electronics

MDPI

Article

Federated Deep Reinforcement Learning-Based Caching and Bitrate Adaptation for VR Panoramic Video in Clustered MEC Networks

Yan Li [1,2]

[1] School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330032, China; yli@nit.edu.cn

[2] School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China

Abstract: Virtual reality (VR) panoramic video is more expressive and experiential than traditional video. With the accelerated deployment of 5G networks, VR panoramic video has experienced explosive development. The large data volume and multi-viewport characteristics of VR panoramic videos make it more difficult to cache and transcode them in advance. Therefore, VR panoramic video services urgently need to provide powerful caching and computing power over the edge network. To address this problem, this paper establishes a hierarchical clustered mobile edge computing (MEC) network and develops a data perception-driven clustered-edge transmission model to meet the edge computing and caching capabilities required for VR panoramic video services. The joint optimization problem of caching and bitrate adaptation can be formulated as a Markov Decision Process (MDP). The federated deep reinforcement learning (FDRL) algorithm is proposed to solve the problem of caching and bitrate adaptation (called FDRL-CBA) for VR panoramic video services. The simulation results show that FDRL-CBA can outperform existing DRL-based methods in the same scenarios in terms of cache hit rate and quality of experience (QoE). In conclusion, this work developed a FDRL-CBA algorithm based on a data perception-driven clustered-edge transmission model, called Hierarchical Clustered MEC Networks. The proposed method can improve the performance of VR panoramic video services.

Keywords: federated deep reinforcement learning; VR panoramic video; caching and bitrate adaptation; hierarchical clustered MEC network

check for updates

Citation: Li, Y. Federated Deep Reinforcement Learning-Based Caching and Bitrate Adaptation for VR Panoramic Video in Clustered MEC Networks. *Electronics* **2022**, *11*, 3968. https://doi.org/10.3390/electronics11233968

Academic Editor: Christos J. Bouras

Received: 12 November 2022
Accepted: 27 November 2022
Published: 30 November 2022

1. Introduction

Among network services, video services have become mainstream [1]. With the accelerated deployment of 5G networks, new types of videos, such as 3D videos and VR panoramic videos, which are more expressive and experiential than traditional videos, have ushered in opportunities for rapid development. Especially in 2021, with the rapid spread of the concept of metaverse, more and more things in the physical world were fully displayed in the metaverse in the form of VR panoramic videos. The VR panoramic video market is expected to continue to grow at a rate of 34% per year from 2018 to 2024 [2].

Edge computing has been proposed to use the processing power of various devices between the cloud center and the terminal to perform computing and storage tasks on the spot. Most data processing and analysis can be carried out at the edge nodes near the terminal without transmitting them to the cloud center, which can reduce the bandwidth consumption of the backbone network. Since mobile equipment now includes mainstream internet terminals, mobile edge networks have become the main form of edge computing, called mobile edge computing [3–5].

In the age of steadily increasing wireless bandwidth, although video transmission optimization technologies [6,7] such as content distribution network (CDN), adaptive bitrate streaming (ABR) and peer-to-peer communication (P2P) have been applied, VR panoramic video still has a much higher bitrate than traditional video of the same resolution. In order to acquire an optimal quality of experience, VR panoramic videos often adopt ultra-high-definition resolutions such as 4K and 8K. Users often change their perspectives when enjoying VR panoramic videos, which undoubtedly increases the difficulty of video caching and bitrate adaptation in advance.

The coding scheme of VR panoramic video is also different from that of traditional video, which may lead to difficulty in achieving the desired effect with the existing joint optimization strategy of video coding and transmission. At the same time, the amount of VR panoramic video data is large, which inevitably requires bandwidth consumption. These problems lead to the urgent need for edge networks to provide adequate computing and caching resources.

In recent years, many scholars have carried out a lot of work on virtual reality video transmission research [8–11], including introducing reinforcement learning (RL) algorithms [12] to solve problems such as video proactive cache and rate adaptation. Based on the ability of deep reinforcement learning (DRL) to solve complex problems with changing environments [13,14], it has also been widely used in video transmission optimization. However, the DRL-based algorithm has the defects of instability and being susceptible to falling into the local optimal solution. In order to effectively improve learning efficiency, federated learning (FL) has been proposed to establish a federated learning model based on distributed datasets, which not only solves data transmission consumption, but also shares different training data [15]. Furthermore, how to better meet the computing, caching and communication resources required in VR panoramic video transmission and intelligent decision making is still an urgent problem.

In this work, to meet the needs of computing, caching and communication capabilities in VR panoramic video transmission and intelligent decision making, a hierarchical clustered MEC network was constructed. Then, under the proposed network architecture, we integrated deep reinforcement learning with federated learning to improve the stability and efficiency of the deep reinforcement learning algorithm. FDRL was applied to the clustered-edge network to realize joint cache and bitrate adaptive intelligent optimization, for the purpose of improving the service quality of VR panoramic video. The main contributions of this paper are summarized as follows:

- Hierarchical Clustered MEC Networks: This paper constructs a regional hierarchical management model for clustered MEC edge nodes, called Hierarchical Clustered MEC Networks (HC-MEC). Based on the edge–edge collaboration between edge nodes in the cluster, the storage and processing capabilities of the MEC server are improved to meet the needs of VR panoramic video services.
- Data perception-driven clustered-edge transmission model: A clustered-edge transmission model for VR panoramic video driven by user data perception is proposed. According to the VR panoramic video area of interest predicted by user data, the joint decisions about caching the video tiles of field of view (FoV) and choosing the high-quality bitrate version are determined.
- Federated deep reinforcement learning-based caching and bitrate adaptation (FDRL-CBA): We take the lead in proposing a joint optimization scheme for VR panoramic video caching and bitrate adaptation. The joint optimization problem is transformed into a Markov Decision Process (MDP), and federated deep reinforcement learning is employed to solve it for achieving adequate performance in cache hit ratio and QoE.

