{C5:} & \ 0 \le f_k^{\text{loc}}[n] \le f_k^{\max}, \forall k \in \mathcal{K}, n \in \mathcal{N} \\
\text{C6:} & \ 0 \le p_k[n] \le P_k^{\max}, \forall k \in \mathcal{K}, n \in \mathcal{N} \\
\text{C7:} & \ t_k^{fin}[n] \le t^{\max}, \forall k \in \mathcal{K}, n \in \mathcal{N} \\
\text{C8:} & \ |\triangle \ \delta_z| \le \varepsilon_z, \forall z \in \mathcal{Z} \\
\text{C9:} & \ R_{\text{r}}[n] \ge R_{\text{r}}^{\min}, \forall n \in \mathcal{N}
\end{aligned} \tag{23}$$

where $f_{\text{mec}}^{\max}$ and $f_k^{\max}$ signify the maximum processing rates of the MEC server and UE k for tasks, respectively, while $P_k^{\max}$ denotes the maximum transmitting power of UE k, and $t^{\max}$ constrains the slot time for a single task. The term $\Delta\delta_z$ is bounded within a radius of ε_z, and $R_{\text{r}}^{\min}$ represents the minimum radar estimation information rate. Constraint C1 pertains to the proportion of the task offloaded to the MEC server. C2 and C3 describe the type of task generated by UE k. C4 limits the maximum computational resources assigned to UE k by the MEC server. C5 restricts UE k's computational resources. C6 governs the offloading transmission power of UE k. C7 delineates the task duration. C8 manages the computational error. C9 prescribes the lower bound for the minimum performance of radar perception.

4. Proposed Algorithm

4.1. Modeling of Single-Agent MDP

A Markov decision process (MDP) is employed to model the challenge in this paper, which involves a nonlinear objective function and environmental uncertainties. To address the challenges posed by time-varying channels and multitasking, DRL is utilized to discover optimal strategies. Despite the increased complexity due to high-dimensional state spaces and the synchronization delay, the PPO variant of DRL algorithms is adopted. PPO ensures stable and effective policy learning for a single agent in complex environments and is instrumental in optimizing resource allocation and task execution within MEC networks to achieve minimal energy consumption. The training framework is shown in Figure 2.

MDP provides a potent mathematical framework to tackle optimization problems [20]. In the MDP paradigm, a decision maker strategically selects actions from a defined set of states. Each action precipitates a state transition coupled with an associated immediate reward. The decision maker's goal is to orchestrate a sequence of actions that amplifies the expected cumulative reward from the present state to a designated future juncture. MDP is especially applicable to contexts where environmental predictability is limited. Through the strategic application and resolution of MDP, the most advantageous policy sequence can be ascertained, offering a theoretically optimal solution to intricate decision-making problems.

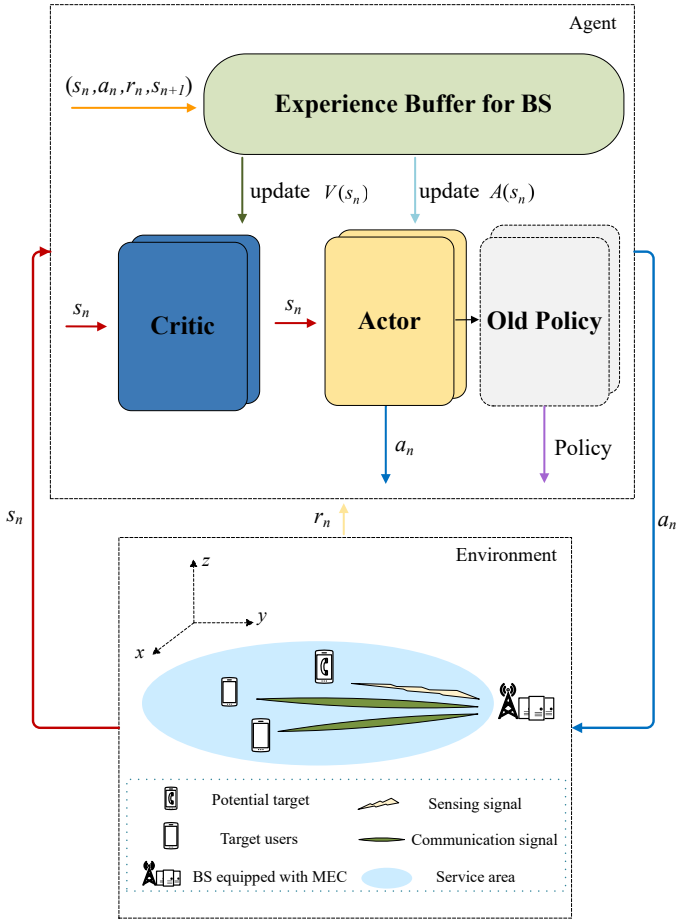


Figure 2. Training framework.

