Deep Reinforcement Learning for Dynamic Access Point Activation in Cell-Free MIMO Networks

Charmae Franchesca Mendoza†, Stefan Schwarz† and Markus Rupp
†Christian Doppler Laboratory for Dependable Wireless Connectivity for the Society in Motion
Institute of Telecommunications, Technische Universität (TU) Wien
Email: {charmae.mendoza,stefan.schwarz,markus.rupp}@tuwien.ac.at

***Abstract*—The cell-free network architecture suppresses inter-cell interference by eliminating cell boundaries through the joint operation of distributed access points (APs). While a dense deployment of these APs may lead to performance gains, the corresponding increase in energy consumption poses environmental and economic issues. One way to improve energy efficiency is to turn off underutilized APs. In this work, we present a deep reinforcement learning-based framework that derives the set of active APs given the spatial user information. The flexible design of the reward function for AP selection allows easy adjustment of performance targets in terms of quality of service and power consumption, as well as studying their trade-off. We also demonstrate how the proposed framework intelligently selects and activates only a subset of APs that contributes significantly to user performance, thereby enabling the cell-free network to provide good service while achieving power savings.**

***Index Terms*—deep reinforcement learning, dynamic AP activation, cell-free MIMO, quality of service, energy efficiency**

I. Introduction

A key enabler for 6G [1] is *cell-free massive multiple-input multiple-output* (MIMO) introduced in [2]. It consists of a large number of distributed cooperating access points (APs) simultaneously serving a much smaller number of users using the same time-frequency resources. Each AP is connected to a centralized processing unit (CPU) via the fronthaul link. The joint operation of APs eliminates cell boundaries to suppress inter-cell interference, which degrades performance in conventional cellular networks. The coherent joint transmission and reception of multiple distributed antennas/APs enable uniformly good service throughout the coverage area [2], [3].

The global mobile data traffic is projected to continuously grow with the increased usage of data-intensive applications and advancement of device capabilities [4]. A dense deployment of APs becomes crucial to support such traffic demand. This, however, also triggers a substantial increase in energy consumption, which is neither sustainable in the environmental sense nor in the perspective of networks operators who are burdened with high operational costs.

In order to reap the benefits of cell-free MIMO, its energy efficiency was investigated in [5], [6]. In our previous work [7], we focused on the AP selection and the optimal user-centric cluster size for a given scenario. However, these prior studies assumed that all APs are always activated. In reality, load traffic conditions vary over space and time. Thus, some APs may be underutilized at certain time periods (off-peak hours) and only contribute to the increased overall energy consumption of the network. Recently, the deactivation of a subset of APs in the context of cell-free MIMO has been studied to improve energy efficiency. In [8], an optimization problem was formulated to minimize the total power consumption over the transmit powers and the number of active APs, while in [9], a joint power allocation and AP selection approach was proposed to minimize the total energy consumption. Different AP switch on/off strategies were presented in [10], where the focus was on suboptimal heuristic algorithms to reduce the complexity when solving the NP-hard AP selection problem. However, all these previous works still relied either on heuristics or conventional optimization methods.

One technique that we can leverage to design network optimization algorithms is *(deep) reinforcement learning* (DRL) [11], [12]. It is a subfield of machine learning, where an *agent* autonomously learns to make a sequence of decisions through continuous interaction with its surroundings without the need for training data. This then serves as a powerful tool to enable self-optimization in dynamic networks. It is for this reason that we see a recent trend of adopting DRL to tackle different challenges in telecommunications, such as solving resource allocation problems [13], [14].

In this paper, we present an intelligent approach for realizing energy-efficient cell-free MIMO networks. The main aspects of our proposed framework are summarized as follows:

1) We utilize DRL to select the best APs to activate given the spatial information of users. In particular, the framework jointly considers the position of all users when deriving the set of active APs. This prevents having an AP turned on even if only one user is connected to it, which is rather inefficient.
2) Central to the decision-making of DRL is the *reward* function. We designed the reward to allow for a straightforward configuration of performance targets in terms of quality of service (QoS) and power savings, followed by an automatic reconfiguration of the network. We consider the guaranteed QoS to all users (instead of sum rate) to ensure uniform performance regardless of user location.
3) The DRL-based scheme has its advantages over using conventional optimization techniques for AP selection. For one, DRL does not strictly require the full channel state information (CSI) to be gathered at the CPU to

achieve our objective. The framework can alternatively let the users report their experience (for instance, actual SINR). Based on that and the resulting reward, it can dynamically activate the best APs. Another advantage is that the DRL agent stores its past experiences/interaction with the environment. Although this is not the main focus of our paper, it is worth highlighting that the framework can, therefore, retain (statistical) information about the spatiotemporal user demand and use that to adapt the network in advance. In contrast, real-time implementation of conventional optimization techniques can be computationally intensive and time consuming, making it difficult to deploy them in practice.

II. Cell-free MIMO System Model

We consider a downlink cell-free MIMO network with M APs having N_t antennas each and K single-antenna UEs. We assume an independent and identically distributed (iid) Rayleigh fading channel between AP m and UE k expressed as:

$$\mathbf{h}_{k,m} = \sqrt{g_{k,m}}\,\widetilde{\mathbf{h}}_{k,m} \in \mathbb{C}^{N_t \times 1}, \tag{1}$$

with $\widetilde{\mathbf{h}}_{k,m} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_t})$ denoting the small-scale fading. The macroscopic channel gain $g_{k,m}$ follows a distance-dependent path loss model with shadow fading:

$$g_{k,m} = \left(\frac{\lambda_c}{4\pi}\right)^2 \left(\frac{1}{d_{k,m}}\right)^{n_c} s_{k,m}, \tag{2}$$

where λ_c is the wavelength of the carrier frequency, $d_{k,m}$ is the distance between UE k and AP m, n_c is the path loss exponent and $s_{k,m} \sim \mathcal{LN}(0, \sigma_c^2)$ is the random lognormally distributed shadow fading.

Each AP performs Maximum Ratio Transmission (MRT) for the downlink data transmissions. The precoder matrix of AP m is represented by $\mathbf{F}_m \in \mathbb{C}^{N_t \times K}$, with column $\mathbf{f}_{k,m} = \mathbf{h}_{k,m}(\|\mathbf{h}_{k,m}\|)^{-1} \in \mathbb{C}^{N_t \times 1}$ corresponding to the beamforming vector of UE k. The input-output relationship of UE k is then:

$$y_k = \sum_{m=1}^{M} b_m \sqrt{P_{k,m}}\,\mathbf{h}_{k,m}^H \mathbf{F}_m \mathbf{s}_m + z_k, \tag{3}$$

where b_m is a binary variable indicating whether an AP is activated or not (i.e., $b_m = 1$ if AP m is turned on and $b_m = 0$ otherwise), $P_{k,m}$ denotes the power allocated to UE k by AP m, $\mathbf{s}_m \in \mathbb{C}^{K \times 1}$ includes the data symbols for the users such that $\mathbb{E}(\mathbf{s}_m \mathbf{s}_m^H) = \mathbf{I}_K$ and z_k is the receiver noise with variance σ_z^2.

The instantaneous signal to interference and noise ratio (SINR) of UE k is given by:

$$\text{Inst. SINR}_k = \frac{\left|\sum_{m=1}^{M} b_m \sqrt{P_{k,m}}\,\mathbf{h}_{k,m}^H \mathbf{f}_{k,m}\right|^2}{\sum_{\substack{j=1 \\ j \neq k}}^{K} \left|\sum_{m=1}^{M} b_m \sqrt{P_{j,m}}\,\mathbf{h}_{k,m}^H \mathbf{f}_{j,m}\right|^2 + \sigma_z^2}. \tag{4}$$