The remainder of this paper is organized as follows. Section 2 presents the system model. Section 3 introduces in detail the framework design of FDRL-CBA. The performance evaluation is presented in Section 4, followed by conclusions and future work in Section 5.

2. System Model

This paper establishes a VR panoramic video transmission network model based on mobile edge computing. It mainly includes the following parts of HC-MEC networks: clustered-edge transmission model, cache model and bitrate adaptation model.

2.1. The Hierarchical Clustered MEC Networks

The amount of VR panoramic video data is large, and high computing and storage capacities are required for caching and transcoding processing. A single MEC edge node can hardly meet the needs of processing capacity. At the same time, the number of MEC nodes at the network edge is relatively large. The differences in computing, storage and communication capabilities between nodes are also relatively large. Based on the above considerations, this paper proposes a regional hierarchical management model for clustering MEC edge nodes, called Hierarchical Clustered MEC Networks. The HC-MEC networks are given in Figure 1. In the vertical direction, the hierarchical structure is realized. From top to bottom, they are cloud center, edge cloud, edge cluster, cluster head and cluster element. The cluster structure is also implemented in the horizontal direction. Multiple edge clusters can be connected under an edge cloud, and the edge nodes in each edge cluster can be different. Therefore, the vertical layering and horizontal clustering of HC-MEC networks can be described in detail as follows.

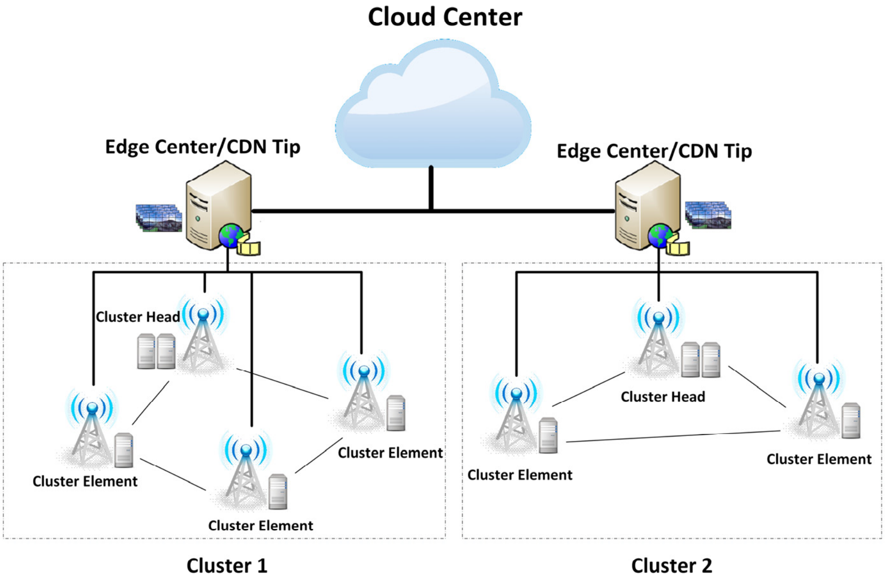


Figure 1. Hierarchical Clustered Mobile Edge Computing (HC-MEC) networks.

(a) Vertical layering: In this network, the VR panoramic video transmission model is seamlessly connected with the current general video transmission CDN technology. Vertically, it established a hierarchical network structure to provide integrated computing resources to meet the needs of task processing. The edge cloud is the CDN regional server. The MEC edge nodes are deployed to the macro base station, small base station, intelligent terminal and other locations, which can form a cluster. In this way, a layered transmission model of VR panoramic video is established.

(b) Horizontal clustering: Due to the large amount of VR panoramic video data, it is difficult for a single MEC edge node with limited capabilities to independently complete the caching, transcoding and distribution of VR panoramic videos. For a large number of MEC edge nodes, according to the clustering algorithm, the establishment of edge clusters not only improves the edge computing power, but also promotes the horizontal cooperation between MEC servers. The edge cluster clustering algorithm is established based on the following basic principles. Firstly, on the basis of considering the service capacity balance between clusters, edge clusters are divided according to the location area of MEC edge nodes. Then, the MEC edge node at the edge cluster boundary is determined by the principle that its service content is similar to the service content of the surrounding edge clusters. Finally, the cluster head is determined according to the capabilities of computing and caching.

2.2. Data Perception-Driven Clustered-Edge Transmission Model

Based on the popularity of FoV video tiles predicted by user data, this paper determines the intelligent decision of caching FoV tiles in VR panoramic video. For the main FoV video tiles, we give priority to caching. With the consideration of user network condition and edge network capabilities, the clustered edge will provide a suitable bitrate version of FoV video tiles as high as possible to obtain better QoE. In this way, with joint consideration of the user's network condition and viewport change prediction and the edge nodes' caching and computing capabilities, the federated deep reinforcement learning method is used to adaptively adjust collaborative caching and bitrate adaptation decisions for achieving data perception-driven VR panoramic video clustered-edge transmission. The specific process is shown in Figure 2.

