Reinforcement Learning Meets Wireless Networks: A Layering Perspective

Yawen Chen*, Yu Liu*, Ming Zeng, Umber Saleem, Zhaoming Lu, *Member, IEEE*, Xiangming Wen, *Senior Member, IEEE*, Depeng Jin *Member, IEEE*, Zhu Han, *Fellow, IEEE*, Tao Jiang, *Fellow, IEEE*, and Yong Li, *Senior Member, IEEE*.

***Abstract*—Driven by the soaring traffic demand and the growing diversity of mobile services, wireless networks are evolving to be increasingly dense and heterogeneous. Accordingly, in such large-scale and complicated wireless networks, the optimal controlling is reaching unprecedented levels of complexity, while its traditional solutions of handcrafted offline algorithms become inefficient due to high complexity, low robustness, and high overhead. Therefore, reinforcement learning, which enables network entities to learn from their actions and consequences in the interactive network environment, attracts significant attentions. In this paper, we comprehensively review the applications of reinforcement learning in wireless networks from a layering perspective. First, we present an overview of the principle, fundamentals and several advanced models of reinforcement learning. Then we review the up-to-date applications of reinforcement learning in various functionality blocks of different network layers, ranging from the low-level physical layer, to the high-level application layer. Finally, we outline a broad spectrum of challenges, open issues, and future research directions of reinforcement learning empowered wireless networks.**

***Index Terms*—Wireless networks, reinforcement learning, communications, protocol layers, optimal controlling.**

The work of Yawen Chen, Zhaoming Lu, and Xiangming was supported in part by the China Postdoctoral Science Foundation under grant 2019M650587, in part by the National Natural Science Foundation of China under grant 61801036. The work of Yu Liu, Umber Saleem, Depeng Jin, and Yong Li was supported in part by The National Key Research and Development Program of China under grant 2018YFB1800804, the National Nature Science Foundation of China under U1936217, 61971267, 61972223, 61941117, 61861136003, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University-Tencent Joint Laboratory for Internet Innovation Technology. The work of Ming Zeng was supported in part by Beijing Institute of Technology Research Fund Program for Young Scholars. The work of Tao Jiang was supported in part by National Key R&D Program of China with Grant number 2019YFB180003400. (*Corresponding author: Yong Li.*)

Yawen Chen, Zhaoming Lu, and Xiangming Wen are with Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing Laboratory of Advanced Information Networks, Beijing, China (e-mail: chenyw@bupt.edu.cn).

Yu Liu, Umber Saleem, Depng Jin, and Yong Li are with Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: liyong07@tsinghua.edu.cn).

Ming Zeng is with School of Information and Electronics, Beijing Institute of Technology, Beijing, China (e-mail: mzengzm@163.com).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA, and also with the Department of Computer Science, University of Houston, Houston, TX 77004, USA (e-mail: zhan2@uh.edu).

Tao Jiang is with Wuhan National Laboratory for Optoelectronics and School of Electronics Information and Communications, Huazhong University of Science and Technology, Wuhan, China (e-mail: taojiang@hust.edu.cn).

* Yawen Chen and Yu Liu give equal contribution to this paper.

I. Introduction

The past decades have witnessed the rapid development of wireless communication, achieving ubiquitous networking and Internet access for various kinds of mobile devices. In consequence, both mobile services and wireless traffic are growing explosively and diversely [1, 2]. To meet the ever-increasing requirements on quantity and diversity, wireless networks are undergoing massive changes, in terms of the network architecture as well as the functionality across different network layers. In particular, wireless networks are turning to be much denser and more heterogeneous [3], along with the surge in wireless resource dimensions and the number of control parameters [4]. Hence, the optimal control of wireless networks is reaching unprecedented levels of complexity.

Traditionally, the optimal control of wireless networks is achieved by the prevalent methodology of offline optimization. Specifically, various control problems are first formulated as different kinds of optimization problems and then solved by the carefully handcrafted algorithms, using mathematical tools from optimization theories, such as convex optimization, game theory, dynamical programming and etc. Moreover, these algorithms are periodically executed to find the optimal network configuration according to the current network situation. Nevertheless, such offline algorithms are insufficient to handle increasingly complex control problems in wireless networks. The main deficiencies lie in the following perspectives:

- High complexity: Most of the optimization problems in wireless networks are non-convex and intractable, which makes it almost impossible to find an optimal solution in polynomial time.
- Low robustness: Offline optimization requires an accurate model of network dynamics. When the model parameters are imperfect or outdated, the performance may be greatly degraded.
- High overhead: Many offline algorithms require centralized processing, as well as information exchange among distributed network entities. In the context of network densification, the timely information exchange would cause extremely high overhead.

Recently, leveraging reinforcement learning (RL) [5] for intelligent wireless networks is emerging as a promising solution. As one of the most attractive machine learning (ML) techniques, RL is a learning process where an agent periodically observes the environment, makes decisions, evaluates rewards and then accordingly adjusts its policy to achieve a

goal. Moreover, recent advances in deep learning (DL) open a new area for RL [6]. Therefore, with the ever-increasing computing capacity of wireless networks, researchers start to exploit RL for increasingly complex control problems therein. In particular, RL is highly desirable for wireless networks owing to the following advantages:

- Solving various optimization problems efficiently: RL provides computation-efficient solutions to diverse optimization problems, not only Markov decision process (MDP) problems and games, but also intractable non-convex optimization problems.
- Learning network dynamics from raw observations: RL doesn't require accurate environment modeling. Through trial-and-error in the interactive environment, RL agents can autonomously learn implicit knowledge of network dynamics from raw high-dimensional observations.
- Learning distributively with local observations: With the distributed multi-agent RL, network entities can learn their own control policy with local observations and without centralized processing, which greatly contributes to overhead and signaling reduction.

According to the above properties, RL enabled online learning can meet the challenges faced by wireless networks, and overcome the shortages of traditional offline optimization, which motivates the tremendous development in this research field. Despite the growing interest of applying RL in wireless networks, most existing works focus on specific optimization problems and a comprehensive yet concise survey is still missing. To understand the development as well as provide a guide for the application of RL in wireless networks, this paper gives a systematic review on the relevant works from a layering perspective. Moreover, the key pros and cons of RL applications in different layers are discussed for further research. We also investigate the challenges and open issues, to shed a light on the future research direction in this field.

We organize the rest of this paper as follows. First, we discuss related survey articles concerning the ML empowered wireless networks in Section II, which helps distinguish the contributions of this paper. Then, the fundamental knowledge, evolution, and categories of RL are presented in Section III. Section IV reviews up-to-date RL applications from a layering perspective. In Section V, we outline challenges, open issues, and future research trends for applying RL in wireless networks. Finally, Section VI concludes this paper.

II. Related Survey Articles and Contributions of This Survey Article

With the fast development of artificial intelligence (AI), corresponding ML methods including RL, DL, and deep reinforcement learning (DRL) have attracted great attention in the area of wireless networks. Also, some papers have surveyed the crossovers between ML methods and wireless networks. Table I summarizes related survey papers and ours according to the paper scope in ML and wireless networks.

According to Table I, this paper is the first one that comprehensively investigates the applications of RL in wireless networks. In comparison, recent research efforts in IoT, CRN, WSN, SON and MEC are reviewed in [7–11, 16, 18, 19], respectively. Bkassiny *et al*. [7] identified the applicable problems of ML in the context of CRN and reviewed the corresponding solutions. Wang *et al*. [9] conducted the same review but with the emphasis on model-free learning techniques. The ML applications in IoT were investigated in [11] [16] [18], where Mohammadi *et al*. [11] reviewed the application of DL in data analysis problem of IoT, Hussain *et al*. [18] presented an in-depth review of ML applications in resource management of IoT, and [18] provided a comprehensive applications of DL in IoT. Alsheikh *et al*. [8] summarized and analyzed ML algorithms used to address the common issues in WSN, in which the comparative guide for other potential applications is also presented. Valente *et al*. [10] reviewed ML based controlling schemes in SON, and presented a clear classification of these schemes in terms of use-cases. Wang *et al*. [19] investigated how DL and MEC can be combined to enable edge intelligence and intelligent edge.

Comprehensive surveys of DL applications in wireless networks were presented in [12–15, 17] with different emphases. Both [12] and [13] reviewed the recent advances of DL as well as their applications in wireless networks. The focus of [12] was primarily on network management, whereas both network analysis and management were considered in [13]. Luong *et al*. [14] presented an overview of DRL empowered communication and networking, which covers network access, security, caching, offloading and management. Elsayed *et al*. [15] and Wang *et al*. [17] provided a comprehensive overview of ML applications in a broad range of wireless networks. The main problems reviewed in [15] included resource allocation, spectrum access, base station (BS) deployment, and energy efficiency. Whereas Wang *et al*. [17] surveyed how ML was applied to obtain Pareto-optimal solutions in multi-objective optimization of wireless networks.

Especially, the works presented in [7–11, 16, 18, 19] primarily focused on how different ML approaches (supervised and unsupervised, rules, fuzzy logic, etc.) were tailored to handle different types of wireless networks. Since each of these papers investigated only one sub-domain of wireless networks, the guidance for applying RL in all scopes of wireless networks is limited. On the other hand, the focus of [12–15, 17] was on the applications of advanced ML methods, such as DL and DRL, while the applications of traditional RL methods were not touched. As RL has long been adopted to handle the control problems in the area of wireless networks, how this research field evolved and what the reasons behind are still open questions. In addition, with virtualization enabled by software-defined networking (SDN) and network function virtualization (NFV) becoming a fundamental building block of future wireless networks [20, 21], an in-depth review of RL applications in virtual networks as well as network slicing is still missing.

Different from existing surveys, this article aims at providing an up-to-date survey of research that lies at the intersection between RL and wireless networks. Overall, the main contributions can be summarized as follows:

- We present the evolution of RL and the cutting-edge RL models from the perspective of wireless networks.

TABLE I
SUMMARY OF EXISTING SURVEY PAPERS RELATED TO RL AND WIRELESS NETWORKS.

Year	Reference	ML methods				Wireless networks							
		RL	DL	DRL	Others	CN	IoT	CRN	WSN	SON	MEC	VN	others
2013	[7]	✓			✓			✓					
2014	[8]	✓			✓				✓				
2016	[9]	✓	✓	✓	✓			✓					
2017	[10]	✓			✓					✓			
2018	[11]		✓	✓			✓						
2018	[12]		✓	✓		✓	✓						✓
2019	[13]		✓	✓		✓	✓		✓	✓	✓		✓
2019	[14]			✓		✓	✓	✓	✓		✓		✓
2019	[15]	✓	✓	✓	✓	✓	✓			✓			✓
2020	[16]			✓		✓	✓				✓		
2020	[17]	✓	✓	✓	✓	✓	✓	✓	✓				✓
2020	[18]	✓	✓	✓	✓	✓	✓						
2020	[19]		✓								✓		
2020	**Ours**	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓

CN denotes cellular network, IoT denotes Internet of Things, CRN denotes cognitive radio network, WSN denotes wireless sensor network, SON denotes self-organizing network, MEC denotes mobile edge computing, VN denotes virtual network.

- We thoroughly summarize and category RL applications in wireless networks in terms of network layers. Based on this, we illustrate how the researchers tailor RL to address various control problems across different network layers.
- We outline the challenges for applying RL in wireless networks, and also figure out the open issues, which we hope to enlighten future research directions in this area.

III. OVERVIEW OF REINFORCEMENT LEARNING

This section provides an overview of RL from fundamentals to advanced models. We start with the basic principle of RL. Next, we introduce the fundamental knowledge of MDP, the main framework for RL. Then we give a brief introduction to several important RL models as well as methods. Finally, we present the evolution of RL and its driving factors in wireless networks.

A. Basic Principle of Reinforcement Learning

RL is a promising ML paradigm which learns through interaction and focuses on sequential decision making. As illustrated in Fig. 1 The learner, which makes sequential decisions is termed as the *agent*. Everything surrounding the agent and it interacts with is defined as the *environment* [22]. These two components interact continually (shown as subscripts t, $t+1$), the agent selecting an action A_t and the environment responding to the action with state S_{t+1} changed and reward R_t returned. Specifically, the reward is a numerical value which represents a long-term objective to be maximized with the agent's choice of actions.

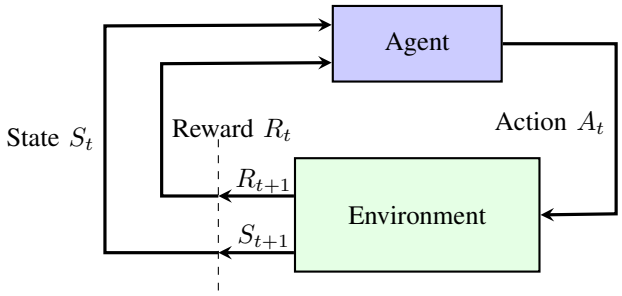


Fig. 1. Illustration of the basic principle of RL.

According to the principle above, action, state and reward characterize the interaction between the agent and its environment. Moreover, an RL agent includes three key elements, *policy*, *value function* and, optionally, *model*. Policy describes how to act according to different situations, and it can be defined as the map from observed environment states to the actions to be taken. Value function is a prediction of future reward, evaluating the goodness/badness of each state and/or action. A model is the agent's representation of the environment, predicting what the environment will do next, which is optional and will be discussed later.

B. Markov Decision Process

As a commonly used stochastic control process for sequential decision making, MDP provides a mathematical framework to formulate the RL problems [23]. Generally, MDP can be defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, which is discussed as follows,

- **States**: $\mathcal{S}$ is the set of states.
- **Actions**: $\mathcal{A}$ is the set of actions.
- **Transition Matrix**: $\mathcal{T}$ is the state transition matrix which maps the state-action pair at time t onto a distribution over possible states at $t+1$, *i.e.*, $\mathcal{T}(s_{t+1}|s_t, a_t) = \mathbb{P}[s_{t+1}|s_t, a_t]$.
- **Reward**: The reward function $\mathcal{R}$ maps a state-action pair at t to the expected value for immediate reward at $t+1$, which can be mathematically described as $\mathcal{R}(s_t, a_t) = \mathbb{E}[R_{t+1}|s_t, a_t]$.
- **Discount Factor**: The discount factor $\gamma \in [0, 1]$ determines how important future rewards are to the current state. For example, the total discounted reward from time t can be defined as $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$.

MDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ also defines policy and value function. Specifically, a policy π maps a state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$, and it can be either deterministic $a = \pi(s)$ or

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3025365, IEEE Internet of Things Journal

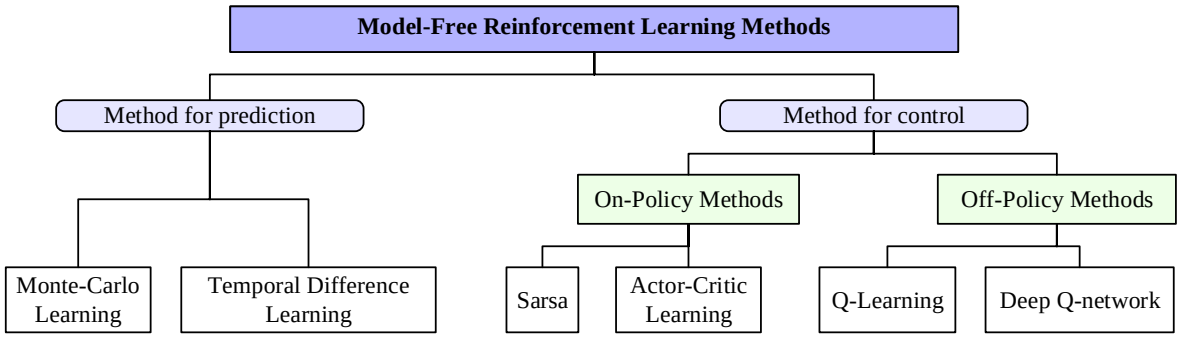


Fig. 2. The classification of model-free reinforcement learning methods.

stochastic $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$. On the other hand, MDP defines two kinds of value functions, *i.e.*, state-value function $v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$ and action-value function $q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$ (also denoted as $Q(s,a)$ and called as Q-value)[1]. The former is the expected value at state s by adopting policy π, while the latter further includes the action a in the condition.

Considering the goal is to find the optimal policy that maximizes the long-term reward, the optimal value functions in MDP can be expressed as

$$v_*(s) = \max_\pi v_\pi(s), \tag{1}$$

$$q_*(s,a) = \max_\pi q_\pi(s,a). \tag{2}$$

Also, these two optimal value functions hold such relation $v_*(s) = \max_a q_*(s,a)$, based on which the Bellman optimality equation for v_* can be obtained as follows [22],

$$v_*(s) = \max_a \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s'|s,a) v_*(s'). \tag{3}$$

However, (3) is a nonlinear equation without a closed-form solution. Thus, several RL methods are proposed to find the optimal policy by solving (3), which will be discussed in Section III-C.

Besides the generally used finite and discrete MDP, some extensions to MDP also attracted much attention and developed recently, such as infinite and continuous MDP on state and action spaces, as well as the classical multi-arm bandit (MAB) framework, which could be seen as a special case of MDP with just one state, and the reward depends only on the action [24]. An emerging direction is the partially observable MDP (POMDP), a generalization of MDP with hidden states [25]. Especially, the current state is not fully observable, and rather an observation, which is a subset of all the available information in a given state. POMDP maintains the complete history of actions and observations to determine the optimal policy. However, instead of tracking a complete history, an agent maintains a belief state based on the probability distribution obtained by a set of observations, observation probabilities and the underlying MDP. Since POMDP is complex and hard to solve, a common approach in the literature is to transform POMDP to MDP.

[1]The terms of q_π and Q are interchangeably used in this paper to refer to state-value function, which is in line with existing literature.

C. Reinforcement Learning Methods

After the basic introduction of RL and its main framework MDP, we now provide an overview of several commonly used as well as advanced RL methods.

As a matter of fact, most traditional MDP problems assume that the full knowledge of environment is known beforehand. Hence we can use dynamic programming (DP) methods such as policy iteration (PI) and value iteration (VI) to solve these MDP problems (see [22] for details). Nonetheless, RL problems focus on dynamic environment, in which the environment model is unknown, *i.e.*, both transition matrix and reward function are unknown/inaccurate. Therefore, several RL methods including model-based methods and model-free methods are proposed. The former learns to fit the environment model, transforming RL problems into traditional DP problems, while the latter is like trial-and-error learning, drawing much attention.

