

Zhang, Tiantian, Dongyang Xu, Amr Tolba, Keping Yu, Houbing Song, and Shui Yu. "Reinforcement Learning-Based Offloading for RIS-Aided Cloud-Edge Computing in IoT Networks: Modeling, Analysis and Optimization." IEEE Internet of Things Journal (08 March 2024).

<https://doi.org/10.1109/JIOT.2024.3367791>. || <https://doi.org/10.1109/JIOT.2024.3367791>

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

Reinforcement Learning-Based Offloading for RIS-Aided Cloud-Edge Computing in IoT Networks: Modeling, Analysis and Optimization

Tiantian Zhang, *Graduate Student Member, IEEE*, Dongyang Xu, *Member, IEEE*, Amr Tolba, *Senior Member, IEEE*, Keping Yu, *Senior Member, IEEE*, Houbing Song, *Fellow, IEEE* and Shui Yu, *Fellow, IEEE*

Abstract—The rapid advancement of wireless communication and artificial intelligence (AI) has led to a plethora of emerging applications that require exceptional connectivity, minimal latency, and substantial computing resources. The widespread adoption of cloud-edge intelligence is propelling the development of future networks capable of supporting intelligent computing. Mobile edge computing (MEC) technology facilitates the movement of computing resources and storage to the network's edge, enabling cost-effective offloading of computational tasks for related applications which needs for reduced latency and improved energy efficiency. However, the offloading efficiency is hindered by limitations of wireless transmission capacity. This paper aims to address this issue by integrating reconfigurable intelligent surfaces (RISs) into a cell-free network within an intelligent cloud-edge system. The core idea is to strategically deploy passive RISs around base stations (BSs) to reconstruct the transmission channel and improve the corresponding capacity. Subsequently, we formulate an optimal problem involving joint beamforming for RISs and BSs, which is characterized by non-convexity and complexity. To tackle this challenge, we employ an alternating optimization scheme to ensure the effectiveness of joint beamforming. In particular, deep reinforcement learning (DRL) is leveraged to reduce the computational complexity involved in optimizing task offloading. Additionally, Lyapunov optimization is utilized to model the latency queue and improve the learning efficiency of the offloading framework. We conduct comprehensive evaluations on the wireless system's capacity, average latency, and energy consumption, considering the integration of RIS with the DRL offloading framework. Experimental results demonstrate that our proposed scheme achieves superior efficiency and robustness.

Index Terms—Reinforcement learning, cloud-edge offloading, reconfigurable intelligent surface, resource allocation.

I. INTRODUCTION

THE rapid advancement of 5G/6G wireless communication, along with the proliferation of intelligent terminal devices, has led to the deployment and application of numerous Internet of Things (IoT) devices in various scenarios. Additionally, the limited computing capacity and security of terminals pose challenges for existing computing frameworks to keep up with the rapid development of emerging services like voice, video, virtual reality, industrial control, and autonomous driving [1]–[3]. Edge artificial intelligence (e-AI) is a remarkable emerging technology that offers a robust solution for various IoT-enabled applications, including ultra-reliable and low-latency communications (URLLC), as well as computation and network resource optimization [4]–[7]. Furthermore, e-AI has the ability to integrate machine learning algorithms, which can provide high computing effectiveness, efficiency, and intelligence, into edge nodes within the network, as emphasized in previous studies [8], [9]. The core objective of mobile edge computing (MEC) involves the deployment of a versatile computing server at the wireless periphery, furnishing computational resources to wireless access devices [51], [52].

However, a significant challenge persists in effectively allocating resources for the offloading of computational tasks. Moreover, given that actual IoT applications typically involve lightweight devices that are sensitive to power constraint, energy consumption during the offloading process of edge or cloud computing is a crucial factor to be investigated, as emphasized in [11], [12]. Given the requirements of delay-sensitive tasks, the average offloading latency is the key indicator for evaluating offloading schemes, as highlighted in [13], [14]. However, the transmission latency of edge computing offload tasks is frequently uncertain, primarily due to the instability of the wireless channel. The ambiguity inherent could potentially render edge offloading an ineffective recourse, as underscored in prior research [15]. Furthermore, the traditional iterative optimization algorithm is characterized by high computational complexity and lacks the ability to diverse application environments. As a result, its generalization ability is limited, as noted in [16].

Reconfigurable intelligent surface (RIS) is a remarkable technology that has the potential to enhance the quality of

This work was supported in part by the National Natural Science Foundation of China under the Grants No. 62001368, in part by the Key Research and Development Program of Shaanxi under Grant No. 2022GY-093, in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University (No.2023D13) and in part by the Researchers Supporting Project No.(RSPD2024R681), King Saud University, Riyadh, Saudi Arabia. (Corresponding author: Dongyang Xu.)

Tiantian Zhang, Dongyang Xu are with School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China. (email: tiantianzhang@stu.xjtu.edu.cn; xudongyang@xjtu.edu.cn.)

Amr Tolba is with the Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia. (email: atolba@ksu.edu.sa)

Keping Yu is with the Graduate School of Science and Engineering, Hosei University, Tokyo 184-8584, Japan. (email: keping.yu@ieee.org)

Houbing Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250 USA. (email: h.song@ieee.org)

Shui Yu is with the School of Computer Science, University of Technology Sydney, NSW, Australia. (email: Shui.Yu@uts.edu.au)

Copyright (c) 2024 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

wireless channels by dynamically modifying the phase weights of RIS elements. This capability has been highlighted in several studies [17]–[19]. Specifically, incorporating a low-cost and efficient wireless communication system has the potential to greatly enhance the network transmission capacity by capitalizing on the advantages offered by both direct links and RIS-reflected links [20], [21], [46]. It is noteworthy that RIS technology is not confined to specific access techniques and can also be seamlessly integrated with other technologies widely studied in academia in recent years, such as non-orthogonal multiple access (NOMA) [47]–[49], rate splitting multiple access (RSMA) [50], and others. RSMA is another multiple access technology that can be configured for various scenarios. It achieves interference management through the splitting of user messages and non-orthogonal transmission of common messages for multi-user decoding and private messages for individual user decoding. Therefore, the deployment of RIS in wireless communication systems is an effective approach to enhance ability of MEC. However, accurately estimating the channel state information (CSI) becomes a challenging problem, particularly when dealing with large scale RIS elements [22]–[24].

Optimally allocating MEC resources based on practical scenarios is of utmost importance [25]. Artificial intelligence (AI) has been widely utilized in MEC offloading applications due to its robustness towards dynamic channel. However, traditional AI schemes often rely on numerous collected samples to train their models, and performance is constrained by the availability of datasets [26]. In contrast, deep reinforcement learning (DRL) offers an auspicious approach to tackle the MEC offloading optimization problem. By utilizing adaptive feedback from the environment, DRL demonstrates potential in improving the efficiency and effectiveness of offloading decisions [55], [56].

In summary, deploying low-cost and energy-efficient RIS provides a means of manipulating the wireless channel, thereby improving its reliability and guaranteeing the capacity [29]. The local computing tasks offloading does not directly have a negative impact on the deployment of RIS. However, the involvement of RIS can improve the transmission capability during the process of offloading tasks. The benefits brought by RIS are realized through the reconstruction of the wireless channel, enabling remarkable communication performance. Consequently, even in scenarios where task offloading is not implemented, the deployment of RIS can still be instrumental in effectively improving BS coverage, enhancing throughput, or ensuring superior quality of service (QoS) for edge users. To address the aforementioned challenges, this paper proposes a cloud-edge offloading framework aided by a RIS while minimizing energy consumption. Additionally, we consider the convergence of average offloading latency as long-term performance evaluation factors [30]. To achieve this, we integrate a DRL framework into our proposed system, which optimizes the offloading latency and energy consumption efficiently.

A. Contribution

To tackle the challenges of cloud-edge offloading optimization in RIS-aided networks, we investigate a hybrid methodol-

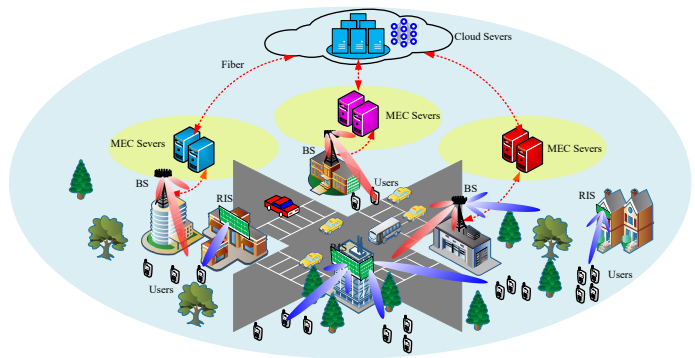


Fig. 1. The architecture of typical cloud-edge offloading system aided by RIS.

ogy that concurrently optimizes RIS beamforming and intelligent offloading of computational tasks. This approach aims to enhance average latency and energy efficiency by involving DRL. With the assumption of perfect CSI can be obtained, a joint optimization problem for RISs and BSs has been established to provide the remarkable transmission capacity. Furthermore, the application of novel DRL framework that prioritizes both the reliability and convergence of the average latency and energy consumption has been established. This paper aims to provide valuable insights into the potential application of RIS and DRL in the optimization of MEC task offloading framework. The detailed contributions are summarized as follows.

- To improve the overall offloading efficiency, this paper proposes a RIS-aided MEC offloading framework within cell-free network that optimizes the average latency and energy consumption. The central concept entails removing the limitations imposed by conventional BS and substituting some of them with RIS to improve system capacity and coverage.
- To tackle the complex beamforming problem of novel cell-free system consist of BSs and RIS, a joint beamforming optimization scheme has been proposed. This scheme employs an iterative algorithm that enables simultaneous phase shift optimization of both RIS and BS. The objective is to maximize the overall capacity of the cell-free network, thereby ensuring efficient task offloading.
- In order to improve the robustness and efficiency of the task offloading scheme, we introduce a novel optimization framework based on actor-critic. Our approach aims to minimize average latency and energy consumption, while adhering to network capacity and power constraints. We utilize deep neural networks (DNNs) to determine the optimal offloading decision and employ the critic component to calculate the associated reward. Through iterative updates to the state information and generation of training datasets, we aim to achieve convergence of offloading latency and energy consumption to a steady state.

B. Organization and Notation

Organization: The organization of this paper is summarized as follows. The related works are discussed in Section II. Section III describes the system model of the wireless communication system aided by RIS and corresponding computing task offloading framework. In Section IV, we discuss the joint optimization framework of beamforming and task offloading based on reinforcement learning. The detail analysis of experiment results about performance of proposed scheme is presented in Section V. Finally, remarkable conclusions are drawn in Section VI.

Notation: In our paper, vectors and matrices are denoted by lower and upper-case, respectively. $[\bullet]^H$ represents the conjugate transpose operation; $[\bullet]^T$ denotes the transpose operation; $[\bullet]^{-1}$ denotes the inverse operation; $\text{diag}[\bullet]$ represents the diagonal operation; σ^2 denotes variance of Gaussian noise; $\|\bullet\|$ represents the norm's operation; $\mathcal{CN}(0, \sigma^2)$ denotes the distribution of Gaussian noise;

II. RELATED WORKS

In recent years, MEC technologies have attracted significant attention from academic researchers and have been applied in various application. The deployment of numerous computing resources at edge, enabling low-latency wireless access to these nodes [51]. Therefore, MEC has enabled the deployment of computing-intensive tasks on lightweight terminals with computing constraints. The primary research areas include offloading decision, resource allocation, and mobility management. Liu *et al.* [11] proposed a joint wireless capacity and computation resource allocation framework to improve energy consumption efficiency in IoT communication systems. They utilized non-orthogonal multiple access (NOMA) to ensure high connectivity and decoupled the optimization problem into two parts: wireless resource and computation optimization.

To evaluate the efficiency of air ground integrated computing task offloading, authors in [12] involved unmanned aerial vehicles (UAVs) to provide more flexibility computing resources. Furthermore, a game combined multi-agent deep deterministic policy gradient (MADDPG) approach is proposed to model the mixed integer non-linear trajectory optimization problem. Simulation results have demonstrated that the proposed scheme could improve the system efficiency. Considering the resource-hungry and compute-intensive in current IoT system, authors in [15] proposed meta reinforcement learning (meta-RL) to solve the migration problem among different clients. Experiment results have confirmed that it could achieve near-optimal offloading decisions and improve the migration ability. Zuo *et al.* [31] proposed an untrusted MEC framework based on blockchain in networks, which can enhance security and privacy of the system. The authors then designed a nonce selection algorithm to maximize the utility of users. To explore the low-latency and computing efficiency constraints in automated vehicles networks, authors in [32] proposed a federated reinforcement learning based offloading framework. Experiment results have demonstrated that the proposed scheme could improve the task offloading efficiency significantly. Although MEC offloading technology

has been extensively developed, the system's performance still cannot meet the requirements of various emerging applications due to the bottleneck of wireless transmission capacity.

