Sparsity-Aware Intelligent Massive Random Access Control for Massive MIMO Networks: A Reinforcement Learning Based Approach

Xiao Tang, Sicong Liu, *Senior Member, IEEE*, Xiaojiang Du, *Fellow, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*— Massive random access of devices brings great challenge to the management of radio access networks. Most of the time, the access requests in the network are sporadic. Exploiting the bursting nature, sparse active user detection (SAUD) is an efficient enabler towards efficient active user detection. However, the sparsity might be deteriorated in case of high concurrent request periods. To dynamically coordinate the access requests, a reinforcement-learning (RL)-assisted scheme of closed-loop access control utilizing the access class barring (ACB) technique is proposed, where the control policy is determined through continuous interaction between the RL agent and the environment. The proposed RL agent can be deployed at the next generation node base (gNB), supporting rapid switching between heterogeneous vertical applications, such as mMTC and uRLLC services. Moreover, a data-driven scheme of deep-RL-assisted SAUD is proposed to resolve highly complex environments with continuous and high-dimensional state and action spaces, where a replay buffer is applied for automatic large-scale data collection. An Actor-Critic framework is formulated to incorporate the strategy-learning modules into the intelligent control agent. Simulation results show that the proposed schemes can achieve superior performance in both access efficiency and user detection accuracy over the benchmark scheme for different heterogeneous services with massive access requests.**

***Index Terms*— Massive random access, active user detection, massive MIMO, compressed sensing, reinforcement learning.**

Manuscript received 1 May 2023; revised 14 October 2023; accepted 5 February 2024. Date of publication 21 February 2024; date of current version 14 August 2024. This work was supported in part by the Natural Science Foundation of Fujian Province of China under Grant 2023J01001; in part by the Open Research Fund of the National Mobile Communications Research Laboratory, Southeast University under Grant 2023D10; in part by the Science and Technology Key Project of Fujian Province, China, under Grant 2021HZ021004; and in part by the Science and Technology Key Project of Xiamen under Grant 3502Z20221027. The associate editor coordinating the review of this article and approving it for publication was Z. Fadlullah. *(Corresponding author: Sicong Liu.)*

Xiao Tang and Sicong Liu are with the Department of Information and Communication Engineering, School of Informatics, Xiamen University, Xiamen 361005, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: liusc@xmu.edu.cn).

Xiaojiang Du is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: dxj@ieee.org).

Mohsen Guizani is with the Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates (e-mail: mguizani@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2024.3365153.

Digital Object Identifier 10.1109/TWC.2024.3365153

I. Introduction

With the popularization of the next-generation communication technology, various types of services have developed rapidly relying on the cost-effective broadband access provided by massive machine-type communication (mMTC) [1]. Since the horizontal expansion of services is supported by the proliferation of sensors ($10^2/\mathrm{km}^2$ to $10^7/\mathrm{km}^2$), higher requirements are placed on the random access of massive devices [2], [3]. Normal-scale cells are configured with orthogonal pilots for each user, and the next generation node base (gNB) can separate multiple signals and accurately detect the active users. However, in massive random access, the existence of numerous devices makes the pilot sequences unable to satisfy the complete orthogonal relationship, and the traditional user detection mechanism can hardly make effort [4].

Fortunately, the access requests of massive users usually have a bursting nature, i.e., the active users who need to make requests to access the network occupy only a small proportion of all the potential users residing within this network [5]. This sparsity makes it possible for the compressed sensing (CS)-based active user detection (AUD), i.e. sparse AUD (SAUD) [6], [7], [8], [9]. Identifying superimposed users by SAUD can counteract the negative impact of non-orthogonal pilots and improve spectrum efficiency. However, as the number of access requests increases rapidly, or when multiple heterogeneous services need to be supported [10], more conflicts among the active users might occur, which breaks the sparsity nature and results in detrimental impact on the accuracy of AUD [11]. Hence, it is necessary to design an effective paradigm to properly manage and dynamically control the access requests of the users in the network.

The access class barring (ACB) mechanism can perform differentiated access control based on the different needs and characteristics of services in the network, thereby effectively reducing network burden. For instance, ultra-reliable low latency communications (uRLLC) services such as Internet of Vehicles are more sensitive to reliability and thus require higher accuracy of AUD, while mMTC services such as Internet of Things tend to enable more users to access at potential cost of reliability. Faced with the diverse requirements and characteristics of various scenarios in the next-generation networks [12], the ACB strategy needs to be dynamically

adjusted to be more adaptable [13]. However, it is difficult for traditional convex optimization algorithms to dynamically achieve accurate modelling and obtain a favorable access control strategy in the time-varying environments.

To this end, a reinforcement learning (RL)-based framework is introduced to facilitate a data-driven intelligent agent, which can proactively interact with the time-varying environment and realize dynamic access management through adaptive control of the ACB factors. Specifically, an RL-assisted SAUD (RL-SAUD) is devised, which adopts Q-learning to obtain experience from trials and errors and learn appropriate decision-making policies. In the proposed RL-SAUD scheme, different ACB factors are assigned to different priority-based classes, making a reasonable trade-off between the permitted access probability and the user detection accuracy. Furthermore, in order to support effective control towards continuous and high-dimensional state and action spaces, a deep reinforcement learning (DRL) assisted SAUD (DRL-SAUD) is devised to mitigate the impact of the state quantization and the dimensional curse in Q-learning. Besides, previous experience is obtained via pre-training and stored in a replay buffer, which is exploited to initialize the parameters of the deep neural networks for more rapid convergence. Then, through the closed-loop interaction between the gNB agent and the network environment, a more up-to-date and precise control can be achieved. The superiority of the proposed scheme is verified by both theoretical analysis and simulation experiments. Consequently, the main contributions of this work are summarized as follows.

- A massive random access control framework based on ACB and SAUD is formulated. The inherent sparsity of access requests is fully exploited to achieve efficient and accurate detection of active users among massive amount of potential users in the network.
- An RL-SAUD is devised to realize access management through dynamic control of the ACB factors, which can adaptively preserve the sparsity of access requests that may be deteriorated by severe conflicts of excessively massive access requests or heterogeneous services in the time-varying environments, thus sustaining the accuracy of SAUD.
- Furthermore, a DRL-SAUD with a built-in Actor-Critic module is proposed, where the previous experience is utilized for faster convergence. A data-driven paradigm is enabled by training deep neural networks, which resolves the performance degradation due to quantization error and the curse of dimensionality considering continuous state and action spaces.

The rest of this article is organized as follows. Section II reviews the related prior work. The system model of massive random access is presented in Section III. The proposed schemes of RL-SAUD and DRL-SAUD for intelligent and dynamic ACB factors and massive access control are described in detail in Section IV and Section V, respectively. The theoretical performance analysis is given in Section VI. Simulation results are reported in Section VII, which is followed by the conclusion in Section VIII.

II. Related Work

In the initial stage of random access, users intending to access the network send their pilots on the physical random access channel, and the gNB detects the active user according to the known pilot sequences [14]. This is the first step in random access, which aims to detect the correct user to establish a connection for information transfer. Unlike traditional random access, in the scenario of massive random access, it is impossible for each user to exclusively occupy an orthogonal pilot resource due to excessively huge amount of users [15]. Therefore, the performance of random active user detection has always been the bottleneck in massive random access [16], [17]. So far, the solutions in the existing literature can be roughly divided into two categories, i.e., scheduling-based random access schemes and scheduling-free random access schemes.

Scheduling-based random access schemes mainly focus on collision avoidance by reasonably allocating orthogonal time-frequency resources to users for user detection, access control and signal transmission. For instance, a dynamic back-off frame adaptation scheme was proposed to mitigate access conflicts [18]. Some solutions allow devices to adaptively occupy available access time slots [19], [20]. Nishimura et al. proposed a strongest user conflict resolution protocol within a grant-based random access paradigm, which significantly reduced failed access attempts [21]. A scheme distributes available random access resources based on learning automata, which can reasonably control the massive access requests from MTC devices [22]. Additionally, a priority-based ACB (PACB) control scheme divides users into multiple classes based on latency requirements, and implements dynamic ACB control to provide satisfactory quality-of-service (QoS) for multiple services [11]. Nevertheless, as the number of potential users grows, limited orthogonal resources impairs the scalability of the network [8].

The scheduling-free random access solution allocates a non-orthogonal pilot sequence to the user upon its initial access into the network. In subsequent access cycles, the user can access the network without requesting orthogonal resources in advance, reducing the signaling overhead and transmission delay [23]. For the received non-orthogonal signals that are mixed up, the sparse recovery-based method utilizes the sparsity of the signals to handle the interference due to non-orthogonality [24]. By taking advantage of the sparsity of active user requests, the SAUD algorithm has been widely studied in the literature. For instance, Hong et al. implemented a CS algorithm for user detection based on channel information with different precisions [9]. A modified approximate message passing algorithm exploited the structured sparsity in non-coherent transmission to enhance the scalability as the number of potential users increases [25]. Based on the inherent structured sparsity of user activities in the non-orthogonal multiple access (NOMA) system, an iterative user detection algorithm was proposed [26]. Zhang et al. employed the block sparse Bayesian learning method to solve the problem of AUD and channel estimation in NOMA, which implemented high-quality data detection and channel estimation with moderate time complexity [27]. Additionally,

the user detection problem was formulated as a joint sparse support recovery problem with multiple measurement vectors [28].

However, although SAUD is more suitable for massive access, its performance cannot be guaranteed without proper user access management due to the variations of channel conditions, random noise, and the number or the proportion of active users. During peak traffic periods, the sparsity of user access activity can be destroyed, resulting in inaccurate user detection and low access efficiency. As a standard access control mechanism, the ACB scheme manages the access of users with different priorities to connect to the network in a time-sharing manner [29], [30]. However, the model of massive random access is complex and the mutual influence between different environmental factors is implicit, making it difficult for conventional convex optimization based ACB schemes to adapt to the dynamic and intricate environments. Even if a reasonable model can be established, it is still difficult to efficiently analyze a large amount of data generated by the massive random access system.

To this end, if RL is introduced in this problem, an RL agent can utilize the utility or value function obtained from interacting with the environment to update its strategy, enabling it to adaptively and rapidly make favorable decisions that adapt to the time-varying environments [31]. At the same time, the closed-loop policy update paradigm eliminates the need for complex modeling of the multi-parameter and dynamic environments [32], [33]. Furthermore, the emerging and popular deep reinforcement learning (DRL) technique, which combines deep neural networks with RL to train enhanced intelligent agents with large-scale datasets, can be utilized to achieve more precise control performance [34], [35]. For example, a DRL-based massive random access scheme is designed to achieve continuous and optimal selection of the access time slot [34]. The dueling deep Q-Network is proposed to achieve a higher level of user satisfaction through the trade-off between access delay, energy consumption and other factors [35]. Moreover, DRL has also been adopted to solve frequent switching, access contention window adaptive determination, etc. [36], [37], and [38].

III. System Model

The model of massive random access to a massive MIMO network considered in this paper is illustrated in Fig. 1. Assume that there are in total N potential users within the network. Let $\mathcal{U}$ represent the set of all the potential users, and let $\mathcal{U}_a$ represent the set of the active users. An activity indicator α_n is used to indicate the activity of the n-th user, i.e., $\alpha_n = 1$ indicates that the n-th user has requested for access at the current time slot, and otherwise it is equal to zero. In the realistic process of establishing a connection with the gNB, an active user with a single antenna sends a unique pilot to the gNB equipped with M antennas in the radio access network, where the pilot is previously assigned to the active user by the gNB.