Thus, we particularly discuss model-free methods here. Especially, these methods can be further categorized into two classes, *i.e.*, methods for prediction and methods for control. The former is to estimate the value function v_π underlying specific policy π in an MDP, while the latter is to learn optimal value function v_* as well as optimal policy π_* together. Moreover, depending on whether the optimal action-value function $q_*(s,a)$ is updated by the policy being followed, methods for control are divided into two categories, *i.e.*, on-policy and off-policy methods. The former estimates state-value function assuming the current policy π continues being followed, while the latter estimates it assuming a greedy policy is followed rather than the current policy π. Fig. 2 presents a classification of the existing model-free RL methods, which will be introduced in the following.

1) Monte-Carlo Learning [26]: Monte-Carlo (MC) learning is commonly used for value prediction, which estimates $v_\pi(s)$ by averaging returns observed after visits to state s in episodes. Here the episode denotes a sequence of interactions between the agent and environment, from some state to terminal state, which can be expressed as $S_t, A_t, R_t, \cdots, S_{t+k}$. Also, MC learning can be applied to action-value prediction [22].

However, the variance of the returns in MC learning can be high, which leads to poor estimation. Besides, when interacting with a system in practice, it might be impossible to reset the system state to some particular state for sample returns [27].

2) Temporal Difference Learning [28]: As one of significant methods of RL, temporal difference (TD) learning is able to address above issues. Compared with MC learning, which update the value until the end of the episode, TD learning only waits until the next time step $(t+1)$ by using bootstrapping. Similar to MC Learning, TD learning can also be applied to action-value prediction.

Therefore, TD learning can learn online after every step, as well as work in continuing (non-terminating) environment, which is quite practical. Besides, the simplest TD learning is to update in the next time step, denoted by TD(0), while TD learning becomes close to MC learning if calculating n-step return with $n \to \infty$.

It is worth noticing that several RL methods for control are based on MC and TD learning concerned with the policy, which are discussed in the following.

3) Sarsa [29]: State-action-reward-state-action (Sarsa) is a typical on-policy method for RL control problems based on TD learning. As the name implies, the Q-value in Sarsa is updated based on the current state S_1, the executing action A_1, the reward R as well as the entered state S_2 after executing this action, and finally the next action A_2 the agent chooses in the new state. Thus, Sarsa updates action-value function as,

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]. \tag{4}$$

On the right side, $Q(s_t, A_t)$, R_{t+1} and $Q(s_{t+1}, A_{t+1})$ denote the old value, observed reward and the estimate of future value following the policy, respectively. α is a constant step-size parameter (also known as learning rate), and $\alpha \in (0, 1]$.

4) Q-Learning [30]: Q-learning is an important off-policy TD control method, defined by,

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]. \tag{5}$$

Compared with the update in Sarsa, Q-learning makes a difference in the estimate of optimal future value, *i.e.*, $\max_a Q(S_{t+1}, a)$, which follows a greedy policy instead of current policy.

Nevertheless, above methods are tabular methods, in which the state and action spaces are small enough to be represented as arrays/tables for finding exact solutions. Obviously, these tabular methods cannot be directly applied in large-scale problems due to memory and time limit. Therefore, several function approximation methods are proposed, in which the approximate value function is represented as a parameterized functional form with weight vector $\boldsymbol{w} \in \mathbb{R}^d$, just like $\hat{v}(s, \boldsymbol{w}) \approx v_\pi(s)$ and $\hat{q}(s, a, \boldsymbol{w}) \approx q_\pi(s, a)$. For instance, $\hat{v}$ might be a linear combination of state features, with $\boldsymbol{w}$ as the vector of feature weights. Some function approximation methods are discussed in the following.

5) Deep Q-Network [6]: Deep Q-network (DQN) is one of important function approximation methods, in which $\hat{q}$ is characterized by deep neural networks, with $\boldsymbol{w}$ as the vector of connection weights in neural networks.

By leveraging the neural network to estimate the Q-value function, the input for the network is the current state, while the output is the corresponding Q-value for each action. Especially, DQN is still based on Q-learning method, and the network training still refers to the Q-learning update equation, which updates the network weights as,

$$\boldsymbol{w}_t \leftarrow \boldsymbol{w}_t + \alpha[R_{t+1} + \gamma \max_a \hat{q}(S_{t+1}, a, \boldsymbol{w}_t) - \hat{q}(S_t, A_t, \boldsymbol{w}_t)] \nabla \hat{q}(S_t, A_t, \boldsymbol{w}_t), \tag{6}$$

DQN is a representative of DRL, which embraces the advantage of deep neural network (DNN) in approximating the value functions, and hence speeding the learning process and improve the RL performance [14]. Moreover, most DRL methods are modifications and extensions to DQN, such as double deep Q-learning methods [31], deep recurrent Q-learning methods [32], etc. In particular, unlike traditional DL techniques, DRL methods train the neural network frequently based on new experiences obtained during the agent-environment interaction.

6) Actor-Critic Method [33]: Most aforementioned methods approximate the value function using parameter $\boldsymbol{w}$, while the policy is generated directly from the value function. In contrast, the actor-critic method is another class of method that learns both policy and value approximations.

Specifically, the actor-critic method directly parameterizes the policy as $\pi(a|s, \boldsymbol{\theta}) = \mathbb{P}[A_t = a|S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}]$[2], with $\hat{v}(s, \boldsymbol{w}) \approx v_{\pi_{\boldsymbol{\theta}}}(s)$ kept for value function, which is also the core concept of policy gradient methods. The policy based actor selects the actions, and the value based critic criticizes the actions taken by the actor. Since the updates are performed based on the policy that the actor is currently following, the actor-critic method is on-policy, which is still based on TD learning. The update of $\boldsymbol{\theta}$ can be expressed as,

$$\boldsymbol{\theta_t} \leftarrow \boldsymbol{\theta_t} + \alpha[R_{t+1} + \gamma \hat{v}(S_{t+1}, \boldsymbol{w}) - \hat{v}(S_t, \boldsymbol{w})] \frac{\nabla_\theta \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})}, \tag{7}$$

which is close to the form of update equation (4) in Sarsa.

7) Multi-Agent RL [34]: Multi-agent RL is the integration of multi-agent systems with RL, which extends MDP framework to a Markov game for multi-agent systems [35]. Through the cooperation of individual RL agents, multi-agent RL is able to solve many complex tasks compared with single-agent RL we discussed above. Moreover, agents communicate with each other and interact with the environment, so that state transitions are controlled by the current state together with all agents' actions.

Specifically, with n agents in the system, the set of actions for agent i is defined as $\mathcal{A}^i$, leading to the joint action set for all agents as $\mathcal{A} = \mathcal{A}^1 \times \mathcal{A}^2 \times \cdots \times \mathcal{A}^n$. The state transition function and the reward function become $\mathcal{T}(s_{t+1}|s_t, a_t^1, \cdots, a_t^n)$ and $\mathcal{R}(s_t, a_t^1, \cdots, a_t^n)$, respectively.

[2]In this paper, we use $\pi_{\boldsymbol{\theta}}$ and $\pi(a|s, \boldsymbol{\theta})$ interchangeably to refer to the parameterized policy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3025365, IEEE Internet of Things Journal

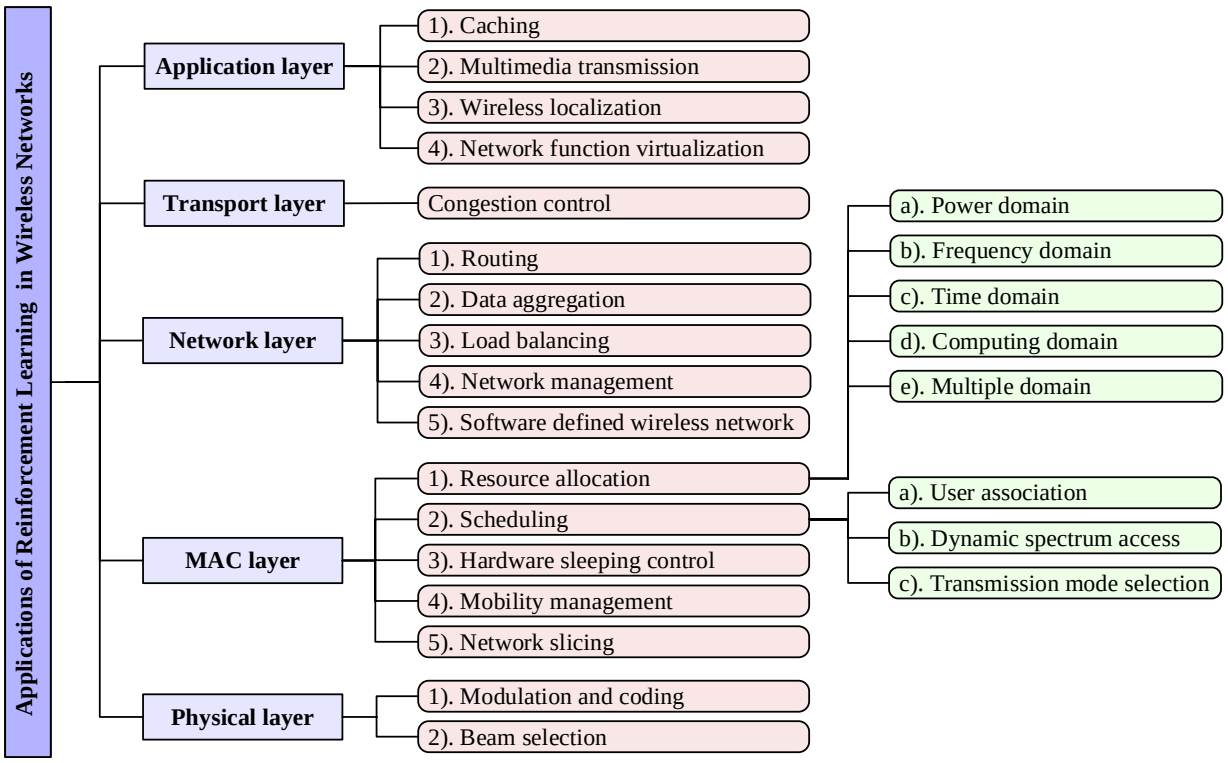


Fig. 3. Overview of reinforcement learning applications in wireless networks from a layering perspective.

Consequently, the above discussion covers several important and commonly used RL methods, including the hot DRL as well as multi-agent RL methods. Note that most of the other RL methods are extended/integrated from these methods above, which share similar algorithm ideas. Different RL methods can be considered for specific tasks and applications in wireless networks.

D. Evolution of RL and Driving Factors

Owning to the promising gains in decision making mentioned above, RL applications are exhibiting huge potential for growth recently. The datasets utilized in the RL framework including information concerning state space, action space and reward are gradually closing to the ground truth: from data randomly generated following certain distributions or models, to data obtained from simulators, then data collected from real world network environments. Though over 70% of the literature still utilizes randomly generated data in the RL framework till 2020, the number of publications utilizing real world data doubles year by year. Many factors are driving the application of RL in wireless networks.

The first key factor is that RL is naturally suitable for issues to be conquered in communication and networking research fields. This is because channel statuses and network statuses are continuously changing, which can be defined as states in RL framework. And changes can be formulated as the state transitions. Entities in wireless networks make decisions to respond to changes in the external environment, which can be described as actions. The achieved system metrics like latency and throughput can be seen as rewards.

The second driving factor is the ever-increasing difficulty and complexity of problems. By applying traditional optimization theory, game theory and queuing theory, explicit factors or elements involved in the formulated models need to be specified. However, when the scale and complexity of problems become intractable, it is almost impossible to describe all various factors and heterogeneous elements in traditional methods. On the other hand, RL takes advantage of such tough problems, since the framework only needs to specify the state space and action space, while the influence of other elements is reflected in the reward from environment. Hence, the inherent feature of RL technology is data-driven.

Another factor leading to the vigorousness of RL methods in wireless networks is the significant value lying in large scale of data, such as trajectory data, call detail record, deep packet inspector data and so on. Thanks to many researchers working on data anonymization, data cleaning and data publishing, data usability is guaranteed, which provides the foundation for applying RL methods.

IV. RL in Wireless Networks

To reveal the wide application of RL in wireless networks, this section gives a comprehensive review of RL applications from a layering perspective, as shown in Fig. 3. Specifically, hot topics and corresponding applications of RL in wireless networks are described layer by layer in the following.

A. RL for Physical Layer

Functionality blocks in the physical (PHY) layer are traditionally manually designed to optimize the radio performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3025365, IEEE Internet of Things Journal

TABLE II
SUMMARY OF RL APPLICATIONS IN PHY LAYER

Reference	Scenario	Addressed Problems	Learning Model	Control Scheme	Proof of Convergence	Performances
[36]	Wireless networks	Modulation/Demodulation	DRL	Cooperation	By simulation: 1000 steps	• Bit error rate: $10^{-3} - 10^{-2}$
[37]	WSN	Adaptive MCS	Actor-critic learning	Centralized	None	• Average throughput: 95% of optimal
[38]	Wireless networks	Adaptive MCS	POMDP	Centralized	None	• Transmission efficiency: 96% of optimal, 125% of P-timer
[39]	Cognitive HetNets	Adaptive MCS	DQN	Centralized	By simulation: 3000 steps	• Average rate: 83% of optimal, 187% of Benchmark
[40]	Indoor mmWave networks	Beam Selection	Q-learning	Centralized	None	• Beam search latency: 975 ms, 60.9% of Exhaustive search
[41]	mmWave cellular networks	Beam Selection	Double DQN	Centralized	By simulation: 3000 steps	-
[42]	mmWave cellular networks	Beam Selection	Policy gradient	Distributed	By theory	• 90% user SE: 9.8 bit/s/Hz, 127% of basic RL, 131% of Baseline
[43, 44]	mmWave vehicle networks	Beam Selection	MAB	Centralized	By theory	• Throughput: 98.27% of optimal, 90% of optimal within 56 min
[45]	mmWave vehicle networks	Beam Selection	DQN	Centralized	None	• Root Mean Square Error: 0.874, 98% of DP
[46]	mmWave UAV networks	Beam Selection	MAB	Centralized	By simulation: 40 iterations	• Expected regret: 15% of greedy
[47]	5G mmWave networks	Beam Selection	DQN	Centralized	By theory	• Sum-rate: 99% of upper bound

under various channel conditions. Approximating these functionality blocks with RL algorithms becomes a popular topic recently. A brief summary of literature is presented in Table II. As can be observed, these works mainly focus on modulation, coding and beam selection in mmWave networks

1) Modulation and Coding: Modulation function transforms digital bit streams into analogy signals which can be transferred over analog wireless channel, while the demodulation function is the inverse. A two-agent RL framework is developed in [36] to implement modulation/demodulation. As illustrated in Fig. 4, the transmitter of each agent is characterized by a DNN which maps the input bit string to a complex number, while the receiver runs k-nearest neighbors (kNN) on each complex number by comparing it with the modulated preamble, and generates a guess for each transmission.

On the other hand, adaptive modulation and coding scheme (MCS) is a widely adopted link adaption technique in the physical layer of wireless networks. Recently, RL has been applied to adaptive MCS from the perspective of optimal control.

For example, an actor-critic learning based adaptive MCS approach is proposed in [37] for both point-to-point communications and multi-node communications in WSN. The intelligent agent residing in the transmitter adjusts the modulation level and also the transmission power to adapt to the buffer size and channel state information (CSI), so as to maximize the long-term energy efficiency. Besides, Feres *et al*. [38] define a new finite-state Markov channel model to address the varying MCS. Specifically, based on POMDP formulation, such problem is solved under the robust header compression control by successive approximation with Markov process. Differently, Zhang *et al*. [39] investigate the adaptive MCS problem using DQN in cognitive heterogeneous networks (HetNets). The primary user (PU) learns the pattern of interference caused by the random spectrum occupation of secondary users (SUs), and accordingly chooses MCS level to improve data rate.

2) Beam Selection: The essential problem in mmWave massive multi-input-multi-output (MIMO) systems is beam selection, which is to choose the transmitting beam that is pointing to user equipments (UEs) and achieves the highest channel gain. Beam selection for single UE in indoor scenario is investigated in [40], where a multi-stage Q-learning approach is developed to determine the optimal beam from multiple BSs for the highest signal quality. Specifically, the Q-learning agent is first trained with an offline dataset. Then, the generated Q-table in offline stage is used to suggest the candidate beam sets for the online beam searching phase.

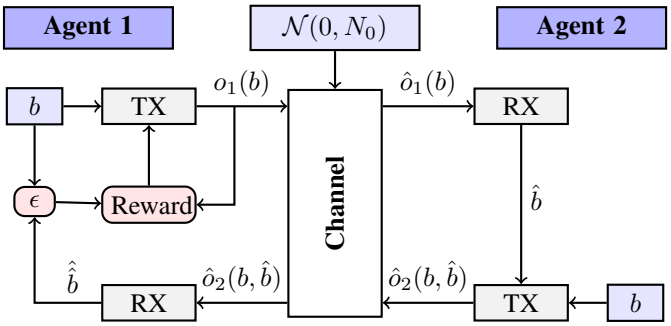


Fig. 4. Framework of two-agent RL for modulation/demodulation.

Shafin *et al*. [41] develop a double DQN based broadcast beam selection algorithm for CN. The agent at BS dynamically adjusts broadcast beam according to the quality of service (QoS) satisfaction of UEs, in order to improve the coverage.

On the other hand, beam selection problem for multiple UEs is studied in [42], where each mmWave BS interacts with each other and learns the best beamwidth and transmission power level in a distributed way, using multi-agent RL framework. Different from [42], where each mmWave BS is assumed to serve only one UE, more practical scenario where BS communicates with multiple UEs simultaneously through beamforming is considered in [43–47].