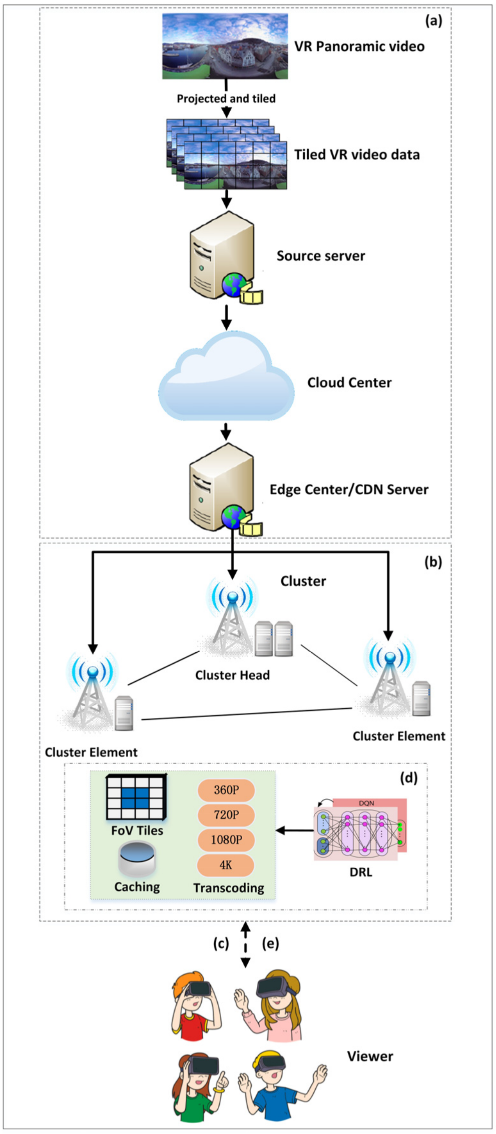


Figure 2. Data perception-driven clustered-edge transmission model for VR panoramic video.

(a) Acquire VR panoramic video: The VR panoramic video is obtained by collecting the VR panoramic video capture device. Then, the VR panoramic video is projected and tiled to obtain streaming media files of a series of 2D image chunks, which are transmitted to the edge CDN server through the cloud center backbone network.

(b) Cache the VR panoramic video: According to the historical popularity of users in this area served by MEC edge nodes in the cluster, the FoV video tiles of VR panoramic video are cached.

(c) VR panoramic video service request: According to the user's VR panoramic video request, the agent obtains information such as the user's viewport selection, network quality and requested bitrate version.

(d) VR panoramic video service decision: According to the user's viewport selection, the network quality and the load situation of MEC edge nodes in a clustered edge, the proposed FDRL-CBA algorithm is used to obtain edge caching and bitrate adaptation decisions.

(e) Implement VR panoramic video services: Due to the edge caching and bitrate adaptation decisions, the corresponding MEC node in a clustered edge implements the task of updating video cache state and transcoding. Then, the clustered edge provides VR panoramic video services to the user.

2.3. Cache Model

The cache model is mainly considered in terms of the rebuffering time of the receiver and the cache hit rate of the cached FoV video tiles.

(a) Rebuffer model: Rebuffering often occurs in the condition that the buffer occupancy $L(t)$ is low, when the playback time of the buffered video chunks is less than the time to download a new video chunk. The calculation of rebuffering time is computed as

$$\begin{gathered} R_b(t) = \max(d(t) - T(t), 0) \\ d(t) = \frac{B_k(i,t)}{W(t)} = \frac{\sum_{x=0}^{X}\sum_{y=0}^{Y} p_{x,y}B_{x,y,k}(i,t)}{W(t)} \end{gathered} \tag{1}$$

where $T(t)$ is the playback buffer video time, $d(t)$ is the total download time for a new video chunk and $W(t)$ represents the wireless transfer rate experienced during time period t. $B_k(i,t)$ is the video bitrate that serves user i at time t. The bitrate for the *x-th* row and *y-th* column tile in the *k-th* chunk is marked as $B_{x,y,k}(i,t)$. The symbol $p_{x,y}$ represents the average viewing probability of the *x-th* row and *y-th* column tile.

(b) Cache hit rate model: The cache hit rate of T requests in the time period can be obtained as

$$CRH(t) = \frac{\sum_{i=1}^{T} \mathrm{l}(H_i)}{T} \tag{2}$$

where $\mathrm{l}(H_i)$ is an indicator function.

2.4. Bitrate Adaptation Model

The bitrate adaptation model of VR panoramic video is mainly considered from the network cost required by the video service and the video quality pursued by users. It mainly includes three parts: video quality, bandwidth cost and transcoding cost.

(a) Video quality: Generally speaking, the bitrate version of the video determines the quality of the video, with higher bitrate videos having higher video quality. The quality of the *x-th* row and *y-th* column tile in the *k-th* chunk can be denoted as

$$q_{x,y,k}(t) = F_t(B_{x,y,k}(i,t)) \tag{3}$$

where the function $F_t(\cdot)$ maps a bitrate to the video quality [16]. Then, the quality of *k-th* chunk can be calculated as

$$q_k(t) = \sum_{x=0}^{X}\sum_{y=0}^{Y} p_{x,y}q_{x,y,k}(t) \tag{4}$$

(b) Bandwidth cost: The corresponding bandwidth cost $C_b(t)$ of MEC servers in one cluster will be generated during the VR panoramic video service process [17]. It can be defined as

$$C_b(t) = \sum_{n=1}^{N} P(n,t)W(n,t) \tag{5}$$

where $P(n,t)$ represents the price of bandwidth and N is the number of MEC severs in one cluster. $W(n,t)$ is the bandwidth cost in MEC server n at time stage t, which can be computed by summing all users' serving video bitrate, which is connected to the MEC server n at time stage t.

(c) Transcoding cost: Generally, the input bitrate, target bitrate, the number of central processing unit (CPU) cores for transcoding and the video length need to be considered in the video pricing model [17]. The video transcoding cost can be denoted as

$$C_t(t) = \sigma(B_{input} - B_{target}) \cdot D \cdot N_{cpu} \tag{6}$$

where B_{input}, B_{target} represent the bitrate of input and target video, respectively. N_{cpu} is the number of CPU cores required for the transcoding task, σ is an adjustable parameter and D is the time length of the video.