To evaluate the performance of RISs deployment, authors in [33] have analyzed the RIS-user/BS association problem to improve the system capacity. Experiment results have confirmed that the proposed scheme could obtain remarkable gains towards the benchmark schemes. Zhang *et al.* [22] investigated the capacity region of a RIS-aided communication system and compared the performance of distributed and centralized deployment. The numerical results showed that centralized scheme can achieve better performance. In [34], the authors evaluated the performance of different phase shifts, taking into account the practical requirements of deployed RISs. They utilized achievable data rate as a metric and conducted simulations to validate the necessity of specific phase shifts for achieving certain data rates. Wang *et al.* [35] investigated the upper bound of system capacity and outage probability in wireless systems assisted by RISs. They specifically examined the influence of various configurations, including the number of RISs, elements number per RIS, and channel characteristics. Through exhaustive simulations, they found that these factors have a significant impact on the performance of RIS-aided systems. To overcome the limitation of wireless capacity, Zhang *et al.* [17] proposed a novel framework, called cell-free aided by RISs, which can significantly improve the capacity. The authors designed a joint optimization scheme. Moreover, the cell-free network can guarantee wireless coverage and reduce inter-cell interference. Simulation results confirmed the effectiveness of proposed approach, demonstrating a significant improvement by employing RISs. Authors in [36] introduces the STAR-RIS technique and incorporates reinforcement learning to enhance the throughput maximization of RSMA networks. Experimental results demonstrate a significant improvement in overall system rate and user fairness with the introduction of STAR-RIS. Authors in [37] integrate RSMA into MEC systems, proposing dynamic rate partitioning to offload tasks to MEC servers. This study also provides an expression for the optimal rate partitioning parameters. Besides the joint optimization of offloading rate and time are involved to maximize the computation success probability in RSMA framework [38]. Although the RIS technology has received extensive research, there are still shortcomings in integrating it with other fields.

The conventional optimization scheme is limited in its ability to solve complex optimization problems. Hu *et al.* [26] proposed a novel block coordinate descending (BCD) algorithm to reduce the complexity of a multi-user MEC system. Additionally, two different machine learning (ML) methods are utilized to estimate the CSI and location information. The results have demonstrated the remarkable offloading efficiency. Feng *et al.* [39] introduced a min-max cost optimization model to achieve latency convergence in distributed machine learning with mobile edge computing. Then, they decomposed it into different sub-problems. Their results demonstrated the effectiveness of the proposed approach. Mao *et al.* [40] proposed a secure offloading framework for MEC aided by RIS. The authors investigated the offloading efficiency under constraints and jointly optimized the problem by utilizing

convex approximation. The results demonstrated the feasibility of the proposed framework in achieving secure and efficient offloading in MEC systems. Furthermore, RIS-assisted MEC task offloading systems are established in [41], [42]. The primary emphasis is on leveraging RIS to provide a robust wireless transmission channel for edge computing services. Yan et al. [30] proposed a DRL framework to jointly solve the offloading task under a time-varying environment. The authors designed a sophisticated actor network to achieve the offloading decision task, while a critic network was applied to evaluate the decision and establish the training datasets. However, traditional iterative optimization algorithms are inadequate for achieving joint intelligent offloading decisions and optimizing energy consumption and latency. Moreover, they lack the capability to dynamically adapt to dynamic environments and often rely heavily on computing resources. As a result, finding an efficient intelligent offloading solution based on DRL that can achieve both average latency and energy efficiency in RIS-aided systems remains a challenging problem.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we aim to provide a comprehensive description of the communication system model and the cloud-edge computing task offloading model employed within the proposed DRL framework. Furthermore, we formulate a hybrid cost-minimization optimization problem to address the challenges.

A. Communications model

As illustrated in Fig. 1, a cloud-edge offloading scenario which is augmented with RISs aided cell-free network is contemplated. The system under consideration is characterized by the deployment of both BS and RISs. Additionally, MEC servers are strategically positioned at the network's edge to offer computational resources. Furthermore, a cloud server is deployed to provide highly capable cloud computing services. Specifically, our system consists of N_{BS} BSs, N_{RIS} RISs, and K users. Besides, we use M_{USER} to denote the number of antennas for each user and M_{RIS} to denote the number of elements in the RIS. Considering the account factors such as feasibility in practical scenarios and user-side decoding complexity, we choose the integration of OFDMA with RIS as the basic wireless communication model. We assume there exists L different multi-carrier during the transmission. With the assumption that the transmitted symbol $x_{l,k}$ has a normalized power. At the transmitter side, the beamforming weight $w_{n_{bs},l,k}$ at the n_{bs} -th BS is utilized to weight all transmitted symbols. Consequently, the transmitted symbol can be represented as

$$\mathbf{x}_{n_{bs},l} = \sum_{k=1}^K \mathbf{W}_{n_{bs},l,k} x_{l,k}, \quad (1)$$

where $\mathbf{H}_{n_{bs},k,l}^H$ denotes the direct link between BS and user. Besides, the BS-RIS-user transmission channel consists

of two different links. Therefore, the entire channel can be denoted as [17]

$$\mathbf{h}_{n_{bs},k,l}^H = \underbrace{\mathbf{H}_{n_{bs},k,l}^H}_{\text{Direct link}} + \underbrace{\sum_{t=1}^{N_{RIS}} \mathbf{P}_{t,k,l}^H \Phi_t^H \mathbf{U}_{b,t,l}}_{\text{Reflect links}}, \quad (2)$$

where $\mathbf{H}_{n_{bs},k,l}^H$, $\mathbf{U}_{b,t,l}$, and $\mathbf{P}_{t,k,l}^H$ denote the transmission links between BS and user, BS and RIS, and the channel between RIS and user, respectively. The phase shift of RIS elements can be written as

$$\Phi_{n_{ris}} \triangleq \text{diag}(\phi_{n_{ris},1}, \dots, \phi_{n_{ris},M_{RIS}}), \forall n_{ris} \in \mathcal{N}_{RIS}, \quad (3)$$

where $\phi_{n_{ris},m_{ris}}$ refers to the phase shift weight of individual elements in the RIS. To simplify the analysis, we assume that the phase weight of the RISs can be independently adjusted and is continuous. More specifically, the signal received by the n_{bs} -th receiver can be mathematically expressed as

$$\begin{aligned} \mathbf{y}_{n_{bs},k,l} &= \mathbf{h}_{n_{bs},k,l}^H \mathbf{x}_{n_{bs},l} \\ &= \left(\mathbf{H}_{n_{bs},k,l}^H + \sum_{t=1}^{N_{RIS}} \mathbf{P}_{t,k,l}^H \Phi_t^H \mathbf{U}_{n_{bs},t,l} \right) \mathbf{x}_{n_{bs},l}. \end{aligned} \quad (4)$$

Taking into account the impact of channel noise $n \in \mathcal{CN}(0, \sigma^2)$, the expression for the received signal in (4) can be reformulated as

$$\mathbf{y}_{k,l} = \sum_{n_{bs}=1}^{N_{BS}} \mathbf{y}_{n_{bs},k,l} + \mathbf{n}_{k,l}, \quad (5)$$

The received signal can be decomposed into two components: the useful and interference signals. Further expanding the expression for user k , we have

$$\begin{aligned} \mathbf{y}_{k,l} &= \sum_{n_{bs}=1}^{N_{BS}} \sum_{j=1}^K \mathbf{h}_{n_{bs},k,l}^H \mathbf{w}_{n_{bs},l,j} x_{l,j} + \mathbf{n}_{k,l} \\ &= \underbrace{\mathbf{h}_{k,l}^H \mathbf{w}_{l,k} x_{l,k}}_{\text{Wanted signal}} + \underbrace{\sum_{j=1, j \neq k}^K \mathbf{h}_{k,l}^H \mathbf{w}_{l,j} x_{l,j}}_{\text{Interference + Noise}} + \mathbf{n}_{k,l}, \end{aligned} \quad (6)$$

Then, the signal-to-interference-plus-noise ratio (SINR) for the k -th user can be acquired through the following expression.

$$\beta_{k,l}^{SINR} = \frac{\mathbf{w}_{l,k}^H \mathbf{h}_{k,l} \mathbf{h}_{k,l}^H \mathbf{w}_{l,k}}{\sum_{j=1, j \neq k}^K \mathbf{h}_{k,l}^H \mathbf{w}_{l,j} \left(\mathbf{h}_{k,l}^H \mathbf{w}_{l,j} \right)^H + n_{k,l}}, \quad (7)$$

Consequently, the capacity of the whole system can be expressed as:

$$R_{\text{Total}} = \sum_{k=1}^K \sum_{l=1}^L B_k \log_2 (1 + \beta_{k,l}^{SINR}), \quad (8)$$

where $\beta_{k,l}^{SINR}$ denotes the SINR of user k . Furthermore, B_k represents the working bandwidth of the k -th user, while R_{Total} denotes the overall system transmission capacity under the assumption of perfect CSI. The objective of this section is to maximize the system capacity by jointly optimizing the phase shifts of the RIS and the BS. This can be formulated as

an optimization problem, given by

$$\mathcal{P}^1 : \max_{\Phi, \mathbf{W}} R_{\text{Total}}(\Phi, \mathbf{W}) = \sum_{k=1}^K B_k \log_2 (1 + \beta_k^{SINR}) \quad (9a)$$

$$\text{s.t.} \quad D_1 : \sum_{k=1}^K \sum_{l=1}^L \|\mathbf{w}_{n_{bs},l,k}\|^2 \leq P_{n_{bs}}^{max}, \quad (9b)$$

$$D_2 : 0 \leq \angle(w_{n_{bs},l,k}) \leq 2\pi, \quad (9c)$$

$$D_3 : 0 \leq \Phi_{n_{ris},m_{ris}} \leq 2\pi, \quad (9d)$$

$$D_4 : \|\mathbf{n}\|^2 \leq \sigma^2, \quad (9e)$$

where $P_{n_{bs}}^{max}$ denotes the maximum transmit power of the BS and $\angle(\cdot)$ represents the phase weight of $w_{n_{bs},l,k}$. However, the joint optimization of the beamforming weight of BS, represented by \mathbf{W} , and the passive phase shift of the RIS, denoted as Φ , is a challenging problem. As a result, we decompose the optimization problem \mathcal{P}^1 into various subproblems and provide a detailed illustration of the optimization process.

B. Partial offloading computing model

In this section, the offloading computing model is illustrated in Fig. 2. There will be different computing data streams at the wireless terminals and the computing tasks can be offloaded accordingly. The offloading policy, especially partial offloading, is a highly complex process influenced by various factors. User preferences, wireless channel capacity, local computational capability, edge server computational capability, and overall system availability are among the factors that determine the ultimate offloading decision for computational tasks (partial, random, or complete offloading) [53], [54]. Except for local computing tasks, all others computing tasks offloaded to edge servers or cloud servers need to be transmitted through wireless channels. RIS can improve the wireless capacity, providing a foundational transmission guarantee for the task offloading. When huge computational tasks are offloaded to edge or cloud, those released resources can significantly enhance the computational efficiency of local tasks. For instance, with numerous complex computational tasks offloaded, additional resources can be obtained to maintain efficient system operations, providing better latency, energy efficiency, and Qos. At the same time, the transmission process of tasks also consumes the system's transmission resources. If local computational tasks rely on wireless aided by RIS to provide services, these local computational services may be affected, leading to an inability to deliver normal and timely services to users.

The system comprises K users, N_{BS} BSs, and N_{RIS} RISs. Furthermore, we assume that the system operates in discrete time frames, which serve as the smallest unit for task offloading. The offloading tasks are divided into two parts: one is transmitted over wireless channels to the cloud-edge servers for computation, and the other is processed locally on the user device. For the k -th user, we use $D_{a,k}^t$, $f_{local,k}^t$, $f_{edge,k}^t$ and $f_{cloud,k}^t$ to represent the data arrival, local computation capacity, edge, and cloud computation capacity at time frame t . With the assumption that $D_{a,k}^t$ follows a general Gaussian distribution, satisfying a basic assumption. Then, the

local computation capacity and energy consumption can be expressed as [25]

$$D_{local,k,L}^t = f_{local,k}^t T / \phi, \forall o_k^t = 0, \quad (10)$$

$$E_{local,k,L}^t = (\alpha_0 (f_{local,k}^t)^3) T, \forall o_k^t = 0, \quad (11)$$

where T represents the duration of the processing subframe, and o_k^t denotes the computation mode for user k at time instant t . Specifically, the values 0, 1, and 2 correspond to local computing, edge computing, and cloud computing, respectively. $D_{local,k,L}^t$ represents the data queue that requires local processing for user k . The variable ϕ denotes the number of computing cycles required to process one bit of a computing task, while α represents the energy consumption efficiency factor. Similarly, we can formulate the edge computing data queue and energy consumption as follows, analogous to (10) and (11)

$$D_{trans,k,L}^t = \frac{1}{v_u} \sum_{l=1}^L B_k \log_2 (1 + \beta_{k,l}^{SINR}), \quad (12)$$

$$D_{edge,k,L}^t = f_{edge,k}^t / \phi, \forall o_k^t = 1, \quad (13)$$

$$E_{edge,k,L}^t = P_{trans,k}^t \tau_k^t T + (\alpha_1 (f_{edge,k}^t)^3) T, \forall o_k^t = 1, \quad (14)$$

where v_u represents the bandwidth utilization ratio, and τ_k^t denotes the transmission duration for user k at time instant t . The variable $f_{edge,k}^t$ represents the computation capability of the edge server, while $P_{trans,k}^t$ represents the transmission power. With the assumption that latency in transmitting the result information after an edge computing task is negligible since the computed information is typically small. For cloud computing, all computation data is first transmitted to the MEC servers and then forwarded to the remote cloud server through fiber transmission. As a result, the average data computation capacity and energy consumption of the cloud server can be expressed as

$$D_{cloud,k,L}^t = f_{cloud,k}^t / \phi, \forall o_k^t = 2, \quad (15)$$

$$E_{cloud,k,L}^t = P_{trans,k}^t \tau_k^t T + P_{edge,k}^t \zeta_k^t T + (\alpha_2 (f_{cloud,k}^t)^3) T, \forall o_k^t = 2, \quad (16)$$

where $f_{cloud,i}$ represents the computational capacity of the cloud server, while $P_{edge,k}^t$ denotes the transmission power of the MEC server. Furthermore, the fiber link between the MEC and cloud servers is assumed to have an ideal transmission rate $D_{fiber,k,L}^t$. For the sake of simplicity, we neglect the transmission delay and energy consumption associated with cloud computation. This is justified by assuming that the remote cloud server has ample transmission and computation capacity. As a result, the data queue to be processed during the entire offloading task can be simplified as follows

$$D_k^t \triangleq o_{i=0}^t D_{local,k,L}^t + o_{i=1}^t \tau_k^t (D_{edge,k,L}^t + D_{trans,k,L}^t) + o_{i=2}^t \zeta_k^t D_{trans,k,L}^t, \quad (17)$$

where D_k^t denote the average data queue needed to be processed. To comprehensively evaluate the performance of the

entire offloading strategy, we define the average equivalent delay and average energy consumption as metrics to measure the system's performance.