In [43], the BS simultaneously selects a subset of beams for multiple vehicles using MAB in order to fully exploit the multiplexing gain. Taking the context information including positions and trajectories of vehicles, [44] develops an MAB based online fast ML approach to maximize the amount of successfully transmitted data. Whereas [45] addresses the joint vehicle scheduling and beam selection problem using DQN. A cascade framework composed of two convolutional neural network (CNN) layers is proposed. The first layer takes position and height of vehicles as input and outputs the achievable signal strength of each beam pair for each vehicle. Then the results are used by the second layer to allocate time slots for each vehicle. Vaezy *et al*. [46] focus on unmanned aerial vehicle (UAV) mmWave networks and propose an MAB based beam selection algorithm. With the unknown dynamics in blockage of mmWave channels, the UAV learns to adjust beams to improve the QoS satisfaction of served UEs. Thompson sampling is adopted to find the optimal solution. Mismar *et al*. [47] exploited DQN for joint beamforming and power allocation problem in 5G mmWave networks. According to signal to interference plus noise ratio (SINR) in receiver, the agent determines the analog beams and

allocated power to different UEs.

3) Discussions on RL Applications in PHY Layer: Though RL has been successfully applied to handle several important problems in PHY layer, there is still room for improvement.

Agent cooperation in [36] would cause extra signaling overhead, and also bring latency in the state collection. Considering modulation/demodulation is performed at the symbol level, bad decision may be made. For instance, only 95% of optimal performance is achieved in [37]. Thus DL may be more appropriate for modulation/demodulation.

As for the beam selection, different observations are taken as the state variables in RL frameworks, such as CSI [42, 47, 48], locations [40, 43, 44], QoS satisfaction [41]. However, low signal to noise ratio (SNR) of mmWave link before beamforming makes it hard to acquire CSI. Location information which is application layer parameters may increase the latency of data collection. In addition, there may exist errors in these state variables due to the dynamical obstacle (as considered in [46]), low precision positioning and CSI estimation. In this regarding, robustness should be considered when designing RL algorithms. Another critical issue is learning delay, as the best beam is frequently changing (ms level). For instance, though the beam searching latency in [40] is reduced by 39.1% compared to the exhaustive search approach, it is still undesirable in the practical mmWave communication systems. On the other hand, both DRL and RL models are adopted for beam selection. However, considering the limited dimension of state and action space, the trade-off between data processing load and optimality should be carefully considered with respect to the network sizes, when designing the learning model. Moreover, results show that an up to $56min$ is needed to achieve the 90% of the optimal performance [40]. There may be a long time during which RL performs badly. Hence fasting the convergence of RL based approaches is highly desirable.

B. RL for Media Access Control Layer

The media access control (MAC) layer of wireless networks is responsible for resource allocation, scheduling, mobility management and interference coordination, etc. RL has long been used to address the control problems in MAC layer and shows the superior performance. We present an overview of RL applications in MAC layer in Table III.

1) Resource Allocation: Resource allocation is a fundamental functionality in MAC layer, the aim of which is to coordinate the resource sharing in multiple domains (power, frequency, time, computation, etc.) among users. Traditionally, resource allocation problems are usually modeled as complex multi-object optimization problems, which, however, are too complex to be solved exactly [4] and may not have closed-form solutions. Inspired by its capability of solving complex optimization problems efficiently, tailoring RL algorithms to better allocate resources, has attracted considerable research efforts. In the following, we discuss related works according to their focused resource domain.

a) Resource Allocation in Power Domain: Researches in this domain can be classified into power control of single transmitter [49–52] and power control of multi-transmitter [53, 54].

Meng *et al*. [49] investigate the power control problem in full-duplex CRN using DQN. The cognitive BS periodically detects the interference power from primary networks, and opportunistically communicates with SUs using appropriate power on the same channel. Mismar *et al*. [50] propose a RL based closed loop power control for voice over long term evolution (VoLTE) users. With the random occurrence of link impairments, the BS can autonomously adjust transmission power allocated to served users to meet their target SINR. Lee *et al*. [51] exploit DQN for user selection and power control in time divide duplex (TDD) multi-access networks. The BS determines the user to serve and also transmission power. The problem is formulated as a constrained MDP, and further transformed into an unconstrained MDP using Lagrangian approach. Particularly, CSI and Lagrangian approach are taken as the state, in order to make the algorithm circumstance-independent. Zhang *et al*. [52] consider the CRN consisting of a primary transmitter-receiver pair, a secondary transmitter-receiver pair and several sensors. Spectrum is shared between primary and secondary pairs. The secondary transmitter collects interference level of primary transmitter through sensors, and accordingly adjust its own transmission power to meet the QoS requirement. Two asynchronous variants of actor-critic learning, *i.e*., asynchronous advantage actor-critic (A3C) and distributed proximal policy optimization are proposed to solve this problem.

Compared with the focus on one transmitter of above works, power control of multi-transmitter studies how to adjust the transmission power of multiple transmitter-receiver pairs simultaneously to mitigate the interference. Most of RL applications in this field adopt the multi-agent framework. For instance, Nasir *et al*. [53] address the power control problem in mobile ad hoc networks (MANET) using multi-agent DQN. Each transmitter acts as an agent, which observes channel condition, location, experienced/generated interference from/to neighbors, and adjust transmission power level accordingly. Xiao *et al*. [54] proposed a multi-agent DRL based distributed power allocation scheme for small cell networks. With the local observations of user density, historical SINR and CSI estimations, each small cell independently determines the transmission power allocated to served users. Both energy consumption, cell throughput and caused interference are considered in the reward function.

b) Resource Allocation in Frequency Domain: RL has been adopted to make decisions of spectrum allocation [55–58], spectrum handoff [59] and hopping [60]. Most of the RL applications in this field adopt multi-agent learning framework, considering that decisions are usually made on the UE side in a distributed manner. The centralized learning framework is only adopted in [60].

Wang *et al*. [55] exploit RL for the spectrum allocation in Carrier Sense Multiple Access (CSMA) based CRN. The problem is formulated as a POMDP with unknown system dynamics. DQN is applied to overcome unknown dynamics and prohibitive computation therein, and achieve near-optimal channel access policy. On the contrary, Naparstek *et al*. [56]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3025365, IEEE Internet of Things Journal

TABLE III
SUMMARY OF RL APPLICATIONS IN MAC LAYER

Problem	Sub-Problem	Reference	Scenario	Learning Model	Proof of Convergence	Performances
Resource Allocation	power	[49]	Full duplex CRN	DQN	By simulation, 35 episodes	• Average Throughput: 85.7% of optimal
	power	[50]	LTE	RL	None	• Throughput for 5 UEs: 101% of Random algorithm
	power	[51]	TDMA networks	DQN	By theory	• Data rate: 90% of optimal
	power	[52]	CRN	A3C	By simulation, 200 steps	• Reward: 90% of optimization method, 120% of DQN
	power	[53]	MANET	DQN	By simulation, 30000 steps	• Sum rate: 105% of WMMSE, 114% of central
	power	[54]	Small cell networks	DRL	By theory	• Average throughput/energy consumption: 125%/66.7% of Bi-SON
	spectrum	[55]	CSMA CRN	POMDP, DQN	By simulation, 70 steps	• Average reward: DQN 0.947, Whittle Index 0.767
	spectrum	[56]	Multi-channel wireless networks	DQN	By simulation, 50 episodes	• Average channel utilization: 200% of Slotted Aloha • Average user rate: 140% of DP
	spectrum	[57]	LTE-LAA	DQN	By theory	• Average airtime allocation: 17% of existing proactive schemes
	spectrum	[58]	D2D Underlaying CN	Multi-agent actor-critic learning	By simulation, 100 steps	• Outage probability: 60% of Q-learning, 62% of DQN • Sum rate: 134% of Q-learning, 131% of DQN
	spectrum	[59]	CRN	Transfer actor-critic learning	By simulation, 30000 steps	• PSNR with 10 dB SNR: 133% of Myopic • MOS: 105% of Q-learning, 122% of Myopic
	spectrum	[60]	CRN	Q-learning	By simulation, 500 steps	• Utility: 103% of Q-learning, 121% of Random
	time	[61]	LTE-U WiFi	Q-learning	None	• Aggregate capacity: 115% of non-dynamic
	time	[62]	Delay-tolerant IoT	DQN	By simulation, 200 episodes	• Average utilization gain: 200% of Heuristic control • Average throughput: 114.7% of Original
	time	[63]	WSN	POMDP, Q-learning	By theory	• Normalized throughput: 125% of fully-connected network
	time	[64]	IRSA	Q-learning	By simulation, 10 steps	• Normalized throughput: 253% of original IRSA
	computation	[65]	MEC	DRL	By simulation, 200 steps	• Average cost: 32% of Greedy, 18% of Local
	computation	[66]	SDN enabled MEC	DQN	By simulation, 1000 episodes	• Average service time: 77.8% of classic algorithm
	computation	[67]	Energy harvesting MEC	POMDP, Policy gradient	By theory	• Average utility: 123% of Greedy, 160% of MEC, 250% of local
	computation	[68]	MEC	POMDP, Multi-agent DRL	By simulation, 8000 episodes	• Average utility: 128% of Greedy, 575% of Random
	computation	[69]	MEC	DQN	By simulation, 1000 steps	• Average cost: 95.8% greedy, 99.44% of Lookahead
	multi-domain	[70]	NOMA networks	POMDP, DQN	By simulation, 200 episodes	• Throughput: 243% of slotted ALOHA NOMA
	multi-domain	[71]	UAV networks	Multi-agent Q-learning	By theory	• Average reward: 250% of random scheme
	multi-domain	[72]	OFDMA networks	DRL	By simulation, 8000 steps	• Average reward: 250% of random
	multi-domain	[73]	Collaborative MEC	Deep MC	By simulation, 10000 steps	• Average latency: 41% of greedy search, 48% of DQN • Average energy consumption: 83% of greedy, 78% of DQN
	multi-domain	[74]	MEC based vehicular networks	DQN	By simulation, 25 episodes	• Utility: 104.4% of Q-learning, 117.1% of local
	multi-domain	[75]	MEC	DRL	By simulation, 8000 frames	• Maximal computation rate:122.5% of edge, 206.9% of local
	multi-domain	[76]	MEC based vehicular networks	DRL	By theory	• Energy consumption: 108% of optimal, 67% of conventional MEC
	multi-domain	[77]	Cache enabled MEC	DQN	None	• Average energy consumption: ĩ13% of exhaustive search
Scheduling	user association	[78]	Energy harvesting IoT	DQN	By simulation, 20000 steps	• Average sum rate: 141% of round-robin scheduling,96% of optimal
	user association	[79]	V2X	Q-learning, DQN		
	user association	[80]	mmWave HetNets	Multi-agent DQN	By simulation, 1000 steps	• Sum-rate: 108% of max-SNR, 104% of heuristic, 98% of optimal
	user association	[81, 82]	HetNets	Multi-agent dueling double DQN	By simulation, 75 episodes	• Smoothing system capacity: 114% of Q-learning, 107% of DQN
	user association	[83]	5G	Multi-agent RL	None	• Successful transmissions: 110% of sensing based, 149% of Q-learning
	dynamic spectrum access	[84]	Ambient backscatter system	Basic RL	By theory	• Average throughput: 136% of random policy, 91% of optimal
	dynamic spectrum access	[85]	Energy harvesting WSN	POMDP, Double DQN	By theory	• Maximal reward gain: 113% of DQN, up to 117% of Q-learning
	dynamic spectrum access	[86]	CRN	POMDP, DQN	By theory	• Decision accuracy: 114% of Q-learning, 200% of random policy
	dynamic spectrum access	[87]	Heterogeneous CRN	DQN	By simulation, 70000 steps	• Spectrum utilization: 130% of half-20step-DQN, 157% of half-DQN
	dynamic spectrum access	[88, 89]	CSMA HetNets	DQN	By theory	• Throughput: 132% of TDMA, 89% of ALOHA
	dynamic spectrum access	[90]	5G V2I	Multi-agent DRL	None	• Cost: 81% of DQN, 70% of Q-learning
	transmit mode selection	[91]	5G D2D	Multi-agent RL	By simulation, 35 episodes	• Data rate: 140% of greedy, 233% of random
	transmit mode selection	[92]	IoT	Multi-agent actor-critic learning	By simulation, 1500 iterations	• Blocking probability: 65% of RL-greedy, 78% of DQN • Throughput: 128% of RL-greedy, 109% of DQN
Hardware Sleeping Control	single-node	[93]	CN	DQN	None	• Sleeping gain: 101% of DQN, 349% of always-on policy
	multi-node	[94]	Cloud-RAN	DQN	By theory	• Total power consumption: 88% of FA
	multi-node	[95]	HetNets	Deep actor-critic learning	By simulation, 1000 steps	• Normalized cost: 72% of Q-learning, 79% of TACT
Mobility Management	BS dominant	[96]	HetNets	MAB	None	• Data rate: 174% of classical, 97% of satisfaction based • Handover failure: 36% of classical, 111% of satisfaction based
	user dominant	[97]	WLAN	DQN	By theory	• Average throughput: 131% of original
	user dominant	[98]	HetNets	Basic RL	By theory	• Throughput: 155% of SBH, 97% of RBH • Number of hand-offs: 55% of RBH, 67% of SBH, 103% of optimal
Network Slicing	content caching	[99]	RAN	DRL	By simulation, 40 episodes	• Average cache hit rate: 101% of First In First Out policy
	multi-domain resource allocation	[100]	RAN	DRL	By theory	• Average utility: 102% of Queue-aware control policy
	spectrum allocation	[101]	RAN	RL, DL	By simulation, 25 steps	• Resource utilization: 165% of heuristic, 112% of classical actor-critic
	spectrum allocation	[102]	SDN-RAN	Multi-agent DRL	By theory	• Spectrum efficiency: 400 bps/Hz; Satisfaction ratio: 99%
	spectrum allocation	[103]	SDN-RAN	Actor-critic learning, LSTM	By simulation, 4000 steps	• Utility: 110% of hard slicing
	slicing configuration	[104]	WiFi	Double DQN	By simulation, 10 steps	• Mean normalized throughput: 94% of optimal
	UAV cooperation	[103]	MEC	RL	By theory	• -

investigate the spectrum allocation in orthogonal frequency division multiple access (OFDMA) systems using dueling DQN [105]. The CSIs, transmission results (success/failure) of last action, and capacity of each frequency channel are jointly modeled as the state space. A long short-term memory (LSTM) layer is adopted to transform the state vector into a low-dimensional internal state. In addition, both competitive reward and cooperative reward are considered in the DQN model.

Spectrum allocation problems are addressed using RL in the context of CN in [57, 58]. Challita *et al*. [57] work on the proactive channel selection problem in long term evolution-license assisted access (LTE-LAA) networks, the aim of which is to guarantee fair spectrum sharing among small cell base stations (SBSs) and access points (APs). In particular, a non-cooperative game framework is developed, and an end-to-end deep LSTM learning based scheme is used to find a mixed strategy Nash equilibrium (NE). The encoder-decoder module first transforms the historical traffic sequence to a fixed dimension vector. Then a multi-layer perception encodes all the vectors together, while the decoder network further decodes the channel selection information from the vector. Differently, Li *et al*. [58] consider the Device-to-Device (D2D) underlayling CN and propose a multi-agent Deep actor-critic learning based spectrum allocation scheme. Each D2D pair chooses the resource block (RB) according to the CSI, interference and occupied RB in the previous time slot. The aim is to maximize the QoS satisfactions of both D2D and cellular UEs.

Koushik *et al*. [59] investigate the spectrum handoff problem

in CRN. Both Q-learning and transfer actor-critic learning are adopted to address this problem. Specifically, the new arrived SU first uses Q-learning to learn the optimal handoff policy if there is no expert SU. On the contrary, if there exists the expert SU, the newly joined SU would first try to find the expert SU using manifold learning, and then get the expert policy. The actor unifies the expert policy and generates native policy to select actions. Through knowledge transfer, the learning process of newly joined SU is greatly sped up.

Han *et al*. [60] exploit Q-learning for the frequency hopping problem in CRN. The SUs make the frequency hopping decisions and decide whether to stay associated with the current BS or move to another BS based on the presence of PUs and achieved SINR in the last time slot. The aim of this scheme is to improve the SINR of the SUs against cooperative smart jammers.

c) Resource Allocation in Time Domain: Time resource allocation usually raises between two different systems, such as LTE and WiFi coexisting systems in [61], and delay-tolerant IoT and real-time application integrated environment in [62]. How to share the time resources between heterogeneous systems is the key problem. Rupasinghe *et al*. [61] propose a multi-agent Q-learning based time sharing scheme. Each LAA BS learns to select the best duty cycle according to the achieved capacity. The cost function here is modeled as the gap between the target and achieved capacity. Chinchali *et al*. [62] leverage basic RL for the time sharing of delay-tolerant IoT traffic and conventional real-time applications. The aim is to maximize the IoT traffic throughput while guaranteeing QoS requirements of conventional real-time applications. The central agents observe load, quality and number of cell sessions, and accordingly determine the fraction of time when IoT data is scheduled on top of conventional traffic.

Time sharing among homogeneous transmitters is investigated in [63] using RL. Nisioti *et al*. [63] propose a decentralized coordinated RL based approach for WSN. Due to the lack of a centralized control point and partial observability of network states, the MAC design is formulated into POMDP framework, where each sensor node is an agent, and utilizes Q-learning to learn the transmitted packets in each time slot. The authors further extend the MAC design under the irregular repetition slotted ALOHA (IRSA) protocol in [64], and theoretically analyze the optimality and near-optimal polices of proposed design.

d) Resource Allocation in Computing Domain: Computation offloading is the main topic in this area. Due to the limited computation capacity, it is necessary for UEs to offload their computation tasks to the nearby edge servers. Therefore, which units and how much should be offloaded to remain a key problem.

Computation offloading in single UE and mobile edge computing (MEC) server case is studied in [65] using DRL. The agent is trained using policy gradient, and learns to choose the offloading number of processing units according to the CPU, memory occupation and requirements of tasks in the job slot and backlog. A DQN based offloading scheme is proposed to determine the target MEC server and also the volume of computations to be offloaded. The state contains radio bandwidths, harvested energy, and battery level. The aim is to minimize the weighted cost, including power consumption, delay, dropping probability and etc.

