Energy-Efficient Optimization for Mobile Edge Computing With Quantum Machine Learning

James Adu Ansere, Dung T. Tran, *Graduate Student Member, IEEE*, Octavia A. Dobre, *Fellow, IEEE*, Hyundong Shin, *Fellow, IEEE*, George K. Karagiannidis, *Fellow, IEEE*, and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—We investigate the joint optimization problem of stochastic computation offloading, content caching strategy, and dynamic resource allocation to maximize the energy efficiency of mobile edge computing in Internet-of-Things. Specifically, we propose a quantum deep reinforcement learning algorithm to exponentially increase the caching learning speed and content caching delivery efficiency in multi-dimensional continuous and large action spaces. Furthermore, we utilize the modified Grover's algorithm with faster computation time to improve the processing efficiency and data-content retrieval for transition quantum state probabilities. The numerical results show that our proposed quantum machine learning scheme significantly outperforms other benchmarks in terms of energy-efficiency maximization subject to transmission power, energy consumption, and transmission latency.**

***Index Terms*—Quantum reinforcement learning, IoT, MEC.**

I. Introduction

IN THE rapidly evolving Internet-of-Things (IoT), where an ever-increasing number of heterogeneous devices are connected to generate a massive amount of data, efficient content delivery and management have become vital. One of the key challenges in such dynamic IoT environments is to ensure timely access to relevant content while minimizing network congestion, transmission latency, and energy consumption. This is where a well-designed content caching strategy for fast processing efficiency plays a critical role. Because IoT devices are battery-powered and resource-constrained with low computing capacity, they can hardly support compute-intensive, time-critical, and latency-sensitive ubiquitous applications and services, such as autonomous vehicles, robotics, augmented reality, and virtual reality. Due to the stochasticity and high dimensionality, the use of traditional optimization methods introduces complexities and challenges in task offloading and content caching to the mobile edge computing (MEC) server, making it computationally expensive for a dynamic IoT environment [1].

A joint strategy of task offloading, distributed resource allocation, and service caching has been optimized to reduce communication latency and energy consumption [2]. A joint optimization of computational task offloading and content caching algorithms was studied to minimize the weighted sum of task execution latency and energy consumption for users [3]. A computational task offloading problem in dynamic MEC-enabled IoT networks has been investigated to minimize the long-term average service cost by considering the power consumption and buffering delay [4]. In [5], a joint optimization problem for content caching strategy, offloading task, and resource allocation is proposed to minimize the average transmission latency in the IoT network. An actor-critic reinforcement learning (RL)-based algorithm is modelled as Markov decision process (MDP) to improve the joint decision-making problem to obtain a stochastic control policy. Notwithstanding, these algorithms pose crucial challenges in real-time scenarios since the joint optimization for high-dimensional discrete-continuous sequential decision problems is practically intractable and too complicated to be solved effectively.

Very recently, quantum machine learning (QML), which combines the principles of quantum mechanics with machine learning schemes, has gained tremendous attention in the industrial and academic communities. QML can potentially solve complex mathematical equations and generate simulations faster than their classical counterparts. In large-scale dynamic environments, where multifaceted computations can become increasingly complex, quantum-based techniques can offer a promising advantage in increasing processing speed [6]. QML exploits quantum superposition principles and quantum parallelism to explore multiple quantum states and evaluate different policy updates simultaneously. A deep reinforcement learning (DRL) with quantum experience replay has been proposed to achieve a better training paradigm compared to the traditional machine learning approach [7]. A quantum-inspired RL algorithm was considered to optimize the trajectory of unmanned aerial vehicle communications [8].

Manuscript received 6 November 2023; accepted 27 November 2023. Date of publication 5 December 2023; date of current version 8 March 2024. The work of Octavia A. Dobre was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery Program. The work of Hyundong Shin was supported by the National Research Foundation of Korea Grant funded by the Korean Government (MSIT) under Grant 2019R1A2C2007037 and Grant 2022R1A4A3033401. The work of T. Q. Duong was supported by the Canada Excellence Research Chair (CERC) Program. The associate editor coordinating the review of this article and approving it for publication was Q. Wu. *(Corresponding authors: Trung Q. Duong; Hyundong Shin.)*

James Adu Ansere is with Sunyani Technical University, Sunyani, Ghana (e-mail: jaansere@stu.edu.gh).