During a typical process of a contention free random access link establishment, three signaling messages denoted by MSG1, MSG2 and MSG3 are sent between the active user and the gNB to convey the unique information of the active user, the response of the gNB, and the acknowledge character of the active user. To be specific, in time slot t, active users modulate their unique pilots $\boldsymbol{\lambda}_k^t = [\lambda_{k,1}^t, \ldots, \lambda_{k,N}^t]^T$, $k = 1, 2, \ldots, K$, onto K OFDM sub-carriers for transmission, which can be regarded as MSG1. The channel matrix of sub-carrier k from N potential users to the gNB is denoted by $\mathbf{H}_k^t = [\mathbf{h}_{k,1}^t, \ldots, \mathbf{h}_{k,N}^t]$, where $\mathbf{h}_{k,n}^t \in \mathbb{C}^M$ denotes the channel response vector from the single-antenna user n to the M-antenna gNB. Thus, the measurement vector $\mathbf{y}_k^t \in \mathbb{C}^M$ on sub-carrier k received by the gNB is represented as

$$\mathbf{y}_k^t = \sum_{n=1}^{N} \alpha_{k,n}^t \lambda_{k,n}^t \mathbf{h}_{k,n}^t + \mathbf{z}_k^t = \underbrace{\mathbf{H}_k^t \boldsymbol{\Lambda}_k^t}_{\tilde{\mathbf{H}}_k^t} \boldsymbol{\alpha}_k^t + \mathbf{z}_k^t, \quad (1)$$

where $\boldsymbol{\Lambda}_k^t = \mathrm{diag}\{\lambda_{k,1}^t, \ldots, \lambda_{k,N}^t\}$ is a diagonal matrix whose diagonal elements are composed of the pilot vector $\boldsymbol{\lambda}_k^t$, and $\mathbf{z}_k^t \in \mathbb{C}^M$ represents the background noise. For simplicity of notation, let $\tilde{\mathbf{H}}_k^t$ represent the normalized channel matrix $\mathbf{H}_k^t \boldsymbol{\Lambda}_k^t$, which represents the original channel matrix $\mathbf{H}_k^t$ normalized by the pilots $\boldsymbol{\Lambda}_k^t$. It is worth noting that, the normalized channel matrix $\tilde{\mathbf{H}}_k^t$ can be regarded as an observation matrix in the framework of CS.

If the unknown activity indicator vector for the k-th sub-carrier $\boldsymbol{\alpha}_k^t = [\alpha_{k,1}^t, \ldots, \alpha_{k,N}^t]^T$ in (1) has a sparse property, i.e., the number of active users are much less than the total users, and the observation matrix satisfies the restricted isometry property, it is high probable that it can be recovered with bounded error from the measurement vector $\mathbf{y}_k^t$ according to the CS theory [4]. Then, the SAUD problem can be modeled as a sparse recovery problem to estimate the active user set, i.e., the positions of nonzero entries, of the unknown activity indicator vector $\boldsymbol{\alpha}_k^t$ for sub-carrier k [39].

The mechanism of massive random access requests with a series of pilot blocks sent by the active users with access requests represented in the time, sub-carrier and antenna domains is visualized in Fig. 2. The sparse recovery problem is actually to estimate the final activity indicator vector $\boldsymbol{\alpha}^t$, which can be decomposed into K sub-problems corresponding to each of the K pilot sub-carriers, as given in Eq. (1). Specifically, each of the activity indicator vectors $\widehat{\boldsymbol{\alpha}}_k^t = [\hat{\alpha}_{k,1}^t, \ldots, \hat{\alpha}_{k,N}^t]^T$, $k = 1, 2, \ldots, K$, can be estimated first

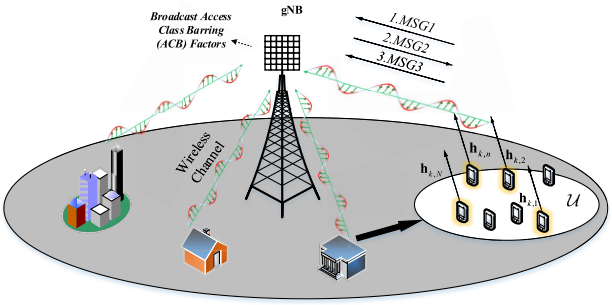


Fig. 1. Massive random access to a massive MIMO network: The gNB serves as an intelligent agent and physical-layer access point for a massive number of users with different priority ACB classes; Different QoS and access requests required by heterogeneous vertical services, time-varying channel environments, and various scenarios should be dynamically supported with adaptive switching capability.

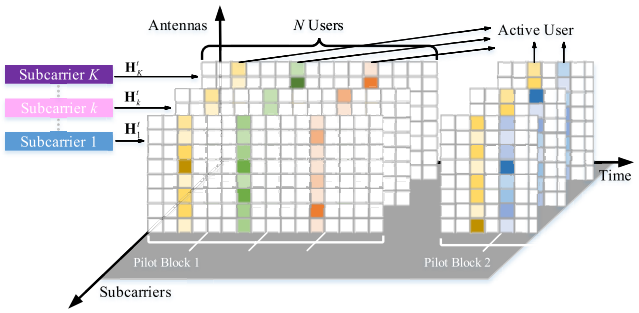


Fig. 2. Visualization of the massive random access requests, with a series of pilot blocks sent by the active users with access requests represented in the time, sub-carrier and antenna domains. The size of each pilot block is determined by the number of potential users in the network and the number of antennas at the gNB. On a certain time-frequency resource, the subset of active users with access requests shows sparse characteristics compared to the set of the total potential users in the network.

by solving its corresponding sub-problem $\mathbf{y}_k^t = \tilde{\mathbf{H}}_k^t \boldsymbol{\alpha}_k^t + \mathbf{z}_k^t$, $k = 1, 2, \ldots, K$; Then, the K estimates $\{\widehat{\boldsymbol{\alpha}}_1^t, \widehat{\boldsymbol{\alpha}}_2^t, \ldots, \widehat{\boldsymbol{\alpha}}_K^t\}$ can be regarded as K voters and combined to jointly determine the estimated final activity indicator vector $\widehat{\boldsymbol{\alpha}}^t = [\hat{\alpha}_1^t, \ldots, \hat{\alpha}_N^t]^T$: For a potential user n in the network, if more than half of the K voters judge it to be active, i.e., $1/K \sum_k \hat{\alpha}_{k,n}^t > 0.5$, user n will be finally detected as active and the n-th element $\hat{\alpha}_n^t$ of vector $\widehat{\boldsymbol{\alpha}}^t$ will be marked as one; otherwise it is considered inactive and marked as zero. This voting process can be represented by

$$\hat{\alpha}_n^t = \begin{cases} 1, & \dfrac{1}{K}\displaystyle\sum_{k=1}^{K} \hat{\alpha}_{k,n}^t > 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

After the active users are detected by SAUD via (2), the gNB sends a random access response, i.e., MSG2, to the detected users.

Starting from sending their pilots, the active users start a timer window to capture the feedback of gNB. If a user successfully parses the response corresponding to the previous MSG1 within this timer window, the device will feed back an acknowledge character (ACK), i.e., MSG3, to the gNB, indicating that the connection has been successfully established. Otherwise, it is considered as reception failure.

IV. Reinforcement-Learning-Assisted Sparse Active User Detection for Massive Random Access Control

The performance of SAUD relies heavily on the sparsity of access requests and the channel state. Thus, the access flow control of the potential users in the network should be carefully and properly managed. To this end, we propose an RL-assisted SAUD (RL-SAUD) scheme in this section, which introduces the ACB mechanism to coordinate the access requests of the users. The scheme aims to adjust the ACB factors reasonably and dynamically, and improve the detection accuracy of the SAUD while allowing as many users as possible to access to satisfy the requirements of various services. We will first introduce the ACB mechanism for flow control, and then present the proposed RL-SAUD scheme in detail in this section.

A. Access Class Barring (ACB) for Flow Control

In various heterogeneous services with excessively massive concurrent access requests, the sparsity of the activity indicator vector in problem (1) may be destroyed, which reduces the probability of accurate user detection and results in access failure [11]. Therefore, the gNB needs a reasonable strategy to control the access traffic, and ACB is a good candidate.

The ACB flow control mechanism divides the users into multiple classes according to their access priorities. Suppose there are L classes, the users in class l are represented by the set $\mathcal{U}_l$, and the number of elements in this set is N_l. The intelligent agent at the gNB generates different ACB factors $\{p_l \in [0,1]\}_{l=1}^L$ for each of the L classes to perform traffic management via a procedure called ACB check. In the process of ACB check, a certain active user $n \in \mathcal{U}_l$ with access requirements randomly samples a value $q_n \in [0,1]$ before sending its pilot: Only if $q_n \le p_l$ will user n send the pilot, otherwise it will back off to a random sampling time within a predefined range and wait for the next ACB check procedure [40]. With the support of the ACB flow control mechanism, the number of devices accessing the network in the same time period can be properly controlled and coordinated, which helps preserve the sparsity of the access requests of active users.

In order to support massive access and the coexistence of various heterogeneous services in a time-varying environment, a closed-loop control scheme based on RL is designed to adjust the ACB factors dynamically. The ACB factors are determined by the intelligent RL agent deployed at the gNB, and will be broadcast to all the potential users within the network, which will be described in detail in the next sub-section.

B. Reinforcement Learning-Assisted Sparse Active User Detection (RL-SAUD) for Massive Random Access

First, we present the model of interactions between the intelligent agent, i.e., the gNB, and the environment, i.e., the users within the network, which can be regarded as a Markov decision process (MDP). In fact, the closed-loop control is actually an MDP process, in which the intelligent agent is enabled by the RL framework. The details of the interactions in the closed-loop access control process are listed as follows.

The proposed RL-assisted SAUD algorithm is summarized in **Algorithm 1**. Specifically, to apply the Q-learning method, the ACB factor for class l is quantized into X_1 levels, i.e., $p_l^t \in \Omega \triangleq \{i/X_1, 1 \le i \le X_1\}$, where Ω is the set of feasible actions in RL, i.e., the value of ACB factors, for a certain class of users. After the selected action, i.e. ACB factor vector $\mathbf{p}^t = [p_1^t, \ldots, p_L^t]^T$ is performed, different classes of users perform the ACB check procedure, which generates a set of active users $\mathcal{U}_a$ and a corresponding activity indicator vector $\boldsymbol{\alpha}^t$. Then, the users in set $\mathcal{U}_a$ will send their unique pilot sequences to the gNB. The gNB estimates the activity indicator vector $\widehat{\boldsymbol{\alpha}}^t$ via the SAUD algorithm, and then transmits MSG2 to the detected active users. The SAUD algorithm is summarized in **Algorithm 2**, where $[\tilde{\mathbf{H}}_k]^*$ and $[\tilde{\mathbf{H}}_k]^\dagger$ represent the conjugate and Moore-Penrose inverse of the normalized channel matrix $\tilde{\mathbf{H}}_k$ of sub-carrier k as given in (1), respectively.

