Digital Twin-Driven MADRL Approaches for Communication-Computing-Control Co-Optimization

Xiaoming Yuan, *Member, IEEE*, Hansen Tian, Xinling Zhang, Hongyang Du, *Member, IEEE*, Ning Zhang, Kaibin Huang, *Fellow, IEEE*, and Lin Cai, *Fellow, IEEE*

***Abstract*—The unpredictability of network environments, limited edge resources, and the high complexity of collaborative policies are significantly hindering the development of the Industrial Internet of Things (IIoT). These challenges are particularly pronounced in healthcare, where high-priority, delay-sensitive medical tasks and large-scale personalized services face substantial obstacles. To address these challenges, this paper proposes the Self-Attention Enhanced QMIX with Multi-Pass Multi-Task Execution (SAE-MT-QMIX) algorithm, aimed at optimizing communication and computing resource allocation as well as task offloading strategies. By leveraging Digital Twin (DT) support, the algorithm achieves collaborative optimization of communication, computing, and control within the Internet of Medical Things (IoMT), significantly enhancing the quality of service for massive personalized applications. The algorithm adopts a distributed execution and centralized training framework: the distributed execution component uses the Multi-Pass Multi-Task Deep Q-Network (MPMT-DQN) algorithm to handle the complexity of parameterized action spaces in multi-task scenarios, while the centralized training component employs the Self-Attention Enhanced QMIX (SAE-QMIX) algorithm to dynamically optimize credit assignment across multiple users. Simulation results demonstrate that SAE-MT-QMIX significantly reduces delay and energy consumption compared to baseline methods. It ensures effective optimization of communication, computing, and control in dynamic IoMT, efficiently addressing diverse demands and tasks while enhancing service quality and system adaptability.**

***Index Terms*—Internet of Medical Things, digital twin, mobile edge computing, parameterized action space, multi-agent deep reinforcement learning, self-attention mechanism.**

Received 27 September 2024; revised 10 January 2025; accepted 28 February 2025. Date of publication 28 May 2025; date of current version 14 October 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62371116; in part by the Research Grants Council of Hong Kong Special Administrative Region, China, under the Fellowship Award, under Grant HKU RFS2122-7S04; in part by NSFC/RGC CRS under Grant CRS_HKU702/24; in part by the Areas of Excellence Scheme under Grant AoE/E-601/22-R; in part by the Collaborative Research Fund under Grant C1009-22G and Grant 17212423; and in part by Shenzhen–Hong Kong–Macau Technology Research Programme (Type C) under Grant SGDX20230821091559018. *(Hansen Tian and Xinling Zhang contributed equally to this work.) (Corresponding author: Xiaoming Yuan.)*

Xiaoming Yuan, Hansen Tian, and Xinling Zhang are with the Hebei Key Laboratory of Marine Perception Network and Data Processing, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China (e-mail: yuanxiaoming@neuq.edu.cn; ths16068481@163.com; XinlingZh34@163.com).

Hongyang Du and Kaibin Huang are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, SAR, China (e-mail: duhy@eee.hku.hk; huangkb@eee.hku.hk).

Ning Zhang is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: ning.zhang@uwindsor.ca).

Lin Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: cai@ece.uvic.ca).

Digital Object Identifier 10.1109/JSAC.2025.3574616

I. Introduction

LEVERAGING the collaborative capabilities of cloud and fog technologies, Cloud-Fog Automation achieves the synergistic optimization of communication, computing, and control, enabling networked computing, intelligent control, and deterministic connectivity [1]. With the rapid advancement of Industrial Internet of Things (IIoT) technologies, the Internet of Medical Things (IoMT) has become a key application area for Cloud-Fog Automation in the healthcare domain.

By effectively connecting sensors, users, and medical institutions, IoMT [2] enables real-time collection, transmission, analysis, and application of medical data. IoMT can provide personalized medical services for a diverse range of users, ensuring real-time and reliable Quality of Service (QoS) support. Additionally, IoMT can also effectively deal with the uneven distribution of medical resources in the process of medical informatization, enhancing the efficiency, intelligence, and remote accessibility of healthcare services.

However, realizing the full potential of IoMT remains challenging due to limited computing and communication resources, as well as the need to deliver efficient, personalized healthcare services to a large, diverse user base while adapting to complex, dynamic network environments [3]. To tackle these challenges, leveraging the advantages of Cloud-Fog Automation is essential [4]. Developing an integrated framework for communication, computation, and control is crucial to enable efficient information sharing, resource allocation, intelligent decision-making, and service management.

In the Cloud-Fog Automation architecture, 5G/6G provides near-deterministic connectivity, ensuring real-time data transmission and the efficient execution of control strategies. Mobile Edge Computing (MEC) [5] extends computing capabilities to the edge of physical systems, enhancing local computation near data sources and significantly improving data processing efficiency and real-time responsiveness. Digital Twin (DT) [6] achieves real-time mapping between physical systems and their digital models, facilitating contin-

0733-8716 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

uous data collection and feedback, continuously optimizing control strategies and enabling intelligent, automated control systems. The integration of these technologies significantly enhances the real-time data processing capabilities of IIoT, improves system flexibility and adaptability, and optimizes both data analysis and decision-making support [7].

To effectively implement Cloud-Fog Automation in IoMT, it is essential to establish a DT-driven IoMT system. This system should comprehensively map physical entities and medical service functions, enabling a deep understanding and accurate prediction of IoMT's complexities and dynamics, while providing MEC strategy support for efficient, real-time data sharing. Furthermore, the strategic integration of 5G/6G, MEC, and DT should facilitate the design of collaborative algorithms for optimizing communication, computation, and control offloading. These algorithms must address the real-time, intelligent, and reliable processing of large-scale medical tasks, while optimizing resource allocation to enhance the overall performance of the IoMT system [4].

When addressing the challenges of collaborative decision-making in communication, computation, and control, Deep Reinforcement Learning (DRL) demonstrates significant advantages over other methods due to its superior adaptability, flexibility, and generalization capabilities [8]. DRL combines the perceptual strengths of Deep Learning (DL) with the strategy-making abilities of Reinforcement Learning (RL), learning through real-time interaction and feedback with the environment. This enables DRL to effectively adapt to the complexity and uncertainty inherent in IoMT. Moreover, Multi-Agent DRL (MADRL) [9] excels in dynamic multi-user decision-making, coordinating user demands and adapting to real-time changes. It can effectively meets stringent requirements for real-time, efficient, and personalized medical services, offering an intelligent and scalable approach to enhancing IoMT performance.

Collaborative decision-making in IoMT presents significant challenges. On one hand, the allocation of communication and computing resources, alongside control strategies, involves both continuous and discrete variables, which are intricately interrelated. Control strategies often exert a profound influence on subsequent resource allocation, resulting in dynamic interactions. However, existing algorithms face difficulties in effectively addressing such parameterized action spaces and adapting to the complexities of IoMT scenarios [10]. On the other hand, designing a reasonable multi-agent training mechanism and implementing dynamic credit assignment to balance control strategies among multiple users becomes crucial [11]. By flexibly adapting to the personalized needs and task differences of agents, it optimizes individual resource utilization while ensuring the overall stability and efficiency of the system. Achieving an equilibrium that optimizes individual resource utilization while maintaining overall system stability and efficiency remains a critical challenge requiring urgent resolution.

In summary, this paper proposes a MADRL algorithm based on Self-Attention Enhanced QMIX with Multi-Pass Multi-Task Execution (SAE-MT-QMIX), aimed at addressing challenges related to resource allocation, task offloading, and multi-user collaboration in DT-driven IoMT. This algorithm enables collaborative optimization of communication, computing, and control. It adopts a distributed execution and centralized training framework, where the Multi-Pass Multi-Task Deep Q-Network (MPMT-DQN) algorithm is designed for the agent network to solve the complex parameterized action space in distributed execution, effectively reducing both time and space complexity. The centralized training part employs the Self-Attention Enhanced QMIX (SAE-QMIX) algorithm, which enhances the credit assignment capability of the Mixing network using a self-attention mechanism, dynamically adapting to environmental changes and ensuring dynamic optimization of global performance. This algorithm enables collaborative optimization of communication, computation, and control in complex and dynamic IoMT environments, providing real-time, efficient, and reliable offloading strategies and resource allocation, ensuring the fulfillment of large-scale users' personalized medical task requirements with low delay, low energy consumption, and high quality. The contributions of this paper are summarized as follows:

- *DT-driven IoMT:* The system architecture is constructed and comprehensively described to achieve high-quality and high-precision virtual mapping of physical entities and service functions within IoMT, supported by robust data to enhance the collaboration of communication, computation, and control optimization.
- *MPMT-DQN Algorithm:* Designed for agent networks, this algorithm tackles high complexity from multiple tasks and complex action spaces in DT-driven IoMT, enabling efficient collaboration of communication and computing resources for optimized resource allocation and task execution.
- *SAE-QMIX Algorithm:* The algorithm utilizes QMIX for collaborative learning and a self-attention mechanism to address limited observation and multi-user control strategy challenges. It improves user adaptability and enables efficient resource allocation and high-quality medical services in dynamic IoMT environments.
- *Simulation:* Simulations show that, compared to baseline algorithms, the proposed algorithm achieves collaborative decision-making in communication, computation, and control, effectively delivering efficient, reliable real-time services to meet users' diverse demands for low delay, energy efficiency, and high reliability in medical services.

The rest of the paper is organized as follows. Section II discusses the related work. Section III presents the system model of DT-driven IoMT. Section IV defines the problem and construct the RL model. Section V describes the proposed algorithm in detail. Section VI simulates and analyzes the results. The conclusion will be presented in Section VII. The notations used throughout this paper are summarized in Table I for clarity and convenience:

II. Related Work

In recent years, many scholars have extensively researched the issue of computing offloading in the IoMT with DT. This