Dung T. Tran is with Duy Tan University, Da Nang 50000, Vietnam.

Octavia A. Dobre is with the Electrical and Computer Engineering Department, Memorial University of Newfoundland, St. Johns, NL A1B 3X5, Canada.

Hyundong Shin is with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, Seoul 02447, South Korea (e-mail: hshin@khu.ac.kr).

George K. Karagiannidis is with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece, and also with the Artificial Intelligence and Cyber Systems Research Center, Lebanese American University, Beirut 1102 2801, Lebanon (e-mail: geokarag@auth.gr).

Trung Q. Duong is with the Memorial University of Newfoundland, St. Johns, NL A1B 3X5, Canada, and also with Queen's University Belfast, BT7 1NN Belfast, U.K. (e-mail: tduong@mun.ca).

Digital Object Identifier 10.1109/LWC.2023.3338571

A quantum-based offloading strategy and communication resource allocation algorithm are proposed to reduce the task processing latency in MEC-based Internet-of-Vehicles (IoV) [9].

Nonetheless, in highly dynamic IoV environments, a massive number of vehicles generate significant data, which requires complex real-time adaptive decisions. Scaling a quantum-inspired DRL algorithm to handle a large volume of content is challenging. Again, the above-mentioned studies only solve the expected reward function maximization using the conventional DRL approaches, which are practically intractable. These approaches become challenging and unpredictable due to the high computational complexity of the increasingly large and multi-dimensional discrete action spaces. Very recently, a quantum computing-supported machine learning algorithm has been applied to solve the joint optimal design for communication and computation of MEC-based IoT systems [10], [11]. However, the work did not consider optimizing adaptive caching policies in real time, which can be challenging in highly dynamic IoT environments.

Inspired by the above-mentioned works, we jointly optimize dynamic resource allocation, content caching strategy, and computation offloading policy to maximize energy efficiency in dynamic large-scale IoT networks. In a quantum uncertainty environment, we propose a novel quantum DRL (QDRL) algorithm to exponentially improve the optimal content caching and data retrieval efficiency in multidimensional continuous actio

II. System Model and Problem Formulation

A. System Model

We consider a MEC-enabled system with L servers to provide computing services to K IoT devices connected to the base stations (BSs). Each IoT device $k \in \mathcal{K} = \{1, \ldots, K\}$ is wirelessly charged and its computationally intensive task is processed in a timely manner at each time interval $\tau \in \mathcal{T} = \{1, 2, \ldots\}$. Each computation task is either executed locally or offloaded and computed at the MEC server $l \in \mathcal{L} = \{1, \ldots, L\}$. The distributed MEC servers are mostly energy and resource constrained and have limited computational capabilities in complex, dynamic environments. However, dynamic IoT environments require real-time adaptation to changing content popularity, user preferences, and network conditions. With static policies and delayed cache updates, traditional caching predictions can become inaccurate, leading to inefficient caching decisions and suboptimal utilization of cache resources. Therefore, we integrate a quantum function into the MEC caching learning and caching processing speed for the dynamic environment.

B. Computation Model

The IoT device produces a computationally intensive and latency-sensitive task to proceed locally or offload to the MEC server. We elaborate the processes as follows.

1) Local Computing: Let $\mathcal{J} = \{1, 2, \ldots, j\}$ be the tasks in which the k-th IoT device can generate a j-th computation-intensive task of $\mathcal{C}(k, j) = ([\varphi_{k,j}(\tau)], [f_{k,j}(\tau)], [\varpi_{k,j}(\tau)])$ in the time slot τ, where $\varphi_{k,j}$ represents the input data (in bits), $f_{k,j}$ is the CPU frequency (in cycles) to complete the task and $\varpi_{k,j}$ is the processing latency (in ms). We assume $\varphi_{k,j} \in [0, 1]$, where $\varphi_{k,j} = 0$ means the IoT device executes the task locally. We define the task processing latency locally $\psi^l_{k,j}(\tau)$ for computing resources at the time slot τ as $\psi^l_{k,j}(\tau) = \frac{\lambda_k}{\mu_{k,j}(\tau)} + q_{k,j}(\tau)$, where $\mu_{k,j}(\tau)$ represents the computing resources; λ_k is the task required in CPU cycles, and $q_{k,j}(\tau)$ is the local queuing delay at time slot τ. Therefore, the energy consumed locally e_l within the time interval τ is given by $e_l(\tau) = \lambda_k(\tau) f_{l,\max}$, where $f_{l,\max}$ is the maximum computation capacity assigned through the uplink transmission power.