Since there are wired connections between edge nodes in the cluster, which can meet the sufficient transmission needs, the load of the federated learning model update can be ignored compared with the bandwidth cost and transcoding cost.

3. Federated Deep Reinforcement Learning-Based Caching and Bitrate Adaptation

This part mainly introduces our proposed optimization objective function and intelligent optimization algorithm based on federated deep reinforcement learning.

3.1. Problem Formulation

In this paper, we formulate the dynamic optimization problem of caching and bitrate adaptation as a Markov Decision Process to achieve high-quality service of VR panoramic video by maximizing the cumulative reward of the VR panoramic video streaming service. Therefore, we define the objective function as

$$\begin{aligned} \max_{M(t),U(t),B_{x,y,k}(i,t)} \ & J(t) = E\left[\sum_{t=0}^{T-1} \gamma^t R(t)\right] \\ s.t.\ & C1: M(t) \in \{0,1,\cdots,N\}, \forall t \\ & C2: U(t) \in \{0,1\}, \forall t \\ & C3: B_{x,y,k}(i,t) \in \{B_1, B_2, \cdots, B_{\max}\}, \forall t \\ & C4: \ \gamma \in (0,1]. \end{aligned} \tag{7}$$

3.2. FDRL-CBA Algorithm

(a) Agent design: In the VR panoramic video service, due to the user network conditions and the network resource status of the clustered edge, the mathematical model of the basic elements such as state, action and reward in the DRL algorithm is established.

- State: According to the clustered-edge transmission model for VR panoramic video, the state space is set as

$$S(t) = \{C_b(t), L(t), R_b(t), q_k(t)\} \tag{8}$$

- Action: The agent action includes caching and bitrate transcoding of FoV tiles. Then, the action space can be described as

$$A(t) = \left\{M(t), U(t), B_{x,y,k}(i,t)\right\} \tag{9}$$

- Reward: Based on the purpose of VR panoramic video service, the reward function includes the cache hit rate, video quality and four penalty items. In the reward, the weighted sum of the short- and long-term cache hit rate for each step can be obtained as $CRH_{sl}(t) = CRH_s(t) + \mu * CRH_l(t)$, where μ is the weight to balance the short- and long-term cache hit rate. The penalty items include video quality changes, video rebuffering time, bandwidth cost and video transcoding cost. Then, the reward space can be defined as

$$\begin{aligned} R(t) = \mu_1 CRH_{sl}(t) + \mu_2 q_k(t) - \mu_3 |q_k(t) - q_k(t-1)| \\ -\mu_4 \mathrm{R}_b(t) - \mu_5 \mathrm{C}_b(t) - \mu_6 \mathrm{C}_t(t) \end{aligned} \tag{10}$$

- QoE: The receiver rendering technology is not considered in this paper, so the QoE of VR panoramic video mainly considers the quality of FoV tiles, cache hit rate and rebuffering time.

$$\mathrm{QoE}_k = \mu_1 CRH_{sl}(t) + \mu_2 q_k(t) - \mu_3 |q_k(t) - q_k(t-1)| - \mu_4 \mathrm{R}_b(t) \tag{11}$$

(b) Training Methodology of FDRL-CBA: Integrating federated learning into the implementation of deep reinforcement learning, federated deep reinforcement learning is proposed to solve this problem. Based on the proposed HC-MEC network model, this paper constructs a federated deep reinforcement learning-based caching and bitrate adaptation scheme, which is shown in Figure 3.

In this framework, the FDRL-CBA algorithm proposed in this paper is implemented based on the clustered edge. Generally, macro base stations and micro base stations are selected as edge nodes. They are located in a local area, which can promote the efficiency of federated learning to a certain extent. Under the HC-MEC network structure, hierarchical federated deep reinforcement learning can also be implemented, including FDRL within edge clusters and FDRL among edge clusters. This paper mainly focuses on intra-cluster federated deep reinforcement learning.

In Figure 3, we show the general process of federated deep reinforcement learning within an edge cluster. The training of deep reinforcement learning is carried out simultaneously on each MEC edge node server in an edge cluster. The cluster head is the federated learning aggregation server, which is responsible for model aggregation and updating. After the local deep learning of each MEC edge node is completed, the local network model parameters are immediately uploaded to the cluster head server through the backhaul link. Then, the overall model is aggregated and updated. After aggregation is complete, each edge node downloads the latest network model and uses local data to validate the update. Then, the cluster server performs the next round of federated learning until the iteration conditions or times are met.

In the clustered-edge network in this paper, the load of federated learning model update can be ignored because of sufficient transmission requirements by wired connections between edge nodes in the cluster. In the proposed FDRL-CBA algorithm, we adopt the Federated Averaging (FedAvg) aggregation algorithm; its iterative model is given as follows:

$$w_t^h = \frac{\sum_{\forall e \in E} w_t^e}{N} \tag{12}$$

where w_t^h is the aggregated weight that the cluster head has obtained from the cluster element's neural network models in the time stage t. N is the number of cluster elements. E is a collection of edge nodes within a cluster that send their weight vectors, denoted as w_t^e, to the cluster head. The local weight vectors are acquired at each cluster element by training based on their own local data.

Then, the FDRL-CBA algorithm will be strictly executed according to Algorithm 1.