State space: To holistically account for the attributes of tasks and the resources of the BS, the state space is defined as

$$s_n = \{L_1[n], L_2[n], \ldots, L_k[n]\ ; \\ C_1[n], C_2[n], \ldots, C_k[n]\} \tag{24}$$

where the set of tasks is described by $L[n] = [L_1[n], L_2[n], \ldots, L_k[n]]$, and UEs' CPU resources are $C[n] = [C_1[n], C_2[n], \ldots, C_k[n]]$.

Action space: The agent, upon receiving the state space s_n, determines actions represented by a_n, which dictate the task offloading choices and the allocation of resources at the BS, thereby quantifying the resultant utility. Therefore, the action can be expressed as

$$a_n = \{\rho[n], f_e[n], w[n]\} \tag{25}$$

Simultaneously, with the goal of minimizing the energy expended in user computations, this paper employs dynamic voltage frequency scaling technology to configure and estimate the CPU frequency, as detailed in the subsequent formula:

$$f_k[n] = \min\{\frac{d_k[n]\zeta_{z,k}[n]c_z}{t_k^{fin}[n]}, f_k^{\max}\} \tag{26}$$

Reward space: To encapsulate the long-term optimization objectives of the problem and to address the fulfillment of constraints, this paper devises a reward function analogous to system energy consumption. The reward encompasses the energy expenditure of the user

as well as penalties incurred for breaches of delay and perception constraints. Consequently, the reward function formulated in this paper is illustrated as follows:

$$r_n = -\left[\sum_{k=1}^{K} E_k[n]\right] P_n^{\text{sen}} P_n^T \tag{27}$$

Among them, the perception constraint penalty P_n^{sen} is a linear penalty function, and the delay constraint penalty P_n^T is an exponential penalty function, described in Equations (28) and (29), respectively:

$$P_n^{\text{sen}} = 1 + \left(\frac{\bar{R}_{\text{sen}} - R_{\text{sen}}^{\min}}{R_{\text{sen}}^{\max} - R_{\text{sen}}^{\min}}\right) \tag{28}$$

$$\begin{aligned} P_n^T &= \frac{1}{K}\sum_{k\in K} P\left(t_k^{fin}[n], T_k[n]\right) \\ &= \frac{1}{K}\sum_{k\in K}\left(2 - \exp(-\left\lceil (t_k^{fin}[n] - T_k[n])/T_k[n]\right\rceil^+)\right) \end{aligned} \tag{29}$$

in which $\lceil\cdot\rceil^+$ means that the value is rounded up.

4.2. PPO-Based DRL Training Framework

To address the problem presented in this paper, PPO, a sophisticated policy gradient technique in DRL, is utilized within the MDP framework to optimize problem-solving methods. PPO is distinguished for its efficiency and capacity for robust optimization in policy spaces, garnering widespread popularity. This algorithm is characterized by a balanced approach to exploration and exploitation, achieved through limiting policy update steps. Such a mechanism is crucial in reducing variance during training and enhancing learning stability. This method is especially beneficial for optimization problems that demand continuous decision making and encompass a broad parameter space, such as dynamic system control and complex resource management tasks. Unlike algorithms that adopt an off-policy approach, the PPO algorithm is an on-policy method, meaning that the behavior policy and target policy are the same. This allows the algorithm to converge more quickly. Based on the on-policy method, agents trained with the PPO algorithm can continuously improve their policies while interacting with the environment, making it easier to adapt to changes in the environment while maintaining exploratory behavior. The effective learning of complex strategies, without compromising performance, is facilitated by PPO, thus providing more precise and robust solutions to the optimization problems.