2) Computation Offloading: When $\varphi_{k,j} = 1$, we assume that the task is transferred to the MEC server for processing. Taking into account the time for data upload and execution, the offloading task latency $\psi^o_{k,j}(\tau)$ can be defined as $\psi^o_{k,j}(\tau) = \frac{d_k(\tau)}{\omega_{k,j}(\tau)}$, where $\omega_{k,j}(\tau)$ represents the task upload processing at time slot τ and $d_k(\tau)$ is the data upload. The corresponding energy consumption to offload the task to the MEC server in time slot τ is given as $e_o(\tau) = p_{k,j}(\tau)\psi^o_{k,j}(\tau)$, where $p_{k,j}(\tau)$ is the transmission power in time slot τ. Hence, the total power consumption $\mathcal{P}_T$ for task processing at time slot τ is defined as $\mathcal{P}_T(\tau) = P_c + \vartheta \sum_{k=1}^{K} \sum_{j=1}^{J} [e_l(\tau) + e_o(\tau)]$, where P_c is the circuit power consumption and $\vartheta \in \{0, 1\}$ indicates the power amplifier efficiency.

C. Caching Model

Since computation offloading between the IoT devices and the BSs causes an extra network communication latency and energy consumption, we model the caching decision as $\lambda_k(\tau) \in \{0, 1\}$ with respect to the IoT device at time slot τ. When $\lambda_k = 1$, the MEC server caches the task offload and $\lambda_k = 0$, otherwise. Thus, the storage capacity $\mathcal{C}_\nabla$ of the MEC server in caching of resources in the time slot τ can be expressed as $\mathcal{C}_\nabla(\lambda_k) = q_{k,j}(\tau) R^c_k(\tau)$, where $q_{k,j}(\tau)$ represents the caching content request rate in the time slot τ and $R^c_k(\tau)$ denotes the caching reward.

D. Communication Model

The MEC server divides the system bandwidth $\mathcal{W}$ into orthogonal subchannels for data transmission. Each subchannel is assigned a maximum one IoT device to avoid interference and minimize energy consumption in real-time scenarios. The maximum achievable data rate for uplink transmission can be expressed as

$$R_k(\tau) = \mathcal{W} \log_2 \left(1 + \frac{p_{k,j}(\tau) g_{k,j}(\tau)}{\alpha_{k,i} \sum_{k=1}^{K} \sum_{j=1, j \neq i}^{J} p_{k,i}(\tau) + \sigma_k^2} \right), \tag{1}$$

where $\mathcal{W}$ is the system bandwidth, $p_{k,j}(\tau)$ is the transmit power at time slot τ, $g_{k,j}(\tau)$ is the channel gain, $\alpha_{k,i}$ indicates the large-scale fading channel coefficient, and σ_k^2 is the complex Gaussian noise density.

Giving a limited computation constraints at each IoT device, the energy efficiency η in bit/Hz/Joules can be written in terms

of the power allocation $\mathcal{P}$, the user scheduling $\mathcal{U}$, and caching content strategy $\mathcal{C}_{\ddagger}$ at the time slot τ as follows:

$$\eta(\tau)\big[\mathcal{P},\mathcal{U},\mathcal{C}_{\ddagger}\big] = \frac{R_k(\tau)\big[\mathcal{P},\mathcal{U},\mathcal{C}_{\ddagger}\big]}{\mathcal{P}_T\big[\mathcal{P},\mathcal{U},\mathcal{C}_{\ddagger}\big]}. \tag{2}$$

E. Formulation of Offloading Task Problem

We aim to maximize the energy efficiency of the considered networks by jointly optimizing computational task offloading, content caching strategy, and dynamic resource allocation while maintaining latency requirements. Mathematically, the joint energy efficiency optimization problem can be written as follows.