Algorithm 1 Reinforcement Learning Assisted Sparse Active User Detection With Traffic Flow Control (RL-SAUD)

Initialization:
Q-values $Q(\mathbf{s}^t, \mathbf{p}^t)$ in Q-table
Initialize a random state $\mathbf{s}^0$
for $t = 1, 2, 3, \ldots$ **do**
  In state $\mathbf{s}^t$, choose action $\mathbf{p}^t$ via (3)
  Calculate number of access-permitted users
  $$N_{\mathrm{p}}^t = \sum_{l=1}^{L} p_l^t N_l^t$$
  Active users to perform ACB check using $\mathbf{p}^t$ and send MSG1 to gNB if passed
  gNB performs SAUD in **Algorithm 2** and feed back MSG2
  Active users detect MSG2 and send MSG3 back to gNB
  gNB counts the number of valid access users N_{v}^t via ACKs in MSG3
  Calculate detection accuracy $c^t = N_{\mathrm{v}}^t / N_{\mathrm{p}}^t$
  Formulate the next state $\mathbf{s}^{t+1} = [\mathbf{p}^t, c^t]$
  Obtain current system utility u^t via (4)
  Update the Q-table using (5)
end

Algorithm 2 Sparse Active User Detection (SAUD)

Input:
Channel matrix $\{\tilde{\mathbf{H}}_k\}_{1 \le k \le K}$
Received signal $\{\mathbf{y}_k\}_{1 \le k \le K}$ and step size s
for $k = 1, 2, 3, \ldots$ **do**
  Initialization:
  $F_0 = \emptyset$, $\theta = 1$, $i = 1$ and residual $\mathbf{r}_0 = \mathbf{y}_k$
  while *true* **do**
    $\mathcal{S}_i = \mathrm{Max}(|[\tilde{\mathbf{H}}_{\mathrm{k}}]^* \mathbf{r}_{\mathrm{i-1}}|, \mathrm{s} \times \theta)$
    $\mathcal{C}_i = F_{i-1} \cup \mathcal{S}_i$
    $F = \mathrm{Max}(|[\tilde{\mathbf{H}}_{\mathrm{k}}]^{\dagger}_{\mathcal{C}_{\mathrm{i}}} \mathbf{y}_{\mathrm{k}}|, \mathrm{s} \times \theta)$
    $\mathbf{r} = \mathbf{y}_k - [\tilde{\mathbf{H}}_k]_F [\tilde{\mathbf{H}}_k]_F^{\dagger} \mathbf{y}_k$
    if $||\mathbf{r}||_2 < 0.01$ **then**
      break
    else if $||\mathbf{r}||_2 > ||\mathbf{r}_{i-1}||_2$ **then**
      $\theta = \theta + 1$
    else
      $F_i = F$, $\mathbf{r}_i = \mathbf{r}$, $i = i + 1$
  end
  $\widehat{\boldsymbol{\alpha}}_k = [\tilde{\mathbf{H}}_k]_F^{\dagger} \mathbf{y}_k$
end
Detect the active user $\widehat{\boldsymbol{\alpha}}$ via (2)
Output: $\widehat{\boldsymbol{\alpha}}$

The active user who successfully receives MSG2 in the timer window returns an ACK to the gNB, and the gNB estimates the detection accuracy c^t accordingly. For the convenience of referring to the past information, the *state* of the RL-based algorithm is composed of the ACB factor vector and the detection accuracy at the previous time slot, i.e., $\mathbf{s}^t = [\mathbf{p}^{t-1}, c^{t-1}] \in \mathcal{S}$. The state implies some information of the system, such as the mapping from $\mathbf{p}^t$ to c^t, which can reflect the influence of some dynamic factors, such as user mobility, the number of active users, and the change of environmental noise, on the accuracy of SAUD at time slot t. Each state-action pair corresponds a Q-value, forming a Q-table of size ${X_1}^L X_2 \times {X_1}^L$, where X_2 is the number of quantization levels of the detection accuracy c^t.

The agent tends to choose the currently optimal action, i.e., $\mathbf{p}^t = \mathbf{p}^*$ that maximizes the feedback Q value $Q(\mathbf{s}, \mathbf{p})$ in the current state $\mathbf{s}^t$, but in this way some actions might not be explored, which might result in stuck in a local optimum. In this regard, the ϵ-greedy method provides a certain small probability to adopt a random strategy by setting an ϵ value, so that every feasible action might be explored, which is given by

$$\Pr\left(\mathbf{p}^t = \mathbf{p}^*\right) = \begin{cases} 1 - \epsilon, & \mathbf{p}^* = \underset{\mathbf{p} \in \Omega^L}{\arg\max}\, Q\left(\mathbf{s}^t, \mathbf{p}\right), \\ \dfrac{\epsilon}{|X_1 + 1|}, & \text{otherwise.} \end{cases} \tag{3}$$

where Ω^L is the action space. Based on (3), the probability of choosing the action with the largest Q-value is $1 - \epsilon$.

The system utility function u^t is designed so as to strengthen the policy of choosing a favorable action over the iterative learning process, which is given by

$$u^t = c^t \sum_{l=1}^{L} p_l^t r_l N_l^t - \rho_1 \frac{1}{L} \sum_{l=1}^{L} \left(p_l^t - \overline{p^t}\right)^2 - \rho_2 (1 - c^t), \tag{4}$$

where r_l is the access priority score for class l, with a higher score indicating a higher access priority. The first term to the right of equation (4) represents the quantity of the valid accessed users weighted by the access priority scores. Intuitively, if more users with higher access priority scores are permitted to access and accurately detected, i.e., valid accessed, the system utility should get a raise. The second and third terms in (4) both play the role of penalty on the utility. Specifically, the second term in (4) represents the variance of the elements in $\mathbf{p}^t$ weighted by a coefficient ρ_1, which plays the role of a penalty on the policy ignoring the access of the users in low-priority classes. If the variance is large, it implies that the users in some of the low-priority classes are hardly permitted to access the network, which is *not* a favorable decision especially for mMTC services with massive users of different priority scores required to access. Thus, the agent can guide the algorithm to learn a policy favorable for massive access control in mMTC services by setting a positive value of the coefficient ρ_1 in the utility function (4).

On the other hand, for uRLLC services, the reliability and stability are utmost important. In this case, the third term in (4) plays the role of a penalty on the detection error weighted by a coefficient ρ_2. Thus, the agent can easily switch to a policy favorable for accurate and reliable detection in uRLLC services simply by setting a positive value of ρ_2 to include penalty on detection error. Consequently, properly setting the two coefficients ρ_1 and ρ_2 for the two penalty terms will lead

to an appropriate tradeoff between different QoS requirements of various heterogeneous services, and provide good support of flexible switching between them.

In state $\mathbf{s}^t$, the agent performs the action $\mathbf{p}^t$, and the state is transferred to $\mathbf{s}^{t+1}$, which will trigger the update of the record $Q(\mathbf{s}^t, \mathbf{p}^t)$ in the Q-table using the Bellman equation as given by

$$Q(\mathbf{s}^t, \mathbf{p}^t) \leftarrow (1-\varpi^t)Q(\mathbf{s}^t, \mathbf{p}^t) + \varpi^t(u^t + \beta \max_{\widehat{\mathbf{p}} \in \Omega^L} Q(\mathbf{s}^{t+1}, \widehat{\mathbf{p}})), \tag{5}$$

where $\varpi^t \in (0,1)$ and $\beta \in (0,1)$ represent the decaying learning rate and the discount rate, respectively. The update of the Q-table is driven by the currently obtained system utility function u^t as given in (4), which allows the RL policy to keep up with the time-varying environment.

It is worth noting that, the Q-learning method adopted by the proposed RL-SAUD scheme is a discrete action control approach, which is equivalent to sampling from the exact policy. In order to convey more specific information to achieve satisfactory performance, the quantization level of the states and actions should be smaller. However, this results in an exponential increase in the size of the Q-table, which costs too much computational complexity and storage overhead. In addition, each interaction is only learned once and the experience is not well exploited for future learning, which limits the potentials of big data-driven approaches. Therefore, it is necessary to introduce the DRL technique, i.e., a data-driven paradigm enabled by training the deep neural networks, to resolve the performance limitation due to discrete quantification and dimensional curse of continuous state and action spaces, and make full use of previous experiences for faster convergence towards learning the optimal strategy.

V. Deep-Reinforcement-Learning-Assisted Sparse Active User Detection for Massive Random Access Control

In this section, a twin delayed deep deterministic (TD3) policy gradient algorithm based on the Actor-Critic framework is introduced to dynamically adjust the ACB factors and properly coordinate the access requests, thereby improving the number of valid access users and the accuracy of SAUD. Compared with the RL-SAUD scheme, the proposed DRL-SAUD scheme uses data-driven approaches to train deep neural networks to resolve the problem of quantization loss and high-dimensional state and action spaces, and accelerate the convergence rate of the access control strategy. In fact, the TD3 algorithm adopted in the DRL-SAUD scheme is a cutting-edge DRL-based algorithm with some favorable features: i) The Actor-Critic framework underlying TD3 is very helpful for offline model testing; ii) Deep neural networks enable the state of the DRL agent to convey more complex information such as the channel state information, which is closely related with the dynamism of the environment, making it easier for the agent to capture the user mobility; iii) Better to realize continuous action control, and iv) Accelerating the convergence of learning via experience replay.

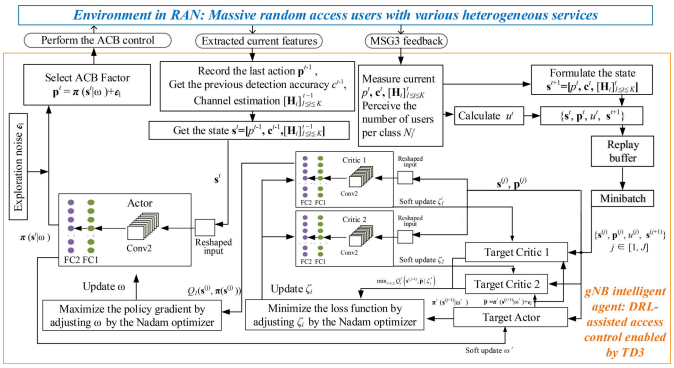


Fig. 3. Architecture of TD3-enabled dynamic control of ACB factors: Active users get access via the ACB factors determined by the intelligent agent at the gNB, powered by the DRL-assisted access control policy.

The DRL-based approach utilized in this paper is illustrated in Fig. 3. Specifically, an evaluation network, i.e., the Critic, processes and learns the experience obtained by the policy network, i.e., the Actor, and then passes the Q-value to the Actor for learning. In this way, the Actor is responsible for action decisions, and the Critic is responsible for scoring the actions. In the framework of TD3, closed-loop offline training can be performed using the Actor-Critic networks to optimize for an effective model before it is applied to the realistic environment, which can significantly improve the testing performance of the model and the user experience compared with the purely online learning method, especially at the beginning of the testing phase [41].

As shown in Fig. 3, the TD3 architecture consists of six networks, including the current Critic 1, Critic 2 and Actor, and their corresponding target networks. The current networks are intended to interact with the environment in real time, and the target networks are responsible for providing reference values for updating the current networks. By employing two Critic networks, the agent can choose a smaller Q-value during the update process to avoid overestimation of Q-values.

The proposed DRL-SAUD algorithm enabled by TD3 is summarized in detail in **Algorithm 3**. Different from the RL-based scheme, the state $\mathbf{s}^t = [\mathbf{p}^{t-1}, c^{t-1}, [\mathbf{H}_k]_{1\le k\le K}^{t-1}]$ is formulated by directly concatenating the real continuous-valued action $\mathbf{p}^{t-1}$, detection accuracy c^{t-1}, and channel matrices $[\mathbf{H}_k]_{1\le k\le K}^{t-1}$ at the previous time slot without quantization. The high-dimensional features of the state can be extracted by the Actor, and then a continuous action can be determined and output via the policy $\pi(\mathbf{s}|\omega)$. Random exploration of the agent is achieved by adding an additive noise term with variance of $\epsilon_1 \sim \mathcal{N}(0,\sigma)$ instead of the ϵ-greedy method, so that the final action to be performed is slightly modified as given by $\mathbf{p}^t = \pi(\mathbf{s}^t|\omega) + \epsilon_1$. A too small value of the additive noise will make the action no longer exploratory, while a too large value will refrain the exploitation of the learnt policy, so a proper tradeoff between exploration and exploitation can be achieved by setting an appropriate value of the additive noise.

