




Backscatter-Aided Hybrid Data Offloading for Mobile Edge Computing via Deep Reinforcement Learning

Yutong Xie^{1,2}, Zhengzhuo Xu³, Jing Xu³, Shimin Gong⁴ , and Yi Wang^{5,6}

¹ Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences, Beijing, China
evnxie@foxmail.com

² University of Chinese Academy of Sciences, Beijing, China

³ School of Electronic Information and Communications,
Huazhong University of Science and Technology, Wuhan, China
1157567638@qq.com, xujing@hust.edu.cn

⁴ School of Intelligent Systems Engineering,
Sun Yat-sen University, Shenzhen, China
gong0012@e.ntu.edu.sg

⁵ SUSTech Institute of Future Networks,
Southern University of Science and Technology, Shenzhen, China
wangy37@sustc.edu.cn

⁶ Pengcheng Laboratory, Shenzhen, China

Abstract. Data offloading in mobile edge computing (MEC) allows the low power IoT devices in the edge to optionally offload power-consuming computation tasks to MEC servers. In this paper, we consider a novel backscatter-aided hybrid data offloading scheme to further reduce the power consumption in data transmission. In particular, each device has a dual-mode radio that can offload data via either the conventional active RF communications or the passive backscatter communications with extreme low power consumption. The flexibility in the radio mode switching makes it more complicated to design the optimal offloading strategy, especially in a dynamic network with time-varying workload and energy supply at each device. Hence, we propose the deep reinforcement learning (DRL) framework to handle huge state space under uncertain network state information. By a simple quantization scheme, we design the learning policy in the Double Deep Q-Network (DDQN) framework, which is shown to have better stability and convergence properties. The numerical results demonstrate that the proposed DRL approach can learn and

The authors would like to thank the anonymous reviewers for their valuable comments. This work is partially supported by an NSFC project grant (ref. no. 61872420), and the project of “PCL Future Regional Network Facilities for Large-scale Experiments and Applications (ref. no. PCL2018KP001)”. The work of Shimin Gong was supported in part by National Science Foundation of China (NSFC) under Grant 61601449, 61503368, and the Shenzhen Talent Peacock Plan Program under Grant KQTD2015071715073798.

converge to the maximal energy efficiency compared with other baseline approaches.

Keywords: Deep reinforcement learning · Double DQN · Computation offloading · Backscatter communications

1 Introduction

Mobile edge computing (MEC) provides the IoT devices in the network edge with cloud-like computation capability at the easy-to-access and resource-rich MEC servers, which can be integrated with the wireless access points or small-cell base stations [11]. The edge devices (e.g., wireless sensor nodes) are allowed to offload sensing data and computation tasks (e.g., data compressing and encryption) to the MEC servers, and then the MEC servers return the processed data for fulfilling the application requests at the edge devices. Data offloading of IoT devices is conventionally achieved by wireless RF communications, which is inherently power consuming by using RF communication radios (referred to as the active radios) to generate RF carrier signals [6]. The high power consumption in active radios may not be affordable by low-power edge devices and hence prevents them from using the MEC servers. Hence, the edge devices have to optimally balance the use of precious energy supply, depending on the channel conditions, energy status, and the users' workloads.

Wireless backscatter is recently introduced as novel communication technology with extremely low power consumption. The backscatter radios operate in *passive* mode by modulating and reflecting the incident RF signal via load modulation [1]. The passive radios are featured with low power consumption and low data rate [8]. Whereas the active radios can transmit in a higher data rate by adapting the transmit power against the channel fading. Hence, we expect to achieve a radio diversity gain by switching data offloading in two radio modes, e.g., [5] and [14]. In this paper, we consider a hybrid data offloading scheme combining local computation, passive and active offloading in a wireless powered MEC scenario, which allows a more flexible control to balance the power consumption in computation and offloading. The critical problem is to determine the optimal time scheduling and workload allocation strategies in each computation scheme, taking into account the time-varying channel conditions, energy supply, workload dynamics, and various resource constraints [2].