Within the proposed framework, the actor–critic architecture plays a crucial role. The actor network is responsible for making decisions, essentially determining the action to be taken given the current state. Conversely, the critic network evaluates these actions by estimating the value function, providing feedback on the quality of the decisions made by the actor. In this architecture, the actor network is divided into new and old components, corresponding to parameters θ and θ_{old}, respectively. Furthermore, the critic network corresponds to network parameter ς. This collaborative adjustment ensures that the policy update is directed towards the enhancement of expected returns.

In the actor–critic framework, the state value function $V(s_n)$ becomes a pivotal component for strategy refinement. It encapsulates the expected returns from the current state s_n under the policy π, providing a crucial metric for the strategic adjustment process. The specific mathematical expression is as follows:

$$\begin{aligned} V(s_n) &= \mathbb{E}\{\mathbb{R}(a_n|s_n), \pi\} \\ &= \mathbb{E}\left\{\sum_{l=0}^{\infty} \gamma^l \mathbb{R}(a_{n+l}|s_{n+l})\right\} \end{aligned} \tag{30}$$

where $\mathbb{E}$ denotes the expected value operator, and γ represents the discount factor for future rewards, indicating the relative importance of future rewards compared to immediate ones. $\mathbb{R}$ symbolizes the reward obtained from a given state and action pair.

The action value function, represented as $Q(s_n, a_n)$, is a critical tool for evaluating the expected return of selecting an action a_n in a given state s_n and adhering to a policy π for subsequent actions. It comprehensively accounts for the immediate rewards and the aggregated impact of potential future rewards. The action value function may be indicated as

$$Q(s_n, a_n) = \mathbb{E}\left\{\sum_{l=0}^{\infty} \gamma^l \mathcal{R}(a_{n+l}|s_{n+l})\right\} \tag{31}$$

Building on this foundation, the advantage function is introduced to assess the efficacy of the selected actions within the given policy framework.

$$A(s_n) = Q(s_n, a_n) - V(s_n) \tag{32}$$

To guarantee the stability of policy updates, the framework employs the general advantage estimation (GAE) approach. The estimated advantage function $\hat{A}(s_n)$ is denoted as

$$\hat{A}(s_n) = \sum_{l=0}^{\infty} (\gamma\lambda)^l (r_{n+l} + \gamma V(s_{n+l+1}) - V(s_n)) \tag{33}$$

where λ is the GAE coefficient. Subsequently, the evaluation network and the action network are updated by optimizing the ensuing objective function. The critic network ς and actor network θ can be renewed by employing the following function:

$$G(\varsigma) = [V(s_{n+1}) - V(s_n)]^2 \tag{34}$$

$$\begin{aligned} G(\theta) = \mathbb{E}\Bigg\{ & \min\left[\frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{\text{old}}}(a_n|s_n)}\hat{A}(s_n),\right. \\ & \left.\text{clip}\left(\frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{\text{old}}}(a_n|s_n)}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}(s_n)\right]\Bigg\} \end{aligned} \tag{35}$$

where π_θ and $\pi_{\theta_{\text{old}}}$ represent the new and old policy functions, respectively, and the update ratio is denoted by $\frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{\text{old}}}(a_n|s_n)}$. The introduction of the clipping factor ε serves to constrain the policy's update ratio, ensuring controlled and stable optimization steps.

Algorithm 1 presents the pseudocode for the DRL training procedure utilizing the PPO algorithm.

4.3. Complexity Analysis

The complexity of the PPO algorithm, as proposed by the Computing Institute, is calculated in this section. Algorithm 1's computational complexity is gauged by the count of multiplication operations executed in a single iteration. Within the DRL framework, the observed state values are initially transmitted to a multi-layer perceptron (MLP) by the agent. A typical MLP consists of an input layer, an output layer, and multiple hidden layers. The state values enter the MLP through the input layer, are processed through the hidden layers, and they are ultimately outputted by the output layer. Given that the input layers as well as output layers have little impact on performance, they are typically disregarded. Consequently, the complexity of each layer can be characterized as follows:

$$\mathcal{O}(N_{i-1}N_i + N_iN_{i+1}) \tag{36}$$

where N_i represents the quantity of neurons in a specific hidden layer. Thus, the computation complexity of the layer I MLP is as follows:

$$\mathcal{O}\left(\sum_{i=2}^{I-1} (N_{i-1}N_i + N_iN_{i+1})\right) \tag{37}$$

Algorithm 1 Proposed PPO training framework

Initialize the maximum training episodes l_m, the maximum episode length l_e, the PPO epochs l_p, the BS's location (x_{BS}, y_{BS}).
Initialize Critic network ς, Actor network θ.
for $m \in \{1, \ldots, l_{\mathrm{m}}\}$ **do**
  Initialize users' location (x_k, y_k) and users' tasks
  for $n \in \{1, \ldots, l_e\}$ **do**
    Obtain state s_t from the environment
    Make decisions π_θ from the state s_t
    Choose action a_n based on π_θ
    Execute action a_n and update to next state s_{n+1}
    Calculate rewards r_n
    Store experience (s_n, a_n, r_n, s_{n+1})
  end for
  for $n \in \{1, \ldots, l_e\}$ **do**
    calculate $\hat{A}(s_n)$
  end for
  for $n \in \{1, \ldots, l_p\}$ **do**
    update ς and θ
  end for
end for

Both the actor and critic networks are constituted by an MLP. Based on the previous analysis, the total computational complexity of Algorithm 1 is, thereby, deduced as follows:

$$\mathcal{O}\left(l_m\left(l_e\left(\sum_{i=2}^{I-1}\left(N_{i-1}N_i + N_iN_{i+1}\right)\right)\right)\right) \tag{38}$$

Further, we reference the scheme proposed in [21], which addresses a scenario similar to ours, employing a WMMSE algorithm for optimizing resource allocation along with sensing and communication beamforming. The study in [22] introduces a beamforming framework enhanced by an LSTM network to augment communication efficiency, this method has also applied to our scenario for comparative analysis. The complexity comparisons of these three algorithms are systematically presented in Table 1.

For this comparison, we assume a setup with 16 users and 4 antennas. Considering that the learning algorithm incorporates convolutional layers, we set the maximum number of iterations to 300 and have 2 intermediate convolutional layers. The analysis clearly demonstrates that our algorithm is better. The complexity of the WMMMSE algorithm and the complexity of the LSTM algorithm are 10 and 2 times higher, respectively, compared to the proposed algorithm.

Table 1. Comparison of computational complexity of different algorithms.

Algorithm	Computational Complexity
WMMMSE algorithm [21]	$\mathcal{O}(l_m(2K^2W^3 + 2K^3 + K^{1/2}(4K+W)(3K+W)^2 + 6K^2))$
LSTM algorithm [22]	$\mathcal{O}(4l_m \cdot N_e \cdot l_e(\mathcal{G}_1\mathcal{G}_2 + \mathcal{G}_2^2 + \mathcal{G}_2)))$
Proposed algorithm	$\mathcal{O}\left(l_m\left(l_e\left(\sum_{i=2}^{I-1}\left(N_{i-1}N_i + N_iN_{i+1}\right)\right)\right)\right)$

5. Simulation Results

This section defines the simulation data to verify the impact of the MEC-assisted ISCC network based on the PPO algorithm on the system's overall energy use. The simulation

environment is constructed using the PyTorch framework. A thorough assessment of the performance of the proposed solution is conducted as follows.

5.1. Parameter Settings

The number of BSs is set to one, with the BS fixed at the coordinates $(0, 0)$ and positioned at an elevation of 100 m. Consider the initiation of activities within a terrestrial square region with a specified area of 1000 m $\times$ 1000 m. The users are randomly distributed within this area. Task data sizes are uniformly distributed in $[L_{\min}, L_{\max}]$, with $L_{\min}$ defaulting to 0.5 Mb and $L_{\max}$ defaulting to 1.5 Mb. The mean number of cycles per bit is $C_k[n] \in [500, 1500]$ cycles/bit. The duration of the task cycle, denoted as T, is set as 200 s. Unless explicitly stated otherwise, the default configuration comprises 16 users, with the communication channel bandwidth between the users and the BS being preset to $B = 10$ MHz. The noise-related power levels, denoted by parameters η^2 and η^2_{sen}, are set as -70 dBm uniformly. Some of the default parameters for the environmental settings are delineated in Table 2, while parameters pertinent to algorithmic training are enumerated in Table 3.

Table 2. Environment settings.