After the ACB check, the active users start requesting access to the gNB, and then the gNB obtains the next state $\mathbf{s}^{t+1} = [\mathbf{p}^t, c^t, [\mathbf{H}_k]_{1\le k\le K}^t]$ and u^t. The information including the current state, the next state, the current utility, and the current action will be packed into a transition as an experience,

i.e., $\Im^t = \{\mathbf{s}^t, \mathbf{p}^t, u^t, \mathbf{s}^{t+1}\}$, and stored in an experience replay buffer $\mathcal{B}$. The replay buffer enables previous experiences to be learned from repeatedly to achieve faster convergence to optimal strategy. Meanwhile, outdated transitions will be replaced by the latest ones on a rolling basis to keep track of the variation of the environment.

When a number of transitions have been captured in the replay buffer, the agent randomly selects $\mathcal{J}$ transitions (a mini-batch) from the replay buffer, i.e., $\{\mathbf{s}^{(j)}, \mathbf{p}^{(j)}, u^{(j)}, \mathbf{s}^{(j+1)}\}, j \in [1, \mathcal{J}]$ to update the weights of the networks in real time. The Target Actor first computes a reference action $\widetilde{\mathbf{p}}$ for $\mathbf{s}^{(j+1)}$

$$\widetilde{\mathbf{p}} \leftarrow \pi'(\mathbf{s}^{(j+1)}|\omega') + \epsilon_2, \epsilon_2 \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -g, g), \tag{6}$$

where the policy noise ϵ_2 is a random variable following a normal distribution clipped by $\pm g$. The additive policy noise ϵ_2 can smooth the Q-function as the output of the Critic network, and enhance the reliability of the Q-value provided by the Critic with the fluctuation of the actions. Then a reference value y_{r} of the Q-value $\left\{Q_i(\mathbf{s}^{(j)}, \mathbf{p}^{(j)})\right\}_{i=1,2}$ is given by

$$y_{\text{r}}^{(j)} \leftarrow u^{(j)} + \gamma \min_{i=1,2} Q_i{}' \left(\mathbf{s}^{(j+1)}, \widetilde{\mathbf{p}}|\zeta_i{}'\right), \tag{7}$$

where γ is a discount factor. The Critic network uses the Nadam optimizer to minimize the loss function as given by

$$\zeta_i \leftarrow \arg\min_{\zeta} \frac{1}{\mathcal{J}} \sum_j \left(y_{\text{r}}^{(j)} - Q_i(\mathbf{s}^{(j)}, \mathbf{p}^{(j)}|\zeta)\right)^2. \tag{8}$$

Different from the update of the RL-based scheme, the DRL-based scheme manipulates more data in one epoch of training, and a newly recorded experience can be learned several times in subsequent training.

The policy network is updated more slowly than the evaluation network, which ensures that the Critic has minimized its own estimation error before providing scores for policy updates [41]. In the design of the proposed scheme in this paper, the Critic is updated d times every time the Actor is updated, and the Nadam optimizer is adopted to maximize the policy gradient as given by

$$\omega \leftarrow \arg\max_{\omega} \frac{1}{\mathcal{J}} \nabla_{\mathbf{p}} Q_1(\mathbf{s}, \mathbf{p}|\zeta_1)|_{\mathbf{s}=\mathbf{s}^{(j)}, \mathbf{p}=\pi(\mathbf{s}^{(j)})} \nabla_{\omega} \times \pi(\mathbf{s}|\omega)|_{\mathbf{s}=\mathbf{s}^{(j)}}. \tag{9}$$

In (9), $\nabla_{\mathbf{p}} Q_1(\mathbf{s}, \mathbf{p}|\zeta_1)$ represents the gradient of $Q_1(\mathbf{s}, \mathbf{p}|\zeta_1)$ with respect to $\mathbf{p}$, and $\nabla_{\omega}\pi(\mathbf{s}|\omega)$ is the gradient of $\pi(\mathbf{s}|\omega)$ with respect to ω.

Every time the Actor is updated, a soft update is also performed on each of the target networks, which is given by

$$\zeta_i{}' = \delta\zeta_i + (1-\delta)\zeta_i{}', \omega' = \delta\omega + (1-\delta)\omega', \tag{10}$$

where $\delta \in (0, 1]$ is a memory coefficient that can be properly set to achieve tradeoff between convergence rate and accuracy.

Algorithm 3 DRL-Assisted Sparse Active User Detection Enabled by TD3 for Access Flow Control (DRL-SAUD)

Initialization:
Actor network $\pi(\mathbf{s}|\omega)$
Critic1 network $Q_1(\mathbf{s}, \mathbf{p}|\zeta_1)$, Critic2 network $Q_2(\mathbf{s}, \mathbf{p}|\zeta_2)$
Target network $\pi'(\mathbf{s}|\omega')$, $Q_1{}'(\mathbf{s}, \mathbf{p}|\zeta_1{}')$, $Q_2{}'(\mathbf{s}, \mathbf{p}|\zeta_2{}')$
Reset replay buffer $\mathcal{B}$
Generate a random initial state $\mathbf{s}^0$
for $t = 1, 2, 3, \ldots$ **do**
  Determine the action $\mathbf{p}^t = \pi(\mathbf{s}^t|\omega) + \epsilon_1$ for state $\mathbf{s}^t$
  Active users send MSG1
  gNB performs SAUD in **Algorithm 2** to detect active users, and then feeds back MSG2
  Detected active users receives MSG2 and returns ACK
  gNB formulates next state $\mathbf{s}^{t+1} = [\mathbf{p}^t, c^t, [\mathbf{H}_k]^t_{1\le k\le K}]$ and derive current system utility u^t
  Store transition $\{\mathbf{s}^{(j)}, \mathbf{p}^{(j)}, u^{(j)}, \mathbf{s}^{(j+1)}\}$ in replay buffer $\mathcal{B}$
  Randomly sample $\mathcal{J}$ transitions from $\mathcal{B}$
  Obtain the reference action $\widetilde{\mathbf{p}} \leftarrow \pi'(\mathbf{s}^{(j+1)}|\omega') + \epsilon_2$
  Calculate the reference Q-value y_{r} using (7)
  Update ζ_i via (8)
  if (t *mod* $d = 0$) **then**
    Update ω via (9)
    Soft update weights of target networks $\zeta_i{}'$ and ω' using (10)
  end
  $t = t + 1$
end

VI. Performance Analysis and Evaluation

In this section, we present theoretical performance analysis and evaluation on some important issues related with the proposed schemes. First, for the SAUD algorithm, the detection accuracy with respect to the sparsity of the user access requests is theoretically analyzed. Then, the convergence of the proposed RL-based schemes is demonstrated. Moreover, for the RL-based schemes, the theoretical bound of the utility function is derived, and the computational complexity of the proposed schemes is evaluated.

A. Performance Analysis of SAUD

The SAUD problem as formulated in (1) can be convex relaxed by ℓ_1-norm minimization of the unknown sparse activity indicator vector [42], which yields a convex optimization problem as given by

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{C}^N} \left\{ \frac{1}{2M} \left\| \mathbf{y} - \tilde{\mathbf{H}}\boldsymbol{\alpha} \right\|_2^2 + \varepsilon_N \|\boldsymbol{\alpha}\|_1 \right\}, \tag{11}$$

where the first term in the minimization problem represents the regularization of the ℓ_2-norm error of sparse recovery. The second term is the ℓ_1-norm of the unknown sparse activity indicator vector $\boldsymbol{\alpha}$, which encourages a sparse solution of $\boldsymbol{\alpha}$. A coefficient ε_N is adopted to make tradeoff between the measurement error due to sparse recovery and the sparsity requirement of the active users, which is given by

$\varepsilon_N = \sqrt{\frac{2\varphi \log N}{\eta N}}, \varphi > 2$. For example, when the number of user access requests increases sharply, the value of ε_N can be reduced to moderately relax the requirement of access sparsity. After performing SAUD, the set of active users can be obtained from the support of the recovered activity indicator vector $\hat{\boldsymbol{\alpha}}$. The active user detection accuracy c is then obtained by the ℓ_1-norm difference between the recovered and the real activity indicator vectors, which is given by

$$c = 1 - \frac{\|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\|_1}{N}. \tag{12}$$

Theorem 1: The active user detection accuracy c is lower-bounded by

$$c \geq 1 - a_1 \exp\left(-a_2 \min\left\{\|\boldsymbol{\alpha}\|_1, \log(N - \|\boldsymbol{\alpha}\|_1)\right\}\right), \tag{13}$$

if the following conditions are satisfied

$$\begin{aligned} \|\boldsymbol{\alpha}\|_{1,\max} &\geq \frac{\eta N(1 - \frac{1}{\varphi})}{2 \log N} \\ |\lambda_{k,n}|_{\min} &> \vartheta \varepsilon_N = \vartheta \sqrt{\frac{2\varphi \log N}{\eta N}}. \end{aligned} \tag{14}$$

Proof. See the details in Appendix A. ■

Remark 1: It is indicated from **Theorem 1** that, the active user detection accuracy is closely related with the sparsity of the user access requests. The two constraints in (14) are the constraint on the sparsity of the user access requests and the constraint on the minimum amplitude of the user pilot $|\lambda_{k,n}|_{\min}$, respectively. $\|\boldsymbol{\alpha}\|_{1,\max}$ denotes the maximum sparsity level of the activity indicator vector that is tolerable in the network, which is subject to the constraint as given by

$$M = 2\left(\|\boldsymbol{\alpha}\|_{1,\max} + \frac{1}{\varepsilon_N^2}\right) \log\left(N - \|\boldsymbol{\alpha}\|_{1,\max}\right). \tag{15}$$

For the sparse measurement model as given in (1), $\eta = M/N$ represents the compressive measurement ratio, which is the ratio of the measurement data size, i.e., the number of antennas, to the length of the sparse vector, i.e., the total number of potential users. The parameters of ϑ and φ are set manually in the optimization process.

B. Convergence Analysis of Reinforcement Learning Based Scheme

For the proposed RL-SAUD scheme, assuming that $Q^{\pi}(\mathbf{s}, \mathbf{p})$ is the target Q-table in a control task, we use $Q^t(\mathbf{s}, \mathbf{p})$ to represent the updated value of the Q-table in the t-th update iteration, and it is updated by

$$\begin{aligned} Q^{t+1}(\mathbf{s}, \mathbf{p}) \leftarrow\ & Q^t(\mathbf{s}, \mathbf{p}) \\ & + \varpi^t [u^t + \beta \max_{\mathbf{b} \in \Omega^L} Q^t(\mathbf{s}', \mathbf{b}) - Q^t(\mathbf{s}, \mathbf{p})]. \end{aligned} \tag{16}$$

Besides, we define the update operator of the Q-table as $\mathbf{T}$, which is given by

$$\mathbf{T}Q(\mathbf{s}, \mathbf{p}) = \sum_{\mathbf{s}' \in \mathcal{S}} \Upsilon_{\mathbf{p}}(\mathbf{s}, \mathbf{s}')[u(\mathbf{s}, \mathbf{p}, \mathbf{s}') + \beta \max_{\mathbf{b} \in \Omega^L} Q(\mathbf{s}', \mathbf{b})], \tag{17}$$

where $\Upsilon_{\mathbf{p}}(\mathbf{s}, \mathbf{s}')$ is the probability that the environment changes from state $\mathbf{s}$ to $\mathbf{s}'$ when the agent chooses action $\mathbf{p}$. When the Q-table converges, a further update iteration will not produce any new changes, that is

$$\mathbf{T}Q^{\pi}(\mathbf{s}, \mathbf{p}) = Q^{\pi}(\mathbf{s}, \mathbf{p}). \tag{18}$$

Then, the RL-SAUD scheme will finally converge to the target policy, which is supported by the following theorem:

Theorem 2: Denoting the error between the calculated Q-table of RL-SAUD and the target Q-table in the training process as $\Delta^t(\mathbf{s}, \mathbf{p}) = Q^t(\mathbf{s}, \mathbf{p}) - Q^{\pi}(\mathbf{s}, \mathbf{p})$, over the iterations of training, we have:

$$\lim_{t \to \infty} \Delta^t(\mathbf{s}, \mathbf{p}) = 0, \tag{19}$$

which shows that $\Delta^t(\mathbf{s}, \mathbf{p})$ will converge to zero as t increases.