$$\max_{(\mathcal{P},\mathcal{U},\mathcal{C}_{\ddagger})} \quad \eta_{EE}\big(\mathcal{P},\mathcal{U},\mathcal{C}_{\ddagger}\big) \tag{3a}$$

$$\text{s.t.} \quad 0 < p_{k,j}(\tau) \le P_{\max}, \forall k \in K, l \in L \tag{3b}$$

$$0 < r_{\min} \le R_k(\tau), \forall k \in K \tag{3c}$$

$$\sum_{k\in K} \lambda_k(\tau)\varphi_{k,j}(\tau) \le \mathcal{C}_{\nabla}, \forall k \tag{3d}$$

$$\sum_{k\in K} \lambda_{k,i}(\tau) \le 1, k \in K, \tag{3e}$$

$$\lambda_{k,i}(\tau) \in \{0,1\}, \forall i \in I, k \in K, \tag{3f}$$

where (3b) denotes the transmit power constraint $P_{\max}$; (3c) maintains the QoS requirement at IoT device and $r_{\min}$ is the minimum data rate requirement; (3d) means that cached contents must not exceed the storage capability $\mathcal{C}_{\nabla}$; (3e) and (3f) ensures the assignment of BS to handle the IoT devices. In dynamic large-scale IoT environments, (3a) is considered a mixed discrete-continuous nonconvex problem and NP-hard with no practical solutions. Applying conventional caching optimization techniques is much challenging with high computational costs and longer convergence time. Therefore, we propose a quantum-enhanced DRL-based caching algorithm to exponentially increase the processing efficiency and data retrieval capability in high-dimensional and complex search spaces.

III. Proposed QDRL Algorithm Design

Due to the intractability of (3a), the use of conventional cache optimization methods is challenging. Conventional cache optimization methods struggle to dynamically adapt to the exploration-exploitation trade-off as the environment changes. In this section, we investigate an innovative method that focuses on continuous learning and real-time adaptive decisions, making it more suitable for situations where the environment evolves over time.

A. Markovian Stochastic Computation Model

We transform the offload task problem in (3a) into an MDP form and solve it using the proposed QDRL algorithm to obtain a stochastic optimal policy control. MDP is defined in 4-tuple, expressed as $(\mathcal{S},\mathcal{A},\mathcal{R},P,\gamma)$ with $\mathcal{S}$, $\mathcal{A}$, $P(s,s')$ being the set of state space, action space, the transition state probability, respectively. We define the formulation of the MDP-based optimization problem as follows:

1. System state space: At each time step τ, the DRL agent detects the set of observations, denoted as $\mathcal{S}_\tau \in \mathbb{S}$; $s_k(\tau) = [p_{k,j}(\tau), f_{k,j}(\tau), \mathcal{W}_k(\tau)]$.

2. System action space: A DRL agent takes action $a_k \in [0,1]$; $a_k(\tau) = [\varphi_{k,j}(\tau), p_{k,j}(\tau), f_{l,\max}, \lambda_{k,i}(\tau)]$.

3. System reward function: The DRL agent uses a policy strategy of $\pi(\mathcal{A} \mid \mathcal{S}) = \mathcal{P}[\mathcal{A}_\tau \mid \mathcal{S}_\tau]$ to attain a state transition probability of $P(s_{\tau+1}|s_\tau, a_\tau)$ for the next state $\mathcal{S}_{\tau+1}$. In MDP, the value function $V^\pi(s)$ within a state s subject to a policy π, is the expected return for all $s \in \mathcal{S}$ and can be shown as

$$V^\pi(s) = \sum_a \pi(a \mid s)\left\{\sum_{s',r} P\big(s',r \mid s,a\big)\big[r + \zeta V^\pi\big(s'\big)\big]\right\}, \tag{4}$$

where $P(s'|s,a)$ is the probability function of the state transition matrix and ζ is the discount rate, i.e., $0 \le \zeta \le 1$. The DRL agent adapts the resource scheduling and unceasingly updates its strategy to maximize the expected rewards in the long term. Therefore, following a similar approach as in [10], the optimal action-value function Q^* can be written as

$$Q^*(s,a) = \sum_{s',r} P\big(s',r \mid s,a\big)\left\{r + \zeta \max_{a'} Q^\pi\big(s',a'\big)\right\}. \tag{5}$$

B. Quantum-Inspired Deep Reinforcement Learning Approach

1) Quantum States Representation of Quantum Machine Learning: Let $|\mathcal{S}\rangle = \sum_k \alpha_k |s_\tau\rangle$ and $|\mathcal{A}\rangle = \sum_k \beta_k |a_\tau\rangle$ respectively be the orthogonal set of eigenstates $|s_\tau\rangle$ and eigenactions $|a_\tau\rangle$, for computational basis 2^q and 2^q in the Hilbert spaces. Therefore, the unitary transformation can be written as

$$\left|\mathcal{S}_\tau^{(\omega)}\right\rangle \to \left|\mathcal{S}_\tau^{(q)}\right\rangle = \sum_{\mathcal{S}=00\ldots0}^{\overbrace{11\cdots1}^{q}} C_s|\mathcal{S}\rangle, \sum_{\mathcal{S}=00\ldots0}^{\overbrace{11\cdots1}^{q}} |C_s|^2 = 1, \tag{6}$$

$$\left|\mathcal{A}_\tau^{(\mu)}\right\rangle \to \left|\mathcal{A}_\tau^{(c)}\right\rangle = \sum_{\mathcal{A}=00\ldots0}^{\overbrace{11\cdots1}^{c}} C_a|\mathcal{A}\rangle, \sum_{\mathcal{A}=00\ldots0}^{\overbrace{11\cdots1}^{c}} |C_a|^2 = 1, \tag{7}$$

where ω and μ indicate the eigenstates and eigenactions, q and c are the number of qubits, respectively [6]. We set the quantum states $|0\rangle$ and $|1\rangle$, where the quantum superposition enables the representation of qubits in multiple states simultaneously and quantum parallelism accelerates content data processing, making IoT systems more responsive. Therefore, a system of q qubits can only assume the quantum states with probability amplitudes of $\{x\}_{x=0}^{2^q-1}$ and $\{x\}_{x=0}^{2^c-1}$, respectively.

2) Quantum Machine Learning-Based Policy and Updating: The intelligent agents select their actions to enhance real-time adaptive decisions in dynamic environments. The accumulated expected reward function maps the state-action space pair as $f(s) = \pi : S \to A_i$. From a unitary matrix based on the parallelism on quantum representation states 2^q, the temporal-difference to update value function V according to immediate reward is given as

$$V(s) \leftarrow V(s) + \varepsilon\big\{R + \zeta V\big(s'\big) - V(s)\big\}, \tag{8}$$

where ε denotes a learning rate and $\zeta \in \{0,1\}$.

C. Proposed QDRL Algorithm

Our proposed QDRL algorithm is based on the modified Grover's search algorithm design. In Grover's quantum algorithm, an element $\mathcal{N}$ is searched in an unsorted database in $O(\sqrt{\mathcal{N}})$ time, which is exponentially faster than classical algorithms that require $O(\mathcal{N})$ time and a quantum register to represent eigenfunctions of $\mathcal{N}$ is initialized. Under time-varying channel conditions, the Grover iterations obtain the optimal computation policy for unstructured search in a multi-dimensional large-scale IoT environment.

Let $\pi : \mathcal{A} \times \mathcal{S} \rightarrow [0,1], \pi(a,s) = \mathrm{P}(a_\tau = a \mid s_\tau = s)$ represent the policy mapping state to action. We assume that $f(\mathcal{A}_s) = |\mathcal{A}_\tau^{(\mu)}\rangle = |\mathcal{A}_\tau^{(c)}\rangle$ denotes the action space taken at the state, where the modified Grover algorithm is implemented with an action having a higher reward function to update the probability amplitude. Hence, we implement the modified Grover search algorithm via the following procedure

1) Initialization Phase: We define the unitary operator of oracle U_o on the computational basis state $|\phi\rangle$ as $U_o|\phi\rangle = (-1)^{f(\phi)}|\phi\rangle$. Let a system have $\mathcal{D} = 2^c$ quantum states representation of c qubit strings. First, we construct the equally weighted superposition of all eigenactions

$$H^{\otimes c}|\overbrace{00\cdots0}^{c}\rangle = \frac{1}{\sqrt{2^c}}\left(\sum_{\mathcal{A}=00\cdots0}^{\overbrace{11\cdots1}^{c}} |\mathcal{A}\rangle\right). \tag{9}$$

From the Hadamard gates [12] with initial states $|0\rangle$, we obtain

$$f(\mathcal{A}_s) = \left|\mathcal{A}_0^{(c)}\right\rangle = \frac{1}{\sqrt{2^c}}|\mathcal{A}\rangle + \sqrt{\frac{2^c-1}{2^c}}\left|\mathcal{A}^{\perp}\right\rangle, \tag{10}$$

where $\langle\mathcal{A}|\mathcal{A}_0^c\rangle = 1/\sqrt{2^c}$. We initialize the probability amplitude as $\frac{1}{\sqrt{2^c}}$ to improve the complex quantum state vector and the system observations.

2) Applying the Quantum Oracle: Let $|\phi\rangle \in \{0,1\}^c$ represent the oracle functions on computational states with a returned function output $f(\phi) = 1$. We construct the quantum oracle O_1 for the proposed algorithm by:

$$|\psi_1\rangle : \left|\mathcal{A}_\tau^{(c)}\right\rangle \overset{O_1}{\rightarrow} \frac{1}{\sqrt{2^c}}\sum_{\mathcal{A}=0}^{2^c-1}(-1)^{f(\mathcal{A})}|\mathcal{A}\rangle|\mathcal{A}^{\perp}\rangle, \tag{11}$$

where $|\mathcal{A}^{\perp}\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2^c}}$.

3) Using the Operator of Quantum Diffusion: Let $G_o = O_1(2|\psi_0^c\rangle\langle\psi_0^c| - I)$ signify the operator of Grover's iteration. By using the unitary transformation, the Grover iteration can be expressed as:

$$|\psi_2\rangle : |\psi_1\rangle \overset{G_o}{\rightarrow} [(2|\psi_0^c\rangle\langle\psi_0^c| - I)|\psi_1\rangle]^F \approx |A_s\rangle\left|\mathcal{A}^{\perp}\right\rangle, \tag{12}$$

where $\mathcal{F}$ denotes the Grover iteration time and I is the unit matrix. The Grover search algorithm $\mathcal{U}_{G_o}$ for $\mathcal{N}$ iterations over $|\psi_0^{\mathcal{N}}\rangle$, is given by:

$$\mathcal{U}_G\left|\psi_0^{\mathcal{N}}\right\rangle = \sin(2\mathcal{N}+1)|\mathcal{A}\rangle + \cos(2\mathcal{N}+1)\left|\mathcal{A}^{\perp}\right\rangle, \tag{13}$$

where $|\mathcal{A}^{\perp}\rangle = \sqrt{\frac{1}{2^d-1}}\sum_{\mathcal{A}\neq\mathcal{A}_j}|\mathcal{A}\rangle$, $\sin\theta \equiv \langle\psi_o|\psi_0^{\mathcal{N}}\rangle = 1/\sqrt{2^c}$. One Grover iteration rotates $|\mathcal{A}_s\rangle$ by 2θ. At $\sum_{\mathcal{A}}|\mathcal{C}_a|^2 = 1$, the probability amplitude is normalized after each updating.

Algorithm 1: Proposed QDRL Algorithm for Caching System

Input: $I \leftarrow$ unity matrix; $\psi \leftarrow$computational state basis
Output: $\mathcal{R}$ (accumulated expected reward)

Initialize $V(s), \left|\mathcal{S}_\tau^{(q)}\right\rangle, \left|\mathcal{A}_\tau^{(c)}\right\rangle$
Initialize $\tau = 0$
for $0 \leq \psi \leq \psi_{\max}$; $\psi == \psi_{\max}$; $\psi = \psi + 1$ **do**
  while $\tau > \psi$ **do**
    observe $f(s) = \left|A_\tau^{(c)}\right\rangle$ to obtain $|\mathcal{A}\rangle$.
    if $|\mathcal{A}\rangle$ *obtained, take the next state* $\left|S_\tau'\right\rangle$, *and* $\mathcal{R}$ **then**
      Construct the quantum states in (9)
      Compute the Hadamard transformation using (10)
    else
      Update the Grover's iterations of caching request $2\left|\psi_0^c\right\rangle\left\langle\psi_0^c\right| - I$
    Content caching $\mathcal{K}_\tau \leftarrow \mathcal{R}$ (reward).
    Update τ.

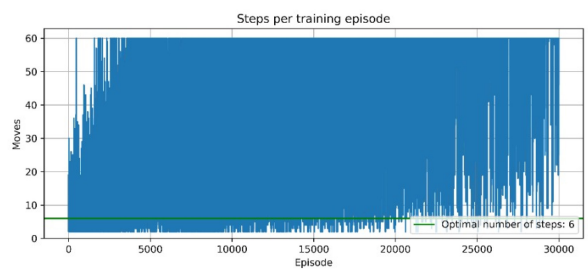

Fig. 1. The steps per training episode with learning ability.

By computing the eigenaction $|\mathcal{A}\rangle$, the probability amplitude $|\psi_0^c\rangle$ value can be updated through quantum collapse postulate to significantly improve quadratic speed. **Algorithm 1** illustrates the practical implementation of the proposed algorithm. However, the proposed algorithm faces the time complexity, coming from the Grover's iteration and updating of state-value $V(s)$. Hence, the total time complexity is given by $\mathcal{O}(\mathcal{N} \times (|\mathcal{N} \times O(\mathcal{D})| + |\sqrt{\mathcal{N}}|))$.

IV. Numerical Results and Discussions

In this section, we compare the performance of the proposed algorithm with benchmarks using a quantum virtual machine with the support of Python and Pytorch. For the system setup, the IoT devices are randomly distributed in a single cell radius of 200 m × 200 m, the bandwidth is 1 MHz, the energy of the IoT devices is initiated at 10 mJ, the transmission power is 2 dBm, and the SINR threshold is 16 dB [10].

Fig. 1 illustrates the learning performance of the proposed QDRL algorithm through the training episode. To rapidly optimize resource utilization, we integrated quantum features into a caching system with data processing to enhance real-time responsiveness. The caching strategy accelerates data

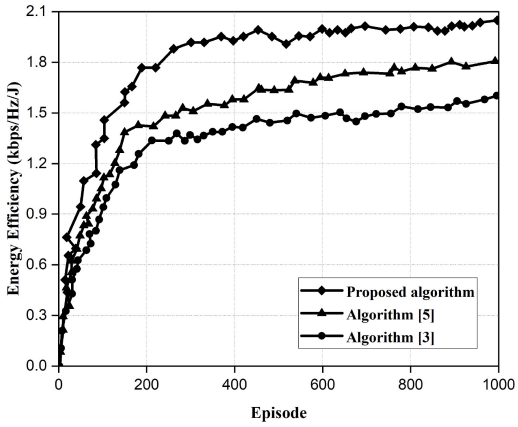

Fig. 2. The performance comparison between the proposed QDRL and other ML algorithms without quantum.

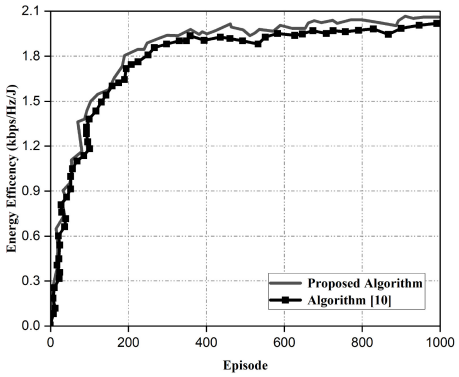

Fig. 3. The performance comparison between the proposed QDRL and quantum ML algorithm without caching.

retrieval by storing frequently requested content items closer to the end user, to ensure efficient content delivery and reduce transmission latency. Most importantly, in Fig. 1, we observe that as the reward function increases with the number of the episodes, to obtain optimal number of content retrieval and stochastic policy control in dynamic IoT environments.

In Fig. 2, we compare the energy efficiency performance of the proposed scheme and the two conventional approaches, namely, the JCOCC (a joint optimization of computational task offloading and content caching) algorithm [3] and the JCORA (a joint optimization problem for content caching strategy, offloading task, and resource allocation) algorithm [5]. As can be clearly observed from Fig. 2, the energy efficiency performance improves when the number of training episodes increases. By leveraging the modified Grover's iterations, our proposed solution optimally accelerates content caching delivery and data retrieval efficiency. Thus, it achieves a higher energy efficiency performance compared to the other conventional algorithms.

Fig. 3 shows the performance of energy efficiency in terms of training episodes in the proposed QDRL algorithm (that is, with caching strategy) against the existing quantum algorithm (that is, without caching) and the JOCC (joint communications and computation without caching) algorithm in [10]. Initially, we observe that all algorithms achieve similar energy performance. However, as the number of episodes increases, the proposed algorithm achieves higher energy efficiency to improve the performance of content caching.

V. Concluding Remarks

We investigated the computational task offloading problem, with the aim of jointly optimizing dynamic resource allocation, computational offloading, and content caching strategies to maximize the system's energy efficiency while satisfying the IoT users' transmission latency requirement. To improve task processing and data retrieval performance, we proposed a quantum DRL algorithm to exponentially increase the processing efficiency to obtain a stochastic optimal policy for high-dimensional state-action spaces in time-varying wireless channel conditions. The numerical results demonstrated that the proposed algorithm converges faster and achieves better energy efficiency performance in various MEC-enabled IoT scenarios compared to conventional algorithms.

References

[1] J. A. Ansere et al., "Optimal computation resource allocation in energy-efficient edge IoT systems with deep reinforcement learning," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 4, pp. 2130–2142, Dec. 2023.

[2] S. Bi, L. Huang, and Y.-J. A. Zhang, "Joint optimization of service caching placement and computation offloading in mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4947–4963, Jul. 2020.

[3] Z. Chen, Z. Chen, and Y. Jia, "Integrated task caching, computation offloading and resource allocation for mobile edge computing," in *Proc. IEEE Glob. Commun. Conf.*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[4] H. Hu, D. Wu, F. Zhou, S. Jin, and R. Q. Hu, "Dynamic task offloading in MEC-enabled IoT networks: A hybrid DDPG-D3QN approach," in *Proc. IEEE Glob. Commun. Conf.*, Madrid, Spain, Dec. 2021, pp. 1–6.

[5] W. Yifei, R. F. Yu, S. Mei, and H. Zhu, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor–critic deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019.

[6] T. Q. Duong, J. A. Ansere, B. Narottama, V. Sharma, O. A. Dobre, and H. Shin, "Quantum-inspired machine learning for 6G: Fundamentals, security, resource allocations, challenges, and future research directions," *IEEE Open. J. Veh. Technol.*, vol. 3, pp. 375–387, Aug. 2022.

[7] W. Qing, M. Hailan, C. Chunlin, and D. Daoyi, "Deep reinforcement learning with quantum-inspired experience replay," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9326–9338, Sep. 2022.

[8] Y. Li, A. H. Aghvami, and D. Dong, "Intelligent trajectory planning in UAV-mounted wireless networks: A quantum-inspired reinforcement learning perspective," *IEEE Wireless Commun. Lett.*, vol. 10, no. 9, pp. 1994–1998, Sep. 2021.

[9] D. Wang, B. Song, P. Lin, F. R. Yu, X. Du, and M. Guizani, "Resource management for edge intelligence (EI)-assisted IoV using quantum-inspired reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12588–12600, Jul. 2022.

[10] J. A. Ansere, T. Q. Duong, S. R. Khosravirad, V. Sharma, A. Masaracchia, and O. A. Dobre, "Quantum deep reinforcement learning for 6G mobile edge computing-based IoT systems," in *Proc. Int. Wireless Commun. Mobile Comput. Conf.*, Marrakesh, Morocco, Jun. 2023, pp. 1–6.

[11] J. A. Ansere, E. Gyamfi, V. Sharma, H. Shin, O. A. Dobre, and T. Q. Duong, "Quantum deep reinforcement learning for dynamic resource allocation in mobile edge computing-based IoT systems," *IEEE Trans. Wireless Commun.*, early access, Nov. 14, 2023, doi: 10.1109/TWC.2023.3330868.

[12] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.