We coherently add the contributions of active APs only in both the signal power and inter-user interference calculation. In this work, we use the average SINR expressed as:

$$\text{SINR}_k = \frac{P_{sig,k}}{\sum_{\substack{j=1 \\ j \neq k}}^{K} P_{int,jk} + \sigma_z^2}, \tag{5}$$

where

$$\begin{aligned} P_{sig,k} &= \mathbb{E}\left\{\left|\sum_{m=1}^{M} b_m \sqrt{P_{k,m}}\,\mathbf{h}_{k,m}^H \mathbf{f}_{k,m}\right|^2\right\} \\ &= \sum_{m=1}^{M} b_m P_{k,m} g_{k,m} \mathbb{E}\{\underbrace{\widetilde{\mathbf{h}}_{k,m}^H \mathbf{f}_{k,m} \mathbf{f}_{k,m}^H \widetilde{\mathbf{h}}_{k,m}}_{= \widetilde{\mathbf{h}}_{k,m}^H \widetilde{\mathbf{h}}_{k,m} \widetilde{\mathbf{h}}_{k,m}^H \widetilde{\mathbf{h}}_{k,m}}\} \\ &+ \sum_{m=1}^{M} \sum_{n \neq m}^{M} b_m b_n \sqrt{P_{k,m} g_{k,m} P_{k,n} g_{k,n}} \mathbb{E}\{\underbrace{\widetilde{\mathbf{h}}_{k,m}^H \mathbf{f}_{k,m} \widetilde{\mathbf{h}}_{k,n}^H \mathbf{f}_{k,n}}_{= \widetilde{\mathbf{h}}_{k,m}^H \widetilde{\mathbf{h}}_{k,m} \widetilde{\mathbf{h}}_{k,n}^H \widetilde{\mathbf{h}}_{k,n}}\} \\ &= \sum_{m=1}^{M} b_m P_{k,m} g_{k,m} + \sum_{m=1}^{M} \sum_{n \neq m}^{M} b_m b_n \sqrt{P_{k,m} g_{k,m} P_{k,n} g_{k,n}}, \end{aligned} \tag{6}$$

and

$$\begin{aligned} P_{int,jk} &= \mathbb{E}\left\{\left|\sum_{m=1}^{M} b_m \sqrt{P_{j,m}}\,\mathbf{h}_{k,m}^H \mathbf{f}_{j,m}\right|^2\right\} \\ &= \sum_{m=1}^{M} b_m P_{j,m} g_{k,m} \mathbb{E}\{\widetilde{\mathbf{h}}_{k,m}^H \mathbf{f}_{j,m} \mathbf{f}_{j,m}^H \widetilde{\mathbf{h}}_{k,m}\} \\ &+ \sum_{m=1}^{M} \sum_{n \neq m}^{M} b_m b_n \sqrt{P_{j,m} g_{k,m} P_{j,n} g_{k,n}} \mathbb{E}\{\widetilde{\mathbf{h}}_{k,m}^H \mathbf{f}_{j,m} \widetilde{\mathbf{h}}_{k,n}^H \mathbf{f}_{j,n}\} \\ &= \sum_{m=1}^{M} b_m \frac{P_{j,m} g_{k,m}}{N_t}. \end{aligned} \tag{7}$$

Note that we simplify the first term in (7) using the following:

$$\mathbb{E}\{\underbrace{\text{tr}(\mathbf{f}_{j,m} \mathbf{f}_{j,m}^H)}_{= \mathbf{f}_{j,m}^H \mathbf{f}_{j,m} = 1}\} = \text{tr}\{\mathbb{E}(\mathbf{f}_{j,m} \mathbf{f}_{j,m}^H)\} = \text{tr}\left(\frac{1}{N_t} \mathbf{I}_{N_t}\right) = 1, \tag{8}$$

while the second term vanishes since the inner product of $\widetilde{\mathbf{h}}_{k,m}$ and $\mathbf{f}_{j,m}$ is 0 (independent channels). As we will see in the next section, the SINR directly affects the AP selection process. In the case of instantaneous SINR, considering the microscopic CSI implies that the AP activation becomes sensitive even to small user movements, which then leads to undesirable effects such as turning on/off APs too often. This is, however, not a problem for the average SINR that is based on the macroscopic gain only. The average SINR ensures that the AP activation is more stable for a longer period of time, and thus, it is well suited for both static and mobile scenarios.