Proof. See the details in Appendix B. ■

C. Theoretical Bound and Computational Complexity of Reinforcement Learning Based Schemes

To evaluate the theoretical performance of the proposed RL-based and DRL-based schemes, the system utility as in (4) can be derived in closed-form for a typical case with each access priority score identical, which is presented in the following theorem.

Theorem 3: If the access priority score of each user class is identical, the problem of maximizing the system utility function u as given in (4) is turned into a convex problem that has a tractable theoretical bound.

Proof. See the details in Appendix C. ■

Remark 2: When there is no difference in access priority of the users in the network, i.e., $L = 1$, the first penalty term in (4) intended to mitigate the variance of the ACB factors of different priority classes will disappear. In this case, the RL-based and DRL-based schemes are actually dynamically searching for a proper policy of access control to approximate the optimal solution or a sub-optimal solution towards the theoretical bound derived in **Theorem 3**.

When different user classes have different access priority scores, more uncertainty is brought to the network. Considering the time-varying property of the environment and the diverse channel conditions and QoS requirements of different user classes, it is not guaranteed that the optimal solution of the original problem is still tractable in closed-form. Hence, the ability of RL-based schemes in searching for a sub-optimal solution towards the system utility in complex environments over a reasonable time frame can be exploited. In the decision process of the RL-based and DRL-based schemes, there are two tradeoffs that need to be considered: The tradeoff between the number of high-priority permitted-access users and the variance in the number of permitted-access users with different priority scores, and the tradeoff between the total number of permitted-access users and the active user detection accuracy.

Next, we will investigate the computational complexity of the proposed RL-based and DRL-based schemes. According to the related research in literature [43], when the number of training episodes of an RL-based algorithm is τ with each

episode including ξ time slots, the computational complexity is given by

$$\mathcal{T}_1 = \mathcal{O}(\tau\xi) \tag{20}$$

if $\tau\xi > \text{poly}({X_1}^L X_2, {X_1}^L, \tau)$, where $\text{poly}(a, b, c)$ is a third-order polynomial whose three roots are a, b, and c.

With the increase of quantization level of the state and action spaces, the number of feasible actions and states increases dramatically, which costs more time slots ξ in each episode for the RL-SAUD scheme to converge. As the side-effect to reduce the performance loss caused by quantization error, the increase of action-state pairs makes the RL-based algorithm cost more searching and computational overhead in random exploration at the early stage of the learning process. For the DRL-SAUD scheme, the computational complexity of float-point calculation in the deep neural networks, which is measured by float-point operations per second (FLOPs), is the main contributor to the overall complexity. In the devised architecture, a network model including a single convolutional neural network (Conv) layers with C_{i} input channels and C_{o} output channels, and two fully connected (FC) layers is considered. Then, the computational complexity of the proposed DRL-SAUD scheme is derived by the following theorem.

Theorem 4: In the training process of the DRL-SAUD algorithm with τ episodes and ξ time slots for each episode, the computational complexity is given by

$$\mathcal{T}_2 = \mathcal{O}\left(\tau\xi C_{\text{i}} w_{\text{i}}^2 C_{\text{o}} ((w_{\text{i}} - v)/s + 1)^2\right) \tag{21}$$

where v and s denote the size and the stride of the convolutional kernel, respectively; w_{i} denotes the size of each input channel of the Conv layer.

Proof. See the details in Appendix D. ■

Remark 3: The concept of open service in next-generation communication enables manufacturers to expand the functions of their network designs, and mobile operators can also support the coexistence of multiple vertical services. In practical deployment of multiple heterogeneous services, there is a significant increase in the amount of input parameters to the neural networks. Thus, it is difficult to analyze huge amount of information of the environment and the system by merely using a simple network of FC layers. Moreover, numerous parameters in the FC layers slow down the computation and it is more likely to cause overfitting. By utilizing Conv layers, one can reuse the parameters of the convolution kernel without consuming too much computational complexity overhead as shown in this theorem. This helps better extract the high-dimensional features of the complex system and realize a more efficient data-driven intelligent scheme for the agent.

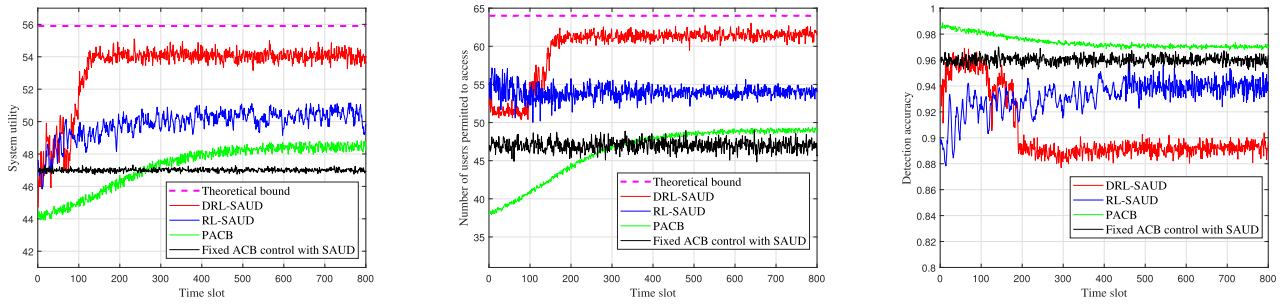
VII. Simulation Results

In this section, the performance of the proposed RL-SAUD and DRL-SAUD schemes is evaluated through extensive simulations. Some typical metrics, such as the number of users permitted to access and the active user detection accuracy, are investigated to show the performance of the massive random access control schemes. The number of users permitted to access the network is investigated to show the performance of access efficiency and throughput of the users, while the active detection accuracy is investigated to show the reliability and stability of the access requests. Further, we explore the relationship among bit error rate (BER), detection accuracy and proportion of active users. The effectiveness and adaptability of the proposed schemes are validated in different scenarios and various heterogeneous vertical services, such as mMTC and uRLLC services. The tendence of the access control strategy determined by the proposed schemes is also demonstrated for different classes of users with different access priorities.

The simulation configuration is set up as follows[1]: The number of potential users residing within the network is set to $N = 300$ and evenly divided into $L = 2$ classes with different access priorities. The number of antennas of the gNB is $M = 128$. The carrier central frequency is located at 2GHz, the total number of OFDM sub-carriers is $N_{\text{sc}} = 1024$, and the number of pilot sub-carriers is is $K = 64$. The parameter configuration of the RL-SAUD scheme is as follows: The value of ACB factor and user detection accuracy have both been divided into five levels ($X_1 = 5$ and $X_2 = 5$). The learning rate and discount rate of the Q-table are set to $\varpi^t = 1/t$ and $\beta = 0.3$, respectively. For the DRL-SAUD scheme, the learning rate of the Critic applied for TD3 is set as 0.00001, and the learning rate of the Actor is set as 0.00005 due to delayed update. The Actor and the Critic share an identical network architecture, which consists of one Conv layer and two FC layers. The capacity of the replay buffer is set as 1000. The size of a mini-batch is 128. The variance of the additive noise to encourage exploration is set as $\epsilon_1 = 0.15$. The variance of the policy noise for the Target Actor is set as $\epsilon_2 = 0.25$, whose clip boundary g is set as 0.4. The delayed update time is $d = 4$.

The performance of the proposed RL-SAUD and DRL-SAUD schemes in the system utility, access efficiency, and detection accuracy is reported in Fig. 4, where an mMTC service with massive access requests from huge number of users is considered. The fixed ACB control scheme with SAUD [9] and the proactive PACB scheme [11] are evaluated as the benchmarks, and the theoretical bound is also depicted for comparison. First, the performance of system utility is reported in Fig. 4(a). It can be observed from Fig. 4(a) that, the proposed RL-SAUD and DRL-SAUD schemes significantly outperform the benchmark schemes, which validates the effectiveness of the proposed RL-assisted mechanism in achieving a better access control utility for massive access scenarios. It is also demonstrated that the DRL-SAUD scheme is approaching the theoretical bound of system utility, which verifies the superior performance of the DRL-assisted architecture in case of complex environments and high-dimensional state and action spaces. The performance gain of DRL-SAUD over RL-SAUD

[1]In order to focus on the proposed RL-based model of massive access control, the factor of hardware impairment has not been considered in this paper, while its impact on the performance can be modeled using non-linear filtering [44], evaluated by simulations, and effectively compensated for in massive MIMO systems with a large number of antennas.



(a) (b) (c)

Fig. 4. Performance of the proposed RL-SAUD and DRL-SAUD schemes in (a) system utility, (b) access efficiency, i.e., number of users permitted to access, and (c) active user detection accuracy. An mMTC service with massive access requests from huge amount of users is considered and supported. Two benchmark schemes, i.e. the fixed ACB control scheme with SAUD and the proactive priority-based ACB (PACB) scheme, are also depicted for comparison.

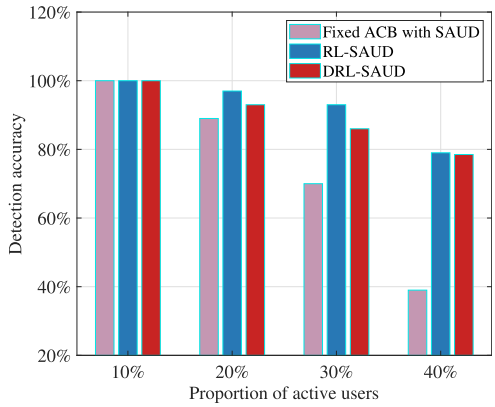


Fig. 5. The accuracy of active user detection versus the number, i.e., the proportion, of active users.

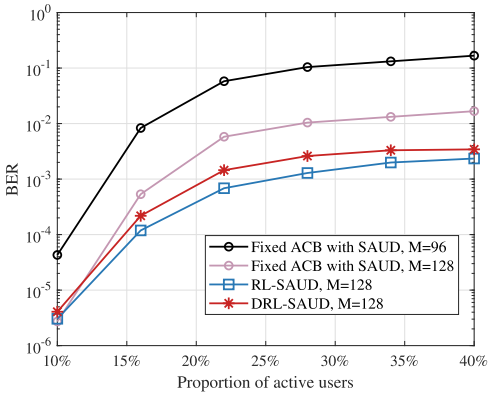


Fig. 6. The BER performance versus the number, i.e., the proportion, of active users.

validates the effectiveness of utilizing the TD3 architecture with deep neural networks to extract more complex information from the environment, and the degradation on the RL-based scheme caused by quantization error.

Second, the performance of access efficiency, which is indicated by the number of users permitted to access the network, is reported in Fig. 4(b). It is observed from Fig. 4(b) that, DRL-SAUD permits about 62 users to access the network, while RL-SAUD permits about 53 users to access. Thus, DRL-SAUD can support more users than RL-SAUD because it overcomes the bottleneck of quantization error and can find a better solution approaching the optimal bound, which is favorable for mMTC services with massive users intended to access. In comparison, the benchmark scheme with fixed ACB only permits 48 users to access, reflecting lack of flexibility to massive access requests. Since active users raise access requests using orthogonal resources in the PACB scheme, the maximum number of permitted users is the total number of orthogonal resources, which causes stronger limitation on the amount of users compared to the sparse active user detection scheme. In addition, it can also be observed that, the convergence rate of the three proactive schemes is ordered as DRL-SAUD, RL-SAUD and PACB. Since a faster convergence rate means a more up-to-date and precise control, DRL-SAUD performs the best among them.