$$D_{latency,k}^t = \max\left\{\frac{s_k^{local}\phi}{f_{local,k}^t}, \frac{s_k^{edge}\phi}{f_{edge,k}^t} + \frac{s_k^{edge} + s_k^{cloud}}{D_{trans,k,L}^t}\right\}, \quad (18)$$

$$E_{total,k}^t = o_{k=0}^t E_{local,k,L}^t + o_{k=1}^t E_{edge,k,L}^t + o_{k=2}^t E_{cloud,k,L}^t, \quad (19)$$

where $D_{latency,k}^t$ denotes the maximum latency among the local, edge, and cloud computation offloading is denoted, while $E_{total,k}^t$ represents the average energy consumption for completing all the computing tasks in the existing queue without any additional computing tasks coming in. These two metrics are the core evaluation indicators in computational offloading tasks. In addition to considering the length of data queue to be processed. To accomplish this, a dynamic data queue model can be constructed.

$$\begin{aligned} D_{q,k}(t+1) &= D_{q,k}(t) - D_i^t + D_{a,k}^t, k = 1, 2, \dots \\ &= s_k^{local} + s_k^{edge} + s_k^{cloud}, \end{aligned} \quad (20)$$

where D_k^t denotes the system's computation ability, and the initial data queue is assumed as $D_{q,k}(0) = 0$. With these assumptions in mind, the objective is to minimize the average computational latency $D_{latency,i}^t$ through the joint optimization of the offloading task $S = [s_1, s_2, \dots, s_K]^T$, the beamforming weight of the BSs W , and the phase shift of the RIS Φ . Furthermore, the computation latency optimization problem can be summarized as (21).

$$\mathcal{P}^2: \underset{\mathbf{o}, \mathbf{\tau}, \mathbf{f}, \mathbf{E}}{\text{minimize}} \lim_{N \rightarrow \infty} \frac{1}{N} \cdot \sum_{t=1}^N \sum_{k=1}^K c_k D_{latency,k}^t \quad (21a)$$

$$\text{s.t. } D_1: D_{edge,k,L}^t + D_{cloud,k,L}^t \leq R_{Total}, P^1 \quad (21b)$$

$$D_2: D_{local,k,L}^t + D_{edge,k,L}^t + D_{cloud,k,L}^t \leq D_{q,k}(t), \forall k, t \quad (21c)$$

$$D_3: 0 \leq \sum_{i=1}^K (\tau_k + \zeta_k) \leq 1, \forall k, t, \quad (21d)$$

$$D_4: 0 \leq f_{local,k}^t \leq f_{local}^{max}, \forall k, t, \quad (21e)$$

$$D_5: 0 \leq f_{edge,k}^t \leq f_{edge}^{max}, \forall k, t, \quad (21f)$$

$$D_6: 0 \leq f_{cloud,k}^t \leq f_{cloud}^{max}, \forall k, t, \quad (21g)$$

$$D_7: o_k^t \in \{0, 1, 2\}, \forall k, t, \quad (21h)$$

$$D_8: P_{trans,k} \leq P_{trans}^{max}, P_{edge,k} \leq P_{edge}^{max}, \forall k, t, \quad (21i)$$

$$D_9: 0 \leq E_{total,k}^t \leq E_{thre}, \forall k, t, \quad (21j)$$

where c_i denotes the weight of user k . The constraint (21b) limits the transmission capacity of the wireless communication system. R_{Total} represents the total transmission capacity of the RIS-aided wireless communication system. The constraint (21c) represents the data computation task queue constraint, indicating that the maximum data processing capacity of the system cannot exceed the corresponding data queue. Constraint (21d) imposes a time constraint on the offloading strategy. Constraints (21e), (21f), and (21g) restrict the maximum computation frequency of the local terminal, MEC, and cloud server, respectively. Here, f_{local}^{max} , f_{edge}^{max} , and f_{cloud}^{max}

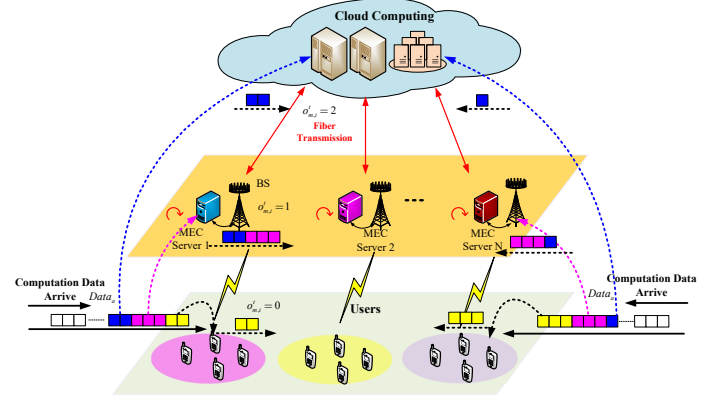


Fig. 2. The typical architecture of cloud-edge-client offloading scheme.

denote the maximum computation capabilities of the local, edge, and cloud server, respectively. Additionally, constraint (21h) indicates the computation offloading mode. Constraints (21i) restricts the maximum transmission power of the local terminal and MEC server, while constraint (21j) limits the overall system energy consumption not to exceed a given threshold E_{thre} . It can be observed from (18) and (20) that the computation data queue in the system exhibits stability equivalent to the average computation latency. Little's law states that with random task computation data arriving, there exist upper and lower bounds for latency. Therefore, the average system queue length can be utilized as an equivalent measure to characterize the average computation latency.

IV. OPTIMIZATION SOLUTION FOR COMMUNICATION AND COMPUTATION OFFLOADING

In this section, we present the optimization solution for the communication and computation offloading model with the objective of minimizing the average offloading latency and energy consumption. To achieve this, the proposed RIS-aided beamforming optimization problem is divided into several subproblems in Section IV-A. Then, a novel DRL framework is employed to optimize the computation offloading problem with fixed W and Φ in Section IV-B. Finally, we discuss the complete optimization algorithm in detail.

A. Joint optimization for RIS precoding

In particular, it is assumed that perfect CSI is available to both the BS and the RIS. Based on this assumption, a joint precoding weight is designed using CSI to maximize the system capacity and support edge computation task offloading. To accomplish this, the maximization optimization problem \mathcal{P}^1 is decoupled into subproblems using the Lagrangian dual method, as discussed in [43]. Furthermore, introducing an auxiliary variable $\delta \in \mathbb{R}$ to the problem \mathcal{P}^1 , and rewrite it

as

$$\widehat{\mathcal{P}}^1 : \max_{\Phi, \mathbf{W}, \delta} g(\Phi, \mathbf{W}, \delta) \quad (22a)$$

$$\text{s.t. } D_1 : \sum_{k=1}^K \sum_{l=1}^L \|\mathbf{w}_{n_{bs}, l, k}\|^2 \leq P_{n_{bs}}^{max}, \quad (22b)$$

$$D_2 : 0 \leq \text{angle}(w_{n_{bs}, l, k}) \leq 2\pi, \quad (22c)$$

$$D_3 : 0 \leq \Phi_{n_{ris}, m_{ris}} \leq 2\pi, \quad (22d)$$

$$D_4 : \|\mathbf{n}\|^2 \leq \sigma^2, \quad (22e)$$

where the auxiliary variable δ refers to $(\delta_{1,1}, \delta_{1,2}, \dots, \delta_{K,L})$ and is related to the ratio term in (7). The equivalent objective function $g(\Phi, \mathbf{W}, \delta)$ can be denoted as

$$\begin{aligned} g(\Phi, \mathbf{W}, \delta) = & \sum_{k=1}^K \sum_{l=1}^L \beta_k \ln(1 + \delta_{k,l}) - \sum_{k=1}^K \sum_{l=1}^L \beta_k \delta_{k,l} \\ & + \sum_{k=1}^K \sum_{l=1}^L \beta_k (1 + \delta_{k,l}) g_{k,l}(\Phi, \mathbf{W}), \end{aligned} \quad (23)$$

where $g_{k,l}(\Phi, \mathbf{W})$ is defined by following

$$g_{k,l}(\Phi, \mathbf{W}) = \frac{\mathbf{w}_{l,k}^H \mathbf{h}_{k,l} \mathbf{h}_{k,l}^H \mathbf{w}_{l,k}}{\sum_{j=1}^K \mathbf{h}_{k,l}^H \mathbf{w}_{l,j} \left(\mathbf{h}_{k,l}^H \mathbf{w}_{l,j} \right)^H + n_{k,l}}, \quad (24)$$

where $g_{k,l}$ is a concave and differentiable function with respect to δ , given that the other variables (Φ, \mathbf{W}) are fixed in (23). Furthermore, it can be observed that once the optimized solution $\delta_{k,l}^*$ is obtained from equation (24), the resulting expression will be equivalent to equation (9). It is important to note that the channel noise $n_{k,l}$ is also taken into consideration in (23).

Proof. See Appendix A. ■

Hence, the optimal beamforming weights of the BS and RIS can be obtained by solving the equivalent problem. Firstly, the optimized $\delta_{k,l}$ is calculated as described above. Then, the optimized weights of the BS \mathbf{W} , and the RIS Φ , can be determined. To obtain \mathbf{W} , the $\delta_{k,l}$ and Φ are fixed, and the corresponding optimization problem in (23) is solved. This leads to a simplified optimization problem that can be denoted as

$$\begin{aligned} \widehat{\mathcal{P}}^{\text{BS}} : \max_{\mathbf{W}} & \sum_{k=1}^K \sum_{l=1}^L \beta_k \left(1 + \widehat{\delta}_{k,l} \right) g_{k,l}(\widehat{\Phi}, \mathbf{W}) \\ \text{s.t. } & 22(b), 22(c), 22(e), \end{aligned} \quad (25)$$

where \mathbf{W} represents the beamforming weights that need to be optimized. It is evident that this sub-problem, denoted as $\widehat{\mathcal{P}}^{\text{BS}}$, remains challenging to solve using common fractional programming algorithms. To address the non-convexity and high complexity of this high-dimensional problem, we employ the multidimensional complex quadratic transform (MCQT) and the alternating direction method of multipliers (ADMM) [17], [44]. Similarly, by fixing \mathbf{W} and $\delta_{k,l}$, we can obtain the optimized RIS phase shift denoted as Φ^* . The equivalent

optimization problem can be rewritten as follows

$$\begin{aligned} \widehat{\mathcal{P}}^{\text{RIS}} : \max_{\Phi} & \sum_{k=1}^K \sum_{l=1}^L \beta_k \left(1 + \widehat{\delta}_{k,l} \right) g_{k,l}(\Phi, \widehat{\mathbf{W}}) \\ \text{s.t. } & 22(d), 22(e), \end{aligned} \quad (26)$$

where $\widehat{\mathbf{W}}$ and $\widehat{\delta}_{k,l}$ represent the corresponding fixed BS weight and auxiliary variable that are optimized in the previous process. Similar to the optimization sub-problem in (25), MCQT and ADMM can also be applied to further decouple and solve the optimization problem [17]. Additionally, the joint beamforming optimization problem can be solved by iteratively addressing the problems in (23)-(26) until the system capacity reaches convergence. It is worth noting that an improved system capability can lead to greater benefits in cloud-edge offloading.