Algorithm 1: Federated Deep Reinforcement Learning-Based Caching and Bitrate Adaptation (FDRL-CBA)

```
Input: the video requests information of users $U_{requests}$, the video service environment information at time $t$ $\{C_b(t)$,$L(t)$,$R_b(t)$,$q_k(t)\}$
Output: bitrate of FoV tiles in VR panoramic video $B_{x,y,k}(i,t)$, cache edge node location $M(t)$, update cache status $U(t)$
Initialization:
Initialize the DNN parameters $w_t^h$ at cluster head with random weights
Initialize the replay memory $\mathrm{D_m}$ to edge node $N_m$
Initialize clustered-edge network service matrix $V_h$ of requests
for iteration = 1, 2, . . . , $T_1$ do
    Download parameters $w_t^h$ from cluster head server and assign them to the DNN model of each edge node
    Generate the DNN parameters $w_t^e$ at each cluster element through local data training by implementing DRL-CBA
    Each edge node uploaded the parameters $w_t^e$ of the training model to the cluster head
    Perform DNN model aggregation at cluster head by calculating as illustrated in Equation (12)
    Assign $w_t^h$ to the DNN model of cluster head
    Implement the DRL-CBA algorithm using the cluster header's local data
    Update DNN model parameters $w_t^h$
    Update clustered-edge network service matrix $V_h$ of requests
end for
```

Sub-algorithm: Deep Reinforcement Learning-Based Caching and Bitrate Adaptation (DRL-CBA)

```
Initialization:
Initialize the DNN parameters by $w_t^h$
Initialize MEC service matrix $V_e$ of requests
for episode = 1, 2, . . . , $M$ do
    Generate the users' requests data
    Observe initial state $s_1$ as illustrated in Equation (8)
    for $t$ = 1, 2, . . . , $T_2$ do
        Give a random probability $\varsigma \in [0,1]$
        Choose action $A(t)$ which listed in Equation (9) as
$A(t) = \begin{cases} a^*(t) = \arg\ \max\limits_{a} Q(s,a;\theta),\ \varsigma > \varepsilon \\ a(t) \neq a^*(t), randomly\ \ select\ \ a(t), others \end{cases}$
        Observe the reward $R(t)$, state $S(t+1)$
        Store the transition ($S(t)$, $A(t)$, $R(t)$, $S(t+1)$) into Buffer pool $\mathrm{D_m}$
        Update MEC network service matrix $V_e$ of requests
        Sample random minibatch of transitions ($S(t)$,$A(t)$,$R(t)$,$S(t+1)$) from Buffer pool $\mathrm{D_m}$
        Set $y_j = \begin{cases} r_j & for\ terminal\ s' \\ r_j + \gamma \max_{a'} Q(s',a';\theta_{i-1}) \big| s,a) & for\ non-terminal\ s' \end{cases}$
        Implement a gradient descent step according to $L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot)}\left[(y_i - Q(s,a;\theta_i))^2\right]$
        Update the parameters within the eval network
        Reset the parameters within the target network in every G time stages
        Update the learning rate
    end for
end for
```

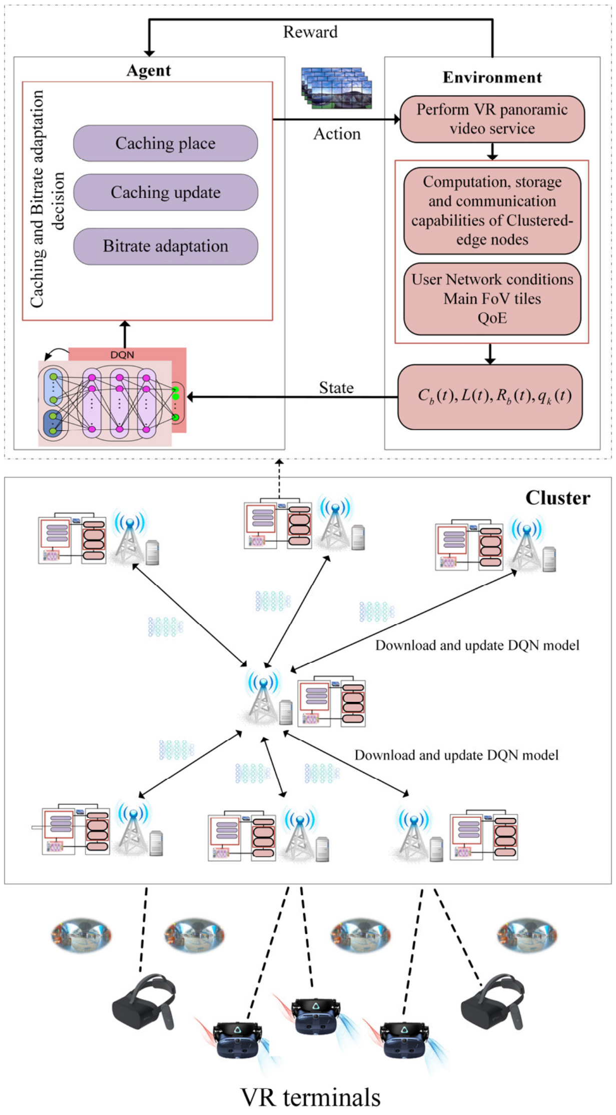

Reward
Agent
Caching and Bitrate adaptation decision
Caching place
Caching update
Bitrate adaptation
DQN
Action
Environment
Perform VR panoramic video service
Computation, storage and communication capabilities of Clustered-edge nodes
User Network conditions
Main FoV tiles
QoE
State
$C_b(t), L(t), R_b(t), q_k(t)$
Cluster
Download and update DQN model
Download and update DQN model
VR terminals

Figure 3. The VR panoramic video service scheme based on FDRL-CBA within an edge cluster.

4. Performance Evaluation

We verify the performance of the proposed FDRL-CBA algorithm for VR panoramic video services in clustered-edge networks through simulation experiments. In order to demonstrate the effectiveness of the clustered edge transmission model and the proposed FDRL method, simulation experiments for comparative methods of FDRL-CBA without transcoding, DRL-CCT [11] and DRL-CCT for single MEC are carried out. The DRL-CCT method is acquired by transforming the application of the DRL-CCT algorithm into VR panoramic video transmission. The experiments use the Python platform for simulation, and the federated deep reinforcement learning modules are implemented by TensorFlow.

