Hierarchical Meta-Reinforcement Learning for Resource-Efficient Slicing in O-RAN

Xianfu Chen*, Celimuge Wu†, Zhifeng Zhao‡, Yong Xiao§, Shiwen Mao¶, and Yusheng Ji‖
*VTT Technical Research Centre of Finland Ltd, Oulu, Finland
†The University of Electro-Communications, Tokyo, Japan
‡Zhejiang Lab, Hangzhou, China
§Huazhong University of Science and Technology, Wuhan, China
¶Auburn University, Auburn, AL, USA
‖National Institute of Informatics, Tokyo, Japan

GLOBECOM 2023 - 2023 IEEE Global Communications Conference | 979-8-3503-1090-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GLOBECOM54140.2023.10437350

***Abstract*—Open radio access network (O-RAN) slicing allows the flexible control of network components and resources to satisfy the ever increasing demand of mobile applications. To optimize service provisioning, efficient management of limited radio resources is challenging due to the orchestration among network slices in the long-timescale and the slice configurations according to the mobile user (MU) statistics in the short-timescale. In this paper, we first propose a novel meta Markov decision process framework to mathematically formulate the problem of two-timescale radio resource management (RRM) in O-RAN slicing. The original RRM problem is then decoupled into a long-timescale master problem and a short-timescale subproblem, which are solved by a hierarchical reinforcement learning (RL) mechanism. Our proposed hierarchical RL mechanism includes a deep RL algorithm, solving the optimal long-timescale RRM policy, and a linear-decomposition based meta-RL algorithm, solving the optimal short-timescale RRM policy. Numerical experiments verify the theoretical analysis and show that our proposed hierarchical RL mechanism outperforms the most representative state-of-the-art baselines.**

***Index Terms*—O-RAN, spectral efficiency, two-timescale optimization, hierarchical RL, meta-learning.**

I. Introduction

Future wireless networks are expected to provide pervasive connectivity for a wide variety of mobile applications with diverse quality-of-service (QoS) and quality-of-experience requirements. To meet such a trend, the next-generation radio access network (RAN) architecture will be built upon the advances in softwarization and programmability. This brings the opportunity to slice the RAN functionalities, which are tailored to satisfy the specific requirements [1]. Under the open RAN (O-RAN) framework [2], RAN slicing is fully supported by disaggregating the hardware from the software to allow flexible control across the network components. As shown in Fig. 1, the O-RAN is split into central unit (CU), distributed unit (DU) and radio unit (RU) implementing different protocol stacks that are coordinated by a RAN intelligent controller (RIC). In O-RAN slicing, the RIC consists of a non-real-time RIC and a real-time RIC, which deploy, respectively, the slice-level orchestration and the fine-grained user-level configurations [3].

This research was supported in part by the Business Finland under Projects Eware-6G and Cloudify-6G, in part by the ROIS NII Open Collaborative Research under Grant 23S0601, in part by the JSPS KAKENHI under Grants 20H00592 and 21H03424, and in part by the NSF under Grant ECCS-1923717.

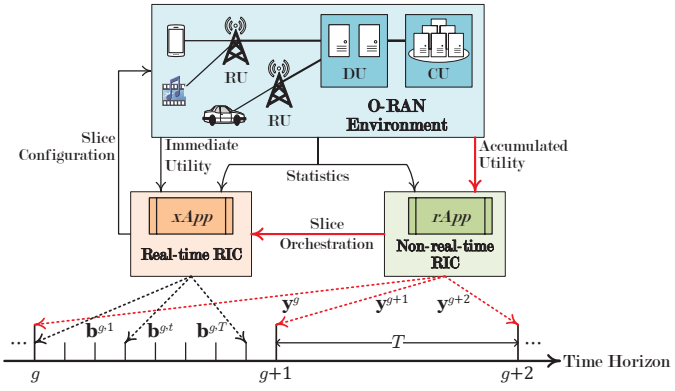


Fig. 1. Two-timescale resource efficiency slicing in O-RAN, where T is the length of an episode, while $\mathbf{y}^g$ and $\mathbf{b}^{g,t}$ are the slice-level action during each episode $g \in \mathbb{N}_+$ and the user-level actions at each time slot t.

In order to optimize service provisioning, radio resource management (RRM) in O-RAN slicing is challenging:

- The limited spectrum has to be efficiently utilized among the network slices as well as the mobile users (MUs) in a network slice;
- The strict QoS requirements for the MUs are guaranteed by the service agreement [4];
- The traffic from MUs exhibits highly dynamic spatial and temporal variations.

In the literature, there exist some related efforts. For example, in [3], Puligheddu et al. proposed a greedy approximation algorithm to solve the problem of semantic flexible O-RAN slicing. In [5], D'Oro et al. designed low-complexity solutions to support the real-world applications of intelligence orchestration in O-RAN slicing. In [6], Thaliath et al. derived a long short-term memory (LSTM) based predictive resource provisioning scheme for O-RAN slicing with the purpose of QoS protection. In [7], Rezazadeh et al. put forward a federated deep RL approach, which adapts the RRM policy to the long-term traffic dynamics in O-RAN. In [8], Lacava et al. combined the random ensemble mixture and the state-

of-the-art convolutional neural network into an intelligent user-specific O-RAN traffic steering framework for optimal handover. In [9], Abedin et al. employed a two-sided matching game and an actor-critic model to solve the RRM problem under varying O-RAN traffic.

However, the flexibility in O-RAN slicing makes the RRM even more challenging. Different from the real-time (or near real-time) user-level radio resource configurations, the slice-level radio resource orchestration is usually performed in the long-timescale (i.e., non-real-time), resulting in a two-timescale optimization problem in O-RAN slicing [10]. In our priori work [11], we formulated the network slicing as a Markov decision process (MDP) and adopted a deep RL algorithm to identify the optimal RRM policy. The performance is constrained due to the simplification of the user-level RRM in the short-timescale using a round-robin policy [12]. In this work, we concentrate on investigating the two-timescale RRM in O-RAN slicing. Attacking such a stochastic optimization problem is challenging since the radio resource has to be simultaneously adapted to different traffics in different timescales [13].