B. Offloading optimization based on reinforcement learning

In this section, a hybrid DRL framework is established to solve the offloading optimization problem \mathcal{P}^2 . Firstly, it is assumed that the environment is nearly quasi-static, allowing accurate modeling of channel fading, transmission capacity, and offloading data queues. To evaluate the computation rate of the system, a computing ability factor C_k^t can be defined as follows

$$C_k^t = \frac{o_{i=0}^t f_{local,i}^t}{\phi} + o_{i=1,2}^t (\tau_k^t + \zeta_k^t) D_{trans,k}^t, \quad (27)$$

where o_i^t represents the offloading mode, and ϕ denotes the number of basic cycles consumed per bit. Additionally, \mathcal{P}^2 can be decoupled into subproblems through Lyapunov optimization. Specifically, the energy consumption $I_k(t)$ for the k -th user can be expressed as follows

$$I_k(t+1) = \max \left(I_k(t) + \nu E_{total,k}^t - \nu E_{thre,k}, 0 \right), \quad (28)$$

where $E_{total,k}^t$ represents the average energy consumption of the system, and ν denotes the energy scale factor. Maintaining a stable energy consumption is crucial to ensure the effectiveness of the overall offloading policy. In order to meet the requirements for both latency and energy consumption, a joint Lyapunov function model and drift are proposed and can be expressed as follows

$$\begin{aligned} L(\mathbf{R}(t)) &= \kappa \left(\sum_{k=1}^K D_{q,k}(t)^2 + \sum_{k=1}^K I_k(t)^2 \right) \\ \Delta L(\mathbf{R}(t)) &= \mathbb{E}\{L(\mathbf{R}(t+1)) - L(\mathbf{R}(t)) \mid \mathbf{R}(t)\}, \end{aligned} \quad (29)$$

where κ is a constant factor, and $\Delta L(\mathbf{R}(t))$ denotes the Lyapunov drift. The objective is to maximize the system's computation ability, which is equivalent to minimizing the average latency. The system's data computation capability is employed to characterize its average latency characteristics. By utilizing the drift-plus-penalty approach, a new expression can be formulated to penalize deviations from the optimal policy in order to minimize the bound on punishment [55]. Therefore,

$\Lambda(\mathbf{R}(t))$ can be defined as

$$\Lambda(\mathbf{R}(t)) \triangleq \Delta L(\mathbf{R}(t)) - Z \cdot \sum_{k=1}^K \mathbb{E} \{c_k C_{Rate,k}^t \mid \mathbf{R}(t)\}, \quad (30)$$

where Z is the scale factors for penalty. To obtain the upper bound of the above expression, models for the data queue and the average energy consumption are established. To derive the upper bound of $\Lambda(\mathbf{R}(t))$, the drift of system average computation queue can be constructed by combining (28) and (29).

$$\begin{aligned} \Delta L(D_q(t)) &\triangleq \mathbb{E}\{L(D(t+1)) - L(D(t)) \mid \mathbf{R}(t)\} \\ &= \kappa \sum_{k=1}^K \mathbb{E}[(D_{a,k}^t - D_k^t)^2] + \sum_{k=1}^K D_{q,k} \mathbb{E}[(D_{a,k}^t - D_k^t) \mid \mathbf{R}(t)], \end{aligned} \quad (31)$$

Similarly, the drift of system's average energy consumption can be represented as:

$$\begin{aligned} \Delta L(I(t)) &\triangleq \mathbb{E}\{L(I(t+1)) - L(I(t)) \mid \mathbf{R}(t)\} \\ &= \kappa \sum_{k=1}^K \mathbb{E}[(E_{total,k}^t - E_{thre,k})^2] + \\ &\quad \sum_{k=1}^K I_k \mathbb{E}[(E_{total,k}^t - E_{thre,k}) \mid \mathbf{R}(t)], \end{aligned} \quad (32)$$

Then, the upper bound of $\Delta L(\mathbf{R}(t))$ can be acquired by combining (31) and (32).

$$\begin{aligned} \Delta L(\mathbf{R}(t)) &\leq O + \sum_{k=1}^K D_{q,k}(t) \mathbb{E}[(D_{a,k}^t - D_k^t) \mid \mathbf{R}(t)] + \sum_{k=1}^K \\ &\quad (I_k(t) \mathbb{E}[E_{total,k}^t - E_{thre,k} \mid \mathbf{R}(t)] - Z \cdot \mathbb{E}\{c_k C_{Rate,k}^t \mid \mathbf{R}(t)\}), \end{aligned} \quad (33)$$

where O represents the constraint on the average data and energy consumption queue, which is defined in (34).

$$\kappa \left(\sum_{k=1}^K \mathbb{E}[(D_{a,k}^t - D_k^t)^2] + \mathbb{E}[(E_{total,k}^t - E_{thre,k})^2] \right) \leq O, \quad (34)$$

The maximization of the queue upper bound is employed to impose constraints on energy consumption. In particular, it is noteworthy that O and $\mathbf{R}(t)$ are independent of each other, allowing us to exclude O from the subsequent optimization process. Moreover, the joint offloading and resource allocation decisions directly impact the bound $\Delta L(\mathbf{R}(t))$. Consequently, the system's reward can be represented as $G_r(\cdot)$ by substituting (34) into (33).

$$\begin{aligned} G_r &\approx - \sum_{k=1}^K D_{q,k}(t) \mathbb{E}[(D_{a,k}^t - D_k^t) \mid \mathbf{R}(t)] - \sum_{k=1}^K (I_k(t) \\ &\quad \mathbb{E}[E_{total,k}^t - E_{thre,k} \mid \mathbf{R}(t)] + Z \cdot \mathbb{E}\{c_k C_k^t \mid \mathbf{R}(t)\}) \\ &= \sum_{k=1}^K (D_{q,k}(t) + Z c_k) C_k^t \mid_{T=1} - I_k(t) E_{total,k}^t, \end{aligned} \quad (35)$$

where C_k^t denotes the data computation ability at time frame t . For simplicity, we assume a time duration of $T = 1$.

The hybrid indicator G_r considers the average data queue and energy consumption. The objective of this function is to strike a balance between the average system delay and energy consumption during the optimization process. Eq.(35) aims to select clients with superior computational abilities while penalizing those with average power consumption exceeding the threshold. Therefore, at each time instant t , optimization problem (21) can be decomposed into subproblems and equivalently transformed into the following problems

$$\mathcal{P}^3 : \underset{\mathbf{o}^t, \tau^t, \mathbf{f}^t, \mathbf{E}^t}{\text{maximize}} \sum_{k=1}^K (D_{q,k}(t) + Z c_k) C_k^t \mid_{T=1} - I_k(t) E_{total,k}^t \quad (36a)$$

$$\text{s.t. } D_1 : D_{edge,k,L}^t + D_{cloud,k,L}^t \leq R_{Total}, \quad (36b)$$

$$D_2 : D_{local,k,L}^t + D_{edge,k,L}^t + D_{cloud,k,L}^t \leq D_{q,k}(t), \forall k, t \quad (36c)$$

$$D_3 : 0 \leq \sum_{k=1}^K (\tau_k + \zeta_k) \leq 1, \forall k, t, \quad (36d)$$

$$D_4 : 0 \leq f_{local,k}^t \leq f_{local}^{max}, \forall k, t, \quad (36e)$$

$$D_5 : 0 \leq f_{edge,k}^t \leq f_{edge}^{max}, \forall k, t, \quad (36f)$$

$$D_6 : 0 \leq f_{cloud,k}^t \leq f_{cloud}^{max}, \forall k, t, \quad (36g)$$

$$D_7 : o_k^t \in \{0, 1, 2\}, \forall k, t, \quad (36h)$$

$$D_8 : P_{trans,k} \leq P_{trans}^{max}, P_{edge,k} \leq P_{edge}^{max} \forall k, t, \quad (36i)$$

$$D_9 : 0 \leq E_{total,k}^t \leq E_{thre}, \forall k, t, \quad (36j)$$

where Z represents the weight factor of the penalty variable in (29). The joint offloading optimization problem can be equivalently transformed into a reward maximization problem defined by $G_r(\cdot)$. However, it is worth noting that the above optimization problem is non-convex and involves multiple-variable constraints. Designing the computational task offloading policy under such conditions can be extremely complex. To tackle this issue, the remarkable scheme named model-free DRL framework is proposed to solve this dynamic offloading policy problems. As illustrated in Fig. 3, this DRL framework mainly includes System state, Action, Reward and Experience reply. Furthermore, the user terminal is set as intelligent agent which can acquire the best offloading decisions. Besides, the deployment of RIS can provide a better transmission capability for computing task offloading. The process related to DRL can be further summarized as follows.

State: The system state S comprises various variables related to the current environment, including the system's channel state information $\mathbf{h}_{nbs,k,l}^H$, the computation task queue $D_{q,k}(t)$, the average energy consumption $I_k(t)$, and the offloading parameters $o_i^t, f_i^t, \tau_k^t, \zeta_k^t$. These relevant parameters are a combination of continuous and discrete variables. The main objective is to acquire the optimal task offloading decisions [32]. The state space at time t can be denoted as:

$$s_t = \{\mathbf{h}_{nbs,k,l}^H, D_{q,k}(t), I_k(t), o_i^t, f_i^t, \tau_k^t, \zeta_k^t\}, \quad (37)$$

where s_t denotes the system real time status. Besides, the action part will produce an optimized offloading decision based on the input system state.

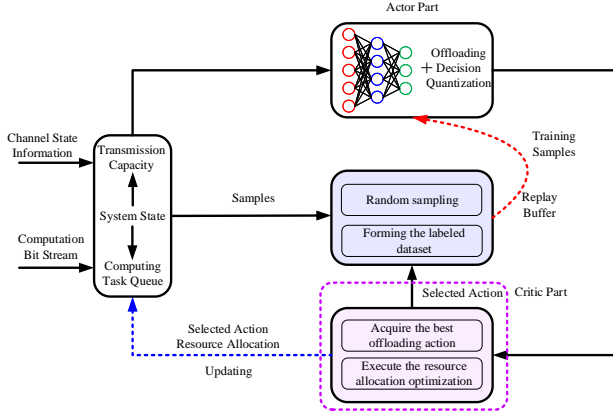


Fig. 3. The workflow of proposed offloading optimization based on deep reinforcement learning.

Action: As illustrated in Fig. 3, the actor part is responsible for generating the potential offloading decisions. Besides, all the variables in the state space and system parameters are considered as inputs. A carefully designed CNN model is responsible for generating offloading decisions o_i^t . The model parameters θ^t are randomly initialized at the start of the DRL learning process. The output of the CNN model is a continuous probability value corresponding to the Sigmoid activation function. Considering that the offloading decision of the MEC system is a binary distribution $[0, 1, 2]$. Therefore, an efficient quantization method illustrated in [55] is applied to generate the discrete offloading decision. To improve the robustness and convergence performance of the proposed DRL framework, we introduced the noise order-preserving quantization scheme to generate M different potential offloading decisions. For the first offloading decision o_1^t , the hard decision method can be obtained by

$$o_{1,i}^t = \begin{cases} 2, & 2/3 < \hat{o}_i^t \leq 1, \\ 1, & 1/3 < \hat{o}_i^t \leq 2/3, \\ 0, & 0 \leq \hat{o}_i^t \leq 1/3, \end{cases} \quad (38)$$

where \hat{o}_i^t is the output of CNN model at current time t . To obtain the next $M/2 - 1$ potential decisions, all the original outputs must be offset and rearranged to center around $[1/3, 2/3]$. Subsequently, the remaining $M/2 - 1$ offloading decisions can be achieved by

$$o_{m,i}^t = \begin{cases} 2, & \begin{cases} \hat{o}_i^t > \hat{o}_{m-1}^t & \& (\hat{o}_i^t - \hat{o}_{m-1}^t \geq (1 - \hat{o}_{m-1}^t)/3), \\ \hat{o}_i^t = \hat{o}_{m-1}^t & \& \hat{o}_{m-1}^t \geq 2/3 \end{cases} \\ 1, & \begin{cases} \hat{o}_i^t > \hat{o}_{m-1}^t & \& (\hat{o}_i^t - \hat{o}_{m-1}^t < (1 - \hat{o}_{m-1}^t)/3), \\ \hat{o}_i^t < \hat{o}_{m-1}^t & \& |\hat{o}_i^t - \hat{o}_{m-1}^t| < (\hat{o}_{m-1}^t)/3, \\ \hat{o}_i^t = \hat{o}_{m-1}^t & \& 1/3 \leq \hat{o}_{m-1}^t < 2/3 \end{cases} \\ 0, & \begin{cases} \hat{o}_i^t < \hat{o}_{m-1}^t & \& |\hat{o}_i^t - \hat{o}_{m-1}^t| \geq (\hat{o}_{m-1}^t)/3, \\ \hat{o}_i^t = \hat{o}_{m-1}^t & \& \hat{o}_{m-1}^t \leq 1/3 \end{cases} \end{cases} \quad (39)$$

where $o_{m,i}^t$ is the m -th potential offloading decisions and i denotes the corresponding clients. Besides, the mapping

function completely ensures the random fairness without bias. To enhance the DRL framework's exploration and learning convergence ability, we propose a noise model to facilitate trial and exploration.

$$\bar{o}_i^t = \mathcal{S}(\hat{o}_i^t + n), \quad (40)$$

where n denotes the normalization Gaussian noise which can enhance the exploratory nature and learning efficiency of offloading optimization. Besides, this random variable also can improve the system robustness during the experimental process. Furthermore, the Sigmoid function is represented by \mathcal{S} . Subsequently, the remaining $M/2$ offloading decisions are generated using equations (39) and (40). The training process utilizes the Adam function to update the parameter θ_t . Finally, all potential offloading decisions are evaluated by the Critic component, which selects the optimal decision.

