Collaborative Mobile Edge Caching Strategy Based on Deep Reinforcement Learning

Jianji Ren, Tingting Hou, and Shuai Zheng(✉)

HeNan Polytechnic University, Jiaozuo, Henan, China
zhengshuai@home.hpu.edu.cn

Abstract. Recently, with the advent of the 5th generation mobile networks (5G) era, the emergence of mobile edge devices has accelerated. Nevertheless, the generation of massive edge data brought by massive edge devices challenges the connectivity and cache computing capabilities of the internet of things (IoT) devices. Therefore, mobile edge caching, as the key to realize efficient prefetc.h and cache of edge data and improve the performance of data access and storage, has attracted more and more experts and scholars' attention. However, the complexity and heterogeneity of the devices in the edge cache scenario make it unable to meet the low latency requirements of 5G. In order to make the mobile edge caching more intelligent, based on the widely deployed macro base stations (ξBSs) and micro base stations (μBSs) in 5G scenarios, the ξBS cooperation space and μBS cooperation space is conceived in this paper. Besides, deep reinforcement learning (DRL) algorithms with perception and decision-making capabilities are also used to implement collaborative edge caching. DRL agents perform original and high-dimensional observation training on high-dimensional edge cache scenes, which can effectively solve the dimensionality problem. Then, we jointly deployed federated learning (FL) locally to train DRL agents, which not only solved the problem of resource imbalance, but also realized the localization of training data. In addition, we formulate the energy consumption problem in the collaborative cache as an optimization problem. The simulation results show that the solution greatly reduces the cost of caching and improves the user's online experience.

Keywords: Mobile edge caching · Deep reinforcement learning · Federated learning

1 Introduction

With the arrival of 5G and artificial intelligence (AI), we will enter the intelligent age of the internet of everything (IoE). AI makes it convenient for people to interact with everything, while 5G makes the IoT a reality. The three application scenarios of 5G (eMBB, uRLLC and mMTC) have high requirements on bandwidth, delay and connectivity of devices in IoT, which requires 5G network to be

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved
X. Wang et al. (Eds.): 6GN 2020, LNICST 337, pp. 3–15, 2020.
https://doi.org/10.1007/978-3-030-63941-9_1

more decentralized and intelligent. The traditional centralized storage method cannot meet the low latency requirements required in the 5G scenario. Therefore, it is necessary to deploy small or portable data centers on the edge of the network and conduct intelligent local processing of terminal requests with the help of AI technology to meet the ultra-low delay requirements of uRLLC and mMTC. However, considering the high expenses of deploying a data center on the edge of the network, the mobile edge caching is a promising solution.

As an extended concept of mobile edge computing [1], mobile edge caching not only caches resources in edge nodes (ENs) closer to users, but also provides real-time and reliable data storage and access for edge computing. It is regarded as a data storage method to improve network efficiency and alleviate the high demand for radio resources in the future network. Considering the localization mode of edge caching, which is faster to access than cloud storage, and it can be used offline even when the internet connection is interrupted, so local applications that rely on edge storage are more resilient to service interruptions. However, a particular concern in edge caching is that when the local node cannot satisfy the user's request, the user's internet experience will be reduced. The above problems can be solved through the cooperation between ENs. Through the cooperation among ENs, the geographically distributed ENs can cooperate together to form a more intelligent distributed storage network, so as to address the problem of resource imbalance and ensure quality of service (QoS) at the same time.

To make the cooperation between ENs more intelligent, not only the perception ability but also the decision-making ability is needed. As an AI method closer to the way of human thinking, DRL combines the perception ability of deep learning (DL) with the decision-making ability of RL. With complementary advantages, it can directly learn control strategies from high-dimensional original data. Therefore, in this paper, we conceived the ξBS cooperation space and μBS cooperation space based on the commonly deployed ξBSs and μBSs in 5G scenario and made the perception and decision in the cooperation of edge nodes with the help of DRL. When the user equipment (UE) sent a cache request, based on the perception of the DRL agent placed at the node, The DRL agent learns the optimal strategy, and then designs an optimal resource allocation plan between the cache requester and the cache provider. However, since the training of the DRL agent requires a large amount of multi-dimensional data, which may involve cross-enterprise data transmission and cause excessive pressure on the network in the process of training data uploading, distributed training DRL agent is a promising choice. FL, as an implementable path and "data island" solution for machine learning under privacy protection, allows the construction of collection models from data distributed across data owners without getting through the data to meet the requirements of joint modeling. Therefore, in the process of training the DRL agent, this paper deploys FL local training the DRL agent, which can not only share the model and solve the problem of resource imbalance but also ensure the localization of training data and the security of data.

The main contributions of this paper are as follows:

- Based on the generally dense deployment of ξBSs and μBSs in 5G scenarios, the ξBS cooperation space and μBS cooperation space is proposed, and devices in the collaboration space are used for collaborative transmission and caching.
- Due to heterogeneous resources and the complexity of a large number of devices in mobile edge nodes, mobile edge caching involves complex scheduling problems. Therefore, when solving the problem of cooperative transmission of content, we use the perception and decision-making ability of AI algorithm, namely DRL algorithm, to place DRL agent in each node to decide the best cooperative mode.
- To ensure the security and privacy of the data, when training the DRL agent of each node, the FL uses local data to perform training and parameter update of the DRL model without prior knowledge of the global data, to realize the sharing of the model.
- Simulation results show that the DRL and FL based collaborative edge caching strategy has better performance than the centralized edge caching strategy.

The rest of this paper is organized as follows. In the next section, we review the related work. In Sect. 3, we present the system model of collaborative edge caching and formulate the energy minimization problem. In Sect. 4 we introduce the background and fundamentals of DRL and FL, then we describe the process of training DRL agents with FL. Simulation results are discussed in Sect. 5. Finally, Sect. 6 concludes the paper and look forward to the future research direction.

2 Related Work

The increase of massive intelligent devices brings a huge burden on the internet, which poses a great challenge to the current internet infrastructure. This also causes users to suffer severe network congestion and delays frequently, so the emergence of cloud caching [2] has alleviated this situation. As a centralized storage model of long-distance data transmission, cloud caching requires data to cross the geographical location limitation, and has a significant delay in data transmission and the possibility of network fluctuation, which is difficult to meet the real-time requirements of edge applications. Fortunately, edge caching [3] distributes data across adjacent edge storage devices or data centers, dramatically reducing the physical distance between data generation, data calculation, and data storage, thereby overcoming the high latency, network dependency and other issues caused by long-distance data transmission in cloud storage, providing high-speed, low-latency data access for edge computing.

Edge caching is not a new topic and has generally been extensively studied by many communities. With aim to reduce the delay of data transmission by IoT devices, [4] focused on the pre-caching of video to reduce the amount of repeated data, a collaborative joint caching and processing strategy for on-demand video streaming in mobile edge computing network was proposed to effectively select the cache video and its version to solve the problem of code switching between different video. The author described the collaborative joint caching and processing problem as an integer linear programming that minimizes backhaul network costs when limited by cache storage and processing power. [5] proposed a cooperative cache allocation and computational offloading scheme to cope with the delay and backhaul pressure caused by a large number of data exchanges in the 5G application scenario, and the cache and computing resources on the MEC server are allocated according to their needs and payments. For service requesters, mobile network developers allocate resources according to weighted proportions to maximize resource utilization. Based on the size and popularity of cached content, the authors in [6] introduced an objective function to assess the popularity of content to ensure global cache hit ratios, and use 0–1 knapsack dynamic programming to maximize wireless The local cache hit ratio of the access points (APs), thereby reducing the content acquisition delay in the wireless network and improving the network throughput.

Inspired by the success of deep reinforcement learning in solving complex control problems the author in [7] proposed a DRL-based content cache strategy, to improve the long-term cache hit ratio of base-station stored content without knowing the popularity of content in advance. Considering the multi-level structure of the network, [8] proposed a network cache setting that abstracts the server into a parent node that is connected to multiple leaf nodes, using leaf nodes closer to the characteristics of the end-user, through locally stored files. Or get the file from the parent node to process the request. Besides, the author proposes an efficient caching strategy using DRL, and proves that the strategy can learn and adapt to the dynamic evolution of file requests and the caching strategy of leaf nodes.

Also, as a data storage method that provides real-time and reliable data storage and access for edge computing, mobile edge caching has been studied by many scholars. The scholars in [9] have studied the edge buffer and computational offload in the mobile edge system and proposed the "In-Edge AI" architecture. With the cooperation between the device and the edge nodes, the learning parameters are exchanged to better train and infer the model for dynamic system-level optimization and application-level enhancement while reducing unnecessary system communication load. Inspired by artificial intelligence algorithm, the author in [10] considered the calculation unloading problem in edge computing. The DRL was used to indicate the decision-making of IoT devices and conducted distributed training on DRL agents. In [11], aiming at the complexity of unloading drive caused by dynamic topological changes caused by vehicle mobility in the internet of vehicles, considering the dynamic road condition information and with the help of mobile edge cache, the author proposed a predictive mode transmission scheme for uploading task files, which reduced the cost of task transmission and improved the efficiency.

The difference between this paper and related work is as follows: I) The cooperative space of ξBS and μBS is conceived, which can realize the cooperative caching of mobile edge scene. II) The problem of energy consumption in a collaborative cache is formulated as an optimization problem. III) DRL and FL are combined to make decisions in the process of collaboration based on DRL perception. To ensure the privacy and security of data, federated learning and localization are adopted to train DRL agents, to share models.

3 System Architecture

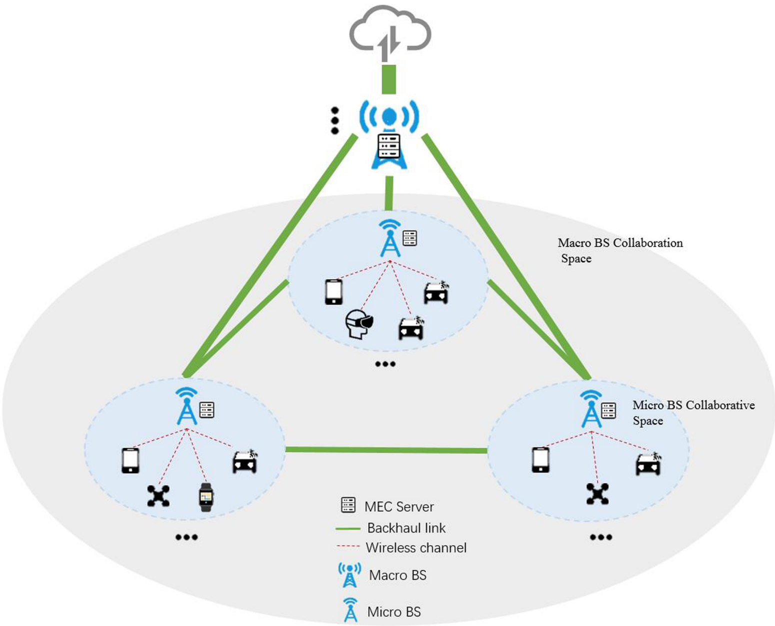


Fig. 1. Collaborative mobile edge caching architecture

3.1 Scenario Analysis

In 5G scenarios, the following three factors need to be considered in order to improve the network capacity: bandwidth, spectrum efficiency, and the number of scenarios covered by the network. As is known to all, the frequency band in 5G scenario is higher than that in 4G, ranging from 3.5 GHz to 30 GHz, or even higher. However, the problem existing at the same time is that the higher the frequency band, the worse the penetration capability and the smaller the coverage area. That also means that 5G network needs a huge number of μBSs, so μBSs have become the key to solving network coverage and capacity in the future. Therefore, in this paper, we focus on the cooperation of user nodes among a large number of commonly deployed μBSs to improve the QoS.

As shown in Fig. 1, the architecture of collaborative mobile edge cache includes several ξBSs and several μBSs within the coverage of a specific ξBS, which constitute a cellular network. There are three types of network connection: I) the UE creates communication and data transmission with EN through the wireless channel assigned by the μBS within the coverage of the μBS. II) The μBS and the ξBS are connected through a backhaul link. III) the backhaul link connects the ξBS to the internet through the gateway. Each ξBS covers the users in its coverage area and is equipped with local cache storage.

We assume that both the ξBS and the μBS are equipped with MEC servers to realize collaborative caching of user devices in the collaboration space. Then we define a two-level collaboration space:

- The first-level collaboration space between IoT devices covered by the same μBS is called the μBS collaboration space, and the EN in this collaboration space is called a direct EN helper.
- The secondary cooperative space formed by the μBS where UE is located is counted in L_b hop and covered by the same ξBS, which is called the ξBS cooperative space, and EN in this scope is called indirect EN helper.

Through the cooperation of nodes between the two levels of collaborative space, the pressure of data center can be relieved and the QoS can be improved.

When the UE sends a request, the request is accepted and processed by BS or sent to the remote cloud for processing, this process involves the BS and the cloud, and the transmission between the UE and the cloud. We use a specific UE to send a cache request as an example to illustrate the model. The UE first sends the cache request to the direct EN helper, the DRL agent placed in the direct EN helper observes the available computing and cache resources at the edge of the network, and design a corresponding resource allocation plan based on perception. If the requested content is available in the direct EN helper, the cached content can be transmitted directly through wireless transmission. When the method cannot meet the requirements of the UE, the ξBS covering the UE needs to be used, searching for requested content at other indirect EN helper, and if the content exists in a indirect EN buffer, then send the requested content to the content requester. If the above steps do not meet the user's needs, the agent will send content requests to the cloud data center.

We give two examples from different areas to illustrate the scenarios, which are strong proof of the advantages of collaborative caching in real scenarios: i) Video stream collaboration cache scenario: video stream caching often occurs in real life. With the help of direct EN helper and indirect EN helper, it can reduce the download of duplicate data, alleviating the backhaul of the backbone network and the pressure of the cloud. ii) Unmanned vehicle cooperation scenario: 5G has high transmission speed and millisecond delay characteristics, with the help of ξBS collaboration space and μBS collaboration space through the vehicle networking will realize data sharing between cars, can take the initiative to avoid obstacles, cooperative path planning, overspeed monitoring, etc., greatly improve the response data and scene recognition accuracy.

3.2 Dynamic System Architecture

Table 1. Notations used in experiment

Notation	Definition
$\mathcal{U}$	The users equipment set
$\mathcal{W}$	A set of wireless channels in the same μBS
$\mathcal{N}$	A EN set in the same μBS coverage as UE
$\mathcal{M}$	A set of μBSs within the coverage of the same ξBS
L	The maximum radio resources
p	The transmit power of each UE in each assigned subchannel
q_f	User request for content f
S_i	Cooperative transmission mode indicator
d_i	The amount of data requested for transmission
v_i	The total number of CPUs required to transfer the requested content
r_i^n	The maximum uplink rate can be achieved by the UE in the subchannel
g_i	The channel gain of UE$_i$ in each subchannel
t_i^n	The shortest transmission time of data in the μBS cooperative space
e_i^n	The energy consumption of data transmission in the μBS cooperative space
e_c	The energy consumed during resource transmission between μBS using indirect EN helper
p_i^a	The transmitted power of the requested target content in the indirect EN helper
p_i^b	The transmitting power of the μBS where ue resides
r_i^a	The channel transmission rate corresponding to the indirect wireless EN helper collaboration
r_i^b	The channel transmission rate of the μBS to the UE where the UE is located
e_i	The energy consumed to assist device$_i$ in content transmission

In this paper, the system model of 5G IoT with ENs is adopted for analysis, as shown in Fig. 1. The popularity of content cached at mobile nodes is described by the Zipf distribution. We assume that the IoT devices in Fig. 1 all have computing and caching resources and communication capabilities, and that the wired communication capability between the μBSs within the coverage of the same ξ station is very strong. Each BS has a mobile edge caching server, which we consider to be a small data center with computing and storage capabilities that can aggregate DRL agent model parameters. For the clarity of the following discussion, the key notations are summarized in Table 1.

The DRL agent placed at EN determines whether the content requested by UE is stored at EN by sensing user preferences, social relationships, and so on. In our simulation, content requests from UE are generated by the Poisson distribution and expressed by q_f, where f represents the requested content. We represent the set of user equipments as $\mathcal{U} = \{1, 2, ..., U\}$, the EN set within the coverage of the same μBS as UE is $\mathcal{N} = \{1, 2, ..., N\}$, and the wireless channel in the same μBS is $\mathcal{W} = \{1, 2, ..., W\}$. $\forall U \in \mathcal{U}$ can establish communication with $\forall N \in \mathcal{N}$ and collaborate with the wireless bandwidth of WHz allocated randomly.

The transmission mode indicator $S_i = \{0, 1\}$, $S_i = 0$ means that the request sent by UE$_i$ is executed in the direct EN helper mode within the μBS collaboration space, $S_i = 1$ means the cooperative transmission in the indirect EN helper mode within the ξBS collaboration space. We represent the requested content transfer of UE$_i$ as (d_i, v_i), where d_i represents the data amount of the requested content transfer, and v_i represents the total CPUs required for the content transfer.

Considering that the channel gain fading in a small range is average and only the fading in a large range can affect it, we assume that the channel gain of different subchannels in the μBS collaboration space is the same for UE and can be different for different UE. Therefore, the power allocated to each subchannel is equal. Then, the uplink rate of the UE in each subchannel is:

$$r_i^n = W \cdot log_2[(1 + p \cdot g_i/(W \cdot K)] \tag{1}$$

Where p is the transmit power of each UE in each assigned subchannel, g_i is the channel gain of UE$_i$ in each subchannel. W is the subchannel bandwidth and K is the noise power spectrum.

The shortest transmission time of the data in the μBS cooperation space is:

$$t_i^n = d_i/r_i^n \tag{2}$$

Therefore the corresponding transmission energy consumption is:

$$e_i^n = p_i^n/t_i^n \tag{3}$$

p_i^n represents the transmission power of the UE$_i$ transmission content distribution.

When the resource transmission between μBSs is performed by means of the indirect EN helper, the corresponding energy consumed is:

$$e_c = e_i^a + e_i^b \tag{4}$$

That is

$$e_c = P_i^a \cdot d_i/r_i^a + P_i^b \cdot d_i/r_i^b \tag{5}$$

Where P_i^a is the transmitted power of the requested target from indirect EN helper, and r_i^a is the corresponding channel transmission rate in this cooperative mode. Similarly, P_i^b is the transmitting power of the μBS where UE resides, and r_i^b is the channel transmission rate from the content provider to UE. We ignored the data transfer time and the energy transfer between the BSs, considering the short distance of the wire transfer time is negligible.

To minimize the transmission consumption, combined with the above formula, the energy consumed by the UE$_i$ to perform device transmission is formalized into the following form:

L is the maximum radio resource, the optimization is:

$$\begin{aligned}
&e_i = \min_{S_i\theta} \sum_{i=1}^{U} S_i(P_i^a \cdot \tfrac{d_i}{r_i^a} + P_i^b \cdot \tfrac{d_i}{r_i^b}) + (1 - S_i)(P_i^a \cdot \tfrac{d_i}{r_i^a}) \\
&s.t. \\
&\ S_i \in \{0, 1\} \\
&\sum_{i=1}^{U} \theta_i \leq L \\
&0 \leq \theta_i \leq S_i L
\end{aligned} \tag{6}$$

4 Deep Reinforcement Learning and Federated Learning

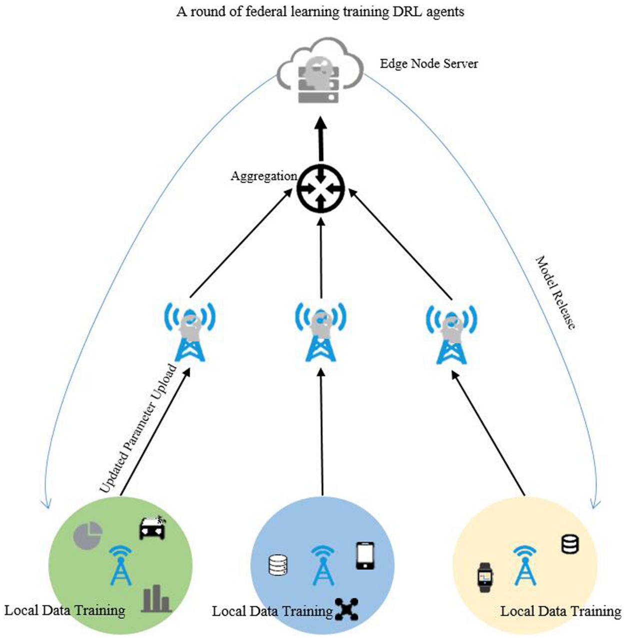


Fig. 2. DRL agent training process based on FL

Due to the complexity of edge caching and EN assistance, with the help of AI to obtain efficient resource scheduling strategies in complex environments with heterogeneous resources and a large number of devices, we introduce the Double Deep Q Network (DDQN) algorithm to solve the search problem of the requested content. As an end-to-end perception and control algorithm, the learning process of DRL is as follows : i) at each moment, agents interact with the environment to obtain a high-dimensional observation, and use DL method to perceive the observation, so as to obtain an abstract and concrete representation of state features; ii) evaluate the value function of each action based on the expected report, and map the current state into the corresponding action through strategies; iii) react to the environment and actions and get the next observation. Through the above steps, the optimal strategy is finally obtained. It is effective for DRL technology to find the optimal strategy for a dynamic edge system, but it also needs a lot of computing resources. Therefore, we should consider the issue of DRL agent deployment (Fig. 2).

In a traditional DRL training process, a central server located at a BS can access the entire training data set within its coverage. Therefore, the server can divide the μBS covered by it into a subset that follows a similar distribution. These subsets are then sent to the participating nodes for distributed training. However, this centralized training method may face privacy and security risks, and mass data uploading process will increase the pressure on the network, which is contrary to the low latency attribute of 5G. Therefore, FL, which is considered a "data island" solution, is used in this paper to deploy DRL agents. The DRL agent and FL are jointly deployed at the EN, the DRL agent is trained locally based on local data, and the agent parameters are uploaded, aggregated, downloaded and updated with the help of FL, so as to realize user privacy protection and solve the problem of resource imbalance.

FL has always been considered an AI technology that guarantees that participants have the best models while protecting the data security and privacy of all parties involved. It allows users to collaboratively train a shared model while saving personal data on the device, thereby alleviating user privacy concerns. Therefore, FL can be used as an enabling technique for machine learning model training in mobile edge networks. In a FL system, the data owner acts as an FL participant, collaborating to train the machine learning model required by the aggregation server. A basic assumption is that the data owners are honest, which means they use real private data for training and submit real local models to the FL server. It allows us to build a collection model from data distributed across data owners, making it our first choice for distributed training DRL agents. Advantages of using FL training DRL agents: 1) Data isolation: Data can be encrypted and decrypted in real time during transmission to meet user privacy protection and security requirements. 2) Relieve network pressure: Locally train the model and upload only the model parameters, which alleviates the pressure on the network.

5 Simulation Test and Results

5.1 Experimental Setting

In the experiment, Zipf distribution was used as the content popularity. With the help of the fitting value of real data captured in [2], we set the Zip parameter $\partial = 1.58$. Requests from UE were expressed by using Poisson distribution. Once the EN receives a request from UE, the DRL agents located in the indirect and direct EN helpers will determine the location of the requested content. We set the total bandwidth between UE and EN as W= 4.8 Hz, and divide it into

Algorithm 1. DRL agent training and aggregation based on FL.

Initialization:
With respect to the global DRL agent in the EN:
  Initialize the DRL agent with random weight λ_0;
  Initialize the gross training times T_0 of all devices;
With respect to each user equipment $U \in \mathcal{U}$:
  Initialize the experience replay memory $\mathcal{M}_0^U$;
  Initialize the local DRL model λ_0^U;
  Download θ_0 from the EN and let $\lambda_0^U = \lambda_0$;
Iteration:
For each round $t = 1$ **to** T **do**;
$E_t \leftarrow \{$ random set of m available user equipments$\}$;
  For each user equipment $U \in E_t$ in parallel do;
    Fetch λ_t from the EN as let $\lambda_t^U = \lambda_t$;
    Sense and update $\mathcal{M}_t^U$;
    Train the DRL agent locally with λ_t^U on $\mathcal{M}^U$;
    Upload the trained λ_{t+1}^U to the EN;
    Notify the EN the times T_t^U of local training;
  End For
  With respect to the EN:
    Receive all model updates;
    Refresh the statistical $T_t = \sum_{U \in E T_t^U}$;
    Perform model parameter aggregation as:
      $\lambda_{t+1} \leftarrow \sum_{U \in E} (T_t^U / T_t) \cdot \lambda_{t+1}^U$;
End For

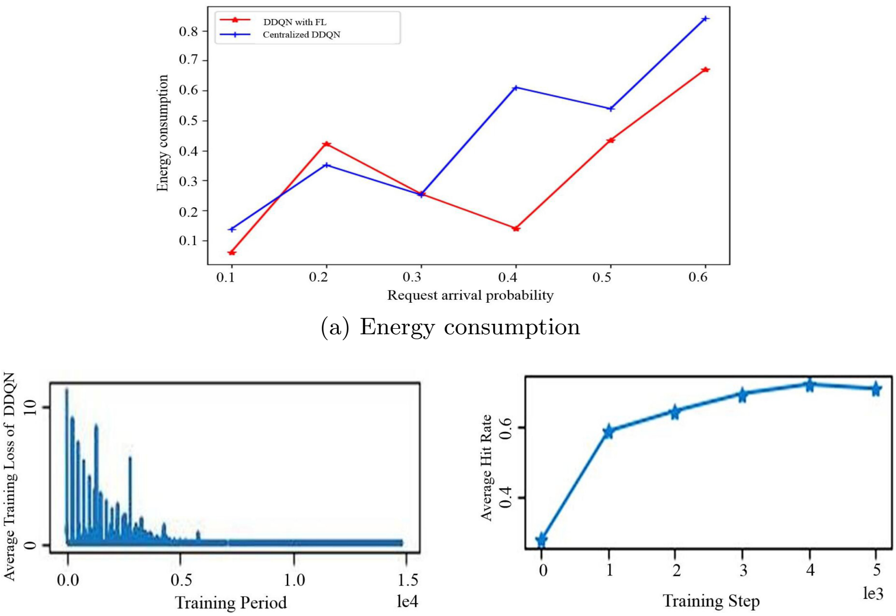
subchannels of equal bandwidth according to the number of ENs in the μBS cooperative space where the UE resides. We investigated a collaborative cache in an IoT scenario with 7 ξBSs, 16 μBSs, and the amount of UE was randomly generated and distributed in μBSs to evaluate the capabilities of our proposed approach.

For the training of DDQN agents placed in EN, BS and cloud, we set relevant parameters as follows: exploration probability 0.001, replay memory capacity 5000, learning rate 0.005, discount factor 0.9, full connection of two layers, 200 neurons in each layer activated by tanh function, replacement of target Q network every 250 times, minimum batch of 200. To evaluate our proposed collaboration strategy, we also implemented in the experiment the strategy that UE's request is satisfied in a centralized way, that is, all sensor data collected by IoT devices are uploaded to a central server for centralized DRL training.

Algorithm 1 shows the process of training DRL agent using FL.

5.2 Simulation Results Analysis

In this section, we present simulation results to evaluate the performance of the collaborative caching strategy presented in this paper. First, we evaluate the total energy consumption of the UE collaborative cache in the collaborative



(a) Energy consumption

(b) Training loss of DDQN

(c) Average hit rate

Fig. 3. Performance of DRL based on FL training

scenario proposed in this paper, and compare it with the energy consumption of the traditional transmission method. The result is shown in Fig. 3(a). With the increase of the probability of the arrival of the cache task, the energy consumption of the FL-based DDQN cooperative cache scheme proposed in this paper is lower than that of the traditional centralized training DRL agent method, which indicates that the collaborative cache strategy proposed in this paper has better energy-saving performance. Then we evaluated the change of DDQN training loss with training duration. It can be seen from Fig. 3(b) that as the training time increases, the training loss gradually decreases. Finally, we assume that the task request probability is the same, select three UEs, and evaluate the cache hit rate according to the changes in the number of training. As shown in Fig. 3(c), we can see that the hit rate is gradually increasing, which shows the effectiveness of the proposed scheme and can guarantee QoS.

6 Conclusion and Future Work

In this paper, we consider the collaborative edge caching problem for multiple users in 5G scenarios. Firstly, we propose the ξBS cooperation space and μBS cooperation space and carry out collaborative edge caching of UE by EN cooperation in the cooperative space. Then, we deploy the DRL agent at the EN, and use the DRL's perception and decision-making ability to make collaborative decision-making through the agent's perception. Afterward, to ensure data security and ease network stress, FL is used to localize DRL agents. Furthermore, we formulated the energy consumption problem in the collaborative cache as an optimization problem. The final simulation results verify that our strategy can reduce energy consumption in the cache and QoS. However, the current model proposed in this paper has not considered device to device (D2D) cooperative caching. In future work, we will consider D2D collaborative edge caching to improve user experience and save spectrum resources. Besides, we will further improve the structure and experiment proposed in this paper.

References

1. ETSI. Mobile Edge Computing - Introductory Technical White Paper, September 2014
2. Dash, D., Kantere, V., Ailamaki, A.: An economic model for self-tuned cloud caching. In: 2009 IEEE 25th International Conference on Data Engineering. IEEE (2009)
3. Lobo, A.R., Nadgir, D.K., Kuncolienkar, S.T.S.: Intelligent edge caching. U.S. Patent Application 13/548,584[P]. 2014-1-16
4. Tran, T.X., Pandey, P., Hajisami. A., et al.: Collaborative multi-bitrate video caching and processing in mobile-edge computing networks. In: 2017 13th Annual Conference on Wireless On-Demand Network Systems and Services (WONS), pp. 165–172. IEEE (2017)
5. Ndikumana, A., Ullah, S., LeAnh, T., et al.: Collaborative cache allocation and computation offloading in mobile edge computing. In: 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 366–369. IEEE (2017)
6. Ren, J., Hou, T., Wang, H., et al.: Increasing network throughput based on dynamic caching policy at wireless access points. Wirel. Netw. **26**, 1577–1585 (2020). https://doi.org/10.1007/s11276-019-02125-0
7. Zhong, C., Gursoy, M.C., Velipasalar, S.: A deep reinforcement learning-based framework for content caching. In: 2018 52nd Annual Conference on Information Sciences and Systems (CISS), pp. 1–6. IEEE (2018)
8. Sadeghi, A., Wang, G., Giannakis, G.B.: Deep reinforcement learning for adaptive caching in hierarchical content delivery networks. IEEE Trans. Cogn. Commun. Netw. **5**(4), 1024–1033 (2019). https://doi.org/10.1109/TCCN.2019.2936193
9. Wang, X., Han, Y., Wang, C., et al.: In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning[J]. IEEE Netw. **33**(5), 156–165 (2019)
10. Ren, J., Wang, H., Hou, T., et al.: Federated Learning-Based Computation Offloading Optimization in edge computing-supported Internet of Things. IEEE Access **7**, 69194–69201 (2019)
11. Zhang, K., Mao, Y., Leng, S., et al.: Mobile-edge computing for vehicular networks: a promising network paradigm with predictive off-loading. IEEE Veh. Technol. Mag. **12**(2), 36–44 (2017)