In line with the above discussions, this work first establishes a novel meta-MDP framework, within which the user-level configurations in the short-timescale are conditioned on the slice-level orchestration in the long-timescale. Then the original RRM problem is decoupled into a long-timescale master problem and a short-timescale subproblem. To find the optimal RRM control policy, the single-agent deep actor-critic algorithm [14] and the meta-RL [15] together form an innovative hierarchical deep RL mechanism, which is another major contribution from this work. To our best knowledge, this work is among the first that comprehensively study the two-timescale RRM in O-RAN slicing. The rest of the paper is organized as follows. In the next section, we elaborate on the system model and the assumptions. In Section III, we propose a novel meta-MDP framework and mathematically formulate the two-timescale optimization problem of RRM in O-RAN slicing. In Section IV, we present in details the proposed hierarchical deep RL mechanism. In Section V, we verify our theoretical analysis by numerical experiments. Finally, we draw conclusions in Section VI.

II. System Model and Assumptions

As depicted in Fig. 1, this work considers an O-RAN architecture, under which a set $\mathcal{K}$ of MUs with different QoS requirements are served by a set $\mathcal{J}$ of different network slices. All network slices share a frequency band of bandwidth W, which is equally divided into a total of N sub-bands. We denote $\mathcal{K}_j$ as the set of MUs belonging to a network slice $j \in \mathcal{J}$ with the corresponding QoS criteria defined by (R_j, L_j), where R_j and L_j are the minimum data rate and the maximum latency requirements. We assume that $\cup_{j\in\mathcal{J}}\mathcal{K}_j = \mathcal{K}$ and $\mathcal{K}_{j'} \cap \mathcal{K}_j = \emptyset$, $\forall j' \neq j$ ($j' \in \mathcal{J}$). The O-RAN deploys a two-timescale RRM scheme. In the long-timescale, the rApp orchestrates the limited resource among the network slices. In the short-timescale, the xApp configures each network slice based on the MU statistics. The time horizon consists of an infinite number of slots, each of which is of duration τ.

Without loss of generality, we assume that the sub-band orchestration is performed every T ($T \geq 2$) time slots, which constitute an episode. We designate by y_j^g the number of sub-bands dedicated to each network slice $j \in \mathcal{J}$ during each episode $g \in \mathbb{N}_+$. In each network slice j, y_j^g sub-bands are then mapped to the MU traffic across each of T time slots. Let $b_{j,k}^{g,t}$ denote the number of sub-bands that are allocated to each MU $k \in \mathcal{K}_j$ at time slot $t \in \mathcal{T} = \{1, \cdots, T\}$ during episode g, which is upper bounded by B_j. Over the infinite time horizon, a time slot t in an episode g can be interchangeably indexed by $(g-1)\cdot T + t$ as well. To ensure resource isolation among the network slices, one sub-band can be allocated to at most one MU at a time slot, indicating that

$$\sum_{j\in\mathcal{J}} y_j^g \leq N, \forall g \in \mathbb{N}_+, \tag{1}$$

$$\sum_{k\in\mathcal{K}_j} b_{j,k}^{g,t} \leq y_j^g, \forall j \in \mathcal{J}, \forall g \in \mathbb{N}_+, \forall t \in \mathcal{T}. \tag{2}$$

Moreover, a MU maintains a local queue to buffer the arrived but not yet transmitted traffic data. To ease the following analysis, we assume that new traffic arrives only at the end of a time slot and no new traffic data will be accepted until the local queue is empty.

For each MU $k \in \mathcal{K}_j$ in a network slice $j \in \mathcal{J}$, we let $q_{j,k}^{g,t}$ be the queue length at the beginning of each time slot t during each episode g, while let $d_{j,k}^{g,t} \in \mathcal{D}_j$ and $a_{j,k}^{g,t}$ be the initial data size and the corresponding arrival time of the traffic buffered in the queue at the end of the time slot, where $\mathcal{D}_j$ is a finite space. With $w_{j,k}^{g,t} = \frac{b_{j,k}^{g,t}}{N}\cdot W$ allocated bandwidth, the achievable data rate can be calculated as

$$r_{j,k}^{g,t} = \begin{cases} 0, & \text{if } w_{j,k}^{g,t} = 0; \\ w_{j,k}^{g,t}\cdot\log_2\left(1+\frac{P\cdot h_{j,k}^{g,t}}{w_{j,k}^{g,t}\cdot\sigma^2}\right), & \text{otherwise}, \end{cases} \tag{3}$$

where P is the transmit power of the MUs and σ^2 is the noise power spectral density, while $h_{j,k}^{g,t} \in \mathcal{H}$ is the channel gain with $\mathcal{H}$ denoting a finite space. Then the local queue dynamics can be expressed by

$$q_{j,k}^{g+\lfloor\frac{t}{T}\rfloor, t+1-T\cdot\lfloor\frac{t}{T}\rfloor} = \begin{cases} d_{j,k}^{g,t}, & \text{if } a_{j,k}^{g,t} = (g-1)\cdot T + t; \\ \max\left\{q_{j,k}^{g,t} - \tau\cdot r_{j,k}^{g,t}, 0\right\}, & \text{if } a_{j,k}^{g,t} < (g-1)\cdot T + t \text{ and } q_{j,k}^{g,t} > 0; \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $\lfloor\cdot\rfloor$ is the floor operator. For the buffered traffic in the local queue (i.e., $q_{j,k}^{g,t} > 0$), the accumulated experienced

latency until the end of the current time slot can be accordingly obtained as follows

$$l_{j,k}^{g,t} = \begin{cases} l_{j,k}^{g-\left\lfloor \frac{T+1-t}{T} \right\rfloor, t-1+T\cdot\left\lfloor \frac{T+1-t}{T} \right\rfloor} + \tau, & \text{if } r_{j,k}^{g,t} = 0; \\ l_{j,k}^{g-\left\lfloor \frac{T+1-t}{T} \right\rfloor, t-1+T\cdot\left\lfloor \frac{T+1-t}{T} \right\rfloor} + \min\left\{ \frac{q_{j,k}^{g,t}}{r_{j,k}^{e,t}}, \tau \right\}, & \text{otherwise.} \end{cases} \tag{5}$$

Note that if $q_{j,k}^{g,t} = d_{j,k}^{g-\lfloor \frac{T+1-t}{T} \rfloor, t-1+T\cdot\lfloor \frac{T+1-t}{T} \rfloor}$, it is straightforward to have $l_{j,k}^{g-\lfloor \frac{T+1-t}{T} \rfloor, t-1+T\cdot\lfloor \frac{T+1-t}{T} \rfloor} = 0$ in (5). To make the notation consistent, we particularly set $a_{j,k}^{g,t} = 0$ and $l_{j,k}^{g,t} = 0$ when the current local queue is empty.