Reward: System reward $r_t = G_r$ is an important indicator for guiding model convergence under the DRL framework. As illustrated in Fig. 3, the critic part is utilized to evaluate the potential offloading decisions. Different from the conventional CNN model-free network, an optimization problem has been involved to evaluate the offloading decisions. The criterion of critic part is maximizing the following

$$o^t = \arg \max_{o_i^t \in [0,1,2]} G_r(o_j^t, h_{nbs,k,l}, D_{a,i}^t, I_i^t), \quad (41)$$

where o_i^t is the potential offloading decisions of actor module and $a_t = o^t$ is the best offloading decision selected by critic modules. In order to improve the precision of value assessment and optimize system learning efficiency, we utilize Lagrangian optimization schemes [55] to evaluate the offloading strategy. To simplify the analysis, we neglect the transmission delay and energy consumption of the cloud server, as it is presumed to possess adequate computing capability and energy resources. Consequently, the offloading tasks primarily account for transmission delay and energy consumption. In this context, the critic component does not employ a CNN model for computation but relies on traditional optimization to perform evaluation. Therefore, within the entire actor-critic framework, there is no introduction of a target network architecture because the critic component does not require updates to its model parameters. This reduces the error introduced by the bootstrapping problem.

Experience Replay: Experience replay is a commonly utilized technique in the DRL framework to address issues related to high correlation and non-stationary distribution among training samples. It involves randomly selecting training samples to mitigate these problems and improve learning efficiency. As shown in Fig. 3, a valid training sample s_t, a_t, r_t, s_{t+1} is formed from the selected action a_t , current state space s_t , reward r_t , and next state space s_{t+1} . These training batches are then used to train the CNN network model in the actor component. By executing the final decision calculated in the previous step, the optimal offloading decision and resource allocation can be obtained. The overall optimization process of the system is summarized in Algorithm 1¹.

¹https://github.com/TiantianZhang/RIS_DRL_MEC

Algorithm 1 RIS-aided Cloud Edge Offloading Joint Optimization (CEOJO) Algorithm

Input: Channel information $h_{nbs,k,l}$; Average queue of data and energy consumption (D_i^t, I_i^t) ; Other default parameters.

Output: Offloading decision o_i^t ; average latency $D_{latency}$, average energy consumption E_{total} ;

- 1: Random initializing DNN parameters $\theta(t)$, \mathbf{W} , Φ ;
- 2: **Joint optimization \mathbf{W}^{t+1} and Φ^{t+1} with fixed \mathbf{D}_i^t , $\mathbf{E}_{total,i}^t$** ; solve the problem by utilizing (22)-(26)
- 3: **Joint optimization \mathbf{D}_i^{t+1} and $\mathbf{E}_{total,i}^{t+1}$ with fixed $\mathbf{W}^{t+1}, \Phi^{t+1}$** ; calculate the average result by (36)
- 4: Calculate the convergence factor:

$$\chi = \frac{|obj(\mathbf{W}, \Phi, \mathbf{D}_i, \mathbf{E}_{total,i})^{t+1} - obj(\mathbf{W}, \Phi, \mathbf{D}_i, \mathbf{E}_{total,i})^t|}{obj(\mathbf{W}, \Phi, \mathbf{D}_i, \mathbf{E}_{total,i})^{t+1}}$$
- 5: **if** $\chi \geq \epsilon$ and $t \leq t_{max}$ **then**
- 6: $t = t + 1$;
- 7: Execution step 2
- 8: **else**
- 9: Calculate the average latency and energy consuming by

$$D_{latency,k}^t = \max\left\{\frac{s_k^{local}\phi}{f_{local,k}^t}, \frac{s_k^{edge}\phi}{f_{edge,k}^t} + \frac{s_k^{edge} + s_k^{cloud}}{D_{trans,k,L}^t}\right\}$$

$$I_k(t+1) = \max\left(I_k(t) + \nu E_{total,k}^t - \nu E_{thre,k}, 0\right);$$
- 10: Output $\mathbf{W}^*, \Phi^*, o_i^*, D_{latency,k}^*, I_k^*$
- 11: **end if**

C. Computational Complexity Analysis

The computational aspect of the proposed approach encompasses the training of the actor component and the optimization of the critic component. The overall computational complexity involved in wireless transmission capacity optimization can be expressed as

$$\mathcal{O}(t_b N_{BS}^2 M_{BS}^2 L^2 K^2 + t_r N_{RIS}^2 M_{RIS}^2), \quad (42)$$

where t_b denotes the optimization iteration of (25) and t_r represents the iteration of (26). As for the complexity of model training, we can describe the corresponding computation for actor part by [45]

$$\mathcal{O}\left(J_0 J_L + \sum_{l=1}^{L-1} J_l J_{l+1}\right), \quad (43)$$

where L denote the model layers and J_0, J_L represents the input and output layer, respectively. Besides, J_l denotes the l -th layer of the network model. The model learning process can be simplified into basic computational units. The computational complexity of the random sampling process can be represented as $\mathcal{O}(\log_2 U)$, where U is the size of experience replay buffer. Additionally, the optimization complexity of the critic module can be denoted as [55]

$$\mathcal{O}\left(\left[K \log_2\left(\frac{\rho}{\epsilon}\right) + K^3 V\right]\right), \quad (44)$$

where ρ denotes the upper bound of the search range, and ϵ represents the absolute error between the boundaries. As

described in Algorithm 1, the training process will conduct t rounds of training until the system converges. Therefore, the overall computational complexity corresponding to the entire optimization process can be denoted as

$$\mathcal{O}\{t[(t_b N_{BS}^2 M_{BS}^2 L^2 K^2 + t_r N_{RIS}^2 M_{RIS}^2) + (J_0 J_L + \sum_{l=1}^{L-1} J_l J_{l+1}) + (K \log_2(\frac{\rho}{\epsilon}) + K^3 V) + \log_2 U]\}, \quad (45)$$

According to the discussion above regarding computational complexity, it is evident that the system optimization process consists of two main computational components: wireless capacity optimization and task offloading decision optimization based on DRL. Importantly, the computational complexity of the actor training component is primarily linear in terms of the model depth and size. This characteristic presents a significant advantage over conventional iterative optimization algorithms, as it substantially reduces the overall computational complexity and improves execution efficiency once the training is completed. However, it is important to note that since task offloading optimization depends on wireless transmission capacity, and it generally converges earlier than DRL during the iterative optimization process. Therefore, the computational complexity described in (45) corresponds to the upper bound of the overall computation complexity.

V. EXPERIMENT SETUP AND RESULTS ANALYSIS

A. Experiment Setup

To establish the representative simulation scenario, a typical RIS-aided cell-free network is adopted, as depicted in Fig. 4. In this setup, it has been assumed that three distinct BSs are deployed in separate locations, while computation servers are placed at the edge to offer abundant computing resources. For simplicity but without loss of generality, the distances between BS and users are scaled down by a factor of about 10 compared to practical distances. Moreover, a typical RIS is deployed in the system to enhance the overall throughput. Considering that in practical scenarios, user distribution is randomized, all users are randomly located within an $5m$ radius along the x-axis [17]. To reflect practical deployment scenarios, all BSs are set at a height of $3m$, while the RIS is elevated to $6m$ to ensure a higher reflection link gain. Specifically, the BSs are situated at $(20m, -30m)$, $(50m, -30m)$, and $(80m, -30m)$, whereas the RIS is positioned at $(50m, 20m)$. Although our parameters are scaled proportionally, this only affects channel loss in the signal transmission process. Other relevant parameters and the architecture remain unchanged. This approach allows us to assess all channel characteristics and system performance. Moreover, the transmitting power of the BSs is set as $P_{trans} = 10$ dBm and the noise level σ^2 is assumed as -80 dBm. Assuming independent and identically distributed wireless transmission channels, the channel path loss can be constructed by using (46).

$$P_{loss} = P_t + 10 \log_{10}\left(\frac{\lambda}{4\pi d_0}\right) - 10\gamma \log_{10}\left(\left(\frac{d}{d_0}\right)\right), \quad (46)$$

where d_0 is the reference point distance, γ is the path loss scale factor, and λ denotes the speed of light. In addition,

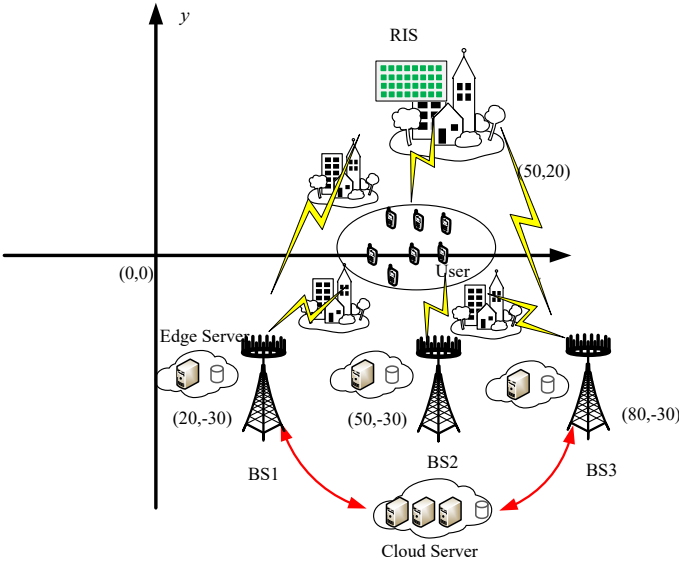


Fig. 4. The architecture of simulation scenario which contains three BSs assisted by RIS and several mobile users.

TABLE I
THE PARAMETERS LIST OF SIMULATION EXPERIMENT SETUP.

Simulation Parameters	Setup
Energy scale factor α	10^{-26} [55]
One bit computing cost ϕ	100 cycles
Energy threshold E_{thre}	0.8W
Computation duration T	1s
Offloading duration τ	$[0, 1]$
Bandwidth utilization v_u	83.33%
Data arrival rate $Data_a^{max}$	2.5Mbps
Cloud computing ability f_{cloud}^{max}	3000GHz
Edge computing ability f_{edge}^{max}	300GHz
Local computing ability f_{local}^{max}	300MHz
Working bandwidth B_k	4 MHz
Number of subcarriers	4
Number of antennas at user M_{user}	2
Number of antennas at BS M_{BSs}	4
RIS elements M_{RIS}	128
Transmitter power P_{trans}	10dBm
Noise level σ^2	-80dBm

we deploy RIS to further improve the system capacity of the cell-free network. The path loss of communication links between BSs and RIS or RIS and users also follow principles described in (46). To provide a more comprehensive evaluation of wireless transmission system performance, the Rician distribution is utilized to characterize more representative wireless transmission channel. Each transmission path is composed of

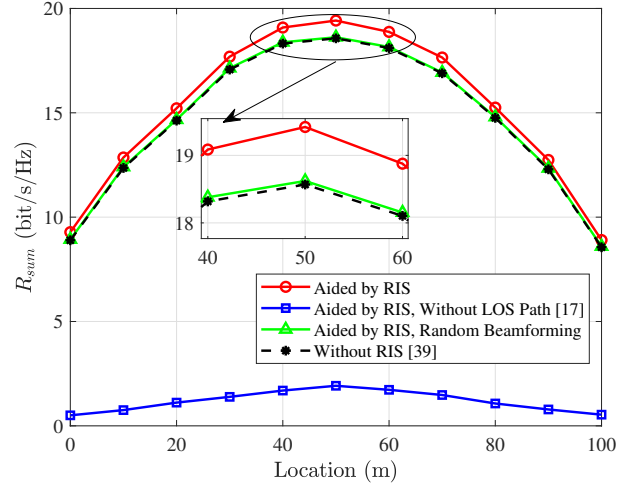


Fig. 5. System total capacity R_{Total} via different user location.

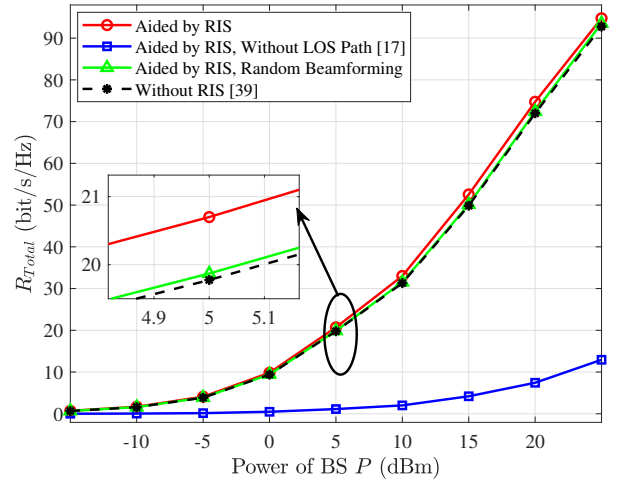


Fig. 6. The influence of base station transmission power P among system capacity.

two different paths: line-of-sight (LOS) and non-line-of-sight (NLOS). Furthermore, the transmission link between BS and user can be denoted by

$$\mathbf{H} = \sqrt{\frac{U}{U+1}} \mathbf{H}_{Los} + \sqrt{\frac{1}{U+1}} \mathbf{H}_{Nlos}, \quad (47)$$

where \mathbf{H}_{Los} represents the corresponding LOS transmission link. Besides, \mathbf{H}_{Nlos} denotes the dynamic NLOS channel. The channel information for the other transmission links is similar to the channel model described above. Recognizing that NLOS transmission paths may exist in practical application, we also conducted comparative analyses of this scenario in subsequent experiments.