Third, the active user detection accuracy is reported in Fig. 4(c). It is shown from Fig. 4(c) that the user detection accuracy of RL-SAUD and DRL-SAUD reaches approximately 94.93% and 88.07%, respectively. Although RL-SAUD has a relatively higher detection accuracy than DRL-SAUD, however, the number of permitted-access users of RL-SAUD is much fewer than that of DRL-SAUD, which leads to a lower system utility as reported in Fig. 4(a). This implies that a bit decrease in detection accuracy can be compensated by a great increase in the number of permitted-access users. Note that, the intelligent agent bears in its mind that maximizing the system utility function as given in (4) is its goal. Therefore, an optimal trade-off in between should be pursued in order to obtain a higher system utility, and this best trade-off strategy is just the solution that the DRL-SAUD scheme is searching for and finally converges to. Moreover, the user detection accuracy of PACB converges to around 97%, which implies that a highly reliable detection performance is achieved at the cost of strict orthogonal resource requirements and limited user access volume.

The performance of user detection accuracy and BER for different access control schemes versus the number, i.e., the proportion, of active users, is reported in Fig. 5 and Fig. 6, respectively. The performance of user detection accuracy is as shown in Fig. 5, with the number of antennas $M = 128$. It is observed from Fig. 5 that, with the growing number of active users, the RL and DRL-based schemes benefit more and more evidently from their capability of adaptive learning and controlling. When 40% users are active to request access, the user detection accuracy of RL-SAUD and DRL-SAUD is 39.7% and 39.3% higher than that of the fixed ACB scheme with SAUD, respectively. This verifies that the RL-SAUD and DRL-SAUD schemes can guarantee the connection reliability

and efficiency by adaptively controlling the massive user access behaviors.

The BER performance for different access control schemes versus the number of active users is reported in Fig. 6. It can be observed from Fig. 6 that, the BER performance degrades with the number of active users growing. Meanwhile, it is noted that this performance degradation can be compensated for by increasing the number of antennas, which is feasible in practice for massive MIMO systems. As the proportion of active users grows, it is shown that the RL-SAUD and DRL-SAUD schemes can help the gNB configure more appropriate ACB factors for different user classes to achieve efficient and intelligent access control, thereby improving the BER performance compared to that of the fixed ACB scheme with SAUD. In addition, if an mMTC service is considered, the coefficient ρ_2 for user detection error penalty in the utility function u in (4) can be set relatively small, and then the intelligent agent will be led to be more concentrated on the amount of accessed users rather than the reliability, as implied by Fig. 4(b) and Fig. 4(c). In this case, the DRL-SAUD scheme prefers to allow more users to access to maximize the system utility u, at the cost of a degradation in user detection accuracy as shown in Fig. 4(c), and a slight degradation in BER as shown in Fig. 6. A proper tradeoff between the number of accessed users and the reliability of connections can be achieved by adjusting the value of the coefficient ρ_2.

As reported in Fig. 7, we verify the ability of the DRL-SAUD scheme to adaptively switch between different heterogeneous services. As described in Section IV-B, for the uRLLC service, the third term of the system utility function in (4) is activated by setting a positive value of the coefficient ρ_2, which plays a role of penalty on the user detection error and thus encourages better reliability of connection. Specifically, for the uRLLC service in the simulations, the coefficient is set as $\rho_2 = 100$.

The performance of active user detection accuracy for the DRL-SAUD scheme applied in an mMTC service and a uRLLC service is reported in Fig. 7(a). It is shown by the results in Fig. 7(a) that, the detection accuracy of the uRLLC service is about 5% higher than that of the mMTC service. This improvement is beneficial for the agent, i.e., the gNB, to make a prompt and effective response to a user who initiates an access request in an uRLLC service. It can also be noted from Fig. 7(a) that, in the early stage of the learning process, the curve of detection accuracy for the uRLLC service has a deep valley, but is then pulled up rapidly. This indicates that the DRL-SAUD scheme firstly performs initial random exploration to probe the environment, and then can rapidly adjust its strategy and converge to an optimized solution because of the influence of the penalty of detection accuracy on the system utility.

On the other hand, the performance of access quantity, i.e., the number of users permitted to access, for the mMTC and uRLLC service is reported in Fig. 7(b). It is shown that the proposed scheme permits about 62 users to access for the mMTC service, while about 51 users are permitted to access for the uRLLC service. This result implies that the proposed scheme aims to permit more users to access the network for the

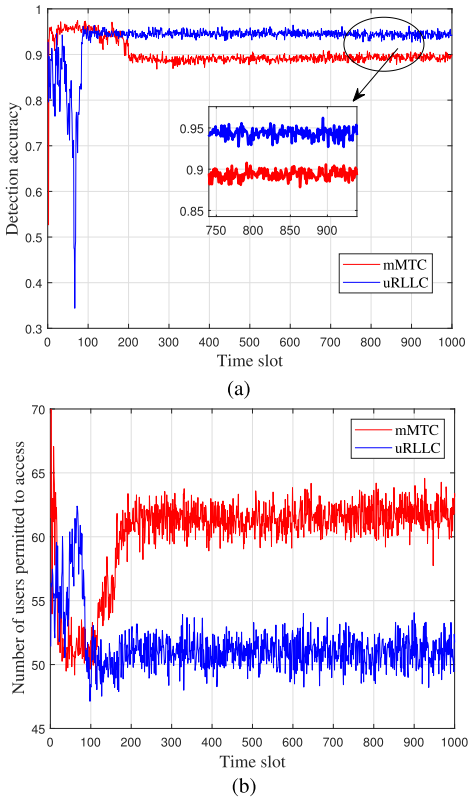
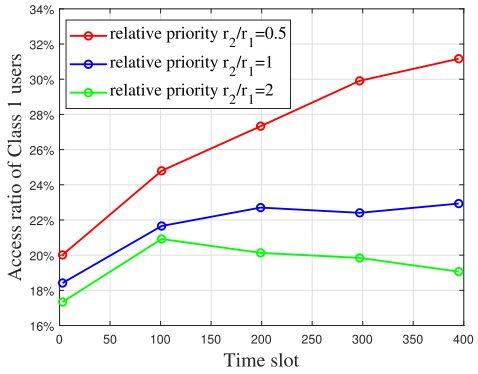


Fig. 7. Performance of the proposed DRL-SAUD scheme applied in a uRLLC service compared with that of an mMTC service: (a) Active user detection accuracy; (b) Number of users permitted to access the network.

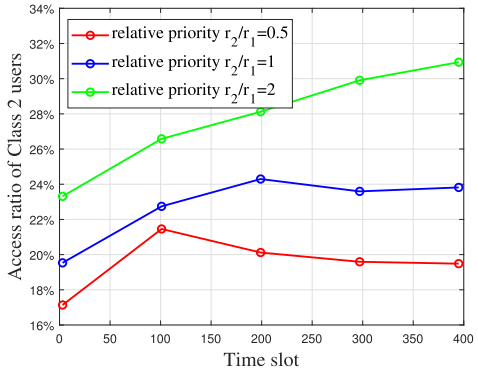
mMTC service, while it determines to sacrifice a bit of access quantity to improve the reliability and stability of the access connections for the uRLLC service. In the proposed RL-based framework, the agent can easily switch from an mMTC service with a policy favorable for a larger access quantity, to a uRLLC service with a policy favorable for accurate user detection and reliable access connection, simply by setting a positive value of ρ_2 to include the second penalty term on detection error in the utility function in (4). In practical implementation, rapid switching between different heterogeneous services can be realized by simply adjusting the penalty coefficients.

To observe the behavior of the proposed DRL-SAUD scheme when faced with users with different access priorities, the access ratio of two differently prioritized classes of users, i.e., Class 1 and Class 2, is reported in Fig. 8. The access ratio of a certain class of users is defined as the percentage of the users permitted to access the network with respect to all the potential users in that class. Let r_2/r_1 represent the relative priority between the two classes considered, which is defined as the ratio of the priority score of Class 2 r_2 with respect to the priority score of Class 1 r_1. In this case, the first penalty term on the system utility in (4), i.e., the penalty due to the variance of the access quantities between different prioritized classes of users, is activated by setting the corresponding coefficient as $\rho_1 = 120$.

Specifically, the performance of access ratio for the users in Class 1 and Class 2 are reported in Fig. 8(a) and Fig. 8(b), respectively. Three cases with different values of relative priority are investigated for comparison, i.e., r_2/r_1 is set as 0.5, 1, and 2. From Fig. 8 (b), it is observed that the access



(a)



(b)

Fig. 8. The performance of access ratio, i.e. the percentage of users permitted to access the network, for two classes with different access priority scores: (a) Access ratio of Class 1 users with access priority score r_1; (b) Access ratio of Class 2 users with access priority score r_2.

ratio of Class 2 is 30.9% in case of $r_2/r_1 = 2$, which is 9.39% and 35.27% higher than the cases of $r_2/r_1 = 1$ and $r_2/r_1 = 0.5$, respectively, which indicates that the proposed scheme has learned the tendence to allow more higher-prioritized users to access the network to improve the system utility. Regardless of whether $r_2/r_1 = 0.5$ or $r_2/r_1 = 2$, during the early stage of training, i.e., over time slots $[0, 100]$, the agent allows a growing number of users for both two classes to access, which is an exploration behavior to improve utility. However, allowing too many low-priority users to access will obstruct high-priority users to access, which limits a further increase in utility. Therefore, the agent determines to prohibit some low-priority users from accessing over time slots $[100, 400]$, sparing for more high-priority users.

It is can also be noted by comparing Fig. 8(a) and Fig. 8(b) that, the access ratio of the two classes is similar to each other in case of $r_2/r_1 = 1$, which implies that the proposed scheme has learned to permit approximately the same amount of users to access with the same access priority. This is because when the access priorities of the two classes are equal, the agent assigns similar ACB factors to them to minimize the negative impact of the variance penalty on system utility. The results in Fig. 8 have verified the adaptability of the proposed DRL-assisted scheme to different prioritized users or various heterogeneous services with different QoS requirements.

VIII. Conclusion

Faced with the challenge of massive random access control in the next-generation radio access networks, this paper has proposed an RL-assisted framework of dynamic access control, which can be deployed in the intelligent agent at the gNB. In order to preserve the sparsity of the access requests to guarantee the accuracy of SAUD in case of ultra-dense traffic, the proposed RL-SAUD scheme can dynamically adjust the ACB control strategy in a closed-loop access control process. A system utility function, which is in favor of increasing the quantity of the users permitted to access the network, has been devised and utilized to train the RL model, and two penalty terms related with the variance of access ratio and the detection accuracy are adopted to support a proper tradeoff and flexible switching between different heterogeneous vertical applications, such as mMTC and uRLLC services.

Furthermore, in order to overcome the quantization error of the RL-based scheme due to discretizing the actions and states using, the DRL-SAUD scheme has been designed based on the Actor-Critic underlying TD3 framework. The information of the environment can be better extracted by the deep neural networks, and the policy and actions can be chosen from a continuous space to obtain an improved solution approaching the optimal bound. Past experiences are exploited to accelerate the convergence of learning by using experience replay buffer. The theoretical analysis and simulation results have validated the efficiency, adaptability, and reliability of the proposed schemes in dynamic and intelligent massive access control for different QoS requirements, different vertical services and different prioritized classes of users. The technique is promising to be applied in the next-generation network architectures to provide an efficient and effective solution for the ever-crowded and ever-complex radio access environments and services.