III. Novel Meta-MDP Formulation

In this section, we first formulate the two-timescale RRM as a meta-MDP, and then specify the long-timescale as well as the short-timescale optimization objectives.

A. Novel Meta-MDP Framework

In the O-RAN under consideration, the two-timescale RRM for resource-efficient slicing falls into the realm of a MDP with hierarchy [16], which motivates a novel meta-MDP formulation. More specifically, at the beginning of each time slot t during each episode g, the global state can be encapsulated as $\mathbf{x}^{g,t} = (\mathbf{x}_{j,k}^{g,t} : \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j)$, where $\mathbf{x}_{j,k}^{g,t} = (h_{j,k}^{g,t}, q_{j,k}^{g,t}, l_{j,k}^{g,t}) \in \mathcal{X}_j$ is the local state of a MU $k \in \mathcal{K}_j$ in a network slice $j \in \mathcal{J}$ with $\mathcal{X}_j$ being a finite local state space[1]. During each episode g, the long-timescale RRM policy ρ selects the slice-level action $\mathbf{y}^g = (y_j^g : \forall j \in \mathcal{J}) \in \mathcal{Y}$ with a probability $\rho(\boldsymbol{\chi}^g, \mathbf{y}^g)$, where $\boldsymbol{\chi}^g = (\mathbf{x}_{j,k}^{g,t} : \forall t \in \mathcal{T})$. At each time slot t, the short-timescale RRM policy φ determines the user-level action $\varphi(\mathbf{x}^{g,t}|\mathbf{y}^g) = \mathbf{b}^{g,t} \triangleq (b_{j,k}^{g,t} : \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j)$, which is conditioned on $\mathbf{y}^g$. Each MU k realizes the immediate utility $u_{j,k}(\mathbf{x}_{j,k}^{g,t}, y_j^g, b_{j,k}^{g,t}) = \alpha_j \cdot e_{j,k}(\mathbf{x}_{j,k}^{g,t}, y_j^g, b_{j,k}^{g,t}) + \beta_j \cdot s_{j,k}(\mathbf{x}_{j,k}^{g,t}, y_j^g, b_{j,k}^{g,t})$ that measures not only the spectral efficiency (SE)

$$e_{j,k}\left(\mathbf{x}_{j,k}^{g,t}, y_j^g, b_{j,k}^{g,t}\right) = \begin{cases} \frac{r_{j,k}^{g,t}}{w_{j,k}^{g,t}}, \text{ if } r_{j,k}^{g,t} > 0; \\ 0, \quad \text{otherwise,} \end{cases} \tag{6}$$

but also the service level satisfaction rate (SLSR)

$$s_{j,k}\left(\mathbf{x}_{j,k}^{g,t}, y_j^g, b_{j,k}^{g,t}\right) = \begin{cases} \frac{\mathbb{1}_{\left\{\bar{r}_{j,k}^{g,t} \geq R_j\right\}} \cdot \mathbb{1}_{\left\{l_{j,k}^{g,t} \leq L_j\right\}}}{1+\lambda_{j,k}^{g,t}}, \text{ if } \tau \cdot r_{j,k}^{g,t} \geq q_{j,k}^{g,t} > 0; \\ 0, \quad \text{otherwise.} \end{cases} \tag{7}$$

In (7), α_j and β_j are two constants. $\mathbb{1}_{\{\cdot\}}$ is the indicator function that equals 1 if the condition is met and otherwise, 0. If the new traffic arrived at the end of the time slot is dropped, we have $\lambda_{j,k}^{g,t} = 1$, and otherwise, $\lambda_{j,k}^{g,t} = 0$. The mean data rate $\bar{r}_{j,k}^{g,t}$ is given by $\bar{r}_{j,k}^{g,t} = d_{j,k}^{g-\lfloor \frac{T+1-t}{T} \rfloor, t-1+T\cdot\lfloor \frac{T+1-t}{T} \rfloor} / l_{j,k}^{g,t}$. After performing $\mathbf{b}^{g,t}$ under $\mathbf{x}^{g,t}$, the global state transits to a subsequent global state with the following controlled probability

$$\mathbb{P}\left(\mathbf{x}^{g+\left\lfloor \frac{t}{T} \right\rfloor, t+1-T\cdot\left\lfloor \frac{t}{T} \right\rfloor} | \mathbf{x}^{g,t}, \mathbf{b}^{g,t}\right) = \prod_{j\in\mathcal{J}} \prod_{k\in\mathcal{K}_j} \mathbb{P}\left(\mathbf{x}_{j,k}^{g+\left\lfloor \frac{t}{T} \right\rfloor, t+1-T\cdot\left\lfloor \frac{t}{T} \right\rfloor} | \mathbf{x}_{j,k}^{g,t}, b_{j,k}^{g,t}\right), \tag{8}$$

where $\mathbb{P}(\cdot)$ denotes the probability of an event.

