

# Adaptive Edge Association for Wireless Digital Twin Networks in 6G

Yunlong Lu, *Member, IEEE*, Sabita Maharjan, *Senior Member, IEEE*, and Yan Zhang, *Fellow, IEEE*

**Abstract**—6G is envisioned to be characterized by ubiquitous connectivity, extremely low latency, and enhanced edge intelligence. However, enriching 6G with these features requires addressing new, unique and complex challenges specifically at the edge of the network. In this paper, we propose a wireless digital twin edge network model by integrating digital twin with edge networks to enable new functionalities such as hyper-connected experience and low-latency edge computing. To efficiently construct and maintain digital twins in the wireless digital twin network, we formulate the edge association problem with respect to the dynamic network states and varying network topology. Furthermore, according to the different running stages, we decompose the problem into two sub-problems, including digital twin placement and digital twin migration. Moreover, we develop a Deep Reinforcement Learning (DRL) based algorithm to find the optimal solution to the digital twin placement problem, and then use transfer learning to solve the digital twin migration problem. Numerical results show that the proposed scheme provides reduced system cost and enhanced convergence rate for dynamic network states.

**Index Terms**—Digital twin, Wireless network, Edge association, Deep reinforcement learning, Transfer learning

## I. INTRODUCTION

With the rapid development of wireless technologies such as the Internet of Things (IoT) and widely deployed fifth-generation (5G) networks, the envisioning and planning for sixth-generation mobile networks (6G) [1] has already begun. It is believed that 6G will provide extremely high data rates, low latency, and improved edge intelligence [2], [3]. According to the 6G white paper released by Samsung, 6G is expected to bring the next hyper-connected experience to every corner of our life [4]. Also, the vision of China Mobile Research Institute for 6G is to realize ubiquitous intelligence and to fully connect all things on the basis of 5G and other techniques such as artificial intelligence [5].

To meet these requirements, the quality of services and experience of applications in 6G networks need to be significantly enhanced. However, hundreds of billions of end devices

This work was partially supported by the the Fundamental Research Funds for the Central Universities 2021RC233, and partially supported by the State Key Laboratory of Rail Traffic Control and SafetyContract No. RCS2021ZQ003, Beijing Jiaotong University.(Corresponding author: Y. Zhang)

Yunlong Lu is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail:yunlong.lu@ieee.org).

Sabita Maharjan and Yan Zhang are with the Department of Informatics, University of Oslo, 0316 Oslo, Norway, and also with the Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (e-mail: sabita@ifi.uio.no; yanzhang@ieee.org).

Copyright (c) 2021 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.

will be connected to the 6G network, and a huge amount of data will be generated from these devices. The massive amount of data requires tremendous computation and communication resources to be provided by edge servers. Thus the huge gap between users' requirements and what the edge servers can provide, is a major challenge for 6G systems in providing high-quality services for emerging and new applications.

Digital twin emerges as a promising technology to bridge the connection gap between physical spaces and digital systems [6]. The digital replicas of physical entities such as devices, machines, and physical objects are constructed at the server based on historical data and real-time running status. Digital twin enables close monitoring, real-time interaction, and reliable communication between the digital space and the physical systems, which can in turn optimize the running of the physical systems. With these advanced capabilities, digital twin is believed to be one of the most important enabling technologies for 6G. Several works have explored utilizing digital twins to enhance the performance of wireless communications, for applications such as computation offloading, content caching, and resource sharing [7]–[9]. In [7], the authors proposed a distributed framework based on federated learning and digital twin to execute edge computing in industrial IoT. In [8], the authors proposed to construct digital twins in edge networks by applying blockchain and federated learning. In [9], the computation offloading problem in digital twin edge networks was formulated and the optimal offloading policy was obtained based on deep reinforcement learning.

Although several applications have been preliminarily explored, the huge number of connected devices and the heterogeneous network structure is still a challenge for the application of digital twin in 6G networks. With delay requirements much stricter than in 5G, achieving the required level of computation and communication latency is one of the biggest challenges for 6G applications. In addition, more heterogeneous networks with dynamic network states lead bring in additional dimensions and thus require finding solutions to even more complex resource optimization problems. Further investigation and more focused research in applying digital twin in edge networks is therefore of utmost importance to alleviate the huge gap between high service quality requirements and the limited edge resources.

The integration of digital twin with Mobile Edge Computing (MEC) [10] opens up new possibilities for mitigating the resource limitation issues in applying digital twins in 6G networks. MEC- one of the key enabling technologies for 6G, can considerably reduce the system latency by executing

computation at the edge of the network. MEC has been widely investigated for accomplishing edge intelligence tasks such as computation offloading, content caching, and data sharing. In [11], the authors proposed an edge intelligence scheme in the Industrial Internet of Things (IIoT) framework based on MEC techniques. In [12], the authors proposed a Deep Reinforcement Learning (DRL) based intelligent content caching scheme for an edge computing framework.

In the convergence of digital twins and edge computing, the placement and maintenance of digital twins in edge networks is a fundamental problem that needs to be further investigated. The dynamic network states such as the channel states and available computation resources may hinder the digital twins from improving the instant application performance. In this context, the mapping relations between end devices and digital twins in the edge server should be carefully designed. Moreover, the mobility of end users makes it less practical for a digital twin server to keep real-time interaction with the mobile users that can affect provisioning of continuous services to the users. To cope with these issues, in this paper, we first propose a digital twin empowered edge network model by incorporating digital twin into MEC systems. Then, we formulate a new edge association problem by placing digital twins in the dynamic network and migrating digital twins of mobile end users between edge servers. Finally, we find the solution to the formulated edge association problem by leveraging DRL for digital twin placement and transfer learning for digital twin migration. The main contributions of this paper are summarized as follows.

- We design a wireless digital twin network model for 6G networks, exploiting digital twins to mitigate the unreliable and long-distance communication between end users and edge servers.
- We formulate the edge association problem for placing and migrating digital twins of users in the edge servers according to the dynamic network states and mobile end users, to reduce the average system latency and to improve user utility in the digital twin empowered edge network.
- We develop a deep reinforcement learning based algorithm to find the optimal solution to the formulated problem, by jointly considering the digital twin placement strategy and corresponding system latency. In addition, we propose the digital twin migration method based on transfer learning to deal with the mobility of users.

The rest of this paper is organized as follows. We review the related work in Section II. In Section III, we present the digital twin empowered edge network model. In Section IV, we formulate a new edge association problem for digital twin placement and migration, to minimize the average system latency while also addressing the mobility of end users. In Section V, we design a deep reinforcement learning based algorithm for digital twin placement and a transfer learning based algorithm for digital twin migration. Numerical results are presented in Section VI. Finally, Section VII concludes the paper.

## II. RELATED WORK

While 5G is still in the process of wide scale deployment, we observe significant work [13]–[15] on 6G vision and huge research interest in solving technological challenges that can enable the 6G paradigm. In [1], the authors suggested that human-centric mobile communications would be one of the most important characteristics of 6G networks, together with high security, secrecy, and privacy. Wireless communications in 6G can also be improved through machine learning techniques. In [16], the authors provided an overview of how machine learning will impact wireless communications in 6G, including how to solve problems in the wireless domain and how to optimize the wireless networks. Although still in its early stage, it is generally believed that 6G networks will be featured with extremely low latency, highly intelligent connectivity, and enhanced security.

MEC plays an important role in providing latency-sensitive services for wireless communications by executing computation and storage tasks at the edge. The application of MEC in achieving edge intelligence has been widely studied. A series of emerging technologies such as blockchain [11], [17], [18] and deep reinforcement learning [12] have also been exploited in MEC systems for optimal network resource allocation. In [18], the authors proposed a blockchain empowered federated learning scheme for data sharing in vehicular networks. The proposed scheme adopted permissioned blockchain at the aggregation server to distribute and store model parameters generated in the federated learning process. In [19], the authors studied the joint task offloading and resource allocation problem, to maximize the task offloading gain for users. Convex and quasi-convex optimization techniques are used to solve the problem to reduce their task completion time and energy consumption. However, in the era of 6G, the pressing demand for reduced latency, improved intelligence, increased connections raise new challenges for MEC systems.

To fulfill the requirements of 6G communications, deep reinforcement learning [20] and digital twin [21], [22] will be two promising enabling technologies for MEC systems. DRL has been investigated comprehensively for edge network optimization, in areas of content caching, computation offloading, and resource allocation. For example, in edge caching, the authors in [23] formulated a constrained minimization of the aggregate cost problem considering the time-varying fetching and caching costs. Since the caching decisions in one slot will affect the available content in the next slots, the authors proposed to use deep Q-learning to find optimal fetch-cache decisions. In [24], the authors proposed to use deep Q-learning to maximize the utility function to achieve optimal transmit power scheduling. While DRL has been widely used in network optimization for MEC systems, the huge gap of running states between physical entities and MEC servers still exists, which may be a potential hindrance for real time optimization of the dynamic edge network.

The emergence of digital twin opens up new possibilities for applying MEC systems in 6G networks with respect to the performance requirements for wireless communication in 6G networks. As the replica of physical systems, digital twin

bridges the gap between physical systems and digital spaces [6], [25]. As one of the most critical enabling technologies for 6G communications, digital twin has been studied in several advanced works for improving the performance of MEC systems in edge networks [26]–[28]. In [26], the authors used digital twin to capture the dynamic network states and used the Unmanned Aerial Vehicle (UAV) driven federated learning for air-ground networks. In [27], the authors proposed the model of digital twin edge networks, and developed a communication-efficient federated learning for improving the running efficiency of their proposed model. Furthermore, due to the lack of mutual trust between IoT devices, the authors in [29] further integrated blockchain with digital twins to enhance system security and user privacy.

We note that some works have explored the use of digital twin as a key enabling paradigm for improving the performance of MEC systems. However, few works can be found focusing on the construction and maintenance mechanism of digital twins in edge networks. How to ensure continuous interactions between digital twins and end users considering the mobility of end users and the dynamic network states, remains a largely unexplored problem, and thus requires in-depth investigation. Motivated by such considerations, to address this gap, in this paper, we propose a wireless digital twin network framework, and formulate the edge association problem considering the placement and migration of digital twins. We then design the DRL and transfer learning based algorithms to find the optimal solution to the edge association problem in order to meet the latency requirements of 6G networks.

### III. SYSTEM MODEL

In this section, we first introduce the architecture of our wireless digital twin network model. Then we further provide the details of the system model, and formulate the digital twin edge association problem, respectively.

#### A. Digital Twin Empowered Network Model

We consider a digital twin empowered edge network as shown in Fig. 1. There are three layers in our wireless digital twin network system: radio access layer (i.e., end layer), digital twin layer (i.e., edge layer), and cloud layer. The radio access layer consists of entities such as mobile devices and vehicles that have limited computing and storage resources. Through wireless communications, these entities connect to the base stations and request for services provided by network operators. In the digital twin layer, some base stations are equipped with MEC servers to execute computation tasks, while other base stations provide wireless communication services to end users. The digital twins of physical entities are modeled and maintained by the MEC servers. Since the number of entities in the physical layer is much larger than the number of MEC servers in the digital twin layer, an MEC server may maintain multiple digital twins of physical entities. In the cloud layer, cloud servers are equipped with a large amount of computing and storage resources. Tasks that

are computation-sensitive or require global analysis can be executed in the cloud layer.

Since digital twins reproduce the running of physical entities, maintaining the digital twins of massive devices consumes a large number of resources including computing resources, communication resources, and storage resources. To relieve the resource limitation in the edge layer, we model digital twins as of two types: device digital twin and service digital twin. The device digital twin is a full replica of physical devices that includes the information of hardware configuration, historical running data, and real-time states. For user  $u_i$ , its device digital twin can be expressed as

$$DT^f(u_i) = \Theta(\mathcal{D}_i, S_i(t), \mathcal{M}_i, \Delta S_i(t+1)), \quad (1)$$

where  $\mathcal{D}_i$  is the historical data of user device  $i$  such as the configuration data and historical running data.  $S_i(t)$  represents the running state of device  $i$  that consists of  $r_1$  dimensions, and varies with time, which can be denoted as  $S(t) = \{s_i^1(t), s_i^2(t), \dots, s_i^{r_1}(t)\}$ .  $\mathcal{M}_i$  is the set of behavior model of  $u_i$  that consists of  $r_2$  behavior dimensions,  $\mathcal{M}_i = \{m_i^1, m_i^2, \dots, m_i^{r_2}\}$ .  $\Delta S_i(t+1)$  is the state update of  $S_i(t)$  in time slot  $t+1$ . Taking meteorological IoT device as an example,  $S(t)$  can be temperature, humidity, wind speed location and so on. The behavior models  $\mathcal{M}_i$  may consist of the variation models of the temperature, humidity, and wind speed. In this paper, we mainly focus on the scenarios of device digital twins to conduct our study.

Different from device digital twins, the service digital twin is a lightweight digital replica constructed by extracting the running states of several devices towards a specific application. Similar to Eq. (1), the service digital twin can be expressed as

$$DT(u_i, \zeta) = \Theta(\mathcal{D}_i(\zeta), S_i^\zeta(t), \mathcal{M}_i^\zeta, \Delta S_i^\zeta(t+1)), \quad (2)$$

where  $\zeta$  is the target service.  $\mathcal{D}_i(\zeta)$ ,  $S_i^\zeta(t)$ ,  $\mathcal{M}_i^\zeta$ , and  $\Delta S_i^\zeta(t+1)$  are corresponding terms related to the target service  $\zeta$ . For example, vehicles driving in the same region can be modeled into a specific service digital twin for supporting autonomous driving on a particular stretch of the road. In such a case, the service digital twin for autonomous driving only collects driving information of these vehicles and analyzes their driving behavior to guide the moving vehicles. Depending on the required scale, service digital twins can be constructed in the edge server or the cloud server.

#### B. Communication and Computation Model

As shown in Fig. 3, the communication between end users and edge servers contains the uplink communication for transmitting data from user devices to edge servers and the downlink communication for sending back results from edge servers to user devices. Note that the size of results returning to users is much smaller than the updated data, we only consider the uplink communication latency in our communication model [30]. The maximum achievable uplink data  $r_{ij}$  between user  $i$  and base station  $j$  is given as:

$$r_{ij} = W \log\left(1 + \frac{p_{ij} h_{ij}}{WN_0}\right), \quad (3)$$

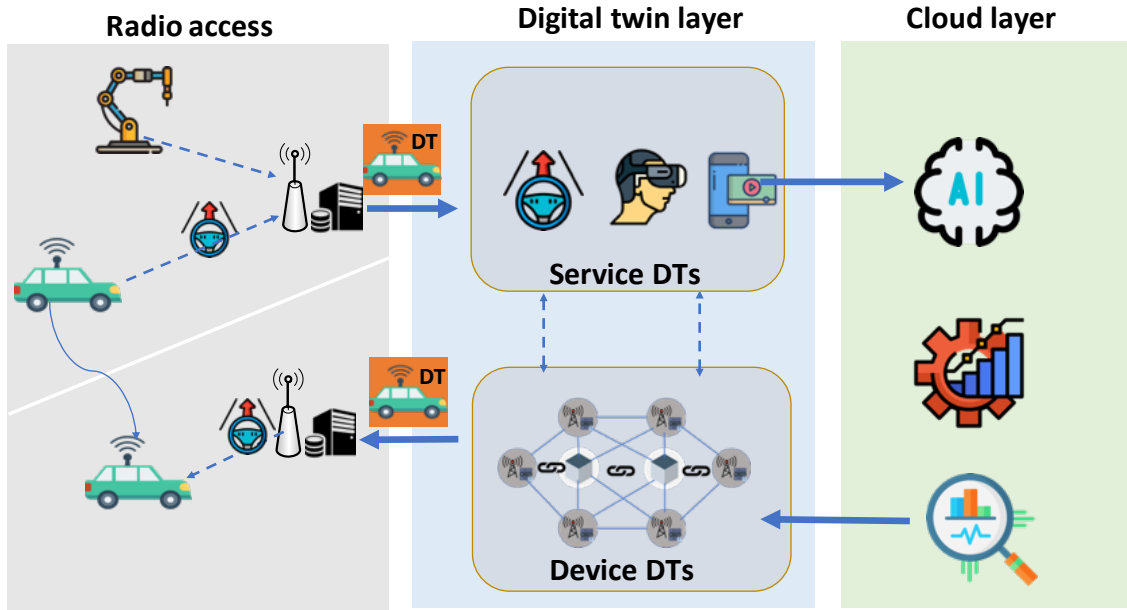


Fig. 1: The architecture of wireless digital twin networks

where  $h_{ij}$  denotes the channel power gain of user  $i$ ,  $p_{ij}$  denotes the corresponding transmission power for user  $i$ ,  $N_0$  is the noise power spectral density, and  $W$  is the channel bandwidth. The transmission latency for uploading  $D_i$  from user  $i$  to base station  $j$  can be expressed as

$$T_{ij}^{com} = \frac{D_i}{r_{ij}}. \quad (4)$$

The wired transmission latency between base stations is highly correlated to the transmission distance. Let  $\phi$  be the latency required for transmitting one unit data in each unit distance. Then the wired transmission latency can be written as

$$T_{j_1 j_2}^{com} = \phi \cdot D_j \cdot d(j_1, j_2), \quad (5)$$

where  $D_j$  is the size of transmitted data, and  $d(j_1, j_2)$  is the distance between base station  $j_1$  and  $j_2$ .

Denote the total computation resource of edge server  $j$  as  $F_j$ . The computation resources of edge server  $j$  may be allocated to multiple user devices for maintaining their digital twins in server  $j$ . Let  $f_{ij}$  denote the computation resource assigned to the digital twin of user  $i$ . Then the time to execute tasks from user  $i$  can be expressed as

$$T_{ij}^{cmp} = \frac{D_i}{f_{ij}}, \quad (6)$$

where  $D_i$  is the size of computation task from user  $i$ ,  $\sum_{i=1}^N x_{ij} f_{ij} \leq F_j$ , and  $x_{ij} = 1$  if  $f_{ij} > 0$ . Otherwise,  $x_{ij} = 0$ .

### C. System Latency Model

The latency of maintaining a digital twin mainly consists of two parts: the construction delay and the synchronization delay. Fig. 2 shows the complete process to construct a digital twin of user  $u_i$ . At the beginning, the running data

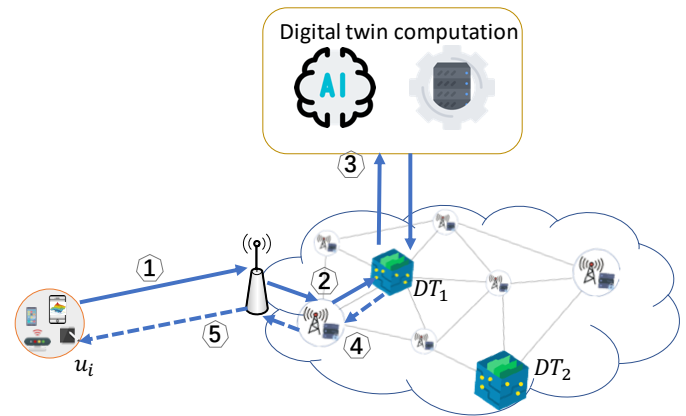


Fig. 2: The process of constructing a digital twin

$D_i$  of  $u_i$  is transmitted to its nearby base station through wireless communication. Then its nearby base station transmits the running data  $D_i$  to the digital twin server  $DT_1$  that is responsible for constructing and maintaining the digital twin of  $u_i$ , through wired communication. The digital twin server  $DT_i$  runs the computation to process and analyze the received data, and build a digital twin model for user  $u_i$ , as given by Eq. (1). During the digital twin computation process, artificial intelligence related algorithms are used for extracting the data features and training the digital twin model. Finally, the results of the digital twin model are transmitted back to user  $u_i$  through wired and wireless communications. The feedback results provide  $u_i$  with insights for improving its service quality or running efficiency for specific applications. The system latency consists of the following items.

- 1) *Wireless data transmission*: In the construction phase of  $DT(u_i)$ , the historical running data of user  $i$  requires to be transmitted to its digital twin server through its nearby

base station. Let  $D_i$  denote the size of historical data to be transmitted, the wireless communication latency  $T_{ij}^{com}$  from user  $i$  to its base station  $j$  can be calculated according to Eq. (4).

- 2) *Wired data transmission*: The wired transmission time from the nearby base station of  $u_i$  to its digital twin server  $k$  is

$$T_{jk}^{com} = \phi \cdot D_i \cdot d(j, k). \quad (7)$$

Thus the total communication time for transmitting historical data of  $u_i$  to its digital twin server is

$$T_{ik}^{com} = T_{ij}^{com} + T_{jk}^{com}. \quad (8)$$

- 3) *Digital twin data computation*: The computation time at digital twin server  $k$  is

$$T_{ik}^{cmp} = \frac{D_i}{f_{ij}}. \quad (9)$$

The total latency for constructing the digital twin of user  $i$  is

$$T_{ik}^{ini} = T_{ik}^{com} + T_{jk}^{com} + T_{ik}^{cmp}. \quad (10)$$

The digital twin of user  $i$ , that is  $DT(u_i)$ , is constructed in its digital twin server  $DT_k$ . Then,  $DT(u_i)$  requires to interact with  $u_i$  constantly to keep consistent with the running states of  $u_i$ . Denote the size of updated data as  $\Delta D_i$ , the latency for one update can be given as

$$T_{ik}^{upd} = \frac{\Delta D_i}{r_{ij}} + \phi \cdot \Delta D_i \cdot d(j, k) + \frac{\Delta D_i}{f_{ij}}. \quad (11)$$

The synchronization latency in one unit time slot can be written as

$$T_{ik}^{syn} = \frac{1}{\Delta t} T_{ik}^{upd}, \quad (12)$$

where  $\Delta t$  denotes the time gap between every two updates.

#### D. Adaptive Edge Association

Due to intensive resource consumption for maintaining digital twins, constructing digital twins in each MEC server can incur a considerable increase in transmission load, computation overhead, and energy consumption. To address this issue, in our digital twin networks, instead of maintaining digital twins in every base station, we select a subset of base stations as the digital twin servers to maintain the digital twins with reduced time cost and energy consumption. In our network model, user device  $u_i \in U$  belongs to base station  $b_j$ , where  $b_j \in B$ . The digital twin of user  $u_i$ , denoted as  $DT(u_i)$ , is constructed in the digital twin server  $s_k$ , where  $s_k \in S$ .

Two fundamental problems require to be addressed in our digital twin empowered networks. The first one is digital twin placement. For constructing digital twins in our system, in the digital twin placement phase, we select the optimized subset of base stations as digital twin servers, that is,  $S \subset B$ . Fig. 2 depicts the digital twin placement in the edge network. Two servers  $DT_1$  and  $DT_2$  are selected as the digital twin servers to construct and maintain the digital twins of user devices. The digital twin servers also cooperate with each other through the cloud server. The second issue is the migration of digital twins. The mobility of devices including mobile

devices and vehicles raises new challenges to the construction and maintenance of digital twins in edge networks, and may thus hinder efficient and effective service provisioning from edge servers to user devices. Fig. 3 depicts the migration process in our digital twin networks. At first, digital twin server  $DT_1$  maintains the digital twin of user  $u$  at location  $A$ . However, when user  $u$  moves to location  $B$  and location  $C$ , maintaining  $DT(u)$  will increase the communication resource and interaction latency due to the long transmission distance. In such a case,  $DT_2$  should take over user  $u$  for maintaining its digital twin  $DT(u)$ .  $DT(u_A)$  should be migrated from  $DT_1$  to  $DT_2$  for continuously maintaining  $DT(u)$  and providing further services to user  $u$ .

Thus, to cope with the dynamic network and to provide real-time high-quality service to users, the digital twins need to be incorporated into the edge network in two phases: placement and migration. In the placement phase, digital twin servers are selected from the base stations to construct and maintain digital twins of end users. In the migration phase, the digital twins of users are migrated between digital twin servers for reducing transmission overhead and communication latency.

The main notations used in this paper are summarized in Table II.

TABLE I: Summary of main notations

$u_i$	The $i$ -th user
$U$	The set of users
$DT(u_i)$	The digital twin server of user $u_i$
$S_i(t)$	The running states of user $u_i$ at time slot $t$
$r_1$	The running state dimension of digital twins
$r_2$	The behavior dimension of digital twins
$\mathcal{M}_i$	The behavior model of $u_i$
$\Delta S_i(t)$	The update states of user $u_i$ at time slot $t$
$D_i$	The running data set of user $u_i$
$b_j$	The $j$ -th base station
$B$	The set of base stations
$W$	The bandwidth of the transmission channel
$r_{ij}$	The achievable data rate between $u_i$ and $b_j$
$p_i$	The transmission power of $u_i$
$T_{ij}^{com}$	The time cost for communication between $u_i$ and $b_j$
$T_{j_1 j_2}^{com}$	The transmission time cost between $b_{j_1}$ and $b_{j_2}$
$T_{ij}^{cmp}$	The computation time of tasks from $u_i$ in server $j$
$G(\cdot)$	The network model
$A$	The association matrix
$pl$	The digital twin placement strategies
$S(t)$	The DRL state at time slot $t$
$\mathcal{A}(t)$	The DRL action at time slot $t$
$\mathcal{R}(t)$	The DRL reward at time slot $t$

#### IV. DIGITAL TWIN EDGE ASSOCIATION: PROBLEM FORMULATION

In this section, we formulate the digital twin edge association problem incorporating the system cost for digital twin construction and maintenance. The placement of digital twins should carefully consider the latency performance and energy consumption of the system. The edge association in our system model is an adaptive process that changes with the dynamic network states such as the varying channel states and network topology. The overall system latency function can be written as

$$T_{sys} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M (T_{ik}^{ini} + \eta T_{ik}^{syn}), \quad (13)$$

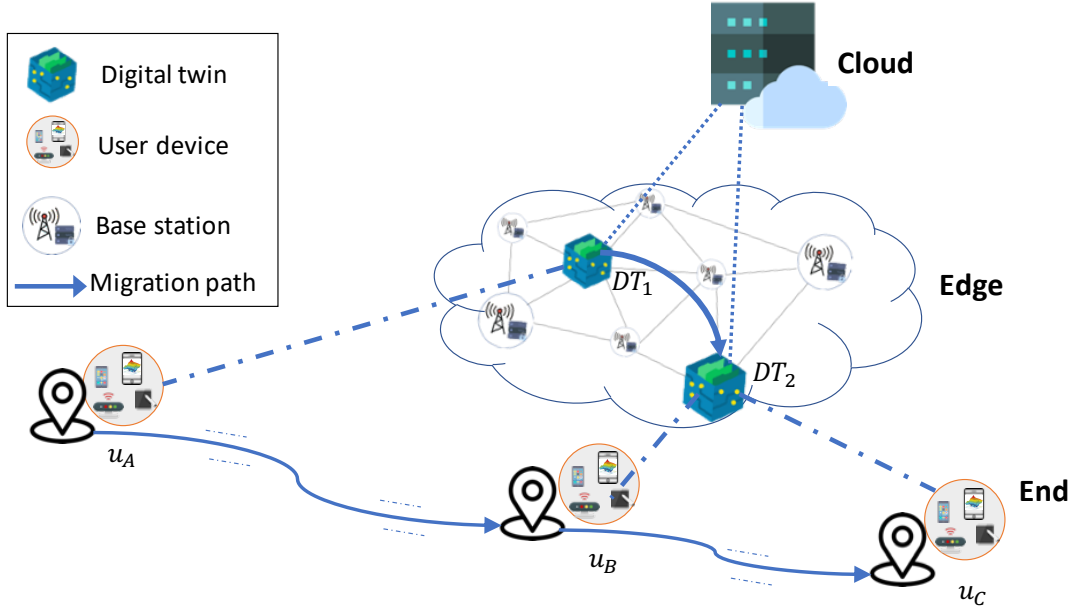


Fig. 3: The digital twin migration process in edge association

where  $\eta$  is the weighting coefficient. The objective of the edge association problem is to minimize the average system latency, and to provide service for each user  $u_i$  within its delay requirement, given as

$$\min T_{sys} \quad (14)$$

$$\text{s.t. } t_{u_i} \leq \tau_{u_i}, \forall i \in \mathcal{N}, \quad (15)$$

where  $t_{u_i}$  is the service interaction latency between end users and their corresponding digital twins as in Eq. (10),  $\tau_{u_i}$  is the latency constraint of user  $u_i$ .

According to the running phases of digital twins, the digital twin edge association consists of two sub-problems, that is, the digital twin placement problem and the digital twin migration problem. The digital twin placement is the first step for digital twin construction. The goal of the placement problem is to find the optimal position to locate digital twin servers and to decide the association strategies between servers and end users, based on the instant system states. In the second phase, the digital twin migration problem is to adjust the placement positions and the association relations of digital twins, to adapt to the dynamic network states. In this phase, the digital twins of end users need to be migrated between digital twin servers to ensure low-latency interactions with moving end users.

#### A. Digital Twin Placement Problem

We use  $G(U, B, DT, E)$  to describe our digital twin empowered network, where  $U$  is the set of end users,  $B$  is the set of base stations,  $DT$  is the set of digital twin servers, and  $E$  is the set of physical links between base stations. The capacity of digital twin server  $DT_k$ , that is,  $N_k$ , denotes the maximum number of digital twins  $DT_k$  can maintain.

We use weight matrix  $A = [a_{ik}]$  to represent the association relations between the user devices and the digital twin servers, where  $a_{ik} = 1$  if the digital twin of user  $i$  is maintained by

digital twin server  $k$ . Otherwise,  $a_{ik} = 0$ . For example, in Fig 2, since the digital twin of  $u_i$  is maintained by  $DT_1$ , thus we have  $a_{i1} = 1$ , and  $a_{i2} = 0$ . The association matrix takes the form

$$\begin{bmatrix} a_{11} & a_{12} & a_{1k} & \dots & a_{1M} \\ a_{21} & a_{22} & a_{2k} & \dots & a_{2M} \\ a_{21} & a_{22} & a_{2k} & \dots & a_{2M} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ a_{N1} & a_{N2} & a_{Nk} & \dots & a_{NM} \end{bmatrix}$$

We use  $pl$  to denote the placement policy of digital twin servers, which is given by

$$pl = \{loc_k | k \in [1, M]\}, \quad (16)$$

where  $loc_k$  denotes the location of digital twin server  $DT_k$ , and  $M$  is the number of base stations in the network.

The average latency is a crucial index to quantify the placement policy of the digital twin servers, which illustrates the running efficiency of the digital twin empowered network. The average placement latency can be written as

$$T(pl, A) = \frac{1}{MN} \sum_{i \in \mathcal{N}, j \in \mathcal{M}} T_{pl}(u_i, DT_j) a_{ij}, \quad (17)$$

where  $T_{pl}(u_i, DT_j)$  denotes the time cost between user  $i$  and digital twin server  $DT_j$  under placement policy  $pl$ , and  $a_{ij} \in A$ . From Eq. (17) we can see that the average latency is determined by the digital twin server placement policy and the association relations between user devices and the digital twin servers.

The objective of the digital twin placement problem is to find the solution with placement locations and association relations that minimize the average latency. The association relations have a direct effect on the latency performance, which

can also determine the digital twin placement locations. Thus we formulate the optimization problem as

$$\begin{aligned}
 P1: \quad & \min_{pl, a \in \mathcal{A}} T(pl, A) & (18) \\
 \text{s.t.} \quad & \sum_{j \in \mathcal{M}} a_{ij} = 1, \forall i \in \mathcal{N}, & (18a) \\
 & \sum_{i \in \mathcal{N}} a_{ij} \leq N_j, \forall j \in \mathcal{M}, & (18b) \\
 & \sum_{i=1}^N x_{ij} f_{ij} \leq F_j, & (18c) \\
 & T_{u_i}(t) \leq \tau_{u_i}, & (18d) \\
 & a_{ij} \in \{0, 1\}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}. & (18e)
 \end{aligned}$$

Constraint (18a) ensures that the digital twin of a user device can only be maintained by one digital twin server. Constraint (18b) indicates that the number of digital twins a server maintains, cannot exceed its capacity. Constraint (18c) ensures that the consumed computation resource does not exceed its total computation capability. Constraint (18d) indicates that the interaction time between user  $u_i$  and its corresponding digital twin should not exceed the delay requirement of  $u_i$ , that is,  $\tau_{u_i}$ . Constraint (18e) denotes that  $a_{ij}$  is a boolean variable, which only has two states, 0 and 1. Problem (18) is a combinational problem. Since there are several products of variables in the objective function, and the time cost of each BS is also affected by the resource states of other BSs, problem (18) can not be solved in polynomial time.

### B. Digital Twin Migration Problem

In the digital twin placement, the average latency is an important index for quantifying the running efficiency of the system. Different from the placement phase, the migration phase aims at dealing with the dynamic network states in consideration of the mobility of users. The objective of digital twin migration is to improve the service quality and experience for users. Thus, instead of focusing on the global system performance, we define the utility of users  $U(t)$  to quantify the effect of digital twin migration at time slot  $t$ , based on the latency and the energy cost for  $u_i$ . The energy cost for  $u_i$  to synchronize its running data to the digital twin is

$$E_{u_i}(t) = p_{u_i}(t) \cdot T_{u_i}^{com}. \quad (19)$$

Thus, the utility of user  $u_i$  can be expressed as

$$U_i(t) = \beta e^{\tau_{u_i} - T_{u_i}(t)} - (1 - \beta) E_{u_i}(t), \quad (20)$$

where  $\tau_{u_i}$  is the delay requirement of  $u_i$ ,  $T_{u_i}(t)$  is the real interaction time cost of  $u_i$ , and  $\beta \in (0, 1)$  is the weight factor.  $U_i(t)$  is negatively correlated to the latency of user  $u_i$ , i.e.,  $T_{u_i}(t)$ .

The objective of digital migration is to maximize the utility function given by Eq. (20) by migrating digital twins between servers, which can be formulated as

$$P2: \quad \max_{a \in \mathcal{A}} U_i(t) \quad (21)$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{N}} a_{ij} \leq N_j, \forall j \in \mathcal{M}, \quad (21a)$$

$$\sum_{j \in \mathcal{M}} a_{ij} = 1, \forall i \in \mathcal{N}, \quad (18b)$$

$$T_{u_i}(t) \leq \tau_{u_i}, \quad (21c)$$

where  $a \in \mathcal{A}$  denotes the migration policy that changes the association correlations between user devices and digital twin servers. Constraint (21a) and (21b) ensure that the digital twin of a user device can only be associated to one digital twin server, while one digital twin server can maintain multiple digital twins. Constraint (21c) denotes that the latency for  $u_i$  should not exceed its delay requirement to guarantee the service quality.

## V. DRL FOR DIGITAL TWIN PLACEMENT AND TRANSFER LEARNING FOR DIGITAL TWIN MIGRATION

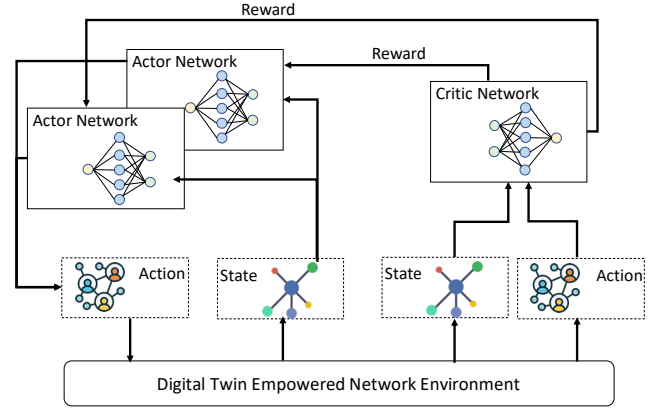


Fig. 4: The process of proposed DRL method

### A. Reinforcement Learning Empowered Digital Twin Placement

We formulate the optimization problem as an MDP  $\mathcal{M} = \{\mathcal{S}(t), \mathcal{A}(t), \mathcal{R}, \mathcal{S}(t+1)\}$ , where  $\mathcal{S}(t)$  is the current system state,  $\mathcal{A}(t)$  is the action adopted at time slot  $t$ ,  $\mathcal{R}$  is the reward function, and  $\mathcal{S}(t+1)$  is the next state. We consider that the network states including available computing and communication resources follow the Markov property. We define the set of system states  $\mathcal{S}$ , action set  $\mathcal{A}$ , and the reward function  $\mathcal{R}$  in our wireless digital edge network model as follows.

- **System State:** The system state in our proposed scheme consists of the physical states of end users  $s_i$ , the placement of digital twin servers  $pl_j$ , and the association relations  $l_{ij}$ . The physical states of users contain the distances between users and servers  $d_{ij}$ , the data of user  $i$  to be transmitted  $D_i$ , and the available data rate between users and servers  $r_{ij}$ . We denote the system state as

$$\mathcal{S}(t) = \{s_i(t), pl_i^0(t), l_{ij}^0(t)\}. \quad (22)$$

- *Agent Action:* The agents are base stations equipped with MEC servers in our system. Each agent selects its action  $a_j(t)$  including digital twin placement policy and association strategy. The action of digital twin placement is expressed as  $\{0, 1\}$  that denotes whether the server acts as the digital twin server. The association strategy of server  $j$  is denoted as  $\{l_{j1}, l_{j2}, \dots, l_{jn}\}$ , where  $l_{ji} = 1$  if end user  $u_i$  is associated to server  $j$ . Otherwise,  $l_{ji} = 0$ . The action space of multiple agents can be written as

$$\mathcal{A}(t) = \{a_i | a_{ij} = (p_{li}, l_{ij}), \forall i \in \mathcal{N}, \forall j \in \mathcal{M}\}. \quad (23)$$

- *Reward Function:* The reward function should be carefully designed to let each agent make the optimal decision for edge association. The goal of each agent is to minimize the average latency and deployment cost for maintaining digital twins within its coverage. The reward function mainly contains two parts: the latency reward  $R_L$  and the cost function  $R_C$ . The latency reward is

$$R_L = -T(p, A). \quad (24)$$

The cost for digital twin server placement is

$$R_C = \varphi \cdot m + E^{com}(u), \quad (25)$$

where  $\varphi$  is the unit cost factor for placing a digital twin server,  $m$  is the number of digital twins, and  $E^{com}(u)$  is the energy cost for users to transmit their data calculated according to Eq. (19). The total reward can be written as

$$R_t = \alpha \cdot R_L - \beta \cdot R_C, \quad (26)$$

where  $\alpha, \beta$  are the weight factors for the latency and cost reward, respectively.

We propose to use deep reinforcement learning to search for the best digital twin placement strategy. Fig. 4 shows the running mechanism of our proposed DRL-based method. There are mainly two components in our proposed scheme, that is, the actor network (deterministic policy network) and the critic network (Q network). Each agent has its own actor network to make association decision with users under its coverage. The goal of training is to maximize the expectation of the cumulative discounted reward, which is written as

$$\mathcal{R} = \sum_{t=1}^T \gamma^{t-1} R_t, \quad (27)$$

where  $\gamma \in (0, 1]$  is the reward discount factor. We use  $\pi = [\pi_1, \pi_2, \dots, \pi_n]$  to denote the policies of the  $n$  agents, whose parameters are denoted as  $\theta = [\theta_1, \theta_2, \dots, \theta_n]$ . Thus we have the policy gradient for agent  $i$  as

$$\begin{aligned} \nabla_{\theta_i} J(\pi_i) &= E_{\{\theta_i\}, a \sim D} [\nabla_{\theta_i} \pi(a_i | o_i) \cdot \\ &\nabla_{a_i} Q_i^{\pi}(\{\theta_i\}, a_1, \dots, a_n) |_{a_i = \pi_i(o_i)}], \end{aligned} \quad (28)$$

where  $\{\theta_i\}$  is the observation of agent  $i$ , that is, the state of each agent. In our scheme, since the placement of digital twin requires global coordination, we consider that all the agents share the same system state through information exchange between servers. For agent  $i$ , it determines its action  $a_i$  through its actor DNN  $\pi(s_t | \theta_{\pi})$ , denoted as

$$a_i(t) = \pi_i(s_t | \theta_{\pi_i}) + \mathfrak{N}, \quad (29)$$

where  $\mathfrak{N}$  is the random noise for generating a new action. The actor DNN is trained as

$$\theta_{\pi} = \theta_{\pi} + \alpha_{\pi} \cdot \mathbb{E}[\nabla_{a_i} Q(s_t, \mathbf{a}_1, \dots, \mathbf{a}_i | \theta_Q) |_{\mathbf{a}_i = \pi(s_t | \theta_{\pi})} \cdot \nabla_{\theta_{\pi}} \pi(s_t)], \quad (30)$$

where  $\alpha_{\pi}$  is the learning rate of the actor DNN.

The critic DNN of agent  $i$  is trained as

$$\theta_{Q_i} = \theta_{Q_i} + \alpha_{Q_i} \cdot \mathbb{E}[2(y_t - Q(s_t, \mathbf{a}_i | \theta_{Q_i})) \cdot \nabla Q(s_t, \mathbf{a}_1, \dots, \mathbf{a}_i)], \quad (31)$$

where  $\alpha_{Q_i}$  is the learning rate,  $y_t$  is the target value,  $(\mathbf{a}_1, \dots, \mathbf{a}_i)$  is the actions of the agents in our system.

The proposed deep reinforcement learning based digital twin placement algorithm is summarized as Algorithm 1. At the beginning, all the actor network and critic network are initialized randomly as the initial training parameters. Then the replay memory is initialized for storing the experiential samples in the training process. In each episode, the agent selects its action towards its current observation state, and obtains the reward for its current action. Then the new observation of the system state is obtained. The experience tuple  $(s_t, a_i, r_t, s_{t+1})$  is then stored to the replay buffer. Finally, the agents train their critic network and actor network by sampling records from the replay buffer.

---

**Algorithm 1** Deep reinforcement learning based digital twin placement algorithm

---

**Input:** The user set  $\mathcal{U}$ , the base station set  $\mathcal{B}$

**Output:** The digital twin server placement strategy  $\Pi$

- 1: **for** each agent  $i \leq M$  **do**
- 2: Randomly initialize critic network and actor network
- 3: Initialize replay memory
- 4: **end for**
- 5: **for** episode  $e \leq E$  **do**
- 6: Initialize network environment setup
- 7: **for** each time slot  $t$  **do**
- 8: **for** each agent  $i \leq M$  **do**
- 9: Observe current system state  $s$  and execute action  $a_i$  according to Eq. (29);
- 10: Obtain reward  $R_i(t)$  according to Eq. (26), and get a new state observation  $s_{t+1}$  based on actions  $(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$
- 11: Store  $(s_t, \mathbf{a}_i(t), R_i(t), s_{t+1})$  to replay memory
- 12: **end for**
- 13: **end for**
- 14: **for** agent  $i \leq M$  **do**
- 15: Update actor network and critic network according to Eq. (30) and Eq. (31)
- 16: Update target network of agent  $i$
- 17: **end for**
- 18: Generate the digital twin placement strategy towards current system state  $s$ , that is,

$$\Pi(t) = \mathcal{M}(s, t)$$

19: **end for**

---



### B. Transfer Learning Empowered Digital Twin Migration

In the migration phase, we focus on migrating the digital twin of a user device from the source server to the target server. The migration of digital twins can continuously provide users with high-quality services. Since training a complete reinforcement learning model can consume a huge amount of resources and can incur long latency, we propose to use transfer learning for digital twin migration between servers. In the digital twin migration process, we aim at constructing the digital twin model of a user device instantly in the target server by leveraging transfer learning to reuse the obtained knowledge in the source digital twin server, as denoted in Fig. 5. For user  $u_i \in \mathcal{U}$  moving from location A to location B and location C, the transfer learning empowered digital twin migration processes are as follows.

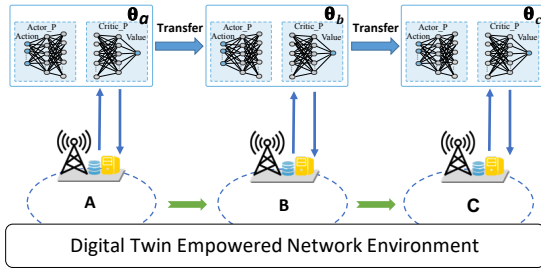


Fig. 5: The process of transfer learning for digital twin migration

- 1) *Target server selection*: We first select the target server according to the current distance and communication channel state, that is, select the server for user  $u_i$  with the minimal time delay given by Eq. (11).
- 2) *Model parameter aggregation*: The reinforcement learning model learned at server  $DT_A$  contains the running knowledge of user  $u_i$ . Thus we transfer the parameters of the trained DRL network model, denoted as  $\theta_{u_i}^A$ , to the target server. The new model at the target server is calculated as

$$\theta_{u_i} = \sum_{p_j \in \mathcal{T}r_{u_i}} w_{p_j} \theta_{u_i}^{p_j}, \quad (32)$$

where  $p_j$  is the location point in the trajectory of user  $u_i$ . For example,  $p_j = \{A, B, C\}$  in the example case.  $w_{p_j}$  is the weighting coefficient and  $\sum_{p_j} w_{p_j} = 1$ . Then we obtain the initial model  $\theta_{u_i}$  for user  $u_i$  in the target server.

- 3) *Experience samples update*: The experience samples including states, actions, and rewards related to user  $u_i$ , are also transmitted from the source server to the target server. Thus we have the new action and state space

$$(\mathcal{A}_t, \mathcal{S}_t) = (\mathcal{A}_{t-1}, \mathcal{S}_{t-1}) \cup (\mathcal{A}_s, \mathcal{S}_s), \quad (33)$$

where  $(\mathcal{A}_s, \mathcal{S}_s)$  is the experienced states and actions of the source model.

- 4) *Train the new model*: Finally, we use the samples  $(\mathcal{A}_t, \mathcal{S}_t)$  to train the new model  $\theta_{u_i}$ , and obtain the migrated model as

$$\theta_{u_i}(t) \leftarrow \theta_{u_i}(t-1) + \rho \cdot \theta_{u_i}^{src}, \quad (34)$$

where  $\theta_{u_i}^{src}$  is the source model, and  $\rho$  is the weight factor.

The complete process of our proposed transfer learning empowered digital migration algorithm is summarized in Algorithm 2.

---

**Algorithm 2** Transfer learning empowered digital twin migration algorithm

---

**Input:** The trained source digital twin model  $\theta_{u_i}^{src}$

**Output:** The target digital twin model  $\theta_{u_i}(t)$

- 1: **for** each user  $u_i \in \mathcal{U}$  **do**
  - 2:   Select the target server for constructing new digital twin models
  - 3:   Transmit  $\theta_{u_i}^{src}$  to the target server
  - 4:   Initialize the target model with  $\theta_{u_i}^{src}$
  - 5:   **for** each time slot  $t$  **do**
  - 6:     Merge the experience sample space according to Eq. (33), and get  $\mathcal{S}_t \rightarrow \mathcal{A}_t$
  - 7:     Train the target model based on the merged samples
  - 8:     Obtain the target digital twin model  $\theta_{u_i}(t) \leftarrow \theta_{u_i}^{src}$
  - 9:   **end for**
  - 10: **end for**
  - 11: Obtain the target digital twin model set  $\theta_{u_i}(t) = \{\theta_{u_i}(t) | u_i \in \mathcal{U}\}$
- 

Specifically, in the target server selection phase, the goal is to select a target digital twin server that has minimal interaction latency and maximum utility  $U_i$  with user  $u_i$ . The interaction latency  $T_{int}$  can be obtained according to Eq. (10). The detailed target server selection algorithm is listed in Algorithm 3.

---

**Algorithm 3** Target digital twin server selection algorithm

---

**Input:** The set of digital twin server  $DT^s$ , source digital twin server ( $DT(u_i, src)$ )

**Output:** The target digital twin server  $DT(u_i, tar)$

- 1: Initialize the target server set  $TS = \emptyset$ , calculate current interaction time cost  $T_{int}(u_i, ini)$  and  $U_i(ini)$
  - 2: **for** each digital twin server  $DT_j \in DT^s \wedge DT_j \notin TS$  **do**
  - 3:   Calculate the time cost between  $u_i$  and  $DT_j$ ,  $T_{int}(u_i, DT_j)$ , further obtain  $U_i(DT_j)$
  - 4:   **if**  $U_i(DT_j) \geq U_i(ini)$  **then**
  - 5:      $DT(u_i, tar) = DT(u_i, j)$
  - 6:   **else**
  - 7:     Keep current  $DT(u_i, tar)$
  - 8:   **end if**
  - 9:   Add  $DT_j$  to  $TS$
  - 10: **end for**
  - 11: Obtain the target digital twin server  $DT(u_i, tar)$
- 

## VI. NUMERICAL RESULTS

We consider a network topology with one MBS as the cloud server,  $M$  base stations, and  $N$  end users, as shown in Fig. 6. The users and BSs are randomly distributed in an area of  $1500\text{m} \times 1500\text{m}$ . The locations of the users and BSs, the available computing resources, and the channel states vary with time in our simulation system. The maximum

TABLE II: Simulation Parameters

Parameter	Setting Value
Transmission channel bandwidth $W$	20 MHz
User device transmit power $p_i$	0.2 W
CPU frequency of edge server $F_j$	[8,24] GHz
Noise power function $N_0$	-174 dBm/Hz
User number $N$	{100, 80}
Base station number $M$	{10, 5}
Training iterations	5000
Discounting factor $\gamma$	0.7
Batch size	32
Learning rate of actor network	0.001

transmission power of the user device is set to 200 mW. The CPU computation capacity is randomly derived from the range [8, 24] GHz. The training iterations of our proposed method is set to 5000. The discounting factor  $\gamma$  in our DRL based algorithm is set to 0.7. In our actor network, the batch size is set to 32, and the learning rate is set to 0.001. The detailed simulation parameters are shown in Table II.

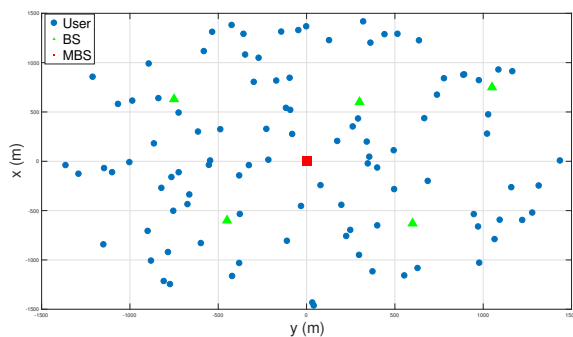


Fig. 6: Illustration of the simulation network

Fig. 7 shows the cumulative system costs of our proposed algorithm for edge association with different numbers of BSs and users. The proposed algorithm is executed and trained in 5000 rounds. As can be seen from Fig. 7, our proposed algorithm achieves good convergence performance in 5000 iterations. By comparing the performance of different numbers of BSs and users, we can see that the number of users has notable affection on the system cost. When the number of BSs  $M$  is set to 5, the system cost of  $N = 100$  is much larger than that of  $N = 80$ . That is because the increased users make the limited resources much more scarce in our proposed wireless digital twin networks. The converged results show that the cumulative system cost increases with the number of users and the number of BSs. While the number of users has a greater effect on the system cost than the number of BSs.

Fig. 8 depicts the optimization process of our proposed adaptive edge association method in 25 steps, with different numbers of users and BSs. The system cost is minimized to a quite small value in 10 steps, which shows the good convergence of our proposed algorithm. From the decreasing system cost, we can see that our proposed scheme converges fast and reaches the convergence points in a few steps. Moreover, from the sub-figure, we can see that the increase in the number of users has a greater impact on the system cost compared to increase in the number of BSs. The trend further

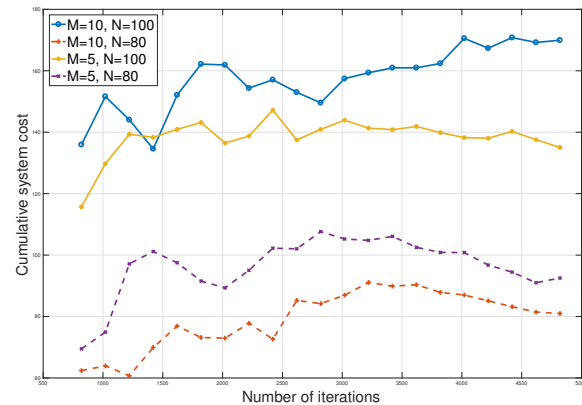


Fig. 7: The cumulative system cost

confirms our observation in Fig. 7. Moreover, by comparing the performance of different number of BSs, we can draw the following insight: the fewer BSs, the higher the system cost. For example, when we take  $N = 80$  and  $M = 5$ , the system cost is much larger than when  $N = 80$  and  $M = 10$ . Similarly, when we take  $N = 100$  and  $M = 5$ , the system cost is also always larger than that of  $N = 100$  and  $M = 10$ . The reason is that with decrease in the number of BS, the users have fewer available resources to access, which inevitably increases the system cost for completing data transmission and edge computing.

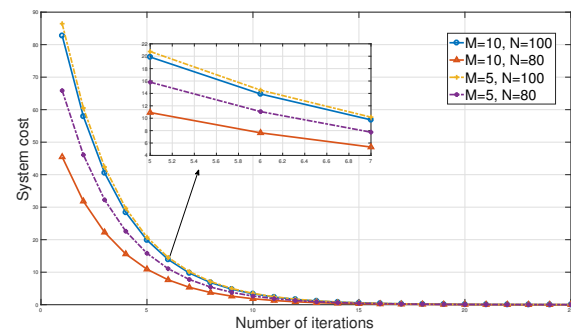


Fig. 8: The optimization process of the edge association cost

We further compare our proposed method with two benchmark methods in the association. The comparison results are shown in Fig. 9. In the nearest association method, the users always select the close servers as the digital twin servers to construct and maintain their digital twins. While in the random association method, the digital twin servers are placed and associated randomly in the network. From Fig. 9 we can clearly see that the proposed method achieves a significant reduction in total time cost compared to the benchmark methods. Our proposed method can search for the optimal edge association strategies according to the instant system states based on our trained actor-critic networks towards different network states.

Fig. 10 illustrates the total time cost (in seconds) of training our DRL based model with different numbers of BSs and

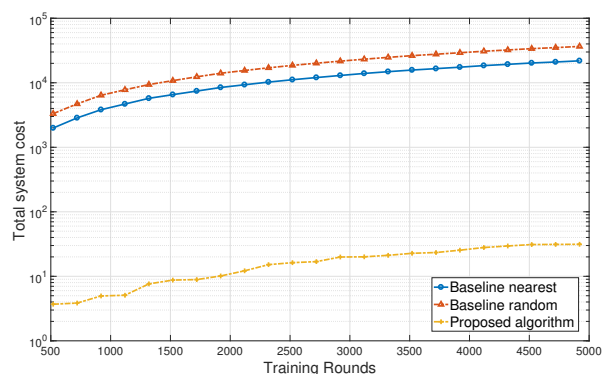


Fig. 9: The comparison of time cost

users. The training iteration ranges from 1 to 5000. From the four experimental sets in Fig. 10 we can see that it costs thousands of seconds for training our proposed models. The experimental set with 80 users and 5 BSs consumes the least time for training compared with other experimental sets, while the experimental set with 100 users and 10 BSs consumes the most time. Taking iteration 5000 as an example, when the number of users  $N$  is set to 100, the running time with 10 BSs increases by 95% compared to the case when  $M = 5$ . While the number of users is 80, the increase is by 57%. Similarly, when the number of BSs  $M = 10$ , the running time for the case  $N = 100$  increases by 25% compared to when  $N = 80$ . While  $M = 5$ , the increased rate is 22%. From the above observations, we can conclude that the increase of BS consumes more training time than the increase of users. The reason is that the number of BSs has a significant impact on the complexity of the edge association problem, including the digital twin server selection and the user association, while the number of users only affects the complexity of association strategies in our proposed wireless digital twin networks. The results in Fig. 10 illustrate that our proposed scheme can be trained quickly towards dynamic network states. It can be also inferred that the optimal decision can be made by our proposed scheme within a few milliseconds.

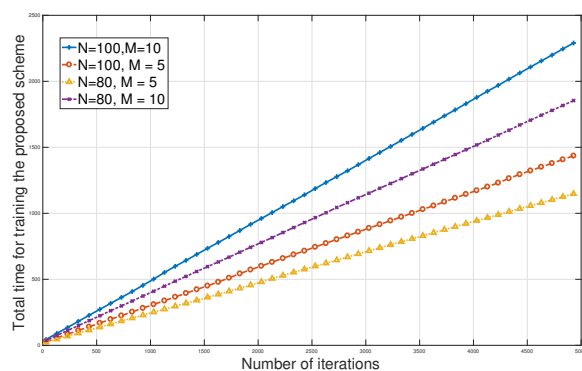


Fig. 10: The total running time for training

## VII. CONCLUSION

In this paper, we proposed a new digital twin empowered network model for 6G. We first presented the digital twin empowered system model for 6G networks including end users, BSs, and cloud servers. We formulated the edge association problem consisting of digital twin placement and digital twin migration for reducing system latency and for increasing utility for the users. To improve the efficiency of the proposed scheme with limited resources, we derived an optimal solution to the edge association problem by exploiting deep reinforcement learning and transfer learning. Numerical results for various network states demonstrated that the proposed scheme effectively reduces the average system latency and improves the convergence rate.

## REFERENCES

- [1] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?" *Nature Electronics*, vol. 3, no. 1, pp. 20–29, 2020.
- [2] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6g wireless communications: Vision and potential techniques," *IEEE Network*, vol. 33, no. 4, pp. 70–75, 2019.
- [3] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.
- [4] S. Research, "6g white paper," <https://research.samsung.com/next-generation-communications>, 2021.
- [5] G. Liu, Y. Huang, N. Li, J. Dong, J. Jin, Q. Wang, and N. Li, "Vision, requirements and network architecture of 6g mobile network beyond 2030," *China Communications*, vol. 17, no. 9, pp. 92–104, 2020.
- [6] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia Cirp*, vol. 61, pp. 335–340, 2017.
- [7] W. Sun, S. Lei, L. Wang, Z. Liu, and Y. Zhang, "Adaptive federated learning and digital twin for industrial internet of things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.
- [8] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6g networks," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.
- [9] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.
- [10] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.
- [11] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5g beyond for the industrial internet of things," *IEEE Network*, vol. 33, no. 5, pp. 12–19, Sep. 2019.
- [12] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, June 2019.
- [13] M. Maier and A. Ebrahimzadeh, *Toward 6G: A New Era of Convergence*. John Wiley & Sons, 2020.
- [14] I. Ahmad and J. Huusko, *White Paper on Machine Learning in Wireless Communication Networks*, ser. 6G Research Visions, S. Ali, W. Saad, and D. Steinbach, Eds. Finland: University of Oulu, Jun. 2020, no. 7.
- [15] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6g communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, 2020.
- [16] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick *et al.*, "6g white paper on machine learning in wireless communication networks," *arXiv preprint arXiv:2004.13875*, 2020.
- [17] H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, accepted, 2019.
- [18] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

- [19] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.
- [20] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.
- [21] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technology, challenges and open research," 2019.
- [22] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in cps-based production systems," *Procedia Manufacturing*, vol. 11, pp. 939 – 948, 2017, 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2351978917304067>
- [23] A. Sadeghi, F. Sheikholeslami, A. G. Marques, and G. B. Giannakis, "Reinforcement learning for adaptive caching with dynamic storage pricing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2267–2281, 2019.
- [24] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.
- [25] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
- [26] W. Sun, N. Xu, L. Wang, H. Zhang, and Y. Zhang, "Dynamic digital twin and federated learning with incentives for air-ground networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.
- [27] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning for digital twin edge networks in industrial iot," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.
- [28] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6g," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12 240–12 251, 2020.
- [29] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet of Things Journal*, pp. 1–1, 2020.
- [30] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.