# Joint Beamforming and Metasurface Reflection: A Lightweight Design for Energy Efficiency via Deep Reinforcement Learning

Mina Yonan *, Mohammad Galal Khafagy §, Karim Banawan ‖, Karim G. Seddik *
* School of Sciences and Engineering, The American University in Cairo (AUC), Egypt.
§ Radio Networks, Technology Department, Vodafone Egypt, Giza, Egypt.
‖ Electrical Engineering Department, Faculty of Engineering, Alexandria University.
E-mail: {m.yonan, kseddik}@aucegypt.edu, mohamed.khafagy@vodafone.com, kbanawan@alexu.edu.eg

***Abstract*—Intelligent reflecting surfaces (IRSs) continue to gain a growing research interest for their potential to support next-generation wireless communications without incurring additional power consumption. In this work, we propose a deep reinforcement learning (DRL)-driven and IRS-aided active/passive beamforming solution for multi-user multiple-input single-output (MISO) settings in beyond 5G networks, which is both lightweight and energy-efficient. The proposed solution is based on a hybrid finely-engineered design that leverages two Twin-Delayed DDPG (TD3) agents. Compared to classical optimization techniques, our numerical evaluation shows that the proposed DRL approach achieves $60\%$ reduction in online computation complexity at the expense of only $1$ dB higher power consumption.**

***Keywords*—beyond 5G, beamforming, energy efficiency, intelligent reflecting surfaces, metasurfaces, reinforcement learning.**

## I. Introduction

Reconfigurable intelligent reflecting surfaces (IRSs) [1]–[4] are continuing to receive a growing research interest in beyond 5G networks for their potential to support next-generation wireless communications without incurring additional power consumption bills. Such interest is driven by the recent advances on *electronically-reconfigurable* metasurfaces that led to generalized laws of reflection and refraction for electromagnetic wave propagation, with special emphasis on optical waves [5]–[8]. Similar to relaying, an IRS can act as an intermediate node that adaptively focuses transmissions from a source towards some designated receiver(s). Unlike relays, however, IRSs do not need to use power of their own, since they can *passively* beamform transmissions following the *software-defined* properties/orientation of their reflecting elements. In multi-user multiple-input single-output (MISO) settings, which is the scope of this work, the *timely* joint design of base station (BS) active beamformer and IRS reflection is of utmost importance for satisfactory communication.

In [9]–[12] and the references therein, different optimization techniques are proposed for beamformer design in IRS-aided single-/multi-user MISO scenarios. In [9], a single-user downlink signal-to-interference-plus-noise ratio (SINR)/throughput maximization problem is addressed under a transmit power constraint. Since no interference exists in such a single-user setting, maximum-ratio transmission (MRT) is the optimal BS transmit beamforming solution. Hence, the problem boils down to an individual optimization of the IRS reflection matrix, with the BS beamformer being a mere function of it. Nonetheless, the marginal reflection matrix optimization is found to be an NP-Hard non-convex quadratically-constrained quadratic program (QCQP). As a work-around, semi-definite relaxation (SDR) is applied yielding a solvable semi-definite program (SDP) followed by Gaussian randomization; a known approach to counteract the relaxation step.

The previous method is extended to *multi-user settings* in [10], [12], where joint design is inevitable due to interference. Specifically, a BS transmit power minimization problem is studied in [10] subject to user SINR constraints (see Section II). Since the joint problem is non-convex, a sub-optimal *alternating optimization (AO) technique* is proposed, in which a second-order cone program (SOCP) is solved for the marginal design of BS beamformer, followed by a reflection matrix SDP. The algorithm keeps alternating between the two simplified subproblems until convergence to a local optimal solution is attained. A slightly different problem variant is considered in [11] with the objective to maximize an energy efficiency metric (in bits/joule). Such AO schemes can take longer time than channel coherence time, especially in fast fading channels, which may return outdated optimized solutions. It should be also noted here that another dual-phase scheme is proposed in [10] for *quicker* results, despite being more sub-optimal compared to the alternating optimization approach.