Appendix A
Proof of Theorem 1

Proof. According to related research in literature [42], it has been proved that the active user detection accuracy c is lower-boudnded by

$$c \geq 1 - a_1 \exp\left(-a_2 \min\left\{\|\boldsymbol{\alpha}\|_1, \log(N - \|\boldsymbol{\alpha}\|_1)\right\}\right), \quad \text{(A.1)}$$

subject to the following constraint,

$$M \geq 2\left(\|\boldsymbol{\alpha}\|_1 + \frac{1}{\varepsilon_N^2}\right)\log(N - \|\boldsymbol{\alpha}\|_1)$$
$$|\lambda_{k,n}|_{\min} > \vartheta\varepsilon_N \quad \text{(A.2)}$$

where N, M, and $||\boldsymbol{\alpha}||_1$ denote the number of all the potential users in the network, the measurement vector size, and the sparsity level of the active user requests. According to the CS theory [4], the sparsity level that can be recovered accurately should be smaller than the number of measurement data M, so we have $M > \|\boldsymbol{\alpha}\|_{1,\max}$. Then, we can derive that,

$$\begin{aligned} 2(||\boldsymbol{\alpha}||_1 + \frac{1}{\varepsilon_N^2})\log(N - M) &\\ \leq 2(||\boldsymbol{\alpha}||_1 + \frac{1}{\varepsilon_N^2})\log(N - ||\boldsymbol{\alpha}||_1) &\\ \leq 2(||\boldsymbol{\alpha}||_1 + \frac{1}{\varepsilon_N^2})\log N. & \quad \text{(A.3)} \end{aligned}$$

Substituting the constraint in (15) into (A.3), we have

$$\begin{aligned} M &= 2\left(\|\boldsymbol{\alpha}\|_{1,\max}+\frac{1}{\varepsilon_N^2}\right)\log\left(N-\|\boldsymbol{\alpha}\|_{1,\max}\right) \\ &\le 2\left(\|\boldsymbol{\alpha}\|_{1,\max}+\frac{1}{\varepsilon_N^2}\right)\log\left(N\right) \\ &\le 2\left(\|\boldsymbol{\alpha}\|_{1,\max}+\frac{M}{2\log N}\frac{1}{\varphi}\right)\log\left(N\right), \end{aligned} \tag{A.4}$$

which is equivalent to

$$\|\boldsymbol{\alpha}\|_{1,\max} \ge \frac{\eta N(1-\frac{1}{\varphi})}{2\log N}. \tag{A.5}$$

Therefore, if the constraint in (14) is satisfied, the lower-bound of the detection accuracy can be derived in (A.1), which concludes the proof. ■

Appendix B
Proof of Theorem 2

Proof. Let a random process Δ^t be defined as:

$$\Delta^{t+1}(x) = (1-\alpha^t(x))\Delta^t(x) + \alpha^t(x)F^t(x). \tag{B.1}$$

Lemma 1: Δ^t will converge to 0 when the following conditions are met [45]:

- $0 \le \alpha^t \le 1, \sum_t \alpha^t(x) = \infty$ and $\sum_t [\alpha^t(x)]^2 < \infty$
- $\exists \gamma < 1, \text{s.t.} ||\mathbb{E}[F^t(x)]||_\infty \le \gamma ||\Delta^t||_\infty$
- $\exists C > 0, \text{s.t.} \text{var}[F^t(x)] \le C(1+||\Delta^t||_\infty^2)$

As defined in **Theorem 2**, we can get the following relationship

$$\begin{aligned} \Delta^{t+1}(\mathbf{s},\mathbf{p}) &= Q^{t+1}(\mathbf{s},\mathbf{p}) - Q^\pi(\mathbf{s},\mathbf{p}) \\ &= (1-\varpi^t)\Delta^t(\mathbf{s},\mathbf{p}) + \varpi^t[u(\mathbf{s},\mathbf{p},\mathbf{s}') \\ &\quad + \beta \max_{\mathbf{b}\in\Omega^L} Q(\mathbf{s}',\mathbf{b}) - Q^\pi(\mathbf{s},\mathbf{p})] \\ &= (1-\varpi^t)\Delta^t(\mathbf{s},\mathbf{p}) + \varpi^t F^t(\mathbf{s},\mathbf{p}). \end{aligned} \tag{B.2}$$

Thus, (B.2) has exactly conformed to the format of (B.1). According to basic series theory, since the learning rate ϖ^t decays in the manner of $\varpi^t = 1/t$, the series $\sum_t \varpi^t$ will diverge, while the series $\sum_t (\varpi^t)^2$ will converge, so the first condition in **Lemma 1** is satisfied. The second condition can be expressed as

$$\begin{aligned} \mathbb{E}[F^t(\mathbf{s},\mathbf{p})] &= \sum_{\mathbf{s}'\in\mathcal{S}} \Upsilon_{\mathbf{p}}(\mathbf{s},\mathbf{s}')[u(\mathbf{s},\mathbf{p},\mathbf{s}') \\ &\quad + \beta \max_{\mathbf{b}\in\Omega^L} Q(\mathbf{s}',\mathbf{b}) - Q^\pi(\mathbf{s},\mathbf{p})] \\ &= \mathbf{T}Q^t(\mathbf{s},\mathbf{p}) - Q^\pi(\mathbf{s},\mathbf{p}) \\ &= \mathbf{T}Q^t(\mathbf{s},\mathbf{p}) - \mathbf{T}Q^\pi(\mathbf{s},\mathbf{p}). \end{aligned} \tag{B.3}$$

Then, we can prove that $\mathbb{E}[F^t(\mathbf{s},\mathbf{p})]$ is contractible in the case of infinite norm, as follows

$$\begin{aligned} ||\mathbb{E}[F^t(\mathbf{s},\mathbf{p})]||_\infty &= ||\mathbf{T}Q^t(\mathbf{s},\mathbf{p}) - \mathbf{T}Q^\pi(\mathbf{s},\mathbf{p})||_\infty \\ &= \max_{\mathbf{s},\mathbf{p}} \beta |\sum_{\mathbf{s}'\in\mathcal{S}} \Upsilon_{\mathbf{p}}(\mathbf{s},\mathbf{s}')[\max_{\mathbf{b}\in\Omega^L} Q^t(\mathbf{s}',\mathbf{b}) \\ &\quad - \max_{\mathbf{b}\in\Omega^L} Q^\pi(\mathbf{s}',\mathbf{b})]| \\ &\le \max_{\mathbf{s},\mathbf{p}} \beta \sum_{\mathbf{s}'\in\mathcal{S}} \Upsilon_{\mathbf{p}}(\mathbf{s},\mathbf{s}') \max_{\mathbf{a},\mathbf{b}} |Q^t(\mathbf{a},\mathbf{b}) \\ &\quad - Q^\pi(\mathbf{a},\mathbf{b})| \\ &= \max_{\mathbf{s},\mathbf{p}} \beta \sum_{\mathbf{s}'\in\mathcal{S}} \Upsilon_{\mathbf{p}}(\mathbf{s},\mathbf{s}') \|Q^t - Q^\pi\|_\infty \\ &= \beta \|Q^t - Q^\pi\|_\infty = \beta \|\Delta^t(\mathbf{s},\mathbf{p})\|_\infty. \end{aligned} \tag{B.4}$$

Since the discount rate $\beta \in (0,1)$, the second condition in **Lemma 1** is also satisfied. For the third condition

$$\begin{aligned} \text{var}[F^t(\mathbf{s},\mathbf{p})] &= \mathbb{E}\big[\big(u(\mathbf{s},\mathbf{p},\mathbf{s}') + \beta \max_{\mathbf{b}\in\Omega^L} Q(\mathbf{s}',\mathbf{b}) - Q^\pi(\mathbf{s},\mathbf{p}) \\ &\quad - \mathbf{T}Q^t(\mathbf{s},\mathbf{p}) + Q^\pi(\mathbf{s},\mathbf{p})\big)^2\big] \\ &= \mathbb{E}\big[\big(u(\mathbf{s},\mathbf{p},\mathbf{s}') + \beta \max_{\mathbf{b}\in\Omega^L} Q(\mathbf{s}',\mathbf{b}) \\ &\quad - \mathbf{T}Q^t(\mathbf{s},\mathbf{p})\big)^2\big] \\ &= \text{var}\big[u(\mathbf{s},\mathbf{p},\mathbf{s}') + \beta \max_{\mathbf{b}\in\Omega^L} Q^t(\mathbf{s}',\mathbf{b})\big], \end{aligned} \tag{B.5}$$

Since the value of the system utility u and the Q-table is bounded, the variance in (B.5) is bounded, thus there exists a constant such that

$$\text{var}[F^t(\mathbf{s},\mathbf{p})] \le C(1+||\Delta^t(\mathbf{s},\mathbf{p})||_\infty^2). \tag{B.6}$$

This concludes the proof of the convergence. ■

Appendix C
Proof of Theorem 3

Proof. If the access priority score of each class is identical, or equivalently, the number of user classes is only one, i.e., $L=1$, thus the total number of potential users in the network and the ACB factor can be denoted by $N = N_1$ and $p = p_1$, respectively. Since there is no difference in ACB factors assigned to different classes, the first penalty term of the utility function in (4) disappears, which is given by

$$\begin{aligned} u &= cp_1r_1N_1 - \rho_2(1-c) = cprN - \rho_2(1-c) \\ &= f(p)\,prN - \rho_2(1-f(p)) \\ &= f(p)(prN+\rho_2) - \rho_2, \end{aligned} \tag{C.1}$$

where the ACB factor $p \in [0,1]$, and the function $f(p)$ is defined by a monotonically decreasing convex curve [11] with properties given by

$$\begin{aligned} &\frac{\partial f(p)}{\partial p} \le 0, \frac{\partial^2 f(p)}{\partial p^2} < 0, \\ &\left.\frac{\partial f(p)}{\partial p}\right|_{p\to 0^+} \to 0^-, \\ &f(p)|_{p\to 1^-} \to 0^+. \end{aligned} \tag{C.2}$$

Hence, maximizing the system utility function as given in (B.1) is a convex optimization problem. It can be verified that the first derivative of the system utility u with respect to the action p satisfies

$$\begin{aligned} \left.\frac{\partial u}{\partial p}\right|_{p\to 0^+} &= \frac{\partial f(p)}{\partial p}(prN+\rho_2) + rNf(p) = rNf(0^+) > 0, \\ \left.\frac{\partial u}{\partial p}\right|_{p\to 1^-} &= \frac{\partial f(p)}{\partial p}(prN+\rho_2) + rNf(p) \end{aligned}$$

$$= \frac{\partial f(p)}{\partial p}(rN + \rho_2) < 0. \tag{C.3}$$

The second derivative of the system utility u with respect to the action p is strictly negative, which is as given by

$$\frac{\partial^2 u}{\partial p^2} = \frac{\partial^2 f(p)}{\partial p^2}(prN + \rho_2) + \frac{\partial f(p)}{\partial p}(2rN + \rho_2) < 0. \tag{C.4}$$

Therefore, it can be derived that the first derivative $\frac{\partial u}{\partial p}$ has a unique zero solution within the feasible range of $p \in [0, 1]$. Consequently, the convex function u with respect to p has a maximum value in the interval $p \in [0, 1]$. ■