Parameters	Values
Time slot δ_n	1.0 s
Constant ψ	$\pi/\sqrt{3}$
Radar duty cycle factor δ	0.01
Radar pulse duration μ	2×10^{-5} s
Predefined threshold ε_z	15
MEC maximum frequency $f_{\text{mec}}^{\max}$	8 GHz
UE maximum frequency $f_k^{\max}$	1.5 GHz
UE maximum transmitting power $p_k^{\max}$	0.4 W
Minimum radar estimation information rate $R_{\text{rad}}^{\min}$	10^3 dB
Effective capacitance coefficients ε	10^{-27}

Table 3. Training settings.

Parameters	Values
Learning rate lr	5×10^{-3}
Maximum training set l_m	300 episodes
Length of each training set l_e	200 steps
Discount factor γ	0.95
GAE parameters λ	0.96

5.2. Simulation Evaluation

To validate the algorithm's performance, this study will conduct the following comparative assessments:

(1) Baseline PPO Design: This approach utilizes the PPO algorithm, devoid of enhancements for computation robustness.

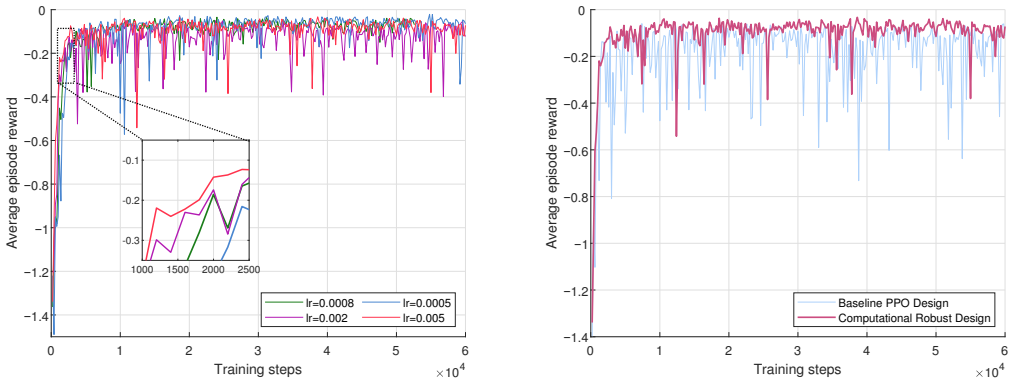
(2) Computational Robust Design: This variant integrates robust computational design, leveraging the DRL-enhanced PPO algorithm introduced herein, which seeks to determine nearly optimal actions for the ensuing time increment.

(3) Complete Offloading: This scenario entails the wholesale relegation of tasks to the BS for execution, thereby absolving users from computational duties.

(4) Random Offloading: In this model, users offload tasks to the BS in a stochastic manner, retaining the responsibility to compute any portion of the tasks not designated to the BS.

The convergence of the PPO algorithm is first substantiated through the analysis presented in Figure 3a. It is clear that with a growing number of training steps, the reward associated with the proposed solution correspondingly ascends. The enhancement of the stated reward by the agent is significant, which substantiates the great performance of the PPO algorithm in the context of computational offloading. Employing PyTorch for the computational experiments, an ensemble of data was amassed through 60,000 training steps, where each result is the sum of reward values within a round. Throughout the training phase, the strategies employed by the agent, particularly in communication, perception, and computation, undergo optimization. Concomitantly, the observed performance fluctuations become smaller, eventually allowing the algorithm to stabilize at the stable reward threshold.

In order to examine the impact of learning rate on algorithmic convergence, a comparative analysis of reward value convergence curves at different learning rates was undertaken. As depicted in Figure 3a, the learning rate of 5×10^{-3} achieves convergence at approximately 2000 training steps, while a learning rate of 5×10^{-4} requires about 8000 training steps for convergence. Additionally, when the learning rates are set at 2×10^{-3} and 8×10^{-4}, the convergence of the curves is observed to lie between the rates mentioned above. Despite the variability in convergence steps necessitated by different learning rates, once convergence was achieved, the resultant reward values were found to be insignificantly varied, residing within a stable range. These findings elucidate that the learning rate exerts a discernible effect on the PPO algorithm's convergence velocity, yet its influence on performance efficacy remains marginal.



(**a**) (**b**)

Figure 3. Performance differences in different trainings steps. (**a**) Convergence with different learning rate; (**b**) comparison with robust and non-robust.