4.1. Experiment Setup

In the experiments, we set up seven edge nodes in one edge cluster to provide video services for 30 users in the local area. There are 10 VR panoramic videos for the edge nodes to cache in advance. The mapped VR panoramic videos' data can be divided into 30 chunks with a duration of 10 s, and each chunk can be decomposed into 4 × 8 FoV video tiles of the same size. The available bitrate set for each tile is {1 Mbps, 2 Mbps, 4 Mbps, 10 Mbps, 20 Mbps}. The cached video is generally the highest bitrate version. Then, we generate the user data of requests randomly. The video data of users' requests followed the Zipf distribution.

In the deep Q network (DQN), we built a four-layer network structure with two hidden layers of size 256 and 512, and the neural network was fully connected. The loss function we used in the deep neural network (DNN) is the mean square error. During the training process, as the learning progresses, we continuously reduce the exploration probability of the greedy algorithm to promote its convergence as soon as possible. The size of the experience reply pool in DQN is set to 2000, and the batch size in Stochastic Gradient Descent (SGD) is 32. The parameters used in the experiment are given in Table 1.

Table 1. Summary of major parameters' values.

Parameter	Value	Parameter	Value
D	10 s	μ_4	1.0
γ	0.9	μ_5	0.1
learning rate	0.01	μ_6	0.1
μ	0.6	$p_{x,y}$	1/32
σ	1.2	$\{N_{cpu}\}$	{2, 4, 6, 8}
μ_1	1.2	X	4
μ_2	1.5	Y	8
μ_3	1.0	ε	0.9

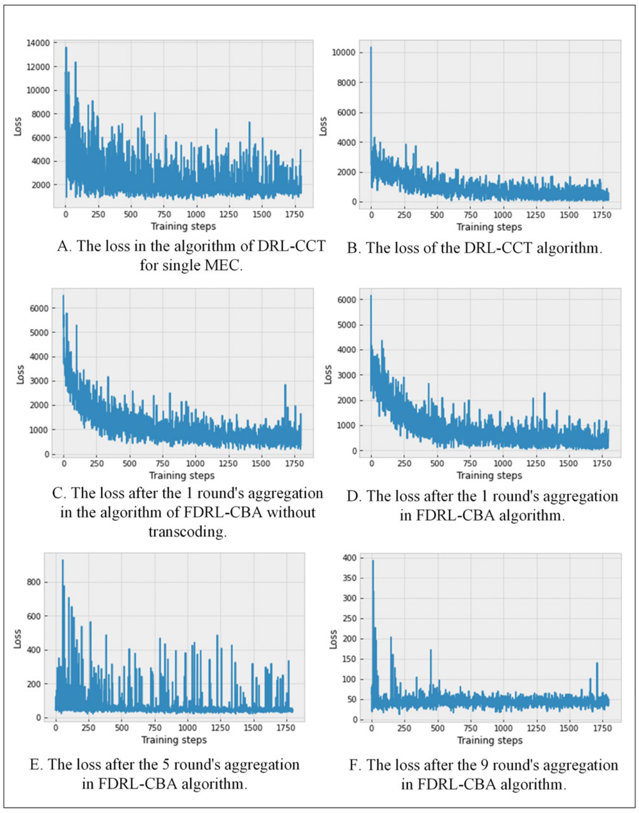
4.2. Performance Analysis

In simulation experiments, the above comparison methods are simulated under our experimental framework. In DRL-based methods (DRL-CCT, DRL-CCT for single MEC), the DQN agents runs 40 episodes, and one episode contains 50 VR video service requests. To facilitate comparison, DQN agents also apply the same settings in FDRL-based methods (FDRL-CBA, FDRL-CBA without transcoding). The federated learning of DQN is conducted in 10 rounds. Based on the characteristics of deep reinforcement learning, for all experimental algorithms, all simulation results are obtained from an average of 20 algorithm executions.

Figure 4 shows a comparison of the convergence of the four methods. It can be seen from the figure that the loss value of all methods has volatility, which is determined by the characteristics of deep reinforcement learning. However, with the progress of federated learning, the loss value range of the FDRL-based methods gradually becomes smaller, especially the FDRL-CBA method. Moreover, compared with the DRL-based methods, the loss value of federated deep reinforcement learning converges faster after aggregation, and the training efficiency is higher, which confirms that federated learning can improve the training effect of deep reinforcement learning.

Figure 5 shows how the reward changes as learning progresses. As can be seen from Figure 5, compared to the other three algorithms, the reward stability of our proposed FDRL-CBA algorithm is better. With the progress of federated learning, the fluctuation bars in the reward function continue to become sparser, indicating that the fluctuation of the reward becomes smaller and remains more stable.

Figure 6 is a comparison of cache hit ratios among the four methods. It can be seen from the experimental results that the cache hit rate of FDRL-CBA algorithm is the highest. Especially when the cache resources are insufficient, the advantage of our proposed algorithm in cache hit rate is more obvious. In addition, the clustered edge-based method has a significantly higher cache hit rate than the single MEC method. It shows that the video service mode based on the clustered edge improves the video cache hit rate compared with the independent MEC server video service mode.