During the experimental phase, we set the energy threshold for each user at 0.8W, representing the maximum allowable

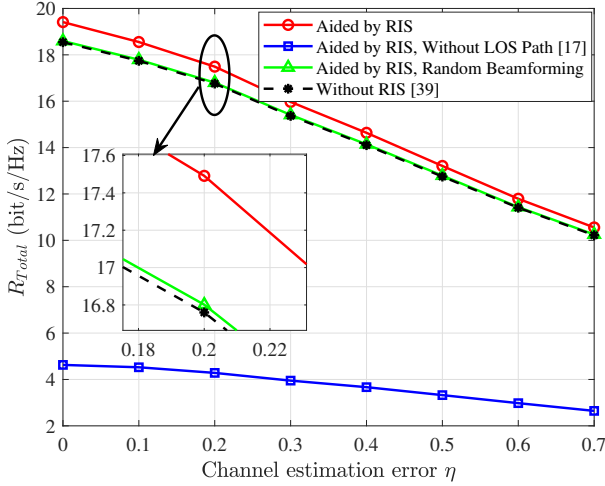


Fig. 7. The influence of channel estimation error among system capacity.

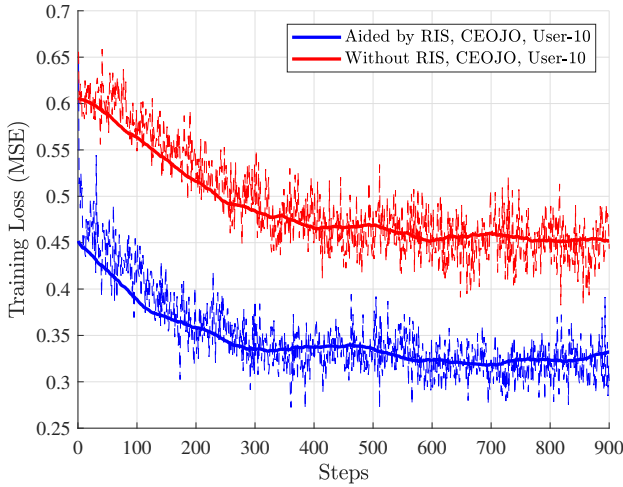


Fig. 8. The training cost via different learning steps.

task computation energy consumption for an individual user, which should not exceed this specified limit. Additionally, the RIS comprises a total of 128 elements, providing additional channel transmission degrees of freedom to enhance the wireless transmission capacity of the system, consequently leading to reductions in system latency and energy consumption. For the sake of simplifying the system calculation process, we assume that the total duration, denoted as T for task offloading is fixed at 1 second. Furthermore, the quantity of computational tasks arriving at the terminal side is represented as $Data_a^{max}$. To precisely characterize the computational capabilities of the cloud, edge, and terminal components, we employ f_{cloud}^{max} , f_{edge}^{max} and f_{local}^{max} to denote the computing capacity at various layers within the system. Indeed, it is crucial to highlight that our assumption considers a single bit of computation at the terminal side consuming 100

cycles [55]. Nevertheless, in real-world systems, this parameter can experience dynamic variations due to shifts in terminal specifications and computing capabilities. It is important to underscore, however, that these variations do not affect the implementation of our comprehensive optimization scheme. For the dynamic conditions, different computational capabilities can be uniformly converted into corresponding computing capacities C_k^t . This adaptability allows us to account for real-world fluctuations and complexities in our assessments. Moreover, the offloading decision for each user can be obtained through the joint optimization of the RIS weights and the offloading metrics. Furthermore, evaluating the performance of system transmission capacity and offload strategy for different user scale (10, 20). All relevant default parameters involved in the simulation experiment are shown in Table I. To determine the offloading decision, a multi-layer perceptron network is employed, and a novel preprocessing approach is implemented to eliminate redundancy. The purpose of this paper is to thoroughly evaluate the effectiveness and robustness of the joint optimization algorithm, and thus the experiment is focused on two primary aspects of research and analysis.

- In a cell-free architecture, the phase shift weights of base stations and RIS are jointly optimized using (9). Furthermore, the impact of varying BS transmit power levels and CSI estimation errors on the overall system transmission capacity is thoroughly investigated.
- The experiments conducted involve a comprehensive comparative analysis of the system performance under various conditions. These conditions include different user scales, RIS deployments, and offloading schemes. Additionally, the impact of different computation rate arrival thresholds on the average system latency and energy consumption is examined. This analysis provides valuable insights into the behavior of the system and allows for the evaluation of its performance in different scenarios.

B. Properties of proposed DRL offloading scheme for RIS-aided MEC

The system capacity of four different joint beamforming schemes is compared by observing the movement of all users from position (0,0) to (100,0), as depicted in Fig. 4. Indeed, the RIS-aided cell-free network demonstrates superior performance in all positions. This is attributed to the passive RIS's capability to alter the transmission path by adjusting the phase of RIS elements, thereby effectively transforming the channel characteristics. By optimizing the phase shifts, the RIS can enhance the SNR at the receivers end, leading to substantial improvements in the total system capacity. It can be found from the Fig. 5 that the highest system capacity is achieved when the user cluster is located at (50,0). This can be attributed to the fact that at this location, the users are relatively closer to both the BS and RIS, resulting in lower channel path loss. A typical modeling method named without RIS scheme is designed for scenarios lacking a direct line-of-sight (LOS) transmission path. This method serves as a basis for comparing and analyzing the advantages conferred

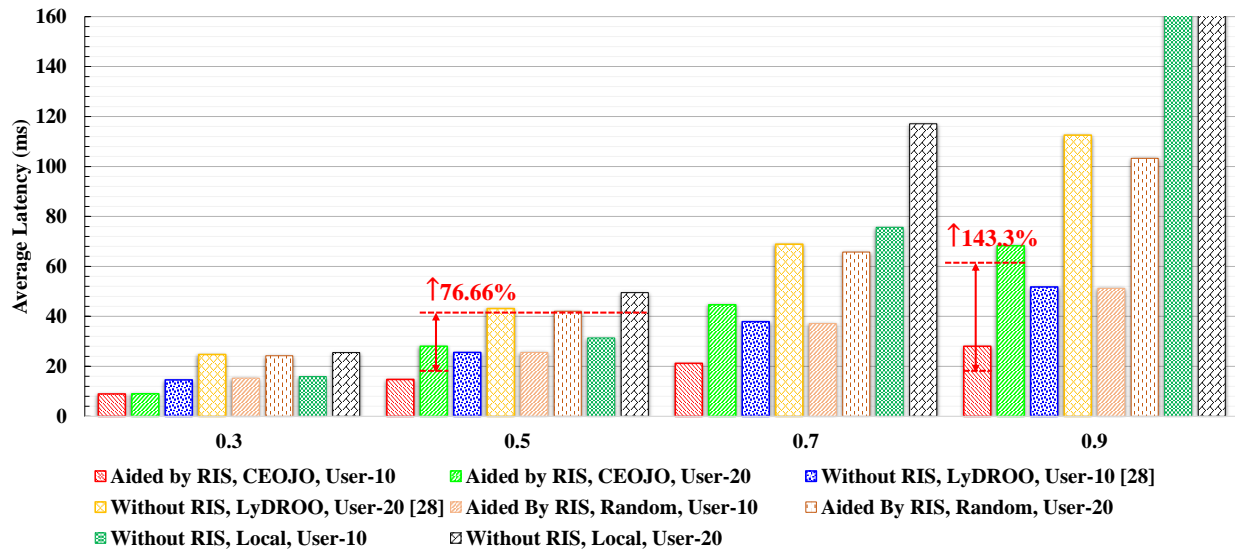


Fig. 9. Performance analysis of latency with different users and data arrival rate

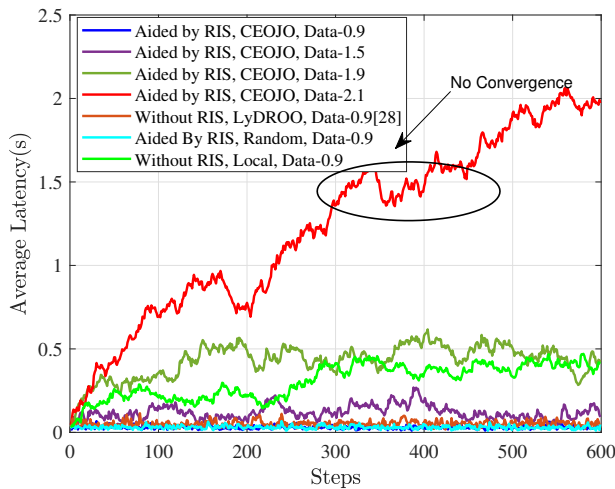


Fig. 10. Latency convergence analysis among different schemes.

by RIS in NLOS scenarios. The random beamforming at the RIS side, focusing solely on precoding optimization at the BS side. This comparison aims to assess the benefits of RIS-side optimization versus random beamforming. However, the random phase scheme and the without RIS scheme cannot effectively improve the capacity due to the absence of reflection path. Similarly, the without direct link scheme exhibits poor performance due to the low SNR at the receiver side. In other experiments, the position center of users is located at (50, 0).

The transmitting power of the BS is a crucial factor that affects the system capacity. However, it can also generate interference between users, thereby limiting the capacity improvement. Fig. 6 shows that as the transmission power of the BS increases, all optimization schemes brings an improvement in

the system throughput. To provide a more objective evaluation of the system's practical application, we assume a transmitting power of 10 dBm for the BS. While perfect CSI acquisition is often assumed in theoretical scenarios, it is important to consider the impact of channel noise and estimation errors that occur in practical applications. Fig. 7 demonstrates that the system capacity significantly declines as the CSI error increases. However, the capacity loss remains acceptable as long as the CSI error is below 10%. In our experiments, to explore the upper bound of computation offloading, we assume perfect CSI, simplifying the analysis without sacrificing generality.

The training loss of the DRL model during the learning process is illustrated in Fig. 8. The results demonstrate a rapid decrease in the system loss function as the number of training iterations increases, eventually converging to a stable state. Comparing the two curves in the chart, it is evident that the system loss function is lower when utilizing RIS assistance compared to without RIS assistance. This can be attributed to the improved transmission performance of the system with RIS assistance, which facilitates faster learning and convergence to the global optimal solution.

As depicted in Fig. 9, the average latency versus the computation arrival rate under different offloading schemes and user scale. The local mode is to put all computation tasks on the local server for computing without edge and cloud offloading. It is obvious that, the RIS-aided cloud-edge offloading scheme has a better performance in latency. Besides, the "LyDROO" scheme, as described in [55], represents an offloading optimization approach based on Lyapunov principles, operating without any assistance from RIS. This algorithm employs Lyapunov optimization and RL to achieve offloading optimization which does not consider cloud offloading and RIS-assisted wireless communication links. Additionally, two other benchmark algorithms are included: random offloading and local computing. Towards the random offloading, the offloading decisions are randomly generated. On the other

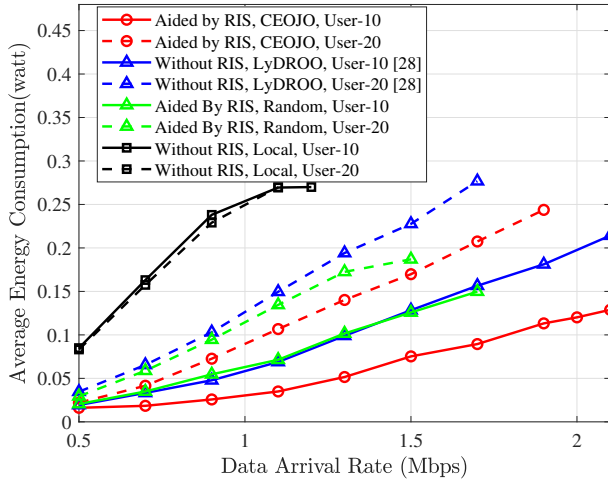


Fig. 11. The average energy consumption of different schemes via data arrival rate.

hand, local computing refers to any non-affordable computation tasks, where all tasks are processed locally. This represents a typical computing task pattern and is therefore considered the most representative baseline for comparison. To comprehensively evaluate the merits and demerits of various offloading optimization schemes, we constructed the core framework for computing offloading task optimization using Python. Simultaneously, we invoked certain DLL functions to perform the optimization for RIS. Especially, when arrival data is 0.9 the without-RIS scheme brings about 143.3% additional latency. Compared to local mode when arrival rate is 0.5, there is about 76.66% additional latency. It is obvious that the average latency escalates with the growth in the number of users, primarily due to the restricted computing resources that hinder the system from allocating an adequate amount of computing resources to all users. This trend becomes more pronounced as the quantity of user computing increases. The results demonstrate that the proposed CEOJO scheme can efficiently reduce the average latency.