Figure 3b provides a clear comparative analysis of the performance between the PPO method with the robust design computations and the non-robust method. The computationally robust PPO algorithm, as proposed in this paper, is observed to achieve higher and more stable overall rewards compared to the baseline PPO algorithm. This improved performance is largely attributable to the computationally robust strategies.

Figure 4 provides an in-depth comparative analysis of the changes in system-weighted energy consumption for four distinct methods across varying user scales. It is observed that the PPO algorithm, designed with a focus on computational robustness as proposed in this paper, outperforms others in terms of efficiency. The system's average weighted energy consumption under various user sizes is found to be lower when employing this algorithm, in contrast to the baseline PPO approach. Moreover, it is noted that the energy consumption resulting from strategies such as complete offloading or random offloading is significantly higher, which substantiates the effectiveness of incorporating partial offloading strategies based on offloading ratio ρ in the joint optimization process, thereby effectively reducing system energy consumption and enhancing performance. Additionally, an upward trend in average energy consumption between adjacent user

numbers is observed in Figure 4. This increase is attributed to the growing number of users accessing the network, which escalates signal interference among users, reduces transmission rates, and consequently raises transmission costs. Such developments lead to a decrease in the volume of tasks offloaded to BS and an increase in locally computed tasks, necessitating greater computational resources from users and ultimately resulting in a rise in system energy consumption.

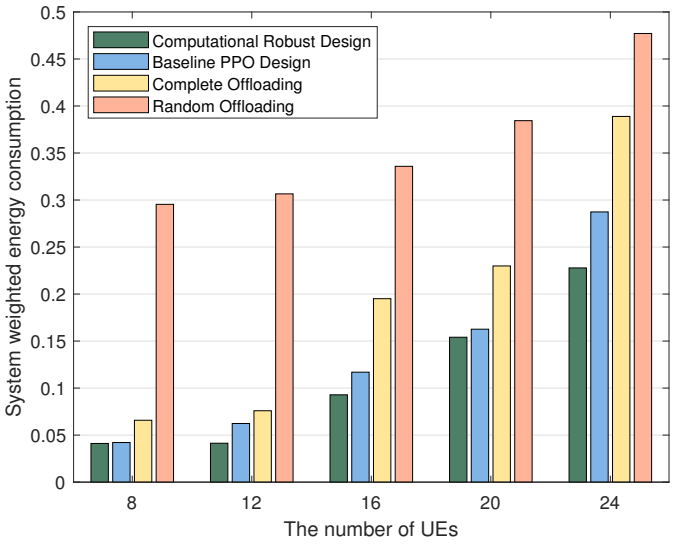
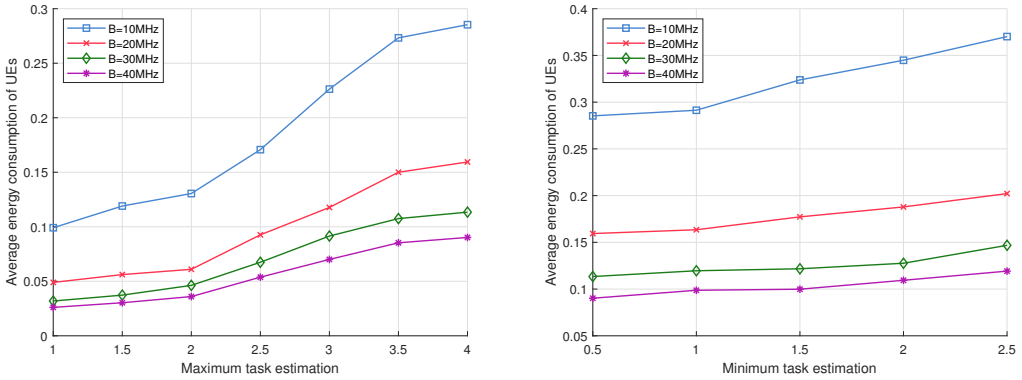


Figure 4. Performance comparison between different numbers of UEs.