Due to network dynamics and close couplings among different network entities, the optimization of MEC offloading strategy become very challenging as the dimensionality and complexity rapidly increase. It is further complicated by the interactions among multiple wireless users, base stations, and MEC servers [10]. For example, different wireless users may compete for resources (e.g., channel, and computation capacities) to fulfill individuals' computation workloads. To deal with those complexities, we propose the model-free DRL-based framework to learn the optimal MEC offloading strategy with uncertain network information. We observe that the MEC offloading decisions are generally continuous

variables. By a simple quantization and encoding scheme, we turn the strategy space into a finite discrete set and then design the learning policy in the double deep Q-network (DDQN) framework to stabilize the learning process. Our numerical results verify that the proposed DDQN framework can achieve the maximum energy efficiency compared to other baseline approaches.

2 System Model

We consider a wireless edge network consisting of one hybrid access point (HAP) and N user devices that can sense and process data independently. To assist their data processing, the user edge devices can offload their sensing data to the HAP, which is co-located with an MEC server. The MEC server will return the processed data to the edge devices after the completion of computation workload. The system model is depicted in Fig. 1. We assume that the MEC server has enhanced computation capability and persistent power supply. Its computation and transmission of results can be performed instantly. Let $\mathcal{N} = \{1, 2, \dots, N\}$ denote the set of all edge nodes and S_i denote the i -th edge node for $i \in \mathcal{N}$. Each node is equipped with single antenna capable of harvesting energy from the HAP. The complex uplink and downlink channels between HAP and node S_i are denoted by $h_i \in \mathcal{C}$ and $g_i \in \mathcal{C}$, respectively. Each S_i is allocated a time slot t_i for its data offloading and capable of energy harvesting in other time slots. The workload of each edge node S_i is given by L_i , which is defined as the number of data bits to be processed either locally or remotely at the MEC center. We assume that the workload of each device is generated at the beginning of each time slot, and it has to be processed before the end of data frame.

2.1 Hybrid Data Offloading Scheme

The data offloading from each edge node to the HAP or MEC server can be performed in either passive backscatter communications or the conventional active RF communications, depending on its energy profile and the channel conditions. The switch between passive and active mode can be achieved by tuning the load impedance, e.g., [7]. As each edge node has only one antenna, we assume that it can only transmit in one radio mode or harvest energy from the HAP. Each edge node can switch its radio mode according to this channel conditions and energy status. As such, we further divide each time slot t_j allocated to S_j into two sub-slots, as shown in Fig. 1(b). One sub-slot $t_{a,j}$ is used for data offloading in active mode and the other sub-slot $t_{p,j}$ is for backscatter communications. The active data offloading is powered by the energy harvested from the HAP over consecutive time slots. While the passive data offloading is powered by power-splitting (PS) scheme, i.e., a part of the incident RF signals, denoted by the PS ratio ρ_j , is harvested to power the operation of backscatter radio, and the other part $1 - \rho_j$ is modulated and instantly reflected back to the HAP. Besides data offloading, the data computation can be also performed locally at the edge devices. In this case, the computation can be parallel to data offloading, as shown in Fig. 1(b).

