Green Collaborative Inference in RIS-Assisted MEC Networks under Computing Backlog Constraints

Yang Yang and M. Cenk Gursoy
Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244
Email: yyang82@syr.edu, mcgursoy@syr.edu

***Abstract*—In this paper, we analyze collaborative inference in a mobile edge computing (MEC) network aided by a reconfigurable intelligent surface (RIS). In particular, we consider multiple user equipments (UEs) with collaborative inference tasks. The goal is to minimize the long-term average energy consumption subject to a long-term average computing queue backlog constraint. We first transform the considered problem into a Lyapunov optimization problem and then propose a deep reinforcement learning (DRL)-based algorithm to solve it. An optimization subroutine is embedded in the proposed algorithm to directly obtain the optimal RIS coefficients, while the UEs' deep neural network (DNN) partition decisions and computational resource allocations at the MEC server are obtained from the DRL-based algorithm. Numerical results show that the proposed algorithm solves the problem efficiently, and the introduced RIS improves the long-term average energy consumption significantly. Furthermore, the impact of various parameters (e.g., bandwidth and the maximum CPU frequency at the MEC server) is analyzed.**

***Index Terms*—Energy consumption, edge computing, collaborative inference, reconfigurable intelligent surface (RIS), deep reinforcement learning.**

I. Introduction

In recent years, an increased demand for high data rates and advanced computing capabilities (e.g., due to heightened computational requirements in mobile data services and applications) has resulted in congestion challenges within cellular networks and data servers [1]. Addressing these challenges, mobile edge computing (MEC) has emerged as a promising solution, allowing user equipments (UEs) to fully or partially offload their tasks to nearby edge servers instead of relying on remote data servers [2].

Along with advances in edge computing, the progress in deep learning (DL) has enabled the management and processing of more intelligent tasks. However, the execution of complex deep neural networks (DNN) for inference applications on UEs consumes a substantial amount of energy, particularly as DL models grow in complexity and size.

As both UEs and the MEC server possess the capability to execute learning and perception tasks (such as classification, recognition, reasoning, etc.) to varying extents, collaborative inference in MEC networks has recently attracted interest as a strategy to diminish energy consumption while adhering to constrained inference latency [3]. With such collaboration, UEs can handle more machine learning applications that involve substantial computational demands through the adoption of collaborative inference with the MEC server. For instance, in [4], the authors have introduced a method to partition a DNN task into multiple sub-tasks. These sub-tasks can be processed locally by the UEs or offloaded to one or more powerful edge nodes or servers, such as those in fog networking.

The cooperation between devices in the industrial internet of things (IoT) and edge networks is of paramount importance for supporting computation-intensive DNN-based inference tasks that demand low latency and high accuracy. In the study in [5], the authors explored the collaborative inference challenge in industrial IoT networks and formulated the problem as a constrained Markov decision process (CMDP). This formulation has jointly considered sampling rate adaptation, inference task offloading, and edge computing resource allocation, with the aim of minimizing the average service delay for various inference services.

As yet another technological advance, reconfigurable intelligent surfaces (RISs) have recently been extensively studied and shown to significantly enhance both the propagation environment and spectral efficiency [6]. In particular, the proper design of the phase shift matrix at the RIS leads to substantial improvements on the wireless propagation environment and performance [7].

Motivated by the aforementioned challenges in effective management of resources under latency requirements, in this paper we consider an RIS-assisted MEC network aiming to minimize the long-term average energy consumption in collaborative inference under a long-term average computing queue backlog constraint. Unlike our previous work in [8], we have further considered the latency requirements by analyzing the long-term computing queue backlog at the MEC server. Such a long-term consideration makes the problem more challenging in pursuing an optimal solution. Below are the primary contributions of this paper:

1) In a multi-UE MEC network, we define and evaluate the computing queue backlog at the MEC server.
2) We analyze the minimization of the long-term average energy consumption with constrained long-term average computing queue backlog.
3) We construct a DRL-based algorithm combined with an embedded optimization subroutine (for RIS coefficients) to solve the formulated optimization problem.

The remainder of this paper is organized as follows. We first describe the collaborative inference model and the RIS-aided transmission model, and then we specify the computing queue backlog at the MEC server as well as the energy consumption model in Section II. In Section III, we state the optimization problem under a long-term average computing queue backlog constraint and then provide a DRL-based algorithm to address it, within which an embedded subroutine for RIS coefficient optimization is introduced and investigated. Simulation results are provided in Section IV. Finally, in Section V, we draw conclusions.

II. System Model

An RIS-aided MEC network where the BS is equipped with an MEC server is considered in this paper. Each of the M UEs in the network has a single antenna. The BS has N antennas and the RIS has K reflecting components. A wireless controller is used by the BS to operate the RIS so that it is capable of dynamically adjusting the RIS phase shift matrix (i.e., the phase shift of each reflecting element).

Fig. 1 illustrates the considered network, where each UE in the UE set $\mathcal{M} = \{1, 2, ..., M\}$ needs to complete computation-intensive DNN inference tasks for e.g., classification, recognition or reasoning applications. The UE can partially offload the DNN tasks to the MEC server via wireless links. In this paper, the same pre-trainted DNN will be processed among all

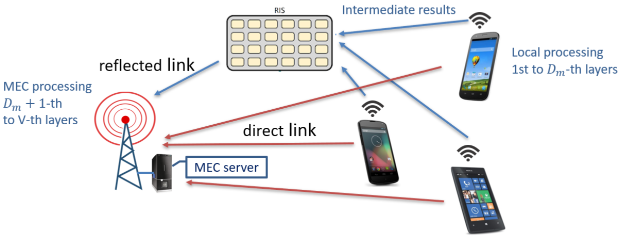


Fig. 1: An illustration of the considered MEC network.

UEs and MEC server. Similar to [4], we use a directed acyclic graph (DAG) $\mathcal{G}$ to represent the DNN, consisting of V layers and $\mathcal{L}$ links. Each link refers to the order of processing and the input-output relationship between two connected layers.