Figure 5a and Figure 5b show the average weighted energy consumption of users under varying computational task sizes and different bandwidth settings, respectively. In Figure 6, with the minimum task size set at $L_{\min} = 0.5\,\text{Mb}$, it is observed that user energy consumption incrementally rises with an increase in the maximum task size $L_{\max}$, whereas energy consumption diminishes with the expansion of bandwidth. Similarly, in Figure 5b, with the maximum task size established at $L_{\max} = 4.0\,\text{Mb}$, it is noted that user energy consumption escalates as the minimum task size $L_{\min}$ increases. Furthermore, energy consumption intensifies as the available bandwidth narrows. This phenomenon can be attributed to the fact that the augmentation of communication resources enhances the users' transmission rates, whereas the escalation in computational tasks results in an upsurge in users' average computation energy consumption. This is because more sizable tasks necessitate a greater allocation of computational resources, thereby leading to increased energy expenditure. Moreover, the data depicted in the graphics indicate that with the increment in bandwidth, a divergence in energy consumption emerges under various bandwidth conditions. This primarily originates from the fact that, for the BS, an expanded bandwidth implies a reduction in transmission latency, and the shortened transmission time is afforded to BS for the processing of more computational tasks. The increase in bandwidth incentivizes users to offload a greater number of tasks to the BS, thus alleviating the local computational workload. Hence, the enhancement of bandwidth conserves computational CPU resources for users, thereby effectively reducing the computation energy consumption.

Figure 6 shows the influence of estimation error bounds and task complexity on performance. It is evident from the figure that when the estimation error bound increases, the energy consumption rises. This is attributed to the fact that a bigger error bound results in better uncertainty in the calculations.



(**a**) (**b**)

Figure 5. Energy consumption in different bandwidths and task estimation. (**a**) Performance comparison of different maximum task volumes and bandwidth; (**b**) performance comparison of different minimum task volumes and bandwidth.

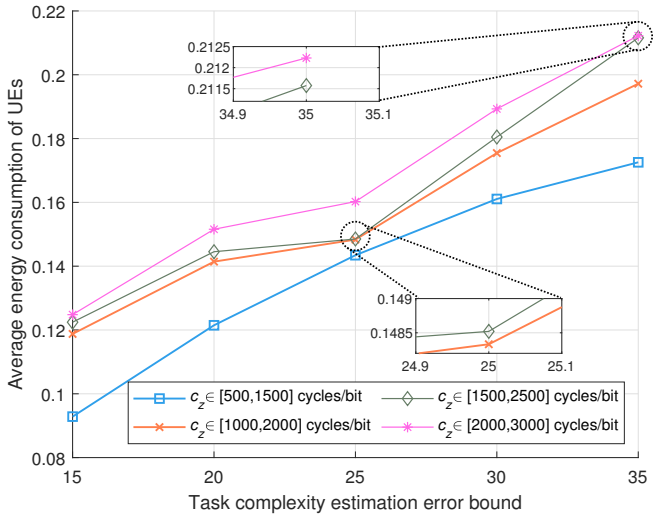


Figure 6. Performance comparison of different task volumes and task complexity estimation error bounds.

6. Conclusions

In this article, the computation uncertainties of tasks in ISCC systems was investigated, leading to the robust offloading and resource allocation scheme. By jointly optimizing transmit beamforming, offloading factors, communication and computation resource allocation, the system energy consumption minimization problem was formulated. To effectively address this optimization challenge, a PPO framework was developed to facilitate the efficient implementation of optimal learning policies. Extensive numerical results have highlighted the superiority of the proposed scheme in terms of energy consumption reduction as compared with baseline approaches. In future research, further investigations into resource allocation and optimization decisions in ISCC networks will be pursued by considering various task processing environments.