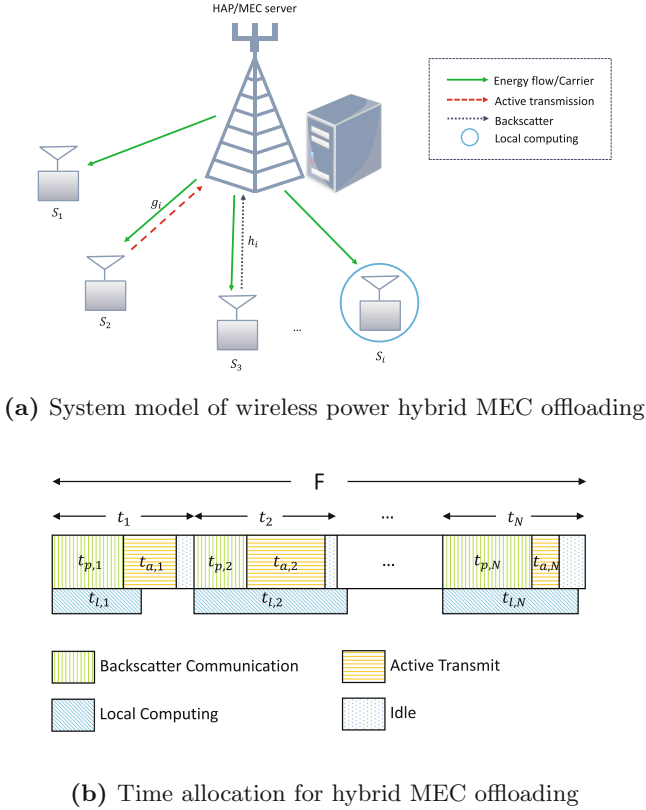


Fig. 1. Backscatter-aided hybrid MEC offloading scheme.

2.2 Workload Allocation

The workload generated in each time slot can be allocated among local computation, active and passive offloading. Note that different computation schemes have different processing capabilities and power consumption. Hence, the design of optimal MEC offloading scheme aims to divide the workload into three schemes, according to the dynamics in workload, channel conditions, and the energy supply of each edge device.

Active Offloading. Let $p_{a,i}$ denote the transmit power in active data offloading. The received signal at HAP is given by $y = \sqrt{p_{a,i}}h_i s(t) + \nu_d$, where $s(t)$ denotes the information with unit power and $\nu_d \sim \mathcal{CN}(0, \sigma^2)$ denotes the noise at the HAP. Then, the data rate in active mode can be denoted by

$$r_{a,i} = B \log_2 (1 + p_{a,i}|h_i|^2/\sigma^2), \tag{1}$$

where B denotes the bandwidth of active data transmission. The relationship between $p_{a,i}$ and $r_{a,i}$ is given by:

$$p_{a,i} = \beta(r_{a,i}) \triangleq \left(2^{r_{a,i}/B} - 1\right) \sigma^2 / |h_i|^2. \quad (2)$$

Hence, the total power consumption in active mode is given by $\tilde{\beta}(r_{a,i}) \triangleq \beta(r_{a,i}) + p_{c,i}$, where $p_{c,i}$ denotes the constant power to excite the circuit.

Passive Offloading. For passive data offloading, the data rate can be viewed as a constant, i.e., $r_{p,i} = r_p$, which relates to the ambient symbol rate and the signal detection scheme at the receiver [8]. Typically the backscatter communications rate r_p is less than that of active RF communications. However, power consumption for backscatter communications can be significantly less than the active RF communications and sustainable via wireless energy harvesting. In particular, a part of the incident RF power can be harvested to power the circuit of backscatter radio [9]. This implies that the edge device prefers to use high rate RF communications when energy is sufficient, and turns to backscatter communications if energy becomes insufficient.

Local Computation. The edge device can also perform local computation in parallel with MEC offloading, similar to [11]. Let f_i denote the processor's computing speed (cycles per second) and $0 \leq t_{l,i} \leq F$ denote the time for local computation. Here F can be the total number time slots during one data frame. Then, the amount of information bits processed locally by the edge node is given by $f_i t_{l,i} / \phi$, where $\phi > 0$ denotes the number of cycles needed to process one bit of task data. The energy consumption of local computation is constrained by $k_i f_i^3 t_{l,i} \leq E_i$ [3], where k_i denotes the coefficient of computation energy efficiency. To maximize the data processing capability, each edge device should exhaust the harvested energy and perform computation throughout the data frame. Hence, we have $f_i^* = \left(\frac{E_i}{k_i F}\right)^{\frac{1}{3}}$ and the local computation rate (in bits per second) is given by $r_{l,i} = \frac{f_i^* t_{l,i}^*}{\phi F}$.

