Multi-user Edge-assisted Video Analytics Task Offloading Game based on Deep Reinforcement Learning

Yu Chen[†], Sheng Zhang[†], Mingjun Xiao[§], Zhuzhong Qian[†], Jie Wu[‡], and Sanglu Lu[†]
[†]State Key Lab. for Novel Software Technology, Nanjing University, P.R. China
[§]School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China, P.R. China
[‡]Center for Networked Computing, Temple University
Email: sheng@nju.edu.cn

***Abstract*—With the development of deep learning, artificial intelligence applications and services have boomed in the recent years, including recommendation systems, personal assistant and video analytics. Similar to other services in the edge computing environment, artificial intelligence computing tasks are pushed to the network edge. In this paper, we consider the multi-user edge-assisted video analytics task offloading (MEVAO) problem, where users have video analytics tasks with various accuracy requirements. All users independently choose their accuracy decisions, satisfying the accuracy requirement, and offload the video data to the edge server. With the utility function designed based on the features of video analytics, we model MEVAO as a game theory problem and achieve the Nash equilibrium. For the flexibility of making accuracy decisions under different circumstances, a deep reinforcement learning approach is applied to our problem. Our proposed design has much better performance compared with some other approaches in the extensive simulations.**

Keywords*—*edge computing; video analytics; task offloading; decentralized algorithm; game theory; Markov decision process; deep reinforcement learning;

I. Introduction

With the emergence of smart devices and numerous new applications, network traffic is growing rapidly. The conventional centralized network architecture cannot meet the needs of users due to high transmission delay and heavy loads on the backhaul links. Therefore, mobile edge computing (MEC) has been proposed, and its main characteristic is to bring the computation and storage resources to the edge of networks. It connects users directly to the nearest service-enabled edge networks and provides computing and caching capabilities. In the past years, lots of issues [1]–[3] related to edge computing have been studied, such as multi-user resource allocation, optimal network control, etc.

Meanwhile, with the development of deep learning, artificial intelligence (AI) applications and services have boomed in the recent years, including recommendation systems, personal assistant and video surveillance [4], [5]. Since 2009, Microsoft has conducted continuous research on what kinds of AI applications should be transferred to the edge, ranging from real-time video analytics, VR/AR, voice command recognition, interactive cloud gaming, etc. Among them, real-time video analytics is envisioned as a killer application in the edge computing environment. Most of video analytics applications running at MEC servers process the video data to detect some specific configurable events, such as causing-trouble vehicle, abandoned luggage, lost child. Video analytics tasks continuously collect a tremendous amount of high-definition videos, and it requires high bandwidth, high computation and low latency. Thus, the edge computing is regarded as a promising solution to meet the strict requirements.

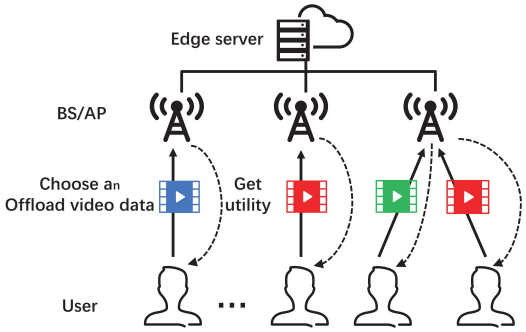


Fig. 1: Multi-user edge-assisted video analytics task offloading

In the edge computing environment, a large number of users offload their video data to the network edges for the video analytics services. In video analytics tasks, video frames are extracted at various sampling rates, compressed into different resolutions, and processed by convolutional neural network (CNN) models [6]. Same as the works in [7], [8], we refer to the combination of frame rate and resolution as a configuration. Obviously, different configurations lead to different accuracies and resource consumptions. In our previous work [7], we study the specific relationship between the analytics accuracy and configuration. To obtain high video analytics accuracy, users have incentives to raise the frame rate and resolution, which results in more video data offloaded to the edge.

For different users, they may have different requirements on the accuracy of video analysis results. Some applications involving privacy protection, like phone unlocking by face identification, may require extremely high accuracy of video analysis. However, other applications don't, like recording traffic flow with surveillance cameras. Usually, the edge network resources are limited, and users interact with others to obtain a stable allocation of network resources and meet the video analytics service requirements. At this point, the approach of game theory (GT) can be applied to enhance the usage of the networking edge resources. It can be used to analyze the interactions among multiple self-interested and independent players [9], [10], and helps to design a decentralized system, where no player will deviate unilaterally.

Game theory is a branch of applied mathematics, and it has been used to formulate, design and optimize the operations in many networking scenarios [11]–[13]. Usually, these scenarios involve multiple players with conflicting goals, and we know from some research [14], [15] that GT has an important role in analyzing the network algorithms and optimizing the configuration parameters. However, to the best of our knowledge, little existing work has focused on applying the game theory method to the video analytics task offloading problem so far.

We study the Multi-user Edge-assisted Video Analytics task Offloading (MEVAO) problem (shown in Fig. 1), where users have the video analytics tasks with various accuracy requirements. Since different video analytics configurations lead to different accuracies, all users independently choose their accuracy decisions, satisfying the accuracy requirements, and offload the video data with corresponding configuration to the edge server. To obtain a stable situation where none of them have an incentive to change the accuracy decision unilaterally, we formulate the MEVAO problem as a GT problem. For it, we design the appropriate utility function for each user based on the video analytics features. We first propose a decentralized algorithm for MEVAO, which can achieve Nash equilibrium (NE) with information sharing (e.g. connection bandwidth, accuracy requirement). However, users in the real world may be unwilling to share their personal information because of security and privacy concerns. Thus, we apply the deep reinforcement learning (RL) approach based on the Advantage Actor Critic (A2C) model to our problem, and extend it to the scenario without information sharing.

The contributions of this paper are summarized as follows.

- We study the multi-user edge-assisted video offloading and analyzing problem, and formulate it as a GT problem by designing the appropriate utility function based on the video analytics features. We propose the algorithm which achieves the NE to solve the problem.
- We apply the deep reinforcement learning approach to our problem without information sharing, and propose the RL-based algorithm to tackle the problem. Based on the A2C model, users adjust their accuracy decisions and finally achieve the converged reward.
- By extensive simulations, our design has better performance when compared with some other approaches, and we study how the parameters in our design influence the simulation results in various settings.