III. A Deep Reinforcement Learning Approach to Energy Efficiency

A. Preliminaries

In reinforcement learning [11], the *environment* is described by its state s_t at time step t. The *agent* makes an observation and decides which action a_t to perform. This triggers a change in the state of the environment to s_{t+1}, as well as the agent receiving a corresponding *reward* r_{t+1}. The goal of the agent is to maximize (or minimize) the cumulative reward received over time or the *expected discounted return*:

$$\mathbb{E}[G_t] = \mathbb{E}\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots\right], \tag{9}$$

where $\gamma \in [0, 1]$ is the *discount factor* that dictates the present value of future rewards [11].

We define *policy* π as a function that takes in a state as input and then outputs an action. The *action-value function* (*Q-function*) for policy π is:

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi\left[G_t | s_t, a_t\right]. \tag{10}$$

This attaches a value for selecting action a_t at state s_t following policy π. The agent estimates the Q-values as it interacts with the environment, which introduces the trade-off between *exploration* and *exploitation*. It exploits its current knowledge of estimates by selecting the action with the highest Q-value, while it explores by settling for a lower-valued action to improve its estimates [11]. While classical algorithms utilize a lookup table to store the state-action pairs with their corresponding Q-values, deep reinforcement learning [12] uses a deep neural network (DNN) to estimate the Q-values.

B. Deep Reinforcement Learning-based Framework

In this section, we present our proposed DRL-based framework for enabling energy-efficient cell-free MIMO. We first go through the individual components of the framework, then we later discuss how they fit together.

1) State: The state space $\mathcal{S}$ includes all possible states describing the environment. In general, the state vector is represented by:

$$s_t = [b_1, \cdots, b_M, p_1, \cdots, p_K]. \tag{11}$$

The first M elements correspond to the on/off state of the M APs. The remaining elements are used to describe the position of users. For example, these can hold the path loss values of users with respect to APs. However, in this work, we only consider instantaneous network load conditions and static users in the coverage area. Thus, the spatial user information remains constant throughout the learning process, and only the first M elements change at each time step (after performing an action). Following these assumptions, we can simplify the state vector representation as:

$$s_t = [b_1, \cdots, b_M]. \tag{12}$$

Note that such simplification is not valid when considering user mobility. In that case, we need to keep track of the changing user locations by including them in the state as in (11). However, even with the simplified state representation in (12), the impact of (static) user positions on AP activation (decision-making) is still taken into account by incorporating it in the reward calculation as explained later.

2) Action: The action space $\mathcal{A}$ contains all possible actions, with the action vector given by:

$$a_t = [c_1, \cdots, c_M]. \tag{13}$$

Each action signals which APs to (de-)activate. That is, the agent configures c_m to 1 in order to turn on AP m at time step t, or 0 to switch it off. Alternatively, we can (de-)activate only a subset of APs or a single AP at a time to reduce the complexity at the expense of slower convergence.

3) Action Selection: The agent considers the Q-value estimates when deciding which action to perform. To strike a balance between exploration and exploitation, we use the *decaying* ϵ*-greedy* algorithm with $\epsilon \in [0, 1]$:

$$a_t = \begin{cases} \arg\max_a Q(s, a) & \text{with probability } 1 - \epsilon \\ \text{any action} & \text{with probability } \epsilon. \end{cases} \tag{14}$$

The agent chooses the greedy action that maximizes the Q-function with probability $1 - \epsilon$, while it selects a random action with uniform distribution from the set of all possible actions with probability ϵ. Instead of having a fixed ϵ value, we decrease it over time at a rate $\epsilon_{decay} \in [0, 1]$ as:

$$\epsilon_{t+1} = \max\{\epsilon_t(1 - \epsilon_{decay}), \epsilon_{min}\}. \tag{15}$$

We start with $\epsilon_0 = \epsilon_{max}$, where we set ϵ_{max} to a large value to initially promote exploration. We gradually decrease ϵ until it reaches ϵ_{min} to exploit the improved Q-value estimates.