As illustrated in Fig. 11, the relationship between the average energy consumption and computation arrival rate for different offloading schemes are presented. The chart clearly demonstrates that the average energy consumption increases with the number of users for all offloading strategies. However, the RIS-aided scheme can significantly reduce the average energy consumption across different user scales. Furthermore, it can be observed that although there is an intelligent decision-making process based on DRL for offloading systems without RIS, the decrease in system transmission capacity has resulted in increased transmission power consumption. On the other hand, for random offloading schemes with RIS assistance, despite the randomness in selective offloading, the performance is still favorable due to the transmission gain provided by the RIS assistance. Additionally, the energy consumption in the local computing mode remains unaffected by the number of users since there is no offloading involved

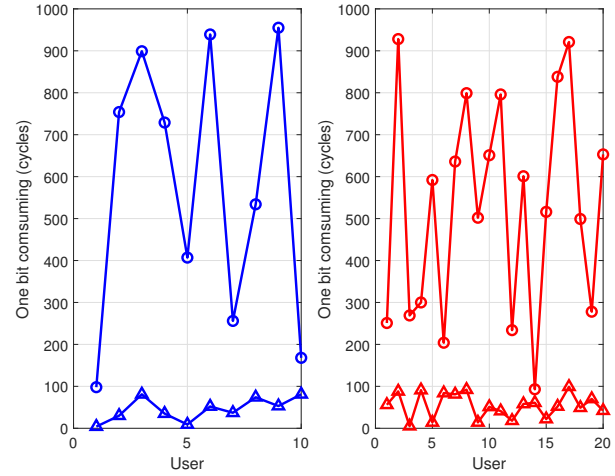


Fig. 12. The different one bit computing cost configuration for users.

in this process. As shown in Fig. 10, the average system latency is observed to converge and stabilize rapidly when subjected to a particular computation data arrival rate. It is worth noting that, as the arrival rate increases, the average system latency also increases. Enhanced transmission links are expected to reduce the average system latency. However, when the arrival rate exceeds a certain threshold, the system's latency becomes unstable and increases with each optimization step. This instability can be attributed to the fact that the arrival rate exceeds the maximum processing capacity of the cloud-edge servers, preventing the system from finding the optimal offloading decision. Consequently, for any given cloud-edge offloading system, there exists a threshold for the number of computing offloading tasks that the system can process. Beyond this threshold, the system cannot support additional users and computing tasks, leading to increased latency and potential performance degradation.

In order to comprehensively evaluate the system under different one-bit computing costs ϕ arising from various parameters such as different operating systems, computational capabilities, and computation offloading task decomposition schemes in a heterogeneous network, we conducted a comprehensive comparative analysis of the proposed algorithm under varying computational capabilities. As shown in Fig. 12, we employed two comparative scenarios. The first scenario involved randomly generating user one-bit computing costs parameters within the range of [0, 1000], while the second scenario entailed generating within the range of 0-100. The left figure depicts the random uniform distribution results between 0-100. From the distribution results in the Fig. 12, it is evident that the average one-bit computing costs are approximately distributed around 500 for the range of [0, 1000] and around 50 for the range of [0, 100].

As depicted in Fig. 13(a),(b)and (c), the statistical results illustrate the average energy consumption under different configurations. It is evident that, when one bit computation

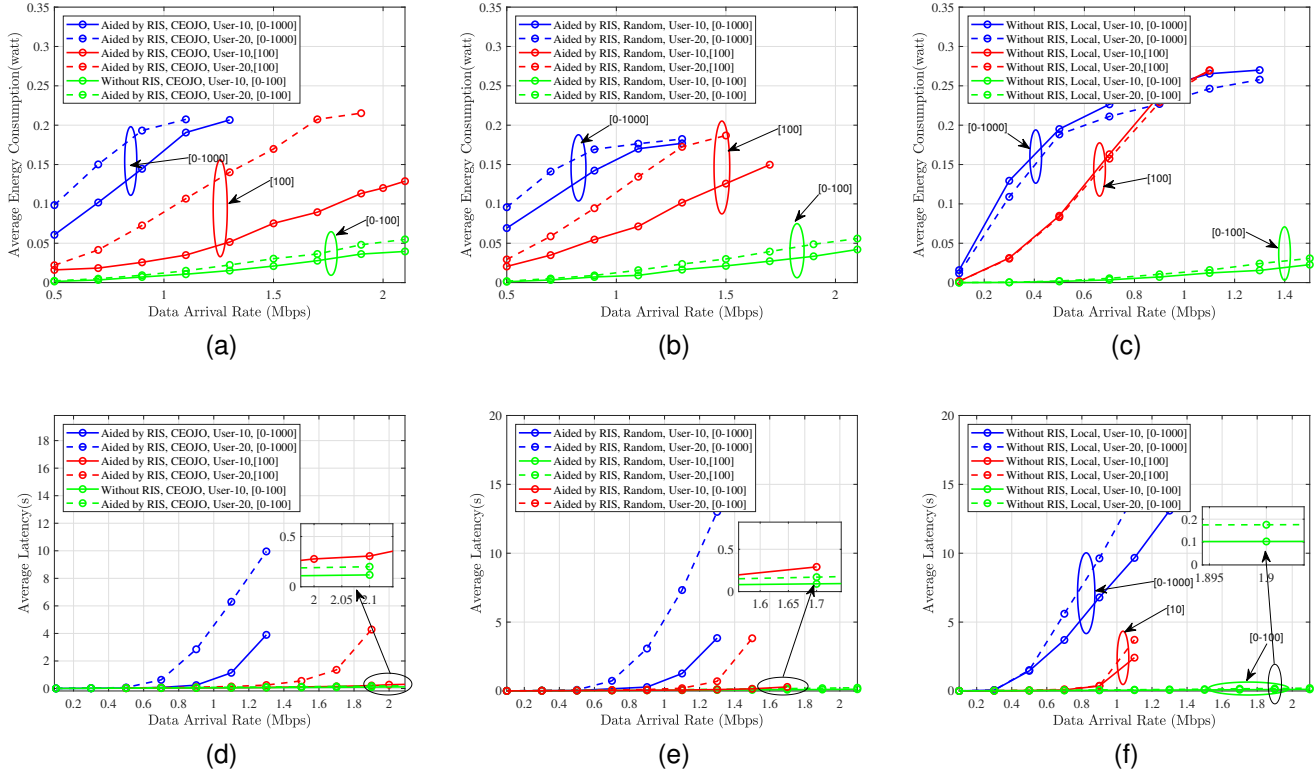


Fig. 13. The average energy consumption and latency via different schemes. (a) average energy consumption, CEOJO. (b) average energy consumption, Random. (c) average energy consumption, Local. (d) average latency, CEOJO. (e) average latency, Random. (f) average latency, Local.

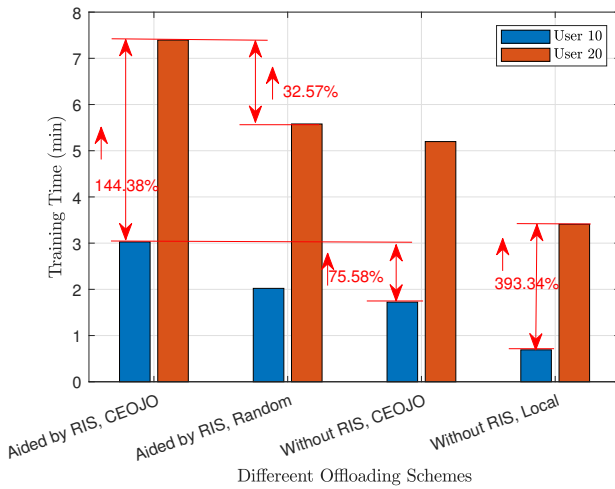


Fig. 14. Computational complexity analysis among different offloading schemes.

cost is randomly distributed within the range of $[0, 100]$, the CEOJO, Random, and Local task computation methods exhibit significant reductions in average power consumption compared to the fixed 100 cycles setting. Additionally, the differences between different user scales are also diminished. On the contrary, when user one-bit computing costs are distributed

TABLE II
THE AVERAGE DECISION-MAKING LATENCY VIA DIFFERENT STAGE.

Latency (ms) \ Schemes	Stage	RIS/BS	Offloading
		Optimization	Optimization
		User(10/20)	User(10/20)
Aided by RIS, CEOJO		202.1/ 557.9	100.5/ 181.6
Without RIS, CEOJO		69.1/ 340.9	103.3/ 178.9
Aided by RIS, Random		205.3/ 547.4	0/ 0

in the range of $[0, 1000]$, there is a substantial increase in average power consumption under the same computation task arrival flow. It is notable that the increase in average power consumption is most pronounced under the local computation mode. This is primarily attributed to the fact that all tasks are locally computed, and inadequate computational capabilities lead to additional power consumption.

As illustrated in Fig. 13(d), (e), and (f), the diagram illustrate the corresponding equivalent average computation latency under different computation task arrival flows. It is obvious that under various offloading schemes, as user one-bit computing costs ϕ increase, local computation capabilities gradually decrease. Consequently, within the same time frame, the device cannot process more computation tasks, leading to an increase in equivalent processing latency. In summary, different configurations of ϕ reflect the diverse computa-

tional capabilities. The higher ϕ signify poorer equivalent computational capabilities. In scenarios with relatively small computation task arrival flows, various offloading schemes exhibit minimal differences in average energy consumption and latency. However, with the increase of computation task flow, the performance differences between different offloading schemes and user scales gradually become more pronounced.

To comprehensively evaluate the effectiveness of the proposed scheme, simulation and comparative analysis on the computational complexity of different offloading schemes under different application scenarios are conducted. As shown in Fig. 14, as the user scale increases, the computational complexity of all involved schemes increases regardless of whether RIS is used. For fair comparison of the training time needed for different computation task offloads, the optimization of BS also is included within the local computing mode. Each completed training process comprises 600 training iterations, with each iteration involving both RIS and BS weight optimization and offloading decision optimization processes. It can be seen from the chart that computational complexity of CEOJO under the assistance of RIS has increased by approximately 144.38% when the user scale increase from 10 to 20. Meanwhile, the computational complexity of the local computing mode without RIS assistance has increased by approximately 393.34%. The computational complexity of this offloading mode mainly comes from the precoding optimization on the BS side, which brings the nonlinearity increase due to the increase in user scale. When the user scale is 20, the CEOJO offloading mode increases the complexity by approximately 32.57% compared to the random offloading mode, mainly due to the additional computational complexity by DRL and related computing modules. When the user scale is 10, under the CEOJO offloading mode, the scheme with RIS assistance increases the computational complexity by approximately 75.58% compared to the method without RIS assistance. The main reason for this is that in the case of RIS assistance, joint optimization with the beamforming on the BS side is required, which increases the corresponding computational complexity and iteration times, resulting in additional computational complexity. However, the random offloading strategy assisted by RIS consumes more computational complexity than the same strategy without RIS, due to the beamforming optimization on the RIS side is more complex.

To obtain a more detailed understanding of the time required for each training or online decision-making, we conducted a statistical analysis of two different computation processes. As shown in Table II, the latency here specifically refers to the decision-making time for different optimization stage. It can be found from the table that each optimization computation is predominantly occupied by the RIS optimization, while the computing offloading optimization requires approximately half of the latency. As the number of users increases, the computational complexity exhibits a nonlinear increase. During the training process mentioned above, the RL component, involving actions related to offloading strategy also encompasses the critic part. According to testing log, executing the actor network alone, without utilizing the critic section for fine-tuning improvements, requires approximately 10% of the

time cost for the training and learning stage with the same computing resources. All the aforementioned computational processes were run and statistically derived on a Supermicro server with 96 computing cores and 8 RTX 3090 GPUs. In summary, the involved RIS in wireless communication transmission has brought benefits in terms of computational offloading, average latency, and energy consumption, but at the same time, it also brings additional computational complexity.

VI. CONCLUSIONS

This paper is dedicated to developing an efficient cloud-edge offloading framework empowered by utilizing DRL to improve the efficiency and robustness of offloading decision. To achieve this, we employ sophisticated Lagrangian iterative algorithms to optimize the phase shift of both BS and RISs simultaneously. Moreover, we propose an efficient DRL framework based on the actor-critic method to further improve the efficiency and robustness of computation task offloading. Through rigorous experimentation, the significant benefits of involving RIS in the MEC offloading system has been evaluated. The extensive experimental results provide compelling evidence that our proposed RIS-aided cloud-edge offloading framework significantly reduces average latency by approximately 76.66% and 143.3% at arrival rates of 0.5 and 0.9, respectively. Additionally, the average energy consumption and system robustness are thoroughly examined and discussed. Once within the system's capabilities, all computational offloading tasks can be optimized significantly. Moreover, our approach is driven by a well-defined model and can be easily replicated. Our future research will concentrate on designing an efficient federated learning framework for heterogeneous resource allocation.