The remainder of this paper is organized as follows. We discuss the related works in Section II. Section III presents the description and formulation of our problem. In Section IV, we give the algorithm design for our problem based on the game theory. In Section V, we apply the deep RL approach to our problem without information sharing. Section VI evaluates the performance of our design and compares it with other existing approaches using extensive simulations. Finally, Section VII concludes the paper and discusses some possible future work.

II. Related Work

Mobile edge computing. Many research efforts in the past years have been carried out in the field of MEC, with respect to storage, latency and computational offloading. Jalali et al. [16] propose a time-based and flow-based energy consumption model, and conduct a number of experiments using centralized nano data centers, which can lead to energy savings. Jararweh et al. [17] design a software defined system for MEC (SDMEC) and software defined storage is the focus of the proposed framework that enables applications requiring storage resources to benefit from SDMEC. Kumar et al. [18] propose a smart grid data management scheme based on vehicular delay-tolerant network, with which the data is transmitted to multiple smart grid devices in the MEC environments.

Video analysis and processing. Some existing work has studied the video analytics in MEC. Instead of processing the video analytics in the central cloud [19], the system can avoid the network congestion caused by video uploading, and benefit from the low latency by offloading the video analytics task to the edge. Ren et al. [20] propose a multiuser video compression offloading approach and minimize the latency in local compression, partial compression and edge cloud compression offloading scenarios. Kang et al. [21] design the NoScope to accelerate video analysis by using a difference detector that highlights temporal differences across frames.

Game theory application. There are some works that apply the game theory to the multiuser computation offloading problem and provide a solution by achieving the Nash equilibrium. Chen et al. [22] propose an efficient computation offloading model based on the game theory, which simultaneously helps connected users to decide the correct wireless channels based on the strategic interactions. Zhan et al. [23] design a decentralized offloading game in which each user decides the portion of task offloaded to the edge server. Hu et al. [24] design a minority game based scheme, where the tasks are divided into subtasks to form some groups, and the subtasks left are scheduled to adjust the decisions in a probabilistic way.

To the best of our knowledge, very little existing work has focused on applying the game theory method to the problem of video analytics task offloading. We study the multi-user edge-assisted video offloading and analyzing problem, where all users independently choose their accuracy decisions satisfying the accuracy requirement, and offload the video data to the edge server. Our problem MEVAO is formulated as a game theory problem, and we design the appropriate utility function for each user based on the features of video analytics. Furthermore, we finally solve it by using the RL approach.

III. Problem Description

In this section, we present the description of our problem. We design the utility function based on the features of video analytics task offloading and formulate the problem. Some important notations are listed in Table I.

We consider a set of N users, denoted by $\{1, 2, ..., N\}$, each of which has a video analytics task to be executed. They are connected to the same edge server nearby, on which the video analytics applications are deployed. As shown in Fig. 1, each user offloads the data of its video analytics task to the edge server. Some applications involving privacy protection, like phone unlocking by face identification, require very high accuracy of video analysis. However, other applications don't, like recording traffic flow using surveillance cameras. For

TABLE I: List of Important Notations

Notation	Description
N	Number of users
M_n	Minimum requirement on accuracy for user n
a_n	Accuracy decision of user n
$\boldsymbol{a}_{-n}$	Accuracy decisions of users except n
$F(a_n)$	Frame rate when accuracy decision is a_n
r_n, s_n, t_n	Fitting parameters related to user n in function F
$T(a_n)$	Transmission cost when accuracy decision is a_n
K	Size of each video frame
b_n	Network bandwidth assigned to user n
$C(a_n)$	Computation allocation when accuracy decision is a_n
E	Amount of computation resource at edge server
$Sat(a_n)$	Accuracy satisfaction when accuracy decision is a_n
$\alpha_n, \beta_n, \gamma_n$	Weight of $T(a_n)$, $C(a_n)$ and $Sat(a_n)$ in utility for user n
$u_n(a_n, \boldsymbol{a}_{-n})$	Utility for user n

different users, they may have different minimum requirements on the accuracy of video analysis results. We let M_n denote the minimum requirement on the video analysis accuracy for user n. The user n can choose the accuracy decision a_n satisfying the accuracy requirement, which means that $M_n \leq a_n \leq 1$. For instance, when user n choose the $a_n = 1$, the video analysis result has the highest accuracy. However, when a_n does not exceed M_n, the result fails to meet the minimum accuracy requirement. More generally, if user n has no requirement on the analysis result, we can set $M_n = 0$.

A. Utility Function Design

1) Transmission cost:

Same as the existing works [7], [8], we refer to the combination of frame rate and resolution as a configuration, and different configurations lead to different accuracies and resource consumptions. In some video analysis applications, the video data is captured from the surveillance cameras [8], where the video resolution is fixed and frame rate can be adjusted. In this paper, we mainly focus on the influence of the frame rate on the analysis accuracy.

In our previous work [7], the specific relationship between the analytics accuracy and the frame rate is derived from our real experiments. We implement YOLO [25], an object detector CNN on NVIDIA Jetson TX2 (shown in Fig. 2) to perform vehicle counting on the clips from surveillance videos. The results are plotted in Fig. 3, and the relationship between the analytics accuracy and the frame rate can be formulated as a fitted convex function

$$F(a_n) = \frac{1}{r_n}(e^{a_n - s_n} - t_n), \tag{1}$$

where r_n, s_n and t_n are the fitting parameters for user n. For example, we vary the frame rate from 2fps to 30fps, and the blue line in Fig. 3 can be fitted as $\frac{1}{0.348}(e^{a_n - 1.828} + 7.177)$.

To obtain high video analytics accuracy, users have incentives to improve the frame rate, which results in much more video data offloaded to the edge server. And we formulate the transmission cost as

$$T(a_n) = \frac{K \cdot F(a_n)}{b_n} = \frac{K \cdot \frac{1}{r_n}(e^{a_n - s_n} - t_n)}{b_n}, \tag{2}$$

where K is the size of each video frame, and b_n is the network bandwidth assigned to user n. It is clear that when the accuracy decision is raised, the transmission cost will increase.

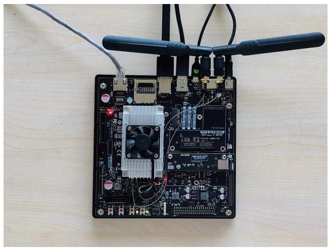


Fig. 2: We implement YOLO on NVIDIA Jetson TX2

2) Computation allocation:

We let E denote the amount of computation resource at the edge server. Following the works in [26], [27], the edge server allocates the computational resource to the users depending on the proportion of their uploaded video data amount on the server. Thus, the computation allocation for user n is

$$C(a_n) = E \cdot \frac{\frac{1}{r_n}(e^{a_n - s_n} - t_n)}{\sum_{i=1}^{N} \frac{1}{r_i}(e^{a_i - s_i} - t_i)}. \tag{3}$$

As shown in (3), the more video data user n offloads to the edge server, the more computation resources it can obtain. Specifically, when the user number is 1, the only user can get all the computation resources from the edge server.

3) Accuracy satisfaction:

If we use the deep learning approach like CNN for video analysis, the accuracy will be more difficult to improve when it is close to 100% [7]. Thus, the user will feel more satisfied with the analysis result if the accuracy can increase from 95% to 100%, compared to that from 85% to 90%. This property of accuracy satisfaction is consistent with the convex functions, so we use a convex function to formulate it. In this paper, we describe the accuracy satisfaction as

$$Sat(a_n) = e^{a_n}. \tag{4}$$

We should mention that the exponential function (4) is taken as an example to formulate the accuracy satisfaction in the paper, but actually we can also use any other convex function to formulate accuracy satisfaction for utility function.

B. Problem Formulation

In terms of transmission cost, computation allocation and accuracy satisfaction, user n's utility function is defined as

$$u_n(a_n, \boldsymbol{a}_{-n}) = -\alpha_n T(a_n) + \beta_n C(a_n) + \gamma_n Sat(a_n), \tag{5}$$

where $\boldsymbol{a}_{-n} = (a_1, ..., a_{n-1}, a_{n+1}, ..., a_N)$ is the accuracy decisions from all users except user n. The three positive coefficients α_n, β_n and γ_n mean the weights of transmission cost, computation allocation and accuracy satisfaction for user n. Usually, we set the parameters α_n, β_n and γ_n satisfying $\alpha_n + \beta_n + \gamma_n = 1$. From (5), we can intuitively see that more computation allocation, higher accuracy satisfaction and less transmission cost lead to the higher utility for each user.

Given other users' decisions $\boldsymbol{a}_{-n}$, each user n will choose the optimal accuracy decision $a_n \in [M_n, 1]$ to maximize its utility defined in Eqn. (5). Thus, for each user n,

$$\begin{aligned} \max \quad & u_n(a_n, \boldsymbol{a}_{-n}) \\ s.t. \quad & a_n \in [M_n, 1]. \end{aligned} \tag{6}$$

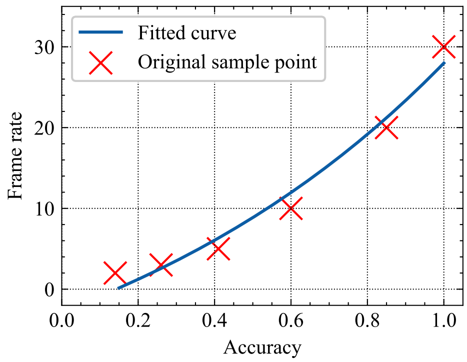


Fig. 3: Fitting the relationship between analytics accuracy and frame rate as a convex function

In the following two sections, we propose the GT-based and RL-based approaches to solve the formulated problem (6).

IV. GT-based Algorithm Design

In this section, we introduce the definition of Nash equilibrium, figure out the accuracy decision $a_n{}^*$ for each user n at the Nash equilibrium, and propose the GT-based algorithm to achieve the Nash equilibrium of the MEVAO problem.

A. Achieving Nash equilibrium

Definition 1. (Nash equilibrium): For each user n, the strategy set $\{a_1{}^*, a_2{}^*, ..., a_N{}^*\}$ constitutes a Nash equilibrium in the game of MEVAO problem if the individual utility cannot be improved by changing the accuracy strategy, i.e.,

$$u_n(a_n{}^*, \boldsymbol{a}_{-n}{}^*) \geq u_n(a_n, \boldsymbol{a}_{-n}{}^*). \tag{7}$$

Usually, the edge network resources are limited, and users interact with others to obtain a stable allocation of network resources and meet the video analytics service requirements. Game theory and Nash equilibrium can be used to analyze the interactions among multiple self-interested and independent users, and helps to design a decentralized system, where no users will change their accuracy decisions unilaterally.

1) Determining $a_n{}^*$ corresponding to $\boldsymbol{a}_{-n}{}^*$:

For convenience, we set $x_n = \frac{e^{a_n - s_n} - t_n}{r_n}$ and $\mathbb{O}_n = \sum_{i \neq n} \frac{e^{a_i - s_i} - t_i}{r_i}$. By Eqn. (5), we define the function

$$\begin{aligned} U_n(x_n) &= u_n(a_n, \boldsymbol{a}_{-n}) \\ &= -\frac{\alpha_n K x_n}{b_n} + \frac{\beta_n E x_n}{x_n + \mathbb{O}_n} + \gamma_n e^{s_n}(r_n x_n + t_n), \end{aligned} \tag{8}$$

and our objective is to maximize $U_n(x_n)$ for each user n. The first-order derivative of U_n with respect to x_n is

$$\frac{\partial U_n}{\partial x_n} = \frac{\beta_n E b_n \mathbb{O}_n}{(x_n + \mathbb{O}_n)^2} + (-\alpha_n K + \gamma_n e^{s_n} r_n b_n),$$

and the second-order derivative of U_n with respect to x_n is

$$\frac{\partial^2 U_n}{\partial x_n{}^2} = \frac{-2\beta_n E b_n{}^2 \mathbb{O}_n}{(x_n + \mathbb{O}_n)^3} < 0.$$

Thus, the function U_n is strictly concave in x_n. If

$$-\alpha_n K + \gamma_n e^{s_n} r_n b_n \geq 0, \tag{9}$$

U_n is monotone increasing. When (9) holds, the size of each video frame K is small enough. Thus, users have the incentives to offload more video data to the edge server, and then the optimal accuracy decision a_n is 1. Otherwise, i.e.,

$$-\alpha_n K + \gamma_n e^{s_n} r_n b_n < 0, \tag{10}$$

we set

$$\frac{\partial U_n}{\partial x_n} = \frac{\beta_n E b_n \mathbb{O}_n}{(x_n + \mathbb{O}_n)^2} + (-\alpha_n K + \gamma_n e^{s_n} r_n b_n) = 0. \tag{11}$$

By solving Eqn. (11), we obtain

$$x_n = \sqrt{\frac{\beta_n E b_n \mathbb{O}_n}{\alpha_n K - \gamma_n e^{s_n} r_n b_n}} - \mathbb{O}_n. \tag{12}$$

For each user n, if $a_n \in [M_n, 1]$, i.e.,

$$\frac{e^{M_n - s_n} - t_n}{r_n} \leq \sqrt{\frac{\beta_n E b_n \mathbb{O}_n}{\alpha_n K - \gamma_n e^{s_n} r_n b_n}} - \mathbb{O}_n \leq \frac{e^{1 - s_n} - t_n}{r_n}, \tag{13}$$

we have

$$x_n{}^* = \sqrt{\frac{\beta_n E b_n \mathbb{O}_n{}^*}{\alpha_n K - \gamma_n e^{s_n} r_n b_n}} - \mathbb{O}_n{}^*, \tag{14}$$

where $x_n{}^* = \frac{e^{a_n{}^* - s_n} - t_n}{r_n}$, and $\mathbb{O}_n{}^* = \sum_{i \neq n} \frac{e^{a_i{}^* - s_i} - t_i}{r_i}$.

According to Eqn. (14), we obtain the accuracy decision $a_n{}^*$ corresponding to other users' accuracy decisions $\boldsymbol{a}_{-n}{}^*$ when (13) holds. Next we figure out $a_n{}^*$.

2) Figuring out $a_n{}^*$:

By moving the terms in Eqn. (14) and squaring both sides of the equation, we obtain

$$\frac{\beta_n E b_n (\sum_{i=1}^{N} x_i{}^* - x_n{}^*)}{\alpha_n K - \gamma_n e^{s_n} r_n b_n} = (\sum_{i=1}^{N} x_i{}^*)^2. \tag{15}$$

Further moving the terms in Eqn. (15), we have

$$\sum_{i=1}^{N} x_i{}^* - x_n{}^* = \frac{\alpha_n K - \gamma_n e^{s_n} r_n b_n}{\beta_n E b_n} (\sum_{i=1}^{N} x_i{}^*)^2. \tag{16}$$

By adding $\sum_{n=1}^{N}$ to both sides of Eqn. (16), we get

$$(N-1)\sum_{i=1}^{N} x_i{}^* = \sum_{n=1}^{N} \mathbb{S}_n (\sum_{i=1}^{N} x_i{}^*)^2,$$

where $\mathbb{S}_n = \frac{\alpha_n K - \gamma_n e^{s_n} r_n b_n}{\beta_n E b_n}$, and it is a constant. Thus,

$$\sum_{i=1}^{N} x_i{}^* = \frac{N-1}{\sum_{n=1}^{N} \mathbb{S}_n}. \tag{17}$$

Plugging Eqn. (17) into Eqn. (16), we obtain

$$x_n{}^* = \frac{N-1}{\sum_{n=1}^{N} \mathbb{S}_n}(1 - \frac{\mathbb{S}_n(N-1)}{\sum_{n=1}^{N} \mathbb{S}_n}), \tag{18}$$

and since $x_n = \frac{e^{a_n - s_n} - t_n}{r_n}$, we have

$$a_n{}^* = \ln(\frac{r_n(N-1)}{\sum_{n=1}^{N} \mathbb{S}_n}(1 - \frac{\mathbb{S}_n(N-1)}{\sum_{n=1}^{N} \mathbb{S}_n}) + t_n) + s_n \tag{19}$$

Algorithm 1 GT-based Algorithm

```
for each user $n = 1, 2, 3, ..., N$ do
    Prepare user $n$'s information including $r_n$, $s_n$, $t_n$, $b_n$, $\alpha_n$, $\beta_n$, $\gamma_n$, $M_n$.
    Publish information to a specified shared storage area.
    repeat
        Gather other users' information except user $n$.
    until All of other users' information is collected.
    Calculate the optimal accuracy decision ${a_n}^*$ according to Eqn. (19).
end for
```

if (10) and (13) hold for each user n. From the analytical solution (19), we can intuitively observe that when the fitting parameters r_n, s_n and t_n in the relationship between analytics accuracy and frame rate is large enough, the transmission cost in (2) will get small. And consequently, users will choose a higher accuracy decision ${a_n}^*$ and offload more video data to the edge server. Note that the solution (19) does not apply to scenarios with a single user because (10) is not satisfied, and it is reasonable in the multi-user video offloading problem.

B. Game Theory-based Algorithm

Given the fact that each user's private information (e.g., r_n, s_n and t_n) satisfies (10) and (13), we propose Algorithm 1 to figure out the optimal accuracy decision ${a_n}^*$ for each user n at the Nash equilibrium.

As shown in Algorithm 1, each user n firstly publishes its private information. Usually, users send their information to a specified shared storage area. Then each user gathers all of the other users' information. With all the information collected, each user n finally calculates the optimal accuracy decision ${a_n}^*$ through Eqn. (19). It is worth mentioning that Algorithm 1 can also be applied to online situations. For example, when the users change their requirements on analytics accuracy, we only need to rerun Algorithm 1, and then obtain the new decision.

V. RL-based Algorithm Design

Although the optimal accuracy decision can be calculated through Algorithm 1, it is unrealistic for users in the real world to share their private information because of security and privacy concerns. The motivation of utilizing reinforcement learning is to improve the flexibility of making video analysis accuracy decisions under different circumstances, and finally obtain the nearly optimal accuracy decision after a long time running without knowing other users' information. In this section, we model the Markov decision process for the video analysis accuracy decision making. Based on the Markov decision process, we utilize the A2C model to design the RL-based decentralized Algorithm 2 for each user.

A. Modeling Markov Decision Process

The Markov decision process is represented as $\mathcal{M} = \langle \mathcal{A}, \mathcal{ST}, \mathcal{R}, \mathcal{P} \rangle$, which consists of an action space $\mathcal{A} = \{\mathcal{A}_n | n = 1, 2, ..., N\}$, a state space $\mathcal{ST} = \{\mathcal{ST}_n | n = 1, 2, ..., N\}$, a reward space $\mathcal{R} = \{\mathcal{R}_n : \mathcal{ST}_n \times \mathcal{A}_n \times \mathcal{ST}_n \to \mathbb{R}\}_{n \in \{1,...,N\}}$ and a state transition probability function set $\mathcal{P} = \{\mathcal{P}_n : \mathcal{ST}_n \times \mathcal{A}_n \times \mathcal{ST}_n \to [0, 1]\}_{n \in \{1,...,N\}}$.

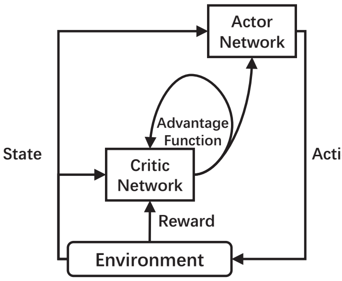


Fig. 4: Advantage actor critic network structure

Action space $\mathcal{A}$: The action space is denoted as $\mathcal{A} = \{\mathcal{A}_n | n = 1, 2, ..., N\}$, where $\mathcal{A}_n = \{a_n^k | k \in \mathbb{N}\}$. At time k, the user n makes the accuracy decision a_n^k. Similar to some existing works [23], [28], users can only acquire their own private information and the past strategy set $\{\boldsymbol{a}^{k-1}, \boldsymbol{a}^{k-2}, ..., \boldsymbol{a}^{k-B}\}$ in the Markov decision process, where B is the size of the past strategy set.

State space $\mathcal{ST}$: There is a state space $\mathcal{ST} = \{\mathcal{ST}_n | n = 1, 2, ..., N\}$, where $\mathcal{ST}_n = \{st_n^k | k \in \mathbb{N}\}$ and $st_n^k = [a_n^k, \boldsymbol{a}_{-n}^k, ..., a_n^{k-B}, \boldsymbol{a}_{-n}^{k-B}]$. The state of user n at time k is denoted by st_n^k, and it consists of the accuracy decisions made by all users at the current time and previous B time slots. $\mathcal{D}_n^0$ denotes the initial state probability distribution of user n.

Reward space $\mathcal{R}$: We have a reward space $\mathcal{R} = \{\mathcal{R}_n : \mathcal{ST}_n \times \mathcal{A}_n \times \mathcal{ST}_n \to \mathbb{R}\}_{n \in \{1,...,N\}}$, where $\mathcal{R}_n = \{r_n^k | k \in \mathbb{N}\}$ and the reward for user n is calculated according to its utility function. At time k, user n gets the reward $r_n^k = u_n(a_n^k, \boldsymbol{a}_{-n}^k)$.

State transition probability function set $\mathcal{P}$: The state transition probability function set is denoted by $\mathcal{P} = \{\mathcal{P}_n : \mathcal{ST}_n \times \mathcal{A}_n \times \mathcal{ST}_n \to [0, 1]\}_{n \in \{1,...,N\}}$, and $\mathcal{P}_n(st_n^{k+1} | st_n^k, a_n^k)$ is the probability of state st_n^k transiting to st_n^{k+1} through action a_n^k at time k.

Video analytics task offloading policy set Π: We have a video analytics task offloading policy set $\Pi = \{\pi_{\boldsymbol{\theta}_n}\}_{n \in \{1,...,N\}}$, where $\pi_{\boldsymbol{\theta}_n} : \mathcal{ST}_n \times \mathcal{A}_n \to [0, 1]$ is the video analytics task offloading policy for user n and it is parameterized by $\boldsymbol{\theta}_n$.

Thus, our objective is to obtain

$$\begin{aligned} {\boldsymbol{\theta}_n}^* &= \arg\max_{\boldsymbol{\theta}_n} \mathbb{E}[V^{\pi_{\boldsymbol{\theta}_n}}(st_n^0) | \mathcal{D}_n^0] \\ &= \arg\max_{\boldsymbol{\theta}_n} \mathbb{E}[Q^{\pi_{\boldsymbol{\theta}_n}}(st_n^0, a_n^0) | \mathcal{D}_n^0, \pi_{\boldsymbol{\theta}_n}], \end{aligned} \tag{20}$$

where the value function $V^{\pi_{\boldsymbol{\theta}_n}}(st_n^0)$ for observation and the value function $Q^{\pi_{\boldsymbol{\theta}_n}}(st_n^0, a_n^0)$ for observation and action are defined as

$$V^{\pi_{\boldsymbol{\theta}_n}}(st_n^0) = \mathbb{E}[\sum_{i=0}^{k} r_n^i | st_n^0, \mathcal{P}, \Pi], \tag{21}$$

$$Q^{\pi_{\boldsymbol{\theta}_n}}(st_n^0, a_n^0) = \mathbb{E}[\sum_{i=0}^{k} r_n^i | st_n^0, a_n^0, \mathcal{P}, \Pi]. \tag{22}$$

After modeling the MEVAO problem as a multi-agent Markov decision process, we apply the deep reinforcement learning approach A2C to optimize the video analytics task offloading policy for each user.

Algorithm 2 RL-based Algorithm

```
Initialize $\boldsymbol{\theta}_n$, $\boldsymbol{w}_n$, $\alpha^{\boldsymbol{\theta}_n}$, $\alpha^{\boldsymbol{w}_n}$ and $st_n^0$.
for time slot $k = 0, 1, 2, ...$ do
    for each user $n = 1, 2, 3, ..., N$ do
        Acquire the past strategy set.
        Update its state $st_n^k$ into $st_n^{k+1}$.
        Input $st_n^k$ into the Actor network $\pi_{\boldsymbol{\theta}_n}$.
        Obtain the accuracy decision $a_n^k$ from $\pi_{\boldsymbol{\theta}_n}$.
        Calculate reward $r_n^k = u_n(a_n^k, \boldsymbol{a}_{-n}^k)$ according to (5).
        Update $\boldsymbol{w}_n$ and $\boldsymbol{\theta}_n$ according to (24), (25).
    end for
end for
```

B. Reinforcement Learning-based Algorithm

The reinforcement learning approach of Advantage Actor Critic (A2C) is based on the Actor-Critic network which combines policy function $\pi_{\boldsymbol{\theta}_n}(a_n^k|st_n^k;\boldsymbol{\theta}_n)$ and value function $V^{\pi_{\boldsymbol{\theta}_n}}(st_n^k, \boldsymbol{w}_n)$, where $\boldsymbol{\theta}_n$ and $\boldsymbol{w}_n$ are weights in the Actor-Critic network. As shown in Fig. 4, in the A2C-based learning approach, each user acts as an agent to make the video analysis accuracy decision a_n^k in the state st_n^k according to the policy $\pi_{\boldsymbol{\theta}_n}(a_n^k|st_n^k;\boldsymbol{\theta}_n)$. The Critic network estimates the value of the actions; meanwhile the Actor network optimizes the policy with the value to maximize the future reward.

Specifically, when the Critic network estimates the value of the action, the weight $\boldsymbol{w}_n$ is updated as:

$$\delta = Q^{\pi_{\boldsymbol{\theta}_n}}(st_n^k, a_n^k) - V^{\pi_{\boldsymbol{\theta}_n}}(st_n^k), \tag{23}$$

$$\boldsymbol{w}_n = \boldsymbol{w}_n + \alpha^{\boldsymbol{w}_n}\delta\nabla V^{\pi_{\boldsymbol{\theta}_n}}(st_n^k), \tag{24}$$

where $\alpha^{\boldsymbol{w}_n}$ means the step size.

The Actor network ascends the gradients of policy $\pi_{\boldsymbol{\theta}_n}(a_n^k|st_n^k;\boldsymbol{\theta}_n)$ by updating the parameters based on the value from the Critic network. We calculate the gradient accumulation of parameter $\boldsymbol{\theta}_n$ as :

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_n + \alpha^{\boldsymbol{\theta}_n}\delta\nabla\ln\pi_{\boldsymbol{\theta}_n}(a_n^k|st_n^k;\boldsymbol{\theta}_n), \tag{25}$$

where $\alpha^{\boldsymbol{\theta}_n}$ is the step size.

Note that the advantage function in the learning scheme of Advantage Actor Critic is described as:

$$A^{\pi_{\boldsymbol{\theta}_n}}(st_n^k, a_n^k) = Q^{\pi_{\boldsymbol{\theta}_n}}(st_n^k, a_n^k) - V^{\pi_{\boldsymbol{\theta}_n}}(st_n^k). \tag{26}$$

Based on the above details of Advantage Actor Critic, we propose the RL-based algorithm for each user n. As shown in Algorithm 2, each user n firstly initializes the parameters in the Actor-Critic network and its state. At the beginning of each time slot, each user acquires the past strategy set $\{\boldsymbol{a}^{k-1}, \boldsymbol{a}^{k-2}, ..., \boldsymbol{a}^{k-B}\}$ and updates its state. Then user n inputs its state into the Actor network and gets the video analysis accuracy decision a_n^k. According to its utility function u_n, the reward r_n^k is calculated. After that, each user n optimizes its Actor-Critic network. The Critic network estimates the action value and updates $\boldsymbol{w}_n$ through (24). The Actor network ascends the gradients of policy $\pi_{\boldsymbol{\theta}_n}(a_n^k|st_n^k;\boldsymbol{\theta}_n)$ and updates $\boldsymbol{\theta}_n$ through (25).

VI. Performance Evaluation

In this section, we evaluate the performance of our algorithms through simulations with various settings. Besides, we compare our proposed designs with other existing approaches.

A. Simulation Settings

In the simulation, each user is allocated the network bandwidth $b_n \sim N(1, 0.1)Mb/s$, and has the video analytics task with various accuracy requirement $M_n \sim U(0, 1)$. We set the computation capacity at edge server as $E = 32Mb/s$, and there is $K = 0.1Mb$ video data caused by each unit of frame rate increase. We set the parameters α_n, β_n and γ_n in the utility function, which satisfies $\alpha_n + \beta_n + \gamma_n = 1$. Similar to the existing work [20], the fitting parameters r_n, s_n and t_n are selected to formulate the relationship between frame rate and accuracy as convex functions; meanwhile (8) and (11) are satisfied. We select the hyper-parameters in Advantage Actor Critic depending on the learning ability and performance. Specifically, the Critic network and the Actor network both have two fully-connected layers, each of which has 64 nodes.

Besides, we compare our work, which is referred to as MA2C, with 4 other baseline approaches:

- MPPO: This is a modified PPO [29], which is implemented for competitive multi-agent training.
- AccuracyPrior: Each user gives priority to the accuracy when making the decision in the task offloading game.
- LatencyPrior: Each user gives priority to the latency when making the decision in the task offloading game.
- Greedy: Each user makes the decision with the maximum reward for each time slot.

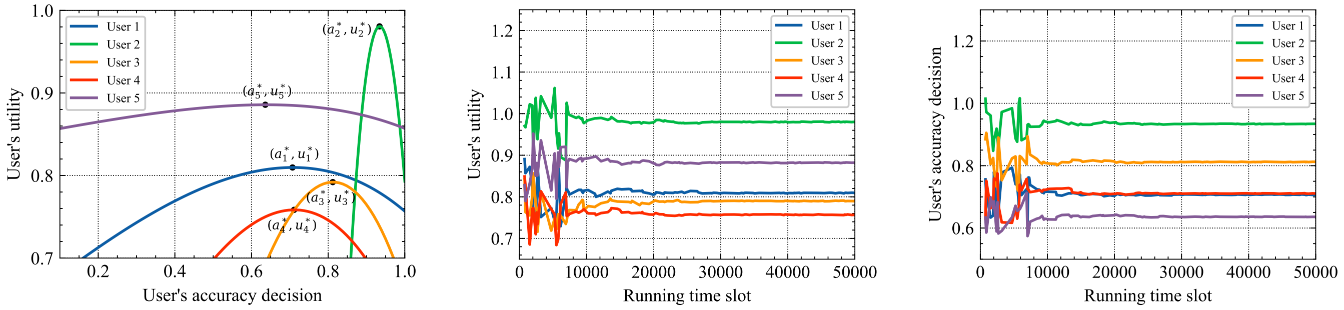
B. Results for GT-based Approach

We first study the performance of the proposed Algorithm 1 when the user number is 5, and we set $M_1 = 0.7$, $M_2 = 0.9$, $M_3 = 0.8$, $M_4 = 0.7$ and $M_5 = 0.6$. As shown in Fig. 5(a), the optimal accuracy decision ${a_n}^*$ and utility ${u_n}^*$ for each user are obtained with algorithm 1.

From Fig. 5(a), we observe that the optimal utilities for the users are about 0.81, 0.98, 0.19, 0.76 and 0.88, and the optimal accuracy decisions are about 0.71, 0.93, 0.81, 0.71 and 0.64. When $a_n \in [M_n, 1]$, we find that $u_n(a_n, \boldsymbol{a}_{-n})$ is concave and ${a_n}^*$ is locally optimal. The optimal accuracy decision ${a_n}^*$ and utility ${u_n}^*$ satisfy (7) in NE definition that each user has no incentives to raise or lower its accuracy decision, and the utility cannot be improved by changing the accuracy strategy.

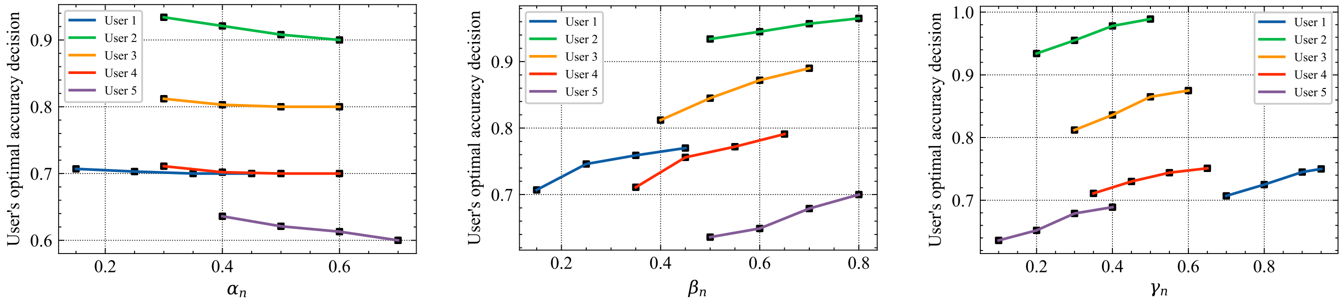
C. Results for RL-based Approach

We investigate the convergence of the proposed RL-based Algorithm 2. As shown in Fig. 5(b) and Fig. 5(c), each user's utility and accuracy decision converge at about 20000 time slots. From Fig. 5(b) and Fig. 5(c), we observe that each user's utility and accuracy decision obtained from algorithm 2 are consistent with the results shown in Fig. 5(a). Thus, we finally get the the nearly optimal utility and accuracy decision in the RL-based Algorithm 2.



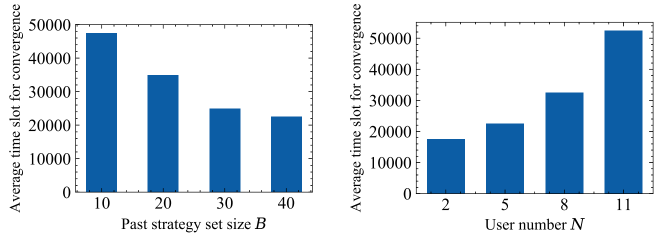
(a) Optimal accuracy decision and utility (b) Utility convergence (c) Accuracy convergence

Fig. 5: Simulation results of GT-based Algorithm and RL-based Algorithm



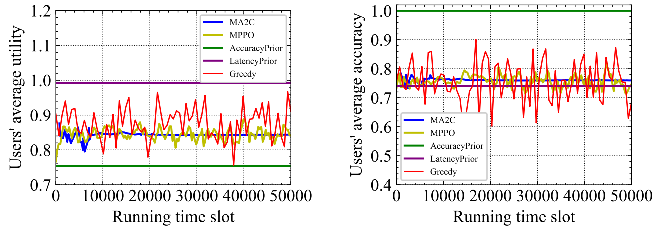
(a) Influnece of weight α_n (b) Influnece of weight β_n (c) Influnece of weight γ_n

Fig. 6: Influence of weight α_n, β_n and γ_n on user's optimal accuracy decision



(a) Varying the past strategy set size (b) Varying the user number

Fig. 7: Average time slots for convergence



(a) Users' average utility (b) Users' average accuracy

Fig. 8: Comparing the performance of 5 algorithms

We study the influence of weight α_n, β_n and γ_n. As demonstrated in Fig. 6, each user's optimal accuracy decision changes when we vary α_n, β_n and γ_n. From the definition of utility function (5), we observe that when we increase the weight of α_n, higher accuracy decision will result in higher transmission cost and reduce the utility. Thus, As Fig. 6(a) shows, users will choose lower accuracy decision when the weight α_n gets larger. Similarly, when we raise the weight of β_n and γ_n in (5), user n will obtain more computation allocation from the edge server and higher accuracy satisfaction. Thus, it is shown in Fig. 6(b) and Fig. 6(c) that when β_n and γ_n become larger, users will make higher accuracy decisions.

We vary the past strategy size B and the user number N in the simulation of video analytics task offloading game. As shown in Fig. 7(a), the number of time slots for convergence decreases when we enlarge the past strategy set. Since more information can be learned from the larger past strategy set, the RL-based algorithm can converge to the Nash equilibrium after fewer time slots. From Fig. 7(b), we observe that when the user number ranges from 2 to 11, it will take more time slots to reach the convergence of the RL-based algorithm. Competition for limited computing resources will be more intense and users will change their accuracy decisions more frequently when more users are involved in the task offloading scenario.

