Diffusion-Driven Optimization for Mobility-Aware User Allocation in Computing Power Networks

Xiaofei Wang, *Senior Member, IEEE,* Chenxuan Hou, Chao Qiu, *Member, IEEE,*
Chenyang Wang, *Member, IEEE,* Kai Dong, *Member, IEEE,* and Tarik Taleb, *Senior Member, IEEE*

***Abstract*—Computing Power Networks (CPNs) represent an innovative, collaborative architecture that integrates resources via the communication network, optimizing resource allocation to support service demands. Due to the increased need for services powered by artificial intelligence across various domains, CPNs are increasingly required to allocate users efficiently to appropriate servers to meet the low-latency needs of service computing. However, challenges such as users' dynamic mobility, weak communication paths, and high-dimensional solution spaces persist in optimizing user allocation in CPNs. In this context, we propose a diffusion-driven optimization approach for mobility-aware user allocation. To tackle the challenge of users' dynamic mobility, we adopt a user location prediction approach incorporating the users' movement patterns to forecast future movement, called *CAMPE*. To tackle the challenge of weak communication paths, we establish the new transmission path by reconfigurable intelligent surface and enhance the quality of the communication link by adjusting the phase configurations. Moreover, faced with the challenge of high-dimensional solution spaces associated with phase adjustment and user allocation decisions, we devise an action-generation strategy based on diffusion models named *DiffUser*. This approach motivates the generation of optimal solutions even in complex and dynamic environments. Finally, we conduct extensive simulations in user location prediction and system latency optimization. Compared with other solutions, the superiority of our approach has been demonstrated.**

***Index Terms*—Computing Power Networks, mobility prediction, user allocation, diffusion model.**

I. Introduction

COMPUTING Power Networks (CPNs) [2] have become a groundbreaking collaborative architecture that integrates resources via the network, providing users with efficient and adaptable services. By dynamically constructing self-organizing networks using various communication technologies, CPNs aim to dismantle the isolation of distributed computing resources, thereby significantly boosting the efficiency of network computational resource utilization.

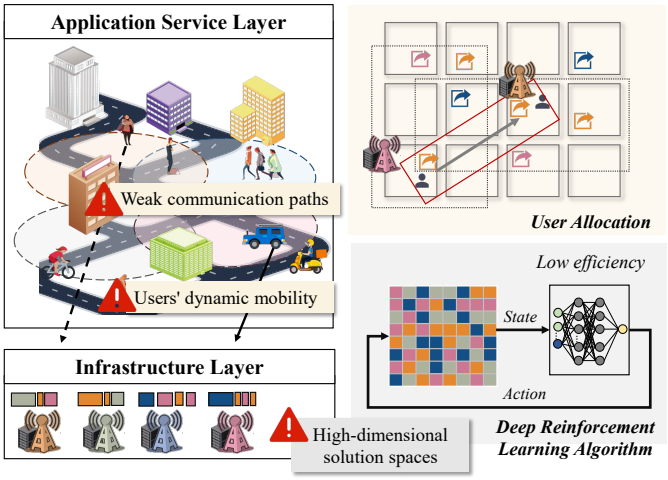


Fig. 1. User allocation: architecture, traditional solutions and challenges.

The surge in demand for mobile services, propelled by the pervasive integration of artificial intelligence (AI) across diverse industries [3], has positioned CPNs at the forefront of enabling applications characterized by high mobility and sensitivity to latency. These applications, necessitating rapid data processing and minimal latency, are serviced by servers dispersed over multiple regions. Within such frameworks, CPNs are indispensable for orchestrating an efficient user-to-server allocation mechanism [4].

Central to the functioning of CPNs is the ability to link users with the most appropriate servers, taking into account the specific characteristics and requirements of their services, known as the User Allocation (UA) problem [5]. This matching of resources with user services is essential for maximizing the network's performance.

In practical scenarios, servers within CPNs exhibit varying processing capabilities depending on the type of services [6]. However, an overreliance on matching services to servers based on their specific strengths can lead to inefficient resource utilization and potential overloads, increasing processing times. Moreover, user mobility can modify the distance between users and the server, which may significantly degrade connection quality, leading to heightened transmission latency. The interrelation among these characteristics complicates the solution of effective user allocation strategies.

Therefore, these dynamic characteristics in CPNs necessi-

This work was supported in part by Beijing-Tianjin-Hebei Basic Research Cooperation Special Project, Research on Key Technologies for Efficient Crowd Intelligence Understanding and Situation Deduction in Intelligent Connected Vehicle Environments, under Grant No. F2024201070.

An earlier and abridged version of this work [1] was presented at the IEEE Global Communications Conference (GLOBECOM), 4–8 December 2023, Kuala Lumpur, Malaysia. (Corresponding author: Chao Qiu)

X. Wang, C. Hou and C. Qiu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300354, China (e-mail: xiaofeiwang@tju.edu.cn; chenxuanhou@tju.edu.cn; chao.qiu@tju.edu.cn).

C. Wang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518061, China; and the Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen 518000, China (e-mail: chenyangwang@ieee.org).

K. Dong is with the Center for Wireless Communications, University of Oulu, 90570 Oulu, Finland (e-mail: kai.dong@oulu.fi).

T. Taleb is with the Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, 44801 Bochum, Germany (e-mail: tarik.taleb@rub.de).

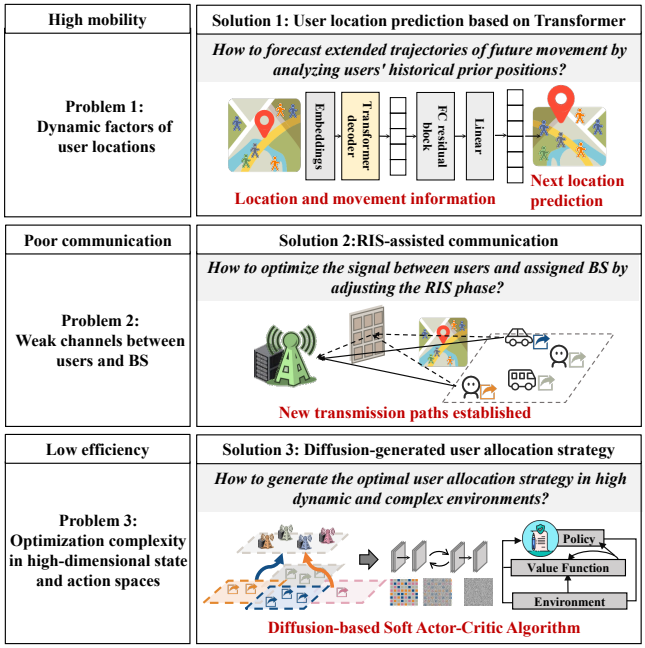


Fig. 2. The issue and solution of user allocation.

tate a joint optimization approach to overcome the challenges, summarized as **users' dynamic mobility**, **weak communication paths**, and **high-dimensional solution spaces** in Fig. 1.

- **User dynamic movement requires adaptive prediction solutions.** A typical environment for user allocation is highly dynamic, with user locations frequently changing due to mobility. The users' trajectories, influenced by various factors such as time of day and movement modes, present significant challenges in accurately predicting user locations. This requires incorporating a more granular understanding of mobility patterns.
- **Weak channels lead to connectivity challenges.** Most existing UA methods mainly focus on the computational resource requirements of application users, such as the Central Processing Unit (CPU), memory, and storage resources, without fully considering the complexity of communication resource allocation. In CPNs, users and servers attempting to connect face challenges due to poor communication channels.
- **High-dimensional state and action spaces lower system optimization efficiency.** In multi-regional scenarios with highly dynamic changes in user locations and demands, allocation strategies must adapt to various channel conditions and user behavior needs. The system faces optimization difficulties in handling environments characterized by high-dimensional states and action spaces, resulting in lower efficiency.

In this work, we propose a diffusion-driven optimization approach for mobility-aware user allocation by the following three contributions, as shown in Fig. 2.

- **User location prediction based on mobility pattern awareness:** We develop an approach for predicting user locations by leveraging a Transformer-based architecture integrated with *Context-Aware Mobility Pattern Embedding (CAMPE)*. By integrating user movement patterns directly into the network, the model can acquire a more profound comprehension of user behavior, thereby enhancing the precision of subsequent location predictions.
- **Reconfigurable intelligent surface assisted communication:** Our approach integrates the Reconfigurable Intelligent Surfaces (RISs) to augment the efficiency of communication resource utilization. The phase configurations of the RISs are adaptively modified to accommodate variations in the distribution of users. This modification enhances communication efficiency and provides users with additional connectivity options at specific locations.
- **Diffusion-driven action generation strategy:** We utilize the diffusion model to generate strategies for RISs phase adjustment and user allocation. Our proposed algorithm generates high-quality samples using environmental information as adjustment factors for optimization. The denoising mechanism enables the learning of complex action distributions, leading to more stable and optimal solutions under environmental uncertainty and variability.

Finally, we conduct extensive comparative simulations, and results demonstrate the superior performance of this work, increasing predicted accuracy by 1.54% without eliminating any locations with infrequent user access. Moreover, our method reduces average latency by 31.7%, 53.1%, 66.3% and 59.9% compared with PPO-L [7], SAC-MSPI [8], DOA [9] and COA [10] under 500 user allocation situations while accelerating convergence by 33.4% over SAC-MSPI and 76.7% over PPO-L. With prediction enabled, decision-making completes ahead of execution, consistently lowering latency and retaining benefits as the number of users increases.

The rest of this paper is organized as follows. Sec. II reviews the related works and Sec. III introduces the overall framework and presents the system model. Sec. IV introduces our proposed solution, which first predicts user mobility based on awareness of mobility patterns and then performs collaborative optimization using diffusion-driven strategies for RIS phase adjustment and user allocation. In Sec. V, we present the evaluation results, and the conclusion is provided in Sec. VI.

II. Related Works

In this section, we briefly review related studies that support mobility-aware user allocation in CPNs.

A. Computing Power Networks

CPNs represent a novel network architecture designed to address the emerging convergence of networking and computing. Tang *et al.* [11] introduce that CPNs are adept at satisfying the multi-tiered deployment and adaptable scheduling requirements that future 6G services will demand across computing, storage, and networking domains. Li *et al.* [12] describe a general CPNs framework and optimized the collaboration between cloud computing, edge computing, and network resources. However, these studies did not specifically focus on the impact of access device mobility on services.

It is necessary to prioritize the mobility of dispersed terminals in CPNs, facilitating integrated collaborative scheduling through existing network infrastructures. Zeng *et al.* [13] have

TABLE I
SUMMARY OF ARTICLES ADDRESSING THE CHALLENGES.

Type	Articles	Mobility Prediction	Improving Comm.	Multi -cell
CPNs	[11]–[13]	×	✓	✓
	[14]	×	×	✓
Reconstruct Comm.	[15], [16]	×	✓	×
	[17]–[19]	×	✓	✓
	[20]	✓	✓	×
User Allocation	[21]	×	×	✓
	[22]	✓	×	✓
	[23]	×	✓	×
Dynamic Scenario	[24]	×	✓	✓
	[19], [25]	×	✓	✓
	[26]–[29]	✓	×	×
	Our Work	✓	✓	✓

integrated wireless power transmission into mobile scenarios within CPNs, which is a highly efficient method for enhancing the self-sufficiency of networks. Furthermore, employing heterogeneous nodes within CPNs enables the provisioning of ubiquitous intelligent services by leveraging computational and networking resources for mobile computing tasks, as explored by Sun *et al.* [14]. Overall, while existing studies provide a foundation for understanding the architecture of CPNs, they offer limited insight into how these resources should be effectively bound to user requests. This highlights the need to further investigate user allocation mechanisms, particularly under multi-cell coverage and different computing capabilities across Base Stations (BSs).

B. User Allocation Problem

A major challenge in CPNs is determining the user server binding strategy that optimally facilitates the routing of service requests. This problem is often referred to as the UA problem, which attempts to effectively allocate server resources between users while overcoming the multifaceted limitations of resource availability and latency [5]. To further reflect real CPN scenarios, users may also lie within the overlapping coverage of multiple BSs. Such multi-cell situations imply that several BSs are simultaneously capable of serving the same user, thereby enlarging the candidate set of feasible allocation decisions and increasing the complexity of UA strategies.

In recent years, many methods have been proposed to address the UA problem with diverse optimization goals, such as minimizing system cost [30], reducing system energy consumption [31], and improving user quality of experience [32]. Cao *et al.* [31] characterize the UA problem by integrating server resources and geographic proximity, adopting heuristic strategies to enhance allocation efficiency. Liu *et al.* [33] improve UA performance by leveraging user role differentiation. In work [23], Chen *et al.* have formulated the problem as a mixed-integer linear programming problem with the objective of minimizing system cost under latency constraints.

However, these explorations predominantly presuppose static user environments, thereby overlooking the dimension of user mobility. Although the authors [24] have primarily examined the dynamic attributes of user access to the system in their research, a comprehensive investigation into the effects of user mobility on allocation issues remains notably absent.

C. User Location Prediction

Location prediction is a pivotal component for user allocation problems in CPNs. Early approaches to predicting future user locations rely primarily on Markov chains [26], which treat locations as states in a Markov model to infer transitions. However, these models struggle to analyze real mobility traces because they inherently assume that the current state depends only on a limited history of past states.

Recently, Recurrent Neural Networks (RNNs) based methods, such as Long Short-Term Memory (LSTM) networks, have significantly advanced the accuracy of location predictions over Markov-based models by capturing enhanced features of locations. For instance, Sun *et al.* [28] propose a two-stage self-attention architecture to represent long-term mobility information. Similarly, Feng *et al.* [27] apply the historical attention module with RNNs, and Wu *et al.* [29] utilize the LSTM model to refine predictive accuracy.

Despite significant advancements, many models still overlook the complete trajectory of user mobility. Advanced predictive models should account for both spatial and temporal aspects of user movement, including visits to infrequent locations. A comprehensive approach is needed to integrate various mobility patterns without oversimplifying the user trajectories.

D. Reconstructing the Communication Environment

In CPNs, communication links connect computing resources with user demands, but severe path loss in Millimeter Wave (mmWave) [34] limits resource utilization. To mitigate these challenges, the deployment of RISs [15] has emerged as a promising technique for improving energy and spectral efficiency. By adjusting the amplitude and phase of incident signals, RISs reconstruct the radio propagation environment and offer additional access paths between servers and mobile users [17]. Sun *et al.* [16] emphasize the necessity of RIS-assisted resource integration. Yang *et al.* [18] have introduced RISs as a communication resource in multi-user systems. These works generally leverage RIS for signal enhancement and latency reduction.

However, employing RIS mainly for signal enhancement underutilizes its potential in user allocation. In response, Zhuang *et al.* [19] introduce a multi-RIS collaborative framework using multi-agent reinforcement learning to jointly handle RIS configuration and user allocation across regions. Most of these studies overlook RIS's crucial role as a link within CPNs, which affects network's topology. In particular, it entails determining how to effectively leverage RIS to dynamically adjust user allocation with optimal network connectivity options.

Based on the preceding discussion, the issue of joint allocation optimization in CPNs introduces new challenges in Table I that must be addressed. Future research needs to develop an integrated approach that not only accurately predicts user location changes but also optimizes the use of communication

and intelligently schedules the user allocation. This method should aim to achieve optimal matching of users and BSs across multiple regions while ensuring low latency.

III. System Model

In this section, we formalize the system model, including the user allocation framework, the RIS-assisted signal model, the latency model, and the joint optimization objective.

A. User Allocation Framework

To mitigate the challenges posed by users' dynamic mobility, weak communication paths, and high-dimensional solution spaces, our proposed architecture integrates a four-layer structure, as delineated in Fig. 3. At the application service layer, diverse user devices (e.g., smartphones, laptops, vehicles) generate service demands. The infrastructure layer consists of edge nodes, servers, and BSs that collectively form the CPN resource pool. For simplicity, we refer to all these nodes as BSs. The signal reconfigurable layer utilizes RISs to dynamically modify the communication environment, thereby enhancing resource accessibility for users. Concurrently, the CPN scheduling layer aggregates service demands and allocates users to the infrastructure layer based on location and service-specific attributes.

For formalization, in this four-layer structure, BSs and users are denoted by sets $\mathcal{M} = \{1, 2, ..., M\}$ and $\mathcal{N} = \{1, 2, ..., N\}$, respectively, with Z representing the number of antennas at each BS. Furthermore, the RISs are represented by the set $\mathcal{U} = \{1, 2, ..., U\}$. Each user is assumed to be active and randomly generates a service request, with the service type indicated by the set $\mathcal{S} = \{1, 2, ..., S\}$. It is posited that each user generates a single service, which is then assigned to a specific server within a BS. The tuple $J_{n,s} = (D_{n,s}, Q_{n,s})$ denotes the service requirements for user n, where $D_{n,s}$ represents the service data size and $Q_{n,s}$ is the overall number of the CPU cycles to complete the service.

B. RIS-assisted Signal Model

In this paper, RIS with K reflection elements enhances communication between multi-antenna BSs and single-antenna users. Each BS covers a specific area with a coverage radius of d_{bs} and within each covered area, there is a single RIS with a coverage radius of d_r. Once the uplink between the associated BS and the requesting user becomes too weak because of the long inter distance, the BS schedules the reflection link via the corresponding RIS to enhance the signal quality to meet the performance requirements. Furthermore, we define the three-dimensional (3D) spatial coordinates of the user n during each time interval $t \in \mathcal{T} = \{1, 2, ..., T\}$ remains unchanged and is denoted as $[x_n(t), y_n(t), z_n]$. The Euclidean distance between the user n and the BS m (located at $[\hat{x}_m, \hat{y}_m, \hat{z}_m]$) is given by

$$d_{n,m}(t) = \sqrt{(x_n(t) - \hat{x}_m)^2 + (y_n(t) - \hat{y}_m)^2 + (z_n - \hat{z}_m)^2}. \quad (1)$$

At each time slot t, the narrow-band block-fading channel vector $\mathbf{h}_{m,n}(t) \in \mathbb{C}^{Z \times 1}$ between the user n and the BS m is given by [35]

TABLE II
Key Symbols Table

Symbol	Description
$\mathcal{M}$	The set of BSs
$\mathcal{N}$	The set of users
$\mathcal{U}$	The set of RISs
$\mathcal{S}$	The set of service types
$D_{n,s}$	Service data size
$Q_{n,s}$	Overall number of the CPU cycles to complete service
Z	The number of BS antennas
K	RIS reflection elements
d_r	RIS coverage radius
d_{bs}	BS coverage radius
$\mathbf{h}_{n,m}$	Channel vector between user n and BS m
$\mathbf{h}_{u,n}$	Channel vector between user n and RIS u
$\mathbf{h}_{m,u}$	Channel vector between BS m and RIS u
α_{pl}, ι	The path-loss and the path-loss exponent
$\mathbf{\Theta}$	Diagonal phase shift matrix
η_K	Reflection amplitude
$\phi_{u,k}$	Phase shift adjustment at the associated RIS u
P_n	Transmit power of the user n for data offloading
g_n	The indicator of whether there is a reflection link for user n through RIS
$\gamma_{m,n}$	Received signal-to-interference-plus-noise ratio (SINR) for the user n
$r_{m,n}$	Achievable data rate for transmissions
B_n	Allocated bandwidths for the user n
f_m^{up}	Computation capability for dominant BSs
f_m^{down}	Computation capability for non-dominant BSs
ρ_n	The execution indicator indicates that the user n is allocated to the dominant BS or non-dominant BS
$\Phi_{m,n}$	The user allocation decision whether the user n is allocated to BS m

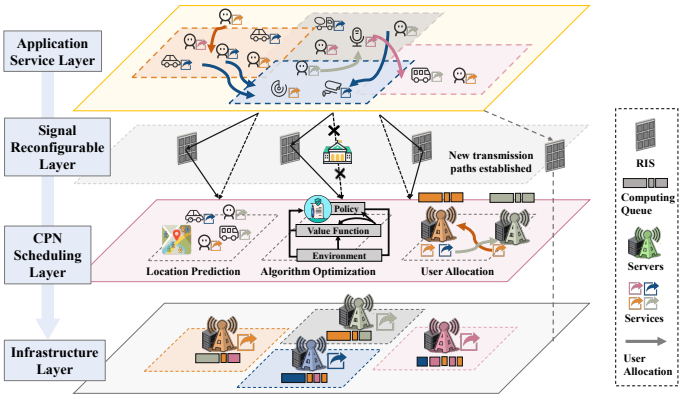


Fig. 3. The framework of our proposed architecture for user allocation.

$$\mathbf{h}_{n,m}(t) = \sqrt{\alpha_{pl} d_{n,m}^{-\iota}(t)} \widetilde{\mathbf{h}}_{n,m}(t), \quad (2)$$

where α_{pl} denotes path-loss and ι denotes the path-loss exponent. $d_{n,m}$ denotes the distance between the user n and BS m which can be calculated by Eq. (1). $\widetilde{\mathbf{h}}_{n,m}$ are random scattering components following Gaussian distribution. It should be noted that the channel vector $\mathbf{h}_{n,m}(t)$ is time varying according to the user's mobility at different time slots. To improve communication performance, the BS coordinates the uplink signal reflection via the RIS, enabling each RIS to dynamically adjust its phase and reflection coefficient.

Specifically, for the RIS located in area m, the corresponding phase shift matrix is configured as a diagonal matrix, $\mathbf{\Theta} \in \mathbb{C}^{K \times K}$, which is given by $\mathbf{\Theta}_u(t) = \text{diag}\{\eta_1 e^{j\phi_{u,1}}(t), \cdots, \eta_k e^{j\phi_{u,k}}(t), \cdots, \eta_K e^{j\phi_{u,K}}(t)\}$, where η_k, $\phi_{u,k}$ are reflection amplitude and phase shift adjustment at the associated u-th RIS, and $\eta_k \in [0, 1]$ and $\phi_{u,k} \in [0, 2\pi)$.

By accounting for realistic hardware limitations, we assume that the phase shift at each RIS element can only take on a limited number of distinct values. Assuming that the phase shifts at each RIS element are uniformly quantized using b bits [36] to indicate the number of phase shift levels F, then the available phase shifts for each RIS element is represented by $\mathcal{P}_b = \{0, \Delta\phi, 2\Delta\phi, \ldots, (F-1)\Delta\phi\}$, where $\Delta\phi = 2\pi/2^b$. Furthermore, we assume an ideal electromagnetic wave signal reflection through each RIS [1], i.e., $\eta_1 = \eta_2 = \cdots = \eta_K = 1$.

For the RIS-assisted links, we denote the propagation channels from the user n to the RIS u and from the RIS u to the BS m as $\mathbf{h}_{u,n} \in \mathbb{C}^{K\times 1}$ and $\mathbf{h}_{m,u} \in \mathbb{C}^{Z\times K}$, respectively. Both channel vectors are assumed to follow Rician block-fading [38], and are expressed as

$$\mathbf{h}_{u,n}(t) = \sqrt{\alpha_{pl} d_{u,n}^{-\iota}(t)} \left(\sqrt{\frac{\xi}{1+\xi}} \mathbf{h}_{u,n}^{\mathrm{LoS}}(t) + \sqrt{\frac{1}{1+\xi}} \mathbf{h}_{u,n}^{\mathrm{NLoS}}(t) \right), \tag{3}$$

and

$$\mathbf{h}_{m,u}(t) = \sqrt{\alpha_{pl} d_{m,u}^{-\iota}(t)} \left(\sqrt{\frac{\xi}{1+\xi}} \mathbf{h}_{m,u}^{\mathrm{LoS}}(t) + \sqrt{\frac{1}{1+\xi}} \mathbf{h}_{m,u}^{\mathrm{NLoS}}(t) \right), \tag{4}$$

where $d_{u,n}$ and $d_{m,u}$ denote the distance from the user n to the RIS u and from the BS m to the RIS u, respectively. ξ denotes the Rician factor.

Assume the transmitted data symbol from the user n to the BS m is s_n, which has zero mean and unit variance. Then the received signal $\mathbf{y}_{m,n}$ (by ignoring the time index t for brevity) at the BS m from the user n can be expressed as

$$\mathbf{y}_{m,n} = \underbrace{\sqrt{P_n}\left(g_n \mathbf{h}_{m,u}\mathbf{\Theta}_u \mathbf{h}_{u,n} + \mathbf{h}_{m,n}\right) s_n}_{\text{Communication signal}} + \underbrace{\mathbf{v}_n}_{\text{Noise}} + \underbrace{\sum_{i=1, i\neq n}^{N_m} \sqrt{P_i}\left(g_i \mathbf{h}_{m,u}\mathbf{\Theta}_u \mathbf{h}_{u,i} + \mathbf{h}_{m,i}\right) s_i}_{\text{Interference from other users}}, \tag{5}$$

where P_n denotes the transmit power of the user n for data offloading; $\mathbf{v}_n$ is the additive white Gaussian noise (AWGN) vetor and $\mathbf{v}_n \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_Z)$; $g_n \in \{0, 1\}$ is the indicator of whether the reflection link via RIS is scheduled, $g_n = 1$ means that the uplink data offloading from the user n is assisted by a scheduled RIS $u \in \mathcal{U}$, otherwise $g_n = 0$. In particular, the last term in Eq. (6) denotes the interference from other users and N_m represents the number of users allocated to the BS[2]. Furthermore, we assume that the channel vectors are well known and the channel estimation approaches refer to [39], [40], which is out of the scope of our work.

By implementing the receive beamforming vector $\mathbf{w}_n \in \mathbb{C}^{Z\times 1}$ at the BS m for the user n, the decoded signal $\hat{y}_{m,n}$ can be expressed as[3]

$$\begin{aligned}\hat{y}_{m,n} =& \mathbf{w}_n^H \mathbf{y}_{m,n} \\ =& \sqrt{P_n}\mathbf{w}_n^H \left(g_n \mathbf{h}_{m,u}\mathbf{\Theta}_u \mathbf{h}_{u,n} + \mathbf{h}_{m,n}\right) s_n + \mathbf{w}_n^H \mathbf{v}_n \\ &+ \sum_{i=1, i\neq n}^{N_m} \sqrt{P_i}\mathbf{w}_n^H \left(g_i \mathbf{h}_{m,u}\mathbf{\Theta}_u \mathbf{h}_{u,i} + \mathbf{h}_{m,i}\right) s_i. \end{aligned} \tag{6}$$

Therefore, the received signal-to-interference-plus-noise ratio (SINR) for the user n at the t-th time interval is given by

$$\gamma_{m,n} = \frac{P_n \left|\mathbf{w}_n^H \left(g_n \mathbf{h}_{m,u}\mathbf{\Theta}_u \mathbf{h}_{u,n} + \mathbf{h}_{m,n}\right)\right|^2}{\sum_{i=1, i\neq n}^{N_m} P_i \left|\mathbf{w}_n^H \left(g_i \mathbf{h}_{m,u}^H \mathbf{\Theta}_u \mathbf{h}_{u,i} + \mathbf{h}_{m,i}\right)\right|^2 + \sigma^2}. \tag{7}$$

Then, the achievable data rate for transmissions between the user n and the BS m at each time interval t is obtained as

$$r_{m,n} = B_n \log_2\left(1 + \gamma_{m,n}\right), \tag{8}$$

where B_n denotes the allocated bandwidth for the user n and is treated as the system parameter, thus not included as an optimization variable.

C. Latency Model

We denote the service request from the users as $J_{n,s} \triangleq (D_{n,s}, Q_{n,s})$. Additionally, each service has a maximum latency $T_n^{\max}$. Each BS handles requests at different processing speeds. Therefore, there are two categories of latency: **transmission latency** $T_{m,n}^{tr}$ and **computation latency** $T_{m,n}^{co}$.

1) Transmission latency: The transmission latency is affected by factors such as the size of the data $D_{n,s}$, the transmission data rate $r_{m,n}$, RIS phase shift matrix $\mathbf{\Theta}_u$, and the distance among the user, BS, and RIS. We model the wireless transmission using Shannon's capacity theorem, the specifics of which have been elaborated in the preceding section. Consequently, transmission latency is formulated as

$$T_{m,n}^{tr} = \frac{D_{n,s}}{r_{m,n}}, \tag{9}$$

where $r_{m,n}$ denotes the transmission data rate, which can be calculated by Eq. (8). Due to the typically small size of the returned results, the downlink transmission latency is disregarded here.

2) Computation latency: The computation latency $T_{m,n}^{co}$ is affected by the size of the processed service $D_{n,s}$, the total number of CPU cycles required to complete the service $Q_{n,s}$ and BS's computing capacity, which varies for different types of user services. When a user is assigned, the BSs with higher computation capability f_m^{up} for the specific service type are designated as the dominant BSs for that service, while others act as non-dominant BSs with computation capability f_m^{down}.

For example, suppose BS_1 is more efficient at handling video-processing services, while BS_2 is more efficient at processing text-based services. If the user requests a video service, BS_1 would be the dominant BS (f_m^{up}), and BS_2 would be the non-dominant BS (f_m^{down}). Conversely, if the user requests a text-based service, BS_2 becomes the dominant BS (f_m^{up}), and BS_1 is the non-dominant BS (f_m^{down}). Thus, the computation latency can be expressed as

$$T_{m,n}^{co} = \frac{D_{n,s} Q_{n,s}}{\left(\mathbb{I}_{\{\rho_n=1\}} f_m^{up} + \mathbb{I}_{\{\rho_n=0\}} f_m^{down}\right)}, \tag{10}$$

[1]Although the reflection amplitude $\eta_k, \forall k$ is related to phase shift implemented [37], we consider an ideal reflection coefficient at each RIS element in our paper for simplicity because it is out the scope of our work.

[2]Noted that interference from adjacent BSs is assumed to be disregarded for brevity, which is often true in practice due to the large inter-cell distance.

[3]Given the known channel, the optimal receive beamforming vector can be obtained by widely used singular value decomposition approach [41], [42].

where $\rho_n \in \{0,1\}$ denotes the execution indicator, which indicates that the user is allocated to the dominant BS with processing capacity f_m^{up} when $\rho_n = 1$, and 0 is otherwise. Therefore, at the time interval t, the service request latency can be calculated as $T_{m,n}(t) = T_{m,n}^{tr}(t) + T_{m,n}^{co}(t)$. Note that the uplink transmission of each allocated request is assumed to be completed within interval t under the current RIS configuration, whereas the computation may continue across later slots and is independent of subsequent RIS adjustments.

D. Problem Formulation

Besides the service request latency defined in Sec. III-C, the decision is generated at the CPN scheduling layer shown in Fig. 3 at the beginning of each slot, and the resulting allocation and configuration are then executed by the system to perform the data transmission and computation. We denote the corresponding decision-making latency as $T^{dec}(t)$, which is primarily determined by the computational complexity of the adopted decision algorithm and the hardware platform used for execution. From the perspective of a single service request, the latency experienced by user n when it is served by BS m can therefore be written as $T^{dec}(t) + T_{m,n}(t)$. It is worth noting that $T^{dec}(t)$ is incurred once per slot for generating a joint scheduling decision for all allocated users and RIS phase adjustment, while $T_{m,n}(t)$ is incurred per service request.

This paper aims to enhance system performance by minimizing the total system latency, which consists of the CPN decision-making latency and the service request latency of all allocated users. The service request latency is optimized through user allocation and RIS phase adjustment, while the decision-making latency is effectively reduced by leveraging user location prediction, which enables decision generation ahead of actual data transmission and computation.

We define the user allocation decision $\Phi_{m,n} \in \{0,1\}$, where $\Phi_{m,n} = 1$ denotes the user n is allocated to BS m, and $\Phi_{m,n} = 0$ otherwise. Thus, the user allocation decisions can be defined as $\mathbf{\Phi} := \{\Phi_{m,n}\}_{m\in\mathcal{M},n\in\mathcal{N}}$ and RIS phase shift matrix is $\mathbf{\Theta}$. The optimization problem of total system latency minimization across T time intervals can be formulated as

$$\mathbf{(P)}: \min_{\mathbf{\Theta},\mathbf{\Phi}} \quad \sum_{t=1}^{T}\left(T^{dec}(t) + \sum_{m=1}^{M}\sum_{n=1}^{N}\Phi_{m,n}(t)\,T_{m,n}(t)\right),$$

$$\text{s.t.} \quad \text{C1}: d_{n,m}(t), d_{m,u}(t) \le d_{bs}(t), \forall n \in \mathcal{N}, m \in \mathcal{M}, u \in \mathcal{U},$$

$$\text{C2}: d_{u,n}(t) \le d_r, \forall n \in \mathcal{N}, u \in \mathcal{U},$$

$$\text{C3}: 0 \le \phi_{u,k}(t) < 2\pi, \forall u \in \mathcal{U}, k \in \mathcal{K},$$

$$\text{C4}: \sum_{m=1}^{M}\Phi_{m,n}(t) = 1, \forall m \in \mathcal{M}.$$

C1 and C2 ensure that all allocated users are within the effective coverage areas of both BSs and RISs. C3 represents the phase shift adjustment within the range of $[0, 2\pi]$. C4 ensures that each user can only be served by one BS.

IV. AI-generated Location Prediction and Collaborative Optimization Strategy

In this section, we present the solution for solving the optimization problem formulated in Sec. III. To address mobility-aware user allocation more effectively, we first predict future user locations so that allocation decisions can be conducted in advance, which will further reduce decision-making latency. Since RIS phase adjustment directly affects communication link quality and thereby influences allocation decisions, we combine RIS control and allocation into a unified optimization step. Accordingly, our approach consists of two stages: location prediction and collaborative optimization.

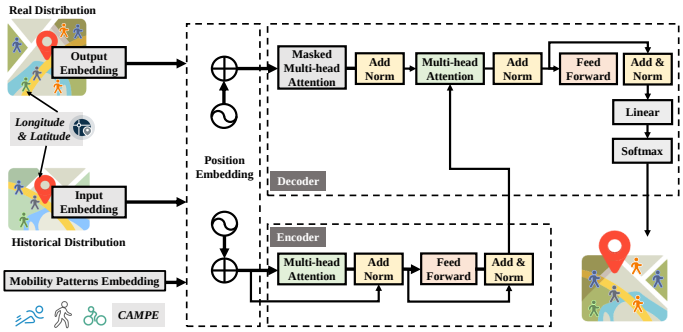


Fig. 4. CAMPE-assisted user location prediction model.

A. User Location Prediction Model

To refine the first stage of the solution, the pivotal initial step involves acquiring precise predictions of location movement. This section introduces a location prediction model empowered by *CAMPE*. Our approach utilizes a transformer-based architecture to predict users' subsequent locations by integrating contextual information from mobility patterns into the model. The context-aware method integrates additional contextual information, such as time of day, day of the week, and mobility patterns, to enhance the model's understanding of user trajectories. By considering mobility patterns, we capture the user movement behavior across different trajectories.

We define the user's trajectory as a sequence of time-ordered records. Each record is denoted as (n, L_i^n, κ_i), indicating user n visit location L_i^n at timestamp κ_i. Note that the term timestamp in prediction refers only to real-world recording time, which is different from the time slot used in the allocation optimization process. L_i^n represents spatial coordinates that include latitude and longitude. Thus, given a query trajectory $\mathcal{V} = [(n, L_1^n, \kappa_1), \ldots, (n, L_i^n, \kappa_i)]$, the objective is to predict the n-th user's next location, denoted as L_{i+1}^n, based on the observed trajectory up to timestamp κ_i. A trajectory is conceptualized as a time-ordered sequence of points, each encapsulated by a temporal stamp and spatial coordinates. When considering the problem of predicting location, we focus on the historical sequence of locations as shown in Fig. 4.

1) Position Embedding: The transformer predicts each user's future trajectory by encoding the current and historical locations. Each user exhibits personal movement patterns [43]. Therefore, our model not only embeds historical and current locations but also introduces *CAMPE* to exploit temporal and mobility pattern embeddings into the predictive framework.

Specifically, *CAMPE* allows the embeddings M_i to capture distinct behavioral nuances associated with different speeds, ranging from slow-paced walking to rapid vehicular movement. Moreover, time features are encoded at two granularities:

hours κ_i^h and days κ_i^d to distinguish different levels of periodicity in location visits. Subsequently, we map the trajectory to a high-dimensional vector space E_n, formulated as

$$\mathrm{E}_i = \tau\left(L_i^n, M_i, \kappa_i^h, \kappa_i^d, I_i, \omega\right), \tag{11}$$

where $\tau(\cdot)$ denotes the position embedding function that transforms these categorical features into fixed-dimensional real-valued vectors and adds all sequence features together, given as $E_i^l = \omega_l L_i^n$, $E_n^h = \omega_h \kappa_i^h$, $E_i^m = \omega_m M_i$, $E_n^d = \omega_d \kappa_i^d$, $E_i^I = \omega_I I_i$. L_i^n, M_i, κ_i^h and κ_i^d denote the one-hot encoded original features and ω represents the weight parameters in the embedding process.

These embeddings are integrated into the transformer architecture, interacting with spatial and temporal features to provide a comprehensive input sequence. The method captures both the static elements of user location and stay duration and the speed-related attributes of user mobility, thus better understanding user location-specific trends (from stay duration) and user-specific preferences (from mobility patterns).

2) Location-Adaptive Transformer: An efficacious model for predicting future locations must effectively discern patterns and comprehend the intricate multilevel periodicity inherent in complex spatio-temporal historical data sequences. Leveraging a transformer-based framework, our approach extracts and learns the nuances of location transition dynamics from historical data encompassing location points and temporal intervals, encapsulated within a comprehensive embedding vector E_i.

The core of transformer architecture is the multi-head self-attention mechanism. This mechanism is structured to integrate queries Q, keys K and values V in order to merge information from various representation subspaces, formally expressed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V, \tag{12}$$

where D is the dimension of the key vector. Subsequently, multi-head attention is formed by concatenating the outputs of k attention functions.

$$\text{MultiHead}\left(H^z\right) = \text{Concat}\left(\textit{head}_1, \ldots, \textit{head}_K\right)W^o, \tag{13}$$

$$\textit{where head}_j = \text{Attention}\left(H^z W_j^Q, H^z W_j^K, H^z W_j^V\right).$$

H^z represents the current state characteristics, z denotes the step within the recurrent computational sequence and W^o represents the output weight matrix. The self-attention mechanism enables the model to extract information from each step of the historical sequence and assess its significance.

Moreover, we introduce forward masking operations to prevent the attention function from being affected by information after the time series. This process guarantees the extraction of long-term dependencies from past patterns. Due to the multi-head design, the model retains multiple sets of parameter matrices concentrated in the historical sequence, capturing the movement's periodic characteristics.

3) Loss Function: The location prediction model subsequently receives the aggregated vector g_i, which encapsulates both the predictive information of location and mobility patterns. The probabilities of each location are calculated using a linear projection followed by a softmax function, as demonstrated in Eq. (14).

$$\begin{aligned} P(\hat{L}_{i+1}^n) &= \text{softmax}\left(\text{Linear}_a(g_i)\right), \\ P(\hat{M}_{i+1}) &= \text{softmax}\left(\text{Linear}_b(g_i)\right), \end{aligned} \tag{14}$$

where $P(\hat{L}_{i+1}^n)$ and $P(\hat{M}_{i+1})$ contain the probabilistic distribution across all potential locations and mobility patterns. The location with the highest value in this probability distribution is the most likely target for the next access.

In the given training set, the prediction task can be formulated as a multi-class classification problem. We employ the multi-class cross-entropy loss function in Eq. (15) to quantitatively assess the discrepancies between the predicted probabilities and the actual categorical labels.

$$\begin{aligned} \mathcal{L}_{loc} &= -\sum_{j=1}^{|\mathcal{X}|} \vartheta_j^L \log\left(P_j(\hat{L}_{i+1}^n)\right), \\ \mathcal{L}_{mp} &= -\sum_{j=1}^{|\mathcal{Y}|} \vartheta_j^M \log\left(P_j(\hat{M}_{i+1})\right), \end{aligned} \tag{15}$$

where $\mathcal{X}$ is the set that contains all known positions and $\mathcal{Y}$ contains category of mobility pattens. ϑ_j^L and ϑ_j^M are the one-hot vectors to represent true next location and mobility patterns. Therefore, this dual prediction framework is managed by a combined loss function, which includes a separate cross-entropy term for mobility pattern discrepancies as

$$\mathcal{L}_{total} = \varphi\mathcal{L}_{loc} + (1-\varphi)\mathcal{L}_{mp}, \tag{16}$$

where $\varphi \in (0, 1)$ represents the hyperparameter that balances the relative contributions of location and mobility patterns prediction losses during the training phase.

B. DiffUser Algorithm

To tackle the second stage problem of collaborative optimization of RISs phase adjustment and user allocation, we propose the *DiffUser* algorithm, which utilizes a diffusion model to generate optimal decision-making. As an advanced generative model, the diffusion model adeptly captures complex distributions by inverting the denoising process to synthesize novel data. Such expressive capability is essential for representing the high-dimensional, tightly coupled decision space formed by joint RIS phase adjustment and user allocation, enabling *DiffUser* to produce more stable and higher-quality actions [44] than conventional deep reinforcement learning-based policy parameterizations.

Specifically, we define the environment state as $\mathbf{s}(t) = \{s_n(t)|n = 1, 2, ..., N\}$, where the state space of the user n denotes $s_n(t) = \{L_n(t), d_{n,m}(t), S_n(t)\}$. $L_n(t)$ denotes the location of user n, $d_{n,m}(t)$ means the distance between the user and the corresponding BS, and $S_n(t)$ represents the users' current service types. Furthermore, the environment needs to evaluate the joint optimization actions, including the allocation strategy $\mathbf{a}_n(t) = \{u_n(t)|n = 1, 2, ..., N\}$ and the RIS phase shift strategy $\mathbf{a}_u(t) = \{\phi_{u,k}(t)\}$. $u_n(t)$ denotes BSs' index assigned by users and $\phi_{u,k}(t)$ is the RIS phase shift, where $\phi_{u,k} \in \mathcal{P}_b = \{0, \Delta\phi, \cdots, (F-1)\Delta\phi\}$. Therefore, the action space is denoted as $\mathbf{a}(t) = \{a_n(t), a_u(t)\}$.

DiffUser aims to optimize the user allocation and RIS phase adjustment strategy of CPNs by reducing total service latency. We first define the reward of the *DiffUser* as the negative value of average service requset latency $\frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{N}\Phi_{m,n}(t)\,T_{m,n}(t)$ in problem **P**. Since time is inherently a positive value, to improve clarity, we then apply

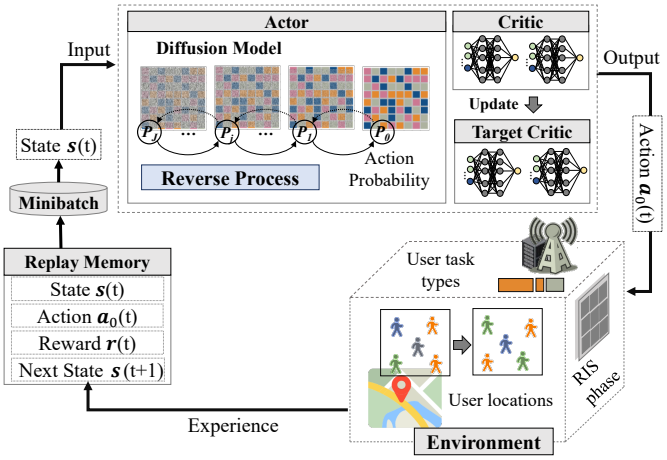


Fig. 5. Diffusion-driven user allocation model.

the exponential function exp(x) to make the reward positive. The optimization algorithm is driven by the diffusion model, which optimizes reward by determining the optimal action.

1) Diffusion-driven Action Generation Strategy: The diffusion process enhances the model's ability to capture and represent complex data patterns accurately and significantly simplifies data processing during training. This enables our algorithm to effectively represent complex data distributions, providing a foundation for more precise modeling of user allocations and strategies for adjusting the phases of RISs. Specifically, the generation of diffusion-driven action strategies is a process of denoising. Within the dynamic environment, the optimal decision solution for user allocation and RIS phase adjustment is the goal of the denoising network.

In the proposed action strategy context, the output is a probability distribution x_0 over decision choices conditional upon the observed environmental state. Given a specified target probability distribution, the forward process introduces a sequence of Gaussian noise at each timestep, denoted as

$$q\left(x_j \mid x_{j-1}\right)=\mathcal{N}\left(x_j ; \sqrt{1-\beta_j} x_{j-1}, \beta_j \mathbf{I}\right). \tag{17}$$

For β_j in the interval (0,1), the sequence is strictly increasing, i.e., $\beta_1<\beta_2<\ldots<\beta_J$. The posterior probability from the original input x_0 to the final data $x_{1:J}$ is expressed in the form as follows

$$q\left(x_{1:J} \mid x_0\right)=\prod_{j=1}^{\infty} q\left(x_j \mid x_{j-1}\right). \tag{18}$$

Simultaneously, a reverse process characterized by Gaussian transitions parameterized by θ is delineated as

$$p_\theta\left(x_{j-1} \mid x_j\right)=\mathcal{N}\left(x_{j-1} ; \mu_\theta\left(x_j, j\right), \mathbf{\Sigma}_\theta\left(x_j, j\right)\right). \tag{19}$$

Gaussian parameters mean and covariance are denoted as $\mu_\theta\left(x_j, j\right)$ and $\mathbf{\Sigma}_\theta\left(x_j, j\right)$, respectively, for the network to estimate. Therefore, the formula for the process of reverse diffusion can be written as

$$p_\theta\left(x_{0:J}\right)=p_\theta\left(x_J\right) \prod_{j=1}^{J} p_\theta\left(x_{j-1} \mid x_j\right). \tag{20}$$

Unlike traditional backpropagation algorithms in neural networks that directly optimize model parameters, *DiffUser* uses diffusion models to improve user allocation and the phase adjustment of RISs by iterative denoising the initial distribution [45], resulting in a more effective reward function. This module receives environmental information $\mathbf{s}(t)$ as input and makes decisions on user allocation and RIS phase adjustment based on the defined policy π_θ, which is denoted as

$$\begin{aligned}\pi_\theta(\mathbf{a}(t) \mid \mathbf{s}(t)) &= p_\theta\left(\mathbf{a}(t)_{0:J} \mid \mathbf{s}(t)\right) \\ &= \mathcal{N}\left(\mathbf{a}(t)_J ; \mathbf{0}, \boldsymbol{I}\right) \prod_{j=1}^{J} p_\theta\left(\mathbf{a}(t)_{j-1} \mid \mathbf{a}(t)_j, \mathbf{s}(t)\right).\end{aligned} \tag{21}$$

Here, the reverse process with parameter θ is expressed as

$$\begin{aligned}&p_\theta\left(\mathbf{a}(t)_{j-1} \mid \mathbf{a}(t)_j, \mathbf{s}(t)\right) \\ &=\mathcal{N}\left(\mathbf{a}(t)_{j-1} ; \mu_\theta\left(\mathbf{a}(t)_j, \mathbf{s}(t), j\right), \mathbf{\Sigma}_\theta\left(\mathbf{a}(t)_j, \mathbf{s}(t), j\right)\right),\end{aligned} \tag{22}$$

where Gaussian parameters mean and covariance are denoted as $\mu_\theta\left(\mathbf{a}(t)_j, \mathbf{s}(t), j\right)$ and $\mathbf{\Sigma}_\theta\left(\mathbf{a}(t)_j, \mathbf{s}(t), j\right)$ for the network to estimate. According to [46], the mean of reversed step is

$$\mu_\theta\left(\mathbf{a}(t)_j, \mathbf{s}(t)_j, j\right)=\frac{1}{\sqrt{\alpha_j}}\left(\mathbf{a}(t)_j-\frac{\beta_j}{\sqrt{1-\bar{\alpha}_j}} \epsilon_\theta\left(\mathbf{a}(t)_j, \mathbf{s}(t), j\right)\right), \tag{23}$$

where ϵ_θ denotes the function approximation based on neural network, $\alpha_j=1-\beta_j$ and $\bar{\alpha}_j=\prod_{j=1}^{J} \alpha_j$. The covariance matrix is denoted as $\mathbf{\Sigma}_\theta\left(\mathbf{a}(t)_j, \mathbf{s}(t), j\right)=\beta_j \mathbf{I}$.

Initially, we sample $\mathbf{a}_J(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, followed by constructing the reverse diffusion chain

$$\mathbf{a}_{j-1}(t) \mid \mathbf{a}_j(t)=\frac{\mathbf{a}_j(t)}{\sqrt{\alpha_j}}-\frac{\beta_j}{\sqrt{\alpha_j\left(1-\bar{\alpha}_j\right)}} \epsilon_\theta(\mathbf{a}_j(t), \mathbf{s}(t), j)+\sqrt{\beta_j} \epsilon. \tag{24}$$

To improve the sample quality, when j equals 1, ϵ is defined as zero [44]. Therefore, the decision for user allocation and RIS phase adjustment $\mathbf{a}_0(t)$ can be obtained.

2) Network Update Process: The *DiffUser* algorithm employs Soft Actor-Critic (SAC) to train the ϵ_θ in Eq. (24) with parameter θ. Our algorithm aims to learn a policy that maximizes cumulative reward while preserving exploration. The optimal policy in maximum entropy reinforcement learning is characterized as

$$\pi^*=\underset{\pi_\theta}{\operatorname{argmax}} \quad \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\left(r(t)+\xi \mathcal{H}\left(\pi_\theta\left(\cdot \mid \mathbf{s}(t)\right)\right)\right)\right], \tag{25}$$

where ξ represents the temperature parameter that balances the trade-off between optimizing reward and entropy and $\gamma \in [0,1)$ is the discount factor. The entropy is given by

$$\mathcal{H}(\pi_\theta(\cdot \mid \mathbf{s}(t)))=-\mathop{\mathbb{E}}_{\mathbf{a}_0(t) \sim \pi_\theta(\cdot \mid \mathbf{s}(t))}[\log(\pi_\theta(\mathbf{a}_0(t) \mid \mathbf{s}(t)))]. \tag{26}$$

As illustrated in Fig. 5, *DiffUser* utilizes the diffusion-driven action generation model as the actor network to predict the optimal action $\mathbf{a}_0(t)$ by ϵ_θ. During the training process for each episode, the actor network selects an action $\mathbf{a}_0(t)$ according to the current policy ϵ_θ. This action is executed in the environment, leading to the next state $\mathbf{s}(t+1)$ and receiving a reward $r(t)$. The tuple $(\mathbf{s}(t), \mathbf{a}_0(t), r(t), \mathbf{s}(t+1))$ is stored in the replay buffer $\mathcal{R}$.

At each learning step, a batch of transitions $(\mathbf{s}(i), \mathbf{a}_0(i), r(i), \mathbf{s}(i+1))$ is sampled from the replay buffer. The target value $z(i)$ is calculated as the sum of the reward and discounted minimum Q-value of the next state-action pair, minus the log probability of the action, thereby promoting exploration. This is used to update the

Algorithm 1: *DiffUser* Algorithm

Initialized parameters for the critic network τ_1, τ_2, actor network ϵ_θ. An empty experience replay memory $\mathcal{R}$;
for *the training episode* $e = 1$ *to* $\mathcal{E}$ **do**
  Initialize a random process $\mathcal{N}$ for allocation exploration.;
  for $t = 1, 2, \ldots, T$ **do**
    Observe the environment and set the action $\mathbf{a}_J(t)$ as Gaussian noise;
    for *the denoising process* **do**
      Use deep neural networks to learn noise distribution;
      Generate the action $\mathbf{a}_0(t)$ by denoising $\mathbf{a}_J(t)$ through ϵ_θ;
    end
    Execute action, collect reward $r(t)$, and observe next state $\mathbf{s}(t+1)$;
    Store $(\mathbf{s}(t), \mathbf{a}_0(t), r(t), \mathbf{s}(t+1))$ into replay memory;
    Sample the minibatch of $N_\mathcal{R}$ from $\mathcal{R}$;
    for $i = 1, 2, \ldots, N_\mathcal{R}$ **do**
      Calculate the target value in Eq.(28);
      Update the critic networks in Eq.(27);
      Update the actor network utilizing the sampled policy gradient.;
      Update target networks: $\hat{\tau}_1, \hat{\tau}_2$ using $\hat{\tau}_l \leftarrow \omega\tau_l + (1-\omega)\hat{\tau}_l$, for $l \in \{1, 2\}$;
    end
  end
end
return The optimal policy π^*;

critic networks by minimizing the loss function, which is averaged over the batch

$$\mathcal{L}(\tau_l) = \frac{1}{N_\mathcal{R}} \sum_{i=1}^{N_\mathcal{R}} \left(Q_{\tau_l}(\mathbf{s}(i), \mathbf{a}_0(i)) - z(i)\right)^2. \quad (27)$$

The target value is calculated by

$$\begin{aligned} z(i) = r(i) + \gamma \min_{l=1,2} Q_{\hat{\tau}_l}(\mathbf{s}(i+1), \hat{\mathbf{a}_0}(i+1)) \\ - \xi \log \pi_\theta(\hat{\mathbf{a}_0}(i+1) \mid \mathbf{s}(i+1)), \end{aligned} \quad (28)$$

where $\hat{\mathbf{a}_0}(i+1)$ can be obtained using $\epsilon_{\hat{\theta}}$. Following this, the actor network can be updated by

$$\frac{1}{N_\mathcal{R}} \sum_{i=1}^{N_\mathcal{R}} \left(\xi \log \pi_\theta(\mathbf{a}_0(i) \mid \mathbf{s}(i)) - \min_{l=1,2} Q_{\tau_l}(\mathbf{s}(i), \mathbf{a}_0(i))\right). \quad (29)$$

Finally, the parameters of the critic networks τ_1 and τ_2 are updated using soft updates with the target networks' parameters $\hat{\tau}_1$ and $\hat{\tau}_2$:

$$\begin{cases} \hat{\tau}_1 \leftarrow \omega\tau_1 + (1-\omega)\hat{\tau}_1, \\ \hat{\tau}_2 \leftarrow \omega\tau_2 + (1-\omega)\hat{\tau}_2, \end{cases} \quad (30)$$

where ω confined within the interval (0, 1], represents the rate of the soft update for target networks.

Conclusively, the *DiffUser* optimizes policy and Q-function parameters in reinforcement learning by integrating a diffusion model approach in Algorithm 1. The algorithm generates actions by denoising the noisy inputs through the actor network, executes them in the environment, collects rewards, and stores the experience in the replay memory. A minibatch $(\mathbf{s}(i), \mathbf{a}_0(i), r(i), \mathbf{s}(i+1))$ is sampled from the memory to update the policy of the networks. Specifically, the convergence proof of *DiffUser* is provided in Appendix A.

C. Complexity Analysis

In this section, we discuss the time and space complexity of our algorithm, which can be divided into two parts:

1) User location prediction: For the time complexity, the transformer-based architecture is influenced by the input sequence length ν_p, the dimension of the embedding vector ν_d. The self-attention operation requires $\mathcal{O}\left(\nu_p^2 \times \nu_d\right)$ and the feed-forward neural networks within each transformer layer contribute a time complexity of $\mathcal{O}\left(\nu_p \times \nu_d^2\right)$. Therefore, the time complexity of the user location prediction algorithm is obtained by $\mathcal{O}\left(\nu_p^2 \times \nu_d + \nu_p \times \nu_d^2\right)$.

Regarding space complexity, the model consumes $\mathcal{O}\left(\nu_p^2\right)$ space for attention weight matrices and $\mathcal{O}\left(\nu_p \times \nu_d\right)$ for query, key, and value vectors. The feed-forward neural networks add an additional $\mathcal{O}\left(\nu_p \times \nu_d^2\right)$ space requirement. Therefore, the space complexity of the user location prediction algorithm is obtained by $\mathcal{O}\left(\nu_p^2 + \nu_p \times \nu_d + \nu_p \times \nu_d^2\right)$.

2) DiffUser algorithm: For the time complexity, the algorithm largely relies on training and denoising process, which is expressed as $|\mathcal{E}| \times T \times J \times |\xi_a| + |\mathcal{E}| \times T \times 2|\xi_c|$, where $|\mathcal{E}|$ and T are the number of training episode and step and J is the number of denoising steps. $|\xi_a|$ and $|\xi_c|$ denote the number of parameters in the actor and critic networks.

For the space complexity, the algorithm requires storage for the parameters of both the actor and critic networks, as well as for the experience replay buffer. Therefore, the overall space complexity of the *DiffUser* algorithm is $\mathcal{O}\left(|\xi_a| + |\xi_c| + N_\mathcal{R}\right)$.

V. Performance Evaluation

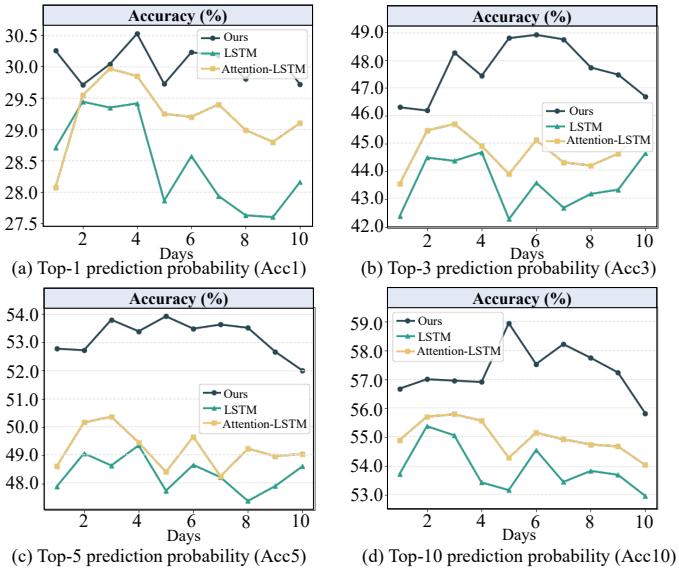
This section focuses on the performance evaluation of our proposed framework in three aspects: ***location prediction accuracy*** assisted by *CAMPE*, ***collaborative optimization effectiveness*** of *DiffUser* algorithm and ***prediction effectiveness in user allocation***. The simulation experiments are implemented in Python 3.9 and performed on a laptop with an NVIDIA GeForce RTX 3080 Ti GPU and Intel Core i9-12900K CPU.

A. Location Prediction Evaluation

1) Dataset Description: We utilized a GPS trajectory dataset collected from the GeoLife project [47]. This dataset was gathered by Microsoft Research Asia from April 2007 to August 2012, spanning three years. It comprises the travel modes and movement trajectories of 182 users. The trajectories are represented by a series of points with timestamps, each containing latitude, longitude, and altitude information. The dataset includes ten mobility modes, such as *walking, biking,*

TABLE III
PERFORMANCE EVALUATION RESULTS OF USER LOCATION PREDICTION.

Historical Info.	Algorithm	Acc_1	Acc_3	Acc_5	Acc_{10}
1-DAY	Markov Chain [51]	25.65	40.9	47.58	49.43
	LSTM [49]	28.71	42.39	47.9	53.72
	Attention-LSTM [50]	28.08	43.56	48.61	54.9
	Ours	**30.25**	**46.31**	**52.78**	**56.68**
3-DAY	Markov Chain [51]	25.65	40.9	47.58	49.43
	LSTM [49]	29.35	44.38	48.64	55.06
	Attention-LSTM [50]	29.97	45.71	50.37	55.79
	Ours	**30.04**	**48.27**	**53.8**	**56.96**
5-DAY	Markov Chain [51]	25.65	40.9	47.58	49.43
	LSTM [49]	27.87	42.28	47.75	53.16
	Attention-LSTM [50]	29.24	43.91	48.41	54.28
	Ours	**29.72**	**48.79**	**53.92**	**58.95**
10-DAY	Markov Chain [51]	25.65	40.9	47.58	49.43
	LSTM [49]	28.16	44.66	48.61	52.96
	Attention-LSTM [50]	29.1	45.72	49.05	54.03
	Ours	**29.71**	**46.69**	**52.00**	**55.82**



(a) Top-1 prediction probability (Acc1) (b) Top-3 prediction probability (Acc3)
(c) Top-5 prediction probability (Acc5) (d) Top-10 prediction probability (Acc10)

Fig. 6. Location prediction accuracy with different benchmarks.

bus and *car*. In our experiments, these labels are categorized into three groups: *fast*, *slow*, and *driving*. For instance, both bus and car are classified under the *driving* category.

In contrast to prior approaches that filter out locations with low visit frequencies, we retain all locations, as infrequent visits may still reflect meaningful variations in user mobility caused by random events. For each user, all trajectory records are sorted by their recorded time, with the first 60% used for training, the subsequent 20% for validation, and the remaining 20% for testing.

2) Implementation Details: We utilize the ***top-k accuracy*** as the evaluation metric for next-location prediction. The test set consists of S prediction samples. For the s-th sample, the model outputs a probability distribution $P(\hat{L}^{(s)})$ [48] over all candidate locations, from which the top-k locations form the prediction set $\mathcal{C}_k^{(s)}$. The top-k accuracy is defined as

$$\mathrm{Acc}_k = \frac{1}{S}\sum_{s=1}^{S}\mathbb{I}_{\left\{L_{\mathrm{true}}^{(s)}\in\mathcal{C}_k^{(s)}\right\}}, \tag{31}$$

where $L_{\mathrm{true}}^{(s)}$ denotes the ground-truth next location of the s-th prediction sample. $\mathbb{I}(\cdot)$ is the indicator function that evaluates whether the true next location of the user lies within the model-generated top-k candidate set. It outputs 1 when the prediction successfully includes the correct location, and 0 otherwise. Thus Acc_1 represents the accuracy of predicting the single most probable location as the true next location. Acc_3 indicates whether the user's actual next location appears among the three most likely locations provided as possible candidates. Acc_5 and Acc_{10} follow the same definition.

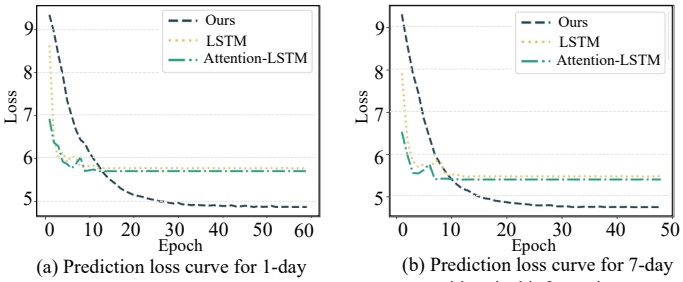
To demonstrate the benefits of the prediction algorithm, we compare it with the following two benchmarks: 1) *LSTM network [49]:* It is a recursive neural network structure suitable for processing and memorizing long-term dependencies. 2) *Attention-LSTM [50]:* It introduces a self-attention mechanism between the current and previous hidden states, which can dynamically adjust the weight of hidden states based on the importance of each time step in the input sequence.

3) Numerical Results: ***Analyze the performance between different strategies.*** There are significant differences in the performance of various algorithms when considering the accuracy of user location prediction. The Markov chain, grounded on the premise that the next state is determined only by current state, lacks the ability to leverage comprehensive historical data. This characteristic results in Markov chains exhibiting poor performance in all comparisons, especially in scenarios that require the use of historical data for prediction.

In contrast, LSTM networks mark a substantial advancement in predictive accuracy, demonstrating a 4.29% improvement in Acc_{10} over Markov chains. Moreover, integrating a self-attention mechanism in the Attention-LSTM algorithm further refines prediction accuracy in Fig. 6 by dynamically adjusting the relevance of past states. The performance of our proposed *CAMPE* assisted prediction model is the most robust, particularly in terms of the prediction loss rate shown in Fig. 7. A lower loss rate is directly associated with higher predictive accuracy, reflecting our model's stronger learning capacity and efficiency in processing user mobility patterns, especially with more historical information. An accuracy of 30.25% was attained on the Acc_1, surpassing the highest accuracy of 28.71% on LSTM by approximately 1.54%.

The implementation of the multi-head self-attention can simultaneously handle many representative subspaces, effectively capturing the overall patterns in user position sequences. Furthermore, by integrating contextual information from mobility patterns, the model has enhanced its capability to comprehend intricate relationships and extended dependency structures within sequences. Our method shows a stable growth trend in the prediction accuracy metric of Acc_3, Acc_5, and Acc_{10}, which can be shown in Fig. 8.

Analyze the performance between different historical information. Fig. 6 shows that the prediction accuracy does not improve monotonically with longer historical data. Using only 1-day of history yields insufficient behavioral context, preventing the model from capturing stable mobility patterns and thereby reducing accuracy. When extending the historical



(a) Prediction loss curve for 1-day historical information

(b) Prediction loss curve for 7-day historical information

Fig. 7. Prediction loss curve under different historical information (1&7 days).

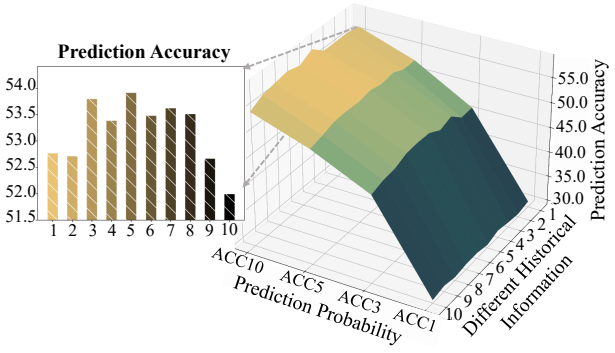


Fig. 8. Location prediction accuracy with different prediction probability.

window to 10 days, the prediction accuracy declines again. This indicates that overly long histories introduce outdated or noisy movement information, which increases the data complexity of the learning process and hampers the model's ability to extract relevant mobility features effectively.

Overall, although Table III shows some performance fluctuations as the number of historical days varies, these fluctuations remain relatively small. This indicates that the proposed prediction model is robust to variations in the duration of historical data.

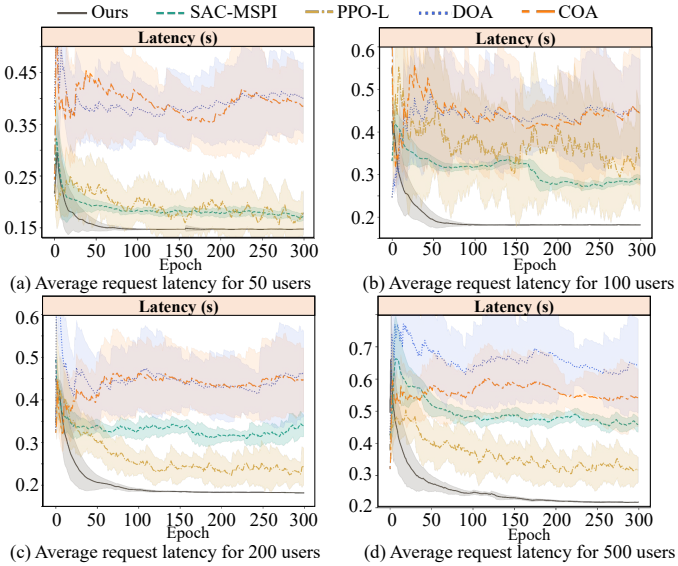
B. Collaborative Optimization Evaluation

1) Implementation Details: In this study, we utilize the publicly available EUA dataset [52], sourced from real-world data repositories, specifically focusing on the geographical locations of end-users within the Australian region.

Edge Servers (BSs): Our experiment simulates an area of 1200m x 1200m to represent the characteristics of a RIS-assisted communication environment accurately. The region is furnished with a total of 8 BSs and 8 RISs. The RISs are situated randomly, with only one BS within the assistive range. RISs with a constant height of 5 meters provide communication forwarding capabilities for all user requests within the specified region. Furthermore, it is assumed that every BS has pre-cached a particular service.

Edge Users: The user allocation is derived from geospatial information in the EUA dataset. We define 4 categories of service demand, and each user is randomly assigned a corresponding computing requirement. Their service demands vary over time according to predefined transition probabilities.

To verify the effectiveness of our proposed algorithm, we classify the comparison methods into learning-based and non-learning-based baselines. The learning-based methods are



(a) Average request latency for 50 users

(b) Average request latency for 100 users

(c) Average request latency for 200 users

(d) Average request latency for 500 users

Fig. 9. The average service request latency comparison with different benchmarks.

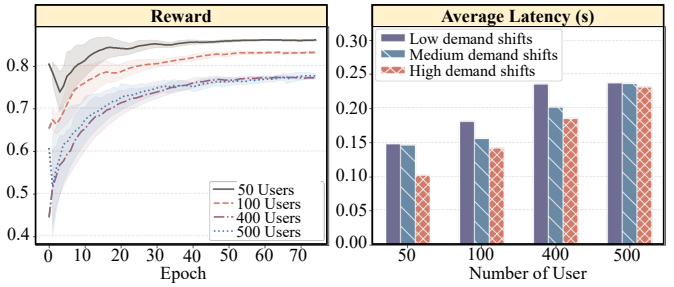
1) *Proximal Policy Optimization (PPO-L) [7]:* The system employs the PPO framework for policy learning.
2) *Soft Actor-Critic Algorithm (SAC-MSPI) [8]:* The system operates within the framework of maximum entropy reinforcement learning and employs entropy as a reward mechanism to incentivize agents to explore states.

The non-learning-based baselines are

1) *Distance-Optimal Allocation (DOA) [9]:* This strategy assigns each user to the nearest available BS, aiming to improve transmission efficiency.
2) *Computation-Optimal Allocation (COA) [10]:* It allocates each user to the BS with the highest processing capability for its service type. In our work, this corresponds to selecting the dominant BS, i.e., the BS with higher computation capability for that service type.

2) Numerical Results: ***Learning performances.*** We evaluate the effects of different strategies on different user numbers. In Fig. 9, under all testing conditions, our proposed algorithm exhibits the highest average return and optimal performance stability. Specifically, for a relatively small number of user assignments (50 and 100), our algorithm is more stable and has better convergence performance. Compared with the PPO-L, SAC-MSPI, DOA and COA algorithms, our algorithm has a lower latency under an allocation situation for 500 users, demonstrating the ability to find better solutions with improvements of 31.7%, 53.1%, 66.3% and 59.9%. Our algorithm demonstrates a 33.4% faster convergence rate compared to SAC-MSPI and a 76.7% improvement over PPO-L. Additionally, the wider shaded areas for SAC-MSPI and PPO-L reflect greater performance fluctuations and inconsistency, indicating that they are more sensitive to the effects of dynamic environments and high-dimensional allocation scenarios.

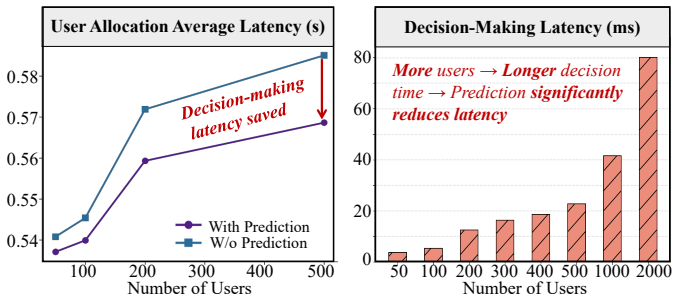
As the number of users increases, from 50 to 500, we observe that the performance of all algorithms decreases. However, our proposed algorithm is the least affected. For example, when the number of users increases to 200, our algorithm outperforms PPO-L, SAC-MSPI, DOA and COA algorithms



(a) Comparison of reward curves for different numbers of users

(b) Comparison of average request latency in different demand shift situations

Fig. 10. Average service request latency under different demand shifts.



(a) Average latency ($T^{dec} + \frac{1}{N}\sum_1^N T_{m,n}$) of user allocation with and without prediction

(b) Decision-making latency (T^{dec}) of *DiffUser* across different user numbers

Fig. 11. Evaluation of prediction effectiveness in user allocation.

by 23.24%, 46.76%, 60.57% and 59.40%, respectively. This indicates our algorithm is more effective in expanding to a larger user group, and its performance degradation is significantly smaller than the other algorithms. When dealing with multi-user allocation problems, our algorithm provides significantly better robustness.

Algorithm efficiency. As shown in Fig. 10 (a), when the number of users increases from 50 to 500, the reward of *DiffUser* decreases by only 12.7%. Considering that the size of the action space grows dramatically with more users, such a minor reduction demonstrates the strong scalability of the proposed method. More importantly, the convergence speed remains nearly identical across all settings, stabilizing at around 30 epochs. This stable convergence indicates that *DiffUser* can effectively handle the enlarged state–action space by generating actions through a progressive denoising procedure, which smooths policy updates and prevents instability in high-dimensional environments.

We further test *DiffUser* under dynamic service demands by varying the probability of user service types. As shown in Fig. 10 (b), although the average latency increases with the number of users, medium and high demand shifts reduce latency by 7.4% and 19.1%, respectively. This improvement occurs because demand variations introduce opportunities for more efficient reallocation. When user services change, the system can utilize computational and communication resources more flexibly. *DiffUser*, which learns to generate high-quality allocation samples under such variability, adapts quickly to these changes and maintains low latency even under highly dynamic conditions.

C. Prediction Effectiveness in User Allocation

To evaluate whether location prediction brings benefits to user allocation, we deploy the prediction results obtained in Sec. V-A into the RIS-assisted communication environment described in Sec. V-B. Each predicted coordinate is selected as the *top-1* output of the prediction model, and then mapped into a 1200m × 1200m region containing 8 BSs and 8 RISs. Two allocation schemes are evaluated under the same environment: ***1) With prediction:*** The algorithm generates decisions ahead using predicted locations. ***2) Without prediction:*** The allocation decision is computed during execution, meaning each request must wait for decision generation before transmission, and computation can begin.

Fig. 11 illustrates the performance difference between these two allocation schemes. As shown in Fig. 11 (a), the proposed prediction-enhanced user allocation consistently achieves lower average allocation latency compared with the non-prediction baseline. This reduction comes from the fact that our model provides future location information ahead of execution, allowing the algorithm to complete decision-making before the slot begins. The shaded area visualizes the decision-making latency that is removed from the execution process. This verifies that the accuracy achieved in our prediction module is effective in improving real allocation performance. Fig. 11 (b) further presents the evolution of decision-making latency as the number of users increases. Without prediction, more users introduce a larger search space for allocation and RIS configuration, which slows down decision generation.

VI. Conclusion

CPNs have offered a promising architecture for addressing the computation and communication challenges of mobile services. This work has tackled the UA problem by considering the users' dynamic mobility, weak communication link quality, and high-dimensional decision spaces, and proposed a *CAMPE*-assisted mobility-aware location prediction model and the *DiffUser* algorithm for diffusion-based allocation and RIS phase control. Extensive experiments have demonstrated improvements in terms of prediction accuracy and system latency. Nonetheless, the current framework has assumed that service requests are independent of each other. In practical scenarios, many applications have involved task chains with input–output dependencies, where intermediate results must be transferred to subsequent tasks. Such dependencies, as well as potential task migration costs, have not been modeled in this work and may result in suboptimal performance in workflow-oriented or multi-stage services. Future work will incorporate task-level dependencies and migration-aware optimization to further enhance allocation efficiency.

References

[1] C. Hou, C. Wang, K. Dong, X. Wang, and T. Taleb, "Ris-assisted ad hoc edge for optimal user distribution in service-intensive scenarios," in *IEEE Global Communications Conference, GLOBECOM 2023, Kuala Lumpur, Malaysia, December 4-8, 2023*. IEEE, 2023, pp. 1107–1112.

[2] X. Wang, Z. Ning, L. Guo, S. Guo, X. Gao, and G. Wang, "Online learning for distributed computation offloading in wireless powered mobile edge computing networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 33, no. 8, pp. 1841–1855, 2022.

[3] X. Wang, C. Hou, C. Qiu, X. Ren, Z. Xiong, H. Yao, and D. Niyato, "A resource management strategy for fluid equilibrium in edge-cloud market supporting AIGC services," *IEEE Trans. Serv. Comput.*, vol. 18, no. 4, pp. 1922–1937, 2025.

[4] Y. Luo, Y. Wang, Y. Lei, C. Wang, D. Zhang, and W. Ding, "Decentralized user allocation and dynamic service for multi-uav-enabled MEC system," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1306–1321, 2024.

[5] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between multi-tenancy and interference: A service user allocation game," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 1980–1992, 2022.

[6] C. Wang, H. Yu, X. Li, F. Ma, X. Wang, T. Taleb, and V. C. M. Leung, "Dependency-aware microservice deployment for edge computing: A deep reinforcement learning approach with network representation," *IEEE Trans. Mob. Comput.*, pp. 1–16, 2024.

[7] D. Xu, X. Su, H. Wang, S. Tarkoma, and P. Hui, "Towards risk-averse edge computing with deep reinforcement learning," *IEEE Trans. Mob. Comput.*, pp. 1–18, 2023.

[8] X. Zhang, R. Jia, Q. Yin, Z. Zheng, and M. Li, "Intelligent trajectory design and charging scheduling in wireless rechargeable sensor networks with obstacles," *IEEE Trans. Mob. Comput.*, pp. 1–17, 2024.

[9] X. Zhang, S. Huang, H. Dong, Z. Bao, J. Liu, and X. Yi, "Optimized edge node allocation considering user delay tolerance for cost reduction," *IEEE Trans. Serv. Comput.*, vol. 17, no. 6, pp. 4055–4068, 2024.

[10] J. Zhang, X. Wang, P. Yuan, H. Dong, P. Zhang, and Z. Tari, "Dependency-aware task offloading based on application hit ratio," *IEEE Trans. Serv. Comput.*, vol. 17, no. 6, pp. 3373–3386, 2024.

[11] X. Tang, C. Cao, Y. Wang, S. Zhang, Y. Liu, M. Li, and T. He, "Computing power network: The architecture of convergence of computing and networking towards 6g requirement," *China Communications*, vol. 18, no. 2, pp. 175–185, 2021.

[12] J. Li, H. Lv, B. Lei, and Y. Xie, "A computing power resource modeling approach for computing power network," in *IEEE ICCCN*, 2022, pp. 1–2.

[13] S. Zeng, X. Huang, and D. Li, "Joint communication and computation cooperation in wireless-powered mobile-edge computing networks with NOMA," *IEEE Internet Things J.*, vol. 10, no. 11, pp. 9849–9862, 2023.

[14] W. Sun, Z. Li, Q. Wang, and Y. Zhang, "Fedtar: Task and resource-aware federated learning for wireless computing power networks," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4257–4270, 2023.

[15] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. D. Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 3, pp. 1546–1577, 2021.

[16] C. Sun, W. Ni, Z. Bu, and X. Wang, "Energy minimization for intelligent reflecting surface-assisted mobile edge computing," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 8, pp. 6329–6344, 2022.

[17] B. Zheng, C. You, W. Mei, and R. Zhang, "A survey on channel estimation and practical passive beamforming design for intelligent reflecting surface aided wireless communications," *IEEE Commun. Surv. Tutorials*, vol. 24, no. 2, pp. 1035–1071, 2022.

[18] Y. Yang, Y. Gong, and Y. Wu, "Intelligent-reflecting-surface-aided mobile edge computing with binary offloading: Energy minimization for iot devices," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12 973–12 983, 2022.

[19] S. Zhuang, Y. He, F. R. Yu, C. Gao, W. Pan, and Z. Ming, "When multi-access edge computing meets multi-area intelligent reflecting surface: A multi-agent reinforcement learning approach," in *IEEE/ACM IWQoS*, 2022, pp. 1–10.

[20] L. Li, W. Guan, C. Zhao, Y. Su, and J. Huo, "Trajectory planning, phase shift design, and iot devices association in flying-ris-assisted mobile edge computing," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 147–157, 2024.

[21] G. Cui, Q. He, X. Xia, F. Chen, F. Dong, H. Jin, and Y. Yang, "OL-EUA: online user allocation for noma-based mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 22, no. 4, pp. 2295–2306, 2023.

[22] C. Wu, T. Chiu, C. Wang, and A. Pang, "Mobility-aware deep reinforcement learning with glimpse mobility prediction in edge computing," in *IEEE ICC*, 2020, pp. 1–7.

[23] Y. Chen, S. Zhang, Y. Jin, Z. Qian, M. Xiao, J. Ge, and S. Lu, "LOCUS: user-perceived delay-aware service placement and user allocation in MEC environment," *IEEE Trans. Parallel Distributed Syst.*, vol. 33, no. 7, pp. 1581–1592, 2022.

[24] P. Lai, Q. He, X. Xia, F. Chen, M. Abdelrazek, J. C. Grundy, J. G. Hosking, and Y. Yang, "Dynamic user allocation in stochastic mobile edge computing systems," *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 2699–2712, 2022.

[25] H. Jin, P. Zhang, H. Dong, X. Wei, Y. Zhu, and T. Gu, "Mobility-aware and privacy-protecting qos optimization in mobile edge networks," *IEEE Trans. Mob. Comput.*, vol. 23, no. 2, pp. 1169–1185, 2024.

[26] Z. Chen, H. Yao, L. Gu, D. Zeng, and K. Zheng, "Dynamic service migration via approximate markov decision process in mobile edge-clouds," in *IDCS*, ser. Lecture Notes in Computer Science, G. Fortino, A. B. M. S. Ali, M. Pathan, A. Guerrieri, and G. D. Fatta, Eds., vol. 10794. Springer, 2017, pp. 13–24.

[27] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," in *ACM WWW*, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds., 2018, pp. 1459–1468.

[28] G. Sun, H. Qi, Y. Shen, and B. Yin, "Tcsa-net: A temporal-context-based self-attention network for next location prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20 735–20 745, 2022.

[29] C. Wu, T. Chiu, C. Wang, and A. Pang, "Mobility-aware deep reinforcement learning with seq2seq mobility prediction for offloading and allocation in edge computing," *IEEE Trans. Mob. Comput.*, vol. 23, no. 6, pp. 6803–6819, 2024.

[30] P. Lai, Q. He, X. Xia, F. Chen, M. Abdelrazek, J. C. Grundy, J. G. Hosking, and Y. Yang, "Dynamic user allocation in stochastic mobile edge computing systems," *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 2699–2712, 2022.

[31] J. Cao, X. Song, and F. Si, "Energy efficiency resource allocation for cell-edge users with social-aware based grouping D2D," *ICT Express*, vol. 9, no. 5, pp. 915–920, 2023.

[32] G. Cui, Q. He, F. Chen, Y. Zhang, H. Jin, and Y. Yang, "Interference-aware game-theoretic device allocation for mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 21, no. 11, pp. 4001–4012, 2022.

[33] E. Liu, L. Zheng, Q. He, P. Lai, B. Xu, and G. Zhang, "Role-based user allocation driven by criticality in edge computing," *IEEE Trans. Serv. Comput.*, vol. 16, no. 5, pp. 3636–3650, 2023.

[34] B. Ji, Y. Wang, K. Song, C. Li, H. Wen, V. G. Menon, and S. Mumtaz, "A survey of computational intelligence for 6g: Key technologies, applications and trends," *IEEE Trans. Ind. Informatics*, vol. 17, no. 10, pp. 7145–7154, 2021.

[35] B. Hazarika, K. Singh, S. Biswas, S. Mumtaz, and C. Li, "Multi-agent drl-based task offloading in multiple ris-aided iov networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1175–1190, 2024.

[36] Q. Wu and R. Zhang, "Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1838–1851, 2020.

[37] S. Abeywickrama, R. Zhang, Q. Wu, and C. Yuen, "Intelligent reflecting surface: Practical phase shift model and beamforming optimization," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5849–5863, 2020.

[38] 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.

[39] X. Wei, D. Shen, and L. Dai, "Channel estimation for ris assisted wireless communications—part i: Fundamentals, solutions, and future opportunities," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1398–1402, 2021.

[40] G. Zhou, C. Pan, H. Ren, P. Popovski, and A. L. Swindlehurst, "Channel estimation for ris-aided multiuser millimeter-wave systems," *IEEE Trans. Signal Process.*, vol. 70, pp. 1478–1492, 2022.

[41] K. Dong, S. A. Vorobyov, H. Yu, and T. Taleb, "Beamforming design for integrated sensing, computation over-the-air, and communication in internet of robotic things," *IEEE Internet Things J.*, 2024.

[42] A. M. Elbir, K. V. Mishra, S. A. Vorobyov, and R. W. Heath, "Twenty-five years of advances in beamforming: From convex and nonconvex optimization to learning techniques," *IEEE Signal Process. Mag.*, vol. 40, no. 4, pp. 118–131, 2023.

[43] Y. Hong, H. Martin, and M. Raubal, "How do you go where?: improving next location prediction by learning travel mode information using transformers," in *SIGSPATIAL 2022*, M. Renz and M. Sarwat, Eds. ACM, 2022, pp. 61:1–61:10.

[44] B. B. Moser, A. S. Shanbhag, F. Raue, S. Frolov, S. Palacio, and A. Dengel, "Diffusion models, image super-resolution and everything: A survey," *CoRR*, vol. abs/2401.00736, 2024.

[45] H. Du, Z. Li, D. Niyato, J. Kang, Z. Xiong, H. Huang, and S. Mao, "Diffusion-based reinforcement learning for edge-enabled ai-generated content services," *IEEE Trans. Mob. Comput.*, pp. 1–16, 2024.

[46] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html

[47] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li, *Geolife GPS trajectory dataset - User Guide*, geolife gps trajectories 1.1 ed., July 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/
[48] L. Gong, Y. Lin, X. Zhang, Y. Lu, X. Han, Y. Liu, S. Guo, Y. Lin, and H. Wan, "Mobility-llm: Learning visiting intentions and travel preference from human mobility data with large language models," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 36 185–36 217.
[49] A. Solomon, A. Livne, G. Katz, B. Shapira, and L. Rokach, "Analyzing movement predictability using human attributes and behavioral patterns," *Comput. Environ. Urban Syst.*, vol. 87, p. 101596, 2021.
[50] F. Li, Z. Gui, Z. Zhang, D. Peng, S. Tian, K. Yuan, Y. Sun, H. Wu, J. Gong, and Y. Lei, "A hierarchical temporal attention-based LSTM encoder-decoder model for individual mobility prediction," *Neurocomputing*, vol. 403, pp. 153–166, 2020.
[51] N. Chaurasia, M. Kumar, A. Vidyarthi, K. Pal, and A. Alkhayyat, "An efficient and optimized markov chain-based prediction for server consolidation in cloud environment," *Comput. Electr. Eng.*, vol. 108, p. 108707, 2023.
[52] P. Lai *et al.*, "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *Proc. ICSOC*, Hangzhou, China, Nov. 2018, pp. 230–245.

Xiaofei Wang (Senior Member, IEEE) is currently a Professor with the Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, China. He got master and doctor degrees in Seoul National University from 2006 to 2013, and was a Post-Doctoral Fellow with The University of British Columbia from 2014 to 2016. Focusing on the research of social-aware cloud computing, cooperative cell caching, and mobile traffic offloading, he has authored over 100 technical papers in the IEEE TCC, IEEE JSAC, the IEEE TWC, the IEEE WIRELESS COMMUNICATIONS, the IEEE COMMUNICATIONS, the IEEE TMM, the IEEE INFOCOM, and the IEEE SECON.

Chenxuan Hou received her B.S. degree in electronic information engineering from Southwest University, Chongqing, China, in 2020, and her M.S. degree in communications and signal processing from the University of Manchester, UK, in 2021. She is currently pursuing the Ph.D. degree in the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her current research interests include computing power networks and edge intelligence.

Chao Qiu (Member, IEEE) received her B.S. degree in communication engineering from China Agricultural University, in 2013, and her Ph.D. in information and communication engineering from the Beijing University of Posts and Telecommunications, in 2019. She is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, China. From September 2017 to September 2018, she visited Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. Her current research interests include edge computing, edge intelligence, and blockchain.

Chenyang Wang (Member, IEEE) received B.S. and M.S. degrees in computer science and technology from Henan Normal University, Xinxiang, China, in 2013 and 2017, respectively. He received his Ph.D. degree in 2023 from the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China. He is also a visiting PhD student under the support of the China Scholarship Council (CSC) at the School of Electrical Engineering, Aalto University in 2021. He is currently a postal researcher at the College of Computer Science and Software Engineering, Shenzhen University. His current research interests include edge intelligence, large models, active inference, and deep learning. He received the Best Student Paper Award of the 24th International Conference on Parallel and Distributed Systems by IEEE Computer Society in 2018. He also received the Best Paper Award from the IEEE International Conference on Communications in 2021. In 2022, he received the "IEEE ComSoc Asia-Pacific Outstanding Paper Award".

Kai Dong received the Ph.D. degrees in Information Technology from Politecnico di Milano, Milan, Italy, in 2023. He is currently a Postdoctoral researcher with the Center of Wireless Communications (CWC), University of Oulu, Finland, and a visiting researcher with Department of Information and Communications Engineering (DICE), Aalto university, Finland. His research interests include integrated sensing, computing and communication, deterministic wireless communications, advanced relaying technologies in wireless, optimization algorithms in signal processing.

Tarik Taleb (Senior Member, IEEE) received the B.E. degree (with distinction) in information engineering and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Full Professor at Ruhr University Bochum, Germany. He was a Professor with the Center of Wireless Communications (CWC), University of Oulu, Oulu, Finland. He is the founder of ICTFICIAL Oy, and the founder and the Director of the MOSA!C Lab, Espoo, Finland. From October 2014 to December 2021, he was an Associate Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. Prior to that, he was working as a Senior Researcher and a 3GPP Standards Expert with NEC Europe Ltd., Heidelberg, Germany. Before joining NEC and till March 2009, he worked as an Assistant Professor with the Graduate School of Information Sciences, Tohoku University, in a lab fully funded by KDDI. From 2005 to 2006, he was a Research Fellow with the Intelligent Cosmos Research Institute, Sendai. He has been directly engaged in the development and standardization of the Evolved Packet System as a member of the 3GPP System Architecture Working Group. His current research interests include AI-based network management, architectural enhancements to mobile core networks, network softwarization and slicing, mobile cloud networking, SDN/NFV, software-defined security, and mobile multimedia streaming.