A. Collaborative Inference Model

For the m-th UE, let U_m denote the number of new perception input data (e.g., images in classification) to the DNN. The pre-trained DNN is divided into two portions by a partition decision D_m. The m-th UE processes the first D_m layers locally, while the MEC server processes the remaining $V - D_m$ layers. The total CPU cycles needed for processing the pre-trained DNN is fixed for the majority of inference tasks, such as object identification or image classification, and in this paper it is represented by C cycles. Moreover, $(1-\alpha(D_m))C$ is the amount of CPU cycles needed to process the previous D_m layers, where $\alpha(D_m) \in [0,1]$ is the fraction of CPU cycles needed to complete the DNN after the D_m-th layer over the total required CPU cycles.

In this paper, we use $\boldsymbol{\mathcal{I}}_m$ to represent the intermediate results from the output of layer D_m. Such intermediate results are transmitted to the MEC server to be considered as the input for the remaining $V - D_m$ layers. When the MEC server receives the offloaded data from UE m, it then processes the remaining $V - D_m$ layers and produces the final inference results based on the output of layer V.[1]

B. Transmission Model with RIS

Each channel between the UE and the BS, as well as the RIS is assumed to experience block fading, and thus the channels remain constant throughout a transmission block. In this paper, the length of each frame is assumed to be τ seconds.

As a result of the RIS deployment, the channel from the UE to the BS now consists of two links: the direct link and the two-hop link (UE to RIS to BS), as shown in Fig. 1. Therefore, the channel fading vector from the m-th UE to the BS can be expressed as [9]

$$\boldsymbol{h}_m = \boldsymbol{h}_{d,m}^H + \boldsymbol{h}_{r,m}^H \Theta \boldsymbol{\mathcal{G}}, \tag{1}$$

where the channel vector from the UE to the BS is $\boldsymbol{h}_{d,m}^H \in \mathbb{C}^{1\times N}$, the channel vector from the UE to the RIS is $\boldsymbol{h}_{r,m}^H \in \mathbb{C}^{1\times K}$, and the channel matrix from the RIS to the BS is $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{K\times N}$. The phase shift matrix of the RIS, denoted by Θ, is defined as $\Theta = \beta \mathrm{diag}(e^{i\theta_1}, ..., e^{i\theta_K}) \in \mathbb{C}^{K\times K}$ where $\theta_k \in [0, 2\pi]$, $k \in \{1, 2, ..., K\}$ and the amplitude reflection coefficient $\beta \in [0,1]$ is set to be 1 in this article. Frequency-division multiple access (FDMA) is adopted in the offloading transmissions. As a consequence, the received signal at the MEC server from the m-th UE can be written as

$$\mathbf{y}_m = \boldsymbol{h}_m x_m + \boldsymbol{n}, \forall m \in \mathcal{M} \tag{2}$$

where x_m is the signal transmitted from the m-th UE, with an average power of $\mathbb{E}[|x_m|^2] = P_m$, and $\boldsymbol{n}$ is the additive white Gaussian noise (AWGN) at the MEC server, e.g., $\boldsymbol{n} \sim \mathcal{CN}(0, \sigma^2 \mathrm{I}_N)$.

Therefore, the SNR γ_m can be expressed as

$$\gamma_m = \rho_m ||\boldsymbol{h}_{d,m}^H + \boldsymbol{h}_{r,m}^H \Theta \boldsymbol{\mathcal{G}}||^2, \tag{3}$$

where $\rho_m = \frac{P_m}{\sigma^2}$.

C. Computing Queue Backlog at the MEC Server

The m-th UE requires $(1 - \alpha(D_m))C$ CPU cycles to complete the processing of the first D_m layers in the DNN. Hence, we can compute the local processing time/latency as

$$T_{L,m} = \frac{U_m(1-\alpha(D_m))C}{f_m}, \tag{4}$$

where f_m is the local CPU frequency (in cycles per second) of UE m. Furthermore, the transmission data rate of UE m to the MEC server in offloading can be formulated as

$$R_m = \mathcal{B} \log_2 (1 + \gamma_m), \tag{5}$$

where $\mathcal{B}$ is the bandwidth. Consequently, the offloading time/latency of UE m is expressed as

$$T_{R,m} = \frac{U_m I(D_m)}{R_m}. \tag{6}$$

where $I(D_m)$ is the size (in bits) of the intermediate results $\boldsymbol{\mathcal{I}}_m$. In this paper, we use B^t to denote the MEC computing queue backlog (in bits) in the t-th frame. Thereby, we can define the queue backlog at the MEC server for the m-th UE/task as

$$B_m^t = \left[\left[B_m^{t-1} - \frac{(T_{L,m}^t + T_{R,m}^t) F_m^t}{X_m} \right]^+ + \frac{\alpha(D_m^t)C - (\tau - T_{L,m}^t - T_{R,m}^t) F_m^t}{X_m} \right]^+ \tag{7}$$

where $[x]^+ = \max\{x, 0\}$, $B_m^0 = 0, \forall m \in \mathcal{M}$. Moreover, X_m and F_m^t are the task computation intensity of UE m and the allocated CPU frequency for the m-th UE at the MEC server in the t-th frame, respectively. Note that the first two terms on the right-hand side of (7) represents the possible remaining computing backlog from the $t-1$-th frame before the MEC server starts to process in the t-th frame, and the third term (i.e., $\frac{\alpha(D_m^t)C-(\tau-T_{L,m}^t-T_{R,m}^t)F_m^t}{X_m}$) is the possible computing backlog after processing the DNN tasks of the t-th frame.

D. Energy Consumption of UEs

In this paper, the energy consumption of each UE consists of two parts, i.e., local processing energy consumption $E_{L,m}$ and offloading energy consumption $E_{R,m}$.

According to [2], the local CPU frequency f_m and the local processing time $T_{L,m}$ of the m-th UE determine the energy consumption for local computing, which is expressed as follows:

$$E_{L,m} = T_{L,m} \Gamma_m f_m^3, \tag{8}$$

where Γ_m, which varies depending on the processor's architecture, is the m-th UE's effective capacitance coefficient.

For the m-th UE, $E_{R,m}$ is evaluated as the product of the offloading transmission power P_m and the offloading time $T_{R,m}$ of the UE:

$$E_{R,m} = P_m T_{R,m}. \tag{9}$$

Subsequently, the total power consumption at UE m is computed as

$$E_m = E_{L,m} + E_{R,m}. \tag{10}$$

[1] The time in downloading from the BS/edge server is negligible compared to the time needed for offloading and computation due to the small sizes of the final inference results.