Author Contributions: Methodology, B.L.; software, X.Z.; writing—original draft preparation, L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Students' Platform Innovation and Entrepreneurship Training Program Support Project (Grant No. 202310300022Z) and in part by the National Nature Science Foundation of China (No. 62101277).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Xing, H.; Zhu, G.; Liu, D.; Wen, H.; Huang, K.; Wu, K. Task-Oriented Integrated Sensing, Computation and Communication for Wireless Edge AI. *IEEE Netw.* **2023**, *37*, 135–144.
2. Ding, C.; Wang, J.B.; Zhang, H.; Lin, M.; Li, G.Y. Joint MIMO Precoding and Computation Resource Allocation for Dual-Function Radar and Communication Systems with Mobile Edge Computing. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 2085–2102.
3. Spinelli, F.; Mancuso, V. Toward Enabled Industrial Verticals in 5G: A Survey on MEC-Based Approaches to Provisioning and Flexibility. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 596–630.
4. Dong, S.; Xia, Y.; Kamruzzaman, J. Quantum Particle Swarm Optimization for Task Offloading in Mobile Edge Computing. *IEEE Trans. Ind. Inform.* **2023**, *19*, 9113–9122.
5. Yan, J.; Bi, S.; Zhang, Y.J.A. Offloading and Resource Allocation with General Task Graph in Mobile Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 5404–5419.
6. Feng, Z.; Wei, Z.; Chen, X.; Yang, H.; Zhang, Q.; Zhang, P. Joint Communication, Sensing, and Computation Enabled 6G Intelligent Machine System. *IEEE Netw.* **2021**, *35*, 34–42.
7. Liu, F.; Cui, Y.; Masouros, C.; Xu, J.; Han, T.X.; Eldar, Y.C.; Buzzi, S. Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 1728–1767.
8. Tang, M.; Wong, V.W. Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems. *IEEE Trans. Mob. Comput.* **2022**, *21*, 1985–1997.
9. Li, B.; Yang, R.; Liu, L.; Wang, J.; Zhang, N.; Dong, M. Robust Computation Offloading and Trajectory Optimization for Multi-UAV-Assisted MEC: A Multiagent DRL Approach. *IEEE Internet Things J.* **2024**, *11*, 4775–4786.
10. Wang, J.; Zhao, L.; Liu, J.; Kato, N. Smart Resource Allocation for Mobile Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1529–1541.
11. Chen, Y.; Chang, Z.; Min, G.; Mao, S.; Hämäläinen, T. Joint Optimization of Sensing and Computation for Status Upyear in Mobile Edge Computing Systems. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 8230–8243.
12. Liu, P.; Zhu, G.; Wang, S.; Jiang, W.; Luo, W.; Poor, H.V.; Cui, S. Toward Ambient Intelligence: Federated Edge Learning with Task-Oriented Sensing, Computation, and Communication Integration. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 158–172.
13. Fan, R.; Liang, B.; Zuo, S.; Hu, H.; Jiang, H.; Zhang, N. Robust Task Offloading and Resource Allocation in Mobile Edge Computing with Uncertain Distribution of Computation Burden. *IEEE Trans. Commun.* **2023**, *71*, 4283–4299.
14. Zhao, L.; Wu, D.; Zhou, L.; Qian, Y. Radio Resource Allocation for Integrated Sensing, Communication, and Computation Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 8675–8687.
15. Li, B.; Liu, W.; Xie, W.; Zhang, N.; Zhang, Y. Adaptive Digital Twin for UAV-Assisted Integrated Sensing, Communication, and Computation Networks. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 1996–2009.
16. Li, X.; Gong, Y.; Huang, K.; Niu, Z. Over-the-Air Integrated Sensing, Communication, and Computation in IoT Networks. *IEEE Wirel. Commun.* **2023**, *30*, 32–38.
17. Qi, Q.; Chen, X.; Khalili, A.; Zhong, C.; Zhang, Z.; Ng, D.W.K. Integrating Sensing, Computing, and Communication in 6G Wireless Networks: Design and Optimization. *IEEE Trans. Commun.* **2022**, *70*, 6212–6227.
18. Huang, N.; Wang, T.; Wu, Y.; Wu, Q.; Quek, T.Q.S. Integrated Sensing and Communication Assisted Mobile Edge Computing: An Energy-Efficient Design via Intelligent Reflecting Surface. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 2085–2089.
19. Huang, N.; Dou, C.; Wu, Y.; Qian, L.; Lu, R. Energy-Efficient Integrated Sensing and Communication: A Multi-Access Edge Computing Design. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 2053–2057.
20. Yamansavascilar, B.; Baktir, A.C.; Sonmez, C.; Ozgovde, A.; Ersoy, C. DeepEdge: A Deep Reinforcement Learning Based Task Orchestrator for Edge Computing. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 538–552.
21. Wang, Z.; Mu, X.; Liu, Y.; Xu, X.; Zhang, P. NOMA-Aided Joint Communication, Sensing, and Multi-Tier Computing Systems. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 574–588.
22. Liu, C.; Yuan, W.; Li, S.; Liu, X.; Li, H.; Ng, D.W.K.; Li, Y. Learning-Based Predictive Beamforming for Integrated Sensing and Communication in Vehicular Networks. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 2317–2334.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.