In contrast, [66–69] cope with the computation offloading problem in multi-user MEC systems. Wang *et al*. [66] adopts the centralized learning framework to address this problem. A DQN based routing selection and computation offloading scheme is developed for Software Defined Network (SDN)-enabled MEC system. The agent is deployed at central controller, observes the location of requests, and determines the routing and offloading action accordingly.

Both [67, 68] exploit distributed multi-agent learning framework for computation offloading. Tang *et al*. [67] focus on IoT MEC systems with energy harvesting. To determine the percent of tasks processed locally and remotely (at MEC servers), the decentralized computation offloading optimization problem is formulated as a POMDP problem with partially observed data and energy states. A policy gradient based algorithm is proposed to maximize the reward obtained by task execution while reducing the electricity cost. Computation offloading problem is also formulated as a POMDP in [68], which is addressed by multi-agent DRL with policy gradient and differential neural computers. Without the bandwidth and preference information of others, each user independently learns the optimal offloading policy so as to maximize its utility in terms of processing time and energy consumption.

On the other hand, Zeng *et al*. [69] address the virtual machine (VM) transferring problem in MEC systems using DQN. VM, which takes charge of service data processing, can be flexibly transferred among the edge servers of different BSs. With the aim of minimizing overall cost, the agent dynamically transfers the VMs according to their previous location and user requests.

e) Resource Allocation in Multiple Domain: In order to improve resource utilization, RL has also been adopted for multi-domain resource allocation, and shows significant performance improvement.

Joint power and frequency resource allocation is investigated in [70, 71]. Zhang *et al*. [70] investigate such problem in grant-free non-orthogonal multiple access (NOMA) networks, where long-term throughput maximization is formulated as a POMDP. With the power fading gain of links as observations, the authors propose a DQN-based algorithm for solution and LSTM is utilized to learn network contention states. Whereas Cui *et al*. [71] solve this problem together with UE selection in UAV communication networks using multi-agent RL. It is formulate as a stochastic game, and then solved by a multi-agent Q-learning, where each UAV acts as an agent and makes decisions according to received SINR level.

Joint time and frequency resource allocation problem in OFDMA networks is addressed using DRL in [72]. In this work, the AP learns to orthogonality allocate resource units (minimum allocation unit in time-frequency domain) of a frame to mobile stations, according to condition of channel and traffic request.

On the other hand, RL is applied in joint computation and communication resource allocation in [73–75]. Chen *et al*. [73] focus on the collaborative MEC based IoT and address

this problem using deep MC tree search algorithms. Moreover, DNN is utilized in the learning framework to predict actions in a self-supervised learning manner. Liu *et al*. [74] consider MEC vehicular networks and use Q-learning and DQN to address this problem. In this work, vehicles are assumed to be moving edge server, since static deployment of edge servers may cause "service hole". The agent located in the central controller determines servers to offload, as well as communication and computation resources to allocate for each UE. DRL is adopted to solve this problem for wireless powered MEC systems in [75]. A DRL based online offloading algorithm, termed as DROO is proposed. The agent determines the proportion of time used for wireless power transfer and offloading, as well as the binary offloading decision, *i.e*., either computing locally or at server.

Different from previous researches, [76, 77] investigate the joint frequency, computation and storing resource allocation problem. Peng *et al*. [76] focus on MEC based vehicular networks and propose a DRL based approach, in which MEC server collects the information about moving, position, and task, and makes the allocation decisions in a centralized manner using the deep deterministic policy gradient (DDPG) learning. In [77], the problem in cache enabled MEC systems is first formulated as a mixed integer nonlinear programming (MINLP), and then solved by a MDP based framework. Considering the high dimension of state and action space, DRL is applied in the framework.

f) Discussions on RL Applications in Resource Allocation: Resource allocation is a well-studied problem in wireless networks, and is traditionally addressed by mathematical tools such as integer programming and convex optimization. Different from traditional schemes which focus on the utility maximization in a single channel coherence time (CCT), the RL based schemes are able to maximize the reward in the long-run through trial-and-error.

Existing RL based resource allocation schemes have considered various state information of wireless networks, including raw observations such as CSI [56, 59], QoS priority [59], currently accessed channel [56], the inferred information such as interference power [49, 60], and also the results of last action [61]. Raw observations are easier to acquire, and have higher efficiency in data collection and processing. For instance, CSI are estimated every frame employing channel reciprocity in TDD CN. Hence they are widely taken as the input of conventional handcrafted algorithms. Inferred information cuts down redundancy in the raw observations, at the cost of increased data processing overhead, and has the advantage in training efficiency. Whereas the results of last action requiring extra storing resource. Overall, these are all helpful parameters of network dynamics for finding an optimal policy. Nevertheless, how to construct state space to achieve the balance between accuracy and data processing/storing/acquiring/training efficiency is still an open issue.

Moreover, Table III shows that most works in this field adopt DRL models. Considering the large number of resource elements, combining DL and RL has advantage in accuracy and optimality. However, DRL results in higher data processing load, and thus may not appropriate for some scenarios where learners have limited battery and processing ability, such as MANET [53] and UAVs [71].

In addition, there are various metrics to evaluate the performance of resource allocation, such as throughput, spectral efficiency, buffering cost, energy consumption, energy efficiency, quality of experience (QoE), fairness, and delay reduction. Since existing RL applications in this field only consider a small part of these objectives at the same time, the benefits of RL in multi-objective optimization are not fully exploited.

On the other hand, both centralized and decentralized frameworks are adopted in these RL applications. Learning the allocation policy of multiple nodes through a centralized agent possesses advantages in convergence rate and optimality. However, the timely collection of state information from multiple distributed nodes requires additional signaling and would result in high overhead. Whereas distributed learning framework has advantages in complexity and overhead. However, it would decrease the convergence speed. As can be observed from Table III, though some RL applications such as [57] using non-cooperative game theory to study the convergence performance of distributive learning, a considerable number of works only prove the convergence by simulation. Results show that a long time is required to converge to the optimal policy. For instance, 25,000 time slots are consumed to converge for the DQN in [53]. In this regard, a cooperative RL framework as done in [59] seems to be a promising solution, where only part of important information about environment is exchanged among the distributed agents. How to choose the information to be exchanged in order to balance the convergence rate, overhead and optimality is still an open issue.

Another critical issue of RL based resource allocation approaches is the online bad decisions, especially at the beginning when convergence is not reached. Online bad decisions are unacceptable for some ultra reliable services such as vehicular communications. The occurrence of online bad decisions leads to the performance gap between RL based and the optimal one. For instance, only 85.7% and 90% of optimal performance are achieved by RL applications in [49] and [51], respectively. Pre-training RL agents with traditionally handcrafted algorithms seems to be a potential solution to avoid the bad decisions at the beginning. However, how to deal with random online bad decisions of converged RL to approximate the optimal performance is still untouched in the context of wireless networks.

Furthermore, computing from cloud to edge is an increasing trend in nowadays wireless networks. RL has been successfully applied to build the intelligent edge in [65–69, 73–77], *i.e*., offloading computing tasks, allocation computing and communication resources in a data-driven and adaptive way. However, computation capability of UEs is usually limited and may be not able to make decisions timely. For example, the total execution latency of DQN based approach in [53] is about $0.059s$, so it may be impractical considering the short resource allocation period, such as $20ms$ considered in [37, 50]. Besides, centralized learning would cause overhead and latency in data collection. In this regarding, how to dynamically compromise centralized, edge and UE learning, and build the edge intelligence employing MEC architecture,

is a promising topic.

2) Scheduling: Scheduling is another key functionalities in the MAC layer, which is responsible for the provision of guaranteed QoS in terms of delay, packet dropping rate (PDR), throughput and so on. RL applications in this field mainly lie in user association [78–83], dynamic spectrum access [84–90] and transmission mode selection [91, 92].

a) User Association: User association policy determines which BS serves what and how many users. Traditionally, user association is decided based on the reference signal receiving power (RSRP). However, this kind of scheme is known limited in satisfying long term QoS demands when facing dynamic and unpredictable network conditions.

Chu *et al*. [78] study the user association problems in energy harvesting IoT, where UEs with various battery levels communicate with one BS. At each time slots, BS decides which UEs can access the available channels. An LSTM DQN integrated framework is developed to address this problem. The deep LSTM is trained to predict the current battery level of candidate UEs based on the history battery levels. Then the prediction results together with CSI are taken as the input of the model, which finally outputs the user association policy. Zhao *et al*. [79] propose to use multi-agent DQN to address this problem in HetNets, where each UE makes the local decision and sends the access requests to BSs. The feedback from BSs is used to evaluate the reward and update the DQN parameters. Similar learning framework is applied in [80] to maximize the overall throughput via optimizing user association in mmWave HetNets. At each time step, the user autonomously selects one of its surrounding BSs to send connection request, according to local observations. The macrocell base station (MBS) broadcasts the resulted network throughput to all users to help them evaluate the goodness of their actions.

On the other hand, it is necessary to jointly optimize user association and resource allocation to maximize resource utilization. For instance, the joint optimization problem is investigated in HetNets [81, 82]. Varying channel conditions and interference may prevent UEs from meeting its minimum QoS. In this regarding, UEs need to dynamically choose the BS to associate and also the channels to communicate. This problem is formulated as a stochastic game and the optimal policy is found by multi-agent dueling double DQN. Yao *et al*. [83] apply RL to address the user association and channel access problem in 5G network. The problem is formulated as a potential game, and multi-agent RL framework is proposed to learn the good association and channel access probability action for UEs. Both regret learning and the fictitious play based algorithm are adopted to update the UEs' policy.

b) Dynamic Spectrum Access: Huynh *et al*. [84] leverage RL for the spectrum access scheduling problem in RF-powered ambient backscatter communication systems. The agent at SU transmitter learns to make optimal decisions, *i.e*., staying idle, sending data, harvesting energy, or backscattering data, according to the status of channel and energy, as well as data demands.

Li *et al*. [85] consider the spectrum access scheduling problem in energy harvesting WSN. Since it is costly and impractical to observe the power information of all nodes, only a part of nodes is assumed to be observed therein. Therefore, the authors formulate a POMDP problem with partial information as belief states, and further propose a double DQN scheme to obtain the scheduling policy for spectrum access. The energy cost of unobserved nodes is avoided by the policy inference in POMDP.

Li *et al*. [86] investigate the dynamic spectrum access in CRN. Generally, for the bandwidth requirement, SUs need to sense states of all channels and aggregate vacant channels among them. However, due to the limited aggregation capability, only vacant channels within the aggregation range can be utilized by the user, *i.e*., the whole system is partially observable to the user, falling into a general POMDP. Moreover, a DQN-based algorithm is proposed to leverage channel correlation for system state inference. Xu *et al*. [87] consider the distributed spectrum access problem in heterogeneous CRN. The considered heterogeneous cognitive mesh network is composed of nodes that adopt different spectrum access schemes. Secondary nodes observe the channel occupation, and determine whether to transmit on certain channel or stop transmission independently. This problem is formulated as a POMDP and solved using deep recurrent Q-Network. Yu *et al*. [88, 89] apply DQN in spectrum sharing in CSMA based HetNets. Multiple nodes with different CSMA MAC communicate on same spectrum. DQN based CSMA, termed as CS-DLMA, is proposed to help node maximize long-term utility in such heterogeneous environment. The agent dynamically changes the length of packets to be transmitted according to the sensing results and packet length in the previous time slot. Note that the epoch is non-uniform due to the heterogeneous CSMA.

Ning *et al*. [90] apply DRL to handle the channel selection problem in 5G vehicle-to-infrastructure (V2I). A distributed DRL framework is proposed, where each V2I user chooses its channel for data transmission independently, according to the experienced interference.

c) Transmission Mode Selection: RL is exploited for transmission mode selection in D2D assisted 5G HetNets in [91]. Each virtual reality (VR) user can stay in three modes, *i.e*., macrocell broadcasting, mmWave small cell unicasting and D2D multicasting. In order to maximize system throughput in varying wireless environment, users need to dynamically change the transmission mode. This problem is formulated as a general-sum stochastic games, and then solved by a multi-agent framework, where both Nash-Q-learning and Wolf-PHC algorithms are used to handle different network scales. Multi-agent actor-critic learning is exploited for mode selection of IoT devices in [92]. Each IoT device is modeled as an agent and dynamically chooses to sleep or be active, access or wait at the selected channel or handoff, and also the MCS level, according to the channel condition and traffic load.

d) Discussions on RL Applications in Scheduling: From above review, it can be observed that RL has been widely implemented to obtain the optimal scheduling policy. However, there is still room for improvement in this research field.

Considering the mobility of UEs, user association should be

determined from a global view, in order to achieve both load balancing and utility maximization. However, the overhead for state information collection and data processing would be a serious problem with the trend of network densification. Combining the LSTM and RL to exploit data correlation and improve data efficiency in [78] is a more desirable method. Nevertheless, the communication demand should also be considered.

Moreover, multi-agent RL is widely adopted in addressing user association [81, 82], dynamic spectrum access [90] and transmission mode selection [91, 92]. Though self-organization and state collection efficiency are achieved, the convergence and optimality become a critic issue, especially in dense wireless networks. Moreover, the data processing load and decision latency caused by DNN such as done in [79, 80] should be taken into account, since UE generally has lower data processing ability and is battery constrained.

3) Hardware Sleeping Control: Hardware sleeping is deemed as the most efficient way to cut down the energy consumed by wireless networks [106]. Most of conventional sleeping control approaches are predicative, *i.e*., predict the coming traffic and then activate/deactivate hardware. However, it is hard to predict accurate traffic variation. On the other hand, it is necessary to consider the long term performance when designing sleeping control scheme, since frequent hardware sleeping/activating transition would lead to extra energy consumption and QoS dissatisfaction. Considering the above deficiency, RL is desirable to solve hardware sleeping control problems.

Whereas Liu *et al*. [93] address the single-node sleeping control problem using DQN. A model-assisted scheme termed as DeepNap is proposed to learn the optimal sleeping policies. The traffic belief state is predicted through a traffic modeling module, where the Baum-Welch algorithm is used to fit an Interrupted Poisson Process (IPP) using real traffic data. The state space is modeled as a combination of predicted traffic belief state, queue length, and the operating status of the BS. According to the observed state, the DQN agent chooses to turn on or off the BS at each time slot. The reward function is a weighted sum of the positive reward for served request, the negative penalty for queued, re-transmitted, or failed requests, and also the status transition cost of the BS.

Xu *et al*. [107] address the multi-node sleeping problem for cloud radio access network (Cloud-RAN) using DQN, the goal of which is to minimize the energy consumption while guaranteeing the QoS requirements. At each time slot, the DQN agent selects the working status, *i.e*., active or sleeping, and also beamforming weights of remote radio heads (RRHs), according to the results of last action and the QoS demands. In order to reduce the dimension of action space, a two-step scheme is proposed, in which the DQN agent first determines the working status of RRH, then calculates the optimal beamforming weights using convex optimization with derived active set of RRHs.

Ye *et al*. [95] address the multi-node sleeping control problem in two-tier HetNets using deep actor-critic learning, in which DNN is first used to predict the traffic arrival rate of each SBS. With the rate obtained as environment space, the sleeping control policy of SBSs comes up with deep actor-critic learning algorithms.

Discussions on RL Applications in Hardware Sleeping Control: RL has been successfully applied in hardware sleeping control and shows superior long-term performance. Different state parameters, such as working status [93, 95], buffer status [93], traffic loads [93, 95], and QoS requirement [107] are considered in these RL applications. These parameters are all raw data available at transmitter and cause negligible overhead and latency in data collection/processing. Hence, to have an accurate modeling of the environment, all of these parameters shall be jointly considered. Furthermore, coordination among nodes is helpful for the sleeping control with respect to optimality [108], but unfortunately, it is only adopted in [95, 107]. Considering that centralized learning of the policies for multiple nodes would cause significant overhead and latency in data collection and processing, transferring the learned knowledge among multiple nodes would be a more efficient way to realize coordination. In addition, since the wireless traffic is dramatically changing in both time and spatial domains, using DNN and time-related LSTM to improve the learning efficiency of RL is a promising scheme. On the other hand, most of the existing RL applications use the randomly generated data to train the model, which makes their performance questionable in real wireless networks. Thus, the usage of real data for RL applications in wireless networks should be further encouraged.

4) Mobility Management: Mobility management is essential to provide seamless connectivity to mobile UEs. RL enabled mobility management has attracted considerable research interests recently. Both [96–98] have used RL to design smart handover schemes. Some of them focus on the BS-dominant handover [96], where BSs decide UEs' handover, while the others focus on the user-dominant handover [97, 98], where users determine the BS to handover. These schemes also differ from each other in scenario, optimization objective, and handover trigger condition.

Simsek *et al*. [96] study the BS-dominant handover problem in two-tier HetNets. Each BS learns its optimal biased RSRP value via MAB methods. Then UEs who fulfill the handover condition are handed over to other BSs. After that, RB based scheduling is performed.

Differently, Sun *et al*. [98] investigate the user-dominant handover problem for mmWave HetNets. An RL based scheme is proposed, where UE learns to select the target BS based on the observed CSI to avoid unnecessary handovers, and also maximize the volume of transmitted data between two handovers.

In comparison, Cao *et al*. [97] address the user-dominant handover problem for wireless local area network (WLAN) using DQN. A hybrid framework is proposed to help the user make handover decisions in order to avoid ping-pong effect and improve the long-term throughput. This hybrid framework first uses the CNN and recurrent neural network (RNN) to abstract potential features such as the location information, from the received signal strength indication (RSSI), which are later taken as the input of DQN for handover policy.

Discussions on RL Applications in Mobility Management:

From above review, we can see that RL has been successfully applied to mobility management. BS-dominant handover policy based on MAB [96] can determine the handover policy from the global view and strike a balance between the system utility and users' QoS. However, the handover decision is not directly made by the agent, but is obtained from the results of the selected BS, which would increase the decision latency and cause optimality degradation. As for UE-dominant handover studied in [97, 98], various state parameters including cell load, QoS demands, and resource availability are considered. From the point of optimal controlling, they shall be jointly considered in handover. However, some parameters such as cell load and resource availability, which are only available at BS, are hard to acquire for UEs. In this case, information exchanging between BS and UEs is required, at the cost of overhead and latency of data collection. Moreover, only the selfish rewards are adopted in the RL model, such as avoiding unnecessary handover, and maximizing transferred data. Nevertheless, the utility of the whole network should also be considered in the reward function for load balancing as well as global optimality.

5) Network Slicing: Network slicing [109], a promising direction for future wireless networks, divides a mobile network into multiple slices, and tailors each slice to the needs of various services. Especially, under the network slicing scenario, several works discuss RL application for resource slicing problems in MAC layer.

For example, Xiang *et al*. [99] formulate RAN slicing as a joint optimization problem with content caching and mode selection. Especially, to deal with various user demands and limited resources, the authors propose a DRL algorithm, where the cloud server makes decision for maximum reward. In comparison, Chen *et al*. [100] considers RAN slicing where multiple service providers behave selfishly to maximize the long-term payoff from the competition of communication and computation functionalities. Such problem is modeled as a stochastic game, and an online scheme based on DRL is derived to approach the optimal abstract control policies for NE. Yan *et al*. [101] investigate a collaborative learning framework for resource scheduling in RAN slicing, where DL is used to perform large time-scale resource allocation, while RL is used to perform online resource scheduling with small time-scale network dynamics. Such conjunction guarantees performance isolation between slices with fast convergence and high utilization.

Different from above works, Hua *et al*. [102] present an SDN-based system in RAN scenario, which contains several slices with sharing physical resources across BSs. The authors model the varying service demands as state, the allocated resources as actions, and further leverage DRL for solution. Especially, a generative adversarial network-powered deep distributional Q-network is proposed to overcome the annoying randomness and noise in communication. The authors further incorporate user mobility in resource management of RAN slicing in [103]. The actor-critic algorithm is utilized for varying service demands, and LSTM is integrated to track the user mobility and improve system utility.

The network slicing is also studied in WiFi scenario [104] and UAV scenario [110]. Specifically, Bast *et al*. [104] explore fast-learning DRL for dynamic slicing configuration in unplanned WiFi networks, in which double DQNs are utilized to enhance the convergence speed and stability. Faraci *et al*. [110] extend a 5G network slice with a fleet of UAVs, each providing computing facilities. Based on RL framework, the system controller keeps a cooperation between UAVs for power consumption minimization with job loss probability and queueing delay considered.

Discussions on RL Applications in Network Slicing: As described above, network slicing divides the physical network into multiple virtual networks (slices), where RL is leveraged for dynamic creation and configuration. Among them, resource allocation in RAN slicing is mostly investigated, and the action space in RL is carefully designed to allocate specific resources. For example, the action space is composed by the spectrum resource in [101–103], power resource in [104], computation resource in [100, 110], as well as caching resource in [99]. Compared with the researched scenarios, the real RAN slicing environment is much more complex and dynamic, which should be paid further attention with intelligent and generalized solution proposed. On the other hand, traditional network slicing implementation is categorized into three types of core network-only slicing, RAN-only slicing as well as core-RAN slicing [101]. However, most existing works focus on RL applications in RAN-only slicing, and it is still uncertain for the applications in other two types.

C. RL for Network Layer

Environment dynamics can greatly influence routing, network management and load balancing decisions in wireless networks. Hence, designing context-aware and intelligent decision-making schemes using RL has attracted a lot of research attention, which are discussed point by point in the following.

1) Routing: A wide range of works have adopted RL to design adaptive routing policies with different objectives in dynamically changing networks. Table IV presents a brief comparison of related works.

Energy consumption and delay aware routing is investigated using RL for underwater acoustic sensor network (UASN) [111, 112]. Jin *et al*. [111] propose a Q-learning based routing protocol called RCAR. Once a node has packets to send, it observes status of buffer and battery, as well as the location of neighboring nodes, and perform RCAR to select the next forwarder. RCAR can reduce delay and energy consumption meanwhile avoiding congestion. Similar scenario is considered in [112], where a distributed RL based CARMA protocol is proposed to help node choose next hop according to the number of unsuccessful transmission.

Besides the aforementioned works, RL based routing for throughput and connectivity optimization in MANET is investigated in [113]. The inherent partial observability of both routing and mobility is formulated as POMDP, where RL is adopted to control forwarding decisions and node mobility. Specifically, Q-routing based solution is proposed with the aim to increase the packet transmission rate.

TABLE IV
SUMMARY OF RL APPLICATIONS IN NETWORK LAYER

Domain	Reference	Scenario	Learning Model	Proof of Convergence	Performances
Routing	[111]	UASN	Q-learning	By simulation, after sending 50 packets	• Energy consumption: 86% of QELAR; Average delay: 27% of QELAR
	[112]	UASN	Multi-agent RL	By theory	• Average latency:7% of CARP, 76% of EFlood • Throughput: 128% of CARP, 187% of EFlood
	[113]	MANET	Q-learning	None	• Success rate: 104.5% of Hand-coded, 95.8% of Central controller, 109.5% of Centroidal
	[114]	MANET	MAB	By theory	• Success rate : 80%
	[115]	WSN	Least Squares Policy Iteration	By simulation, 40 episodes	• Minimum success rate: Q-learning: 0% Proposed: 90%
Data Aggregation	[116]	WSN	SMDP, Q-learning	By theory	• Energy cost: 110% of EXPL, 60% of on-demand scheme • Average degree of aggregation: 88% of EXPL, 165% of on-demand scheme
	[117]	VANET	Multi-agent Q-learning	By simulation, after 4km	• Number of observed reports: 88% of Non-learning, 71% of discrete-Q-table
Load Balancing	[118]	Multi-sink WSN	Q-learning	None	• Average hop: 52% of round-robin; • Average network lifetime: 93% of round-robin
	[119]	UASN	Q-learning	By simulation, after sending 50 packets	• Network lifetime: 109%-213% of Vector-based-forwarding
	[120]	SDN networks	RL	None	• Average delay: 77% of round robin; Average utilization: 120% of round robin
Network Management	[121]	SON	Stochastic MAB	By simulation, 10000 steps	• Average rewards: almost 100% of optimal, 107- 132% of default
	[122]	SON	MAB	By simulation, 3000 steps	• Perceived reward: 103% of cognitive policy
	[123]	HetNets	RL	None	-
	[124]	5G	Multi-agent RL	By simulation, 200 minutes	-
SDWN	[125]	SDN networks	MDP, DRL	By simulation, 200 steps	• Average request latency: 102% of optimal, 83% of greedy
	[126]	WLAN	MAB	By simulation, 1000 steps	• Average page load time: 60% of no control
	[127]	IoT	DRL	By simulation, 60 episodes	• Average packet delivery ratio : 102%-111% of OSPF
	[128]	MEC	Q-learning	By theory	• Average delay: 25% of random method, 33% of nearest offloading method

To achieve a longer network lifetime as well as lower energy consumption, the mapping of routing into POMDP is presented in [114]. In order to approximate the optimal node behavior, the decision problem is further reduced to a MAB problem. In addition, a stochastic learning algorithm is proposed to optimize the learning rate.

Wang *et al.* [115] propose a routing scheme that aims at multiple objectives including route path length, load balancing, link reliability and aggregation. Least squares policy iteration (LSPI) technique is adopted in this scheme, which uses parametric functions to approximate the Q-values of policies and has the advantage in data efficiency and insensitivity to the initialization.

2) Data Aggregation: Data aggregation in mobile sensor networks requires to design adaptive and distributed algorithms to capture the dynamics of moving sensors, sink nodes as well as gateways.

Ye *et al.* [116] propose a stochastic decision framework to investigate the fundamental trade-off between energy consumption and transmission delay in case of distributed data aggregation in WSN. The authors formulate the decision problem as a semi-Markov decision process (SMDP), by modeling the state space as the collected data samples. The action set includes two actions: 1) send the queued data immediately, or 2) wait to collect more data, while the latter could save energy at the cost of increased delay. Two algorithms based on adaptive real-time dynamic programming and real-time Q-learning are proposed and compared. Yu *et al.* [117] investigate the delay controlling problem for data aggregation in vehicular ad hoc network (VANET) using distributed RL. A novel scheme termed as CatchUp is proposed, where vehicles adaptively adjust their forwarding speed based on learning results, and in turn, improve the chances of aggregating nearby reports.

3) Load Balancing: Load balancing is important to prevent over or under utilization of a certain network entity which in turn ensures the connectivity of the network.

In a multi-sink WSN, following the commonly devised nearest-sink strategy when an event occurs, would exhaust the energy of the sink node early. As a consequence, that sink node would be isolated and numerous routing paths will be broken. Pertinent to this issue, a Q-learning based adaptive zone partition (QAZP) scheme is proposed in [118]. The state is characterized by the available energy of hotspots and the position of the centralized mobile anchor responsible for network partitioning. QAZP scheme adaptively partitions the network into zones, each of which is associated with a sink node. Sink nodes dynamically adjust the zone size based on the available energy of the nearby hotspots.

Hu *et al.* [119] propose an adaptive and energy efficient routing protocol, QELAR, for UASN. QELAR protocol exploits Q-learning to learn the optimal forwarding decision to balance the workload among nodes. QELAR considers the available energy and also the energy distribution in a group of nodes.

Huong *et al.* [120] propose an RLLR algorithm for load balancing in SDN networks. RLLR adopted RL as the decision framework. The agent at SDN controller computes the load balancing weights for the candidate shortest paths of every two nodes, according to traffic information.

4) Network Management: In the wireless networks, various tasks are automated by the network operators through SON functions including load balancing, mobility robustness optimization and so on. However, the optimal configuration of each instance of a particular SON function in case of simultaneous deployment requires orchestration and coordination. Furthermore, the SON functions are realized as a black box by the designers who are RAN vendors, which makes this task more complex. On the other hand, RL provides a possible solution that enables an agent to adapt to the network dynamics and learn the optimal configuration, which has been widely used to address the network management problem.

Daher *et al.* [121] introduce a policy based SON manager (PBSM) empowered with learning capability to configure SON functions with the aim to meet the operator objectives. The optimal mapping from the SON configurations to the performance indicators are first learned by PBSM. A stochastic MAB learning algorithm, named as the UCB1 is then proposed to maximize the reward function representing the operator's objectives.

Based on a different MAB algorithm named as LinUCB, an alternative learning approach for C-PBSM to achieve better convergence speed at the cost of sub-optimal solution, is

presented in [122]. LinUCB reduces the time for exploring the action space by taking advantage of the SON configuration value structure. Specifically, linear rewards are considered corresponding to their feature vectors associated with actions, and per iteration only one action is tested to evaluate the consequences of different actions. Simulations show that LinUCB has a much faster convergence speed.

A similar problem is considered by Iacoboaiea *et al.* [123] for HetNets, and RL-based SON coordinator (SONCO) approach is provided. Specifically, three SON functions are considered, including mobility robustness optimization, cell range expansion and eICIC. The SONCO makes the accept or decline decisions for the parameter update request from SON functions with the aim to minimize the maximum regret among all requests. The proposed RL framework encompasses a multi-dimensional regret vector and uses linear function approximation to deal with the inherent complexity for large scale problems.

Preciado *et al*. [124] investigate the dynamical SON function configuration in 5G networks. A multi-agent RL framework is adopted, where every SON function is considered as an agent, and they may conflict with each other. Two typical SON functions are considered, which are Mobility Load Balancing (MLB) and Mobility Robustness Optimization (MRO). The MLB agent observes proportion of edge users and cell load, and accordingly sets the cell individual offset to improve the capacity and coverage. Whereas MRO agent takes user speed as state and determines the best handover parameters.

5) Software Defined Wireless Network: Similar to the idea of SDN[129], software defined wireless network (SDWN) [130] decouples the control layer and the data layer, providing both low-latency and low-energy consumption services. Several related works with RL on this technology have been proposed for network layer problems.

For instance, Zhang *et al*. [125] focus on multi-controller SDWN scenario, and formulate the controller synchronization problem therein as MDP, which is further solved by DRL-based algorithm. Considering the low latency brought by SDN, Moura *et al*. [126] explore SDWNs for Web QoE, and formulate the system as a MAB framework. Two closed control loops are proposed for changing the wireless channel and the transmission power, and the UCB algorithm is utilized for solution.

On the other hand, Guo *et al*. [127] investigate the routing problem in SDN based IoT networks. The authors propose a DRL based QoS-aware security routing protocol, where the flow table and channel occupancy rate are the state space, the switch assignment composes the action space, and packet loss rate as well as processing delay are computed as the reward.

Baek *et al*. [128] investigate the load balancing problem in SDN-based Fog/MEC networks, where SDN Fog controller manages network resources, and SDN Fog nodes serve user requests. Based on Q-learning, the latency minimization problem is formulated into RL based decision-making process, where the number of tasks to be allocated is viewed as the state, and the offloading neighbour nodes as well as offloading tasks are determined by the action.

6) Discussions on RL Applications in Network Layer: According to the review above, RL is a promising technique for network layer problems especially the routing problems.

However, RL still faces various challenges in routing problems, such as the dynamic topology due to the mobility of nodes in MANET, the limited battery and processing capability in WSN. Dynamic topologies pose great challenges on convergence and decision accuracy. Since the neighbours are dynamically changing, an optimal routing policy is difficult to obtain and may require a very long time. How to efficiently transfer the learned knowledge to the new cases is an interesting question. Moreover, most of RL applications in this field adopt the distributed learning framework which possesses advantage in data collection/processing and time effectiveness, but a global optimal routing is not guaranteed. Considering the limited learning capability, cooperative learning among neighbors seems to be a promising solution to achieve global optimality. Another possible solution is SDWN, whose controller-infrastructure structure kindly matches RL framework, and SDN controllers can be viewed as agents to balance centralized control, scalability, and reliability requirement. However, the trade-off still needs to be carefully considered.

As for the works on data aggregation, most are focus on the static topology [116, 117], far from the real dynamic network environment. On the other hand, as most works only considered network lifetime, the delay factor is also very important for robust performance. Moreover, some state parameters such as the residual energy considered in [118, 119] are hard to acquire, which requires continually information exchanging and causes overhead and latency.

Another essential issue is the lack of the performance evaluation of RL-based schemes in the more realistic wireless network platforms. Most of existing schemes above have only been evaluated by simulation, without the validation of correctness and feasibility in real. Especially, the scalability of MAB algorithm in [121, 122] might be the main change when facing the complexity of the real network topology.

D. RL for Transport Layer

The key function of the transport layer is to provide congestion control in networks. The traditional transmission control protocol (TCP) protocols like TCP Reno, are rule-based and are suitable for some specific scenarios. However, the existing congestion control standards cannot adapt to network dynamics, and the inherent inflexibility makes them unsuitable for a wide range of network scenarios in the next generation networks. To overcome such shortcomings of TCP congestion control schemes, researchers have used RL to design intelligent and adaptive congestion control schemes. Table V presents a comparison of these works.

Both [131–133] focus on RL based congestion control with high throughput and low outage for high-speed mobile networks. In [131], RL congestion controller (RLCC) scheme is proposed by coupling the action-value prediction methods and TD for action selection and policy improvement. RLCC learns the optimal action to control congestion avoidance based on a signal from an immediate reward evaluator. In

TABLE V
SUMMARY OF RL APPLICATIONS IN TRANSPORT LAYER

Reference	Learning Model	Control Scheme	Proof of Convergence	Performances
[131]	TD learning	Centralized	None	• Throughput: 280 kbps; Cell loss rate: 10^{-8}
[132]	Q-learning	Cooperative	By simulation, 7 episodes	• Throughput: 125 kpbs; Packet loss rate: $5 * 10^{-9}$
[133]	DQN	Centralized	By simulation, $8s$	• Real-time throughput: 975 KB/s, 262% of vegas
[134]	Q-learning	Centralized	None	• Average delivery ratio:82%, 178.3% of RRCC, 221.6% of SR
[135]	Q-learning	Centralized	None	• Maximum throughput: 1250 kbps, 198.4% of Flush
[136]	Q-learning	Centralized	By simulation, $200s$	• Average throughput: 16 Mbps, 160% of NewReno; • Average RTT: 120.0 ms, 94.1% of NewReno
[137]	Actor-critic learning	Centralized	None	• Mean throughput: 0.56 Mbps; Mean packet loss ratio: 5%
[138]	DRL	Centralized	By simulation, $8s$	• Average throughput: 20.7 Mbps, 94.5% of BBR, 86.3% of Optimal • Average latency: 41 ms, 67.2% of BBR, 128.1% of Optimal
[139]	DDPG	Centralized	By simulation, $70s$	• Average total goodputs: 8.2Mbps, 193%-350% of wVegas
[140]	RL	Centralized	By theory	• Downloading time: 86% of Round-Robin
[141]	Q-learning	Distributed	None	• Throughput: 127% of default congestion control for MPTCP
[142]	A3C	Distributed	By simulation, 40 steps	• Mean FCT with low load: 80% of IW-10, 101% of SmartIW
[143]	A3C	Centralized	By simulation, 26 steps	• TCP response time: 95% of optimal; Throughput: 95% of optimal

contrast to [131], cooperative multi-agent congestion controller (CMCC) is proposed in [132] by extending RL to n-agents MDP stochastic game. CMCC consists of the same subsystems as RLCC, however, the best actions are learned based on a signal from a fuzzy reward evaluator utilizing game theory. Whereas Cui *et al*. [133] aim at coping with frequent handover from the perspective of transport layer, and propose Hd-TCP, which is a DQN based congestion control algorithm. In Hd-TCP framework, agent collects round-trip time (RTT), queue delay, records last action and reward, counts packet loss, and accordingly adjusts the window size. Different from previous work, five actions are considered, which allows for finer adjustment.

In [134–136], Q-learning based congestion control schemes are presented for small scale problems. The authors in [134] propose a RL framework to minimize a node's congestion in delay tolerant networks (DTN). Based on Q-learning the framework observes environment and updates Q-values as buffer occupancy, drop ratio and local congestion changes, while using two existing action selection strategies: probabilistic selection based on the Q-values, and Win or Learn Fast selection. In [135], a congestion control scheme named as eqCC is proposed for WSN by formulating the problem as MDP. Q-learning based eqCC scheme adjusts the data rate according to the queue length state at the gateway, QoS and battery levels of sensors. The performance of eqCC is compared with classic TCP and Flush. The results indicate that eqCC achieves better energy efficiency and QoS performance than other schemes in both high and low load cases. The similar performance objective is considered in [136] with throughput and latency, and QTCP is proposed by integrating the Q-learning framework with existing TCP design. In contrast to the Reno-like TCP variants that require handcrafted rule set, QTCP could gradually learn the effective strategies to adjust the congestion window size in an online manner by taking into account the following state parameters: the average transmission interval of packets, the average receiving interval of acknowledges (ACKs), and the average RTT. Most importantly, a Kanerva coding algorithm based on generalization is applied to speed up the training process. As a consequence, the proposed scheme is scalable for large complex state spaces.

Xu *et al*. [137] design a congestion control algorithm using the actor-critic method, where the fuzzy neural network is used to construct the actor and critic. Moreover, genetic algorithm is adopted to optimize the network parameters of actor neural network, which obtains Q-function values from the fuzzy inference system and calculates the fitness of Q-function by the varying of Q parameters. Each fuzzy rule is optimized based on the genetic algorithm, and simulation results show that this scheme can effectively avoid congestion by selecting the optimal dropping probability.

Different from the previous works, Jay *et al*. [138] recently investigate DRL based congestion control. The proposed framework extends the existing performance-oriented congestion control approach, and designs Custard (customized and robust decision). The agent maps statistics vectors accounting for performance measures such as packet loss rate and average latency to variations in sending rate by using a fully connected neural network.

Congestion control for multi-path TCP is investigated in [139–141]. DRL-CC proposed in [139] uses a central agent to control the congestion of all flows. According to the flow rate, goodput, average and deviation of RTTs as well as the congestion window size, the agent decides the operation on congestion window for each flow, *i.e*., increasing, decreasing or keeping the current window size. Whereas Zhang *et al*. [140] propose an RL based scheduler for multi-path TCP (ReLeS). ReLeS determines the subflow for each packet to transmit on, according to the RTT and throughput measurements. ReLeS adopts stacked LSTM to abstract features from raw measurements and the overall LSTM Q-network is asynchronously trained. A Q-learning based congestion control algorithm called SmartCC is proposed for HetNets in [141], which exploits the diversity of multiple TCP path. SmartCC adopts an asynchronous training algorithm and makes online decisions of congestion window size of each subflow for senders according to the interval of two consecutive ACKs and sending rate.

Xie *et al*. [142] focus on 5G MEC systems and propose a A3C based online decision algorithm of initial window (IW), with the objective of minimizing the flow completion time. This algorithm adopts the hierarchical learning scheme, where each MEC server acts as an sub-agent to compute the best IW using local policy, and policy parameters are sent to central

agent to update the global policy. In the learning framework, histogram which is abstracted from raw observations is taken as the state space. In addition, supervised learning using the data collected online is also used to further improve the performance. Joint IW and congestion control scheme (BBR, Cubic, etc.) determination problem is addressed using A3C in [143]. The considered system consists of a frontend server and a brain server. Users are classified into different groups according to network feature. The frontend server collects the performance data such as RTT, throughput, and send these data to the brain server, who performs A3C to determine the IW and congestion control scheme for each group.

Discussions on RL Applications in Transport Layer: According to the analysis above, RL-based congestion control schemes break the rule-based design concept in traditional TCP protocols, which is promising to be a robust method for congestion control in various scenarios of transport layer. Various parameters are considered in the state space of these RL applications, such as buffer status [131, 134, 137], queue status [132], sending rate [131, 132], battery level [135], RTT [133, 140, 143] and others. Nevertheless, though many of these parameters are vital for the optimality of decision, seldom works comprehensively consider them in modeling network dynamics. Besides, considering that high-dimensional state space results in difficulty in data processing, extracting features from raw data seems to be a potential solution. Comparing with the handcrafted abstracting approach in [142], LSTM and other DNN tools in [140] are more promising.

Moreover, different optimization objectives are adopted in these works, such as throughput [131, 132, 144], congestion level [134], delay [144], which are important metrics for congestion control. Therefore, multiple objectives are necessary for RL application in transport layer, handling the trade-offs between minimizing delay, maximizing throughput, and minimizing packet loss, which is only considered in [137, 138].

On the other hand, online bad decisions are critical issues. In this regarding, the integration of RL based congestion control schemes with existing schemes in transport layer is promising. For instance, the network conditions change frequently, which may greatly vary from the training conditions for RL methods, then detecting the network condition change and falling back to classical schemes seem to be potential methods.

E. RL for Application Layer

In the application layer, caching, multimedia transmission, wireless localization as well as NFV are becoming hot issues as users request more and heterogeneous traffic through wireless networks, location based services emerge and are provided to a large number of network subscribers. Therefore, RL has been widely used to design caching strategies, optimize the multimedia transmission, enhance the positioning function, and facilitate intelligent virtual network management. Table VI presents the summary and comparison of RL applications in the application layer.

1) Caching: Basu *et al*. [145] aim to design caching strategies to adapt to the heterogeneous content delivery network (CDN) traffic with bursty and non-stationary requests. A two-level based time-to-live (TTL) caching scheme is proposed to adapt the rare and unpopular content but suitably retaining bursty, and frequently-accessed stationary traffic, respectively.

Different from [145], the authors in [146, 147] investigate what and when to cache instead of caching duration for each content. The objectives are maximizing the caching efficiency which not only considers the hit rate of caching files, but also the file switching cost. For instance, [146] considers both local and global popularity, and uses two Markov processes to model the popularity transition process. Since transition probabilities of states are unavailable, Q-learning is applied to find the optimal caching policy in a decentralized manner at each SBS. Lin *et al*. [147] focus on the cooperation caching in cellular networks with coordinated multipoint (CoMP), where both storage and transmission cooperation can be utilized to optimize the caching performance. The problem is formulated as a MDP considering the practical time-varying content request pattern. Then Q-learning is utilized to select the caching action of the BS, according to the popularity and request distribution.

Unlike previous researches, Zhang *et al*. [148] consider content placement delivery array and propose a double coded caching scheme for ultra dense networks exploiting double DQN. The agent learns from the experience and dynamically selects the content to multicast and also the content to be exchanged among the coded and uncoded content set, according to the link status and file requests. Ning *et al*. [149] investigate the caching resource allocation for MEC based Internet of Vehicles. The agent resides in traffic controller and uses DDPG to learn the optimal computation, caching, bandwidth allocation of MEC servers to vehicles.

2) Multimedia Transmission: With the increasing traffic volume of multimedia traffic spread through wireless networks, which includes entertainment video, surveillance video, medical images and facial images, how to guarantee the QoS or QoE with limited network resources should be considered cautiously. Thus, lots of researchers have exploited RL algorithms to optimize multimedia transmission. In the following, we will discuss existing literature from different aspects. Some papers study the single-agent optimization problem [150–153], while others investigate the case when resource competition exists among multiple agents [154].

[150–153] aim to find the optimal policies from a perspective of single agent. Mao *et al*. [150] deploy a bit rate adapter at the client-side video player to optimize the user's QoE. The authors propose a novel RL framework to learn the adaptive bit rate (ABR) policy. The playback buffer occupancy, actions taken in the past and some raw observations of signals (*e.g*., throughput) are taken as the state of the ABR agent. States are inputted to an actor-critic network to determine the action, *i.e*., the bit rate for the next chunk. In addition, Huang *et al*. [152] consider that the perceptual video quality is related to not only bit rate, but also video properties, such as video type, brightness and number of objects in video. Thus, a quality awareness rate control (QARC) agent based on actor-critic learning is designed for the sender. To solve the state explosion problem, the QARC agent is implemented through

TABLE VI
SUMMARY OF RL APPLICATIONS IN APPLICATION LAYER

Domain	Reference	Learning Model	Proof of Convergence	Performances
Caching	[145]	Actor-critic learning	By simulation, 2 hours	• Cache size requirement: 44.5% and 25.4% of Fixed-TTL
	[146]	Q-learning	By simulation, 3×10^4 steps	• Convergence speed: 2% of Q-learning
	[147]	Q-learning	By simulation, 500 steps	• Cache hit rate: 130% of random caching
	[148]	Double DQN	By simulation, 100 episodes	• The variance in the uniform case: 39% of DQN, 36% of random policy
	[149]	DDPG	By simulation, 500 episodes	• Profits: 141% of non-cooperative scheme
Multimedia Transmission	[150]	Deep actor-critic learning	None	• Outperform ABR algorithms by 12% - 25%
	[151]	Deep actor-critic learning	None	• Quality: 0.25, 55.6% of Standard
	[152]	Actor-critic learning	By simulation, 60 steps	• Average quality: 104% - 109% of Greedy
	[153]	POMDP Parametric policy learning	By theory	• PSNR: 53dB, 118% of Rate-distortion based power Minimization
	[154]	Q-learning	By simulation, $10s$	• Quality stability: 114% of mDASH, 195% of onlineLearner
Wireless Localization	[155]	DQN	By simulation	• Localization accuracy: 123% of supervised learning
	[156]	DQN	By simulation, 10^6 steps	• Mean location error: 64% of DRL
	[157]	Multi-agent RL	By simulation, $1.5s$	• Mean location error: 75% of Client-based, 30% of Kalman Filter
	[158]	Multi-agent DQN	By simulation, 40 episodes	-
NFV	[159]	Q-learning	By theory	• Energy utilization: 0.68, 170% of weightless swarm algorithm, 87% of optimal
	[160]	DRL	By simulation, 500 episodes	• Average service function chain delay: 22% of existing schemes
	[161]	Deep actor-critic learning	By simulation, 80 steps	• Utilization: 80% of DDQN, 82% of DDPG, 325% of Greedy
	[162]	Double DQN	By simulation, 2100 steps	• Throughput: 190% of Eigen decomposition, 102% of MSGAS
	[163]	DQN	By simulation, 10 steps	• Admission ratio: 80%
	[164]	DRL	By simulation, 1000 episodes	• Average revenue: 157% of DDPG, 125% of customized DDPG
	[165]	Q-learning	By simulation, 1000 episodes	• Average reward: 114% of random policy, 116% of particle swarm optimization

two sub-modules. One sub-module is used to predict the video quality of the future frame, the results of which are fed to the other sub-module together with the previous network status to determine the bit rate of the next frame.

Joint power and optimal streaming rate control are investigated in [153] using RL. A two scale decision framework is proposed, where the transmission power is adaptive to the CSI and queue state information at the fast timescale, and streaming rate is only adaptive to the queue state information at the slow timescale. This optimization problem is formulated as a POMDP, based on which singular perturbation is exploited to obtain an asymptotically optimal solution with simplified process. Parametric policy learning algorithm is further used to find the optimal policy.

With the growing demands of HTTP adaptive streaming systems, multiple players might compete for the shared resources. To cope with this issue, Bentaleb *et al*. [154] use three features *i.e*., display resolution, content type, and subscription plan type to construct the virtual topology and cluster the competing players. Then, for each cluster, the bit rate selection problem is modeled as a POMDP, in which the environment states contain statuses of both global network and local players in the same cluster, rewards include the average perceptual quality and penalties (*i.e*., startup delay, stall events and quality oscillations), and the objective is to maximize the long-term revenue by choosing bit rate for each cluster. An online algorithm with value difference is proposed using Q-learning, in which the optimal action is decided in each cluster in parallel and then all clusters' Q-values are centrally processed to accelerate the convergence.

3) Wireless Localization: Location-aware services play an important part in the development of smart cities. Mohammad *et al*. [155] take the iBeacons' RSSI as the input of a DRL agent to identify indoor locations. The states of agent contain RSSI measurements, current location as well as the distance away from the target point. The reward is a function of the distance, and action is eight directions to move. Since a large number of data collected in this application is unlabeled, the authors propose a semi-supervised DRL approach to solve the problem. Specifically, the proposed semi-supervised DRL applies a Gaussian inference network to estimate the label of the unlabeled data before utilized in the DRL framework.

Li *et al*. [156] investigate wireless localization problem in a gridding space system using DQN. Several location labels and RSS features are assumed to be uniformly distributed in the gridding space. According to the previous location and RSS, the agent computes current location and chooses to stay or move a grid towards one of the eight directions. If it moves to a landmark point and has a matching RSS feature, a positive reward would be achieved.

Carrera *et al*. [157] propose a particle filter-based RL approach for robust wireless indoor localization, which is termed as PFRL. PFRL adopts distributed learning framework, and lightweight learning such as hidden Markov model based zone prediction is performed at clients. The heavy learning task, such as the particle filter takes zone prediction, move detection and a floor plan component as the input, and outputs high-precision position, is performed at the MEC server. Since the RL based resampling is adopted, PFRL has high robustness and data error tolerance.

Cooperative localization is investigated in [158]. Nodes equipped with GPS and radar communicate and help with each other to measure their relative or absolute locations. The objective is to localize all nodes as soon as possible in a distributed manner. This problem is formulated as a multi-agent POMDP. Each link between two nodes is modeled as an agent and decides whether to help measure according to distance, the number of neighbors that need help. Then the POMDP is solved by the multi-agent DQN and policy gradient.

4) Network Function Virtualization: NFV [166] is the basic enabler for virtual networks, which decouples network functions from the proprietary hardware, allowing a set of virtual network functions (VNFs) to be deployed on a shared pool of resources. In this part, we present several RL applications in NFV especially application layer problems.

Specifically, Roig *et al*. [161] consider the management and orchestration of resources occupied by VNFs in RAN, and formulate the stochastic resource optimization problem as a parameterized action MDP. To minimize the cost with economic, latency and QoS, the authors propose a novel DRL approach based on parameterized action twin deterministic policy gradient, which leverages an actor-critic architecture to provision resources to VNFs in an online manner. Besides, Gu *et al*. [164] investigate the online VNF orchestration and flow scheduling for network utility maximization. Different from traditional heuristic solutions with many assumptions, the authors apply optimization models such as inventing profiling-exploration and dual replay buffer into DRL framework for the problem. Li *et al*. [165] also study the VNF scheduling problem to minimize the make-span of all services while satisfying end-to-end delay requirements. The authors first formulate the problem as an NP-hard mixed integer linear program, and then reformulate as an MDP problem with variable action set. With a delay-aware reward function carefully designed, Q-learning is utilized for obtaining the best scheduling policy.

On the other hand, Khezri *et al*. [163] focus on NFV placement problem with the maximum number of admitted services additionally considered. Based on DQN, the resource demand and reliability requirement are modeled as states, and the placements of network providers and servers are modeled as actions. [162] also studies the VNF placement problem in SDN/NFV-enabled networks, and formulates the problem as a binary integer programming model. To minimize the placement and running cost, a double DQN based VNF placement algorithm is proposed for efficient solution.

Another important scenario for NFV is IoT. Since IoT networks are ultra-dense with the explosive growth of IoT devices, NFV with flexible network framework and efficient resource management is promising for performance enhancement in IoT networks. Especially, to combine VNFs based on the logic of IoT applications, the service function chain (SFC) needs to be embedded in IoT networks. However, the embedding process is not trivial due to the dynamic nature of IoT networks as well as abundance of IoT terminals. To overcome this, [160] employs DRL to approach the unknown NFV-enabled IoT environment model, which can allocate computing and network resources to SFCs adaptively under varying IoT conditions.

5) Discussions on RL Applications in Application Layer: From the review above, the interaction thought of RL greatly matches with the dynamics lying in the popularity of caching, the various traffic of multimedia transmission, uncertainty of localization as well as dynamics of virtual networks.

Content are usually cached at BSs with storage. Considering the diverse popularity in different areas, BSs learn their own caching policy according to local observations in a distributed way as done in [146] is an intuitive approach. However, the convergence and data collection are two critical issues. Since global popularity should also be considered to achieve the global optimality. Periodical acquirement of dynamic global popularity increases data collection complexity and causes extra signaling overhead. It shows in [146] that about 2000 iterations are required to converge to the optimal policy. Moreover, since there are a lot of contents to be cached, the dimension of state space would be extremely high, which leads to the increased complexity in data collection/processing. Hence DRL models as adopted in [148, 149] are more appropriate for in this case.

Meanwhile, RL has been used to optimize the multimedia transmission based on various tasks, such as bit rate control [150, 152–154]. The perceptual video quality in multimedia transmission is highly related to both bit rate and video properties. However, schemes in [150, 154] only care about the bit rate, which can be further optimized with video properties considered. Furthermore, the convergence speed is also an open issue. As shown in [151], the RL based multimedia transmission system performs poor video quality at the beginning due to slow convergence, while the real network conditions change frequently, which demands fast convergence. Hence, integrating multimedia transmission with fast convergence RL methods is the key to high-quality real-time video. On the other hand, multimedia transmission involves multiple clients. Considering the limited capability of data processing, multi-agent RL based schemes such as [154] are quite appropriate for user cooperation, from the point of optimality.

Wireless localization is usually coped with supervised learning with large volume of labeled data. [155, 156] provide a promising solution to combine the RL with huge volume of both labeled and unlabeled data. However, processing big data and using DRL in these works pose great challenge for mobile devices. Offloading the heavy learning tasks to the MEC server [157] and cooperative learning [158] are more appropriate approaches.

On the other hand, the implementation of virtual networks with NFV faces the dynamic issues in resource allocation, management, scheduling, and etc. Therefore, RL approaches are commonly used to model the dynamic characteristics. Especially, due to the complex system factors therein, powerful RL methods like DRL are mostly utilized in such problems. Moreover, the fast convergence for RL approaches reduces the application-level response time without service disruptions.

V. Challenges, Open Issues, and Future Research Trends

From the layering perspective, the comprehensive review above evidently demonstrates the powerful combination of RL and wireless networks. Moreover, to facilitate the development of this area, the challenges, open issues, as well as future research trends are outlined as follows.

A. Challenges of Applying RL in Wireless Networks

1) Incomplete Environment Information: According to the basic knowledge of RL in Section III, the performance of RL methods is highly related to the extent of obtained environment information. An accurate model of real world always brings lots of benefits, which is, however, hard to be guaranteed in wireless networks.

The environment modeling of wireless works usually embeds many fragile assumptions, and cannot fully describe the dynamics therein. Besides, parameters about environment need

to be available before making decisions, which is difficult in many cases. Moreover, some environment information or state parameters are hard to get. For example, the mobility information of UE, which is quite important but generally unknown at the BS when determining the handover action. Such incomplete environment information usually leads to errors in state parameters, which requires further extrapolation. Although other powerful tools such as deep learning methods may help improve network parameters, the incomplete information of wireless networks is still challenging for the application of RL.

2) Convergence Rate: The convergence rate of RL is challenging when it comes to the online application. As we know, the RL agents actively interact with the wireless environment over several iterations, which eventually learns the optimal policy. During the exploration procedure, agents would even take the wrong actions that degrade the performance greatly. However, both exploration time and wrong actions are not acceptable in some wireless network scenarios like the vehicle-to-anything (V2X) scenario, which demands ultra-reliable communication.

The convergence rate related problems are also found in both centralized and distributed RL frameworks. In the centralized case, the state parameters need to be estimated, collected, processed at the central agent, which is time consuming and requires extra signaling. As for the distributed case, wireless networks are changing dramatically, users access and depart network frequently. Therefore, the optimal policy should be obtained quickly in the short residence time of users. For example, in the mobility management case, agents at UEs need to select the optimal associated BS along its trajectory. Hence, how to overcome the slow convergence rate of RL in some wireless network scenarios is challenging.

3) Gap between Simulation and Real World: The ultimate objective of applying RL is to enable intelligent operation and management for the real wireless networks. Nevertheless, most of existing works only evaluate the proposed RL aided control scheme in the manner of random data/model-based simulation, which is far from the practical environment.

For instance, [167] uses RL to improve the performance of interference alignment and content delivery. However, the number of the candidate transmission pair is assumed to be fixed, which is impossible for the dynamically changing wireless network. Hence, the proposed RL aided control schemes are hard to implement in practical wireless networks.

Considering the high computation requirement and the necessary modification to existing protocols, there is a large gap between simulation and real world in applying RL to wireless networks. On the other hand, the RL aided control schemes have grown more sophisticated without confronting real world problems, leaving them too fragile to operate beyond deterministic and narrowly defined environments. Such evaluation methods are quite challenging for the development of RL applications.

B. Open Issues of Current Researches

1) Large Sate and Action Space: According to the review before, the large number of network environment parameters, as well as a great many of the configurable parameters in wireless networks, lead to an extremely high dimension of state-action space for RL. The large space then leads to slow convergence rate, large computation cost and high complexity, which becomes a serious issue.

Specifically, the involved environment variables in wireless networks include the available channels, channel bandwidth, channel quality, RSRP, RSSI, suffered interference in PHY layer, congestion level, queue size, sliding window size, and RTT setup in transport layer, etc. On the other hand, many state variables carry the same information about the environment, such as the channel quality and RSRP, which both implicitly indicate the location of the users. Thus, how to comprehend multiple state variables together and extract implicit features directly from high-dimensional raw observations, is quite necessary to be solved.

2) Balance between Self-Organization and Optimality: In the multi-agent learning framework, the self-organization and the optimality are the targets of user side and operator side, respectively. Especially, in RL aided control schemes, the self-organization enables the agent to autonomously learn and make the decisions through trial-and-error in interactive environments. On the other hand, the operator pursues the optimality of network management.

Since the action taken by each agent would affect the state of other agents, the iteration between agents could be considered as a non-cooperative game. To achieve the goal of self-organization, each agent adaptively changes its actions according to the environment dynamics, and tries to learn a stable policy that can improve its selfish utility. However, it is well recognized that the cooperation among agents is necessary to improve the optimality and convergence speed. Hence, there tends to be a balance between the self-organization (selfishness) and optimality (cooperation), and how to find the balance for different problems in wireless networks is still an open issue.

3) Transferring of Learned Knowledge: Due to frequent/random arrivals and departures of users, wireless network environment may dramatically change. Moreover, the optimal policy in RL is usually obtained by a large number of trials and errors. Thus, how to exploit the learned knowledge from old scenarios to accelerate the learning process in the new scenario, is a crucial problem.

For example, when applying multi-agent based RL for wireless networks, each user is equipped with an agent to learn in interactive environment. However, when a user leaves the network, his learning experience will be lost, thus the newly joining users have to learn from the very beginning. In this case, it is necessary to transfer the learned knowledge from existing users to the new users in order to help them adapt to changing environments faster, meanwhile improve the global network performance. Besides, central learning is necessary to enable efficient transfer learning.

C. Future Research Trends

1) Integration with Low-Level Function Blocks: As can be seen from Section III, most RL applications in wireless

networks lie in the MAC layer and layers above, while using RL to facilitate the design of intelligent low-level function blocks such as modulation, beamforming and error correcting coding is still less explored.

More efforts should be done to design more mature RL enabled modulation function with an advanced feedback mechanism and improved reward definition. Besides, beamforming weight calculation has always been a tough task, which highly depends on the channel condition and the QoS requirements. Though some studies have used RL to address such problems for mmWave band, the RL based beamforming weight determination for the more general case is still untouched. Moreover, error correcting code is crucial for the reliable transmission over the radio links. Furthermore, computing complexity has always been a key obstruction for the error correcting coding in the wireless networks. The future works may develop an RL aided coding function with both low complexity and high accuracy.

2) Systematic Learning Solution and Cross-Layer Design: Existing researches apply RL to solve diverse isolate control problems in the wireless networks, and most only focus on dedicated functionality. Whether it is possible to combine the learning process of different functions together and propose a systematic learning solution, remains an open issue.

On the other hand, most of existing studies only focus on different individual layers in terms of RL applications. However, it is crucial to consider cross-layer optimization to achieve network utility maximization. For example, to improve the long-term QoE of multimedia service, the RL agent can jointly make the decision of the beamforming weights, modulation and coding scheme, scheduling, resource allocation and congestion control based on the collected information about network environment across different layers.

3) Domain-specific RL Models: Although RL has been exploited for various control problems in wireless networks, most previous works directly use existing RL models in AI area for solutions. Since these models initially are proposed for AI problems, direct utilization may limit the performance. Also, due to the distinct characteristics of wireless networks, only part of RL models can be applied.

A helpful point is that, most problems of wireless networks have long been investigated and some achievements are made. In addition, these online control problems are conventionally solved by the carefully hand-crafted algorithms with validated performance in the real world environment. Therefore, a promising aspect is to use the well-recognized knowledge about the system dynamics and refer to these hand-crafted algorithms, modify the RL model to make it more suitable for the application in wireless networks.

4) Robust RL Models: As described before, the wireless network environment is partially observed to the agent. Specifically, in an RL framework, multiple network parameters act as the input, many of which depend on the accurate estimation and feedback from various nodes. Due to the limited estimation capability and unpredictable link conditions, these parameters may be not that precise and sometimes delayed. In this case, robust RL models are in demand to enable the input tolerance on the data errors or latency. Besides, since deploying an RL framework in real devices is not cheap, a robust RL model for various problems would attract much attention in future research.

5) Mobile Edge Computing Empowered RL: An important issue for RL applications in wireless networks is that, the user nodes have limited computation resources, while RL may bring much burden especially when the multi-agent learning is adopted, which significantly hinders the implementation of RL aided control schemes. On the other hand, mobile edge computing, which provides various cloud resources (*e.g.* computation and storage resources) closer to the user side for computation offloading, becomes a hot topic.

The redundant computation resource at MEC server provides a great solution for applying RL in wireless networks. For instance, in the multi-agent framework, the user device could send its information to nearby MEC servers, where RL methods are executed for the optimal policy. In this way, most computation and energy costs are moved to MEC servers. It can be envisioned that, the MEC empowered RL framework is promising in this area.

VI. Conclusions

With the growing complexity of wireless network management and controlling, the methodology of controlling-by-learning enabled by reinforcement learning has drawn great attention in academia and industry in these years. This paper has comprehensively reviewed the applications of reinforcement learning to wireless networks from a layering perspective. First, we have presented the fundamental knowledge and advanced models of reinforcement learning in the context of wireless networks. Moreover, how reinforcement learning is tailored to address various controlling and management problems in different protocol layers are analyzed detailedly. Finally, we have highlighted the challenges and open issues of applying reinforcement learning in wireless networks, and also listed several potential directions for the future research directions, through which we hope to help readers find some interesting and promising topics to focus on.

References

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 2016.

[2] Cisco, "Global mobile data traffic forecast update, 2016–2021," *white paper*, vol. 7, 2017.

[3] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. T. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: The dominant theme for wireless evolution into 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 82–89, 2014.

[4] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, and P. Soldati, "Learning radio resource management in 5G networks: Framework, opportunities and challenges," *arXiv preprint arXiv:1611.10253*, 2016.

[5] A. G. Barto, "Reinforcement learning," *A Bradford Book*, vol. 15, no. 7, pp. 665–685, 1998.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[7] M. Bkassiny, L. Yang, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, 2013.

[8] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 2014.

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

[10] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self organizing cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2392–2431, 2017.

[11] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, 2018.

[12] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 2018.

[13] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 2019.

[14] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 2019.

[15] M. Elsayed and M. Erol-Kantarci, "AI-enabled future wireless networks: Challenges, opportunities, and open issues," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 70–77, 2019.

[16] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, "Deep reinforcement learning for autonomous internet of things: Model, applications and challenges," *IEEE Commun. Surveys Tuts.*, 2020.

[17] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, 2020.

[18] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and iot networks: Potentials, current solutions, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1251–1275, 2020.

[19] 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 Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2020.

[20] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 2014.

[21] D. B. Rawat, "Fusion of software defined networking, edge computing, and blockchain technology for wireless network virtualization," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 50–55, 2019.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[23] M. L. Puterman, *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[24] M. N. Katehakis and A. F. Veinott Jr, "The multi-armed bandit problem: Decomposition and computation," *Math. Oper. Res.*, vol. 12, no. 2, pp. 262–268, 1987.

[25] W. S. Lovejoy, "A survey of algorithmic methods for partially observed markov decision processes," *Ann. Oper. Res.*, vol. 28, no. 1, pp. 47–65, 1991.

[26] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine Learning*, vol. 22, no. 1-3, pp. 123–158, 1996.

[27] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 4, no. 1, pp. 1–103, 2010.

[28] G. Tesauro, "Temporal difference learning and TD-Gammon," *ACM Commun.*, vol. 38, no. 3, pp. 58–68, 1995.

[29] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK, 1994, vol. 37.

[30] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[31] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *AAAI*, 2016.

[32] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable mdps," in *AAAI*, 2015.

[33] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *NIPS*, 2000, pp. 1008–1014.

[34] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," in *Innovations in multi-agent systems and applications-1*. Springer, 2010, pp. 183–221.

[35] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. Machine Learning*. Elsevier, 1994, pp. 157–163.

[36] C. de Vrieze, S. Barratt, D. Tsai, and A. Sahai, "Cooperative multi-agent reinforcement learning for low-level wireless communication," *arXiv preprint arXiv:1801.04541*, 2018.

[37] C. Pandana and K. J. R. Liu, "Near-optimal reinforcement learning framework for energy-aware sensor communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 788–797, 2005.

[38] C. Feres and Z. Ding, "Low complexity header compression with lower-layer awareness for wireless networks," in *Proc. IEEE ICC*, 2019, pp. 1–7.

[39] L. Zhang, J. Tan, Y.-C. Liang, G. Feng, and D. Niyato, "Deep reinforcement learning for modulation and coding scheme selection in cognitive HetNets," in *Proc. IEEE ICC*, 2019, pp. 1–6.

[40] Y.-J. Chen, W.-Y. Cheng, and L.-C. Wang, "Learning-assisted beam search for indoor mmwave networks," in *Proc. IEEE WCNC Workshop*, 2018, pp. 320–325.

[41] R. Shafin, H. Chen, Y.-H. Nam, S. Hur, J. Park, J. Zhang, J. H. Reed, and L. Liu, "Self-tuning sectorization: Deep reinforcement learning meets broadcast beam optimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4038–4053, Jun 2020.

[42] T. K. Vu, M. Bennis, M. Debbah, M. Latva-aho, and C. S. Hong, "Ultra-reliable communication in 5G mmwave networks: A risk-sensitive approach," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 708–711, 2018.

[43] G. H. Sim, S. Klos, A. Asadi, A. Klein, and M. Hollick, "An online context-aware machine learning algorithm for 5G mmwave vehicular communications," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2487–2500, 2018.

[44] A. Asadi, S. Müller, G. H. Sim, A. Klein, and M. Hollick, "Fml: Fast machine learning for 5g mmwave vehicular communications," in *Proc. IEEE INFOCOM*, 2018, pp. 1961–1969.

[45] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G mimo data for machine learning: Application to beam-selection using deep learning," in *Proc. IEEE ITA Workshop*, 2018, pp. 1–9.

[46] H. Vaezy, M. Salehi Heydar Abad, O. Ercetin, H. Yanikomeroglu, M. J. Omidi, and M. M. Naghsh, "Beamforming for maximal coverage in mmwave drones: A reinforcement learning approach," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1033–1037, May 2020.

[47] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5g networks: Joint beamforming, power control, and interference coordination," *IEEE Trans. on Commun.*, vol. 68, no. 3, pp. 1581–1592, 2019.

[48] K. Feng, Q. Wang, X. Li, and C.-K. Wen, "Deep reinforcement learning based intelligent reflecting surface optimization for miso communication systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 745–749, May 2020.

[49] X. Meng, H. Inaltekin, and B. Krongold, "Deep reinforcement learning-based power control in full-duplex cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2018, pp. 1–7.

[50] F. B. Mismar, J. Choi, and B. L. Evans, "A framework for automated cellular network tuning with reinforcement learning," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7152–7167, Oct 2019.

[51] H.-S. Lee, J.-Y. Kim, and J.-W. Lee, "Resource allocation in wireless networks with deep reinforcement learning: A circumstance-independent approach," *IEEE Syst. J.*, 2019.

[52] H. Zhang, N. Yang, W. Huangfu, K. Long, and V. C. Leung, "Power control based on deep reinforcement learning for spectrum sharing," *IEEE Trans. Wireless Commun.*, 2020.

[53] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, 2019.

[54] L. Xiao, H. Zhang, Y. Xiao, X. Wan, S. Liu, L.-C. Wang, and H. V. Poor, "Reinforcement learning-based downlink interference control for ultra-dense small cells," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 423–434, Jan 2020.

[55] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cogn. Commun. Netw*, vol. 4, no. 2, pp. 257–265, 2018.

[56] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, 2018.

[57] U. Challita, L. Dong, and W. Saad, "Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4674–4689, 2018.

[58] Z. Li and C. Guo, "Multi-agent deep reinforcement learning based spectrum allocation for D2D underlay communications," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1828–1840, 2020.

[59] A. M. Koushik, F. Hu, and S. Kumar, "Intelligent spectrum management based on transfer actor-critic learning for rateless transmissions in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1204–1215, 2018.

[60] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE ICASSP*, 2017, pp. 2087–2091.

[61] N. Rupasinghe and İ. Güvenç, "Reinforcement learning for licensed-assisted access of lte in the unlicensed spectrum," in *Proc. IEEE WCNC*, 2015, pp. 1279–1284.

[62] S. Chinchali, P. Hu, T. Chu, M. Sharma, M. Bansal, R. Misra, M. Pavone, and S. Katti, "Cellular network traffic scheduling with deep reinforcement learning," in *AAAI*, 2018.

[63] E. Nisioti and N. Thomos, "Robust coordinated reinforcement learning for mac design in sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2211–2224, 2019.

[64] E. Nisioti and N. Thomos, "Fast Q-learning for improved finite length performance of irregular repetition slotted aloha," *IEEE Trans. Cogn. Commun. Netw*, 2019.

[65] H. Zhang, W. Wu, C. Wang, M. Li, and R. Yang, "Deep reinforcement learning-based offloading decision optimization in mobile edge computing," in *Proc. IEEE WCNC*, 2019, pp. 1–7.

[66] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput. Intell.*, 2019.

[67] Q. Tang, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Decentralized computation offloading in iot fog computing system with energy harvesting: A dec-pomdp approach," *IEEE Internet Things J.*, 2020.

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

[69] D. Zeng, L. Gu, S. Pan, J. Cai, and S. Guo, "Resource management at the network edge: a deep reinforcement learning approach," *IEEE Network*, vol. 33, no. 3, pp. 26–33, 2019.

[70] J. Zhang, X. Tao, H. Wu, N. Zhang, and X. Zhang, "Deep reinforcement learning for throughput improvement of uplink grant-free noma system," *IEEE Internet Things J.*, 2020.

[71] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for uav networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, 2019.

[72] R. Balakrishnan, K. Sankhe, V. S. Somayazulu, R. Vannithamby, and J. Sydir, "Deep reinforcement learning based traffic-and channel-aware ofdma resource allocation," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.

[73] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, "iRAF: A deep reinforcement learning approach for collaborative mobile edge computing iot networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7011–7024, 2019.

[74] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 158–11 168, 2019.

[75] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, 2019.

[76] H. Peng and X. S. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Trans. Netw. Sci. Eng.*, pp. 1–1, 2020.

[77] Z. Yang, Y. Liu, Y. Chen, and G. Tyson, "Deep reinforcement learning in cache-aided mec networks," in *Proc. IEEE ICC*, 2019, pp. 1–6.

[78] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning based multi-access control and battery prediction with energy harvesting in iot systems," *arXiv preprint arXiv:1805.05929*, 2018.

[79] N. Zhao, C. Tian, M. Fan, M. Wu, X. He, and P. Fan, "Deep reinforcement learning for mobile video offloading in heterogeneous cellular networks," *Int. J. Mobile Comput. Multimedia Commun.*, vol. 9, no. 4, pp. 34–57, 2018.

[80] M. Sana, A. De Domenico, and E. Calvanese Strinati, "Multi-agent deep reinforcement learning based user association for dense mmwave networks," in *Proc. IEEE GLOBECOM*, Dec 2019, pp. 1—-6.

[81] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous networks," in *Proc. IEEE Globecom Workshops*, 2018, pp. 1—-6.

[82] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, 2019.

[83] F. Yao and L. Jia, "A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1024—-1027, 2019.

[84] N. Van Huynh, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, D. Niyato, and P. Wang, "Optimal and low-complexity dynamic spectrum access for rf-powered ambient backscatter system with online reinforcement learning," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5736–5752, 2019.

[85] D. Li, S. Xu, and J. Zhao, "Partially observable double dqn based iot scheduling for energy harvesting," in *Proc. IEEE ICC*, 2019, pp. 1–6.

[86] Y. Li, W. Zhang, C.-X. Wang, J. Sun, and Y. Liu, "Deep reinforcement learning for dynamic spectrum sensing and aggregation in multi-channel wireless networks," *IEEE Trans. Cogn. Commun. Netw*, 2020.

[87] Y. Xu, J. Yu, and R. Buehrer, "The application of deep reinforcement learning to distributed spectrum access in dynamic heterogeneous environments with partial observations," *IEEE Trans. Wireless Commun.*, 2020.

[88] Y. Yu, S. C. Liew, and T. Wang, "Carrier-sense multiple access for heterogeneous wireless networks using deep reinforcement learning," in *Proc. IEEE WCNC Workshop*, 2019, pp. 1–7.

[89] Y. Yu, S. C. Liew, and T. Wang, "Non-uniform time-step deep q-network for carrier-sense multiple access in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, 2020.

[90] Z. Ning, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, B. Hu, and Y. Li, "When deep reinforcement learning meets 5g-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1352–1361, 2019.

[91] L. Feng, Z. Yang, Y. Yang, X. Que, and K. Zhang, "Smart mode selection using online reinforcement learning for vr broadband broadcasting in d2d assisted 5g hetnets," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 600—611, Jun 2020.

[92] H. Yang and X. Xie, "An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive internet of things systems," *IEEE Syst. J.*, 2019.

[93] J. Liu, B. Krishnamachari, S. Zhou, and Z. Niu, "Deepnap: Data-driven base station sleeping operations through deep reinforcement learning," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4273–4282, 2018.

[94] Z. Xu, Y. Wang, T. Jian, W. Jing, M. C. Gursoy, Z. Xu, Y. Wang, T. Jian, W. Jing, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *Proc. IEEE ICC*, 2017, pp. 1–6.

[95] J. Ye and Y.-J. A. Zhang, "Drag: Deep reinforcement learning based base station activation in heterogeneous networks," *arXiv preprint arXiv:1809.02159*, 2018.

[96] M. Simsek, M. Bennis, and I. Guvenc, "Context-aware mobility management in HetNets: A reinforcement learning approach," in *Proc. IEEE WCNC*, 2015, pp. 1536–1541.

[97] G. Cao, Z. Lu, X. Wen, T. Lei, and Z. Hu, "Aif: An artificial intelligence framework for smart wireless network management," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 400–403, 2018.

[98] Y. Sun, G. Feng, S. Qin, Y.-C. Liang, and T.-S. P. Yum, "The smart handoff policy for millimeter wave heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1456–1468, 2018.

[99] H. Xiang, S. Yan, and M. Peng, "A realization of fog-ran slicing via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2515–2527, 2020.

[100] 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, 2019.

[101] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent resource scheduling for 5g radio access network slicing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7691–7703, 2019.

[102] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "Gan-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, 2019.

[103] 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.*, pp. 1–1, 2020.

[104] S. De Bast, R. Torrea-Duran, A. Chiumento, S. Pollin, and H. Gacanin, "Deep reinforcement learning for dynamic network slicing in ieee 802.11 networks," in *Proc. INFOCOM Workshop*, 2019, pp. 264–269.

[105] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.

[106] J. Wu, Y. Zhang, M. Zukerman, and E. K. N. Yung, "Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 803–826, 2015.

[107] X. Hu, S. Liu, R. Chen, W. Wang, and C. Wang, "A deep reinforcement learning-based framework for dynamic resource allocation in multibeam satellite systems," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1612–1615, 2018.

[108] P.-H. Huang, S.-S. Sun, and W. Liao, "Greencomp: Energy-aware cooperation for green cellular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 143–157, 2016.

[109] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5g: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, 2017.

[110] G. Faraci, C. Grasso, and G. Schembra, "Reinforcement-learning for management of a 5g network slice extension with uavs," in *Proc. INFOCOM Workshop*, 2019, pp. 732–737.

[111] Z. Jin, Q. Zhao, and Y. Su, "Rcar: A reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10 881–10 891, 2019.

[112] V. Di Valerio, F. Lo Presti, C. Petrioli, L. Picari, D. Spaccini, and S. Basagni, "Carma: Channel-aware reinforcement learning-based multi-path adaptive routing for underwater wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2634–2647, 2019.

[113] Y.-H. Chang, T. Ho, and L. P. Kaelbling, "Mobilized ad-hoc networks: A reinforcement learning approach," in *Proc. IEEE ICAC*, 2004, pp. 240–247.

[114] P. Nurmi, "Reinforcement learning for routing in ad hoc networks," in *Int. Symp. WiOpt*, 2007, pp. 1–8.

[115] P. Wang and T. Wang, "Adaptive routing for sensor networks using reinforcement learning," in *Proc. IEEE CIT*, 2006, pp. 219–219.

[116] Z. Ye, A. A. Abouzeid, and J. Ai, "Optimal stochastic policies for distributed data aggregation in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1494–1507, 2009.

[117] B. Yu, C.-Z. Xu, and M. Guo, "Adaptive forwarding delay control for vanet data aggregation," *IEEE Trans. Parallel Distrib. Syst*, vol. 23, no. 1, pp. 11–18, 2012.

[118] S.-T. Cheng and T.-Y. Chang, "An adaptive learning scheme for load balancing with zone partition in multi-sink wireless sensor network," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9427–9434, 2012.

[119] T. Hu and Y. Fei, "Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 6, pp. 796–809, 2010.

[120] T. T. Huong, N. D. D. Khoa, N. X. Dung, and N. H. Thanh, "A global multipath load-balanced routing algorithm based on reinforcement learning in SDN," in *Proc. IEEE ICTC*, 2019, pp. 1336–1341.

[121] T. Daher, S. B. Jemaa, and L. Decreusefond, "Cognitive management of self—organized radio networks based on multi armed bandit," in *Proc. IEEE PIMRC*, 2017, pp. 1–5.

[122] T. Daher, S. B. Jemaa, and L. Decreusefond, "Linear UCB for online SON management," in *Proc. IEEE VTC*, 2018, pp. 1–5.

[123] O.-C. Iacoboaiea, B. Sayrac, S. B. Jemaa, and P. Bianchi, "Son coordination in heterogeneous networks: A reinforcement learning framework," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 5835–5847, 2016.

[124] D. F. P. Rojas and A. Mitschele-Thiel, "A scalable son coordination framework for 5g," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–8.

[125] Z. Zhang, L. Ma, K. Poularakis, K. K. Leung, J. Tucker, and A. Swami, "Macs: Deep reinforcement learning based sdn controller synchronization policy design," in *Proc. IEEE ICNP*, 2019, pp. 1–11.

[126] H. D. Moura, D. F. Macedo, and M. A. Vieira, "Automatic quality of experience management for wlan networks using multi-armed bandit," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 279–288.

[127] X. Guo, H. Lin, Z. Li, and M. Peng, "Deep reinforcement learning based qos-aware secure routing for SDN-IoT," *IEEE Internet Things J.*, 2019.

[128] J.-y. Baek, G. Kaddoum, S. Garg, K. Kaur, and V. Gravel, "Managing fog networks using reinforcement learning based load balancing algorithm," in *Proc. IEEE WCNC*, 2019, pp. 1–7.

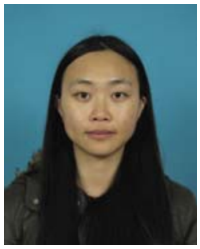
[129] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[130] K. Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportu-

nities and challenges for internet-of-things: A review," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 453–463, 2015.
[131] M.-C. Shaio, S.-W. Tan, K.-S. Hwang, and C.-S. Wu, "A reinforcement learning approach to congestion control of high-speed multimedia networks," *Cybernetics and Systems: An International Journal*, vol. 36, no. 2, pp. 181–202, 2005.
[132] K.-S. Hwang, S.-W. Tan, M.-C. Hsiao, and C.-S. Wu, "Cooperative multiagent congestion control for high-speed networks," *IEEE Trans. Syst., Man, Cybern. Syst*, vol. 35, no. 2, pp. 255–268, 2005.
[133] L. Cui, Z. Yuan, Z. Ming, and S. Yang, "Improving the congestion control performance for mobile networks in high-speed railway via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, pp. 1–1, 2020.
[134] A. P. Silva, K. Obraczka, S. Burleigh, and C. M. Hirata, "Smart congestion control for delay-and disruption tolerant networks," in *Proc. IEEE SECON*, 2016, pp. 1–9.
[135] B. T. Nguyen, L. Murphy, and G.-M. Muntean, "Energy-efficient qos-based congestion control for reliable communications in wireless multimedia sensor networks," in *Proc. IEEE ICC Workshop*, 2018, pp. 1–6.
[136] L. Wei, Z. Fan, K. R. Chowdhury, and W. M. Meleis, "QTCP: Adaptive congestion control with reinforcement learning," *IEEE Trans. Netw. Sci. Eng*, vol. PP, no. 99, pp. 1–1.
[137] T. Xu, L. Gong, W. Zhang, X. Li, X. Wang, and W. Pan, "Network congestion control algorithm based on actor-critic reinforcement learning model," in *AIP Conf. Proc.*, vol. 1955, no. 1, 2018, p. 040167.
[138] N. Jay, N. H. Rotman, P. Godfrey, M. Schapira, and A. Tamar, "Internet congestion control via deep reinforcement learning," *arXiv preprint arXiv:1810.03259*, 2018.
[139] Z. Xu, J. Tang, C. Yin, Y. Wang, and G. Xue, "Experience-driven congestion control: When multi-path tcp meets deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1325–1336, Jun 2019.
[140] H. Zhang, W. Li, S. Gao, X. Wang, and B. Ye, "Reles: A neural adaptive multipath scheduler based on deep reinforcement learning," in *Proc. IEEE INFOCOM*, Apr 2019, pp. 1648–1656.
[141] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, "SmartCC: A reinforcement learning approach for multipath tcp congestion control in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2621–2633, 2019.
[142] R. Xie, X. Jia, and K. Wu, "Adaptive online decision method for initial congestion window in 5G mobile edge computing using deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 389–403, 2020.
[143] X. Nie, Y. Zhao, Z. Li, G. Chen, K. Sui, J. Zhang, Z. Ye, and D. Pei, "Dynamic TCP initial windows and congestion control schemes through reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1231–1247, 2019.
[144] W. Li, F. Zhou, K. R. Chowdhury, and W. Meleis, "QTCP: Adaptive congestion control with reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 445–458, 2019.
[145] S. Basu, A. Sundarrajan, J. Ghaderi, S. Shakkottai, and R. Sitaraman, "Adaptive ttl-based caching for content delivery," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1063–1077, 2018.
[146] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, 2018.
[147] P. Lin, Q. Song, J. Song, A. Jamalipour, and F. R. Yu, "Cooperative caching and transmission in comp-integrated cellular networks using reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5508–5520, 2020.
[148] Z. Zhang, H. Chen, M. Hua, C. Li, Y. Huang, and L. Yang, "Double coded caching in ultra dense networks: Caching and multicast scheduling via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1071–1086, Feb 2020.
[149] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. Kwok, "Joint computing and caching in 5g-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Trans. Intell. Transp. Syst.*, 2020.
[150] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. AC Special Interest Group on Data Commun.*, 2017, pp. 197–210.
[151] B. Cheng, J. Yang, S. Wang, and J. Chen, "Adaptive video transmission control system based on reinforcement learning approach over heterogeneous networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 1104–1113, 2015.
[152] T. Huang, R.-X. Zhang, C. Zhou, and L. Sun, "Qarc: Video quality aware rate control for real-time video streaming based on deep reinforcement learning," in *ACM Multimedia*, 2018, pp. 1208–1216.
[153] S. Chai and V. K. N. Lau, "Joint rate and power optimization for multimedia streaming in wireless fading channels via parametric policy gradient," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4570–4581, 2019.
[154] A. Bentaleb, A. C. Begen, and R. Zimmermann, "ORL-SDN: Online reinforcement learning for SDN-enabled HTTP adaptive streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 3, pp. 71–98, 2018.
[155] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. S. Oh, "Semi-supervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, 2018.
[156] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, and N. El-Sheimy, "Deep reinforcement learning (DRL): Another perspective for unsupervised wireless localization," *IEEE Internet Things J.*, 2019.
[157] J. L. Carrera Villacres, Z. Zhao, T. Braun, and Z. Li, "A particle filter-based reinforcement learning approach for reliable wireless indoor positioning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2457–2473, 2019.
[158] B. Peng, G. Seco-Granados, E. Steinmetz, M. Frohle, and H. W. Wymeersch, "Decentralized scheduling for cooperative localization with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4295–4305, 2019.
[159] X. Bao, H. Liang, Y. Liu, and F. Zhang, "A stochastic game approach for collaborative beamforming in sdn-based energy harvesting wireless sensor networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9583–9595, 2019.
[160] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Dynamic service function chain embedding for NFV-enabled IoT: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, 2019.
[161] J. S. P. Roig, D. M. Gutierrez-Estevez, and D. Gündüz, "Management and orchestration of virtual network functions via deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, 2019.
[162] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal vnf placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE J. Sel. Areas Commun.*, 2019.
[163] H. R. Khezri, P. A. Moghadam, M. K. Farshbafan, V. Shah-Mansouri, H. Kebriaei, and D. Niyato, "Deep reinforcement learning for dynamic reliability aware nfv-based service provisioning," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.
[164] L. Gu, D. Zeng, W. Li, S. Guo, A. Y. Zomaya, and H. Jin, "Intelligent vnf orchestration and flow scheduling via model-assisted deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, 2019.
[165] J. Li, W. Shi, N. Zhang, and X. Shen, "Delay-aware vnf scheduling: A reinforcement learning approach with variable action set," *IEEE Trans. Cogn. Commun. Netw*, 2020.
[166] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 2015.
[167] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10 433–10 445, 2017.

Yawen Chen received his B.S. and Ph.D. degrees in information and communication engineering from Beijing University of Posts and Telecommunications, in 2013 and 2018, respectively. He is now a postdoctoral fellow at Beijing University of Posts and Telecommunications. His research interests span millimeter wave communications, cloud radio access networks, massive MIMO systems and reinforcement learning for communications.

Yu Liu received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in electronic engineering with Tsinghua University, Beijing, China. His research interests include wireless networks, edge computing, and optimization.



Ming Zeng received her B.E. degree and Ph.D degree in communication and information systems from the University of Electronic Science and Technology of China, Chengdu, China in 2009 and 2016, respectively. She worked as a Post Doctor in Tsinghua University. Now she is the assistant Professor in Beijing Institute of Technology. Her research interests include data-driven network optimization, mobile edge computing and spatial temporal data mining.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3025365, IEEE Internet of Things Journal

Umber Saleem received the B.E. degree in information and communication systems engineering in 2013 and the M.S. degree in electrical engineering (telecommunications) in 2016 from the National University of Sciences and Technology, Islamabad, Pakistan. She is currently working toward the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China. Her research focus is resource allocation for mobile edge mobile edge computing.

Zhaoming Lu received the Ph.D in Beijing University of Posts and Telecommunications in 2012, and he then joined the school of Information and communication engineering in Beijing University of Posts and Telecommunications, where he has been an assistant professor since 2017, His research includes software-defined and open-source wireless networks, massive MIMO systems, machine learning and WiFi sensing.

Xiangming Wen received the M.S. and Ph.D. degrees in information and communication engineering from the Beijing University of Posts and Telecommunications. He is currently a professor there and serves as the Director of the Beijing key laboratory of network system architecture and convergence and Beijing Laboratory of Advanced Information Networks. His research interests focus on software-defined and open-source wireless networks, and machine learning for wireless networks.

Depeng Jin received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1995 and 1999 respectively both in electronics engineering. He is a professor at Tsinghua University and vice chair of Department of Electronic Engineering. Dr. Jin was awarded National Scientific and Technological Innovation Prize (Second Class) in 2002. His research fields include telecommunications, high-speed networks, ASIC design and future Internet architecture.

Zhu Han (S'01–M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019 and ACM distinguished Member since 2019. Dr. Han is 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks."

Tao Jiang (M'06-SM'10-F'19) is currently a Distinguished Professor in the Wuhan National Laboratory for Optoelectronics and School of Electronics Information and Communications, Huazhong University of Science and Technology, Wuhan, P. R. China. He received Ph.D. degree in information and communication engineering from Huazhong University of Science and Technology, Wuhan, P. R. China, in April 2004. From Aug. 2004 to Dec. 2007, he worked in some universities, such as Brunel University and University of Michigan-Dearborn, respectively. He has authored or co-authored more 300 technical papers in major journals and conferences and 9 books/chapters in the areas of communications and networks. He served or is serving as symposium technical program committee membership of some major IEEE conferences, including INFOCOM, GLOBECOM, and ICC, etc. He was invited to serve as TPC Symposium Chair for the IEEE GLOBECOM 2013, IEEEE WCNC 2013 and ICCC 2013. He is served or serving as associate editor of some technical journals in communications, including in IEEE Network, IEEE Transactions on Signal Processing, IEEE Communications Surveys and Tutorials, IEEE Transactions on Vehicular Technology, IEEE Internet of Things Journal, and he is the associate editor-in-chief of China Communications, etc.

Yong Li (M'09-SM'16) is currently a Tenured Associate Professor of the Department of Electronic Engineering, Tsinghua University. He received the Ph.D. degree in electronic engineering from Tsinghua University in 2012. His research interests include machine learning and big data mining, particularly, automatic machine learning and spatial-temporal data mining for urban computing, recommender systems, and knowledge graphs. Dr. Li has served as General Chair, TPC Chair, SPC/TPC Member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. He has published over 100 papers on first-tier international conferences and journals, including KDD, WWW, UbiComp, SIGIR, AAAI, TKDE, TMC etc, and his papers have total citations more than 8300. Among them, ten are ESI Highly Cited Papers in Computer Science, and five receive conference Best Paper (run-up) Awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers, Young Talent Program of China Association for Science and Technology, and the National Youth Talent Support Program.