APPENDIX D
PROOF OF THEOREM 4

Proof. For the convolutional neural network (Conv) layer in the architecture as shown in Fig. 3, the input data can be reshaped into a high-dimensional tensor of size $C_{\mathrm{i}} \times w_{\mathrm{i}} \times w_{\mathrm{i}}$, where C_{i} is the number of input channels of the Conv layer. When the Conv layer has C_{o} output channels with each channel equipped with a convolution kernel of size $v \times v$ and stride s, the number of float-point operations consumed by each Conv layer is calculated by

$$N_{\mathrm{FLOPs}}(\mathrm{Conv}) = C_{\mathrm{i}} w_{\mathrm{i}}^2 C_{\mathrm{o}} w_{\mathrm{o}}^2 \tag{D.1}$$

where w_{o} is the size of each output channel of the Conv layer. According to related research in literature [46], in the zero padding mode, the value of w_{o} is related with the input size, convolution kernel size, and stride as given by

$$w_{\mathrm{o}} = \frac{w_{\mathrm{i}} - v}{s} + 1 \tag{D.2}$$

where the influence of bias is reflected by adding one to the right of (C.2). Since the FLOPs of the remaining two fully connected network (FC) layers is much smaller than that of the Conv layer so that it can be neglected, so we have the computational complexity of each time slot of the DRL-based scheme as given by

$$N_{\mathrm{FLOPs}}(\text{time slot}) = C_{\mathrm{i}} w_{\mathrm{i}}^2 C_{\mathrm{o}} \left(\frac{w_{\mathrm{i}} - v}{s} + 1\right)^2 \tag{D.3}$$

Finally, multiplying the number of FLOPs for each time slot of the DRL-based scheme with the number of episodes and time slots, we can derive (17). ■

REFERENCES

[1] S. Verma, Y. Kawamoto, and N. Kato, "Energy-efficient group paging mechanism for QoS constrained mobile IoT devices over LTE-A pro networks under 5G," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9187–9199, Oct. 2019.

[2] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "ColO-RAN: Developing machine learning-based xApps for open RAN closed-loop control on programmable experimental platforms," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5787–5800, Oct. 2023.

[3] J.-C. Jiang and H.-M. Wang, "Massive random access with sporadic short packets: Joint active user detection and channel estimation via sequential message passing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4541–4555, Jul. 2021.

[4] Z. Gao, L. Dai, S. Han, Z. Wang, and L. Hanzo, "Compressive sensing techniques for next-generation wireless communications," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 144–153, Jun. 2018.

[5] L. Liu and W. Yu, "Massive connectivity with massive MIMO—Part I: Device activity detection and channel estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2933–2946, Jun. 2018.

[6] Y. Noh and S. Hong, "Compressed sensing based active user detection in MIMO systems with one-bit ADC," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1313–1317, Jan. 2023.

[7] J. Ahn, B. Shim, and K. B. Lee, "EP-based joint active user detection and channel estimation for massive machine-type communications," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5178–5189, Jul. 2019.

[8] M. Ke, Z. Gao, Y. Wu, X. Gao, and R. Schober, "Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive MIMO," *IEEE Trans. Signal Process.*, vol. 68, pp. 764–779, 2020.

[9] J.-P. Hong, W. Choi, and B. D. Rao, "Sparsity controlled random multiple access with compressed sensing," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 998–1010, Feb. 2015.

[10] X. Du, D. Wu, W. Liu, and Y. Fang, "Multiclass routing and medium access control for heterogeneous mobile ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 270–277, Jan. 2006.

[11] Y. Sim and D. Cho, "Performance analysis of priority-based access class barring scheme for massive MTC random access," *IEEE Syst. J.*, vol. 14, no. 4, pp. 5245–5252, Dec. 2020.

[12] S. Verma, Y. Kawamoto, H. Nishiyama, N. Kato, and C.-W. Huang, "Novel group paging scheme for improving energy efficiency of IoT devices over LTE-A pro networks with QoS considerations," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[13] H. S. Jang, H. Jin, B. C. Jung, and T. Q. S. Quek, "Versatile access control for massive IoT: Throughput, latency, and energy efficiency," *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1984–1997, Aug. 2020.

[14] M. Centenaro, L. Vangelista, S. Saur, A. Weber, and V. Braun, "Comparison of collision-free and contention-based radio access protocols for the Internet of Things," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3832–3846, Sep. 2017.

[15] O. S. Nishimura, J. C. M. Filho, T. Abrão, and R. D. Souza, "Fairness in a class barring power control random access protocol for crowded XL-MIMO systems," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4574–4582, Sep. 2022.

[16] Z. Yuan, W. Li, Y. Hu, H. Tang, J. Dai, and Y. Ma, "Blind multi-user detection based on receive beamforming for autonomous grant-free high-overloading multiple access," in *Proc. IEEE 2nd 5G World Forum (GWF)*, Dresden, Germany, Sep. 2019, pp. 520–523.

[17] Z. Chen, F. Sohrabi, and W. Yu, "Multi-cell sparse activity detection for massive random access: Massive MIMO versus cooperative MIMO," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4060–4074, Aug. 2019.

[18] L. M. Bello, P. D. Mitchell, and D. Grace, "Intelligent RACH access techniques to support M2M traffic in cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8905–8918, Sep. 2018.

[19] D.-D. Tran, S. K. Sharma, and S. Chatzinotas, "BLER-based adaptive Q-learning for efficient random access in NOMA-based mMTC networks," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–5.

[20] M. V. da Silva, R. D. Souza, H. Alves, and T. Abrão, "A NOMA-based Q-learning random access method for machine type communications," *IEEE Wireless Commun. Lett.*, vol. 9, no. 10, pp. 1720–1724, Oct. 2020.

[21] O. S. Nishimura, J. C. Marinello, and T. Abrão, "A grant-based random access protocol in extra-large massive MIMO system," *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2478–2482, Nov. 2020.

[22] C. Di, B. Zhang, Q. Liang, S. Li, and Y. Guo, "Learning automata-based access class barring scheme for massive random access in machine-to-machine communications," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6007–6017, Aug. 2019.

[23] C. Bockelmann et al., "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sep. 2016.

[24] S. Liu, F. Yang, J. Song, and Z. Han, "Block sparse Bayesian learning-based NB-IoT interference elimination in LTE-advanced systems," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4559–4571, Oct. 2017.

[25] K. Senel and E. G. Larsson, "Grant-free massive MTC-enabled massive MIMO: A compressive sensing approach," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6164–6175, Dec. 2018.
[26] B. Wang, L. Dai, T. Mir, and Z. Wang, "Joint user activity and data detection based on structured compressive sensing for NOMA," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1473–1476, Jul. 2016.
[27] Y. Zhang, Q. Guo, Z. Wang, J. Xi, and N. Wu, "Block sparse Bayesian learning based joint user activity detection and channel estimation for grant-free NOMA systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9631–9640, Oct. 2018.
[28] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, "Grant-free massive random access with a massive MIMO receiver," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2019, pp. 23–30.
[29] F. Morvari and A. Ghasemi, "Two-stage resource allocation for random access M2M communications in LTE network," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 982–985, May 2016.
[30] J. Jiao, L. Xu, S. Wu, Y. Wang, R. Lu, and Q. Zhang, "Unequal access latency random access protocol for massive machine-type communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5924–5937, Sep. 2020.
[31] L. Xiao et al., "Reinforcement learning-based downlink interference control for ultra-dense small cells," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 423–434, Jan. 2020.
[32] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 1st Quart., 2020.
[33] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1424–1440, Jun. 2019.
[34] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, Jun. 2019.
[35] A. H. Bui and A. T. Pham, "Deep reinforcement learning-based access class barring for energy-efficient mMTC random access in LTE networks," *IEEE Access*, vol. 8, pp. 227657–227666, 2020.
[36] Y. Cao, S.-Y. Lien, Y.-C. Liang, K.-C. Chen, and X. Shen, "User access control in open radio access networks: A federated deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 3721–3736, Jun. 2022.
[37] A. Kumar, G. Verma, C. Rao, A. Swami, and S. Segarra, "Adaptive contention window design using deep Q-learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4950–4954.
[38] X. Ye, Y. Yu, and L. Fu, "Multi-channel opportunistic access for heterogeneous networks based on deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 794–807, Feb. 2022.
[39] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.
[40] *Radio Resource Control (RRC); Protocol Specification*, document TS 36.331, V13.0.0, 3GPP, Jan. 2016.
[41] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 4, Stockholm, Sweden, Feb. 2018, pp. 2587–2601.
[42] M. J. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso)," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2183–2202, May 2009.
[43] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is Q-learning provably efficient?" in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 4863–4873.
[44] E. Bjornson, P. Zetterberg, M. Bengtsson, and B. Ottersten, "Capacity limits and multiplexing gains of MIMO channels with transceiver impairments," *IEEE Commun. Lett.*, vol. 17, no. 1, Jan. 2013.
[45] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Comput.*, vol. 6, no. 6, pp. 1185–1201, Nov. 1994.
[46] C. C. T. Mendes, V. Frémont, and D. F. Wolf, "Exploiting fully convolutional neural networks for fast road detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 3174–3179.

Xiao Tang received the B.S. degree in communication engineering from Central South University, Changsha, China, in 2020. He is currently pursuing the M.S. degree with the Department of Information and Communication Engineering, Xiamen University, Xiamen, China. His research interests include compressed sensing and AI-assisted communications.

Sicong Liu (Senior Member, IEEE) received the B.S.E. and Ph.D. degrees (Hons.) in electronic engineering from Tsinghua University, Beijing, China, in 2012 and 2017, respectively. He is currently an Associate Professor with the Department of Information and Communication Engineering, School of Informatics, Xiamen University, China. He has authored over 60 journal and conference papers, and four monographs in the related areas. His current research interests include compressed sensing, AI-assisted communications, integrated sensing and communications, and visible light communications.

Xiaojiang (James) Du (Fellow, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 2002 and 2003, respectively. He was a Professor with Temple University from August 2009 to August 2021. He is currently the Anson Wood Burchard Endowed-Chair Professor of the Department of Electrical and Computer Engineering, Stevens Institute of Technology. His research interests include security, wireless networks, and systems. He has authored over 500 journal and conference papers in these areas, including the top security conferences IEEE S&P, USENIX Security, and NDSS. He is an ACM Distinguished Member and an ACM Life Member. He won the Best Paper Award from IEEE ICC 2020 and IEEE GLOBECOM 2014 and the Best Poster Runner-Up Award from ACM MobiHoc 2014. He has been awarded more than eight million U.S. Dollars in research grants from the U.S. National Science Foundation (NSF), the Army Research Office, the Air Force Research Laboratory, the State of Pennsylvania, and Amazon. He serves on the editorial boards for three IEEE journals.

Mohsen Guizani (Fellow, IEEE) received the B.S. (with distinction), M.S., and Ph.D. degrees in Electrical and Computer Engineering from Syracuse University, Syracuse, NY, USA, in 1985, 1987, and 1990, respectively. He is currently a Professor of Machine Learning at the Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. Previously, he worked in different institutions in the USA. He is the author of 11 books, more than 1000 publications and several U.S. patents. His research interests include applied machine learning and artificial intelligence, smart city, Internet of Things (IoT), intelligent autonomous systems, and cybersecurity. He became an IEEE Fellow in 2009 and was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019, 2020, 2021, and 2022. He has won several research awards including the "2015 IEEE Communications Society Best Survey Paper Award," the Best ComSoc Journal Paper Award in 2021 as well five Best Paper Awards from ICC and Globecom Conferences. He is also the recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 AdHoc Technical Committee Recognition Award, and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award. He served as the Editor-in-Chief for IEEE Network and is currently serving on the Editorial Boards of many IEEE TRANSACTIONS and Magazines. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer.