Article:
Liang, Xuanyu, Al-Tahmeesschi, Ahmed Abdulkareem J, Chetty, Swarna Bindu et al. (2026) Scalable machine learning-based approaches for energy saving in densely deployed Open RAN. IEEE Transactions on Green Communications and Networking. pp. 2710-2722. ISSN: 2473-2400

https://doi.org/10.1109/TGCN.2026.3679951

Scalable machine learning-based approaches for energy saving in densely deployed Open RAN

Xuanyu Liang*, Ahmed Al-Tahmeesschi*, Swarna Chetty*, Cicek Cavdar†, Berk Canberk ‡ and Hamed Ahmadi*
*School of Physics Engineering and Technology, University of York, UK
†School of EECS at KTH Royal Institute of Technology, Sweden.
‡School of Computing, Engineering and The Built Environment, Edinburgh Napier University, UK

***Abstract*—Densely deployed base stations are responsible for the majority of the energy consumed in Radio access network (RAN). While these deployments are crucial to deliver the required data rate in busy hours of the day, the network can save energy by switching some of them to sleep mode and maintain the coverage and quality of service with the other ones. Benefiting from the flexibility provided by the Open RAN in embedding machine learning (ML) in network operations, in this work we propose Deep Reinforcement Learning (DRL)-based energy saving solutions. Firstly we propose 3 different DRL-based methods in the form of xApps which control the Active/Sleep mode of up to 6 radio units (RUs) from Near Real time RAN Intelligent Controller (RIC). We also propose a further scalable federated DRL-based solution with an aggregator as an rApp in None Real time RIC and local agents as xApps. Our simulation results present the convergence of the proposed methods. We also compare the performance of our federated DRL across three layouts spanning 6–24 RUs and 500–1000 m regions, including a composite multi-region scenario. The results show that our proposed federated TD3 algorithm achieves up to 43.75% faster convergence, more than 50% network energy saving and 37. 4% lower training energy versus centralized baselines, while maintaining the quality of service and improving the robustness of the policy.**

***Index Terms*—6G, Open Radio Access Network (O-RAN), Federated Learning (FL), DRL, TD3, Energy Efficiency, Sleep Mode Control**

I. Introduction

To accommodate rapidly increasing mobile traffic, Base Stations (BSs) have been widely deployed to satisfy user data rate requirements. Although dense deployments significantly enhance capacity and coverage, they also lead to substantial energy consumption. Studies indicate that BSs account for a major portion of total network energy usage: roughly a decade ago they drew about 57% of the total, whereas recent figures have risen to 73%–77% [1]. This firmly establishes BSs as the dominant energy consumers in mobile networks. A typical 5G macro BS consumes approximately 3.3–9 kW, with higher values observed for massive MIMO and mmWave deployments. The resulting energy demand increases both the operational expenditure of network providers and the associated greenhouse gas emissions [2]. Consequently, improving energy efficiency in BS operations is increasingly crucial. However, traditional BS designs feature tightly integrated hardware with limited flexibility, constraining selective activation or power adaptation of individual components [3] and leading to persistently inefficient energy usage.

In this context, O-RAN introduces an open, disaggregated, and intelligence-ready architecture that better supports energy-aware control through standardized interfaces [4]. It decomposes the monolithic BS into the Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU); the CU/DU together implement Base Band Unit (BBU) functions and are typically cloud-hosted, while the RU remains a physical entity (akin to a streamlined Remote Radio Head (RRH)) responsible for lower-PHY processing, RF chains, and power amplifiers—components that constitute a major share of O-RAN energy consumption [5]. As RUs are dimensioned for peak demand, placing a larger fraction into sleep mode during off-peak periods can yield substantial savings provided Quality of Service (QoS) constraints are respected [6]. O-RAN further introduces the RAN Intelligent Controller (RIC) as a programmable control platform with two layers: the Near Real Time RIC (Near-RT RIC) (control loop $\sim$10–1000 ms) hosting third-party xApps for near-real-time radio control (e.g., traffic steering [7], network slicing [8] and latency sensitive control [9]) and the Non Real Time RIC (Non-RT RIC) (timescales $\geq$1 s) orchestrating policies and analytics through rApps [10]. The Non-RT RIC provides high-level guidance and long-term optimization goals to the Near-RT RIC via the standardized A1 interface [11]. In turn, the Near-RT RIC executes time-sensitive control by running xApps that translate these A1 policies into actionable decisions and interact with O-DU/O-RU nodes through the E2 interface [12]. This hierarchical workflow enables rApps to supply long-term network-wide intelligence (e.g., predicted traffic trends or RU utilization expectations), while xApps react to instantaneous conditions to enforce fine-grained radio control. The whole workflow is illustrated at Fig. 1. Building on this architecture, AI-driven solutions are increasingly deployed in O-RAN to support intelligent and adaptive control at both rApps and xApps [13]. Several prior studies and O-RAN specification documents have analyzed signaling latency, control overhead, and RIC interaction models under similar functional splits. Our framework adheres to these established timelines and control abstractions. In our work, we develop decentralized Machine Learning (ML)-based mechanisms in the form of an xApp that leverages rApp-provided policy constraints to intelligently schedule RUs into sleep mode while maintaining user QoS.

A. Related Work

Early efforts to reduce energy consumption in cellular networks focused on BS sleep-mode control. When BSs are densely deployed for peak traffic, keeping them active during off-peak periods leads to substantial energy waste. Turning off lightly loaded sites—a strategy commonly referred to as *sleep mode*—has therefore been widely studied [14]–[17]. For instance, [15] introduced a random sleep policy for small-cell BSs, while [16] proposed a sequential deactivation algorithm that preserves user rates. In [17], several schemes were analyzed for macro-BS deactivation in heterogeneous networks, maintaining coverage via power readjustment and complementary micro layers. Although such optimization-based approaches can be effective, their scalability degrades with growing network size and multi-objective constraints, leading to high computational complexity and long solution times.

The sleep/active mode decision problem is NP-hard due to the combinatorial on/off choices and temporal coupling, which makes exact optimization intractable at scale. To address this challenge while satisfying QoS constraints, ML-based strategies have been widely studied, particularly in Ultra Dense Network (UDN) scenarios where spatial redundancy enables flexible sleeping. Paper [18] employs an Long Short-term Memory (LSTM)-based predictor to capture temporal patterns in traffic and channel variations, enabling proactive on/off switching. In [19], a Deep Q-Learning Network (DQN) framework is enhanced with an action-selection network that filters out invalid actions and mitigates ineffective exploration. The work in [20] introduces a modular Deep Reinforcement Learning (DRL) pipeline that integrates a Deep Deterministic Policy Gradient (DDPG) policy with a supervised cost estimator and traffic predictor to account for switching costs and QoS degradation. Spatio–temporal correlations are further leveraged in [21], where convolutional-LSTM forecasting is coupled with an actor–critic controller to improve robustness under fluctuating traffic. A different direction is explored in [22], which formulates joint sleep control and renewable-energy sharing as a single Markov Decision Process (MDP), solved through a multi-discrete Proximal Policy Optimization (PPO) that factorizes the exponential action space into tractable subspaces. Paper [23] presents a PPO-based approach that simultaneously considers sleep scheduling, cell zooming, user association, and Reconfigurable Intelligent Surface (RIS) configuration, thus addressing multi-cell coordination under mixed discrete/continuous actions. Risk-aware mechanisms have also been investigated. In [24], a digital twin is integrated with a DQN controller to assess delay risk before executing sleep decisions, triggering re-training or feature suppression under anomalous traffic conditions. From a multi-agent perspective, [25] treats each BS as an agent and applies a multi-agent PPO with a state-similarity heuristic, jointly optimizing BS sleeping and MIMO antenna operations while reducing oscillatory behavior. Recent works have explored heuristic and learning-based Radio Card (RC) switching strategies within the O-RAN framework. For example, [26] proposed heuristic xApps for RC ON–OFF control under static user deployment, where User Equipments (UEs) were assumed to be non-mobile and decisions were made based on instantaneous load and RSS thresholds. While such approaches demonstrate energy savings in snapshot-like scenarios, the decision rules are manually designed and rely on fixed thresholds. Subsequently, [27] introduced a DQN-based xApp under a similar static network setting and showed that learning-based policies outperform the heuristic model, even in stationary environments. This indicates that the RC activation problem is inherently combinatorial and difficult to fully capture through fixed rule-based logic. Both lines of work primarily consider static or quasi-static user distributions. In practical deployments, user mobility and traffic fluctuations introduce temporal dependencies, transforming the RC switching problem from a one-shot optimization into a sequential control problem with long-term consequences.

However, centralized ML approaches typically collect training data at a single server, which raises scalability concerns in large, distributed systems: as RUs/BSs and UEs increase, the communication burden grows rapidly and data heterogeneity across locations hinders convergence and generalization [28]. FL addresses these issues by training locally and aggregating model updates instead of raw data, reducing backhaul load and improving locality adaptation [29]. Nevertheless, Federated Reinforcement Learning (FRL) is sensitive to client heterogeneity; inconsistent local policies can lead to conflicting updates and a less effective global policy [30].

Applying federated DRL to sleep control has shown promise. For example, [31] lets each Small Base Station (SBS) train a local Double Deep Q-Learning Network (DDQN) and periodically aggregates at a macro BS. It treats each SBS as an isolated agent, resembling a loosely coupled multi-agent system; and it aggregates at the macro-site level, which is agnostic to the functional split and control timescales defined in O-RAN. In contrast to prior cellular-centric FL/FRL designs, O-RAN introduces a native control hierarchy (Non-RT RIC for long-timescale policy/orchestration and Near-RT RIC for near-real-time control). Moreover, RUs are major energy consumers in O-RAN due to RF and PA hardware. Our work aligns the learning workflow with this hierarchy and shifts the agent granularity from per-SBS to per-region (i.e., an area comprising multiple RUs and UEs). This broader spatial/temporal context improves policy generalization across heterogeneous regions while remaining faithful to O-RAN interfaces and roles. As a result, the proposed federated deep reinforcement learning framework aggregates richer, more representative local policies and scales better than centralized training in large, distributed O-RAN deployments.

B. Contributions

In this paper, we investigate the energy-efficient operation in the O-RAN architecture, which, due to provisioning for peak demand, often keeps RUs active even during low traffic periods, leading to unnecessary energy consumption. We formulate the joint RU sleep control problem as an NP-hard network-wide energy minimization task under QoS constraints, and transform it into a MDP to enable sequential decision-making

via DRL. To address the challenge of large action spaces, we develop three centralized DRL schemes: (i) a DQN-Multiple Action (DQNMA) baseline, which enumerates all joint RU configurations; (ii) a DQN-Single Action (DQNSA) variant, which sequentially switches the state of only one RU per step; and (iii) a Twin Delayed Deep Deterministic Policy Gradient (TD3) controller, which outputs continuous control values, thereby avoiding exponential action growth. Although these centralized approaches are effective in small or medium scale deployments, their scalability is limited in large and geographically distributed networks. To overcome this, The proposed Fed-DRL architecture provides scalable learning across large O-RAN deployments by enabling each area to train its policy locally while only exchanging neural-network parameters with a global aggregator. This avoids raw data sharing, eliminates the computational bottleneck of centralized training. During execution, each xApp independently makes RU activation decisions based on local observations, further improving scalability. The main contributions of this work are summarized as follows:

- We propose an O-RAN compliant energy control framework in which RUs dynamically switch between active and sleep modes based on traffic and mobility patterns. The design leverages the O-RAN functional split and standardized control interfaces for practical deployment.
- We formulate the activation of RU as MDP considering traffic load, user mobility, and spatial distribution. The reward function incorporates empirically validated energy and QoS tradeoffs to ensure stable and meaningful learning behavior. This formulation is compatible with the O-RAN learning framework aligned with multiple DRL paradigms, supporting both centralized and distributed solutions.
- We propose a federated DRL architecture tailored for O-RAN, where local agents embedded in the Near-RT RIC interact with regional RUs and UEs, and their models are periodically aggregated at the Non-RT RIC. This architecture is consistent with the disaggregated control roles of O-RAN, enabling real-time local responsiveness while ensuring global policy consistency and coordination across regions. By explicitly embedding FL into the O-RAN control framework, our design provides a scalable and communication-efficient solution for large and heterogeneous network deployments.
- We conduct extensive simulations across multiple network layouts and heterogeneous regions. The results show that the proposed Fed-DRL achieves significant network energy savings and scales effectively to large deployments. Moreover, the federated-learning–based solution also reduces the computational and communication overhead associated with centralized training, thereby lowering the overall training energy consumption.

II. System Model

In this section, we describe the system model of the O-RAN-based wireless network under consideration. The network consists of M distributed RUs, each equipped with a single antenna, serving K single-antenna UEs in the downlink. The sets of RUs and UEs are denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$, respectively. We assume a multi-area deployment, where RUs are geographically distributed and managed by corresponding near-real-time RIC instances as illustrated on Fig. 1. The system operates in discrete time slots and the UEs are mobile with a constant speed v, following a random direction mobility model. Each UE is associated with one RU based on signal strength and proximity. To model energy control at the RU level, we define a binary variable $\alpha_m \in \{0, 1\}$ for each RU $m \in \mathcal{M}$. Specifically, $\alpha_m = 1$ indicates that the RU m is in active mode and capable of transmitting data, while $\alpha_m = 0$ represents that the RU is in sleep mode and does not consume transmission power during the interval.

In our system, each RU $m \in \mathcal{M}$ is equipped with a total of Q_m Physical Resource Blocks (PRBs), which can be allocated to its associated UEs. Let U_m^t denote the number of UEs associated with RU m at time slot t. We define $n_{m,k}^t$ as the number of PRBs allocated by RU m to the k-th UE at time t, where $k \in \mathcal{K}$. Based on this allocation, the total number of PRBs utilized by RU m at time t, denoted as N_m^t, can be expressed as:

$$N_m^t = \sum_{k=1}^{U_m^t} n_{m,k}^t. \tag{1}$$

The PRB allocation $n_{m,k}^t$ for each UE is determined according to its individual data rate requirement at time t, ensuring that the allocated resources are sufficient to meet the minimum QoS demands. Based on the PRB allocation, we define the load of RU m at time slot t as the ratio of occupied PRBs to the total available PRBs:

$$l_m^t = \frac{N_m^t}{Q_m}. \tag{2}$$

To model the wireless channel, we adopt the Urban Microcel (UMi) fading model. Let $h_{m,k}^t$ denote the small-scale fading channel gain between RU m and UE k at time slot t. We consider a power allocation scheme where the total transmission power of each RU, denoted as P_{TX}, is evenly distributed across all its available PRBs. Consequently, the transmission power allocated to a UE is proportional to the number of PRBs it receives. Under this model, the received Signal-to-Noise Ratio (SNR) at UE k served by RU m at time t is given by:

$$SNR_{m,k}^t = \frac{P_{\mathrm{TX}} \cdot h_{m,k}^t \cdot \left(\frac{n_{m,k}^t}{Q_m}\right)}{n_{m,k}^t \cdot N_0}, \tag{3}$$

where N_0 represents the noise power spectral density of the Additive White Gaussian Noise (AWGN), and the term $\frac{n_{m,k}^t}{Q_m}$ reflects the fraction of total RU power allocated to UE k based on its PRB share. The achievable downlink data rate for UE k at time t is then expressed by the Shannon capacity formula as:

$$R_k^t = n_{m,k}^t \cdot B \cdot \log_2\left(1 + SNR_{m,k}^t\right), \tag{4}$$

where B denotes the bandwidth of a single PRB.

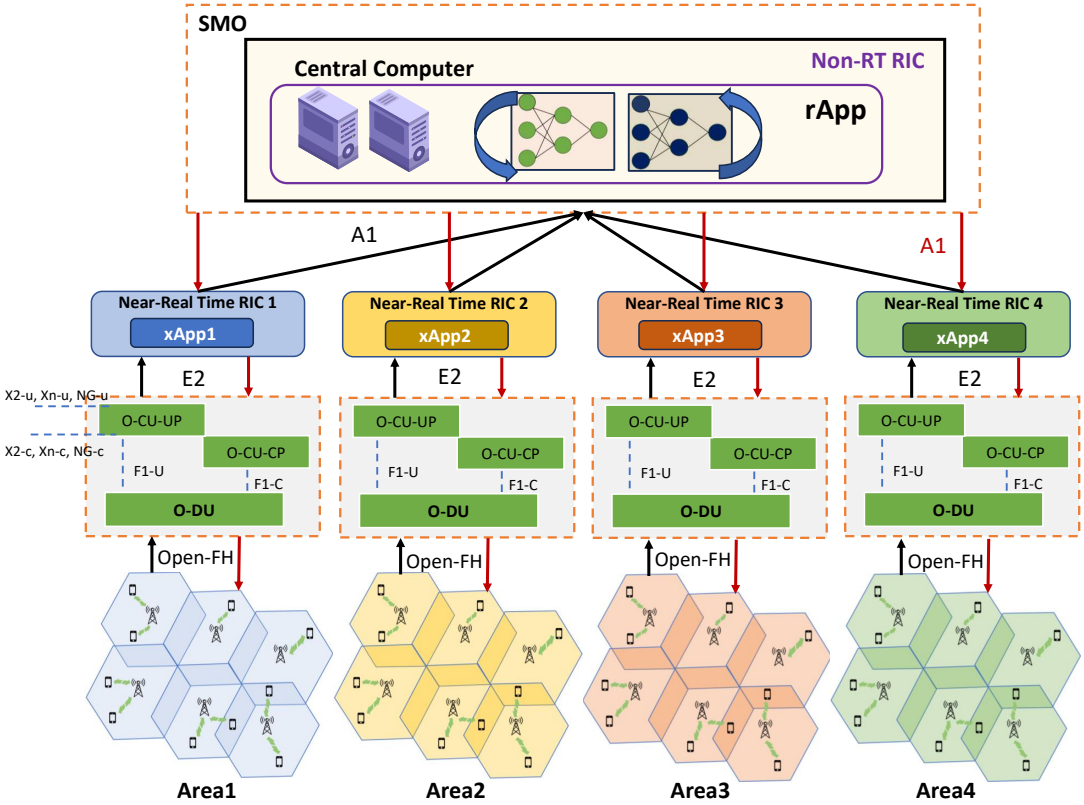

Fig. 1: Illustration of the O-RAN architecture incorporating both O-RAN-defined and 3GPP-standard interfaces. Solid lines represent O-RAN interfaces (e.g., E2, A1, O1, Open FH), while dashed lines indicate 3GPP interfaces.The system spans four geographical areas, each with its own O-DU and O-CU components. Near-RT RICs operate on a per-area basis, deploying multiple xApps. A centralized Non-RT RIC performs global policy aggregation and training coordination using A1 interface.

TABLE I: Summary of Symbols

Notations	Value
α_m	Mode variable of RU m
n_m, k	Number of allocated PRBs from RUm to UE k
N_m	Number of PRBs allocated in RU m
Q_m	Total number of PRBs in RU m
U_m^t	Total number of UEs associate with RU m at time slot t
l_m	Load of RU m
$h_{m,k}$	UMi fading channel model from the RU m to the UE k
$SNR_{m,k}$	SNR of the UE m in RU k
$R_{d,k,min}$	Data rate requirement of UE k
$R_{d,k}$	Current data rate requirement of UE k
η	Power amplifier efficiency
P_{TX}	Maximum transmission power of RU
V_m^{trans}	Mode transition power of RU m
P_m^t	Energy consumption of RU m at time slot t
P_m^{active}	Fixed power consumption of RU m in active mode
P_m^{sleep}	Fixed power consumption of RU m in sleep mode
P_m^{data}	Load dependent power consumption of RU m
P_{tot}^t	Total network energy consumption at time slot t

A. Power Consumption Model and Problem Formulation

The power consumption of each RU is composed of three components [19]. The first is fixed power consumption, denoted by $P_m^{\text{Fix},t}$, which represents the energy used by signal processing, cooling systems, and power supply units. This component depends solely on the operational mode of the RU. The second component is the load-dependent power consumption, denoted by $P_m^{\text{data},t}$, primarily attributed to the Power Amplifier (PA). In this work, the load is quantified based on the PRB utilization, as defined in (2). The third component is the transition power $P_m^{\text{trans},t}$, which is incurred only when the RU switches from sleep mode to active mode. This represents the additional energy required to re-activate hardware components such as the signal processing unit and power amplifier. By summing these three components, the total power consumption of RU m at time slot t is given by:

$$P_m^t = P_m^{\text{Fix},t} + P_m^{\text{data},t} + P_m^{\text{trans},t}. \tag{5}$$

As discussed earlier, the fixed power consumption of an RU depends solely on its operational mode (active or sleep), and remains constant regardless of traffic load. It can be modeled as:

$$P_m^{\text{Fix},t} = \alpha_m^t P_m^{\text{active}} + (1-\alpha_m^t) P_m^{\text{sleep}}, \tag{6}$$

where $\alpha_m^t \in \{0,1\}$ is the binary activity indicator of RU m at time t as we mentioned above. Here, P_m^{active} and P_m^{sleep} represent the fixed power consumption levels of RU m in active and sleep modes, respectively. Generally, P_m^{sleep} is significantly lower than P_m^{active}, as many hardware modules are turned off in sleep mode.

The second component of power consumption is the transmission power, which is proportional to the load of the RU.

Specifically, the transmission power of RU m at time slot t is modeled as:

$$P_m^{\mathrm{data},t} = \alpha_m^t \cdot \frac{P_{\mathrm{TX}}}{\eta} \cdot l_m^t = \alpha_m^t \cdot \frac{P_{\mathrm{TX}}}{\eta} \cdot \frac{N_m^t}{Q_m}, \tag{7}$$

where P_{TX} denotes the maximum transmission power of the RU, and $\eta \in (0, 1]$ is the power amplifier (PA) efficiency. Since power amplifiers are not ideal, only a fraction η of the consumed electrical power is converted into radiated signal power, and the rest is dissipated as heat. In most case, only half of the consumed power contributes to actual transmission. The power consumption increases with the RU load l_m^t, which is defined based on PRB utilization as described in (2). This formulation ensures that higher load levels lead to higher transmission power consumption.

The third component is the transition power, which is incurred when the RU switches its operational state. It is defined as:

$$P_m^{\mathrm{trans},t} = \left|\alpha_m^t - \alpha_m^{t-1}\right| \cdot V_m^{\mathrm{trans}}, \tag{8}$$

where α_m^{t-1} indicates the state of RU m in the previous time slot, and V_m^{trans} denotes the energy cost associated with transitioning from sleep mode to active mode. In this work, we consider transition power only when the RU is turned on (i.e., from sleep to active), while mode deactivation (active to sleep) is assumed to incur negligible overhead.

Finally, the total power consumption of the entire network in time slot t is defined as:

$$\begin{aligned} P_{\mathrm{tot}}^t &= \sum_{m=1}^{M} (P_m^{Fix,t} + P_m^{data,t} + P_m^{trans,t}) \\ &= \sum_{m=1}^{M} (\alpha_m^t P_m^{active} + (1-\alpha_m^t) P_m^{sleep}) + \alpha_m^t \frac{P_{TX}}{\eta} * \frac{N_m^t}{Q_m} \\ &\quad + (\left|\alpha_m^t - \alpha_m^{t-1}\right| * V_m^{trans}). \end{aligned} \tag{9}$$

The objective of network is to minimize the total energy consumption over a given time horizon by dynamically controlling the sleep mode of RUs, while ensuring QoS requirements are satisfied. At the initial time slot ($t = 0$), all RUs are assumed to be active. The optimization problem is formulated as follows:

$$\begin{aligned} \mathcal{P}_1 : \quad \min_{\{\alpha_m^t, n_{m,k}^t\}} \quad & \sum_{t=1}^{T} P_{\mathrm{tot}}^t \\ \text{s.t.} \quad & R_{d,k}^t \geq R_{d,k}^{\min}, \quad \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \\ & N_m^t \leq Q_m, \quad \forall m \in \mathcal{M},\ \forall t \in \mathcal{T} \\ & \alpha_m^t \in \{0,1\}, \quad \forall m \in \mathcal{M},\ \forall t \in \mathcal{T} \\ & \sum_{m \in \mathcal{M}} \delta_{m,k}^t = 1, \quad \forall k \in \mathcal{K},\ \forall t \in \mathcal{T}, \end{aligned} \tag{10}$$

where $R_{d,k}^t$ denotes the achieved downlink rate of UE k at time t, which must satisfy a minimum QoS threshold $R_{d,k}^{\min}$. N_m^t is the total number of PRB allocated by RU m and must not exceed its available PRB budget Q_m. while $\delta_{m,k}^t \in \{0,1\}$ is an association indicator that ensures that each UE is connected to exactly one RU at any given time. Problem ($\mathcal{P}_1$) aims to minimize the cumulative network energy over a horizon T, subject to QoS. Solving this objective with linear or mixed integer formulations in every time interval t would require repeatedly recomputing a large combinatorial program in changing channels and user mobility. Even with relaxations or Model Predictive Control (MPC)-style solvers, the computational burden grows quickly with the number of RUs and must be incurred at each slot. Moreover, per-slot solutions that optimize instantaneous energy P_{tot}^t cannot anticipate future handovers or load variations, often leading to excessive switching and degraded long-term performance. We therefore cast ($\mathcal{P}_1$) as a MDP and learn a policy $\pi(a|s)$ that directly maps the current network state s to RU sleep/active action vector a with the goal of optimizing the long-horizon return. After offline training, online control reduces to a single forward pass of the neural network per slot, which is orders of magnitude lighter than resolving a combinatorial program, while naturally adapting to user mobility since the state already encodes time-varying positions and loads.

III. DRL-based Energy Saving Mechanisms

In this section, we develop a centralized DRL framework aimed at minimizing the overall energy consumption in O-RAN networks by intelligently controlling the sleep mode of distributed RUs. Specifically, the proposed approach seeks to determine an optimal activation policy that dynamically switches RUs between active and sleep modes, while simultaneously guaranteeing the QoS requirements of all UEs in the network.

To the end, we propose three DRL algorithms: DQNSA, DQNMA, and TD3. While DQN-based methods are well-suited to discrete and low-dimensional action spaces, they become less effective in high-dimensional environments with complex combinatorial action structures. In contrast, TD3, which extends the DDPG algorithm, offers enhanced performance in continuous control settings through the use of twin critic networks, delayed policy updates, and target smoothing mechanisms. These features make TD3 particularly suitable for jointly controlling the binary activation states of multiple RUs in a temporally and spatially dynamic wireless environment.

The remainder of this section first outlines the formal MDP formulation of the RU sleep optimization problem, followed by detailed descriptions of the employed DRL algorithms within the centralized control framework.

A. Markov Decision Process Problem

In this subsection, we discuss the state space, action space, and reward function of DRL model to consist the MDP problem.

1) State Space: state comprises essential information used for policy training. At each time step t, it contains the real data rate of the UEs , represented as:

$$\mathbf{R}_d^t = \left[R_{d,1}^t, R_{d,2}^t, \ldots, R_{d,K}^t\right]^T, \tag{11}$$

which captures the system's ability to meet user demands. The activation states of the radio units (RUs) from the previous time step are denoted as:

$$\boldsymbol{\alpha}^{t-1} = \left[\alpha_1^{t-1}, \alpha_2^{t-1}, \ldots, \alpha_M^{t-1}\right]^T. \tag{12}$$

The PRB utilization of the RUs is given by:

$$\mathbf{L}^t = \left[l_1^t, l_2^t, \ldots, l_M^t\right]^T, \tag{13}$$

quantifying the load on the RUs. Finally, the spatial positions of the UEs in the network are represented as:

$$\mathbf{U}^t = \left[(x_1^t, y_1^t), (x_2^t, y_2^t), \ldots, (x_K^t, y_K^t)\right]^T, \tag{14}$$

characterized by their (x, y) coordinates. The UE coordinates determine path loss and association and thus affect data rates and RU loads. Including them in s_t enables the agent to capture the spatial distribution and mobility of UEs. This allows the learned policy to anticipate future handovers, traffic shifts between RUs, and coverage-critical regions, instead of reacting only to instantaneous loads.

In summary, the state can be expressed as:

$$s_t = \left[\mathbf{R}_d^t, \boldsymbol{\alpha}^{t-1}, \mathbf{L}^t, \mathbf{U}^t\right]^T. \tag{15}$$

All state parameters are normalized to facilitate a better interpretation by the agent.

2) Action Space: action $\alpha \in \mathcal{A}$ represents the binary operational states of all RUs, defined as:

$$\mathcal{A} = [\alpha_1, \alpha_2, \ldots, \alpha_M]. \tag{16}$$

Each element α_m corresponds to an RU, where $\alpha_m = 1$ denotes that the RU is active, and $\alpha_m = 0$ indicates that the RU is in sleep mode. These actions are derived from the output of the DRL models, which guides the energy-efficient operation of the network.

3) Reward Function: reward $r \in \mathbb{R}$ is designed to balance energy efficiency and user satisfaction, guiding the model towards an optimal operational policy. It is defined as:

$$r = -w_1 \cdot \frac{P_{\text{tot}}}{P_{\text{max}}} - w_2 \cdot \frac{K_{\text{unsat}}}{K}, \tag{17}$$

where P_{max} corresponds to the energy consumption if all RUs are all active. The term K_{unsat} denotes the number of users with data rates below their required thresholds, normalized by the total number of users K. The weights w_1 and w_2 adjust the relative importance of energy efficiency and user satisfaction. Normalization is used in the reward function to ensure that the contributions of energy consumption and user satisfaction are scaled to comparable ranges, preventing dominance by one factor over the other. The original RU activation problem Eq (10) can be viewed as minimizing network energy subject to minimum rate constraints. To make the problem tractable within the MDP framework, we adopt a penalty-based relaxation in which QoS violations are incorporated into the reward function. This corresponds to a Lagrangian relaxation of the constrained optimization problem, where the penalty weight w_2 acts as a Lagrange multiplier. Under sufficiently large w_2, the relaxed objective discourages persistent violations and approximates the constrained solution in practice, while maintaining stable gradients for DRL training. During training, multiple weight settings such as $(1, 1)$, $(1, 3)$, and $(1, 10)$ were evaluated. The final choice $(1, 5)$ was selected because it provides a reasonable balance between energy-saving exploration and QoS protection, avoiding both overly conservative and overly aggressive behavior. We note that this tradeoff is observed under the considered simulation configurations. Overall, At each time slot t, the agent observes the state s_t, consisting of user data rates, RU loads, RU activation modes, and UE coordinates. Based on this state, the policy $\pi(a_t|s_t)$ outputs an RU activation vector a_t. The environment then evolves according to user mobility and traffic dynamics, producing the next state s_{t+1} and the reward r_t, which jointly reflect both energy consumption and QoS satisfaction. The agent updates the policy to maximize the long-term cumulative reward, thereby learning optimal RU sleep/active decisions.

B. DQN-Based Model Training

The DQN framework extends traditional Q-learning by employing a deep neural network to approximate the action–value function $Q(s, a; \theta)$, where θ denotes the network parameters. Instead of maintaining a tabular representation, the DQN maps input states to Q-values for all possible actions, enabling decision-making in high-dimensional state spaces. During training, the network parameters are updated to minimize the temporal-difference loss:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[\big(r + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta)\big)^2\Big], \tag{18}$$

where θ' is the target network parameter vector, γ is the discount factor, (s, a, r, s') are sampled from the replay buffer D, and the maximization is taken over the discrete action set. The use of a target network and replay experience helps stabilize training by breaking correlations in sequential samples and preventing oscillations.

Building on this framework, we applied DQN to the RU sleep control problem as our initial discrete action solution. In this setting, the agent takes the current network state as input and outputs Q-values for possible RU activation patterns, selecting the action with the highest estimated value. This formulation allows the agent to learn sleep/active scheduling policies without requiring exhaustive optimization at every time step, thus avoiding the computational burden of traditional methods. To investigate how action representation impacts scalability and learning efficiency, we developed two variants: a DQNMA, where each action encodes the joint activation state of all RUs, and a DQNSA, where each action controls the state of only one RU at a time. These two designs highlight the trade-off between expressiveness and tractability in DQN-based sleep control.

1) Multiple-Action DQN (DQNMA): In the DQNMA model, each action represents a complete sleep/active configuration of the entire RU set. Given M RUs, the total number of possible configurations is 2^M. Instead of representing each configuration as a binary vector, we encode each configuration as an integer index:

$$\mathcal{A}_{\text{multi}} = \{0, 1, \ldots, 2^M - 1\} \tag{19}$$

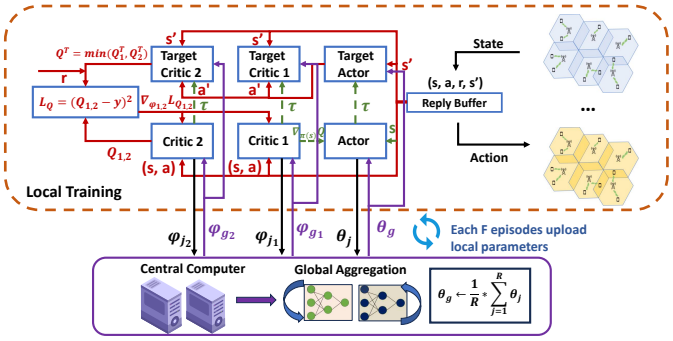

Fig. 2: Illustrates the workflow of Fed-TD3 across distributed agents and a aggregator. Red solid lines indicate local critic training based on temporal-difference loss; green dashed lines represent actor updates guided by critic gradients and soft update target network; black and purple solid lines denote the periodic aggregation and redistribution of actor and critic parameters via the aggregator. Each agent operates within its own local environment and contributes to a globally coordinated policy through federated learning.

Each action $a \in \mathcal{A}_{\text{multi}}$ is mapped to a binary vector of length M, where the i-th bit determines whether RU i is active (1) or asleep (0). For example, in a system with $M = 6$, the action $a = 42$ corresponds to the binary vector $[1, 0, 1, 0, 1, 0]$, indicating the activation states of all six RUs.

This representation is compact and facilitates Q-value lookup through a single scalar action input. However, the action space still grows exponentially with M, making the Q-table (or output layer of the DQN) increasingly difficult to train. As M increases, it becomes challenging for the agent to explore all possible configurations effectively, leading to slower convergence and suboptimal learning.

2) Single-Action DQN (DQNSA): To mitigate the scalability issue inherent in DQNMA, we propose a single-action DQN variant that limits control to one RU per time step. The action space is defined as:

$$\mathcal{A}_{\text{single}} = \{\text{ON}_1, \ldots, \text{ON}_M, \text{OFF}_1, \ldots, \text{OFF}_M\} \tag{20}$$

resulting in a total of $2M$ actions. Each action explicitly represents turning ON or OFF a specific RU. This reduces the action space from exponential to linear size, greatly simplifying the learning problem and accelerating convergence. However, the agent's limited per-step control restricts its ability to rapidly adapt to network-wide changes, potentially resulting in suboptimal policies in highly dynamic environments.

C. TD3 Model Training

Although DQN-based approaches, as we mentioned above, provide effective solutions for discrete RU control, they face significant limitations when applied to large-scale environments with many RUs. In such scenarios, the action space grows exponentially with the number of controllable RUs, making it increasingly difficult for value-based methods to explore and learn optimal policies efficiently. To address these challenges, we adopt a continuous action approach using the TD3 algorithm as shown in Algorithm 1. By modeling the control decisions in a continuous space, TD3 allows the agent to express more accurate preferences over RU activation and significantly reduces the combinatorial complexity of policy learning.

In our adaptation of TD3, the actor network outputs a continuous action vector $\mu_\theta(s) \in [0, 1]^M$, where each element represents the activation likelihood of an RU. To enforce discrete sleep/active decisions in the environment, we apply a deterministic thresholding function with Gaussian exploration noise:

$$a_i = f_d(\mu_\theta(s)_i + \eta_i), \quad a_i = \begin{cases} 1 & \text{if } \mu_\theta(s)_i + \eta_i > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where, $i \in \{1, \ldots, M\}$ indexes the RUs, η_i is a Gaussian noise vector with elements $\eta_i \sim \mathcal{N}(0, \sigma^2)$, adding more exploration in the action decision. $f_d(\cdot)$ denotes the discretization mapping from continuous actor outputs to binary RU activation decisions.This approach enables the policy to explore efficiently in a smoothed action space while still producing discrete activation patterns compatible with RU switching.

The environment includes realistic features such as user mobility, handovers, and RU-specific energy models. Transitions (s, a, r, s') are collected in a replay buffer, and the reward function jointly captures energy efficiency and QoS satisfaction. We use two critic networks Q_{ϕ_1} and Q_{ϕ_2} to compute target Q-values and mitigate overestimation bias:

$$y = r + \gamma \min_{i=1,2} Q_{\phi_i}(s', f_d(\mu_{\theta'}(s') + \epsilon)), \tag{22}$$

where $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma^2_{\text{target}}))$ provides target smoothing. The actor is updated using the deterministic policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\mathcal{D}}[\nabla_a Q_\phi(s, a)|_{a=\mu_\theta(s)} \cdot \nabla_\theta \mu_\theta(s)]. \tag{23}$$

Following TD3's principle of stability, we delay actor and target network updates relative to the critics, and employ soft target updates:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta', \quad \phi'_i \leftarrow \tau\phi_i + (1 - \tau)\phi'_i. \tag{24}$$

While our TD3-based approach helps the large discrete action space problem by leveraging continuous policy learning and threshold-based execution also provides better results as we show in the simulation section, it does not fully eliminate the scalability limitations of centralized training. As the number of RUs increases, for example in ultra-dense urban environments, the centralized controller continues to encounter significant challenges related to computational complexity, communication overhead, and limited policy generalization. To address these issues, we further explore a federated DRL framework in the next section.

IV. Federated DRL for Scalable RU Control in O-RAN

The modular and disaggregated architecture of O-RAN makes it a natural fit for federated DRL, particularly in scenarios that involve large-scale and geographically distributed RU

Algorithm 1: TD3-Based Energy-Aware RU Training

Input: Initial network state s_0, actor parameters θ, critic parameters ϕ_1, ϕ_2, replay buffer $\mathcal{D}$
Output: Trained actor network μ_θ for RU sleep scheduling

for *each episode* **do**
  Initialize environment and receive initial state s ;
  for *each time step* $t = 1, \ldots, T$ **do**
    Add exploration noise $\eta_t \sim \mathcal{N}(0, \sigma^2)$;
    Compute continuous action: $a_c = \mu_\theta(s) + \eta_t$;
    Discretize: $a = f_d(a_c)$ where $a_i = \mathbb{I}[a_{c,i} > 0.5]$;
    Execute action a in the environment ;
    Receive reward r and next state s' ;
    Store (s, a, r, s') into replay buffer $\mathcal{D}$;
    Sample a mini-batch (s_j, a_j, r_j, s'_j) from $\mathcal{D}$;
    Compute target action: $\tilde{a}'_j = f_d(\mu_{\theta'}(s'_j) + \epsilon)$;
    Compute target Q-value: $y_j = r_j + \gamma \min_{i=1,2} Q_{\phi'_i}(s'_j, \tilde{a}'_j)$;
    Update critics ϕ_1, ϕ_2 by minimizing: $\mathcal{L}(\phi_i) = \frac{1}{N} \sum_j \left(Q_{\phi_i}(s_j, a_j) - y_j\right)^2$;
    if *every* d *steps* **then**
      Update actor using deterministic policy gradient ;
      Soft-update target networks: $\phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i$, $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$;
    $s \leftarrow s'$

Algorithm 2: Federated TD3 Training Process

Require: Multi-region environment with R regions, local TD3 agents $\{A_i\}_{i=1}^{R}$, aggregation frequency F and TD3 hyperparameters.
Initialization:
for $i = 1$ **to** R **do**
  Initialize local agent A_i with its actor, critics, and target networks.
end for
Initialize the global model parameters θ_{global}, $\phi_{\text{global\_1}}$ and $\phi_{\text{global\_2}}$ with random value. Establish the connection between local agents and a global server.
Training Loop:
for episode $= 1$ **to** N_{eps} **do**
  for $j = 1$ **to** R **do**
    Initialize state $s_0^{(i)}$.
    for $t = 0$ **to** $T - 1$ **do**
      Agent A_i selects an action $a_t^{(i)}$
      Execute actions $a_t^{(i)}$ in the environment.
      Obtain next states $s_{t+1}^{(i)}$ and rewards $r_t^{(i)}$.
      Agent A_i stores transition $(s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)})$.
      Agent A_i performs a local TD3 update.
      if $(t+1) \bmod F = 0$ **then**
        Global Server Aggregation (FedAvg). Agent A_i sends θ_i, ϕ_{i_1} and ϕ_{i_2} to the global server.
      end if
      Update state: $s_t^{(i)} \leftarrow s_{t+1}^{(i)}$.
    end for
  end for
end for
Global Server Aggregation (FedAvg):
the The global server receives local parameters from all agents.
compute the weighted average of actor and critic model update:
$\theta_{\text{avg}} = \sum_j (\omega_j \cdot \theta_j / \sum_j \omega_j)$
$\phi_{\text{avg\_}i} = \sum_j (\omega_{j\_i} \cdot \phi_{j\_i} / \sum_{j\_i} \omega_{j\_i}), \quad i = 1, 2$. Where ω_i are weight factors
Global Model Update:
Update the global actor and critic model parameters:
$\theta_{\text{global}} = \theta_{\text{avg}}, \quad \phi_{\text{global\_}i} = \phi_{\text{avg\_}i}, \quad i = 1, 2,$
$\theta'_{\text{global}} = \theta_{\text{global}}, \quad \phi'_{\text{global\_}i} = \phi_{\text{global\_}i}, \quad i = 1, 2.$
Global Distribution: For each region j, update local parameters: $\theta_j \leftarrow \theta_{\text{global}}$, $\phi_{j\_i} \leftarrow \phi_{\text{global\_}i} \quad i = 1, 2,$
$\theta'_j \leftarrow \theta'_{\text{global}}$, $\phi'_{j\_i} \leftarrow \phi'_{\text{global\_}i}$.

deployments. In O-RAN, control functionality is split across multiple layers, such as xApps operating at the Near-RT RIC and rApps at the Non-RT RIC, communicating via standardized open interfaces. This structure aligns seamlessly with the federated learning paradigm, where learning agents (e.g., xApps) act as local clients that train policies independently using site-specific data and periodically synchronize with a central coordinator (e.g., an rApp) to build a global policy model [32], [33].

Adopting federated DRL in this context offers significant scalability advantages over centralized DRL. As network size and RU density increase, centralized learning suffers from exploding state and action spaces, as well as communication bottlenecks and slow convergence. By distributing the learning process across O-RAN components, federated DRL reduces computational and communication loads at any single point [34], [35], while also enabling site-level policy customization. Moreover, the reuse of O-RAN's open interfaces allows efficient and standards-compliant integration of FL workflows into real network deployments. An overview of the proposed federated DRL framework within the O-RAN architecture is depicted in Fig. 1. In this design, each O-RAN region is equipped with a Near-RT RIC hosting local DRL agents that interact with the underlying RUs and UEs to make real-time sleep/active decisions. These agents are trained locally using region-specific traffic and mobility dynamics, thereby capturing heterogeneous environmental characteristics without requiring raw data exchange. Periodically, the locally trained models are uploaded to the Non-RT RIC, which serves as the global aggregator. The Non-RT RIC integrates model updates from multiple regions to construct a global policy, which is then redistributed back to the Near-RT RICs. This hierarchical workflow aligns naturally with the functional split of O-RAN: the Near-RT RIC ensures responsiveness to fast-varying network states, while the Non-RT RIC provides global coordination, scalability, and policy generalization across distributed deployments.

A. Federated DRL Training and Global Aggregation

In our federated DRL framework, each agent corresponds to a distinct geographical region, consisting of multiple RUs and UEs. The training process begins with the aggregator distributing an initialized global model to all participating agents. This model includes either Q-network weights for DQN agents or actor–critic parameters for TD3 agents, depending on the algorithm. Fig. 2 illustrates the entire Fed-TD3 training process as a representation. After receiving the global model, each agent independently interacts with its local environment, collecting state–action–reward transitions and updating its DRL model based on region-specific user mobility and traffic dynamics. This local training proceeds over a fixed number of episodes F. Once the local training phase is completed, each agent uploads selected model parameters to the aggregator. The aggregator then performs model aggregation to produce an updated global model. This aggregation strategy varies depending on the type of DRL model used and will be detailed in the following section. The updated global model is subsequently redistributed to all agents for the next round of training. This iterative cycle of local update and global synchronization continues until

convergence. By leveraging this decentralized optimization strategy, our framework ensures that agents collaboratively learn a generalizable policy while preserving privacy and significantly reducing communication overhead.

Depending on the underlying DRL algorithm, we employ two distinct parameter-sharing strategies to support both value-based and policy-based learning. The complete Fed-TD3 training process is summarized in Algorithm 2. For agents using the DQN architecture with discrete action spaces, each agent performs Q-learning on local environment and periodically uploads its Q-network to the server. The global model is computed by averaging the Q-values across all agents:

$$Q_{\text{global}}(s,a) = \frac{1}{R}\sum_{j=1}^{R} Q_j(s,a), \quad \forall s,a, \tag{25}$$

$$Q_j(s,a) \leftarrow Q_{\text{global}}(s,a), \quad \forall s,a, \tag{26}$$

where $Q_j(s,a)$ represents the Q-network of agent j, and R is the number of agents. The aggregated global model $Q_{\text{global}}(s,a)$ is then broadcast to all agents for subsequent training rounds.

For agents using the TD3 algorithm in the continuous action space, each agent maintains an actor–critic architecture. The global policy seeks to learn $|\mathcal{S}|\times|\mathcal{A}|$ table, $\pi_{global}(a|s)$. After several local update episodes, the policy network which is the actor network parameters is uploaded to the server for aggregation:

$$\pi_{\text{global}}(a|s) = \frac{1}{R}\sum_{j=1}^{R} \pi_j(a|s), \quad \forall s,a, \tag{27}$$

$$\pi_j(a|s) \leftarrow \pi_{\text{global}}(a|s), \quad \forall s,a. \tag{28}$$

Although the standard actor–critic framework typically aggregates only the actor to ensure decentralized critic adaptation, in our design we also upload and aggregate the two critic networks across agents. This additional aggregation improves training stability and enhances the generalizability of the learned value functions. After each global aggregation step, both the actor and critic target networks are updated, as detailed in Algorithm 2. In our scenario, the areas share a homogeneous RU activation task and similar traffic patterns, which reduces the degree of heterogeneity. In the future works we will investigate heterogeneous environments that cause further challenges. In this work we mainly want to show that Federated learning is a scalable AI solution for O-RAN networks.

V. Simulation Results

To evaluate the effectiveness of the proposed energy savings framework, we perform simulations across a range of network scenarios and baseline models. Our evaluation includes centralized and federated learning environments to assess performance in terms of reward, energy consumption, and scalability.

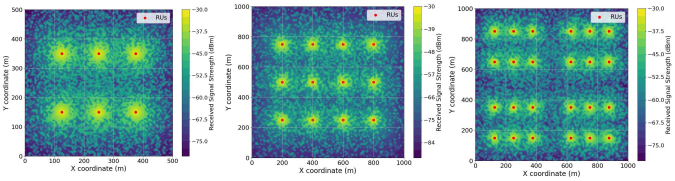

Fig. 3: The left plot shows a radio map of single 500 m×500 m area used for centralized training and evaluation. The right plot depicts a 1000 m×1000 m area composed of four such subregions, representing a composite environment for centralized training and inference. This comparison setting is used to evaluate the generalization capability of the global model trained via federated reinforcement learning versus a centralized model trained on the combined area. The middle plot presents a single large-scale 1000 m×1000 m environment used to evaluate model performance under centralized DRL training.

A. Simulation Settings

In our simulations, each scenario is modeled as an O-RAN environment where M RUs serve K UEs within a square area of size $L \times L\,\text{m}^2$. In practical O-RAN deployments, the Near-RT RIC is commonly deployed in edge cloud or regional edge data centers with latency budgets between 10 ms and 1 s. In our system model, the inference of the DRL agent runs inside the Near-RT RIC. The model training or federated aggregation may take place at the same edge location. This placement ensures that the decision-making latency requirements of the xApp are satisfied. As illustrated in Fig. 3, three layouts are considered: (i) a 500 m×500 m area with 6 RUs, used both as the centralized DRL testbed and as the local region in FL; (ii) a 1000 m×1000 m area with 12 RUs, representing a large-scale single region; (iii) a 1000 m×1000 m composite area with 24 RUs, formed by four independent 500 m×500 m subregions, used to examine the scalability of FL.

The UEs move at a constant speed $v = v_{\text{avg}} \pm v_{\text{std}}$ (mean $v_{\text{avg}} = 2$ m/s, standard deviation $v_{\text{std}} = 0.5$ m/s), with individual speeds randomly drawn from this range. A periodic mobility pattern is applied, where UEs travel from the service area edge toward the center and back, forming a structured cyclic movement. The wireless channel adopts the 3GPP UMi model [36] with both Line of Sight (LOS) and Non-Line of Sight (NLOS) conditions. For LOS links:

$$PL_{m,k}^{\text{LOS}} = \begin{cases} 32.4 + 21\log_{10}(d_{m,k}) \\ \quad + 20\log_{10}(f), & d_{m,k} \le d_{\text{BP}}, \\ 32.4 + 40\log_{10}(d_{m,k}) \\ \quad + 20\log_{10}(f) - 9.5, & d_{m,k} > d_{\text{BP}}, \end{cases} \tag{29}$$

with $d_{\text{BP}} = \frac{4h_{\text{RU}}h_{\text{UE}}f}{c}$ as the breakpoint distance. For NLOS links:

$$\begin{aligned} PL_{m,k}^{\text{NLOS}} &= 35.3 + 22.4\log_{10}(d_{m,k}) \\ &\quad + 21.3\log_{10}(f) - 0.3(h_{\text{UE}} - 1.5). \end{aligned} \tag{30}$$

The LOS probability is:

$$P_{\text{LOS}} = \min\left(\frac{18}{d_{m,k}}, 1\right)\left(1 - e^{-d_{m,k}/36}\right) + e^{-d_{m,k}/36}, \tag{31}$$

and $P_{\text{NLOS}} = 1 - P_{\text{LOS}}$, with $P_{\text{LOS}} = 1$ if $d_{m,k} < 18\,\text{m}$. Other system parameters, including carrier frequency, antenna heights, bandwidth, and RU power consumption values, follow Table II.

TABLE II: Simulation Parameters

Parameter	Value
TD3 Actor learning rate (α_π)	0.0001
TD3 Critic learning rate (α_Q)	0.001
DQN learning rate (α)	0.0001
Mini-batch size (B)	128
Replay memory size ($\mathcal{R}$)	50000
Discount factor (γ)	0.99
Training episodes (N_{eps})	2000
Random seed	42
Soft update coefficient (τ)	0.01
Carrier frequency (f)	2 GHz
RU height (h_{RU})	15 m
UE height (h_{UE})	1.7 m
Network size (L)	[500, 1000] m
Number of RUs (M)	[6, 12, 24]
Number of UEs (K)	[20–80]
Number of local regions (R)	8
Minimum data rate requirement (R_{min})	3 Mbps
Noise power (σ_n^2)	-174 dBm/Hz
Power amplifier efficiency (η)	0.5
Average UE speed (v_{avg})	2 m/s
Std. deviation of UE speed (v_{std})	0.5 m/s
Active mode RU power (P^{active})	20 W
Sleep mode RU power (P^{sleep})	5 W
Maximum transmission power (P_{TX})	1 W / 30 dBm
Mode transition power (P^{trans})	3 W
Reward weights (w_1, w_2)	1, 5

During training, each episode consists of 200 time steps. Considering the transition latency of RUs, each time step is set to 1 s, which is also the interval for both energy consumption measurement and sleep/active mode decisions. The energy consumption recorded at each step corresponds to the instantaneous value, while the total energy consumption is obtained by summing over all steps in an episode. For the TD3 model, both the actor and critic networks consist of four fully connected layers, with activation functions and layer sizes detailed in Table III. The Adam optimizer is used for parameter updates, and Batch Normalization (BN) is applied to improve training stability. For the DQN-based models, each network contains five fully connected layers, with architectures also provided in Table III.

We evaluate three centralized DRL approaches: DQNSA, DQNMA, and TD3, which together provide a representative coverage of both value-based and policy-based methods. To investigate scalability in large-scale or distributed deployments, each of these approaches is further extended to a federated learning setting, resulting in the Fed-DQNSA, Fed-DQNMA, and Fed-TD3 models. In the federated setting, agents are trained on separate local regions and periodically aggregated into a global model, whereas the centralized setting relies on a single combined environment. This design enables a systematic comparison between centralized and federated paradigms across different DRL architectures.

TABLE III: Network Configurations

	Layer1	Layer2	Layer3	Layer4	Output Layer
Actor	BN+relu 512	relu 256	relu 128	None	sigmoid M
Critic	relu 512	relu 256	relu 128	None	linear 1
DQNSA	relu 512	relu 384	relu 256	relu 128	linear M^2
DQNMA	relu 512	relu 384	relu 256	relu 128	linear $2M$

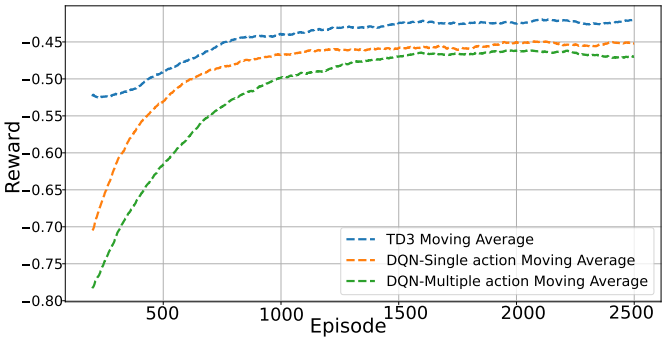

Fig. 4: Illustrates the rewards of TD3, DQNMA and DQNSA model in 500 m×500 m area with 6 RUs and 20 UEs.

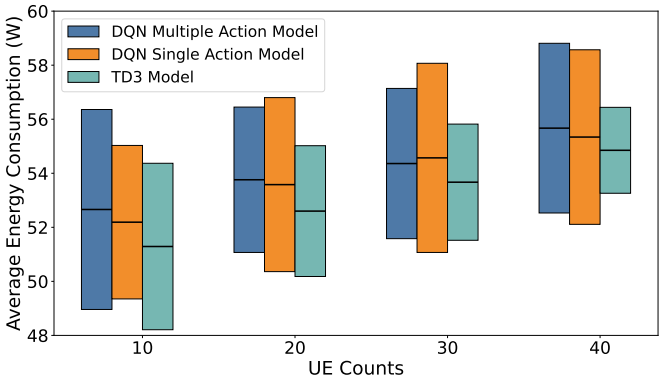

Fig. 5: Illustrates the average energy consumption in 500 m×500 m area among 6 RUs and 10 to 40 UEs with DQNMA, DQNSA and TD3 models. The maximum theoretical energy consumption is 126W.

B. Numerical Results

In Fig. 4, we present the reward performance of the centralized TD3, DQNSA, and DQNMA models over 2500 episodes in a 500,m × 500,m network layout. The results clearly highlight the advantages of continuous-action methods: the TD3 model consistently outperforms both DQN variants, achieving the highest long-term reward with smooth convergence and superior stability. In contrast, the DQNSA model demonstrates moderate performance, yet its reward trajectory exhibits noticeable fluctuations, reflecting the limitations of discrete-action value approximation in adapting to dynamic environments. The DQNMA model, while extending the action granularity with multiple discrete outputs, does not translate this added complexity into performance gains; instead, it yields the lowest reward among the three. This comparison indicates that simply enlarging the discrete action space within a DQN framework cannot effectively capture the nuanced trade-offs required for energy-efficient RU scheduling, whereas TD3's

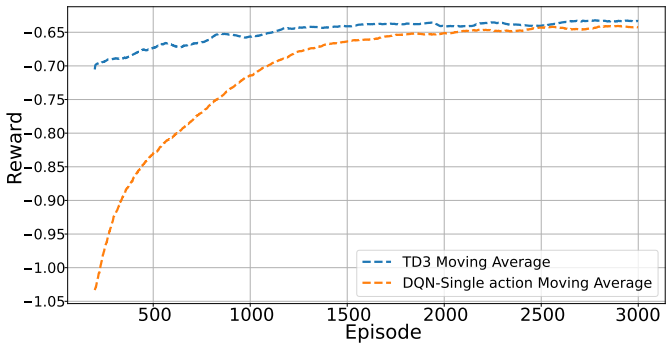

Fig. 6: Illustrates the rewards of TD3 and DQNSA model in 1000 m×1000 m area with 12 RUs and 40 UEs.

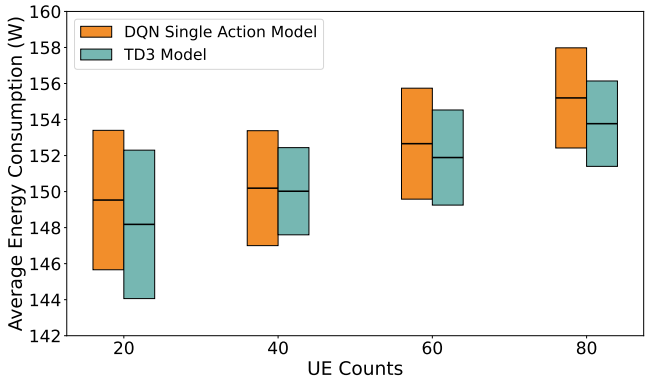

Fig. 7: Illustrates the average energy consumption among 12 RUs and 20 to 80 UEs in 1000 m×1000 m area with TD3 and DQNSA models. The maximum theoretical energy consumption is 252W.

continuous-action formulation enables more fine-grained and adaptive decision-making.

Fig.5 indicates the energy consumption performance in 500 m×500 m area for varying UE counts ranging from 10 to 40, is still compared among the previous three models. The figure distinctly shows the average energy consumption, indicated by the black line in the center of each bar, while the length of each bar represents the variance in energy consumption. With a total of 6 RUs installed in this area, the maximum possible energy consumption reaches 126W when all RUs operate in active mode. The results clearly demonstrate that all three models achieve significant energy savings, exceeding 50% compared to the theoretical maximum. Specifically, the TD3 model achieving up to an additional 6% energy saving compared to the two DQN-based models.

In Fig.6 shows the reward performance over an extended simulation environment from 500 m ×500 m area to 1000 m×1000 m area. But only compare the TD3 model against the DQNSA model due to the excessive expansion of DQNMA action space (e.g. 2^{12}). In the results, the TD3 model still achieves higher and stable reward levels throughout all episodes. Furthermore, the TD3 model demonstrates faster and more efficient convergence toward optimal solutions. The Fig.7 illustrates the energy consumption between the TD3 and DQNSA model, covering UE counts from 20 to 80.

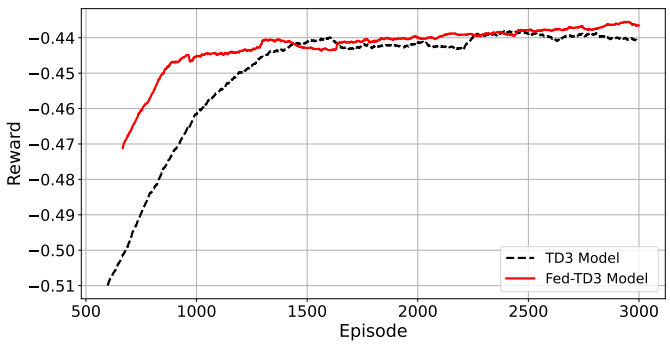

Fig. 8: Illustrates the rewards and convergence speed of TD3 and Fed-TD3 in 500 m×500 m area with 6 RUs and 20 UEs.

Numerically, the TD3 model maintains lower energy consumption, approximately 144w at 20 UEs and up to around 152w at 80 UEs, significantly below the theoretical maximum consumption of 252W (more than 40%) with all 12 RUs active. However, the TD3 model can also achieve 5% over the DQNSA mode, further emphasizing the TD3 model in a complex and large-scale environment.

Having established that TD3 provides the best performance among the centralized DRL approaches, we next extend our analysis to examine the benefits of federated learning. As illustrated in Fig. 8, the training reward curves highlight clear advantages of the federated framework. To statistically validate the convergence behavior, we computed the convergence episode for five independent runs of both TD3 and Fed-TD3 and performed a Welch two-sample t-test. Fed-TD3 converges in 858.8 ± 127.4 episodes on average, while TD3 requires 1421.4 ± 322.6 episodes. Fed-TD3 achieves 40% reduction in convergence episodes compared with TD3. The t-test yields $p = 0.0309 < 0.05$, indicating that the faster convergence of Fed-TD3 is statistically significant at the 95% confidence level. The observed behavior is interpreted as an empirical outcome of the proposed multi-area hierarchical learning setup. The improved learning dynamics can be attributed to the aggregation of diverse policy updates across regions, which enhances generalization and stabilizes the learning process. Moreover, Fed-TD3 attains a higher final reward compared to centralized TD3, indicating superior long-term performance in balancing energy savings and service quality.

Fig. 9 compares the training reward curves of three federated DRL models: Fed-TD3, Fed-DQNMA, and Fed-DQNSA. Both federated DQN variants exhibit significant improvements over their non-federated counterparts, with approximately a 10% increase in final average reward. This demonstrates that federated learning effectively enhances the generalization and learning efficiency of value-based methods even under discrete action settings. Notably, the final rewards achieved by Fed-DQNMA and Fed-DQNSA closely approach that of Fed-TD3, suggesting that with appropriate aggregation, discrete-action methods can benefit substantially from distributed training. However, the convergence speed of Fed-DQNMA and Fed-DQNSA remains noticeably slower than Fed-TD3. While Fed-TD3 stabilizes after approximately 900 episodes, the federated DQN models require around 1,300 episodes to reach similar

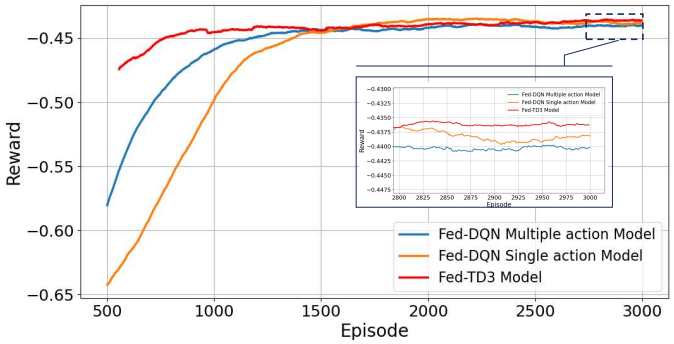

Fig. 9: Illustrates the rewards and convergence speed of Fed-TD3, Fed-DQNSA and Fed-DQNMA model in 500 m×500 m area with 6RUs and 20UEs.

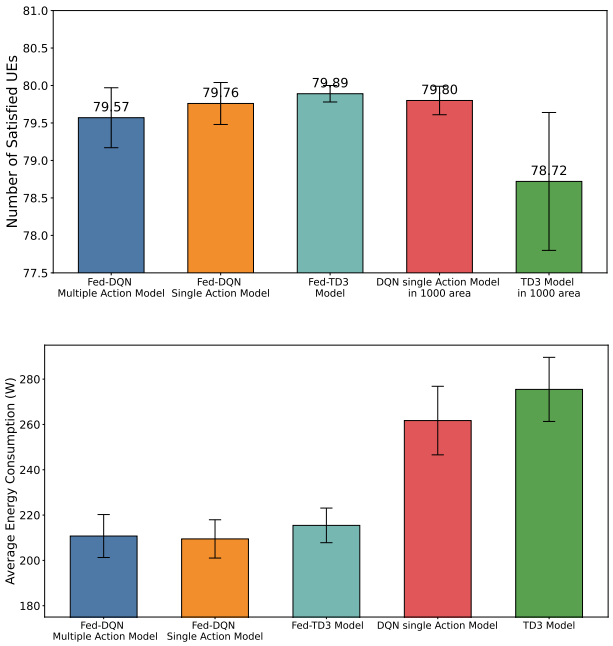

Fig. 10: Illustrates comparison of average energy consumption (right figure) and UE satisfaction (left figure) between federated and centralized reinforcement learning approaches. The first three bars represent federated models (Fed-DQN with multiple and single actions, and Fed-TD3) applied to four independent 500 m×500 m regions with 6 RUs and 20 UEs. The last two bars show the performance of centralized models (DQN and TD3) trained on a merged 1000 m×1000 m region with 24 RUs and 80 UEs.

stability, indicating a relative slowdown of about 30%. This gap reflects the inherent limitations of Q-learning in handling large action spaces compared to policy-based approaches.

Fig. 10 compares the average energy consumption between federated and centralized reinforcement learning approaches. The federated models (e.g. Fed-DQNMA, Fed-DQNSA, and Fed-TD3) are first trained across four geographically separated 500 m×500 m regions. After convergence, the resulting global models are independently evaluated in each of the original

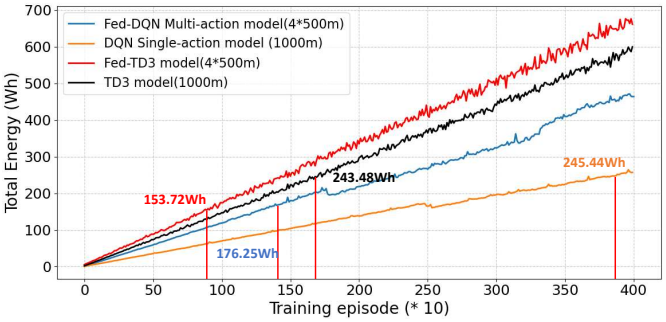

Fig. 11: Illustrates training energy consumption for different DRL models. Each curve represents the cumulative energy usage over time for a specific model. The red labels indicate the total energy consumed by each model upon convergence. The comparison highlights the efficiency differences among federated and centralized approaches.

training regions, and the total test energy consumption is obtained by summing across all four areas. In contrast, the centralized models (DQN and TD3) are trained on a merged 1000 m×1000 m, environment composed of the same four subregions, with user distributions and mobility constrained within each 500 m×500 m area, as illustrated on the third figure of Fig. 3. This setting ensures that both federated and centralized models are tested on identical underlying environments, allowing a fair comparison of generalization and efficiency. The results show that federated models achieve comparable or lower overall energy consumption than their centralized counterparts, despite being trained without direct access to globally aggregated data. In particular, Fed-DQNMA and Fed-DQNSA demonstrate strong energy-saving behavior, while Fed-TD3 incurs slightly higher energy usage. This difference arises because Fed-TD3 prioritizes satisfying the QoS requirements of all users, leading to fewer RUs being turned off during testing. In contrast, the fed-DQN models are more aggressive in RU deactivation, sacrificing service quality for a greater reduction in energy consumption. Nevertheless, Fed-TD3 achieves higher average rewards, indicating a better balance between energy efficiency and user satisfaction. These results confirm that federated reinforcement learning not only enables scalable training across geographically distributed areas, but also yields robust and energy-efficient policies that generalize well when deployed in composite environments. Note that in Fig.10, the TD3 model performs slightly worse than the DQNSA in the 1000 m×1000 m area. This is primarily due to the significantly larger action space in the centralized TD3 setting, which involves 24 RUs and poses challenges even with continuous action control. It is important to note that this comparison focuses solely on the differences between FL and non-FL approaches.

Fig. 11 reports the total energy consumption required for each model to reach convergence during training. The federated models Fed-TD3 and Fed-DQNMA are compared against centralized DQN and TD3 baselines trained on the full 1000 m×1000 m area. Each data point represents the accumulated training energy measured at the episode when

the model achieves stable reward performance. To obtain accurate measurements, we implemented a custom power monitoring module that continuously samples both CPU and GPU power usage during training. The GPU power was measured using the NVIDIA System Management Interface (nvidia-smi), while CPU utilization was monitored using the `psutil` library. Samples were collected every 100 ms, and total energy consumption (in Wh) was computed by integrating the average power over the training duration. All models were trained exclusively on a single RTX 3060 GPU, with no other significant processes running concurrently. Among all methods, Fed-TD3 exhibits the lowest training energy consumption at 153.72 Wh, followed by Fed-DQNMA at 176.25 Wh. In contrast, the centralized DRL models consume 243.48 Wh (TD3) and 245.44 Wh (DQN), respectively—the highest among all configurations. Notably, Fed-TD3 reduces training energy by approximately 37.4% compared to the centralized DQN model, while Fed-DQNMA achieves a 28.2% reduction. These results highlight the efficiency advantage of federated reinforcement learning in distributed training scenarios.

VI. Conclusions

In this work, we explored energy-efficient RU sleep control within the O-RAN framework through both centralized and federated DRL strategies. In the centralized setting, we evaluate DQNMA, DQNSA, and TD3, demonstrating that continuous-action control in TD3 effectively avoids the combinatorial growth of discrete methods while enabling coordinated multi-RU decisions. Building on these insights, we extend the solution to a FL framework aligned with O-RAN's hierarchical control, enabling scalable training across geographically distributed regions without sharing raw data. Extensive simulations across varied network scales and heterogeneous layouts show that the federated global models achieve competitive or superior performance compared to centralized training while significantly improving scalability and generalization. Compared with DQN-based baselines, Fed-TD3 achieves higher rewards, lower training energy consumption, and better adaptability to complex action spaces. These results confirm the practicality of combining centralized DRL optimization with FL to enable flexible and energy efficiency network deployments.

References

[1] J. S. Thompson, S. Fletcher, V. Friderikos, Y. Gao, L. Hanzo, M. R. Nakhai, T. O'Farrell, and P. D. Wells, "Editorial a decade of green radio and the path to "net zero": A united kingdom perspective," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 2, pp. 657–664, 2022.

[2] H. Ahmadi, M. Rahmani, S. B. Chetty, E. E. Tsiropoulou, H. Arslan, M. Debbah, and T. Quek, "Towards sustainability in 6g and beyond: Challenges and opportunities of open ran," *IEEE Communications Standards Magazine*, 2025.

[3] M. Polese, L. Bonati, S. D'oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[4] O-RAN Allience WG1, "O-RAN Architecture Description," June 2025, accessed: June 2025. [Online]. Available: https://https://specifications.o-ran.org/specifications

[5] A. I. Abubakar, O. Onireti, Y. Sambo, L. Zhang, G. Ragesh, and M. A. Imran, "Energy efficiency of open radio access network: A survey," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. IEEE, 2023, pp. 1–7.

[6] P. Lähdekorpi, M. Hronec, P. Jolma, and J. Moilanen, "Energy efficiency of 5G mobile networks with base station sleep modes," in *2017 IEEE conference on standards for communications and networking (CSCN)*. IEEE, 2017, pp. 163–168.

[7] M. Dryjański, Ł. Kułacz, and A. Kliks, "Toward modular and flexible open ran implementations in 6g networks: Traffic steering use case and o-ran xapps," *Sensors*, vol. 21, no. 24, p. 8173, 2021.

[8] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and learning in o-ran for data-driven nextg cellular networks," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 21–27, 2021.

[9] L. Zhou, M. V. Ngo, B. Chen, and T. Q. Quek, "Digital twins meet open ran: Case studies, implementation, and opportunities," *IEEE Communications Magazine*, vol. 63, no. 8, pp. 162–168, 2025.

[10] B. Agarwal, R. Irmer, D. Lister, and G.-M. Muntean, "Open ran for 6g networks: Architecture, use cases and open issues," *IEEE Communications Surveys & Tutorials*, 2025.

[11] O-RAN Allience WG2, "A1 Interface: general aspects and principles 3.0," June 2025, accessed: June 2025. [Online]. Available: https://https://specifications.o-ran.org/specifications

[12] O-RAN Allience WG3, "O-RAN E2 General Aspects and Principles (E2GAP) 8.0," October 2025, accessed: October 2025. [Online]. Available: https://https://specifications.o-ran.org/specifications

[13] L. Zhou, W. Feng, Z. Li, S. Leng, M. Guizani, and T. Q. Quek, "Integrating foundation models with open ran for robots-based mobile scenarios," *IEEE Communications Magazine*, vol. 63, no. 9, pp. 28–34, 2025.

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

[15] C. Liu, B. Natarajan, and H. Xia, "Small cell base station sleep strategies for energy efficiency," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1652–1661, 2015.

[16] E. Oh and K. Son, "A unified base station switching framework considering both uplink and downlink traffic," *IEEE Wireless Communications Letters*, vol. 6, no. 1, pp. 30–33, 2016.

[17] J. Peng, P. Hong, and K. Xue, "Stochastic analysis of optimal base station energy saving in cellular networks with sleep mode," *IEEE Communications Letters*, vol. 18, no. 4, pp. 612–615, 2014.

[18] G. Jang, N. Kim, T. Ha, C. Lee, and S. Cho, "Base station switching and sleep mode optimization with lstm-based user prediction," *IEEE Access*, vol. 8, pp. 222 711–222 723, 2020.

[19] H. Ju, S. Kim, Y. Kim, and B. Shim, "Energy-efficient ultra-dense network with deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6539–6552, 2022.

[20] J. Ye and Y.-J. A. Zhang, "Drag: Deep reinforcement learning based base station activation in heterogeneous networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2076–2087, 2019.

[21] Q. Wu, X. Chen, Z. Zhou, L. Chen, and J. Zhang, "Deep reinforcement learning with spatio-temporal traffic forecasting for data-driven base station sleep control," *IEEE/ACM transactions on networking*, vol. 29, no. 2, pp. 935–948, 2021.

[22] J. Gan, H. Kou, G. Yang, H. Zhang, Z. Cao, and W. Xu, "Joint sleep control and energy sharing strategy with deep reinforcement learning in green ultra-dense networks," *IEEE Transactions on Green Communications and Networking*, 2025.

[23] S. Sun, C. Huang, G. Chen, P. Xiao, and R. Tafazolli, "Deep learning-based traffic-aware base station sleep mode and cell zooming strategy in ris-aided multi-cell networks," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[24] M. Masoudi, E. Soroush, J. Zander, and C. Cavdar, "Digital twin assisted risk-aware sleep mode management using deep q-networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1224–1239, 2022.

[25] Y. Zhen, L. Tao, D. Wu, T. Tang, and R. Wang, "Energy-saving control strategy for ultra-dense network base stations based on multi-agent reinforcement learning," *Digital Communications and Networks*, 2024.

[26] X. Liang, Q. Wang, A. Al-Tahmeesschi, S. B. Chetty, D. Grace, and H. Ahmadi, "Energy consumption of machine learning enhanced open ran: A comprehensive review," *IEEE Access*, vol. 12, pp. 81 889–81 910, 2024.

[27] Q. Wang, S. Chetty, A. Al-Tahmeesschi, X. Liang, Y. Chu, and H. Ahmadi, "Energy saving in 6g o-ran using dqn-based xapp," *arXiv preprint arXiv:2409.15098*, 2024.

[28] Y. Zhou, Y. Shi, H. Zhou, J. Wang, L. Fu, and Y. Yang, "Toward scalable wireless federated learning: Challenges and solutions," *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 10–16, 2023.

[29] B. Zhao, Q. Cui, W. Ni, X. Li, and S. Liang, "Multi-layer collaborative federated learning architecture for 6g open ran," *Wireless Networks*, vol. 31, no. 2, pp. 1377–1390, 2025.

[30] H. Jin, Y. Peng, W. Yang, S. Wang, and Z. Zhang, "Federated reinforcement learning with environment heterogeneity," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 18–37.

[31] T. Pan, X. Wu, and X. Li, "Dynamic multi-sleeping control with diverse quality-of-service requirements in sixth-generation networks using federated learning," *Electronics*, vol. 13, no. 3, p. 549, 2024.

[32] A. Ndikumana, K. K. Nguyen, and M. Cheriet, "Federated learning assisted deep q-learning for joint task offloading and fronthaul segment routing in open ran," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3261–3273, 2023.

[33] J. Wang, P. Chen, J. Wang, and B. Yang, "A hierarchical federated learning paradigm in o-ran for resource-constrained iot devices," in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 2555–2560.

[34] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[35] A. K. Singh and K. K. Nguyen, "Communication efficient compressed and accelerated federated learning in open ran intelligent controllers," *IEEE/ACM Transactions on Networking*, 2024.

[36] 3GPP, "Study on channel model for frequencies from 0.5 to 100 ghz," ETSI, Technical Report TR 138 901, 2017. [Online]. Available: https://www.etsi.org/deliver/etsi˙tr/138900˙138999/138901.pdf