3 Deep Reinforcement Learning Approach for MEC Offloading Optimization

3.1 Optimization of MEC Offloading

We aim to reduce the total energy cost and fulfill the computation workload of every edge node. To this end, we propose an optimization formulation to maximize the energy efficiency in MEC offloading, which is defined as the ratio between the total computation workload and the energy consumption:

$$R(\mathbf{t}) \triangleq \frac{\sum_{i \in N} L_i x_i}{\sum_{i \in N} [\tilde{\beta}(r_{a,i}) t_{a,i} + p_{c,i} t_{p,i} + k_i f_i^3 t_{l,i}]}, \quad (3)$$

which depends on the time or workload allocation among different computation schemes. Here binary $x_i = \{0, 1\}$ denotes the outage event that happens when the workload is not finished within a time deadline or the energy supply is not sufficient. Once outage happens, e.g., $x_i = 0$, the computation becomes invalid and will not generate any useful information to the edge device. The hybrid offloading policy has to satisfy the resource constraints from at least two aspects:

Workload Completion. The edge user's workload generated in each time slot has to be completed before a fixed delay bound. The hybrid MEC offloading model provides three schemes to complete the workload, i.e., local computation, active and passive offloading. To cooperate with other edge nodes, we stipulate that each edge node has to complete active and passive offloading within a time slot, whereas local computing can be completed during different time slots but also within a time frame. Therefore, we have $t_{a,i} + t_{p,i} \leq F/N$, where F denotes the frame length and N denotes the total number slots. The combination of computation capabilities in three schemes have to fulfill the user's application requirement. That is, $l_{a,i} + l_{p,i} + l_{l,i} \geq L_i$, where $l_{c,i}$ for $c \in \{l, a, p\}$ denotes the workload of user S_i completed in different computation schemes, including local computation, active, and passive offloading. Typically we have $l_{c,i} = t_{c,i}r_{c,i}$. Note that the computation capability may vary in different schemes, which implies an optimal division of the user's workload to minimize the task outage probability.

Energy Budget. Without loss of generality, we assume that the battery of each node is initially fully charged with the maximum capacity E_{\max} . Different computation schemes also vary in their energy consumptions. In particular, local computation consumes power in CPU cycles, while active offloading consumes high power in RF communications. For simplicity, we omit the power consumption in wireless backscatter, which is much less than that of RF communications [12]. Hence, the total energy consumption of each edge node in one time slot is denoted by $k_i f_i^3 t_{l,i} + t_{a,i} \tilde{\beta}(\frac{l_{a,i}}{t_{a,i}})$, corresponding to local computation and active offloading, respectively.

At the beginning of each data frame, the energy in battery is equal to the energy left in the previous time frame plus the energy collected in other time slots. As such, we define the energy dynamics in k -th data frame of i -th node as follows:

$$E_{k,i} = \min \left(E_{\max}, E_{k-1,i} + \eta \sum_{j \neq i} p_0 g_i^2 t_{p,j} \right), \quad (4)$$

where η denotes energy conversion efficiency and p_0 represents transmit power of HAP. Given the edge user's battery status, the transmission scheduling in two radio modes has to meet the energy budget constraint.

Problem Formulation. Till this point, we can formulate the optimization problem as follows:

$$\max_{\mathbf{t}} R(\mathbf{t}) \quad (5a)$$

$$s.t. \quad t_{a,i} + t_{p,i} \leq F/N, \quad (5b)$$

$$l_{a,i} + l_{p,i} + l_{l,i} \geq L_i, \quad (5c)$$

$$t_{a,i} \tilde{\beta} \left(\frac{l_{a,i}}{t_{a,i}} \right) + k_i f_i^3 t_i \leq E_{k,i}, \quad (5d)$$

$$\mathbf{t}_l \succeq 0, \mathbf{t}_p \succeq 0, \text{ and } \mathbf{t}_l \succeq 0 \quad (5e)$$

where $\mathbf{t}_l \triangleq [t_{l,1}, t_{l,2}, \dots, t_{l,N}]^T$, $\mathbf{t}_a \triangleq [t_{a,1}, t_{a,2}, \dots, t_{a,N}]^T$, and $\mathbf{t}_p \triangleq [t_{p,1}, t_{p,2}, \dots, t_{p,N}]^T$ denote the allocation of computation time in local computation, active, and passive offloading, respectively. The major difficulties of solving problem (5) are caused by the non-convex problem structure, and the couplings among multiple network entities in a dynamic environment. Hence, the conventional model-based optimization techniques become very inflexible and inefficient.