APPENDIX A PROOF OF FUNCTION (23)

The communication system capacity optimization problem mainly relays on signal-to-noise ratio. As denoted in (A.1), it is a weighted system transmission capacity optimization problem

$$\underset{\Phi, \mathbf{W}}{\text{maximize}} \sum_{m=1}^M w_m \log \left(1 + \frac{\Upsilon_m(\Phi, \mathbf{W})}{\Gamma_m(\Phi, \mathbf{W}) + n_{k,l}} \right) \quad (\text{A.1a})$$

$$\text{subject to } \Phi, \mathbf{W} \in \mathcal{S} \quad (\text{A.1b})$$

where w_m is the weight factors and $n_{k,l}$ denotes the system noise. Besides, Υ_m and Γ_m represents the corresponding terms of SINR. Furthermore, we can simplify it as (A.2) by involving Lagrangian dual transform.

$$\begin{aligned} g(\Phi, \mathbf{W}, \delta) = & \sum_{k=1}^K \sum_{l=1}^L \beta_k \ln(1 + \delta_{k,l}) - \sum_{k=1}^K \sum_{l=1}^L \beta_k \delta_{k,l} \\ & + \sum_{k=1}^K \sum_{l=1}^L \beta_k (1 + \delta_{k,l}) g_{k,l}(\Phi, \mathbf{W}), \end{aligned} \quad (\text{A.2})$$

where $\delta_{k,l}$ denotes the auxiliary variable. Besides, $g_{k,l}(\Phi, \mathbf{W})$ is defined by

$$g_{k,l}(\Phi, \mathbf{W}) = \frac{\mathbf{w}_{l,k}^H \mathbf{h}_{k,l} \mathbf{h}_{k,l}^H \mathbf{w}_{l,k}}{\sum_{j=1}^K \mathbf{h}_{k,l}^H \mathbf{w}_{l,j} \left(\mathbf{h}_{k,l}^H \mathbf{w}_{l,j} \right)^H} + n_{k,l} \quad (\text{A.3})$$

$$= \frac{\mathcal{R}_m(\Phi, \mathbf{W})}{\mathcal{R}_m(\Phi, \mathbf{W}) + \Gamma_m(\Phi, \mathbf{W}) + n_{k,l}},$$

Furthermore, it can be found from (23) that $g_{k,l}$ is a concave differentiable function towards variable δ when the other variables (Φ, \mathbf{W}) are fixed.

$$\frac{\partial g(\Phi, \mathbf{W}, \delta)}{\partial \delta_{k,l}} = \sum_{k=1}^K \sum_{l=1}^L \beta_k \left(\frac{1}{1 + \delta_{k,l}} - 1 + \frac{\mathcal{R}_m(\Phi, \mathbf{W})}{\mathcal{R}_m(\Phi, \mathbf{W}) + \Gamma_m(\Phi, \mathbf{W}) + n_{k,l}} \right) \quad (\text{A.4})$$

when $\frac{\partial g(\Phi, \mathbf{W}, \delta)}{\partial \delta_{k,l}} = 0$, we can acquire the optimized solution $\delta_{k,l}^*$ equivalently.

$$\delta_{k,l}^* = \frac{\mathcal{R}_m(\Phi, \mathbf{W})}{\Gamma_m(\Phi, \mathbf{W}) + n_{k,l}} \quad (\text{A.5})$$

REFERENCES

- [1] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Enabling massive iot toward 6g: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11891–11915, 2021.
- [2] D. Xu, K. Yu, and J. A. Ritcey, "Cross-layer device authentication with quantum encryption for 5g enabled iiot in industry 4.0," *IEEE Trans. Ind. Inf.*, vol. 18, no. 9, pp. 6368–6378, 2022.
- [3] D. Xu and P. Ren, "Quantum learning based nonrandom superimposed coding for secure wireless access in 5g urllc," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2429–2444, 2021.
- [4] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6g: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, 2022.
- [5] L. Song, X. Hu, G. Zhang, P. Spachos, K. N. Plataniotis, and H. Wu, "Networking systems of ai: On the convergence of computing and communications," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20352–20381, 2022.
- [6] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for ai-enabled wireless networks: A tutorial," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 2, pp. 1226–1252, 2021.
- [7] T. Zhou, H. Zhang, B. Ai, C. Xue, and L. Liu, "Deep-learning-based spatial-temporal channel prediction for smart high-speed railway communication networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5333–5345, 2022.
- [8] K. Wang, P. Xu, C.-M. Chen, S. Kumari, M. Shojafar, and M. Alazab, "Neural architecture search for robust networks in 6g-enabled massive iot domain," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5332–5339, 2021.
- [9] D. C. Nguyen, P. Cheng, M. Ding, D. Lopez-Perez, P. N. Pathirana, J. Li, A. Seneviratne, Y. Li, and H. V. Poor, "Enabling ai in future wireless networks: A data life cycle perspective," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 1, pp. 553–595, 2021.
- [10] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [11] B. Liu, C. Liu, and M. Peng, "Resource allocation for energy-efficient mec in noma-enabled massive iot networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1015–1027, 2021.
- [12] A. Gao, Q. Wang, W. Liang, and Z. Ding, "Game combined multi-agent reinforcement learning approach for uav assisted offloading," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 12888–12901, 2021.
- [13] G. Cui, X. Li, L. Xu, and W. Wang, "Latency and energy optimization for mec enhanced sat-iot networks," *IEEE Access*, vol. 8, pp. 55915–55926, 2020.
- [14] J. Yun, Y. Goh, W. Yoo, and J.-M. Chung, "5g multi-rat urllc and embb dynamic task offloading with mec resource allocation using distributed deep reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20733–20749, 2022.
- [15] Z. Zhang, N. Wang, H. Wu, C. Tang, and R. Li, "Mr-dro: A fast and efficient task offloading algorithm in heterogeneous edge/cloud computing environments," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3165–3178, 2023.
- [16] L. Xu, M. Qin, Q. Yang, and K.-S. Kwak, "Learning-aided dynamic access control in mec-enabled green iot networks: A convolutional reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2098–2109, 2022.
- [17] Z. Zhang and L. Dai, "A joint precoding framework for wideband reconfigurable intelligent surface-aided cell-free network," *IEEE Trans. Signal Process.*, vol. 69, pp. 4085–4101, 2021.
- [18] Z. Chu, P. Xiao, M. Shojafar, D. Mi, J. Mao, and W. Hao, "Intelligent reflecting surface assisted mobile edge computing for internet of things," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 619–623, 2021.
- [19] C. Wang, X. Chen, J. An, Z. Xiong, C. Xing, N. Zhao, and D. Niyato, "Covert communication assisted by uav-irs," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 357–369, 2023.
- [20] J. Wang, W. Tang, Y. Han, S. Jin, X. Li, C.-K. Wen, Q. Cheng, and T. J. Cui, "Interplay between rin and ai in wireless communications: Fundamentals, architectures, applications, and open research problems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2271–2288, 2021.
- [21] X. Pang, N. Zhao, J. Tang, C. Wu, D. Niyato, and K.-K. Wong, "Irs-assisted secure uav transmission via joint trajectory and beamforming design," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 1140–1152, 2022.
- [22] S. Zhang and R. Zhang, "Intelligent reflecting surface aided multi-user communication: Capacity region and deployment strategy," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5790–5806, 2021.
- [23] E. Basar, I. Yildirim, and F. Kilinc, "Indoor and outdoor physical channel modeling and efficient positioning for reconfigurable intelligent surfaces in mmwave bands," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8600–8611, 2021.
- [24] L. Jin, X. Xu, S. Han, X. Chi, S. Lv, P. Zhang, and C. Liu, "Computation offloading outage probability analysis and min-max fairness optimization in ris-assisted mec system," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 4615–4627, 2023.
- [25] T. Bai, C. Pan, Y. Deng, M. Elakashan, A. Nallanathan, and L. Hanzo, "Latency minimization for intelligent reflecting surface aided mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2666–2682, 2020.
- [26] X. Hu, C. Masouros, and K.-K. Wong, "Reconfigurable intelligent surface aided mobile edge computing: From optimization-based to location-only learning-based solutions," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3709–3725, 2021.
- [27] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mob. Comput.*, vol. 19, no. 11, pp. 2581–2593, 2020.
- [28] S. Bi, L. Huang, H. Wang, and Y.-J. A. Zhang, "Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7519–7537, 2021.
- [29] S. Mao, L. Liu, N. Zhang, M. Dong, J. Zhao, J. Wu, and V. C. M. Leung, "Reconfigurable intelligent surface-assisted secure mobile edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6647–6660, 2022.
- [30] J. Yan, S. Bi, and Y. J. A. Zhang, "Offloading and resource allocation with general task graph in mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5404–5419, 2020.
- [31] Y. Zuo, S. Jin, and S. Zhang, "Computation offloading in untrusted mec-aided mobile blockchain iot systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8333–8347, 2021.
- [32] Q. Zhang, H. Wen, Y. Liu, S. Chang, and Z. Han, "Federated-reinforcement-learning-enabled joint communication, sensing, and computing resources allocation in connected automated vehicles networks," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 23 224–23 240, 2022.
- [33] W. Mei and R. Zhang, "Performance analysis and user association optimization for wireless network aided by multiple intelligent reflecting surfaces," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6296–6312, 2021.
- [34] H. Zhang, B. Di, L. Song, and Z. Han, "Reconfigurable intelligent surfaces assisted communications with limited phase shifts: How many phase shifts are enough?" *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4498–4502, 2020.

- [35] Y. Wang, W. Zhang, Y. Chen, C.-X. Wang, and J. Sun, "Novel multiple ris-assisted communications for 6g networks," *IEEE Commun. Lett.*, vol. 26, no. 6, pp. 1413–1417, 2022.
- [36] C. Meng, K. Xiong, W. Chen, B. Gao, P. Fan, and K. B. Letaief, "Sum-rate maximization in star-ris assisted rsma networks: A ppo-based algorithm," *IEEE Internet Things J.*, pp. 1–1, 2023.
- [37] P. Chen, H. Liu, Y. Ye, L. Yang, K. J. Kim, and T. A. Tsiftsis, "Rate-splitting multiple access aided mobile edge computing with randomly deployed users," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 5, pp. 1549–1565, 2023.
- [38] H. Liu, Y. Ye, Z. Bai, K. J. Kim, and T. A. Tsiftsis, "Rate splitting multiple access aided mobile edge computing in cognitive radio networks," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2022, pp. 598–603.
- [39] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2687–2700, 2022.
- [40] S. Mao, L. Liu, N. Zhang, M. Dong, J. Zhao, J. Wu, and V. C. M. Leung, "Reconfigurable intelligent surface-assisted secure mobile edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6647–6660, 2022.
- [41] S. Huang, S. Wang, R. Wang, M. Wen, and K. Huang, "Reconfigurable intelligent surface assisted mobile edge computing with heterogeneous learning tasks," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 7, no. 2, pp. 369–382, 2021.
- [42] H. Li, M. Liu, B. Gao, K. Xiong, P. Fan, and K. B. Letaief, "Sum computation rate maximization in self-sustainable ris-assisted mec," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–2.
- [43] K. Shen and W. Yu, "Fractional programming for communication systems—part ii: Uplink scheduling via matching," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2631–2644, 2018.
- [44] K. Shen and W. Yu, "Fractional programming for communication systems—part i: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, 2018.
- [45] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 375–388, 2021. [Online]. Available: <https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=9206080&ref=>
- [46] Z. Zhang, L. Dai, X. Chen, C. Liu, F. Yang, R. Schober, and H. V. Poor, "Active ris vs. passive ris: Which will prevail in 6g?" *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1707–1725, 2023.
- [47] A. Chauhan, S. Ghosh, and A. Jaiswal, "Ris partition-assisted non-orthogonal multiple access (noma) and quadrature-noma with imperfect sic," *IEEE Trans. Wireless Commun.*, vol. 22, no. 7, pp. 4371–4386, 2023.
- [48] R. Zhong, Y. Liu, X. Mu, Y. Chen, and L. Song, "Ai empowered ris-assisted noma networks: Deep learning or reinforcement learning?" *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 182–196, 2022.
- [49] Y. Yang, Y. Hu, and M. C. Gursoy, "Energy efficiency of ris-assisted noma-based mec networks in the finite blocklength regime," *IEEE Trans. Commun.*, pp. 1–1, 2023.
- [50] A. Mishra, Y. Mao, O. Dizdar, and B. Clerckx, "Rate-splitting multiple access for 6g—part i: Principles, applications and future works," *IEEE Commun. Lett.*, vol. 26, no. 10, pp. 2232–2236, 2022.
- [51] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [52] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, 2018.
- [53] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.
- [54] Y. Ding, C. Liu, X. Zhou, Z. Liu, and Z. Tang, "A code-oriented partitioning computation offloading strategy for multiple users and multiple mobile edge computing servers," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4800–4810, 2020.
- [55] S. Bi, L. Huang, H. Wang, and Y.-J. A. Zhang, "Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7519–7537, 2021.
- [56] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, 2020.