4) Max-min SINR Optimization: After deriving the set of active APs, we still need to allocate power among the users by solving the following max-min SINR optimization problem:

$$\begin{aligned} \text{maximize} \quad & \beta \\ \text{w.r.t.} \quad & 0 \leq P_{k,m} \leq P_T \\ \text{subject to} \quad & \text{SINR}_k \geq \beta, \forall k \in \{1, \ldots, K\} \\ & \sum_{k=1}^{K} P_{k,m} \leq P_T, \forall m \in \{1, \ldots, M\}. \end{aligned} \tag{16}$$

The objective in (16) is to maximize QoS (minimum SINR). The per-user SINR constraint guarantees a certain performance level to all users as defined by β. The per-AP power constraint ensures that the total power allocated by an AP does not exceed the maximum transmit power P_T, which for simplicity, we assume to be the same for all APs. We obtain β using the bisection method [15], while we get $P_{k,m}$ $\forall k, m$ by solving a feasibility problem for a given β. The resulting values are then used in the reward calculation.

In principle, we can opt for other (suboptimal) power allocation schemes to reduce complexity. However, in this paper, we solve the above optimization problem for optimal power allocation.

5) Reward: The scalar reward is designed to represent the goal of the framework such that maximizing the reward leads to achieving that goal. Our objective is to maximize the guaranteed QoS to all users while minimizing power consumption. We, therefore, define the reward as:

$$r_t = \underbrace{\alpha\beta_{norm}}_{\text{QoS}} - \underbrace{(1-\alpha)P_{norm}}_{\text{Power}}, \tag{17}$$

where

$$\beta_{norm} = \frac{\beta}{\beta_{max}}, \tag{18}$$

and

$$P_{norm} = \frac{\sum_{m=1}^{M} b_m \left(P_{circuit} + \sum_{k=1}^{K} P_{k,m} \right)}{M(P_{circuit} + P_T)}. \tag{19}$$

The first term corresponds to QoS, where β denotes the max-min SINR in (16) and β_{max} is the highest possible QoS such as when all APs are turned on. The second term captures the total power consumption, where $P_{circuit}$ is the power consumed when an AP is activated. To study the trade-off between QoS and power savings, the two terms are normalized and a weighting factor $\alpha \in [0, 1]$ can be configured. This allows us to easily set performance targets by adjusting α accordingly.

We also note that the reward is user location-dependent, irrespective of state vector representation. Specifically, β is determined by solving the max-min SINR optimization problem in (16), which uses the average SINR that is based on the path loss between each user and AP.

6) Deep Neural Network Policy Representation: We utilize a DNN as a Q-function approximator for realizing the agent's policy. The DNN takes in a state-action pair as input and then outputs a scalar number corresponding to the Q-value estimate. We employ a feedforward, fully-connected DNN with 2 hidden layers having 16 neurons each and a scaling layer before the output. We use the rectified linear unit (ReLU) as the activation function [16].

7) Double Deep Q-network: The goal of reinforcement learning is to find the optimal policy that maximizes the cumulative reward. This is achieved by continuously evaluating and updating the agent's policy. Since the policy is represented by a DNN, the Q-function is now given by $Q(s_t, a_t|\theta_t)$, where θ_t denotes the DNN parameters (i.e., weights) to be updated.

In this work, we use the *double deep Q-network* (double DQN) algorithm [17] to update the policy, specifically the Q-value estimates. Here, we define 2 DNNs with the same architecture: DQN_{train} parameterized by θ_{train} and DQN_{target} parameterized by θ_{target}. The update rule is:

$$Q(s_t, a_t|\theta_{train}) \leftarrow Q(s_t, a_t|\theta_{train}) + \lambda(\Delta Q). \tag{20}$$

The *learning rate* is denoted by $\lambda \in (0, 1]$, and the error ΔQ is expressed as:

$$\Delta Q = \text{Target} - Q(s_t, a_t|\theta_{train}), \tag{21}$$

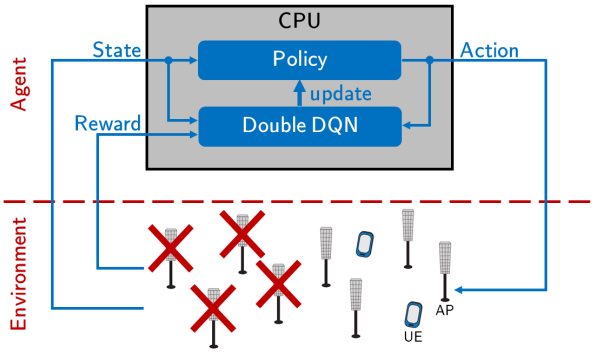

Fig. 1. Proposed DRL-based framework.

where

$$\text{Target} = r(s_t, a_t) + \gamma Q(s_{t+1}, a_{max}|\theta_{target}), \tag{22}$$

and

$$a_{max} = \arg\max_a Q(s_{t+1}, a_{t+1}|\theta_{train}). \tag{23}$$

In contrast, the original DQN algorithm employs a single DNN, which has some disadvantages. For one, we minimize the error at each time step by getting closer to Target. However, when using the same DNN for estimating both Target and Q-value, the target is moving/shifted as well, causing oscillations during training. In addition, the problem of overestimation of Q-values in DQN was first investigated in [17]. It was shown that using separate networks for generating Target in (22) and selecting an action in (23) solves this issue and improves performance.

We use *experience replay* [12] in minimizing the error. At each time step t, the agent stores an experience tuple $\{s_t, a_t, r_{t+1}, s_{t+1}\}$ in a *replay buffer* $\mathcal{B}$. It then samples a random mini-batch of size M from $\mathcal{B}$. Random sampling is necessary to avoid updating the network using correlated experiences. We define the loss function as the mean squared error (MSE) between the target and the current Q-value estimate over the M sampled experiences indexed by i below:

$$L(\theta_{train}) = \frac{1}{M} \sum_{i=1}^{M} \left\{ \text{Target}_i - Q(s_i, a_i|\theta_{train}) \right\}^2. \tag{24}$$

We minimize MSE and update θ_{train} by performing gradient descent. On the other hand, we update θ_{target} through *Polyak averaging* [16] as:

$$\theta_{target} \leftarrow \delta\theta_{train} + (1-\delta)\theta_{target}, \tag{25}$$

where $\delta \in [0, 1]$ is the *rate of averaging*.

Fig. 1 illustrates the agent-environment scenario of our DRL-based framework. We utilize a single agent that is the CPU in a cell-free MIMO network. The environment consists of the APs and users. The CPU knows where the users are located in the coverage area (state). Based on this spatial information, it decides which APs to activate (action) that will maximize the guaranteed QoS while achieving power savings

(reward). It considers the obtained state, action and reward information in its decision-making process. It does so by feeding them into the DNN policy for action selection, as well as into the double DQN for updating the Q-value estimates (policy). The proposed algorithm is summarized below.

Algorithm

for every time step t **do**
  Given s_t, select a_t following decaying ϵ-greedy.
  (De-)activate the APs accordingly.
  Solve the max-min SINR optimization problem.
  Observe the resulting reward r_{t+1} and next state s_{t+1}.
  Store $\{s_t, a_t, r_{t+1}, s_{t+1}\}$ in $\mathcal{B}$.
  Randomly sample M experiences from $\mathcal{B}$.
  Calculate $L(\theta_{train})$ for all samples.
  Perform gradient descent and update θ_{train}.
  Update θ_{target} using Polyak averaging.
  Update ϵ if $\epsilon > \epsilon_{min}$.
end for

IV. Simulations

We consider a downlink cell-free MIMO network with $M = 8$ regularly-placed APs having $N_t = 10$ antennas each. We vary the number of single-antenna users $K \in \{2, 4, 6\}$ that are uniformly distributed in a 60×60 m^2 simulation area. We summarize the simulation parameters in Table I.

TABLE I
Simulation Parameters

Parameter	Value
Carrier frequency f_c	2 GHz
Maximum transmit power P_T	1 W
Circuit power $P_{circuit}$	1 W
Path loss exponent n_c	2
Shadow fading variance σ_c^2	6
Noise variance σ_z^2	10^{-8}
Discount factor γ	0.99
$\epsilon_{max}, \epsilon_{min}, \epsilon_{decay}$	1, 0.05, 0.0005
Learning rate λ	0.01
Replay buffer size $\lvert\mathcal{B}\rvert$	10000
Mini-batch size M	64
Rate of averaging δ	0.001

We define 2 baseline schemes: Baseline 1, where all APs are activated and connected to all users (canonical cell-free network), and Baseline 2, where each user connects to only a single AP with the best channel (largest channel gain) then we deactivate unused APs. We evaluate the performance of the DRL-based framework in terms of the maximum guaranteed QoS or max-min SINR (Fig. 2), total power consumption (Fig. 3) and reward value (Fig. 4). We also provide the AP set solution in Table II for all schemes considering different α settings and number of users. Each 8-bit solution represents the on/off state of the 8 APs. Based on its calculation of long-term reward, the DRL framework derives the AP set that it deems best given the spatial user information.

Increasing the number of users decreases QoS and increases power consumption for all schemes. Baseline 1 achieves the

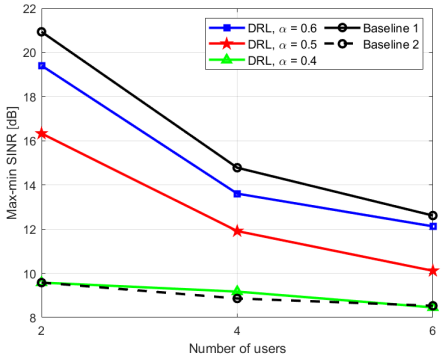

Fig. 2. Maximum guaranteed QoS for different schemes and α values with a varying number of users.

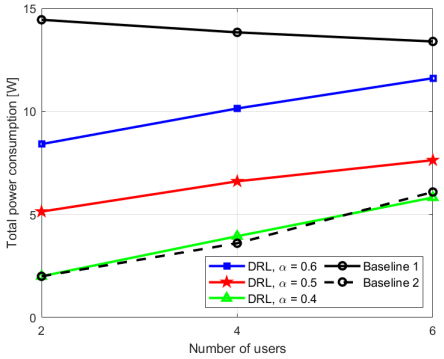

Fig. 3. Total power consumption for different schemes and α values with a varying number of users.

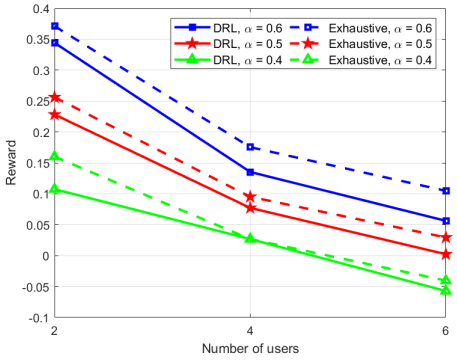

Fig. 4. Reward comparison for DRL scheme and exhaustive search with a varying number of users.

TABLE II
AP set solutions

Scheme	Number of users		
	2	4	6
DRL, $\alpha = 0.6$	00110111	00111111	01111111
DRL, $\alpha = 0.5$	00110001	00110011	00010111
DRL, $\alpha = 0.4$	00010000	00010001	00010011
Baseline 1	11111111	11111111	11111111
Baseline 2	00010000	00010001	00010111

highest SINR level in Fig. 2, since all APs are turned on and serving all users. However, that translates to the most power consumed as depicted in Fig. 3. Comparing Baseline 1 and the DRL framework when $\alpha = 0.6$, we observe at most a 1.5 dB gap in terms of QoS while possibly consuming up to 6 W more with Baseline 1. This amounts to only 7% gain in QoS while increasing power consumption by 37.5%. This highlights the power savings that we can achieve when activating only a subset of APs instead of having all APs always turned on. That is, the framework is still able to provide good QoS while consuming less power by deactivating APs that do not contribute significantly to UE performance. When $\alpha = 0.4$, the framework behaves close to Baseline 2 as they give almost the same AP set solutions in Table II.

We study the trade-off between QoS and power consumption by varying α in (17) for the DRL framework. With $\alpha = 0.5$, we prioritize neither QoS nor power savings. When we increase α, we give more importance to providing better service to all users. This is illustrated in Fig. 2 where we see the max-min SINR shifted up as we move from $\alpha = 0.5$ to 0.6. However, this is at the expense of higher power consumption in Fig. 3. From Table II, we observe that with higher α, the framework activates more APs. For example, in the case of 4 users, we have 4 active APs when $\alpha = 0.5$, which goes to 6 when $\alpha = 0.6$. Thus, demanding better QoS by increasing α forces the framework to activate more APs (to meet such requirement) that also explains the corresponding increase in power consumption. If we increase α further, we expect to approach the upper bound that is Baseline 1. On the other hand, we instruct our framework to turn off more APs when we configure α to be less than 0.5. This is again evident in Table II where we can see that, for the same example of 4 users, we now have only 2 active APs when $\alpha = 0.4$. Setting small α in (17) means that we prioritize minimizing power consumption over maximizing QoS. For this reason, the corresponding SINR values are shifted down in Fig. 2, while the framework is able to save power in Fig. 3 by deactivating more APs.

In Fig. 4, we show the obtained reward, which is the quantity being optimized by the DRL framework. As previously mentioned, QoS goes down while power goes up with increasing number of users. This justifies the decreasing trend in reward for all cases. We also performed an exhaustive search to check how close each DRL solution is to the optimal one. While the framework provides a suboptimal solution, we observe that the gap is not large, which suggests that the framework is indeed able to derive a near-optimal set of active APs considering the performance targets configured using α.

V. Conclusion

In this work, we have proposed a DRL-based scheme for improving the energy efficiency of cell-free MIMO. By exploiting the available spatial information of users, we have shown that our framework activates only a subset of APs that contributes significantly to user performance, which then translates to power savings. We have also demonstrated how we can use the system to study the interplay between QoS and power consumption, where higher QoS requirements tend to activate more APs and vice-versa. This serves as the first step for our future work in which we plan to extend the framework to consider traffic load densities instead of instantaneous static load. We will also focus on designing scalable algorithms (for instance, incorporating the power allocation in the DRL learning process rather than formulating it as a separate optimization problem) to perform large-scale network simulations.

Acknowledgment: The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged.

References

[1] I. F. Akyildiz, A. Kak and S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," in *IEEE Access*, vol. 8, pp. 133995-134030, 2020.

[2] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834-1850, Mar. 2017.

[3] E. Björnson and L. Sanguinetti, "Making Cell-Free Massive MIMO Competitive With MMSE Processing and Centralized Implementation," in *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77-90, Jan. 2020.

[4] Ericsson, *Ericsson Mobility Report*. Nov. 2020. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2020

[5] H. Q. Ngo, L. Tran, T. Q. Duong, M. Matthaiou and E. G. Larsson, "On the Total Energy Efficiency of Cell-Free Massive MIMO," in *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 25-39, Mar. 2018.

[6] L. D. Nguyen, T. Q. Duong, H. Q. Ngo and K. Tourki, "Energy Efficiency in Cell-Free Massive MIMO with Zero-Forcing Precoding Design," in *IEEE Communications Letters*, vol. 21, no. 8, pp. 1871-1874, Aug. 2017.

[7] C. F. Mendoza, S. Schwarz and M. Rupp, "Cluster Formation in Scalable Cell-free Massive MIMO Networks," *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 62-67, 2020.

[8] T. Van Chien, E. Björnson and E. G. Larsson, "Joint Power Allocation and Load Balancing Optimization for Energy-Efficient Cell-Free Massive MIMO Networks," in *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6798-6812, Oct. 2020.

[9] T. X. Vu, S. Chatzinotas, S. ShahbazPanahi and B. Ottersten, "Joint Power Allocation and Access Point Selection for Cell-free Massive MIMO," *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1-6, 2020.

[10] G. Femenias, N. Lassoued and F. Riera-Palou, "Access Point Switch ON/OFF Strategies for Green Cell-Free Massive MIMO Networking," in *IEEE Access*, vol. 8, pp. 21788-21803, 2020.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[12] V. Mnih et al., "Playing Atari with Deep Reinforcement Learning," *arXiv*, 2013. [Online]. Available: https://arxiv.org/pdf/1312.5602.pdf

[13] N. Liu et al., "A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning," *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 372-382, 2017.

[14] Z. Xu, Y. Wang, J. Tang, J. Wang and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," *2017 IEEE International Conference on Communications (ICC)*, pp. 1-6, 2017.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[16] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] H. v. Hasselt, A. Guez and D. Silver, "Deep Reinforcement Learning with Double Q-learning," *arXiv*, 2015. [Online]. Available: https://arxiv.org/pdf/1509.06461.pdf