In the following, we resort to a model-free learning based approach. In particular, we integrate deep neural networks (DNNs) and the conventional reinforcement learning for autonomous decision making [13] with huge state space under dynamic network environment.

3.2 DRL Approach for Hybrid MEC Offloading

Relying on the success of DNNs, DRL is capable of solving high dimensional, non-convex, and even model-free network control problems, e.g., multiple access, transmission scheduling, and resource allocation in a dynamic network environment [10]. These are very difficult to handle by classical techniques such as convex optimization, dynamic and stochastic programming, due to the imprecise modeling, uncertain system dynamics, and huge state spaces. In this part, we propose the DRL approach to learn the optimal MEC offloading policy from past experience, without exact knowledge about the network conditions.

Double DQN Framework. Deep Q -Network (DQN) is a popular DRL approach that uses a set of DNNs to approximate the action value function $Q^\pi(s, a; \theta)$ in conventional reinforcement learning, given the state s and action a . Let θ denote the vector of parameters of the multi-layer DNNs, which can be built with different structures, e.g., deep convolutional neural network (CNN) and recurrent neural network (RNN). An overview of DRL approaches and its applications in wireless networking can be found in the survey paper [10]. In general, DQN employs two key mechanisms, i.e., experience replay and target Q -network, to stabilize the learning process. The experience replay mechanism randomly selects a set of transition samples, i.e., mini-batch, from a replay memory of historical transition samples to train the DNN. This can break the correlations and ensure more efficient training by independent transition samples.

The training of DQN is performed by minimizing the loss function $L_i(\theta)$:

$$L_i(\theta_i) = \mathbb{E} [(y_i - Q(s_i, a_i; \theta_i))^2], \tag{6}$$

where y_i denotes the target Q -value and $Q(s_i, a_i; \theta_i)$ is the output of DNN parameterized by θ_i . To stabilize Q -learning, the DQN algorithm uses a separate Q -network with the parameter θ' to generate the target values as $r + Q(s', a'; \theta')$. The target Q -network keeps θ' fixed between successive updates and only updates it by copying the value from θ every a few steps. This mechanism adds a time delay between the update to the online Q -network and the evaluation of the target value.

DQN usually results in overoptimistic estimation of Q -value as we introduce a positive bias by finding the maximum action value $\max_a Q(s_{i+1}, a; \theta'_i)$ in each decision epoch. The same transition data is used to decide the best action with the highest reward. To correct this, an extension of DQN, namely, Double DQN (DDQN), provides a better estimate by updating the action in the online network, and then using the target network to estimate the value function [4].

$$y_i^d = r_i + \gamma Q'(s_{i+1}, \arg \max_a Q(s_{i+1}, a; \theta_i); \theta'_i), \tag{7}$$

where γ is a discount factor. Note that the selection of action in (7) is still based on the online parameter θ_i , as illustrated in Fig. 2. However, the second parameter θ'_i is used to evaluate the Q -value. Hence, the action is decoupled from the generation of target Q -value, which makes the training faster and more reliable.

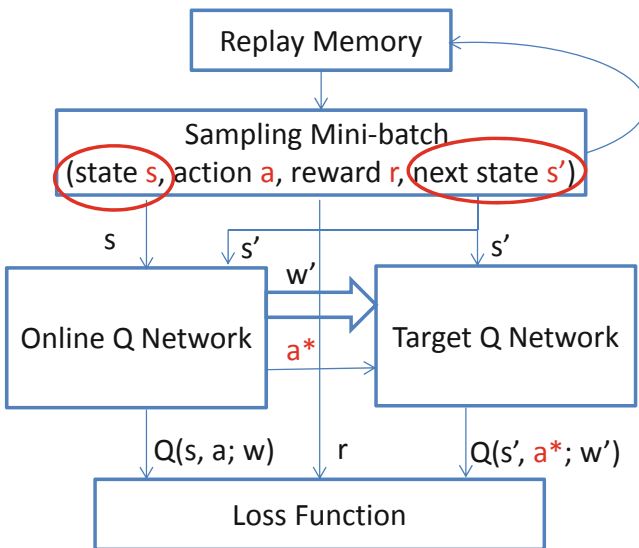


Fig. 2. Information flow of DDQN with DNN parameters \mathbf{w} and \mathbf{w}' .