III. Minimization of the Long-term Average Energy Consumption

In this section, we first formulate and analyze the global optimization problem and subsequently propose a deep RL-based algorithm to tackle the problem.

A. Problem Formulation

Our goal is to minimize the long-term average total energy consumption of all UEs under long-term average computing queue backlog constraint by jointly determining the RIS reflecting coefficients $\boldsymbol{\theta}$, the UEs' partition decisions $\{D_m\}$, and the CPU frequency allocations $\{F_m\}$ to the tasks among all UEs at the MEC server. Consequently, our global optimization problem is the following:

$$\textbf{P1:} \quad \underset{\{D_m, F_m, \boldsymbol{\theta}\}}{\textbf{Minimize}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} E_m^t \tag{11}$$

$$\textbf{s. t.} \quad D_m \in \{1, 2, ..., V\}, \forall m \in \mathcal{M} \tag{11a}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} B_m^t \leq \mathrm{B}_{\mathrm{th}}, \forall m \in \mathcal{M} \tag{11b}$$

$$B_m \leq 2\mathrm{B}_{\mathrm{th}}, \forall m \in \mathcal{M} \tag{11c}$$

$$\sum_{m=1}^{M} F_m \leq F_{\max}, \tag{11d}$$

where $F_{\max}$ is the total available CPU frequency at the MEC server. (11a) represents the feasible range of partition decisions. (11b) and (11c) indicate the long-term average computing queue backlog constraint and the maximum instantaneous computing queue backlog limitation, respectively. (11d) provides the MEC total CPU frequency constraint. Note that offloading of the input raw data directly is not allowed since we want to preserve data privacy, i.e., $D_m \geq 1$.

Due to the non-convex long-term constraint and the highly correlated optimization variables $\{D_m\}, \{F_m\}, \boldsymbol{\theta}$, the determination of the globally optimal solution of the non-convex problem **P1** is extremely challenging. In this paper, we construct a deep RL-based algorithm to solve **P1**. In particular, we obtain the optimal $\{D_m\}, \{F_m\}$ via deep RL, and we determine the optimal RIS coefficients $\boldsymbol{\theta}$ via an optimization subroutine which is embedded in the proposed deep RL learning algorithm.

B. Problem Transformation

Addressing the long-term constraint is one of the main challenges in solving **P1**. To overcome this bottleneck, we employ the Lyapunov approach [10], [11] to transform **P1**. The key technique is to introduce buffer deficit queues to describe the status of the long-term computing queue backlog constraint, and these assist the learning agent to achieve the long-term average computing queue backlog constraint.

For each UE, we first develop computing deficit queues that is updated as follows:

$$Z_m^{t+1} = \left[B_m^t - \mathrm{B}_{\mathrm{th}} + Z_m^t\right]^+, \forall m \in \mathcal{M} \tag{12}$$

where $Z_m^0 = 0$ and B_{th} (as also defined above) is the computing queue backlog limit at the MEC server.

In (12), Z_m^t represents the difference between the attained current computing queue backlog and the required long-term size limitation. Subsequently, according to [5], we construct a Lyapunov function $L(Z_m^t) = \frac{(Z_m^t)^2}{2}$ to describe the degree of contentment in the long-term computing queue backlog constraint. Note that the long-term computing queue backlog limitation is well satisfied when $L(Z_m^t)$ is small.

Moreover, we need to make sure the introduced Lyapunov function $L(Z_m^t)$ will keep a low value constantly, leading to a satisfied long-term computing queue backlog constraint. Similar as in [20], we analyze the one-shot Lyapunov drift to investigate the deviation of the Lyapunov function $L(Z_m^t)$ during two consecutive frames. Such one-shot Lyapunov drift $\Delta(Z_m^t)$ is characterized as follows:

$$\begin{aligned}\Delta(Z_m^t) &= L(Z_m^{t+1}) - L(Z_m^t) \\ &= \frac{(Z_m^{t+1})^2}{2} - \frac{(Z_m^t)^2}{2} \\ &\overset{(a)}{\leq} \frac{1}{2}[(B_m^t - \mathrm{B}_{\mathrm{th}} + Z_m^t)^2 - (Z_m^t)^2] \\ &= \frac{1}{2}(B_m^t - \mathrm{B}_{\mathrm{th}})^2 + Z_m^t(B_m^t - \mathrm{B}_{\mathrm{th}}) \\ &\overset{(b)}{\leq} \frac{(\mathrm{B}_{\mathrm{th}})^2}{2} + Z_m^t(B_m^t - \mathrm{B}_{\mathrm{th}}).\end{aligned} \tag{13}$$

Inequality in (a) above is due to the fact that $[x]^+ \leq |x|$, and hence $([x]^+)^2 \leq x^2$. Inequality in (b) exists due to $\frac{(\mathrm{B}_{\mathrm{th}})^2}{2} \geq \frac{(\mathrm{B}_{\mathrm{th}} - B_m^t)^2}{2} = \frac{(B_m^t - \mathrm{B}_{\mathrm{th}})^2}{2}$ since $0 \leq B_m^t \leq 2\mathrm{B}_{\mathrm{th}}$. Therefore, by employing an auxiliary parameter W, we can construct a one-shot drift-plus-energy function during the t-th frame, and we have

$$\begin{aligned}\sum_{m=1}^{M} W\cdot \Delta(Z_m^t) + \sum_{m=1}^{M} E_m^t \leq \sum_{m=1}^{M} \frac{(W\cdot \mathrm{B}_{\mathrm{th}})^2}{2} + \\ \sum_{m=1}^{M} [W\cdot Z_m^t(B_m^t - \mathrm{B}_{\mathrm{th}}) + E_m^t].\end{aligned} \tag{14}$$

