Towards Coverage-Aware Cooperative Video Caching in LEO Satellite Networks

Ruili Zhao[†], Yongyi Ran[*], Jiangtao Luo[*], Shuangwu Chen[‡]
School of Communication and Information Engineering
Chongqing University of Posts and Telecommunications, Chongqing, China[*†]
University of Science and Technology of China, Hefei, China[‡]
[*]{ranyy, luojt}@cqupt.edu.cn
[†]{D200101029}@stu.cqupt.edu.cn,[‡]{chensw}@ustc.edu.cn

***Abstract*—Video services such as short video sharing have exploded due to the rapid development of Internet social media platforms. Caching video segments on satellites effectively shortens service delay and speeds up video sharing, especially for users without terrestrial Internet access. However, where to place what video and how to replace it in time is by no means an easy task, requiring careful consideration of many factors, e.g., satellite coverage, video popularity, and limited caching resource. In this paper, we propose a coverage-aware cooperative video caching algorithm (CACVC) that considers the prevalence of video in the coverage area and the collaboration between adjacent satellites. In CACVC, we model the cache placement problem of video as a Partially Observable Markov Decision Process (POMDP) to optimize the service delay of video provided by access satellites, neighboring satellites, or ground stations. We derive the optimal cache strategy by utilizing Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm with a centralized training and distributed execution paradigm. Simulation results show that the cache hit ratio can be improved by 4%$\sim$18%, and the average service delay can be reduced by 1%$\sim$14%.**

***Index Terms*—Cooperative Caching, Multi-Agent System, Deep Reinforcement Learning, Satellites Network, Delivery Delay**

I. Introduction

Satellite networks are widely considered a powerful supplement to existing terrestrial mobile communication networks due to the global coverage, seamless access, and infrastructure-free features [1]. Compared with traditional medium and high orbit satellites, low-earth orbit (LEO) satellites have a shorter service delay, lower signal attenuation, and lower operation and maintenance costs [2–4]. With the development of intelligent mobile devices, the demand for low-latency mobile applications and multimedia services has increased significantly. According to Cisco's VNI report [5], IP video traffic will quadruple by 2022, accounting for 82% of total IP traffic. As a solution to this global traffic growth, LEO satellite caching is considered a promising approach. By pre-caching popular videos on LEO satellites, ground user terminals (UTs) can retrieve their requested videos mainly from the cached videos, significantly reducing repetitive video traffic and video service delay.

Although caching on LEO satellites has been widely used, besides the limited cache capacity of satellites, the high-speed movement of satellites imposes additional challenges to the design of a caching strategy for LEO satellites. First, satellites' coverage areas and access users are constantly changing, leading to dynamic video popularity and frequent replacement of cached videos. Second, the same video segments are repeatedly cached between satellites, resulting in low cache space utilization within the satellite constellation. The cache policy needs to dynamically decide which videos need to be cached and where the selected videos are cached (e.g., access satellite or neighbor satellite) by using the dynamics of cache space and video popularity. Third, large-scale constellations and large amounts of video will result in high computation and communication cost problems.

Several studies have begun considering caching schemes on satellites to overcome these obstacles [6–11]. By caching popular content favored by users on the satellite, the optimized caching strategy is adopted to improve the service performance of the satellite in different scenarios and achieve more efficient content distribution in the satellite network. In a single satellite scene design [6–8], the content transmission is usually interrupted due to the satellite's movement. A cache placement scheme for multiple LEO satellites is proposed in [9–11]. These proposals, however, only focus on the optimal placement of cache contents at a specific time, and there is not enough cooperation between satellites, resulting in poor cache space utilization.

In order to address the issues mentioned above, we propose a coverage-aware cooperative video caching algorithm (CACVC) based on Multi-Agent Deep Reinforcement Learning (MADRL) [12] to optimize the video average service delay. We firstly formulate a Partially Observable Markov Decision Process (POMDP) model by designing state space, action space, and reward function according to video request, video cache, video popularity, and video transmission delay within the streets of coverage (SOC). Then, we utilize Multi-Agent Deep Deterministic Policy Gradient (MADDPG) to find the optimal caching strategy. MADDPG works in a centralized training and decentralized execution paradigm. Each agent

*The corresponding authors are Yongyi Ran and Jiangtao Luo.

*This work is jointly supported by National Natural Science Foundation of China (No.62171072, 62172064, 62003067, 62101525), Natural Science Foundation of Chongqing (cstc2021jcyj-msxmX0586) and Chongqing Postgraduate Research and Innovation Project (CYB21204).

makes actions based on its local observations and is trained with all the agents' observations and actions. Our main contributions are summarized as follows:

- We devise a statistical model of video popularity based on SOC for our cooperative caching problem, where the access users in SOC are relatively fixed, which is beneficial for learning users' preferences.
- We formulate the cooperative caching problem as a POMDP-based multi-agent decision problem maximizing the cumulative reward for all satellite caches. This formulation ensures caching collaboration among satellites by optimizing the service delay of cached video from access satellites, cached video from neighboring satellites, and ground stations (GSs).
- Compared with baseline caching strategies, the proposed CACVC algorithm, according to simulation results, can reduce $1\%{\sim}4\%$ of the video average delivery delay and improve the $4\%{\sim}18\%$ hit ratio.

The remainder section of this paper is arranged as follows. Section II summarizes the work related to satellite caching. In Section III, the system model and problem formulation are presented. The details of our proposed CACVC are described in Section IV. In addition, the evaluation results are detailed within Section V. Finally, the conclusion and future work are explained in Section VI.

II. Related Work

In recent years, existing LEO satellite caching strategies can be roughly categorized into single and Multiple satellites.

The first type is a caching strategy that considers only caching multimedia content on a single satellite. Zhong et al. [6] researched Quality of Experience (QoE)-driven placement optimization of video stream caching by considering the required video stream transfer rate and the social relationship between users. Han et al. [7] researched joint cache placement and content delivery for the scenario of multiple base stations and one satellite in a satellite-ground integrated cloud radio access network. The cooperative caching of the Base Stations (BSs) and the users are considered in the satellite backhaul transmission. In [8], the authors discussed the caching problem in the satellite-to-ground relay network.

The second type is a caching strategy that considers multiple satellites cooperating to cache multimedia content. In [9], in order to minimize the content access delay of user terminals, a new caching algorithm was proposed, which optimizes the content placement in the LEO satellite constellation network. Qiu et al. [10] expressed the collective problem of network, cache, and computing resource allocation as a joint optimization problem and used the deep Q-learning method to solve the problem in the software-defined satellite-ground network. In [11], the satellite-ground hybrid network was used for offline edge caching of cellular base stations to reduce the traffic on the ground network. Nevertheless, the above works scarcely consider the impact of changing-with-time satellite network topology on the file distribution process, leading to updating caching location with degraded distribution performance.

III. System Model And Problem Formulation

In this section, we discuss our system model, in particular the service model. Then the problem formulation is given.

A. System Model

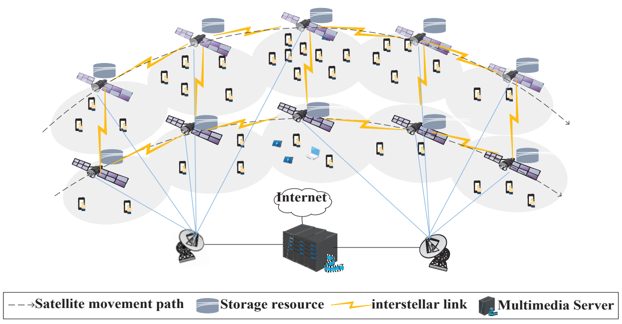

Fig. 1: Cooperative caching system over LEO satellite networks.

As shown in Fig. 1, a typical LEO satellite network for video delivering comprises a set of LEO satellites $\mathcal{n} = \{1, 2, \ldots N\}$, a number of UTs $\mathcal{k}$=$\{1, 2, \ldots K\}$ and a set of GSs $\mathcal{g}$=$\{1, 2, \ldots G\}$. The satellite network is deployed with essential inter-satellite links (ISL, in the same orbital and between orbitals) to achieve wide coverage areas and low access latency. Each satellite has a certain storage capacity to cache some popular videos. UTs get the requested videos by accessing the satellites. Although the storage capacity of a single satellite is limited, the cached videos can be shared between satellites through inter-satellite links, which allows us to coordinate satellites to take full advantage of the storage capacity of satellites. Therefore, the fundamental problem is to develop an efficient cooperative video caching strategy that minimizes the service delay of the requested videos, which is the optimization goal of this paper. The mathematical models of the system mentioned above can be described as follows.

goal, not a problem

1) Coverage Model: The coverage of the satellite network to the ground has become an essential indicator in measuring the communication capabilities of this satellite network. Satellites move according to orbits, and satellites in the same orbit form fixed SOC during the movement, as shown in Fig. 2.

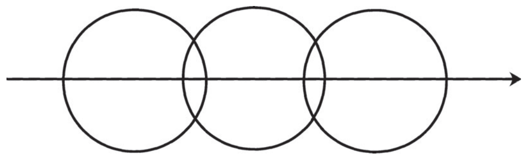

Fig. 2: The SOC of satellites.

The point set $A_n(t)$ of all points within the area covered by satellite n can be express in [13]:

$$A_n(t) = \{x(R,\theta,\Phi)\,|\,\sin\theta\sin\theta_n\cos(\Phi-\Phi_n)+\cos\theta\cos\theta_n \le \cos\psi\} \tag{1}$$

where R is the earth radius, Φ_n is the longitude of satellite n, $\theta_n = \left|\frac{\pi}{2} - n_{\text{latitude}}\right|$, n_{latitude} is the latitude of satellite n, ψ is the half cone angle of the covered area to the core of the earth. $x(R,\theta,\Phi)$ represents the coordinate of any node.

So the point set $A_{SOC}(t)$ of all points can be express in:

$$A_{SOC}(t) = \{A_n(t)\,|\,n = 1, 2, \ldots N\} \tag{2}$$

2) Video Request Model: The UTs are uniformly distributed in SOC. The access satellite n is requested by UT k for video segments, so the request status can be expressed as $\boldsymbol{r}_n(t) = \left\{r^l_{n,k,f}(t) \mid n \in \boldsymbol{n}, f \in \boldsymbol{f}, k(R,\theta,\Phi) \in A_{SOC}(t)\right\}$, $k(R,\theta,\Phi)$ denotes the coordinate of k. $r^l_{n,k,f}(t) = 1$ indicates satellite n receives a request f^l from k; otherwise $r^l_{n,k,f}(t) = 0$.

3) Video Caching Model: We only consider the scenario where the UTs may receive the requested video segments from the accessed satellite. Let $\mathscr{F}=\{1,\cdots,f,\cdots,F\}$ denote the set of videos requested by the UTs. All the F videos can be retrieved from the multimedia server. Different videos consist of different numbers of segments, each of which has the same segment size. We assume that each satellite has the same caching capacity. The video placement is refreshed periodically, and the time slot is indexed by $t = 0, 1, \cdots$. Let $c^l_{n,f}(t) = 1$ denote the video segment f^l cached in the satellite n. The full segments that have been cached in satellite n have to satisfy the constraint:

$$\sum_{f\in\mathscr{f}}\sum_{l\in\ell} c^l_{n,f}(t) b^l_f \le c_n(t), \forall n \in \mathscr{N} \tag{3}$$

where b^l_f is the size of video segment f^l, $c_n(t)$ is the cache state of access satellite n.

4) Popularity Model: Due to the high-speed movement of satellites, the prevalence of video segments within satellite coverage at different moments may exhibit geographical differences. The access users in SOC are relatively fixed, which is beneficial to the statistics and training of popularity. Let $p_{n,o}$ denotes the popularity of the o-th video segment at satellite n, which follows the Mandelbrot-Zipf (MZipf) distribution [14]. Thus we can get:

$$p_{n,o} = \frac{I_n(o)^{-z_n}}{I_{SOC}(o)^{-z_n}} \tag{4}$$

where $I_n(o)$ indicates the popularity rank of the o-th video segment at sateliite n, $I_{SOC}(o)$ indicates the popularity rank of the o-th video segment at SOC, z_n is a skewness factor taking values in $[0.6, 1.2]$.

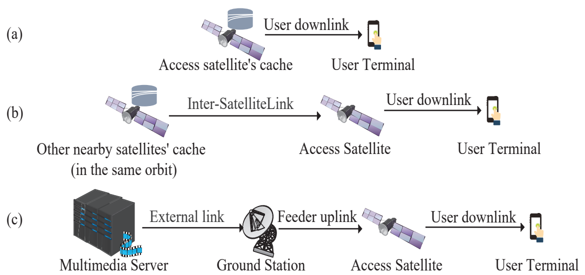

Fig. 3: The service paths for different video placement.

5) Service Model: In the cooperative caching system, the UTs' requests may be accommodated by the GS, access satellite, or nearby satellite, as depicted in Fig. 3, depending on the video placements. We consider the video service delay of UTs as the transmission delay.

Access service mode ($m^{ac} = 1$) : If the requested video segments have been cached in the access satellite, they can be delivered to the UTs directly, as shown in Fig.3 (a). $d^{ac}_n(t)$ indicates the video service delay, which can be calculated:

$$d^{ac}_n(t) = \sum_{k\in\mathscr{k}}\sum_{f\in\mathscr{f}}\sum_{l\in\ell} \frac{c^l_{n,f}(t) r^l_{n,k,f}(t) b^l_f}{v_{n,k}} \tag{5}$$

$$v_{n,k} = B_{n,k}\log_2\left(1 + \frac{q_k g_{n,k}}{\sigma^2 + \sum_{v\in\mathscr{K}\setminus\{k\}:a_v=a_k} q_v g_{n,v}}\right) \tag{6}$$

where $v_{n,k}$, $B_{n,k}$ indicates the video transmission rate and channel bandwidth of satellite n to UT k, respectively. σ^2 represents the background noise power, q_k is the power consumption of n transmission to k, $g_{n,k}$ is the channel gain.

Cooperative service mode ($m^{co} = 1$) : In the cooperative caching system, the segments belonging to the same video may be distributed to multiple satellites. If the access satellite has no segments for the requested video, but nearby satellites cached them, the UT requests can be accommodated by sharing the video segments among those satellites. Due to the fairly limited cache space, if a requested content segment is fetched from a nearby satellite, it will be directly forwarded to the UT. $d^{co}_n(t)$ indicates the video service delay of this mode, as shown in Fig.3 (b), which can be calculated:

$$d^{co}_n(t) = d^{ac}_n(t) + d^H_n(t) \tag{7}$$

where $d^H_n(t)$ is the transmission delay between satellites, $d^H_n(t) = \sum_{k\in\mathscr{k}}\sum_{f\in\mathscr{f}}\sum_{l\in\ell} \frac{H^l_{n,k,f} c^l_{n,f}(t) r^l_{n,k,f}(t) b^l_f}{v_{co}}$. $H^l_{n,k,f}$ is the number of hops between satellites when f^l is sent to k (in the coverage of satellite n). v_{co} indicates the average transmission rate between satellites, $v_{co} = \frac{1}{N}\sum_{n=1}^{N} v_{n,n-1}$. $v_{n,n-1}$ can be calculated according to formula (6).

Source service mode ($m^{so} = 1$) : If the requested video is not cached by satellites, the UT will fetch it from the GS, as shown in Fig.3 (c). $d^{so}_n(t)$ indicates the video service delay of this mode, which can be calculated:

$$d^{so}_n(t) = d^{ac}_n(t) + \sum_{k\in\mathscr{k}}\sum_{f\in\mathscr{f}}\sum_{l\in\ell}\sum_{g\in\mathscr{g}} \frac{r^l_{n,k,f}(t) b^l_f}{v_{g,n}} \tag{8}$$

$v_{g,n}$ is the transmission rate of GS g to satellite n which can be calculated according to formula (6).

B. Problem Formulation

The satellites need to dynamically determine what video should be replaced and where the video request should be served during each episode to adapt to changing environments. Our optimization goal is to minimize the average service delay for requested videos.

d^{avg} is the average service delay of requested videos during the T period, which can be defined as:

$$d^{avg} = \frac{\sum_{t\in\mathscr{t}}\sum_{n\in\boldsymbol{n}}[m^{ac}d^{ac}_n(t) + m^{co}d^{co}_n(t) + m^{so}d^{so}_n(t)]}{\sum_{t\in\mathscr{t}}\sum_{n\in\boldsymbol{n}}\sum_{k\in\mathscr{k}}\sum_{f\in\mathscr{f}}\sum_{l\in\ell}\left|r^l_{n,k,f}(t)\right|} \tag{9}$$

Thus, we formulate the corresponding cooperative caching problem to minimize the objective function and obtain the optimal control policy, which can be expressed as:

$$
\begin{aligned}
\min \quad & d^{avg} \\
\text{s.t.} \quad & C_1 : m^{ac}, m^{co}, m^{so}, c_{n,f}^{l}(t), r_{n,k,f}^{l}(t) \in \{0,1\} \\
& C_2 : m^{ac} + m^{co} + m^{so} = 1 \\
& C_3 : \sum_{f \in \mathscr{f}} \sum_{l \in \ell} c_{n,f}^{l}(t) b_f^l \leq c_n, \forall n, f, t, l \\
& C_4 : H \leq \frac{v_{co}}{v_{g,n}}, \forall n, g
\end{aligned}
\tag{10}
$$

where C_2 guarantees that the video request can be served; C_3 ensures that the total size of video cached on the satellite should not exceed its cache capacity.

IV. The Proposed Coverage-Aware Cooperative Video Caching Algorithm

The above optimization problem is modeled as a POMDP [15, 16], where the information observed by each satellite is only a partial glimpse of the constellation state. In this section, the CACVC algorithm is proposed to solve the average service delay optimization problem.

A. POMDP

The fact that the complete information concerning the states is not entirely observable when making caching decisions motivates us to formulate our caching decision problem as a POMDP. We define the states, actions, and rewards as follows:

- **State Space:** At time slot t, we define the state of the whole system is defined as:

$$\boldsymbol{\chi}(t) = (\boldsymbol{r}_n(t), \boldsymbol{c}_n(t), \boldsymbol{p}_n(t)) \tag{11}$$

where $\boldsymbol{r}_n(t)$ is the state of the request received by satellite n, $\boldsymbol{c}_n(t)$ is the video cache state. $c_{n,f}^{l}(t) = 1$ means satellite n cached f^l; otherwise $c_{n,f}^{l}(t) = 0$. Finally, $\boldsymbol{p}_n(t)$ is the video popularity state, $\boldsymbol{p}_n(t) = \left\{p_{n,f}^{l}(t)\right\}$, $p_{n,f}^{l}(t)$ is the video popularity of f^l under satellite n.

- **Action Space:** In order to accommodate the dynamic changes of the video popularity and satellites, each satellite need to determine which video segments should be replaced and where the video requests should be served during each episode. We define three types of actions following: access actions $\boldsymbol{a}^{ac}(t) = \left\{a_{n,f,l}^{ac}(t)\right\}$, cooperative actions $\boldsymbol{a}^{co}(t) = \{a_n^{co}(t)\}$, and GS actions $\boldsymbol{a}^{so}(t) = \left\{a_g^{so}(t)\right\}$. Therefore, the action vector can be expressed as:

离散动作空间

$$\boldsymbol{\Phi}(\boldsymbol{\chi}(t)) = \{\boldsymbol{a}^{ac}(t), \boldsymbol{a}^{co}(t), \boldsymbol{a}^{so}(t)\} \tag{12}$$

For the access actions, $a_{n,f,l}^{ac} \in \{0,1\}$, $a_{n,f,l}^{ac} = 1$ means that f^l cached in satellite n needs to be replaced by the video segment of currently requested; otherwise $a_{f,l}^{ac} = 0$. For the cooperative actions, $\boldsymbol{a}^{co}(t) = [a_1^{co}(t), \ldots, a_N^{co}(t)]$, $a_n^{co}(t) = 1$ indicates the current request is processed by the cooperative satellite n. For the GS actions, $a_g^{so}(t) = 1$ means the requested video should be provided by GSs.

- **Reward Function:** When the system takes action $\mathbf{A}$ in state $\mathbf{S}$, the system will receive feedback rewards. According to the request processing procedure discussed above, the video requested by UTs may have been cached by access satellite, or maybe cached in the adjacent satellite of access satellite. To achieve the maximum system reward and to ensure the average video service delay is minimized, we use a negative exponential function to normalize the reward function, so the reward function is:

$$
\boldsymbol{R}_n\left(\boldsymbol{\chi}(t), \boldsymbol{\Phi}\left(\boldsymbol{\chi}(t)\right)\right) =
\begin{cases}
p_{n,f}^{l} e^{-\frac{d_n^{ac}(t)}{P_n(t)}}, & \text{Access} \\
p_{n,f}^{l} e^{-\frac{d_n^{ac}(t) + d_n^{H}(t)}{P_n(t)}}, & \text{Cooperation} \\
p_{n,f}^{l} e^{-\frac{d_n^{ac}(t) + d_n^{g}(t)}{P_n(t)}}, & \text{Source} \\
0, & \text{Else}
\end{cases}
\tag{13}
$$

where P_n is cache hit ratio of access satellite n, $P_n = \frac{\sum_{r \in r_n(t)} c_n^r(t)}{|r_n(t)|}$. $r_n(t)$ is the requests received in t, $c_n^r(t) \in \{0,1\}$, $c_n^r(t) = 1$ means satellite n has cached the requested video, otherwise $c_n^r(t) = 0$. $p_{n,f}^{l} e^{-P_n d_n^{ac}(t)}$ is the reward that a UT obtains video segments directly from access satellite. $p_{n,f}^{l} e^{-P_n(d_n^{ac}(t) + d_n^{H}(t))}$ means the UT is served by the satellite-satellite cooperation. When a UT has to be served by GSs, the reward is $p_{n,f}^{l} e^{-P_n(d_n^{ac}(t) + d_n^{g}(t))}$. Otherwise, the reward is 0.

B. CACVC: Coverage-Aware Cooperative Video Caching

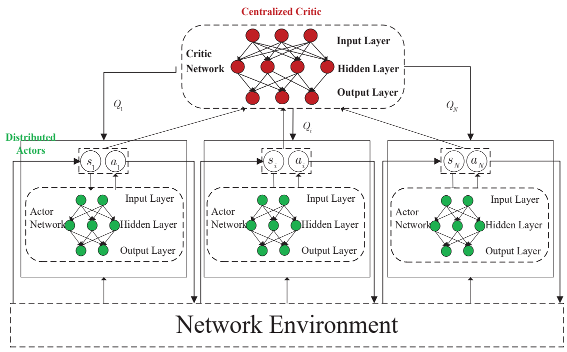

Fig. 4: The multi-agent actor-critic cooperation framework.

In Fig.4, actor and critic are built based on the neural network, where the actor is used to learn the local policy function to choose actions based on the state information from the environment and decide to replace the video segments cached on the satellite. The state information includes video request state, current video cache state, and video popularity state. The critic evaluates the decision of the actor based on the video segment delivery delay and guides the actor to update the policy better so that the probability of better actions increases and the probability of worse actions decreases. When multiple satellites need to make caching decisions, video requests and popularity are the external environments. The negative exponential form of service delay for the entire satellite network is used as the reward for all nodes. The critic input in MADDPG contains information about the actions of all agents so that it can converge to a better state more consistently and faster. The output values of the critic network are used to train the parameters of the actor-network.

Suppose the policy set of n agents is $\pi = \{\pi_1, \pi_2, \ldots \pi_n\}$. $\theta = \{\theta_1, \ldots \theta_n\}$ represents the strategy of n agents. For the cumulative expected reward of the i-th agent, the policy gradient is defined as: $\nabla_{\theta_i} \mathrm{J}(\mu_i) = \mathrm{E}_{o,a \sim D}\left[\nabla_{\theta_i} \mu_i(a_i \mid o_i) \nabla_{a_i} Q_i^{\mu}(o, a_1, \ldots, a_n) \mid a_i = \mu_i(o_i)\right]$, where $Q_i^{\mu}(o, a_1, \ldots, a_n)$ is a centralized action-value function, $o = \{o_1, \ldots o_n\}$ contains the observations.

Thus, we use this gradient to update the actor-network. We need to calculate the mean square error between it and target network as the loss to update parameters for the critic network.

$$\mathscr{L}(\theta_i) = \mathrm{E}_{o,a,r,s'}\left[\left(Q_i^{\mu}(s, a_1, a_2, \ldots, a_n) - y\right)^2\right] \tag{14}$$

$$y = r_i + \gamma \overline{Q_i}^{\mu'}(x', a_1', \ldots, a_n')\Big|_{a_j' = \mu_j'(o_j)} \tag{15}$$

Where $\overline{Q_i}^{\mu'}$ is the target network, $\mu' = [\mu_1', \mu_2', \ldots, \mu_n']$ represents the delayed update parameters of the target network θ_j'. The strategies of other agents can be obtained by fitting approximation without communication interaction.

Finally, according to the description above, the proposed CACVC algorithm for LEO satellite networks can be summarized in Algorithm 1.

Algorithm 1 The proposed CACVC Algorithm

Initialize: the discount factor γ, the maximum learning episode **EP**, the replay buffer **D**, the weights and learning rate of the actor network and the critic network, random process Ψ, the network layout with N LEO satellites.

```
for episode = 1 to EP do
    Each satellite receives the initial state o_i^t according to the task
    requirements and network environment, and the global state s^t.
    for each step t = 1 to T do
        For each agent i, select action a_i = μ_θi(o_i) + N_t w.r.t.
        Execute actions a = (a_1, ..., a_N), observe reward r and new state s'.
        Store (s, a, r, s') in replay buffer D.
        Set s ← s'.
        for agent i = 1 : N do
            Sample a random minibatch of J samples (s^j, a^j, r^j, s'^j) from D.
            Set y according to (14)
            Update the critic network and the actor network.
        end for
        Update the target Q network: θ'_i ← τθ_i + (1 − τ)θ'_i
    end for
end for
```

V. Performance Evaluation

This section evaluates the performance of our proposed CACVC algorithm through comprehensive simulation and comparison with different baseline algorithms.

TABLE I: PARAMETERS SCHEME

Parameter	Value	Parameter	Value
Number of Satellites	11	Learning rate η	0.001
Satellite orbit height	780 Km	Mini-batch size	8
Orbit inclination	86°	Weight decay coefficient	0.0001
Half cone angle ψ	62°	Discount factor	0.9
Elevation mask	8.2°	Number of epochs	200
Latitude threshold	60°	Target network update rate	0.01
Lengths of the videos	4-15s	Exploration fading factor	0.9
Video resolution	720p	Experience replay buffer size	50000
Storage capacity c_n	100-500MB	Reward discount factor γ	0.95
Number of video	10000-30000	Initial exploration coefficient	0.03

A. Simulation Settings

To verify the effectiveness of CACVC, we built a satellite network simulation scenario based on STK and python 3 by taking Iridium as the reference. The satellite network consists of 11 satellites evenly distributed in one orbit. Total 2000 UTs are uniformly distributed in SOC. Satellite parameter setting main reference [17]. We use three FC networks in the actor-network at most, with 256, 128, and 64 neurons, respectively. The same setting for the critic network. The main parameters are listed in Table I.

B. Baseline Algorithms and Metrics

We compare our proposed CACVC algorithm with four baseline solutions:

- First Input First Output (FIFO): The oldest content in the system will be replaced first.
- Least Recently Used (LRU): The least recently used content in the system will be replaced first.
- Least Frequently Used (LFU): The least frequently used content in the system will be replaced first.
- Exchange-Stable Matching (ESM): A stable exchange matching algorithm based on matching game theory [9].

Besides, for quantitatively evaluating the performance of our proposed method, the following performance metrics are used:

1) Hit ratio: The cache hit ratio is usually determined by whether the requested video exists in the local video library. In the CACVC algorithm, since the requested video can be obtained from adjacent satellites, UTs can still obtain lower video service delay. Therefore, we propose the cache hit ratio as a performance indicator, which is defined by the following:

$$P^{hit} = \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_{r\in r_n(t)}\left[c_n^r(t) + (1 - c_n^r(t))\, c_{adjacent}^r(t)\right]}{|r_n(t)|} \tag{16}$$

where $r_n(t)$ is the requests received in a slot t, $c_{adjacent}^r(t) \in \{0, 1\}$ is the cache status of adjacent satellite, $c_{adjacent}^r(t) = 1$ means the adjacent satellite has cached video of request r, otherwise $c_{adjacent}^r(t) = 0$.

2) Average service delay d^{avg}: the average service delay for all requested video within the T period.

C. Convergence Results & Performance Comparison

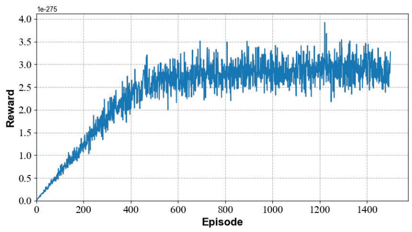

Fig. 5: The convergence speed of CACVC.

The learning process of CACVC is shown in Fig. 5. The reward increases rapidly up to about 500 episodes and then gradually stabilizes. This indicates the learning algorithm converges after about the 500 training episodes, and then the well-trained network can be used for accurately estimating value function.

We plot the average video service delay and hit ratio in Fig. 6 and 7, respectively. On service delay, CACVC achieves the lowest average delay of 1515ms, improving the performance of 1.2%, 12%, 13.8%, and 12.9% compared to ESM, LRU, LFU, and FIFO, respectively. On hit ratio, almost

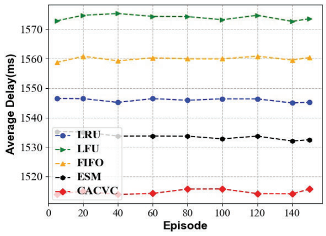

Fig. 6: Average video service delay.

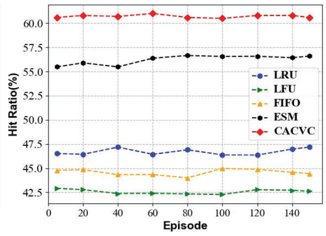

Fig. 7: Cache hit ratio.

60% video requests are satisfied by the CACVC algorithm, and it outperforms ESM, LRU, LFU, and FIFO algorithms with up to 4%, 14%, 18%, and 15% improvements. While individual satellite has limited storage space, CACVC enables the sharing of cached video between satellites, thus avoiding redundant caching of the same video among satellites, which saves storage space with satellites. In contrast, LRU, LFU, and FIFO make caching decisions independently so that satellites may cache duplicate video. They may cache much less video than CACVC in the limited storage space. Meanwhile, since the deep learning model characterizes the variation of content popularity better than the heuristic model, CACVC achieves a higher cache hit ratio and a lower average delay than ESM.

D. Sensitivity Analysis

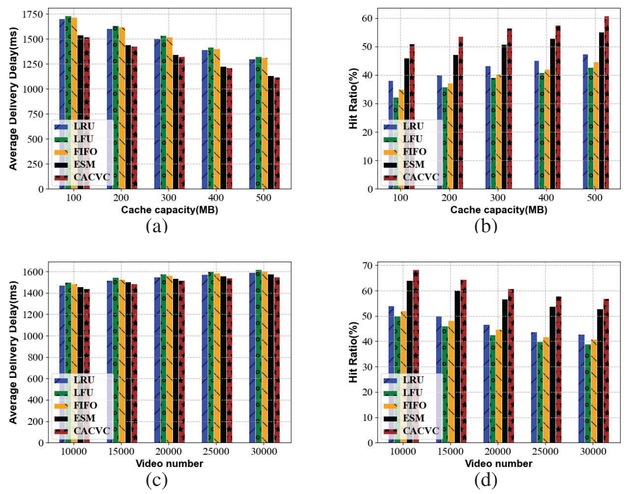

Fig. 8: (a) average delay, (b) hit ratio versus cache capacity; (c) average delay, (d) hit ratio versus video number.

Fig. 8 demonstrates the network performance under different cache capacities and video numbers. According to Fig. 8 (a), (b), the cache hit ratio increases, and the average video service delay decreases in all five algorithms as the cache capacity increases (the video number is 20000). Compared to the LRU, LFU, FIFO, and ESM algorithms, the video cache hit ratio of the proposed CACVC improved by 13.17%, 17.7%, 16%, and 5.5%; the video service delays of proposed CACVC is reduced by 180ms, 208ms, 194ms, and 18ms. Because when the cache capacity is large enough to store more videos, replacement processes rarely occur.

According to Fig. 8 (c), (d), the average video service delay increases, and the cache hit ratio decreases as the number of videos increases. The CACVC algorithm reduces the average video service delay by 2%, 3.8%, 2.9%, and 1.2% compared to the LRU, LFU, FIFO, and ESM algorithms, respectively. The main reason is that when more popular videos are cached on the satellite, the newly requested videos are not cached.

VI. Conclusion

In this paper, we formulated the cooperative caching problem as a POMDP-based multi-agent decision problem, which jointly optimized the service delay of fetching videos from local satellites, adjacent satellites, and GSs. We conceive a CACVC algorithm based on Multi-Agent Deep Reinforcement Learning to solve this problem. Our experimental results verified that the proposed CACVC algorithm facilitated the collaboration between satellites and thus reduced the service delay of fetching videos while improving the cache hit ratio.

In the future, we plan to consider caching base stations, vehicles, and other devices in the system. Each device with caching enabled can act as a cooperation cache node to share content caching with LEO satellites and other devices.

References

[1] X. Zhu, C. Jiang, L. Yin, L. Kuang, N. Ge, and J. Lu, "Cooperative multigroup multicast transmission in integrated terrestrial-satellite networks," *IEEE JSAC*, vol. 36, no. 5, pp. 981–992, May 2018.

[2] X. Jia, T. Lv, F. He, and H. Huang, "Collaborative data downloading by using inter-satellite links in leo satellite networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1523–1532, Mar. 2017.

[3] J. Li, H. Lu, K. Xue, and Y. Zhang, "Temporal netgrid model-based dynamic routing in large-scale small satellite networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6009–6021, Jun. 2019.

[4] J. Li, K. Xue, D. S. L. Wei, J. Liu, and Y. Zhang, "Energy efficiency and traffic offloading optimization in integrated satellite/terrestrial radio access networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2367–2381, Apr. 2020.

[5] Cisco, "Cisco visual networking index: Forecast and trends,2017–2022 white paper." *Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.*

[6] G. Zhong, J. Yan, and L. Kuang, "Qoe-driven social aware caching placement for terrestrial-satellite networks," *China Commun.*, vol. 15, no. 10, pp. 60–72, Oct. 2018.

[7] D. Han, H. Peng, H. Wu, W. Liao, and X. S. Shen, "Joint cache placement and content delivery in satellite-terrestrial integrated c-rans," in *Proc. IEEE Int. Conf. Commun.*, pp. 1–6, 2021.

[8] K. An, Y. Li, X. Yan, and T. Liang, "On the performance of cache-enabled hybrid satellite-terrestrial relay networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1506–1509, Oct. 2019.

[9] S. Liu, X. Hu, Y. Wang, G. Cui, and W. Wang, "Distributed caching based on matching game in leo satellite constellation networks," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 300–303, Feb. 2018.

[10] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Vehic. Tech.*, vol. 68, no. 6, pp. 5871–5883, June 2019.

[11] A. Kalantari, M. Fittipaldi, S. Chatzinotas, T. X. Vu, and B. Ottersten, "Cache-assisted hybrid satellite-terrestrial backhauling for 5g cellular networks," in *Proc. IEEE Glob. Commun. Conf.*, pp. 1–6, 2017.

[12] Z. Hao, G. Xu, Y. Luo, H. Hu, J. An, and S. Mao, "Multi-agent collaborative inference via dnn decoupling: Intermediate feature compression and edge learning," *IEEE Trans. Mobile Comput.*, pp. 1–16, 2022.

[13] H. Zhou, L. Liu, and H. Ma, "Coverage and capacity analysis of leo satellite network supporting internet of things," in *Proc. IEEE Int. Conf.Commun. (ICC)*, pp. 1–6, May 2019.

[14] H. Wu, J. Li, H. Lu, and P. Hong, "A two-layer caching model for content delivery services in satellite-terrestrial networks," in *Proc. IEEE Glob. Commun. Conf.*, pp. 1–6, 2016.

[15] Y. Lyu, Z. Liu, R. Fan, C. Zhan, H. Hu, and J. An, "Optimal computation offloading in collaborative leo-iot enabled mec: A multi-agent deep reinforcement learning approach," *IEEE Trans. Green Commun. Netw.*, pp. 1–1, 2022.

[16] C. Zhan, H. Hu, S. Mao, and J. Wang, "Energy-efficient trajectory optimization for aerial video surveillance under qos constraints," in *IEEE INFOCOM Workshop*, pp. 1559–1568, 2022.

[17] F. Tang, H. Zhang, and L. T. Yang, "Multipath cooperative routing with efficient acknowledgement for leo satellite networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 179–192, Jan. 2019.