A. The loss in the algorithm of DRL-CCT for single MEC.

B. The loss of the DRL-CCT algorithm.

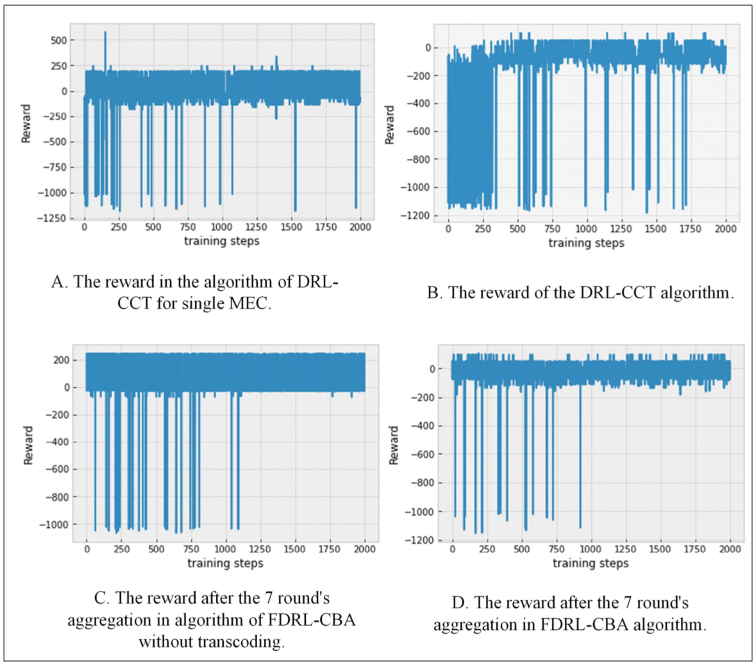
C. The loss after the 1 round's aggregation in the algorithm of FDRL-CBA without transcoding.

D. The loss after the 1 round's aggregation in FDRL-CBA algorithm.

E. The loss after the 5 round's aggregation in FDRL-CBA algorithm.

F. The loss after the 9 round's aggregation in FDRL-CBA algorithm.

Figure 4. Comparison of the loss performance.



A. The reward in the algorithm of DRL-CCT for single MEC.

B. The reward of the DRL-CCT algorithm.

C. The reward after the 7 round's aggregation in algorithm of FDRL-CBA without transcoding.

D. The reward after the 7 round's aggregation in FDRL-CBA algorithm.

Figure 5. Comparison of the reward performance.

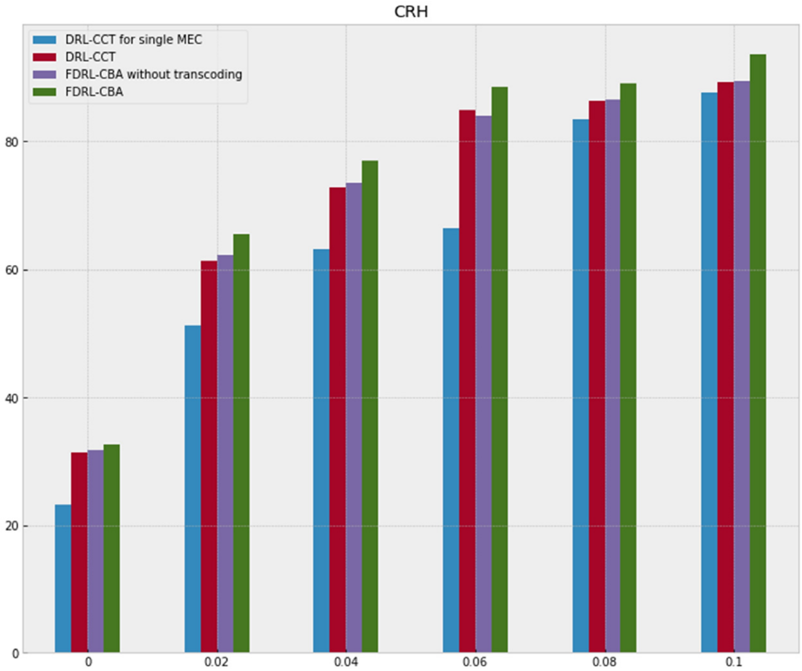


Figure 6. Comparison of cache hit ratio in different cache ratios among the four algorithms.

The QoE performance in different algorithms is given in Figure 7. In terms of QoE performance of users, the algorithm based on FDRL-CBA achieved stability and showed the best performance. Compared with the DRL-based algorithm, the FDRL-CBA algorithm improved the video experience quality. This is because federated learning makes full use of distributed training data, which improves the accuracy of the DRL model to a certain extent. The QoE performance of the FDRL-CBA method outperforms the method of FDRL-CBA without transcoding, which shows the necessity of joint optimization of caching and transcoding. Therefore, the simulation results are consistent with the theoretical results. In conclusion, the method proposed in this paper achieves better performance in terms of cache hit rate and QoE.

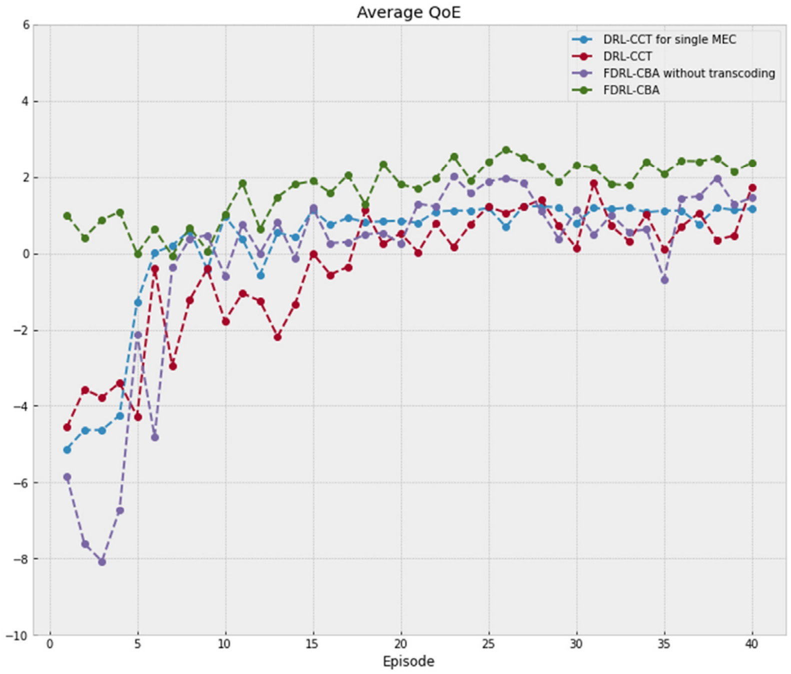


Figure 7. The QoE performance in different algorithms.

5. Conclusions and Future Work

In view of the needs of VR panoramic video services, this paper constructed Hierarchical Clustered MEC Networks to provide sufficient computing and caching resources for caching and bitrate adaptation in VR panoramic video services. The proposed network architecture can support the collaboration between MEC edge nodes within the edge cluster and realize a data perception-driven clustered-edge transmission model for VR panoramic video. Then, the algorithm of federated deep reinforcement learning-based caching and bitrate adaptation was designed to implement adaptive caching and transcoding for VR panoramic video. The proposed method achieves better video service performance in terms of cache hit rate and QoE.

The cost consumption of various methods was not analyzed in this paper, and it also only focuses on the bitrate version of VR panoramic video, which does not consider the video format, mapping method, etc. In addition, many details of VR video transmission need further research, such as VR panoramic video quality perception and evaluation methods, federated learning aggregation methods and improvement of deep reinforcement learning. In the future, we will further explore the details of VR panoramic video edge services based on the proposed network architecture, including intra-cluster edge nodes' load balancing and collaboration between clusters.

Funding: This research was funded by the National Natural Science Foundation of China (no. 61961021) and the Science and Technology Project of Jiangxi Education Department (no. GJJ180251).

Data Availability Statement: The data in this paper are available from the corresponding author upon request.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper. Available online: https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white-paper-c11-738429.html (accessed on 9 March 2020).
2. 360-Degree Camera Market Analysis—Global Forecast by 2018–2024. Available online: https://www.marketresearchengine.com/360-degree-camera-market (accessed on 6 November 2019).
3. Hu, Y.C.; Patel, M.; Sabella, D.; Sprecher, N.; Young, V. Mobile edge computing-a key technology towards 5G. *ETSI White Paper* **2015**, *11*, 1–16. Available online: https://5ghub.us/5g-mobile-edge-computing-mec/ (accessed on 1 September 2015).
4. Zhao, J.; Sun, X.; Li, Q.; Ma, X. Edge caching and computation management for real-time internet of vehicles: An online and distributed approach. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 2183–2197. [CrossRef]
5. Zhao, J.; Li, Q.; Gong, Y.; Zhang, K. Computation offloading and resource allocation for cloud assiste d mobile edge computing in vehicular networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7944–7956. [CrossRef]
6. Mehrabi, A.; Siekkinen, M.; Ylä-Jääski, A. Edge computing assisted adaptive mobile video streaming. *IEEE Trans. Mob. Comput.* **2019**, *18*, 787–800. [CrossRef]
7. Tran, T.X.; Pompili, D. Adaptive bitrate video caching and processing in mobile-edge computing networks. *IEEE Trans. Mob. Comput.* **2019**, *18*, 1965–1978. [CrossRef]
8. Kan, N.; Zou, J.; Li, C.; Dai, W.; Xiong, H. RAPT360: Reinforcement learning-based rate adaptation for 360-degree video streaming with adaptive prediction and tiling. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 1607–1623. [CrossRef]
9. Chu, T.M.C.; Zepernick, H.-J. Performance analysis of an adaptive rate scheme for QoE-assured mobile VR video streaming. *Computers* **2022**, *11*, 69. [CrossRef]
10. Jiang, Z.; Zhang, X.; Xu, Y.; Ma, Z.; Sun, J.; Zhang, Y. Reinforcement learning based rate adaptation for 360-degree video streaming. *IEEE Trans. Broadcast.* **2021**, *67*, 409–423. [CrossRef]
11. Cheng, Q.; Shan, H.; Zhuang, W.; Yu, L.; Zhang, Z.; Quek, T.Q.S. Design and analysis of MEC- and proactive caching-based 360° mobile VR video streaming. *IEEE Trans. Multimed.* **2022**, *24*, 1529–1544. [CrossRef]
12. Liu, S.; Zheng, C.; Huang, Y.; Quek, T.Q.S. Distributed reinforcement learning for privacy-preserving dynamic edge caching. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 749–760. [CrossRef]
13. Wan, Z.H.; Li, Y. Deep reinforcement learning-based collaborative video caching and transcoding in clustered and intelligent edge B5G networks. *Wireless Commun. Mobile Comput.* **2020**, *2020*, 6684293. [CrossRef]
14. Wang, X.F.; Han, Y.; Leung, V.C.M.; Niyato, D.; Yan, X.; Chen, X. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 869–904. [CrossRef]

15. Abreha, H.G.; Hayajneh, M.; Serhani, M.A. Federated learning in edge computing: A systematic survey. *Sensors* **2022**, *22*, 450. [CrossRef]
16. Chen, M.; Ponec, M.; Sengupta, S.; Li, J.; Chou, P.A. Utility maximization in peer-to-peer systems with applications to video conferencing. *IEEE/ACM Trans. Netw.* **2012**, *20*, 1681–1694. [CrossRef]
17. Zheng, Y.; Wu, D.; Ke, Y.; Yang, C.; Chen, M.; Zhang, G. Online cloud transcoding and distribution for crowdsourced live game video streaming. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 1777–1789. [CrossRef]