Accordingly, utilizing the Lyapunov optimization theory, we can transform the original optimization problem **P1** aiming to minimize the long-term average total energy consumption while satisfying the long-term average computing queue backlog constraints into a long-term average drift-plus-energy minimization problem **P2** as follows:

$$\textbf{P2:} \underset{\{D_m, F_m, \boldsymbol{\theta}\}}{\textbf{Minimize}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} \left[W\cdot Z_m^t(B_m^t - \mathrm{B}_{\mathrm{th}}) + E_m^t\right] \tag{15}$$

$$\textbf{s. t.} \quad D_m \in \{1, 2, ..., V\}, \forall m \in \mathcal{M} \tag{15a}$$

$$B_m \leq 2\mathrm{B}_{\mathrm{th}}, \forall m \in \mathcal{M} \tag{15b}$$

$$\sum_{m=1}^{M} F_m \leq F_{\max}, \tag{15c}$$

where the auxiliary parameter W is a positive parameter to balance the weight between the energy consumption minimization and the contentment degree of the long-term average computing queue backlog constraint. This adjustment can be explained as follows. If any violation occurs in the long-term average computing queue backlog constraint, i.e., $Z_m^t > 0$, we may decrease the value of W so that achieving a small computing queue backlog becomes relatively more important than improving the energy consumption.

C. DRL-based Algorithm

By solving **P2**, we not only minimize the global energy consumption, but also guarantee the long-term computing queue backlog requirements. In this section, we introduce and describe the main structure of DRL-based algorithm to address **P2**. Specifically, its action space, state space and reward function are defined as follows:

1) Action Space: The action space in our proposed DRL-based algorithm consists of the UEs' partition decisions $\boldsymbol{D} = \{D_1, D_2, ..., D_m, ...D_M\}, m \in \mathcal{M}$ and the CPU frequency allocations $\boldsymbol{F} = \{F_1, F_2, ..., F_m, ...F_M\}, m \in \mathcal{M}$ at the MEC server, i.e.,

$$a^t = \{\boldsymbol{D}^t, \boldsymbol{F}^t\}. \tag{16}$$

Note that following constraints should be guaranteed in all the selected actions: (i) $D_m \in \{1, 2, ..., V\}\, \forall m \in \mathcal{M}$, which constrains each UEs' partition decision to an integer set; (ii)

$\sum_{m=1}^{M} F_m \leq F_{\max}$ and $0 \leq F_m \leq F_{\max}$ $\forall m \in \mathcal{M}$, which are imposed on MEC CPU frequency allocations.

2) State Space: The state space in this DRL consists of the numbers of UEs' new perception data $\boldsymbol{U} = \{U_1, U_2, ..., U_m, ...U_M\}$, channel states of all UEs $\boldsymbol{H} = \{h_1, h_2, ..., h_m, ...h_M\}$, computing queue backlogs at the MEC server $\boldsymbol{B} = \{B_1, B_2, ..., B_m, ...B_M\}$, i.e.,

$$s^t = \{\boldsymbol{U}^t, \boldsymbol{H}^t, \boldsymbol{B}^{t-1}\}. \tag{17}$$

Note that the computing queue backlogs are obtained from the $t-1$-th frame.

3) Reward Function: We construct the reward function with the objective to minimize the drift-plus-energy in any given frame t, i.e.,

$$r^t = \mathcal{R} - \mathcal{V} \cdot \sum_{m=1}^{M} [W \cdot Z_m^t (B_m^t - \mathrm{B}_{\mathrm{th}}) + E_m^t], \tag{18}$$

where $\mathcal{R}$ and $\mathcal{V}$ are constants to balance the reward. With the above state, action and reward definitions, we propose a DRL-based algorithm, which is extended from the deep deterministic policy gradient (DDPG) framework proposed in [12]. In this section, we aim to solve **P2** without considering the optimization of RIS coefficients, which will be addressed via an embedded optimization subroutine introduced in the next subsection. Such proposed DRL-based algorithm can be deployed at the MEC server since it can collect all the required information about the channel states and apply the policy to all served UEs.

D. Subroutine for the Optimization of RIS Coefficients

In the previous subsection, we introduced our proposed DRL-based algorithm without considering the optimization of RIS coefficients. In this subsection, we investigate the optimization of RIS coefficients in each frame.

1) Relationship between B_m^t and $T_{R,m}^t$: During the t-th frame, **P2** is transformed into **P3** when D_m, F_m are fixed:

$$\textbf{P3: } \underset{\boldsymbol{\theta}}{\textbf{Minimize}} \sum_{m=1}^{M} [W \cdot Z_m^t (B_m^t - \mathrm{B}_{\mathrm{th}}) + E_m^t] \tag{19}$$

$$\textbf{s. t.} \quad B_m^t \leq 2\mathrm{B}_{\mathrm{th}}, \forall m \in \mathcal{M}. \tag{19a}$$

In **P3**, from (12) we observe that Z_m^t is fixed in the t-th frame since it is determined by Z_m^{t-1} and B_m^{t-1}, and thereby minimizing B_m^t and E_m^t is equivalent to minimizing the objective in **P3**. Note that by properly adjusting the RIS coefficients $\boldsymbol{\theta}$, the channel fading vectors of all UEs will change correspondingly, and hence the SNRs will be enhanced, resulting in an improvement in the transmission data rate as well as the offloading time $\{T_{R,m}^t\}$.

Now we explore the inherent property of B_m^t. We first define a value of $\mathrm{T}_{R,m}^{t,\mathrm{eq}}$ so that $B_m^{t-1} = \frac{(T_{L,m}^t + \mathrm{T}_{R,m}^{t,\mathrm{eq}})F_m^t}{X_m}$. Next, we consider two different scenarios of $T_{R,m}^t$:

$$\begin{cases} \text{when } T_{R,m}^t \leq \mathrm{T}_{R,m}^{t,\mathrm{eq}}: B_m^{t-1} \geq \frac{(T_{L,m}^t + T_{R,m}^t)F_m^t}{X_m}, \text{we have } B_m^t = B_m^{t,A}; \\ \text{when } T_{R,m}^t > \mathrm{T}_{R,m}^{t,\mathrm{eq}}: B_m^{t-1} < \frac{(T_{L,m}^t + T_{R,m}^t)F_m^t}{X_m}, \text{we have } B_m^t = B_m^{t,B}, \end{cases}$$

where $B_m^{t,A} = \left[B_m^{t-1} + \frac{\alpha(D_m^t)C - \tau F_m^t}{X_m}\right]^+$ and $B_m^{t,B} = \left[\frac{\alpha(D_m^t)C - (\tau - T_{L,m}^t - T_{R,m}^t)F_m^t}{X_m}\right]^+$.