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF NOTATIONS

Notation	**Definition**
i, j, k	Indices of users, tasks, and servers
$\mathcal{U}, \mathcal{T}_i, \mathcal{M}$	Sets for users, tasks for user i, and servers
N^u, N_i^t, N^m	Total number of users, tasks for user i, and servers
D, C, P	Data size, computing complexity, and priority of a task
$s_{i,j}, f_{i,j}, b_{i,j}$	Offloading strategy, communication and computing resource allocation for task
k_0	Constant related to energy consumption
$g_{i,k}, h_{i,k}, PL\left(\cdot\right)$	Channel gain model, Rayleigh fading and path loss between user i and server k
$d_0, d_{i,k}$	Reference distance and distance between user i and server k
η, σ	Path loss exponent and Rayleigh distribution variance
$\gamma_{i,k}, rate_{i,j}$	Signal-to-noise ratio and transmission rate
P_i^{tr}, N_0	Transmission power of user i and Gaussian noise
$T_{i,j}^{exe}, T_{i,j}^{tr}, T_{i,j}, T_i, T^{all}$	Delays for task execution, transmission, processing, user-level total delay, and system-wide delay
$E_{i,j}^{exe}, E_{i,j}^{tr}, E_{i,j}, E_i, E^{all}$	Energy consumption for task execution, transmission, processing, user-level consumption, and system-wide consumption
F_i, F_k, B_k	Total computing resources in user i and server k, and total communication resource in server k
n_i^t, n_k^t	Number of tasks processed locally by user i and offloaded to server k
$Cap_i, T^{\max}$	Energy capacity of user i and maximum tolerated delay
c, a, μ	Constant scaling task reward, coefficient for priority-reward relationship, baseline priority
γ^t, γ^e	Coefficients for delay and energy consumption decay
$\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{R}, \Omega, \mathcal{P}, \gamma$	State, action and observation space, reward function, state and observation probability, discount factor in Dec-POMDP
$s, s^T, \hat{s}$	Global state, terminated global state, observable global state
a, a^d, a^c	Action, discrete action, and continuous action
$r, \text{cost}^{\max}, \text{cost}$	Reward, max cost, cost of task
$\alpha^T \alpha^E$	Weights for delay and energy
$x_{t,i,j}, x_{t,i,j-}$	Task information and information of other tasks
$o_{t,i}, o_{t,i}^{other}$	Observation and observations of other users
$o_{t,i,j}, \hat{o}_{t,i,j}, \hat{o}_{t,i,j}^{s_{t,i,j}}$	Task observation, corrected task observation, and corrected task observation for multi-batch processing
$\mathbf{Q}_{t,i}^j, i^Q, j^Q$	Q-value matrix generated by the MPTQ network, along with row and column indices
$\mathbf{q}_{t,i}^{s_{t,i,j}}, q_{t,i,j}^{s_{t,i,j}}$	Row vectors of $\mathbf{Q}_{t,i}^j$ and their corresponding elements
$Q_{t,i}^{s_{t,i,j}}, Q_{t,i}, Q_t^{tot}$	Aggregated agent Q-value within a pass, aggregated agent Q-value, and system Q-value
$\mathbf{q}, \mathbf{k}, \mathbf{v}, d_k$	Query, key, value vectors, and the scaling factor
$\mathbf{W}^q, \mathbf{W}^k, \mathbf{W}^v$	Coefficient matrices for query, key, and value vectors
$\alpha, \hat{\alpha}, \mathbf{A}, \hat{\mathbf{A}}$	Weight coefficients and weight matrix
$\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$	Weights and biases in the mixing network
$\hat{Q}_{t,i}, \hat{\mathbf{Q}}$	Weighted Q-value for an agent and the vector of weighted Q-values for all agents
$\mathcal{B}_i, \mathcal{B}$	Agent experience pool and system experience pool
l, N^B	Batch index and batch size
$\theta_{t,i}^{\mu}, \theta_{t,i,j}^{Q}, \theta_t^{Q}$	Model parameters of Actor network, MPTQ network, and system Q-value network

section aims to review and summarize the current state of research on DT-driven MEC architecture, DRL algorithms for computing offloading, and MADRL algorithms for computing offloading in multi-user scenarios.

A. DT-Enabled MEC Architecture

Some researchers utilize the real-time monitoring, analysis, and simulation capabilities of DT to generate high-quality data and offer support and real-time feedback for MEC. References [12] and [13] have proposed similar DT-driven intelligent offloading frameworks. In these frameworks, DT can simulate the behavior of physical entities in MEC by constructing high-fidelity virtual mappings, enabling joint optimization of task offloading and resource allocation strategies. Simultaneously, DT can also play a role in algorithm optimization. By incorporating the concept of DT into aerial MEC paradigms, [14] presents a framework utilizing DT technology in Unmanned Aerial Vehicle (UAV)-assisted ground MEC networks. By monitoring the status of edge nodes and providing training data, DT assists in machine learning for intelligent offloading strategy and dynamic resource allocation. In [15], architecture combining DT and MEC with Federated learning (FL) frameworks is proposed. With real-time data analysis and network resource optimization, DT can optimize device scheduling and manage resource allocation to maximize the utility of FL services.

Additionally, in the medical field, DT is often combined with 5G and 6G technologies to provide patients with real-time remote health monitoring, diagnosis, and treatment services. Reference [16] proposes a DT-assisted quantum FL algorithm. DT enables the real-time utilization of data from IoMT devices to create DT of patients, achieving intelligent diagnosis in 5G mobile networks while protecting patient privacy. Reference [17] developed a DT-driven IoMT system, which projects surgical navigation images in real-time onto the helmet of surgeons for remote lung cancer treatment.

However, some recent literature on DT-driven MEC-based computation offloading primarily focuses on describing the architecture but falls short of providing comprehensive explanations of the specific modules and implementation details within the architecture. These studies often lack in-depth discussions of how the individual components interact and function within the framework, limiting the understanding of the practical application and optimization of such systems.

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

B. DRL Based Computing Offloading

Due to the ability of DRL to handle large-scale, high-dimensional state spaces in MEC and its adaptability to complex dynamic environments, an increasing number of scholars are turning to DRL to address the computing offloading problem.

To tackle the offloading strategy problem, [18] combines the decision tree algorithm and Double Deep Q-Network (DDQN) to propose a DT-assisted offloading scheme. Additionally, [19] proposes a two-stage task offloading optimization algorithm based on DRL and transfer learning. Both algorithms can effectively reduce the processing delay of task offloading and decrease the average energy consumption of the system. Reference [20] proposes an Actor-Critic based DRL solution for minimizing offloading delay while considering accumulated service migration costs. In [21], the multi-task joint computation offloading problem is addressed by modeling tasks with dependencies as directed acyclic graphs (DAG). An attention mechanism-based proximal policy optimization (A-PPO) collaborative algorithm is proposed to optimize the learning process of offloading strategies.

For the joint optimization problem of offloading strategy and resource allocation, [19] presents a scheduling strategy based on an improved DDQN algorithm, while also proposing a low-complexity algorithm to achieve nearly optimal allocation of energy and computing resources. In [22], an energy minimization model is developed by integrating terminal power control, computing resource allocation, and task offloading decisions. To simplify the problem, it is divided into two subproblems: a deep actor-critic-based online algorithm is proposed to address the dimensionality issue in offloading decisions, and a difference-of-convex-based method is designed to optimize power control and resource allocation. Unlike the methods mentioned above that solve the joint problem separately, [23] determines how to allocate resources through the Long Short-Term Memory (LSTM) algorithm and the Parameterized Deep Q-Network (PDQN) algorithm. Similarly, [24] employs a Parameterized Action Space Deep Q-Network (PADQN) to handle hybrid action spaces and integrates a dynamic time factor mechanism (PADQN-D) to enhance IT and cooling subsystem coordination.

However, in recent years, there has been limited research on the collaborative optimization strategies for communication, computation, and control in the context of computation offloading. Most studies tend to address the problem using algorithms such as PA-DDPG, without delving deeply into the specific challenges of this integrated approach. Furthermore, there has been little exploration of the high time and space complexity involved in handling the complex parameterized action spaces in multi-task scenarios while maintaining the potential relationships between tasks. These aspects remain under-explored, despite their importance in optimizing offloading strategies.

C. MADRL Based Computing Offloading

Due to the need to support medical services for multiple users simultaneously in IoMT, traditional DRL algorithms have not considered the challenges of competition and cooperation among users. Therefore, some scholars have delved into research on multi-user computing offloading scenarios.

To tackle challenges of traditional DRL algorithms' inability to adapt to dynamic environments, [25] proposes a offloading algorithm that combines MADRL and meta-learning to reduce the processing delay and energy consumption of tasks. Reference [26] introduces a collaborative MADRL algorithm, which utilizes centralized training and decentralized execution to incorporate collaborative multi-agent techniques, thereby enhancing task offloading efficiency in multi-user scenarios. In [27], a MADRL method based on Multi-Agent Proximal Policy Optimization (MAPPO) and attention mechanism is proposed. This method aims to pursue the optimal computing offloading strategy for multiple users while minimizing system energy consumption.

For more complex multi-agent optimization problems, [28] proposes a DT-assisted multi-agent task scheduling scheme to reduce delay and energy consumption, while improving scheduling success rate, load balancing metrics, and training efficiency. In [29], a hierarchical computing offloading approach is proposed for delay-tolerant and delay-sensitive tasks. Utilizing Multi-Agent DDPG (MADDPG), this approach achieves faster task processing, dynamic real-time resource allocation, and lower system costs. In [30], Lyapunov-based optimization algorithm is employed to determine local task scheduling and computing capabilities. Additionally, an online MADRL algorithm combined with game theory is employed to achieve intelligent and energy-efficient task offloading and transmit power allocation. In [31], a graph attention-based MADRL algorithm is proposed to learn the optimal task offloading and service caching strategy in DT-driven MEC.

Current research seldom involves the use of DT technology to optimize computation offloading algorithms in multi-agent scenarios. Additionally, there is relatively little research on simultaneously addressing complex action space problems and achieving multi-agent collaboration in the context of computation offloading. Therefore, this paper will investigate how to integrate DT to achieve efficient computing offloading in multi-user scenarios with complex action spaces, effectively addressing the aforementioned issues.

III. System Model

In this section, a system model of DT-driven IoMT is described, as illustrated in Figure 1. Serving as a robust framework for collaborative decision-making in communication, computing, and control, this model facilitates efficient resource scheduling and intelligent medical services. According to the definition of DT, the model adopts a three-layer architecture, comprising the physical network layer, twin network layer, and network application layer. The specific details of each layer are outlined below.

A. Physical Network Layer

The physical network layer serves as the foundation for DT, encompassing all physical entities and their functionalities

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

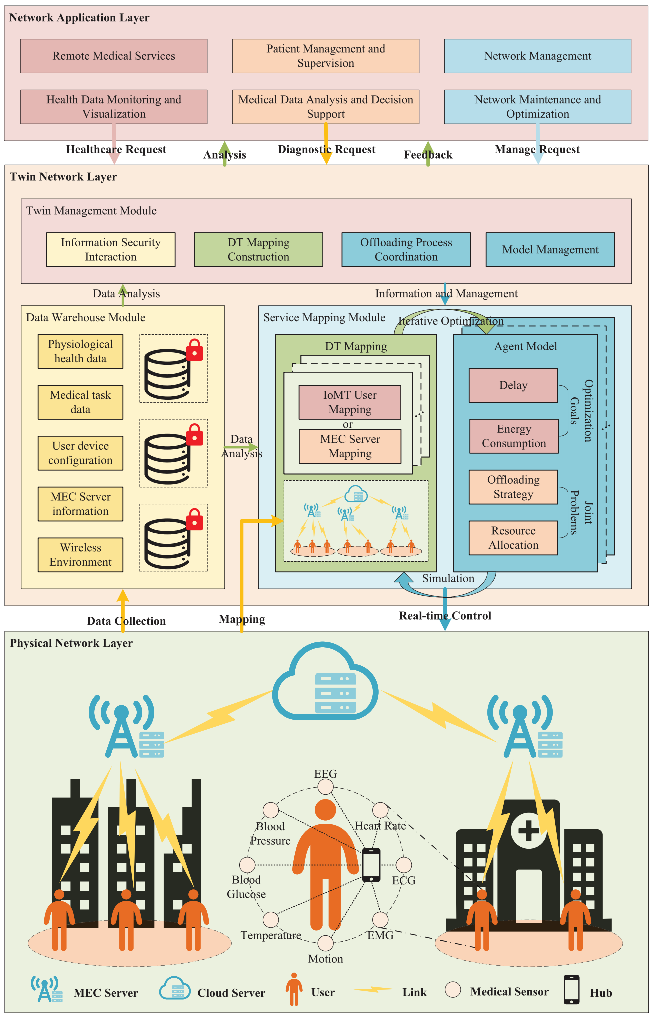

Fig. 1. System model.

that constitute the DT-driven IoMT. It is responsible for establishing and managing data transmission and interconnectivity among various physical devices. Through close collaboration and interaction with the following network twin layer, the physical network layer will not only facilitate the modeling and simulation of the DT but also fosters system optimization and intelligent policy-making. The following will provide a detailed exploration of the physical entities and the process of computing offloading in the physical network layer.

In the physical network layer, there are N^u IoMT users, with the user set represented as $\mathcal{U} = \{i|i = 1, 2, \cdots, N^u\}$. Each user utilizes medical sensors to real-time collect physiological data, generating medical tasks aimed at providing high-quality healthcare services. These medical tasks are represented by the set $\mathcal{T}_i = \{j|j = 1, 2, \cdots, N_i^t\}$, where N_i^t indicates the number of tasks for user i. Each task can be represented by a triplet (D, C, P), where D denotes the data size of the task, C represents the computational complexity required to process the task (measured in computational cycles), and P indicates the priority of the task, reflecting the level of demand for higher priority tasks necessitate greater service quality.

To meet users' service demands for low delay, low energy consumption, and high reliability, computationally intensive tasks need to be processed in real-time and efficiently. Although local devices have a certain level of computational frequency F_i, overloading them with excessive tasks will make it challenging for these tasks to be completed within a limited timeframe T^{max}. Moreover, local processing increases energy consumption, significantly shortening device lifecycles.

Meanwhile, within the physical network layer, there are N^m MEC servers poised to offer computational support to users in close proximity. This collection of edge servers is symbolized as $\mathcal{M} = \{k|k = 1, 2, \cdots, N^m\}$. Users have the flexibility to offload computational tasks onto these edge servers or process them locally, thereby easing the local computational load and improving the QoS for users.

1) Computing Model: User devices and MEC servers need to allocate diverse resources for various tasks to meet differentiated service demands of users. Subsequently, the execution delay $T_{i,j}^{exe}$ and execution energy consumption $E_{i,j}^{exe}$ for the task can be expressed as follows:

$$T_{i,j}^{exe} = \frac{C_{i,j}}{f_{i,j}}, \tag{1}$$

$$E_{i,j}^{exe} = k_0 f_{i,j}^2 C_{i,j}. \tag{2}$$

where k_0 represents a constant related to energy consumption, while $f_{i,j}$ represents the allocation of computational resources to task j of user i. These resources are assigned to the local device during local processing and to the server during offloaded processing.

2) Communication Model: During the transmission process of offloaded tasks, MEC servers will also reasonably allocate communication resources for accessing users' devices to meet the diverse demands of offloaded tasks. The uplink transmission rate of task can be expressed using the Shannon's theorem:

$$rate_{i,j} = b_{i,j} \log_2 \left(1 + \frac{g_{i,k}^2 P_i^{tr}}{b_{i,j} N_0}\right), \tag{3}$$

where the DT-driven IoMT employs frequency division multiple access for transmission, with interference between tasks being disregarded. $b_{i,j}$ represents the channel bandwidth allocated for each task. $g_{i,k}$ denotes the channel gain between user device i and MEC server k, and N_0 represents Gaussian noise. P_i^{tr} signifies the transmission power of user i, neglecting transmission queue delays. It is assumed that once an offloading task acquires sufficient communication resources, it can be transmitted with the transmission power P_i^{tr}.

This paper employs a channel gain model based on Rayleigh fading and path loss for the IoMT wireless communication environment [32]. The channel gain $g_{i,k}$ comprises two components: path loss and Rayleigh fading. Path loss PL considers the distance between users and servers as well as the propagation environment, while Rayleigh fading $h_{i,k}$ describes small-scale fading variations, modeled as $h_{i,k} \sim \mathcal{CN}\left(0, \sigma^2\right)$. The Rayleigh fading model accurately characterizes channel fading under multipath effects, enabling optimized resource

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

allocation and communication strategies. The channel gain $g_{i,k}$ can be expressed as:

$$g_{i,k} = \frac{h_{i,k}}{PL\left(d_{i,k}\right)} = \frac{h_{i,k}}{PL_0 + 10\eta\log_{10}\left(d_{i,k}\right)}, \tag{4}$$

where *PL* denotes path loss, PL_0 is the path loss at reference distance d_0, $d_{i,k}$ is the distance between user device *i* and server *k*, and η is the path loss exponent. The signal-to-noise ratio (SNR) $\gamma_{i,k}$ is defined as:

$$\gamma_{i,k} = \frac{P_i^{tr}}{\left(PL_0 + 10\eta\log_{10}\left(d_{i,k}\right)\right)^2 b_{i,j} N_0}. \tag{5}$$

The expected transmission rate $\mathbb{E}\left[rate_{i,j}\right]$ can then be expressed as:

$$\begin{aligned}\mathbb{E}\left[rate_{i,j}\right] &= b_{i,j} \cdot \mathbb{E}\left[\log_2\left(1 + \gamma_{i,k}\left|h_{i,j}\right|^2\right)\right] \\ &= \frac{b_{i,j}}{\ln 2} \cdot e^{\frac{1}{\gamma_{i,k}\sigma^2}} \cdot E_1\left(\frac{1}{\gamma_{i,k}\sigma^2}\right).\end{aligned} \tag{6}$$

Consequently, the transmission delay and energy consumption of the task can be expressed as:

$$T_{i,j}^{tr} = \frac{D_{i,j}}{\mathbb{E}\left[rate_{i,j}\right]}, \tag{7}$$

$$E_{i,j}^{tr} = P_{i,j}^{tr} T_{i,j}^{tr}. \tag{8}$$

3) Control Model: The offloading strategy for a task is defined as $s_{i,j}$. When $s_{i,j} = 0$, it indicates that the task is executed locally, while when $s_{i,j} \in \mathcal{M}$, the task is offloaded to the corresponding MEC server for execution. Each user must control and optimize the offloading strategies for all tasks to enhance the overall QoS of the IoMT system. Therefore, the total processing delay and energy consumption of the task can be expressed as:

$$T_{i,j} = \left(s_{i,j} \in \mathcal{M}\right) T_{i,j}^{tr} + T_{i,j}^{exe}, \tag{9}$$

$$E_{i,j} = \left(s_{i,j} \in \mathcal{M}\right) E_{i,j}^{tr} + \left(s_{i,j} \notin \mathcal{M}\right) E_{i,j}^{exe}. \tag{10}$$

B. Twin Network Layer

The twin network layer serves as the core of the DT. Through real-time data collection from the physical network layer, it maps elements such as entities, functionalities, and connection relationships to construct a virtual representation, providing a comprehensive understanding of the actual IoMT. Simultaneously, through simulation and analysis, the twin network layer can offer users real-time information feedback, helping enhance user policy-making capabilities and improve the performance of medical services, as well as the utilization efficiency of system resources.

The twin network layer is primarily composed of three modules: data warehouse module, service mapping module, and twin management module. The definitions and functionalities of each module are presented below:

1) Data Warehouse Module: The data warehouse module DW_i stores information collected at each time step during user interactions with the environment from the physical network layer, encompassing information about physical entity configurations, operational statuses, and user task details. However, IoMT devices face significant challenges due to limited storage and processing capabilities. For example, health data from wearables, like heart rate and step counts, can reach tens of megabytes per day, while storage is typically limited to a few hundred megabytes, making long-term data storage and processing challenging. Moreover, this health data often contains highly sensitive personal information (such as medical diagnoses and user habits), requiring secure and privacy-preserving approaches for storage and sharing.

To address these challenges, the data warehouse module employs a strategy of local storage, edge collaboration, and cloud-tiered storage. Recent data is encrypted and stored locally, medium-term data is offloaded to edge nodes, and long-term data is securely archived in the cloud. TLS encryption, differential privacy, and access control further enhance data security and privacy.

2) Service Mapping Module: The service mapping module is the core of the twin network layer, serving as a crucial hub that connects both upper-layer application services and lower-layer physical network entities in the DT. Through in-depth simulation and validation, it can provide reliable intelligent services to the network application layer, ensuring the QoS for tasks. Simultaneously, through intelligent interaction, the service mapping module feeds critical information back to the physical network layer, optimizing user policies.

Each user's service mapping module can collaborate with its data warehouse module DW_i, using local or edge servers for encrypted storage to address the limited storage and computational capabilities of IoMT devices. Intelligent agent models utilize generative AI, conducting replay training independently based on the data warehouse, and exposing only interaction interfaces in a P2P manner to ensure user data privacy. Additionally, the environmental service mapping module, stored on upper-layer servers or distributed storage, builds virtual IoMT environments, enabling real-time, precise reflection of the physical network layer and ensuring high system visibility.

3) Twin Management Module: The twin management module undertakes the management functions of the twin network layer. This module is responsible for managing the information exchange between various twin mappings, coordinating the computational offloading process of medical tasks, and ensuring the collaborative operation and efficient management.

In addition, model management is also one of the functions of the twin management module. This module can employ effective collaborative strategies to share and integrate users' observation information. Through information integration and collaborative strategies, the twin management module can effectively enhance cooperation among users, ensuring more intelligent and efficient policy-making, thereby maximizing the performance of medical services.

C. Network Application Layer

The network application layer serves as the user interface for the DT, aiming to support applications in the DT-driven IoMT and meet users' demands for medical services. This layer provides a user-friendly interface, enabling interaction with applications to conveniently achieve goals such as network optimization, policy improvement, and efficient computational offloading. Additionally, by inputting user requirements

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

through the northbound interface of the twin network layer and instantiating twin mappings in the service mapping module to simulate behavior, the network application layer assists users in gaining a deeper understanding of the operational mechanisms and conducting various simulation experiments.

IV. Problem Statement

In this section, a joint optimization problem is formulated based on the system model to achieve the collaborative management of communication, computation, and control, aiming to jointly minimize the delay and energy consumption of medical tasks. Subsequently, a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) model is constructed to mathematically formalize RL. This model will serve as the foundation for the DRL algorithm in the next section, aiming to achieve intelligent solutions to the optimization problem.

A. Problem Formulation

Based on the system model, a multi-objective function optimization problem *P* is formulated to minimize the processing delay and energy consumption of all tasks among all users. The goal of the optimization problem is to maximize the QoS for medical tasks as much as possible. The specific definition is as follows:

$$P: \begin{cases} \min T^{all}\left(\mathbf{s},\mathbf{f},\mathbf{b}\right) = \sum_{i\in\mathcal{U}} T_i \\ \min E^{all}\left(\mathbf{s},\mathbf{f},\mathbf{b}\right) = \sum_{i\in\mathcal{U}} E_i \end{cases}$$
$$s.t. \begin{cases} C_1: s \in \{0\}\cup\mathcal{M}, \forall s\in\mathbf{s} \\ C_2: \sum_{j'=1}^{n_i^t} f_{j'} \le F_i, \forall i\in\mathcal{U} \\ C_3: \sum_{j'=1}^{n_k^t} f_{j'} \le F_k, \forall k\in\mathcal{M} \\ C_4: \sum_{j'=1}^{n_k^t} b_{j'} \le B_k, \forall k\in\mathcal{M} \\ C_5: \sum_{t=1}^{T}\sum_{j}^{N_i^t} E_{t,i,j} \le Cap_i, \forall i\in\mathcal{U} \\ C_6: T_{i,j} \le T^{\max}, \forall i\in\mathcal{U}, \forall j\in\mathcal{T}^i \end{cases} \tag{11}$$

where $\mathbf{s}$, $\mathbf{f}$, and $\mathbf{b}$ correspond to the offloading strategy, computational resource allocation, and communication resource allocation for all tasks across all users. Moreover, the system's overall delay T^{all} and energy consumption E^{all} are obtained by summing each user's individual delay T_i and energy consumption E_i. The individual delay T_i and energy consumption E_i are closely related to each task's priority $P_{i,j}$ and its corresponding delay $T_{i,j}$ and energy consumption $E_{i,j}$, as represented by the following equation:

$$T_i = \sum_{j\in\mathcal{T}_i}\left(c\cdot e^{\alpha(P_{i,j}-\mu)}\cdot\left(1-\gamma^t T_{i,j}\right)\right), \tag{12}$$

$$E_i = \sum_{j\in\mathcal{T}_i}\left(c\cdot e^{\alpha(P_{i,j}-\mu)}\cdot\left(1-\gamma^e E_{i,j}\right)\right), \tag{13}$$

where c is a constant that adjusts the scale of the overall task reward, α is the coefficient that defines the relationship between priority and reward, and μ is the baseline priority. γ^t and γ^e are the delay and energy consumption decay coefficients, respectively. These equations ensure that tasks with higher priority have higher values upon completion.

For the constraint conditions, C_1 requires that the offloading strategies for all tasks fall within the set $\{0\}\cup\mathcal{M}$, indicating that tasks are either processed locally by user devices or offloaded to servers for processing. C_2, C_3, and C_4 correspond to resource allocation, ensuring that allocated resources do not exceed the total system resource limits. Here, F_i and F_k denote the total computing resources of user *i*'s device and server *k*, respectively, while B_k represents the total communication bandwidth resources available at server *k*. Additionally, n_i^t and n_k^t correspond to the total number of tasks processed locally by user *i*'s device and offloaded to server *k*, respectively. C_5 requires that the energy consumption of all processed tasks from $t=1$ to $t=T$ should be lower than the device's energy capacity Cap_i, ensuring that devices can complete tasks without running out of energy. C_6 requires that the processing delay for all tasks should be less than the maximum tolerated delay T^{max}, ensuring tasks are completed on time and meet delay requirements.

B. Dec-POMDP Model

By analyzing the system model, the DT-driven IoMT exhibits a complex multi-agent scenario. In this scenario, each user can only observe limited states while needing to compete or collaborate with other users under limited resources. To facilitate efficient intelligent policy-making among multiple agents and effectively address the joint optimization problem *P*, this paper treats user-end devices capable of intelligent decision-making, such as hubs, smart wristbands, and other devices, as agents. These devices are not only capable of receiving users' physiological data and generating medical tasks but also possess computing caching abilities and intelligent functions, such as controlling task offloading and processing.

In the following sections of the paper, we will refer to IoMT users as agents in simpler terms and construct a Dec-POMDP model to model the system. This model can be represented by a tuple $\left(\mathcal{S},\{\mathcal{A}_i\}_i^{N^u},\mathcal{P},\mathcal{R},\{\Omega_i\}_i^{N^u},\mathcal{O},\gamma\right)$, with specific explanations as follows:

- $\mathcal{S}$ represents the state space of the entire multi-agent system, encompassing the states of all intelligent agents and the IoMT environment. The system transitions to a terminal state s_T if any agent's device depletes its energy or lacks sufficient energy to process tasks.
- $\mathcal{A}_i$ represents the action space of agent *i*, which includes the offloading strategy $\mathbf{s}_i$ for all tasks, as well as the allocated computational resource $\mathbf{f}_i$ and communication resource $\mathbf{b}_i$. This can be expressed as $A_i = \mathbf{s}_i\times\mathbf{f}_i\times\mathbf{b}_i$.
- $\mathcal{P}$ defines the state transition probability, $P\left(s'|s,a\right)$, representing the likelihood of transitioning from state s to s' after performing action a.
- $\mathcal{R}:\mathcal{S}\times\mathcal{A}\to\mathcal{R}$ represents the reward function. The agent receives a reward *r* when performing action *a* in state *s*. According to problem *P*, the reward function is defined as: $r_i = \sum_{j=1}^{N_i^t}\left(cost^{\max} - cost_{i,j}\right)$, where $cost_{i,j}$ is defined as the task cost and $cost^{\max} = \max cost_{i,j}$. To jointly optimize delay and energy consumption, the cost function is defined as $cost_{i,j} = \alpha_{i,j}^T T_{i,j} + \alpha_{i,j}^E E_{i,j}$, where

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

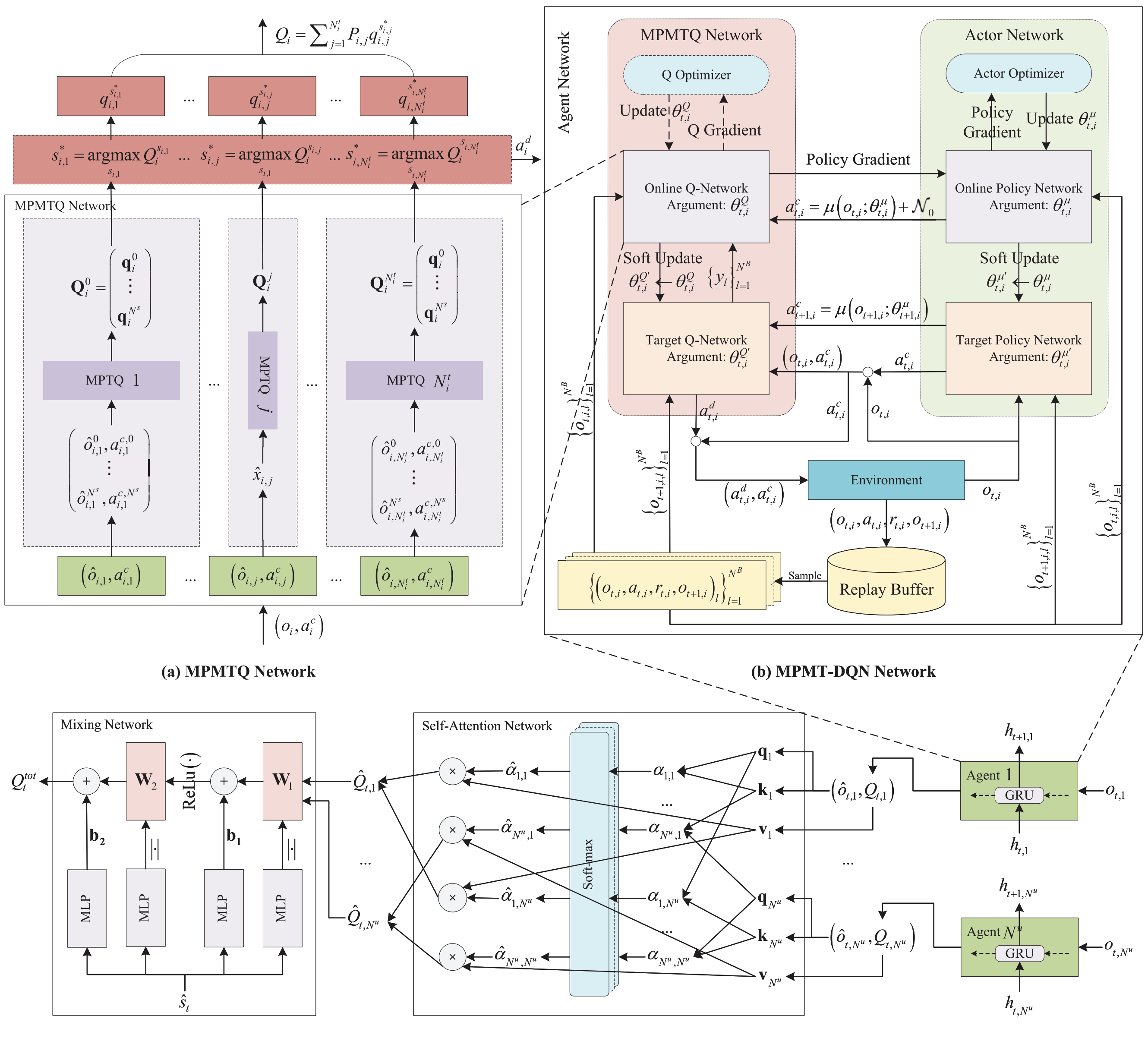

Fig. 2. Algorithm Model. Figure 2(a) shows the MPMTQ network, which addresses the parameterized action space of multiple tasks in agents through multiple MPTQ networks. Figure 2(b) shows the MPMT-DQN network, which utilizes the Actor-Critic architecture to achieve both offloading strategies and resource allocations of multiple tasks for agents. Figure 2(c) illustrates the SAE-QMIX network, which primarily facilitates multi-agent policy coordination. This network comprises three key components: the agent network, the self-attention mechanism module, and the Mixing network.

$T_{i,j}$ and $E_{i,j}$ are normalized, and $\alpha_{i,j}^T$ and $\alpha_{i,j}^E$ are the weights for delay and energy, respectively.

- Ω_i represents the limited observation space of agent i, which can be expressed as:

$$\Omega_i = \left(\{D_{i,j}\}_{j=1}^{N_i^t}, \{C_{i,j}\}_{j=1}^{N_i^t}, \{P_{i,j}\}_{j=1}^{N_i^t}\right) \times \left(P_i^{tr}, Cap_i, F_i\right) \times \left(\{B_k\}_{k=1}^{N^m}, \{F_k\}_{k=1}^{N^m}\right) \quad (14)$$

- $\mathcal{O}$ represents the observation probability, where the agent obtains observation o with the probability $P(o|s', a)$ given the state s' and action a.
- γ represents the discount factor.

V. Optimization Algorithm

In this section, the SAE-MT-QMIX algorithm is designed and comprehensively introduced for efficient computing offloading in DT-driven IoMT. Leveraging data support from the DT, SAE-MT-QMIX enables users to act as independent agents, aiming to resolve critical issues within the Dec-POMDP model. These challenges include the complexity of parameterized action spaces, decision-making under limited observations, and the cooperation and competition among multiple agents. By facilitating collaborative decision-making across communication, computation, and control in multi-user scenarios, the algorithm will significantly enhance the efficiency and reliability of IoMT systems, delivering real-time, high-quality medical services to users.

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

The framework of the algorithm is illustrated in Figure 2, which comprises three key components. First, Subfigure (a) showcases the task network architecture, focusing on addressing parameterized action space issues in multi-task scenarios while optimizing the challenges of high time and space complexity. Second, Subfigure (b) depicts the overall architecture of the agents, emphasizing deep collaboration among communication, computation, and control to enhance resource utilization and system adaptability. Finally, Subfigure (c) presents the IoMT architecture, which aims to coordinate control cooperation and competition among multiple agents enabling efficient task offloading and resource allocating.

A. MPMT-DQN: Handling Parameterized Action Spaces

Based on the system and Dec-POMDP models, the collaborative optimization of communication, computation, and control corresponds to the agent's allocation of communication resources $\mathbf{b}$, computation resources $\mathbf{f}$, and offloading strategies $\mathbf{s}$. Since the action space of the agents simultaneously encompasses discrete and continuous actions with potential interdependencies, this paper first proposes an efficient collaborative optimization network, MPMT-DQN, designed to efficiently handle the computing offloading problem within parameterized action spaces.

Figure 2(b) provides a detailed introduction to the network structure of the MPMT-DQN, which consists of an Actor network and a Multi-Pass Multi-Task Q-Network (MPMTQ). For the Actor network, at each time step t, agent i inputs its current observation $o_{t,i}$ into the network. Subsequently, the Actor network outputs the computing resource allocation $\mathbf{f}_{t,i}$ and communication resource allocation $\mathbf{b}_{t,i}$ for all tasks of the agent. Therefore, the Actor network can be described as the deterministic policy $a^c_{t,i} = \mu_i\left(o_{t,i}; \theta^{\mu}_{t,i}\right)$, where $\theta^{\mu}_{t,i}$ represents the Actor network parameters of agent i. For the MPMTQ network, it takes the agent's observation $o_{t,i}$ and continuous action $a^c_{t,i} = (\mathbf{f}_{t,i}, \mathbf{b}_{t,i})$ as inputs to obtain the agent's Q-value. This network can be represented as $Q_i\left(o_{t,i}, a^c_{t,i}; \theta^Q_{t,i}\right)$, where $\theta^Q_{t,i}$ represents the MPMTQ network parameters of agent i. The next subsection will provide a detailed description of the structure and functionality of the MPMTQ network.

For MPMT-DQN, considering the advantages of attention mechanisms in selective focus, enhancing the utilization of key features, and dynamically adjusting weights, this paper integrates Transformer-inspired Encoder structures into both the Actor network and the MPMTQ network, as illustrated in the Figure 3.

Specifically, the Actor network consists of two Encoder modules and one output module. The first Encoder layer captures the relationships between observation $o_{t,i}$ and incorporates a GRU layer to further enhance the ability to process and remember historical observations. This design enables efficient handling of uncertainties and dynamics in IoMT environments, ensuring robust information flow despite limited agent observations. The second Encoder layer focuses on integrating historical observations with the current observation, and the output module generates the communication and computation coordination strategy $(\mathbf{f}_{t,i}, \mathbf{b}_{t,i})$.

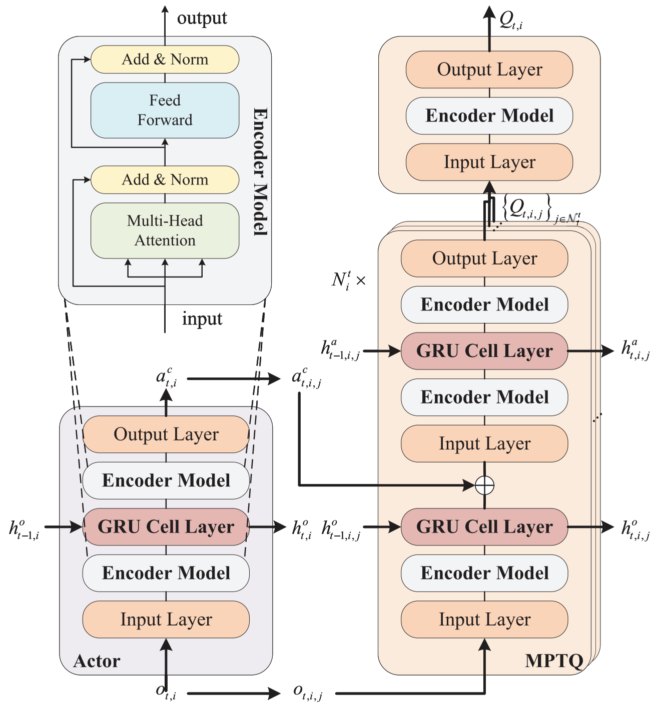

Fig. 3. Agent network model.

The MPMTQ network features a similar structure but with more layers, including three Encoder modules and one output module. The first and third Encoder layers are designed similarly to those in the Actor network. The second Encoder layer processes the joint information of observation $o_{t,i}$ and continuous action $a^c_{t,i}$, leveraging a GRU layer to manage sequential data of historical observations and actions. The third Encoder and output module then generate each agent's Q-value Q_i, enabling precise control strategies.

This multi-level Encoder-based architecture enhances the ability to process and retain historical information, providing efficient and accurate decision support in uncertain and dynamic IoMT environments. The agents' Q-values Q_i are ultimately integrated into a global Q-value Q via the upper-level mixing network detailed in Section V-C, supporting overall system optimization.

B. MPMTQ: Tackling Multi-Task Challenges

The MPMTQ network decomposes the agent's Q-values into N_i^t Task Q-values to address the parameterized action space of computing offloading with multiple tasks. As shown in Figure 2(a), MPMTQ consists of N_i^t Multi-Pass Task Q-Networks (MPTQ), with each MPTQ network dedicated to learning the Task Q-value for its corresponding task. Unlike the structure involving multiple Multi-Pass Q-Networks, MPTQ network takes as input not only the observation $o_{t,i,j}$ and continuous action $a^c_{t,i,j} = (f_{t,i,j}, b_{t,i,j})$ corresponding to task j, but also includes information $x_{t,i,j^-} = \left\{\left(o_{t,i,j'}, a^c_{t,i,j'}\right) | j' \in \mathcal{T}_i, j' \neq j\right\}$ from all other tasks and additional observation $o^{other}_{t,i}$ of the agent i to ensure potential correlations among tasks.

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

MPMTQ network also utilizes the multi-pass approach, but MPTQ network only needs to process the information $x_{t,i,j} = \left(o_{t,i,j}, a^c_{t,i,j}\right)$ of the corresponding task based on the offloading strategy $s_{t,i,j}$, without considering information of other tasks. In other words, the MPTQ network treats x_{t,i,j^-} and $o^{other}_{t,i}$ as part of the observation of the corresponding task. The corrected task observation $\hat{o}_{t,i,j}$ can be expressed as $\hat{o}_{t,i,j} = \left(o_{t,i,j}, x_{t,i,j^-}, o^{other}_{t,i}\right)$. Therefore, the input of MPTQ can be represented as follows:

$$\hat{x}_{t,i,j} = \begin{bmatrix} \hat{x}^0_{t,i,j} \\ \vdots \\ \hat{x}^{N^m}_{t,i,j} \end{bmatrix} = \begin{bmatrix} \left(\hat{o}^0_{t,i,j}, a^{c,0}_{t,i,j}\right) \\ \vdots \\ \left(\hat{o}^{N^m}_{t,i,j}, a^{c,N^m}_{t,i,j}\right) \end{bmatrix}, \tag{15}$$

where $\hat{o}^{s_{t,i,j}}_{t,i,j} = \left(o^{s_{t,i,j}}_{t,i,j}, x_{t,i,j^-}, o^{other}_{t,i}\right)$.

Each MPTQ network of agent i outputs a matrix $\mathbf{Q}^j_{t,i} = \left[\mathbf{q}^0_{t,i}, \cdots, \mathbf{q}^{N^m}_{t,i}\right]^\top$ of $m \times n$ dimensions, where each Task Q-value $q_{i^\mathbf{Q}, j^\mathbf{Q}}$ is an element of $\mathbf{Q}^j_{t,i}$, with $i^\mathbf{Q} \in \{1 \cdots m\}$ and $j^\mathbf{Q} \in \{1 \cdots n\}$. In $\mathbf{Q}^j_{t,i}$, the number of rows *m* corresponds to the number of passes, i.e., the number of available offloading strategies $N^m + 1$ for tasks, and the number of columns *n* corresponds to the tasks number N^i_t of the agent *i*. Each row vector $\mathbf{q}^{s_{t,i,j}}_{t,i}$ represents the Task Q-values of all tasks under the corresponding offloading strategy $s_{t,i,j}$, where $q_{i^\mathbf{Q}, (i^\mathbf{Q}-1)\boldsymbol{od}n+1}$ corresponds to the Task Q-value of task *j*. Therefore, the output of MPTQ can be represented as follows:

$$\mathbf{Q}^j_{t,i} = \begin{bmatrix} q^0_{t,i,j} & q^0_{t,i,0} & \cdots & q^0_{t,i,N^i_t} \\ q^1_{t,i,N^i_t} & q^1_{t,i,j} & \cdots & q^1_{t,i,N^i_t-1} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}. \tag{16}$$

Further, the MPTQ network selects the offloading strategy $s_{t,i,j}$ corresponding to the maximum agent's Q-value aggregated by Task Q-values as the optimal offloading strategy, rather than $s_{t,i,j}$ corresponding to the maximum Task Q-value of the task *j*. The optimal offloading strategy $s^*_{t,i,j}$ can be represented as follows:

$$s^*_{t,i,j} = \underset{s_{t,i,j}}{\operatorname{argmax}}\, Q^{s_{t,i,j}}_{t,i} = \underset{s_{t,i,j}}{\operatorname{argmax}} \sum\nolimits_{j' \in \mathcal{T}_i} P_{i,j'} q^{s_{t,i,j}}_{t,i,j'}, \tag{17}$$

where $Q^{s_{t,i,j}}_{t,i}$ represents the aggregated Q-value of the agent under the offloading strategy $s_{t,i,j}$.

Finally, by aggregating the optimal Task Q-value $q^{s^*_{t,i,j}}_{t,i,j}$ of each task under the optimal offloading strategy $s^*_{t,i,j}$ from all MPTQ networks, the Q-value $Q_{t,i}$ of the agent *i* can be obtained as follows:

$$Q_{t,i} = \sum\nolimits_{j \in \mathcal{T}_i} P_{i,j} q^{s^*_{t,i,j}}_{t,i,j} = Q_i\left(o_{t,i}, a^c_{t,i}; \theta^Q_{t,i}\right). \tag{18}$$

In summary, the objective of the MPMTQ network for the agent *i* is to evaluate $Q_{t,i}$. According to Equation (18), the agent's Q-value is aggregated from the optimal Task Q-value $q^{s^*_{t,i,j}}_{t,i,j}$ output by each MPTQ network that constitutes the MPMTQ network. The MPMTQ network can be trained by minimizing the squared difference between $q^{s^*_{t,i,j}}_{t,i,j}$ and the target Task Q-value for all MPTQ networks.

Therefore, the loss function $L\left(\theta^Q_{t,i,j}\right)$ of the MPTQ network corresponding to task *j* for agent *i* can be represented by the Equation (19), as shown at the bottom of the next page. Here, $\theta^Q_{t,i,j}$ represents the MPTQ network parameters, and $Q^j_{i,j}\left(\hat{o}_{t,i,j}, s_{t,i,j}, a^c_{t,i,j}; \theta^Q_{t,i,j}\right)$ represents the Task Q-value of task *j* output by the MPTQ network under corrected task observation $\hat{o}_{t,i,j}$ and task action $\left(s_{t,i,j}, a^c_{t,i,j}\right)$. Meanwhile, the target Task Q-value $y_{t,i,j}$ can be represented by Equation (20), as shown at the bottom of the next page, which is obtained through the Temporal Difference algorithm (TD).

Then, the goal of the Actor network is to maximize the Q-value $Q_{t,i}$ of the agent as much as possible. Similarly, training the Actor network can be achieved by maximizing all aggregated agent's Q-values $\left\{Q^{s_{t,i,j}}_{t,i} | j \in \mathcal{T}_i\right\}$ under all possible offloading strategies $\{s_{t,i,j} | s_{t,i,j} = 0, \cdots N^m\}$ using the Task Q-values $\left\{Q^{j'}_{i,j}(\cdot) | j' \in \mathcal{T}_i\right\}$ output from all MPTQ networks. The loss function of the Actor network can be represented as shown in Equation (21), as shown at the bottom of the next page.

C. SAE-QMIX: Achieving Multi-Agent Cooperative Learning

To tackle the challenge of multi-user collaborative control, this paper introduces the Self-Attention-Enhanced QMIX (SAE-QMIX) algorithm, which utilizes a QMIX-based Mixing network for the nonlinear aggregation of agent Q-values. The Mixing network effectively models dynamic user interactions, ensuring global Q-value optimization while enhancing collaborative decision-making across communication, computation, and control tasks.

Furthermore, the SAE-QMIX introduces the self-attention mechanism between the mixing network and agent networks, enabling dynamic adjustment of the weights of individual local values to obtain the global value, providing a more accurate representation of interactions among agents. As shown in Figure 2(c), the agent Q-values $\{Q_{t,i} | i \in \mathcal{U}\}$ obtained through MPMT-DRQN, along with the corrected agent observations $\{\hat{o}_{t,i} | i \in \mathcal{U}\}$, which exclude user privacy information, will first be input into the self-attention network to compute the weights of each agent's Q-value. The weight matrix $\hat{\mathrm{A}}$ for the agent's Q-values $\mathbf{v} = \left[Q_{t,1}, \cdots, Q_{t,N^u}\right]^\top$ is represented as follows:

$$\hat{\mathrm{A}} = \operatorname{softmax}(\mathrm{A}) = \operatorname{softmax}\left(\frac{\mathbf{k}^\top \mathbf{q}}{\sqrt{d_k}}\right), \tag{22}$$

where $\mathbf{q} = \mathbf{W}^q[(\hat{o}_{t,1}, Q_{t,1}), \cdots, (\hat{o}_{t,N^u}, Q_{t,N^u})]^\top$ represents the query vector, $\mathbf{k} = \mathbf{W}^k[(\hat{o}_{t,1}, Q_{t,1}), \cdots, (\hat{o}_{t,N^u}, Q_{t,N^u})]^\top$ represents the key vector, and $\sqrt{d_k}$ denotes the scaling factor. In addition, $\mathbf{W}^q$ and $\mathbf{W}^k$ correspond to the weight matrices of the networks mapping $\mathbf{q}$ and $\mathbf{k}$ respectively.

Then, the weighted agent Q-values is represented as follows:

$$\hat{\mathbf{Q}} = \left[\hat{Q}_{t,1}, \cdots, \hat{Q}_{t,N^u}\right]^\top = \hat{\mathrm{A}}\mathbf{v}. \tag{23}$$

The weighted Q-values $\hat{\mathbf{Q}}$ of all agents will further input into the mixing network. At the same time, the weights $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2]$ and biases $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2]$ of the network are generated from weight and bias networks, where the limited global state $\hat{s}_t$ serves as the input. It's worth noting that $\hat{s}_t$

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

is unrelated to user privacy. The representation of the global Q-value Q_t^{tot} is as follows:

$$Q_t^{tot} = \mathbf{W}_2^{\top}\mathrm{ReLu}\left(\mathbf{W}_1^{\top}\hat{\mathbf{Q}} + \mathbf{b}_1\right) + \mathbf{b}_2, \tag{24}$$

where $\mathrm{ReLu}(\cdot)$ is the activation function.

Typically, the value decomposition algorithm needs to satisfy the Individual Global Max (IGM) condition to ensure the consistency between maximizing local values for all agents and the global value. To satisfy the IGM condition, it is necessary to ensure that the derivative of the global value with respect to the Q-values of all agents is greater than zero, as shown below:

$$\frac{\partial Q_t^{tot}}{\partial Q_{t,i}} = \frac{\partial Q_t^{tot}}{\partial \hat{Q}_{t,i}}\frac{\partial \hat{Q}_{t,i}}{\partial Q_{t,i}} > 0, \quad \forall i \in \mathcal{U}, \tag{25}$$

where the mixing network ensures that the derivative of Q_t^{tot} with respect to all input $\hat{Q}_{t,i}$ is greater than zero. Simultaneously, the softmax function within the self-attention network guarantees that all weights $\hat{\alpha} \in \hat{\mathrm{A}}$ are positive, ensuring that the derivative of $\hat{Q}_{t,i}$ with respect to $Q_{t,i}$ is also greater than zero. Therefore, it can be demonstrated that the SAE-QMIX algorithm satisfies the IGM condition.

Finally, similar to updating agent networks, loss function $L\left(\theta_t^Q\right)$ to train the mixing network, the self-attention network, and all agent's MPMTQ networks, as well as loss function $L\left(\theta_{t,i}^{\mu}\right)$ for training agent's Actor network to maximize the global Q-value Q_t^{tot}, can be represented as Equation (26) and Equation (27) respectively:

$$L\left(\theta_t^Q\right) = \mathbb{E}\left[\left(y_t - Q^{tot}\left(\mathbf{o}_t, \mathbf{a}_t, \hat{s}_t; \theta_t^Q, \left\{\theta_{t,i}^{\mu'} | i \in \mathcal{U}\right\}\right)\right)\right], \tag{26}$$

$$L\left(\theta_{t,i}^{\mu}\right) = -\mathbb{E}\left[Q^{tot}\left(\mathbf{o}_t, \mathbf{a}_t, \hat{s}_t; \theta_t^{Q'}, \left\{\theta_{t,i}^{\mu} | i \in \mathcal{U}\right\}\right)\right]. \tag{27}$$

where θ_t^Q denotes the parameters of the networks in Equation (26). And the target global Q-value y_t can be shown as follows:

$$y_t = r_t + \gamma \max_{\mathbf{a}_{t+1}} Q^{tot}\left(\mathbf{o}_{t+1}, \mathbf{a}_{t+1}, \hat{s}_{t+1}; \theta_t^{Q'}, \left\{\theta_{t,i}^{\mu'} | i \in \mathcal{U}\right\}\right). \tag{28}$$

To sum up, with DT-driven IoMT, the proposed SAE-MT-QMIX algorithm conducts model training from both the agents' and the system's perspectives. As shown in Algorithm 1, this approach can not only meet the demands of personalized medical services but also enhance the overall performance of IoMT system.

TABLE II
SYSTEM PARAMETERS

Parameters	Value
Data size $D_{i,j}$	$20 \sim 50$ KB
Computing size $C_{i,j}$	$10^5 \sim 2 \times 10^5$ cycles
Computing frequency of IoMT users F_j	0.1 GHz
Computing frequency of MEC servers F_k	1.0 GHz
Bandwidth B_k	1.0 GHz
Transmitting power P_j^{tr}	$0.2 \sim 0.4$ W
Gaussian noise N_0	4^{-21} W

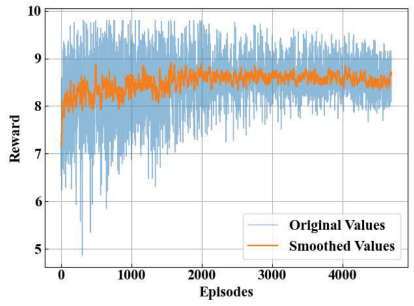

Fig. 4. The convergence curve of MPMT-DQN algorithm.

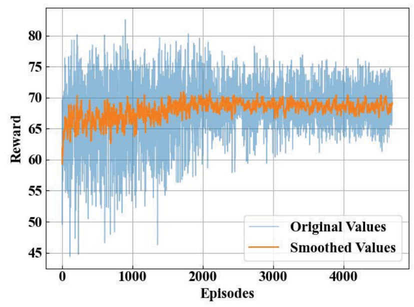

Fig. 5. The convergence curve of SAE-QMIX algorithm.

VI. Performance Evaluation

In this section, this paper simulates the proposed algorithm and analyzes the simulation results. The DT-driven IoMT simulation environment is constructed based on Python 3.9, with the DRL model implemented using the PyTorch framework. The primary evaluation focuses on comparing the performance of the proposed algorithm with baseline algorithms in terms of delay and energy consumption of medical tasks. Key parameters are shown in TABLE II.

A. Convergence Analysis

First, the convergence analysis of the MPMT-DQN algorithm and SAE-QMIX algorithm is carried out, and their convergence curves are shown in Figure 4 and Figure 5 respectively. With the increase of the number of training

$$L\left(\theta_{t,i,j}^Q\right) = \mathbb{E}\left[\left(y_{t,i,j} - Q_{i,j}^j\left(\hat{o}_{t,i,j}, s_{t,i,j}, a_{t,i,j}^c; \theta_{t,i,j}^Q\right)\right)^2\right], \tag{19}$$

$$y_{t,i,j} = r_{t,i,j} + \gamma Q_{i,j}^j\left(\hat{o}_{t+1,i,j}, s_{t+1,i,j}^*, \mu_i\left(o_{t+1,i}; \theta_{t,i}^{\mu'}\right)[(2j-1):2j]; \theta_{t,i,j}^{Q'}\right), \tag{20}$$

$$L\left(\theta_{t,i}^{\mu}\right) = -\mathbb{E}\left[\sum_{j=1}^{N_i^t}\sum_{s_{t,i,j}=0}^{N^m}\sum_{j'=1}^{N_i^t} P_{i,j'} Q_{i,j}^{j'}\left(\hat{o}_{t,i,j}, s_{t,i,j}, \mu_i\left(o_{t+1,i}; \theta_{t,i}^{\mu}\right)[(2j-1):2j]; \theta_{t,i,j}^{Q'}\right)\right]. \tag{21}$$

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 SAE-MT-QMIX

```
Ensure:
    Initialize DT-driven IoMT.
    Initialize all experience replay buffers $\{\mathcal{B}_i | i \in \mathcal{U}\}$ for all agents and global replay buffer $\mathcal{B}$.
    Initialize the network parameters $\theta$ and $\theta'$ for all online networks and target networks respectively.
Require:
    The policy $\pi_i$ for each agent $i \in \mathcal{U}$.
1:  for $episode = 1$ to $M$ do
2:      Reset DT-driven IoMT.
3:      Obtain observation $o_{0,i}$ for each agent and the limited global state $\hat{s}_0$.
4:      for $t = 1$ to $T$ do
5:          Each agent selects an action $a_{t,i}$ based on its current policy $\pi_{t,i}$ with noise $\mathcal{N}_0$ according to MPMT-DQN.
6:          Obtain the global reward $r_t$ and limited global state $\hat{s}_{t+1}$, as well as the corresponding reward $r_{t,i}$ and observation $o_{t+1,i}$ for each agent.
7:          Store the transition $(\hat{s}_t, a_t, r_t, \hat{s}_{t+1})$ to $\mathcal{B}$, and transition $(o_{t,i}, a_{t,i}, r_{t,i}, o_{t+1,i})$ of each agent $i \in \mathcal{U}$ to corresponding $\mathcal{B}_i$.
8:          for each agent $i \in \mathcal{U}$ do
9:              Sample a random min-batch of $N^B$ transitions $\{(o_{t',i}, a_{t',i}, r_{t',i}, o_{t'+1,i})_l\}_{l=1}^{N^B}$ from $\mathcal{B}_i$.
10:             Update network parameters $\theta_i^{\mu}$ of online Actor network by minimizing Eq. (21), and network parameters $\theta_{i,j}^{Q}$ of each online MPTQ network $j \in \mathcal{T}_i$ by minimizing Eq. (19).
11:             Soft update the target Actor network and all target MPTQ networks.
12:         end for
13:         Sample a min-batch of $N^B$ transition sequences $\left\{\left([(o_{t',i}, a_{t',i}, r_{t',i}, o_{t'+1,i})]_{t'=1}^{T}\right)_l\right\}_{l=1}^{N^B}$ from $\mathcal{B}$.
14:         Update network parameters $\theta_i^{\mu}$ of each online Actors network $i \in \mathcal{U}$ by minimizing Eq. (27), and network parameters $\theta^{Q}$ of the mixing network, the self-attention network, and all agent's MPMTQ networks by minimizing Eq. (26).
15:         Soft update the target mixing network, the target self-attention network and all target agent networks.
16:     end for
17: end for
```

iterations, the rewards of the MPMT-DQN algorithm and the SAE-QMIX algorithm gradually increase and tend to be stable. The MPMT-DQN algorithm converges after about 1500 iterations, and the SAE-QMIX converges after about 2000 iterations.

B. MPMT-DQN Algorithm Analysis

Following, a detailed analysis of the performance of the MPMT-DQN algorithm will be conducted. This part will focus on the performance of the agent algorithm, with a fixed number of agent users set to 8, and Value-Decomposition Networks (VDN) employed as the agent value aggregation algorithm. The MPMT-DQN algorithm will be compared with the following algorithms to evaluate performance in terms of delay and energy consumption:

- *PA-DDPG [33]:* Derived from the DDPG algorithm, the PA-DDPG algorithm addresses the challenge of parameterized action spaces by transforming discrete actions into a continuous space.
- *MT-PDQN [34]:* Multiple parallel PDQN algorithms independently train all task policies of each agent, and then merge the task values into the agent's value.
- *MT-MP-DQN [35]:* Similar to MT-PDQN, but solves the problem of parameterized action spaces for multi-task agents using multiple parallel MP-DQN algorithms.

First, the time and space complexity of the proposed algorithms are analyzed. The PDQN and MP-DQN algorithms require considering all task-server combinations, with output dimensions of $(N_i^t)^{(N^m+1)}$ and $(N_i^t)^{2(N^m+1)}$, resulting both in time and space complexities of $O\left((N_i^t)^{(N^m+1)}\right)$ and $O\left((N_i^t)^{2(N^m+1)}\right)$, respectively. These complexities grow exponentially with the number of tasks and servers. Consequently, this paper does not consider parameterizing all task offloading decisions simultaneously.

If tasks are parameterized independently (i.e., MT-PDQN and MT-MP-DQN algorithms), the time and space complexities become $O\left(N_i^t \cdot (N^m+1)\right)$ and $O\left((N_i^t \cdot (N^m+1))^2\right)$, respectively. While this reduces the overall complexity, the potential relationships between tasks are not fully captured, which may impact performance. In contrast, the PA-DDPG algorithm directly handles offloading strategies for all tasks, with time and space complexities of $O\left(3 \cdot N_i^t\right)$, which results in a smaller computational cost, but it struggles to converge and achieve optimal solutions.

The proposed MPMT-DQN algorithm strikes a balance between time and space complexities while fully considering task interdependencies, improving multi-tasks collaborative control efficiency and achieving better performance. For the algorithms proposed above, under the conditions of one batch, one time step, four multi-head attention mechanisms, and a fixed setup of 10 tasks and 4 edge servers, the FLOPs and model parameters for each module and the overall agent network model are shown in the Table III.

TABLE III
FLOPs AND PARAMETERS OF AGENT MODEL

Networks	FLOPs	Parameters
Actor (PA-DDPG)	56312.0	55822.0
Critic (PA-DDPG)	130772.0	129213.0
Actor (MT-PDQN)	55672.0	55172.0
Q-Network (MT-PDQN)	232760.0	233779.0
Actor (MT-MP-DQN)	55672.0	55172.0
Q-Network (MT-MP-DQN)	871720.0	236059.0
Actor (MPMT-DQN)	**55672.0**	**55172.0**
Q-Network (MPMT-DQN)	**887720.0**	**239309.0**
PA-DDPG	187084.0	185035.0
MT-PDQN	288432.0	288951.0
MT-MP-DQN	927392.0	291231.0
MPMT-DQN	**943392.0**	**294481.0**

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

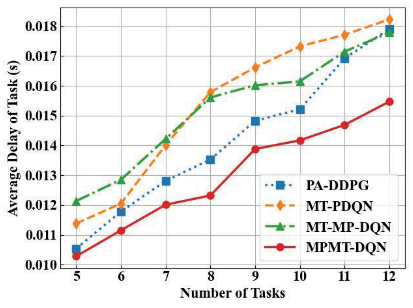

Fig. 6. Average delay for different numbers of tasks.

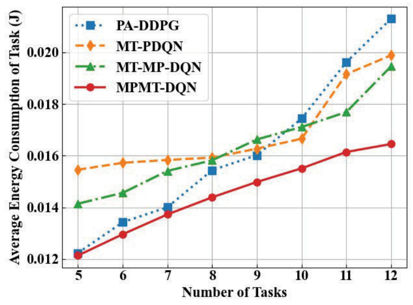

Fig. 7. Average energy consumption for different numbers of tasks.

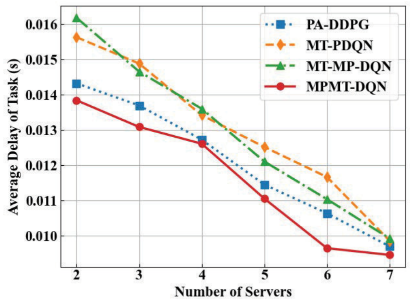

Fig. 8. Average delay for different numbers of servers.

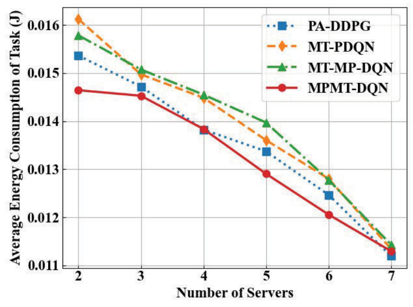

Fig. 9. Average energy consumption for different numbers of servers.

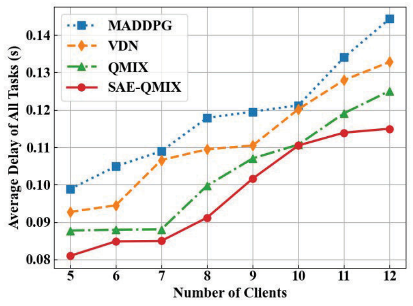

Fig. 10. Average delay for different numbers of clients.

By analyzing the delay and energy consumption under varying numbers of medical tasks with a fixed server quantity of 4, it is evident from Figures 6 and 7 that the proposed algorithm demonstrates certain advantages over other algorithms. The performances of MT-PDQN and MT-MP-DQN algorithms are lower due to a lack of analysis of the potential relationships between tasks. Although PA-DDPG algorithm has certain advantages in performance compared to the former two, as the number of tasks increases, the complexity of the model also increases, leading to higher noise levels and a decrease in training effectiveness. The proposed MPMT-DQN algorithm effectively addresses the issue of parameterized action spaces and avoids overlooking the interactions between tasks.

Furthermore, examining the scenario where the number of tasks is fixed at 8, the delay and energy consumption is analyzed across varying numbers of servers. As depicted in Figures 8 and 9, with an increase in the number of servers, more communication and computing resources become available for executing medical tasks, resulting in a gradual reduction in delay and energy consumption. Notably, the MPMT-DQN algorithm demonstrates a significant advantage in both delay and energy consumption compared to others.

C. SAE-QMIX Algorithm Analysis

Following, an analysis of the performance of the multi-agent algorithm SAE-QMIX will be conducted. This part will primarily focus on the performance of multiple agents in terms of competition and cooperation. Therefore, the number of user tasks will be fixed at 8, and MPMP-DRQN will be employed to implement the agent network. The algorithms for comparative analysis include:

- *MA-PA-DDPG [36]:* MA-PA-DDPG aims to address cooperation issues among multiple agents. It learns the policy of each agent using PA-DDPG to maximize global performance.
- *VDN [37]:* VDN is a multi-agent learning approach based on Q-value decomposition. It decomposes the Q-value of each agent into local value functions, then merges these values to generate a global value function.
- *QMIX [38]:* QMIX algorithm is a multi-agent learning algorithm suitable for parameterized action spaces. It employs a mixing networks to model relationships between agents and optimizes global Q-values.

Firstly, to analyze the impact of different MADRL algorithms on delay and energy consumption as the number of users increases (with 4 servers fixed). Figures 10 and 11 respectively illustrate the trends of delay and energy consumption as the number of users increases. The limited computational resources provided by MEC servers gradually become insufficient to meet the demands of medical task offloading, leading to a gradual decline in task executing performance. Specifically, MA-PA-DDPG experiences performance degradation due to handling all agent states and actions

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

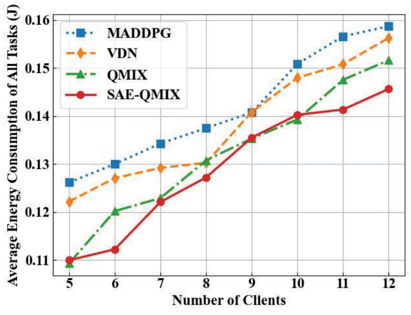

Fig. 11. Average energy consumption for different numbers of clients.

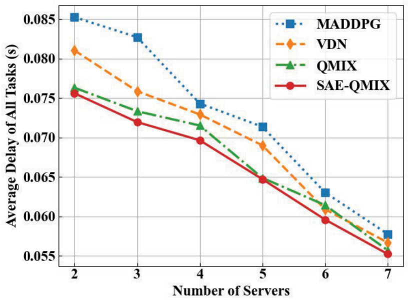

Fig. 12. Average delay for different numbers of servers.

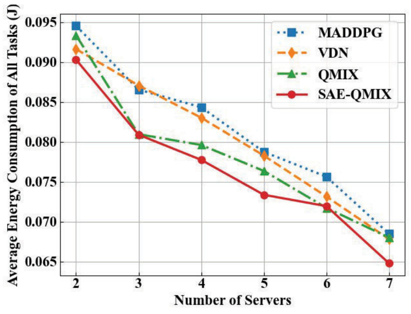

Fig. 13. Average energy consumption for different numbers of servers.

uniformly, a trend that becomes more pronounced as the number of agents increases. In contrast, the VDN algorithm merely combines local value functions into a global one, failing to fully leverage agent cooperation. While the QMIX algorithm shows some capacity for handling complex agent cooperation, its effectiveness can be improved by incorporating self-attention mechanisms to enhance its perception of the global state, improving its ability to manage agent relationships.

Next, with the number of users fixed, Figures 12 and 13 depict the trends of processing delay and energy consumption. As the number of MEC servers increases, more medical tasks are likely to be offloaded to MEC servers for processing, relieving local computational loads and providing additional computing resources to reduce resource competition among offloaded medical tasks. The proposed SAE-QMIX algorithm demonstrates certain advantages in both delay and energy consumption compared to other algorithms.

VII. Conclusion and Future Works

In this paper, the proposed MPMT-DQN and SAE-QMIX algorithms effectively address the multi-user communication, computation, and control collaborative decision-making problem in DT-driven IoMT, reducing the delay and energy consumption for providing medical services to all users. At the same time, these algorithms successfully tackle challenges such as parameterized action spaces and limited observations in multi-agent scenarios, achieving a balance between competition and cooperation in multi-agent control decision-making.

Future work will focus on deploying and testing the algorithms in real hardware environments to verify their performance in practical scenarios. Although existing multi-agent algorithms support data security by preventing the sharing of local experience pools, the upper-layer mixed network analysis still requires comprehensive evaluation of the agents' current states and rewards. A key challenge will be designing effective decision-making mechanisms for the upper-layer server while ensuring privacy protection. Furthermore, while Digital Twin technology currently exists primarily at the architectural level, providing a platform for collaborative control and information exchange training for agents, future work can enhance interaction mapping through generative AI. This will enable better fitting and feedback for each agent, further ensuring the security of user data in upper-layer interactions.

References

[1] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in Industrial Internet of Things: Architecture, advances and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2462–2488, 4th Quart., 2020.

[2] Z. Ning et al., "Mobile edge computing enabled 5G health monitoring for Internet of Medical Things: A decentralized game theoretic approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 463–478, Feb. 2021.

[3] H. R. Chi, C. K. Wu, N.-F. Huang, K.-F. Tsang, and A. Radwan, "A survey of network automation for industrial Internet-of-Things toward Industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 2065–2077, Feb. 2023.

[4] F. Tang, X. Chen, T. K. Rodrigues, M. Zhao, and N. Kato, "Survey on digital twin edge networks (DITEN) toward 6G," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1360–1381, 2022.

[5] B. F. Ji, Y. N. Wang, K. Song, and C. G. Li, "A survey of computational intelligence for 6G: Key technologies, applications and trends," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7145–7154, Oct. 2021.

[6] X. Yuan, J. Chen, N. Zhang, J. Ni, F. R. Yu, and V. C. M. Leung, "Digital twin-driven vehicular task offloading and IRS configuration in the Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24290–24304, Dec. 2022.

[7] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, "Digital twin assisted task offloading for aerial edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10863–10877, Oct. 2022.

[8] X. Yuan et al., "A MEC offloading strategy based on improved DQN and simulated annealing for Internet of behavior," *ACM Trans. Sensor Netw.*, vol. 19, no. 2, pp. 1–20, Dec. 2022.

[9] S. Peng, B. Li, L. Liu, Z. Fei, and D. Niyato, "Trajectory design and resource allocation for multi-UAV-assisted sensing, communication, and edge computing integration," *IEEE Trans. Commun.*, vol. 73, no. 4, pp. 2847–2861, Apr. 2025.

[10] A. Srivastava and S. M. Salapaka, "Parameterized MDPs and reinforcement learning problems—A maximum entropy principle-based framework," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9339–9351, Sep. 2022.

[11] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[12] Y. Zhang, J. Hu, and G. Min, "Digital twin-driven intelligent task offloading for collaborative mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3034–3045, Oct. 2023.

[13] R. Zhang, Z. Xie, D. Yu, W. Liang, and X. Cheng, "Digital twin-assisted federated learning service provisioning over mobile edge networks," *IEEE Trans. Comput.*, vol. 73, no. 2, pp. 586–598, Feb. 2024.

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

[14] B. Wang, Y. Sun, H. Jung, L. D. Nguyen, N.-S. Vo, and T. Q. Duong, "Digital twin-enabled computation offloading in UAV-assisted MEC emergency networks," *IEEE Wireless Commun. Lett.*, vol. 12, no. 9, pp. 1588–1592, Sep. 2023.

[15] Y. He, M. Yang, Z. He, and M. Guizani, "Computation offloading and resource allocation based on DT-MEC-assisted federated learning framework," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 6, pp. 1707–1720, Dec. 2023.

[16] Z. Qu, Y. Li, B. Liu, D. Gupta, and P. Tiwari, "DTQFL: A digital twin-assisted quantum federated learning algorithm for intelligent diagnosis in 5G mobile network," *IEEE J. Biomed. Health Informat.*, early access, Aug. 8, 2023, doi: 10.1109/JBHI.2023.3303401.

[17] Y. Tai et al., "Digital-twin-enabled IoMT system for surgical simulation using rAC-GAN," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 20918–20931, Nov. 2022.

[18] T. Liu, L. Tang, W. L. Wang, Q. B. Chen, and X. P. Zeng, "Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1427–1444, Jan. 2022.

[19] Z. Hu et al., "An efficient online computation offloading approach for large-scale mobile edge computing via deep reinforcement learning," *IEEE Trans. Services Comput.*, vol. 15, no. 2, pp. 669–683, Mar./Apr. 2022.

[20] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12240–12251, Oct. 2020.

[21] F. Chai, Q. Zhang, H. Yao, X. Xin, R. Gao, and M. Guizani, "Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7783–7795, Feb. 2023.

[22] P. Qin, S. Wang, Z. Lu, Y. Xie, and X. Zhao, "Deep reinforcement learning-based energy minimization task offloading and resource allocation for air ground integrated heterogeneous networks," *IEEE Syst. J.*, vol. 17, no. 3, pp. 4958–4968, Sep. 2023.

[23] F. Liu, H. Yu, J. Huang, and T. Taleb, "Joint service migration and resource allocation in edge IoT system based on deep reinforcement learning," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 11341–11352, Apr. 2024.

[24] Y. Ran, H. Hu, Y. Wen, and X. Zhou, "Optimizing energy efficiency for data center via parameterized deep reinforcement learning," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1310–1323, Mar. 2023.

[25] W. Ding, F. Luo, C. Gu, Z. Dai, and H. Lu, "A multiagent meta-based task offloading strategy for mobile-edge computing," *IEEE Trans. Cognit. Develop. Syst.*, vol. 16, no. 1, pp. 100–114, Feb. 2024.

[26] A. Suzuki, M. Kobayashi, and E. Oki, "Multi-agent deep reinforcement learning for cooperative computing offloading and route optimization in multi cloud-edge networks," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 4, pp. 4416–4434, Dec. 2023.

[27] W. Liu, B. Li, W. Xie, Y. Dai, and Z. Fei, "Energy efficient computation offloading in aerial edge networks with multi-agent cooperation," *IEEE Trans. Wireless Commun.*, vol. 22, no. 9, pp. 5725–5739, Sep. 2023.

[28] J. Huang et al., "Digital twin assisted DAG task scheduling via evolutionary selection Marl in large-scale mobile edge network," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2023, pp. 158–163.

[29] W. Hou, H. Wen, H. Song, W. Lei, and W. Zhang, "Multiagent deep reinforcement learning for task offloading and resource allocation in cybertwin-based networks," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16256–16268, Nov. 2021.

[30] F. Zhou, L. Feng, M. Kadoch, P. Yu, W. Li, and Z. Wang, "Multiagent RL aided task offloading and resource management in Wi-Fi 6 and 5G coexisting industrial wireless environment," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 2923–2933, May 2022.

[31] Z. Yao, S. Xia, Y. Li, and G. Wu, "Cooperative task offloading and service caching for digital twin edge networks: A graph attention multi-agent reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3401–3413, Nov. 2023.

[32] Y. Lyu, W. Wang, Y. Sun, and I. Rashdan, "Measurement-based fading characteristics analysis and modeling of UAV to vehicles channel," *Veh. Commun.*, vol. 45, Dec. 2023, Art. no. 100707. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209623001377

[33] L. Tan, S. Guo, P. Zhou, Z. Kuang, and X. Jiao, "HAT: Task offloading and resource allocation in RIS-assisted collaborative edge computing," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 5, pp. 4665–4678, Sep. 2024.

[34] Y. Xu, Y. Wei, K. Jiang, L. Chen, D. Wang, and H. Deng, "Action decoupled SAC reinforcement learning with discrete-continuous hybrid action spaces," *Neurocomputing*, vol. 537, pp. 141–151, Jun. 2023.

[35] Y. Yan, K. Du, L. Wang, H. Niu, and X. Wen, "MP-DQN based task scheduling for RAN QoS fluctuation minimizing in public clouds," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2022, pp. 878–884.

[36] F. Chai et al., "Multi-agent DDPG based resource allocation in NOMA-enabled satellite IoT," *IEEE Trans. Commun.*, vol. 72, no. 10, pp. 6287–6300, Oct. 2024.

[37] M. Meng, B. Hu, S. Chen, and S. Kang, "Dynamic beam pattern based on cooperation multi-agent VDN-D3QN for LEO satellite communication system," *IEEE Trans. Green Commun. Netw.*, vol. 9, no. 2, pp. 725–738, Jun. 2025.

[38] M. Zhang, W. Tong, G. Zhu, X. Xu, and E. Q. Wu, "SQIX: QMIX algorithm activated by general softmax operator for cooperative multiagent reinforcement learning," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 54, no. 11, pp. 6550–6560, Nov. 2024.

Xiaoming Yuan (Member, IEEE) received the Ph.D. degree in communication and information system from Xidian University, China, in 2018. From September 2016 to August 2017, she was a Visiting Scholar with the Broadband Communications Research (BBCR) Group, University of Waterloo, Waterloo, Canada. From December 2022 to December 2024, she was a Post-Doctoral Researcher with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. She is currently an Associate Professor with Qinhuangdao Branch Campus, Northeastern University, China. Her research interests include edge intelligence, network management, intelligent connected vehicles, and generative AI.

Hansen Tian received the B.E. degree in computer science and technology from Northeastern University, Shenyang, China, in 2022. He is currently pursuing the master's degree in computer science and technology with Qinhuangdao Branch Campus, Northeastern University. His research interests include edge intelligence and machine learning.

Xinling Zhang received the B.E. degree in computer science and technology from Liaoning University, Shenyang, China, in 2024. She is currently pursuing the master's degree in computer science and technology with Qinhuangdao Branch Campus, Northeastern University. Her research interests include edge computing and reinforcement learning.

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.

Hongyang Du (Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore. He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include edge intelligence, generative AI, and network management. He was a recipient of the IEEE ComSoc Young Professional Award for Best Early Career Researcher in 2024, the IEEE Daniel E. Noble Fellowship Award from the IEEE Vehicular Technology Society in 2022, the IEEE Signal Processing Society Scholarship from the IEEE Signal Processing Society in 2023, Singapore Data Science Consortium (SDSC) Dissertation Research Fellowship in 2023, and the NTU Graduate College's Research Excellence Award in 2024. He was recognized as an Exemplary Reviewer of IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE COMMUNICATIONS LETTERS. He serves as the Editor-in-Chief Assistant (2022–2024) and an Editor (since 2025) for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, an Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and the Guest Editor for *IEEE Vehicular Technology Magazine*.

Kaibin Huang (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from the National University of Singapore and the Ph.D. degree from The University of Texas at Austin. He is currently the Philip K H Wong Wilson K L Wong Professor of electrical engineering and the Head of the Department of Electrical and Electronic Engineering, The University of Hong Kong (HKU), Hong Kong. He is a member of the Engineering Panel of Hong Kong Research Grants Council (RGC) and a RGC Research Fellow (2021 Class). He is a fellow of U.S. National Academy of Inventors. His work was recognized with seven best paper awards from the IEEE Communication Society. He has served on the editorial boards for five major journals in the area of wireless communications and co-edited ten journal special issues. He has been active in organizing international conferences, such as the 2014, 2017, and 2023 editions of IEEE GLOBECOM, a flagship conference in communication. He has been named as a Highly Cited Researcher by Clarivate in the last six years (2019–2024) and an AI 2000 Most Influential Scholar (Top 30 in Internet of Things) in 2023–2024. He was an IEEE Distinguished Lecturer.

Ning Zhang received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2015. He is currently an Associate Professor and Canada Research Chair with the Department of Electrical and Computer Engineering, University of Windsor. After that, he was a Post-Doctoral Research Fellow with the University of Waterloo and the University of Toronto. His research interests include connected vehicles, mobile edge computing, wireless networking, and security. He is a member of the Royal Society of Canada College, a Distinguished Lecturer of IEEE, and a Highly Cited Researcher (Web of Science). He received several best paper awards from conferences and journals. He serves as the Chair for the IEEE Technical Committee on Cognitive Networks and the Vice Chair for the IEEE Technical Committee on Big Data. He also serves/served as a TPC/general chair for numerous conferences. He serves/served as an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE INTERNET OF THINGS JOURNAL, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.

Lin Cai (Fellow, IEEE) has been with the Department of Electrical and Computer Engineering, University of Victoria, since 2005, where she is currently a Professor. Her research interests include communications and networking, with a focus on network protocol and architecture design supporting ubiquitous intelligence. She has been a Board Member of the IEEE Women in Engineering (2022–2024) and the IEEE Communications Society (2024–2026). She is a Royal Society of Canada Fellow, an NSERC E. W. R. Steacie Memorial Fellow, an Engineering Institute of Canada Fellow, and a Canadian Academy of Engineering Fellow. She has been elected to serve the Board of the IEEE Vehicular Technology Society (2019–2027), and as its VP in Mobile Radio (2023–2025). She has served as an Associate Editor-in-Chief for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. She was a Distinguished Lecturer of the IEEE VTS Society and the IEEE Communications Society.

Authorized licensed use limited to: UNIVERSITY OF VICTORIA. Downloaded on November 03,2025 at 04:56:59 UTC from IEEE Xplore. Restrictions apply.