We compare our work with 4 other baseline approaches when the user number is 5. As shown in Fig. 8(a) and Fig. 8(b), our proposed algorithm converges at about 15000 time slots, and we can obtain both the average of users' optimal utilities and the average of users' optimal accuracy decisions. However, it is hard for the MPPO algorithm to converge within 50000 time slots, and the average utility obtained from the MPPO algorithm fluctuates when the users change their accuracy decisions. Meanwhile, we apply the AccuracyPrior and LatencyPrior algorithm to our problem. When the users give priority to the accuracy, they will choose the highest accuracy decision (i.e., 100%), and the average utility is about 0.99. Similarly, when the users give priority to the latency, they prefer to offload less video data to the edge sever, which will result in the lowest accuracy decision (i.e., M_n), and the average utility is about 0.76. Besides, we design a Greedy algorithm, where users make the decisions with the maximum utility for each time slot. From Fig. 8(a) and Fig. 8(b), we observe that it is difficult for users to get steady rewards, and they have the incentives to change their accuracy decisions. Thus, our design has a better performance than others.

VII. Conclusion and Future Work

We study the multi-user edge-assisted video offloading and analyzing problem in this paper. All users independently choose their accuracy decisions satisfying the accuracy requirement and offload the video data to the edge server. With the utility function designed based on the video analytics features, we achieve the Nash equilibrium and the optimal video analytics accuracy. To improve the flexibility of making decisions under different circumstances, we propose the RL-based algorithm to tackle the MEVAO problem without information sharing. Based on the A2C model, users adjust their accuracy decisions and finally achieve the converged reward.

However, there are a few limitations in our work that demand future research effort. Firstly, we consider the MEVAO problem in the special case where Nash equilibrium can be obtained, and it can be our future work to extend the problem to the general case. Secondly, video analytics tasks can be offloaded to multiple edge servers or the edge server cluster in the real world, and jointly considering the problem of video analytics task offloading and resource allocation within the edge cluster will be challenging. Finally, in the proposed RL-based algorithm, each user updates its state based on the past strategy set, which consists of all users' strategies in the past B time slots, and it is meaningful to set the appropriate size of the past strategy set since it has an influence on the performance of the RL-based algorithm.

Acknowledgments

This work was supported in part by National Key R&D Program of China (2017YFB1001801), NSFC (61872175, 61832008), Natural Science Foundation of Jiangsu Province (BK20181252), Jiangsu Key R&D Program (BE2018116), and Collaborative Innovation Center of Novel Software Technology and Industrialization. Sheng Zhang is the corresponding author.

References

[1] C. Wang, S. Zhang, Z. Qian, M. Xiao, J. Wu, B. Ye, and S. Lu, "Joint server assignment and resource management for edge-based mar system," *IEEE/ACM Transactions on Networking*, 2020.

[2] S. Zhang, Y. Liang, J. Ge, M. Xiao, and J. Wu, "Provably efficient resource allocation for edge service entities using hermes," *IEEE/ACM Transactions on Networking*, 2020.

[3] M. Caprolu, R. Di Pietro, F. Lombardi, and S. Raponi, "Edge computing perspectives: architectures, technologies, and open security issues," in *2019 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2019, pp. 116–123.

[4] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[5] K. Hsieh, G. Ananthanarayanan, P. Bodik, S. Venkataraman, P. Bahl, M. Philipose, P. B. Gibbons, and O. Mutlu, "Focus: Querying large video datasets with low latency and low cost," in *2018 USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. USENIX, 2018, pp. 269–286.

[6] Y. Li, Z. Han, Q. Zhang, Z. Li, and H. Tan, "Automating cloud deployment for deep learning inference of real-time online services," in *2020 IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2020, pp. 1668–1677.

[7] C. Wang, S. Zhang, Y. Chen, Z. Qian, J. Wu, and M. Xiao, "Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics," in *2020 IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2020, pp. 1–10.

[8] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*. ACM, 2018, pp. 253–266.

[9] P. K. Dutta and P. K. Dutta, *Strategies and games: theory and practice*. MIT press, 1999.

[10] R. Gibbons, *A primer in game theory*. Harvester Wheatsheaf New York, 1992.

[11] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 260–288, 2018.

[12] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, "Coalitional game theory for communication networks," *Ieee signal processing magazine*, vol. 26, no. 5, pp. 77–97, 2009.

[13] D. Yang, X. Fang, and G. Xue, "Game theory in cooperative communications," *IEEE Wireless Communications*, vol. 19, no. 2, pp. 44–49, 2012.

[14] W. Wang, A. Kwasinski, D. Niyato, and Z. Han, "A survey on applications of model-free strategy learning in cognitive wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1717–1757, 2016.

[15] M. S. Abdalzaher, K. Seddik, M. Elsabrouty, O. Muta, H. Furukawa, and A. Abdel-Rahman, "Game theory meets wireless sensor networks security requirements and threats mitigation: A survey," *Sensors*, vol. 16, no. 7, p. 1003, 2016.

[16] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728–1739, 2016.

[17] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa, "Sdmec: Software defined system for mobile edge computing," in *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*. IEEE, 2016, pp. 88–93.

[18] N. Kumar, S. Zeadally, and J. J. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 60–66, 2016.

[19] A. Anjum, T. Abdullah, M. Tariq, Y. Baltaci, and N. Antonopoulos, "Video stream analysis in clouds: An object detection and classification framework for high performance video analytics," *IEEE Transactions on Cloud Computing*, 2016.

[20] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5506–5519, 2018.

[21] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia, "Noscope: optimizing neural network queries over video at scale," *arXiv preprint arXiv:1703.02529*, 2017.

[22] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[23] Y. Zhan, S. Guo, P. Li, and J. Zhang, "A deep reinforcement learning based offloading game in edge computing," *IEEE Transactions on Computers*, vol. 69, no. 6, pp. 883–893, 2020.

[24] M. Hu, Z. Xie, D. Wu, Y. Zhou, X. Chen, and L. Xiao, "Heterogeneous edge offloading with incomplete information: A minority game approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 9, pp. 2139–2154, 2020.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[26] L. Xiao, Y. Li, X. Huang, and X. Du, "Cloud-based malware detection game for mobile devices with offloading," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2742–2750, 2017.

[27] X. Wan, G. Sheng, Y. Li, L. Xiao, and X. Du, "Reinforcement learning based mobile offloading for cloud-based malware detection," in *2017 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2017, pp. 1–6.

[28] J. Zou, T. Hao, C. Yu, and H. Jin, "A3C-DO: A regional resource scheduling framework based on deep reinforcement learning in edge scenario," *IEEE Transactions on Computers*, 2020.

[29] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch, "Emergent complexity via multi-agent competition," *arXiv preprint arXiv:1710.03748*, 2017.