DDQN for Hybrid MEC Offloading. We define the state space of i -th edge device as $\mathcal{S} = \{(\mathcal{W}, \mathcal{E}, \mathcal{C})\}$, where $w \in \mathcal{W} \triangleq \{0, 1, \dots, W\}$ represents the workload of edge node at the beginning of each time frame, $e \in \mathcal{E} \triangleq \{0, 1, \dots, E\}$, and $c \in \mathcal{C} \triangleq \{0, 1, \dots, C\}$ represent the finite state energy status and channel conditions, respectively. At the k -th time frame, the system state $S_k \in \mathcal{S}$ consists of the channel conditions, the user's energy supply, and random workload. In particular, the channel from the HBS to each user can be modeled in a finite-state Markov chain. This leads to state transitions in the edge user's offloading rate and power consumption. It further affects the transmit performance in two radio modes. Due to the uncertainty in ambient environment, the harvested energy is random and following an unknown stochastic process. The power consumption also varies with the channel conditions. This implies a dynamic process of the edge user's battery status. The workload of each edge user is also uncertain due to the user's mobility and time-varying behaviors of upper layer applications. We assume that the workload can be divided flexibly and processed separately without affecting the integrity. The state transition function $P(S_{k+1}|S_k, a_k)$ represents the distribution of the next state S_{k+1} given the current state S_k and the offloading action a_k .

We define the action space of i -th edge node as $\mathcal{A} = \{a; a \in \{0, 1, 2\}\}$, where $a = 0, 1$, and 2 correspond to active offloading, passive offloading, and local computation, respectively. Given the dynamics of channel conditions, energy status, and workload, each user device will choose its action accordingly to maximize its reward function. Moreover, the action also needs to divide workload in different computation schemes. To avoid continuous action space, we equally divide each time slot into multiple sub-slots. In each sub-slot, the edge user follows the same DRL framework to optimize its offloading decision. By this quantization, we actually optimize the workload allocation among local computation, passive, and active offloading. To maximize the system performance, we define the reward function as the energy efficiency, i.e., the successfully completed workload per unit energy. It captures the immediate value at each time frame, which is given by $R(\mathbf{t}) = \frac{L_i x_i}{\beta(r_{a,i})t_{a,i} + k_i f_i^3 t_{l,i}}$. If workload is completed successfully, e.g., $x_i = 1$, the reward value is a positive number that represents the throughput per unit of energy. Otherwise, the reward value is 0. This allows the DRL agent to constantly search for a better strategy to maximize the total energy efficiency. Algorithm 1 summarizes the DDQN approach for hybrid MEC offloading.

4 Numerical Evaluation

In this section, we evaluate the performance of the proposed DRL algorithm. A fixed transmit power at HAP is set to $p_0 = 100$ mW and the energy conversion efficiency is $\eta = 0.6$. We compare the performance of our hybrid data offloading scheme with the conventional active offloading and local computing scheme without the support for backscatter communications. Besides, greedy algorithm and random algorithm are compared as well. We assume flat block fading channels, i.e., the channel gains remain the same within one time frame and follow

Algorithm 1. DDQN Approach for Hybrid MEC offloading**Require:** Initial workload, channel and energy conditions.**Ensure:** Convergent hybrid MEC offloading strategy π^* .**Initialize** replay memory D , DNN parameters θ, θ' .**for** episode $k \leq 1, 2, \dots, K$ **do** **if** $\text{mod}(k, 100) == 0$ **then**

Change the initialization to the current best result.

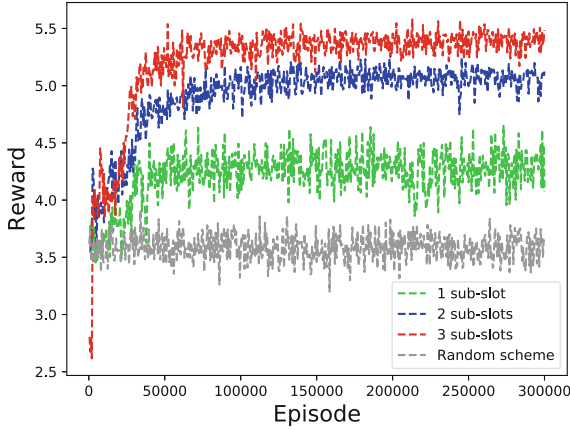
end if Choose a random probability number p . **if** $p < \varepsilon$ **then** $a^*(t) = \arg \max_a Q(s, a; \theta)$. **else** Choose $a(t)$ randomly. **end if** Execute action $a(t)$ and receive immediate reward $r(t)$. Observe next environment state s' . Store transition (s, a, r, s') in replay memory D . Sample random mini-batch of transitions from D . Calculate the target Q -value $y(t)$ from the target network,

$$y(t) = r(t) + \gamma \max_a Q'(s_{t+1}, \arg \max_a Q(s_{t+1}, a; \theta); \theta').$$

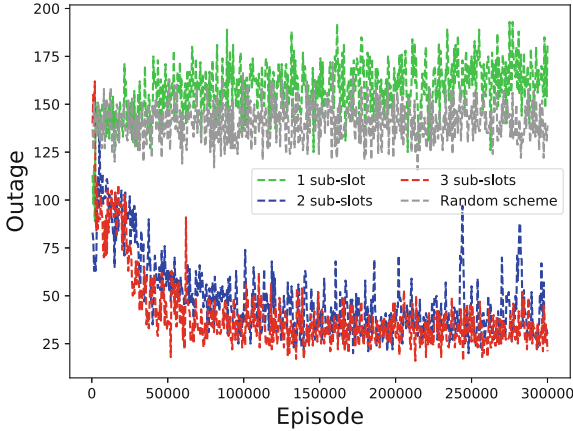
 Update the parameters θ of the online Q -network. Copy θ to the target network for every K steps.**end for****Table 1.** Parameter settings in the DRL framework

Parameters	Value
Number of hidden layers	2
Fully connected neuron network size	64×64
Activation	ReLU
Optimizer	Adam
Learning rate α	0.01
Discount rate γ	0.9
ϵ -greedy	0.9
Mini-batch size	32
Experience replay memory size	2000
Target network update frequency	100

a finite-state Markov chain over consecutive time frames. The workload of each edge node is randomly generated in the range $[4, 32]$ kbits. We set the constant circuit power to $p_c = 1 \mu\text{W}$ and the constant data rate in passive mode as $r_p = 5$ kbps. The noise power is set to $\sigma^2 = -70$ dBm and the bandwidth is given by $B = 400$ kHz. Table 1 lists the parameter settings in our DRL framework.



(a) Rewards with different number of sub-slots



(b) Outage performance with different number of sub-slots

Fig. 3. Performance comparison with different number of sub-slots.

Figure 3 shows the average reward of 500 episodes with different sub-slots at individual edge node. We denote reward as the throughput of hybrid offloading or local computing per energy unit when the workload is fulfilled successfully. To account for workload allocation, we further divide each time slot into a number of sub-slots to realize dual-mode data offloading. When the number of sub-slots is 2, it means that the workload can be assigned to different offloading modes. Firstly, we observe that averaged reward is higher than that of the random scheme, when the number of sub-slot is set to one, which means that the edge user can only work in one mode for data offloading in one time slot. However, the outage performance in this case becomes worse than that of the random

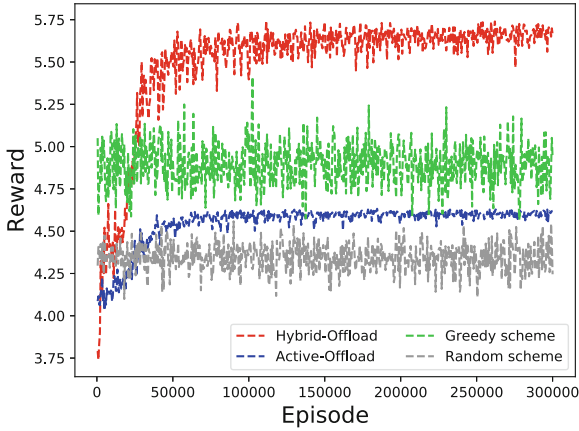


Fig. 4. Rewards in different algorithms.

scheme. The reason is that, the DRL agent may sacrifice outage performance to obtain a higher reward value when the edge user is unable to switch the radio's operating mode during MEC offloading. In Fig. 3(a), we also observe that the average reward grows significantly with the increase in the number of sub-slots, which allows more flexibility in workload allocation. Meanwhile, the number of outage events decreases as shown in Fig. 3(b). This is because, with more sub-slots, the partition of workload can be closer to optimal and thus achieve an improved performance.

We also compare the performance of the proposed DRL-based offloading algorithm with a few baseline approaches, which include the greedy and random algorithms, as well as the conventional active offloading algorithm without passive mode. From Fig. 4, we can see that the DRL-based algorithm achieve the best performance with highest reward. The conventional active offloading algorithm is inferior to the greedy algorithm slightly and much lower than the hybrid offloading algorithm in this set of parameters. This verifies that the hybrid offloading strategy has an significant performance improvement over the conventional offloading scheme, due to its flexibility in mode switch.

5 Conclusions

In this paper, we have proposed a deep reinforcement learning based hybrid offloading algorithm to maximize the energy efficiency in wireless powered MEC networks with hybrid data offloading. We first formulate the energy efficiency function as a non-convex optimization problem. To solve it, we have developed the DDQN-based algorithm to learn the near optimal offloading policy. The numerical results demonstrate that the proposed DRL solution can achieve better performance than the conventional methods.

References

1. Boyer, C., Roy, S.: Backscatter communication and RFID: coding, energy, and MIMO analysis. *IEEE Trans. Commun.* **62**(3), 770–785 (2014)
2. Chen, X., Zhang, H., Wu, C., Mao, S., Ji, Y., Bennis, M.: Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet Things J.* (2019). <https://doi.org/10.1109/JIOT.2018.2876279>
3. Guo, S., Xiao, B., Yang, Y., Yang, Y.: Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing. In: *IEEE INFOCOM*, pp. 1–9, April 2016
4. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning. In: *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 2094–2100, February 2016
5. Hoang, D.T., Niyato, D., Wang, P., Kim, D.I., Han, Z.: Ambient backscatter: a new approach to improve network performance for RF-powered cognitive radio networks. *IEEE Trans. Commun.* **65**(9), 3659–3674 (2017)
6. Li, J., Xu, J., Gong, S., Huang, X., Wang, P.: Robust radio mode selection in wirelessly powered communications with uncertain channel information. In: *Proceedings of IEEE GLOBECOM*, December 2017
7. Li, J., Xu, J., Gong, S., Li, C., Niyato, D.: A game theoretic approach for backscatter-aided relay communications in hybrid radio networks. In: *Proceedings of IEEE GLOBECOM*, December 2018
8. Liu, V., Parks, A., Talla, V., Gollakota, S., Wetherall, D., Smith, J.R.: Ambient backscatter: wireless communication out of thin air. In: *Proceedings of ACM SIGCOMM*, New York, August 2013
9. Lu, X., Wang, P., Niyato, D., Kim, D.I., Han, Z.: Wireless networks with RF energy harvesting: a contemporary survey. *IEEE Commun. Surv. Tutor.* **17**(2), 757–789 (2015)
10. Luong, N.C., et al.: Applications of deep reinforcement learning in communications and networking: a survey. *CoRR abs/1810.07862*. <http://arxiv.org/abs/1810.07862> (2018)
11. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: Mobile edge computing: Survey and research outlook. <https://arxiv.org/abs/1701.01090v3>
12. Niyato, D., Kim, D.I., Maso, M., Han, Z.: Wireless powered communication networks: research directions and technological approaches. *IEEE Wirel. Commun. PP*(99), 2–11 (2017)
13. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge (1998)
14. Xu, L., Zhu, K., Wang, R., Gong, S.: Performance analysis of RF-powered cognitive radio networks with integrated ambient backscatter communications. *Wirel. Commun. Mob. Comput.* **2018**, 16 (2018)