Defining $Y_m^t = B_m^{t-1} + \frac{\alpha(D_m^t)C - \tau F_m^t}{X_m}$, we have the following case:

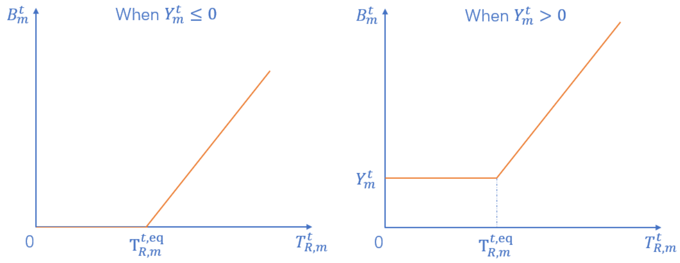
$$\text{when } T_{R,m}^t \geq \mathrm{T}_{R,m}^{t,\mathrm{eq}}: Y_m^t \overset{(c)}{\leq} \frac{(T_{L,m}^t + T_{R,m}^t)F_m^t}{X_m} + \frac{\alpha(D_m^t)C - \tau F_m^t}{X_m} \tag{20}$$

$$= \frac{\alpha(D_m^t)C - (\tau - T_{L,m}^t - T_{R,m}^t)F_m^t}{X_m}.$$

Above, inequality (c) holds because when $T_{R,m}^t \geq \mathrm{T}_{R,m}^{t,\mathrm{eq}}$, we have $B_m^{t-1} \leq \frac{(T_{L,m}^t + T_{R,m}^t)F_m^t}{X_m}$, and adding $\frac{\alpha(D_m^t)C - \tau F_m^t}{X_m}$ to both sides of this inequality leads to inequality (c). We subsequently have the following conclusions:

$$\begin{cases} \text{when } Y_m^t \leq 0 : B_m^{t,A} = 0, \text{and hence } B_m^{t,A} \leq B_m^{t,B}; \\ \text{when } Y_m^t \geq 0 : B_m^{t,A} \leq B_m^{t,B}, \text{due to the inequality (20).} \end{cases}$$

Therefore, we conclude that regardless of the value of Y_m^t, we always have $B_m^{t,A} \leq B_m^{t,B}$. Furthermore, considering $B_m^{t,B}$, increasing $T_{R,m}^t$ leads to a linear increase in $B_m^{t,B}$. Consequently, we have the following figures to demonstrate the characteristics of B_m^t versus $T_{R,m}^t$ in both cases. Note that $B_m^{t,A}$ is a constant that is equal to $[Y_m^t]^+$ when D_m^t and F_m^t are fixed. From Fig. 2a and Fig. 2b, we observe that by decreasing $T_{R,m}^t$ until $T_{R,m}^t = \mathrm{T}_{R,m}^{t,\mathrm{eq}}$, we can obtain improvement in $B_m^{t,B}$ in either case.



(a) (b)

Fig. 2: B_m^t versus $T_{R,m}^t$

2) Equivalent Optimization of RIS Coefficients: According to (9), it is obvious that a small $T_{R,m}^t$ contributes to small energy consumption in transmission. However, since there is no improvement in $B_m^{t,B}$ by decreasing $T_{R,m}^t$ when $T_{R,m}^t < \mathrm{T}_{R,m}^{t,\mathrm{eq}}$, simply aiming to achieve a small value of $T_{R,m}^t$ to achieve the optimal objective in **P3** is not accurate. As a consequence, we construct the following **P4** as an equivalent problem to **P3**:

$$\textbf{P4:} \underset{\boldsymbol{\theta}}{\textbf{Minimize}} \sum_{m=1}^{M} [S_m^t (T_{R,m} - \mathrm{T}_{R,m}^{t,\mathrm{eq}} + |T_{R,m} - \mathrm{T}_{R,m}^{t,\mathrm{eq}}|) + E_m^t] \tag{21}$$

$$\textbf{s. t.} \quad T_{R,m}^t \leq \mathrm{T}_m^{t,\mathrm{th}}, \forall m \in \mathcal{M} \tag{21a}$$

where $\mathrm{T}_m^{t,\mathrm{th}}$ in (21a) is the transmission time requirement when $B_m^t = 2\mathrm{B}_{\mathrm{th}}$, and $S_m^t = W \cdot \frac{Z_m^t F_m^t}{2X_m}$ is the weighted parameter for UE m which remains constant during the t-frame. The underlying rationale in **P4** is the following. When $T_{R,m}^t$ decreases and $T_{R,m}^t \geq \mathrm{T}_{R,m}^{t,\mathrm{eq}}$, we attain improvements in both the computing queue backlog B_m^t and the energy consumption E_m^t. On the other hand, when $T_{R,m}^t$ decreases but $T_{R,m}^t \leq \mathrm{T}_{R,m}^{t,\mathrm{eq}}$, we can only obtain an improvement in E_m^t since $T_{R,m} - \mathrm{T}_{R,m}^{t,\mathrm{eq}}$ will be canceled by its absolute value in this case.

However, **P4** is still nontrivial to solve because of the presence of the absolute value of $T_{R,m} - \mathrm{T}_{R,m}^{t,\mathrm{eq}}$ in the objective.To address this, we introduce an auxiliary parameter κ_m^t and transform **P4** in to **P4-1**:

$$\textbf{P4-1: } \underset{\boldsymbol{\theta}, \{\kappa_m^t\}}{\textbf{Minimize}} \sum_{m=1}^{M} \left(2S_m^t |\kappa_m^t| + E_m^t\right) \tag{22}$$

$$\textbf{s. t.} \quad T_{R,m}^t \leq \mathrm{T}_m^{t,\mathrm{th}}, \forall m \in \mathcal{M} \tag{22a}$$

$$T_{R,m}^t \leq \mathrm{T}_m^{t,\mathrm{eq}} + \kappa_m^t, \forall m \in \mathcal{M} \tag{22b}$$

$$\kappa_m^t \geq -\mathrm{T}_m^{t,\mathrm{eq}}, \forall m \in \mathcal{M}. \tag{22c}$$

In **P4-1**, (22b) and (22c) ensure that $T_{R,m}^t$ is not negative.

Considering the case when $T_{R,m}^t \leq \mathrm{T}_m^{t,\mathrm{eq}}$, i.e., $\kappa_m^t \leq 0$, in **P4-1**, we note that any decrease in $T_{R,m}^t$ will only result in an improvement in E_m^t since κ_m^t will be 0 to have $|\kappa_m^t|$ minimized. This is due to the fact that given any $T_{R,m}^t \leq \mathrm{T}_m^{t,\mathrm{eq}}$, the constraint in (22b) will always be satisfied when we set $\kappa_m^t = 0$. Consequently, decreasing $T_{R,m}^t$ will only improve E_m^t in this case.

On the other hand, when $\kappa_m^t > 0$, any decrease in $T_{R,m}^t$ will not only improve E_m^t, but also lead to a smaller $|\kappa_m^t|$ since we can always tighten the upper bound of $T_{R,m}^t$ in (22b) by adjusting κ_m^t to be smaller, which results in further improvement in the objective of **P4-1**. In this case, the improvement of $|\kappa_m^t|$ corresponds to the enhancement in computing queue backlog B_m^t.

Therefore, when $T_{R,m}^t \leq \mathrm{T}_{R,m}^{t,\mathrm{eq}}$, we can obtain an improvement only in the energy consumption as we keep decreasing $T_{R,m}^t$. We can achieve a weighted improvement both in the energy consumption and computing queue backlog B_m^t when we decrease $T_{R,m}^t$ until $T_{R,m}^t = \mathrm{T}_{R,m}^{t,\mathrm{eq}}$. Consequently, **P4-1** is equivalent to **P4**.

3) Optimization of RIS Coefficients: We can adopt similar methods as in our previous work [8] to solve **P4-1**. The following descriptions pertain to a given frame t, and hence, for brevity, we can eliminate t in the notations of e.g., channel vectors and RIS coefficients. We first define a vector $\boldsymbol{\phi} = [\phi_1, \phi_2, ..., \phi_K]^H$, where $\phi_k = e^{i\theta_k}$. We subsequently define $\boldsymbol{\Phi}_m = \mathrm{diag}(\boldsymbol{h}_{r,m}^H)\boldsymbol{G} \in \mathbb{C}^{K\times N}$ so that $\boldsymbol{h}_{r,m}^H \Theta \boldsymbol{G} = \boldsymbol{\phi}^H \boldsymbol{\Phi}_m$, and thereby we have $||\boldsymbol{h}_{d,m}^H + \boldsymbol{h}_{r,m}^H \Theta \boldsymbol{G}||^2 = ||\boldsymbol{h}_{d,m}^H + \boldsymbol{\phi}^H \boldsymbol{\Phi}_m||^2$. Note that $||\boldsymbol{h}_{d,m}^H + \boldsymbol{\phi}^H \boldsymbol{\Phi}_m||^2 = ||\boldsymbol{h}_{d,m}^H||^2 + \boldsymbol{h}_{d,m}^H \boldsymbol{\Phi}_m^H \boldsymbol{\phi} + \boldsymbol{\phi}^H \boldsymbol{\Phi}_m \boldsymbol{h}_{d,m} + \boldsymbol{\phi}_m^H \boldsymbol{\Phi}_m \boldsymbol{\Phi}_m^H \boldsymbol{\phi}$, similar to [9], we hence introduce an auxiliary variable χ and define

$$\boldsymbol{W}_m = \begin{bmatrix} \boldsymbol{\Phi}_m \boldsymbol{\Phi}_m^H & \boldsymbol{\Phi}_m \boldsymbol{h}_{d,m} \\ \boldsymbol{h}_{d,m}^H \boldsymbol{\Phi}_m^H & 0 \end{bmatrix}, \widetilde{\boldsymbol{\phi}} = \begin{bmatrix} \boldsymbol{\phi} \\ \chi \end{bmatrix}. \tag{23}$$

Therefore, the SNR γ_m can be further expressed as $\gamma_m = \rho_m(||\boldsymbol{h}_{d,m}^H||^2 + \widetilde{\boldsymbol{\phi}}^H \boldsymbol{W}_m \widetilde{\boldsymbol{\phi}})$. Next, a positive semidefinite matrix (PSD) $\boldsymbol{\Psi}$ associated to the RIS reflecting coefficients is constructed. Specifically, we define $\boldsymbol{\Psi} = \widetilde{\boldsymbol{\phi}}\widetilde{\boldsymbol{\phi}}^H$ with the constraints $\boldsymbol{\Psi} \succeq \boldsymbol{0}$ and $\mathrm{rank}(\boldsymbol{\Psi}) = 1$. Consequently, we have $\widetilde{\boldsymbol{\phi}}^H \boldsymbol{W}_m \widetilde{\boldsymbol{\phi}} = \mathrm{Tr}(\boldsymbol{W}_m \widetilde{\boldsymbol{\phi}}\widetilde{\boldsymbol{\phi}}^H) = \mathrm{Tr}(\boldsymbol{W}_m \boldsymbol{\Psi})$. Based on the above analysis, we can express the SNR of UE m in the t-th frame as $\gamma_m^t = \rho_m(||(\boldsymbol{h}_{d,m}^t)^H||^2 + \mathrm{Tr}(\boldsymbol{W}_m^t \boldsymbol{\Psi}))$, and we have

$$T_{R,m}^t = \frac{U_m^t I(D_m^t)}{\mathcal{B}\log_2(1 + \rho_m(||(\boldsymbol{h}_{d,m}^t)^H||^2 + \mathrm{Tr}(\boldsymbol{W}_m^t \boldsymbol{\Psi})))}. \tag{24}$$

In **P4-1**, let $\gamma_{\mathrm{th,m}}^t$ represent the threshold SNR when the equality in (22a) holds. Consequently, (22a) requires us to satisfy $\gamma_m^t = \rho_m(||(\boldsymbol{h}_{d,m}^t)^H||^2 + \mathrm{Tr}(\boldsymbol{W}_m^t \boldsymbol{\Psi})) \geq \gamma_{\mathrm{th,m}}^t, \forall m \in \mathcal{M}$, which is equivalent to the following inequalities:

$$\mathrm{Tr}(\boldsymbol{W}_m^t \boldsymbol{\Psi}) \geq \frac{\gamma_{\mathrm{th,m}}^t}{\rho_m} - ||(\boldsymbol{h}_{d,m}^t)^H||^2, \quad \forall m \in \mathcal{M}. \tag{25}$$

Similarly, we can transform constraint (22b) into following inequality via (24):

$$\mathrm{Tr}(\boldsymbol{W}_m^t \boldsymbol{\Psi}) \geq \frac{1}{\rho_m}\left(2^{\frac{U_m^t I(D_m^t)}{\mathcal{B}(\mathrm{T}_m^{t,\mathrm{eq}} + \kappa_m^t)}} - 1\right) - ||(\boldsymbol{h}_{d,m}^t)^H||^2, \forall m \in \mathcal{M}, \tag{26}$$

where $\mathrm{T}_m^{t,\mathrm{eq}} = \frac{B_m^{t-1} X_m}{F_m^t} - T_{L,m}^t$, which is fixed when both D_m^t and F_m^t are determined in the t-th frame.

Based on all the above analysis, **P4-1** is transformed into the following problem **P4-2**:

$$\textbf{P4-2:} \ \underset{\boldsymbol{\Psi}, \{\kappa_m^t\}}{\textbf{Minimize}} \sum_{m=1}^{M} \left(2S_m^t |\kappa_m^t| + P_m T_{R,m}^t\right) \tag{27}$$

$$\textbf{s. t.} \quad \boldsymbol{\Psi}_{k,k} = 1, \quad k = 1, 2, ..., K+1, \tag{27a}$$

$$\boldsymbol{\Psi} \succeq \boldsymbol{0}, \tag{27b}$$

$$(22c), (25), (26). \tag{27c}$$

In **P4-2**, we relax the non-convex constraint $\mathrm{rank}(\boldsymbol{\Psi}) = 1$ and adopt the semidefinite relaxation (SDR) method to address it. Therefore, **P4-2** becomes a convex semidefinite program (SDP) that can be solved optimally by using a conventional convex optimization tool. Note that only offloading transmission power consumption $E_{R,m}^t = P_m T_{R,m}^t$ is considered in **P4-2** since the local computing energy consumption $E_{L,m}^t$ is fixed when D_m^t has already been determined in the t-th frame. Also note that solving **P4-2** will not necessarily give us a rank-one solution, and in this case the Gaussian randomization [13] can be utilized to retrieve a sub-optimal solution.

In Algorithm 1, we summarize the entire proposed framework for collaborative inference in an RIS-assisted MEC network, combining the learning and optimization steps.

Algorithm 1 Framework for Proposed Collaborative Inference.

```
Initialization:
  1) Initialize computing backlog limitation $\mathrm{B}_{\mathrm{th}}$, parameters
     $\{P_m, \Gamma_m, f_m, X_m\}$ of each UE, bandwidth $\mathcal{B}$ and maximum
     available CPU frequency $F_{\max}$ at the MEC server.
  2) Initialize all neural networks and the experience replay memory.
Actions:
  1) Obtain initial state $s^0$.
  2) For $t = 1 : T$
  3) Check all the DNN tasks and generate $\{U_m^t\}$.
  4) Determine the offloading decisions and CPU frequency allocations $a^t$
     by the actor network according to current state $s^t$;
  5) With given $\boldsymbol{D}^t$ and $\boldsymbol{F}^t$, obtain the RIS coefficients
     $\boldsymbol{\theta}^t$ via solving P4-2.
  6) Obtain the current computing queue backlogs $\{B_m^t\}$, which are given
     by the updated state space.
  7) Observe reward $r^t$ and new state $s^{t+1}$.
  8) Store transition $(s^t, a^t, r^t, s^{t+1})$ in the experience replay memory;
  9) Sample a random minibatch transition from the experience replay memory;
 10) Train the critic and actor network, respectively;
 11) Update target networks.
 12) End for.
```

IV. Numerical Results

In this section, we analyze the performance of the proposed algorithm. In the simulations, the channels are modeled as follows: $\boldsymbol{h}_{l,m} = \sqrt{\xi_0 d_{l,m}^{-\alpha_{l,m}}}\widetilde{\boldsymbol{g}}_{l,m}$, $l \in \{d, r\}$ and $\boldsymbol{\mathcal{G}} = \sqrt{\xi_0 d_B^{-\alpha_B}}\widetilde{\boldsymbol{g}}_B$. $d_{l,m}$, $\alpha_{l,m}$ and $\widetilde{\boldsymbol{g}}_{l,m}$ denote the distance to the RIS/BS, path loss exponent, and complex Gaussian distributed fading components for the m-th UE, respectively. Similarly, d_B, α_B, $\widetilde{\boldsymbol{g}}_B$ are the distance from the RIS to the BS, path loss exponent, and complex Gaussian distributed fading components of such links. The channel parameters are listed below in Table I.

Parameter	Definition	Value
$\alpha_{d,m}$	Path loss exponent for the m-th UE to the BS	5
$\alpha_{r,m}$	Path loss exponent for the m-th UE to the RIS	2
α_B	Path loss exponent from the RIS to the BS	3.5
ξ_0	Path loss at the reference point $d_0 = 1$ m	-30 dB
σ^2	Noise power	-95 dBm

TABLE I: List of channel parameters.

In Fig. 3, we first analyze the average computing backlog at the MEC server attained with the proposed algorithm with and without considering RIS assistance. It is evident that with the deployment of RIS, whose coefficients are obtained from our proposed RIS optimization subroutine, results in a reduced average computing backlog. We further observe that the computing backlog is improved when the maximum CPU frequency constraint $F_{\max}$ at the MEC server increases, which is expected since increasing $F_{\max}$ leads to enhances processing capability at the MEC server.

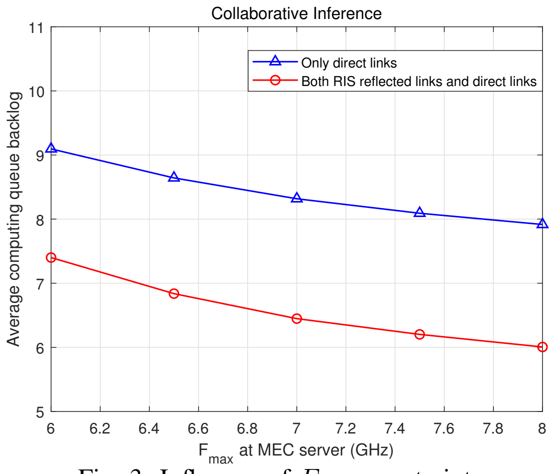


Fig. 3: Influence of $F_{\max}$ constraint.

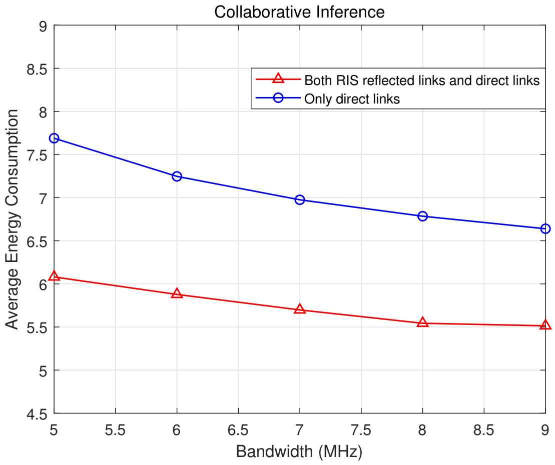


Fig. 4: Influence of bandwidth $\mathcal{B}$.

Next, in Fig. 4, we investigate the impact of bandwidth used for transmission in the offloading phase. This figure plots the curves of long-term average energy consumption versus the bandwidth $\mathcal{B}$, where the red and blue curves represent the long-term average energy consumption with and without deploying RIS. In Fig. 4, it is readily seen that the energy consumption is lower with RIS assistance. We additionally observe that the long-term average energy consumption becomes smaller as the bandwidth $\mathcal{B}$ is increased. This phenomenon occurs because increasing the bandwidth $\mathcal{B}$ mainly enables the UEs to offload more layers of DNN tasks to be processed at the MEC server, providing improvements in the energy consumption.

Finally, we evaluate the convergence of the proposed DRL-based algorithm embedded with the RIS coefficients optimization subroutine, as presented in Fig. 5. This figure shows the reward curve as the number of training episodes grows. It reveals that the training will converge around 720 episodes, which assures the feasibility of our proposed algorithm in addressing the long-term constrained optimization problem.

V. Conclusion

In this paper, we have investigated the long-term average energy consumption of collaborative inference in an RIS-assisted MEC network subject to long-term average computing queue backlog constraints. We have first presented the system model and described the collaborative inference model as well as wireless transmissions when RIS is employed. We then have defined computing queue backlog at the MEC server and introduced the energy consumption for the UEs. We have subsequently formulated an optimization problem aimed at minimizing the long-term average energy consumption for all UEs, subject to long-term average computing queue backlog, maximum computing queue backlog and maximum MEC CPU frequency constraints. To address this problem, we have presented a DRL learning based approach to optimize the offloading decisions $\{D_m\}$ and frequency allocations $\{F_m\}$ at the MEC server. Furthermore, we have proposed an optimization subroutine to find the optimal RIS coefficients for any given $\{D_m\}$ and $\{F_m\}$. Our numerical results indicate that the introduction of RIS results in a better performance, and increasing the maximum CPU frequency $F_{\max}$ at the MEC server improves the computing queue backlog. Furthermore, we have also observed that increasing the bandwidth in offloading transmission leads to a lower average energy consumption. Finally, we have explicitly demonstrated the convergence performance, further validating the effectiveness of our proposed algorithm in this paper.

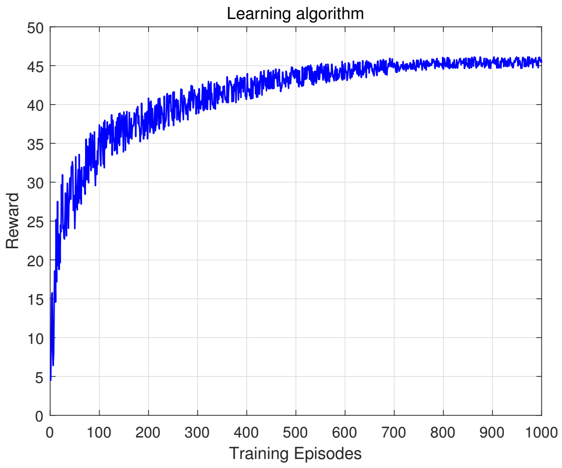


Fig. 5: Convergence.

References

[1] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[3] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[4] T. Mohammed, C. Joe-Wong, R. Babbar, and M. Di Francesco, "Distributed inference acceleration with adaptive DNN partitioning and offloading," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 854–863, IEEE, 2020.

[5] W. Wu, P. Yang, W. Zhang, C. Zhou, and X. Shen, "Accuracy-guaranteed collaborative DNN inference in industrial IoT via deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4988–4998, 2020.

[6] M. Di Renzo, A. Zappone, M. Debbah, M.-S. Alouini, C. Yuen, J. De Rosny, and S. Tretyakov, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2450–2525, 2020.

[7] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface aided wireless communications: A tutorial," *IEEE Transactions on Communications*, 2021.

[8] Y. Yang and M. C. Gursoy, "Energy-Efficient Scheduling in RIS-Aided MEC Networks for Collaborative Inference," in *ICC 2023-IEEE International Conference on Communications*, pp. 5377–5382, IEEE, 2023.

[9] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[10] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Springer Nature, 2022.

[11] J. Luo, F. R. Yu, Q. Chen, and L. Tang, "Adaptive video streaming with edge caching and video transcoding over software-defined mobile networks: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1577–1592, 2019.

[12] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[13] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network: Joint active and passive beamforming design," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2018.