B. Two-Timescale Optimization

During each episode $g \in \mathbb{N}_+$, the execution of a slice-level action $\mathbf{y}^g$ and a sequence $\{\mathbf{b}^{g,t} : \forall t \in \mathcal{T}\}$ of user-level actions leads to the discounted utility $v(\boldsymbol{\chi}^g, \mathbf{y}^g) = \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_j} (\gamma)^{t-1} \cdot u_{j,k}(\mathbf{x}^{g,t}, \mathbf{y}^g, b_{j,k}^{g,t})$, where $\gamma \in (0,1)$ is the discount factor and $(\cdot)^{t-1}$ denotes the $(t-1)$-th power. Given the initial global state $\mathbf{x}^{1,1} = \mathbf{x}$, the objective of RRM in O-RAN slicing is to maximize the expected long-term discounted utility

$$\mathbb{V}(\mathbf{x}, \rho, \varphi) = \mathsf{E}_{\rho,\varphi}\left[\sum_{g=1}^{\infty} (\gamma)^{(g-1)\cdot T} \cdot v(\boldsymbol{\chi}^g, \mathbf{y}^g) \,|\mathbf{x}\right], \tag{9}$$

where the expectation $\mathsf{E}_{\rho,\varphi}$ is taken with respect to the probability measure jointly induced by the long-timescale policy ρ and the short-timescale policy φ. (9) can also be termed as the state-value function. Formally, the two-timescale RRM optimization problem can be formulated as

$$\begin{aligned} \max_{\rho,\varphi} \ & \mathbb{V}(\mathbf{x}, \rho, \varphi) \\ \text{s.t.} \ & \text{Constraints (1) and (2).} \end{aligned} \tag{10}$$

We rewrite $\mathbb{V}(\mathbf{x}, \rho^*, \varphi^*)$ as $\mathbb{V}(\mathbf{x})$, where the optimal long-timescale RRM policy ρ^* and the optimal short-timescale RRM policy φ^* are the solution to (10).

IV. Proposed Hierarchical RL Framework

Solving the optimal RRM policies ρ^* and φ^* in (10) is challenging due to:

- The complex network dynamics statistics as in (8) is infeasible in general;
- The obtaining of global states across an episode is impossible before selecting a slice-level action;
- The short-timescale RRM policy is conditioned on the long-timescale RRM policy.

To address all these challenges, this section proposes a hierarchical RL mechanism.

A. Hierarchical Learning

By decoupling the RRM optimization problem formulated in Section III-B, we obtain the long-timescale RRM master problem and the short-timescale RRM subproblem.

[1]Given the finite spaces $\mathcal{H}$ and $\mathcal{D}_j$ as well as the limited number N of sub-bands, the local state space $\mathcal{X}_j$ is clearly finite.

1) Long-timescale RRM: Given the short-timescale RRM policy φ, the long-timescale RRM optimization problem can be formally reformulated as

$$\begin{aligned} &\max_{\rho} \ \mathbb{V}(\boldsymbol{\chi}, \rho|\varphi) \\ &\text{s.t. Constraint (1),} \end{aligned} \tag{11}$$

where we denote by $\boldsymbol{\chi} = \boldsymbol{\chi}^1$ the global states during the initial episode and

$$\mathbb{V}(\boldsymbol{\chi}, \rho|\varphi) = \mathsf{E}_{\rho,\varphi}\left[\sum_{g=1}^{\infty} (\gamma)^{(g-1)\cdot T} \cdot v(\boldsymbol{\chi}^g, \mathbf{y}^g)|\boldsymbol{\chi}, \varphi\right]. \tag{12}$$

It can be easily shown that (11) is a standard single-agent MDP. However, the dimensionality of $\boldsymbol{\chi}^g$ during each episode g grows exponentially as T and $|\mathcal{K}|$ increase. We hence propose to replace $\boldsymbol{\chi}^g$ with the historical traffic $\mathbf{z}^g = (\mathbf{z}_j^g : \forall j \in \mathcal{J})$ and adopt a deep recurrent actor-critic algorithm [17] to solve ρ^*, where $\mathbf{z}_j^g = (z_j^{g-I-1}, \cdots, z_j^{g-1})$ with each $z_j^{g-i} = \frac{1}{|\mathcal{K}_j|\cdot T} \cdot \sum_{k\in\mathcal{K}_j} \sum_{t\in\mathcal{T}} q_{j,k}^{g-i,t}$, $1 \le i \le I-1$. Given $\mathbf{z}^1 = \mathbf{z}$, we have $\mathbb{V}(\boldsymbol{\chi}, \rho|\varphi) \approx \mathbb{V}(\mathbf{z}, \rho|\varphi)$.

Algorithm 1 Learning Long-Timescale RRM Control Policy

1: Initialize the deep actor network parameters $\boldsymbol{\theta}^g$ and the deep critic network parameters $\boldsymbol{\phi}^g$, for $g = 1$.
2: **for** $i \in \{1, \cdots, I\}$ **do**
3: Randomly select a slice-level action $\mathbf{y}^i \in \mathcal{Y}$.
4: **for** $t \in \mathcal{T}$ **do**
5: Under Constraint (2), randomly choose a user-level action $b_{j,k}^{i,t}$, $\forall j \in \mathcal{J}$, $\forall k \in \mathcal{K}_j$.
6: **end for**
7: Obtain $\{z_j^i : \forall j \in \mathcal{J}\}$ at the end of episode i.
8: **end for**
9: Obtain $\mathbf{z}^g$, for $g = 1$.
10: **repeat**
11: At the actor network, take $\mathbf{z}^g$ as the input and pick a slice-level action $\mathbf{y}^g$ with a probability of $\rho_{\boldsymbol{\theta}^g}(\mathbf{z}^g, \mathbf{y}^g)$.
12: Implement φ from Algorithm 2, obtain $\mathbf{z}^{g+1}$ at the end of episode g and calculate $\mathbb{V}_{\boldsymbol{\phi}^g}(\mathbf{z}^{g+1}|\varphi)$.
13: With $\mathsf{LOSS}^g_{(\text{Actor})}(\boldsymbol{\theta}^g)$ and $\mathsf{LOSS}^g_{(\text{Critic})}(\boldsymbol{\phi}^g)$, update the deep actor and deep critic network parameters according to (13) and (16), respectively.
14: Update the episode index $g = g + 1$.
15: **until** A predefined stopping condition is satisfied.

Specifically, we approximate the optimal long-timescale RRM policy ρ^* and $\mathbb{V}(\mathbf{z}|\varphi) = \mathbb{V}(\mathbf{z}, \rho^*|\varphi)$, $\forall \mathbf{z}$, by a deep actor network $\rho_{\boldsymbol{\theta}}$ and a deep critic network $\mathbb{V}_{\boldsymbol{\phi}}(\mathbf{z}|\varphi)$, where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are the deep neural network parameters. After performing a slice-level action $\mathbf{y}^g$ under the observation $\mathbf{z}^g$ following the long-timescale RRM policy $\rho_{\boldsymbol{\theta}^g}$ at each episode g, the training of the deep actor network follows

$$\boldsymbol{\theta}^{g+1} \leftarrow \boldsymbol{\theta}^g - \eta_{\boldsymbol{\theta}} \cdot \nabla_{\boldsymbol{\theta}^j} \mathsf{LOSS}^g_{(\text{Actor})}(\boldsymbol{\theta}^g), \tag{13}$$

where $\eta_{\boldsymbol{\theta}}$ is the learning rate, $\boldsymbol{\theta}^g$ is the deep actor network parameters at episode g, while

$$\begin{aligned} \mathsf{LOSS}^g_{(\text{Actor})}(\boldsymbol{\theta}^g) = \delta_{\boldsymbol{\phi}^g}(\mathbf{z}^g|\varphi) \cdot \ln(\rho_{\boldsymbol{\theta}^g}(\mathbf{z}^g, \mathbf{y}^g)) \\ + \psi \cdot \rho_{\boldsymbol{\theta}^g}(\mathbf{z}^g, \mathbf{y}^g) \cdot \ln(\rho_{\boldsymbol{\theta}^g}(\mathbf{z}^g, \mathbf{y}^g)), \end{aligned} \tag{14}$$

with $\psi > 0$ being a constant weight, $\boldsymbol{\phi}^g$ being the deep critic network parameters at episode g, and

$$\begin{aligned} \delta_{\boldsymbol{\phi}^g}(\mathbf{z}^g|\varphi) = v(\boldsymbol{\chi}^g, \mathbf{y}^g) + (\gamma)^T \cdot \mathbb{V}_{\boldsymbol{\phi}^g}\left(\mathbf{z}^{g+1}|\varphi\right) \\ - \mathbb{V}_{\boldsymbol{\phi}^g}(\mathbf{z}^g|\varphi). \end{aligned} \tag{15}$$

With the loss function $\mathsf{LOSS}^g_{(\text{Critic})}(\boldsymbol{\phi}^g) = \left(\delta_{\boldsymbol{\phi}^g}(\mathbf{z}^g|\varphi)\right)^2$, the deep critic network parameters are updated according to

$$\boldsymbol{\phi}^{g+1} \leftarrow \boldsymbol{\phi}^g + \eta_{\boldsymbol{\phi}} \cdot \nabla_{\boldsymbol{\phi}^j} \mathsf{LOSS}^g_{(\text{Critic})}(\boldsymbol{\phi}^g), \tag{16}$$

where $\eta_{\boldsymbol{\phi}}$ is the learning rate. The learning procedure can be described as in Algorithm 1.

2) Short-timescale RRM: In a similar way, we reformulate the short-timescale RRM optimization problem as

$$\begin{aligned} &\max_{\varphi} \ \mathbb{V}(\mathbf{x}, \varphi|\rho) \\ &\text{s.t. Constraint (2),} \end{aligned} \tag{17}$$

given the long-timescale RRM policy ρ, where $\mathbb{V}(\mathbf{x}, \varphi|\rho)$ is given by (18). With the slice-level action $\mathbf{y} = \mathbf{y}^g$ during any episode g, the short-timescale RRM policy then straightforwardly becomes an episodic single-agent MDP. Along with the slice-level actions generated from the long-timescale RRM policy, the short-timescale RRM optimization problem as in (17) is a meta-MDP as discussed in Section III-A. Let $\mathbb{V}(\mathbf{x}|\rho) = \mathbb{V}(\mathbf{x}, \varphi^*|\rho)$ and (19) define the optimal Q-function. The optimal short-timescale RRM policy φ^* can then be directly obtained from $\mathbb{V}(\mathbf{x}|\rho) = \max_{\mathbf{b}} \mathbb{Q}(\mathbf{x}, \mathbf{b}|\rho)$. To address the extremely huge global state space, we leverage a deep neural network to approximate the optimal Q-function. That is, $\mathbb{Q}(\mathbf{x}, \mathbf{b}|\mathbf{y}) \approx \mathbb{Q}_{\boldsymbol{\vartheta}}(\mathbf{x}, \mathbf{b}|\mathbf{y})$, $\forall \mathbf{y} \in \mathcal{Y}$, where $\boldsymbol{\vartheta}$ denotes the DQN parameters.

For each slice-level action $\mathbf{y} \in \mathcal{Y}$, let $\mathcal{O}_{\mathbf{y}}^{g,t}$ be the pertained replay memory at the beginning of each time slot t during each episode g. Each experience in the replay memory includes the current global state $\mathbf{x}$, the selected user-level action $\mathbf{b}$, the sum of realized utilities of all MUs $\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_j} u_{j,k}(\mathbf{x}, \mathbf{y}, \mathbf{b})$, and the subsequent global state $\mathbf{x}'$. Then the DQN parameters are updated by minimizing the loss function given by (20), where $\boldsymbol{\vartheta}^{g,t}$ and $\boldsymbol{\vartheta}_-^{g,t}$ are the DQN parameters at a time slot t during each episode g and at a certain previous time slot, while $\tilde{\mathcal{O}}_{\mathbf{y}}^{g,t}$ is a randomly sampled mini-batch. We then obtain the meta-training rule of the DQN parameters,

$$\begin{aligned} \boldsymbol{\vartheta}^{g+\lfloor \frac{t}{T} \rfloor, t+1-T\cdot\lfloor \frac{t}{T} \rfloor} \leftarrow \boldsymbol{\vartheta}^{g,t} \\ + \eta_{\boldsymbol{\vartheta}} \cdot \nabla_{\boldsymbol{\vartheta}^{g,t}} \sum_{\mathbf{y}\in\mathcal{Y}} \mathsf{LOSS}^{g,t}_{(\text{DQN})}\left(\boldsymbol{\vartheta}^{g,t}|\mathbf{y}\right), \end{aligned} \tag{21}$$

where $\eta_{\boldsymbol{\vartheta}}$ is the learning rate. However, the user-level action space explodes with even a small number of MUs.

$$\mathbb{V}(\mathbf{x},\varphi|\rho)=\mathsf{E}_{\rho,\varphi}\left[\sum_{g=1}^{\infty}\sum_{t\in\mathcal{T}}(\gamma)^{(g-1)\cdot T+t-1}\cdot\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}_j}u_{j,k}\left(\mathbf{x}_{j,k}^{g,t},y_j^g,b_{j,k}^{g,t}\right)|\mathbf{x},\rho\right] \tag{18}$$

$$\mathbb{Q}(\mathbf{x},\mathbf{b}|\rho)=\mathsf{E}_{\rho,\varphi^*}\left[\sum_{g=1}^{\infty}\sum_{t\in\mathcal{T}}(\gamma)^{(g-1)\cdot T+t-1}\cdot\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}_j}u_{j,k}\left(\mathbf{x}_{j,k}^{g,t},y_j^g,b_{j,k}^{g,t}\right)|\mathbf{x},\mathbf{b}^{1,1}=\mathbf{b},\rho\right] \tag{19}$$

$$\mathsf{LOSS}_{(\mathrm{DQN})}^{g,t}\left(\boldsymbol{\vartheta}^{g,t}|\mathbf{y}\right)=\mathsf{E}_{\widetilde{\mathcal{O}}_{\mathbf{y}}^{g,t}\subset\mathcal{O}_{\mathbf{y}}^{g,t}}\left[\left(\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}_j}u_{j,k}(\mathbf{x}_{j,k},y_j,b_{j,k})+\gamma\cdot\max_{\mathbf{b}'}\mathbb{Q}_{\boldsymbol{\vartheta}_-^{g,t}}(\mathbf{x}',\mathbf{b}'|\mathbf{y})-\mathbb{Q}_{\boldsymbol{\vartheta}^{g,t}}(\mathbf{x},\mathbf{b}|\mathbf{y})\right)^2\right] \tag{20}$$

B. Linear Decomposition

From the centralized short-timescale RRM, the user-level actions are executed within each network slice, which motivates us to linearly decompose the Q-function,

$$\mathbb{Q}(\mathbf{x},\mathbf{b}|\mathbf{y})=\sum_{j\in\mathcal{J}}\mathbb{Q}_j(\mathbf{x}_j,\mathbf{b}_j|y_j), \tag{22}$$

where $\mathbf{x}_j=(\mathbf{x}_{j,k}:\forall k\in\mathcal{K}_j)$, $\mathbf{b}_j=(b_{j,k}:\forall k\in\mathcal{K}_j)$ and y_j are, respectively, the state, the user-level action and the slice-level action of network slice j at each current time slot, while $\mathbb{Q}_j(\mathbf{x}_j,\mathbf{b}_j|y_j)$ is defined to be the per-slice Q-function for each network slice $j\in\mathcal{J}$ that satisfies

$$\begin{aligned}\mathbb{Q}_j(\mathbf{x}_j,\mathbf{b}_j|y_j)=&\sum_{k\in\mathcal{K}_j}u_{j,k}(\mathbf{x}_{j,k},y_j,b_{j,k})\\&+\gamma\cdot\sum_{\mathbf{x}_j'}\mathbb{P}\left(\mathbf{x}_j'|\mathbf{x}_j,\mathbf{b}_j\right)\cdot\max_{\mathbf{b}_j'}\mathbb{Q}_j(\mathbf{x}_j',\mathbf{b}_j'|y_j),\end{aligned} \tag{23}$$

where $\mathbf{x}_j'=(\mathbf{x}_{j,k}':\forall k\in\mathcal{K}_j)$ and $\mathbf{b}_j'=(b_{j,k}':\forall k\in\mathcal{K}_j)$ are the state and the user-level actions at the subsequent time slot. From the previous discussions, the definition of an identical utility function for MUs in the same network slice indicates the homogeneity during the sub-band allocation and the traffic data transmission, which inspires us to further decompose each per-slice Q-function into a series of per-user Q-functions

$$\mathbb{Q}_j(\mathbf{x}_j,\mathbf{b}_j|y_j)=\sum_{k\in\mathcal{K}_j}\mathbb{Q}_{j,k}(\mathbf{x}_{j,k},b_{j,k}|y_j)\,, \tag{24}$$

where

$$\begin{aligned}&\mathbb{Q}_{j,k}(\mathbf{x}_{j,k},b_{j,k}|y_j)=u_{j,k}(\mathbf{x}_{j,k},y_j,b_{j,k})+\\&\gamma\cdot\sum_{\mathbf{x}_{j,k}'}\mathbb{P}\left(\mathbf{x}_{j,k}'|\mathbf{x}_{j,k},b_{j,k}\right)\cdot\max_{b_{j,k}'}\mathbb{Q}_j\left(\mathbf{x}_{j,k}',b_{j,k}'|y_j\right).\end{aligned} \tag{25}$$

We emphasize that given the slice-level action, the user-level action of each network slice at each time slot is determined so as to maximize the per-slice Q-function.

Based on (24), training a common DQN for the MUs in the same network slice can be a promising alternative, instead of training a separate DQN for each MU. For each MU $k\in\mathcal{K}_j$ in each network slice $j\in\mathcal{J}$, we approximate the per-user Q-function by $\mathbb{Q}_{j,k}(\mathbf{x}_{j,k},b_{j,k}|y_j)\approx\mathbb{Q}_{\boldsymbol{\vartheta}_j}(\mathbf{x}_{j,k},b_{j,k}|y_j)$, $\forall\mathbf{x}_{j,k}$, $\forall b_{j,k}$ and $\forall y_j$, where $\boldsymbol{\vartheta}_j$ is the associated DQN parameters. Accordingly, a replay memory $\mathcal{O}_{j,y_j}^{g,t}$ is maintained for each network slice j and is updated with the most useful experience picked from the most recent interactions between MUs and O-RAN [18]. A most useful experience can be from the MU that achieves the largest immediate utility. The meta-training process follows

$$\begin{aligned}\boldsymbol{\vartheta}_j^{g+\left\lfloor\frac{t}{T}\right\rfloor,t+1-T\cdot\left\lfloor\frac{t}{T}\right\rfloor}\leftarrow&\,\boldsymbol{\vartheta}_j^{g,t}\\&+\eta_{\boldsymbol{\vartheta}}\cdot\nabla_{\boldsymbol{\vartheta}_j^{g,t}}\sum_{y_j}\mathsf{LOSS}_{j,(\mathrm{DQN})}^{g,t}\left(\boldsymbol{\vartheta}_j^{g,t}|y_j\right),\end{aligned} \tag{26}$$

where we express the loss function $\mathsf{LOSS}_{j,(\mathrm{DQN})}^{g,t}\left(\boldsymbol{\vartheta}_j^{g,t}|y_j\right)$ by (27) with $\widetilde{\mathcal{O}}_{j,y_j}^{g,t}$ being a randomly sampled mini-batch. In brief, the process of learning short-timescale RRM control policy is summarized as in Algorithm 2, where $\epsilon\in(0,1)$ is the exploration probability.

V. Numerical Experiments

In this section, we numerically evaluate the proposed hierarchical RL mechanism for two-timescale RRM in O-RAN slicing by conducting experiments with TensorFlow. To implement our proposed hierarchical RL mechanism, Algorithms 1 and 2 are applied to iteratively update the long-timescale and the short-timescale RRM control policies.

A. Experimental Configurations

In experiments, we setup an O-RAN that covers a 300×300 m^2 square area. There are a number 600 of MUs subscribed to a set $\mathcal{J}=\{1,2,3\}$ of three network slices, which support, respectively, voice over LTE, enhanced mobile broadband and ultra-reliable low latency communications services. The frequency bandwidth is $W=10$ MHz and the total number of sub-bands is $N=50$. The MU configurations and the traffic models follow our prior work [11]. During the meta-DQN training, we collect 5000 interaction experiences for

$$\mathsf{LOSS}^{g,t}_{j,(\mathrm{DQN})}\left(\boldsymbol{\vartheta}^{g,t}_{j}|y_j\right)=\mathsf{E}_{\widetilde{\mathcal{O}}^{g,t}_{j,y_j}\subset\mathcal{O}^{g,t}_{j,y_j}}\left[\left(u_{j,k}(\mathbf{x}_{j,k},y_j,b_{j,k})+\gamma\cdot\max_{b'_{j,k}}\mathbb{Q}_{\boldsymbol{\vartheta}^{g,t}_{j,-}}\left(\mathbf{x}'_{j,k},b'_{j,k}|y_j\right)-\mathbb{Q}_{\boldsymbol{\vartheta}^{g,t}_{j}}(\mathbf{x}_{j,k},b_{j,k}|y_j)\right)^2\right] \quad (27)$$

Algorithm 2 Learning Short-Timescale RRM Control Policy

1: Initialize the replay memory $\{\mathcal{O}^{g,t}_{j,y_j}:\forall y_j,\forall j\in\mathcal{J}\}$, and the DQN parameters $\{\boldsymbol{\vartheta}^{g,t}_{j}:\forall j\in\mathcal{J}\}$ and $\{\boldsymbol{\vartheta}^{g,t}_{j,-}=\boldsymbol{\vartheta}^{g,t}_{j}:\forall j\in\mathcal{J}\}$, for $g=1$ and $t=1$.
2: **repeat**
3: Select a slice-level action $\mathbf{y}^g$ from Algorithm 1.
4: **for** $t\in\mathcal{T}$ **do**
5: In each network slice $j\in\mathcal{J}$, observe the local states $\{\mathbf{x}^{g,t}_{j,k}:\forall k\in\mathcal{K}_j\}$, and choose $\mathbf{b}^{g,t}_{j}$ that maximizes $\sum_{k\in\mathcal{K}_j}\mathbb{Q}_{\boldsymbol{\vartheta}^{g,t}_{j}}(\mathbf{x}^{g,t}_{j,k},b^{g,t}_{j,k}|y^g_j)$ with a probability of $1-\epsilon$ or randomly choose a user-level action $b^{g,t}_{j,k}$ with a probability of ϵ, $\forall k\in\mathcal{K}_j$.
6: Update the replay memory $\{\mathcal{O}^{g,t}_{j,y_j}:\forall y_j,\forall j\in\mathcal{J}\}$ from the most recent interaction experiences.
7: With $\{\widetilde{\mathcal{O}}^{g,t}_{j,y_j}:\forall y_j,\forall j\in\mathcal{J}\}$ sampled from $\{\mathcal{O}^{g,t}_{j,y_j}:\forall y_j,\forall j\in\mathcal{J}\}$, update meta-DQN parameters according to (26).
8: Regularly reset $\{\boldsymbol{\vartheta}^{g,t}_{j,-}=\boldsymbol{\vartheta}^{g,t}_{j}:\forall j\in\mathcal{J}\}$.
9: **end for**
10: Update the episode index $g=g+1$.
11: **until** A predefined stopping condition is satisfied.

TABLE I
PARAMETER VALUES IN EXPERIMENTS.

Parameter	Value	Parameter	Value
I	10	P	30 dBm
B_j	10, $\forall j$	τ	0.5 ms
α_j	0.01, $\forall j$	T	2000
β_j	1, $\forall j$	σ^2	-174 dBm/Hz
ψ	0.005	γ	0.9

each replay memory, while each mini-batch consists of 2000 randomly sampled experiences. Other key parameter values are listed in Table I.

B. Results and Discussions

For performance comparisons, we simulate two representative baselines, namely, soft actor-critic with LSTM (SACL) [17] and deep recurrent Q-network (DRQN) [19]. Implementing the baselines, the sub-band allocation in each network slice during an episode adopts the round-robin policy to ensure the fairness among the MUs. Fig. 2 illustrates, respectively, the average utility, SE and SLSR performance across the learning episodes. We can observe from Fig. 2(a) that our proposed hierarchical RL process converges after around 5000 episodes and outperforms the SACL and DRQN baselines. Figs. 2(b) and 2(c) further demonstrate the achieved SE and SLSR from all mechanisms. The superior average SE performance validates that our proposed mechanism adapts the limited number of sub-bands to the varying traffic requests of MUs in three network slices from not only the long-timescale but also the short-timescale. This also tells that the round-robin policy for sub-band allocation during an episode does not sufficiently optimize the SE. Even though our proposed mechanism obtains the best SLSR performance, the average SLSR performance from all three mechanisms deteriorates as the learning progresses. The reason can be that given the weight value settings, all the three mechanisms sacrifice the SLSR performance while concentrating on maximizing the expected long-term SE.

VI. CONCLUSIONS

In this paper, we focus our efforts on investigating the RRM for resource-efficient slicing in O-RAN. Accounting for the traffic dynamics, we propose to formulate the problem of two-timescale RRM as a meta-MDP. The objective is to maximize the expected long-term utility over the infinite time horizon, where an immediate utility at each time slot measures not only the SE, but also the SLSR. By decoupling the meta-MDP into the long-timescale master problem and the short-timescale subproblem, we obtain a hierarchical RL mechanism to solve the optimal slice-level and user-level RRM control policies. Numerical experiments confirm that our proposed hierarchical RL mechanism significantly outperforms the most representative baselines in terms of SE and SLSR.

REFERENCES

[1] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2377–2392, Oct. 2019.

[2] O-RAN WG1, "O-RAN Architecture Description - v7.0," Technical Specification, 2022.

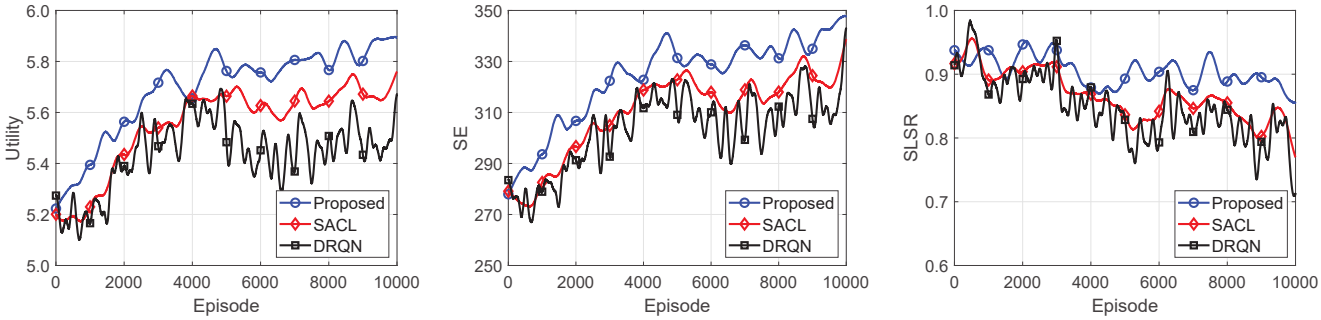
[3] C. Puligheddu, J. Ashdown, C. F. Chiasseriniy, and F. Restuccia, "SEM-O-RAN: Semantic and flexible O-RAN slicing for nextG edge-assisted mobile systems," in *Proc. IEEE INFOCOM*, New York Area, USA, May 2023.

[4] R. Yu, G. Xue, and X. Zhang, "QoS-aware and reliable traffic steering for service function chaining in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2522–2531, Nov. 2017.

[5] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Network automation through orchestrated intelligence in the open RAN," in *Proc. IEEE INFOCOM*, London, United Kingdom, May 2022.

[6] J. Thaliath, S. Niknam, S. Singh, R. Banerji, N. Saxena, H. S. Dhillon, J. H. Reed, A. K. Bashir, A. Bhat, and A. Roy, "Predictive closed-loop service automation in O-RAN based network slicing," *IEEE Commun. Stand. Mag.*, vol. 6, no. 3, pp. 8–14, Sep. 2022.

[7] F. Rezazadeh, L. Zanzi, F. Devoti, H. Chergui, X. Costa-Pérez, and C. Verikoukis, "On the specialization of FDRL agents for scalable and distributed 6G RAN slicing orchestration," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3473–3487, Mar. 2023.



(a) Utility (b) SE (in bits/Hz) (c) SLSR

Fig. 2. Average performance over all the MUs across the learning episodes.

[8] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and customized intelligence for traffic Steering in 5G networks using open RAN architectures," *IEEE Trans. Mobile Comput.*, Early Access Article, 2023.

[9] S. F. Abedin, A. Mahmood, N. H. Tran, Z. Han, and M. Gidlund, "Elastic O-RAN slicing for industrial monitoring and control: A distributed matching game and deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10808–10822, Oct. 2022.

[10] H. Zhang and V. W. S. Wong, "A two-timescale approach for network slicing in C-RAN," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6656–6669, Jun. 2020.

[11] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, Dec. 2018.

[12] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 7, pp. 1024–1039, Sep. 1991.

[13] M. Setayesh, S. Bahrami, and V. W. S. Wong, "Resource slicing for eMBB and URLLC services in radio access network using hierarchical deep learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 8950–8966, Nov. 2022.

[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. ICML*, Stockholm, Sweden, Jul. 2018.

[15] R. Fakoor, P. Chaudhari, S. Soatto1, and A. Smola, "Meta-Q-learning," in *Proc. ICLR*, Virtual, Apr.-May 2020.

[16] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artif. Intell.*, vol. 112, no. 1–2, pp. 181–211, Aug. 1999.

[17] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020.

[18] D. Zha, K.-H. Lai, K. Zhou, and X. Hu, "Experience replay optimization," in *Proc. IJCAI*, Macao, China, Aug. 2019.

[19] X. Chen, C. Wu, T. Chen, H. Zhang, Z. Liu, Y. Zhang, and M. Bennis, "Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2268–.2281, Apr. 2020.