To offer both reliable and timely optimized results, machine learning (ML)-driven solutions are proposed for the problem at hand with/without direct BS-user links for single-user [13]–[15] and multi-user [16], [17] settings. In [15], a single-user downlink MISO system is studied, and the IRS reflection matrix is designed via a deep reinforcement learning (DRL)-based approach for received throughput/SINR maximization. The DRL algorithm in [15] is deep deterministic policy gradient (DDPG), to allow for continuous state/action spaces. The BS beamformer is fixed as a function of the IRS reflection matrix and known instantaneous channels to be the MRT solution. In this single-user scenario taking received throughput/SINR maximization as objective, the MRT solution is optimal. In multi-user scenarios when interference exists, however, joint BS beamformer/IRS reflection design should be addressed.

The multi-user joint beamforming/reflection problem remains to be of more practical interest to cater for the growing spectrum demand in beyond 5G networks along with its known scarcity, via spectrally-efficient non-orthogonal multiple access (NOMA) approaches. Such a *timely* beamformer design is a key enabler of the technology, while the available optimization algorithms are either too suboptimal, or require extended time to converge to the solution of iterative optimization methods. For this problem, ML can play a vital role in expediting the

optimization process, at the expense of a prior offline training.

In this paper, we propose a DRL-driven solution to the multi-user communication setting in [10], taking the classical AO solution as a benchmark. The proposed solution is both lightweight and energy-efficient, and leverages two Twin-Delayed DDPG (TD3) agents. Compared to classical AO techniques, our results show that the proposed DRL approach achieves $60\%$ reduction in online computation complexity at the expense of only $1$ dB higher in power consumption.

The rest of the paper is organized as follows. In section II, the system model is detailed, followed by a summary of the AO technique proposed in [10]. Our DRL-based solution is explained in section III. Numerical results are presented in section IV, and finally concluding remarks in section V.

## II. System Model, Problem Formulation, and Classical Optimization Solution

### A. System Model

Consider a downlink (DL) multi-user setting where an $N_{BS}$-antenna BS communicates with $N_U$ single-antenna users in the presence of an $M$-element IRS, with $N_{BS} \geq N_U$. The received signal at all users is given by the vector:

$$\boldsymbol{y} = \underbrace{(\boldsymbol{H}_r^H \boldsymbol{\Theta} \boldsymbol{H}_t + \boldsymbol{H}_d^H)}_{\boldsymbol{H}^H} \boldsymbol{W}\boldsymbol{s} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{n} \in \mathbb{C}^{N_U}$ denote the vectors corresponding to received signals, transmitted symbols, and receiver additive white Gaussian noise (AWGN), respectively. Also, the narrowband channel matrices $\boldsymbol{H}_t \in \mathbb{C}^{M \times N_{BS}}$, $\boldsymbol{H}_r \in \mathbb{C}^{M \times N_U}$, and $\boldsymbol{H}_d \in \mathbb{C}^{N_{BS} \times N_U}$ represent the first hop (transmitter) channel from the BS to the IRS, the second hop (receiver) channel from the IRS to the set of users, and the direct channel from the BS to the users, respectively, which are assumed to follow a circularly-symmetric complex Gaussian distribution; $\boldsymbol{H}_{\mathrm{i}} \sim \mathcal{CN}\{\boldsymbol{0}, \pi_{\mathrm{i}} \boldsymbol{I}\}$, where $\mathrm{i} \in \{\mathrm{t}, \mathrm{r}, \mathrm{d}\}$. For ease of notation, it is assumed that the path loss effect is absorbed inside the channel gain $\pi_{\mathrm{i}}$. The beamformer matrices $\boldsymbol{W} \in \mathbb{C}^{N_{BS} \times N_U}$ and $\boldsymbol{\Theta} \in \mathbb{C}^{M \times M}$ respectively represent the active and passive beamforming at the BS and IRS. The SINR at the $n$th user,

$$\mathrm{SINR}_n = \frac{|\boldsymbol{h}_n^H \boldsymbol{w}_n|^2}{\sum_{k \neq n}^{N_U} |\boldsymbol{h}_n^H \boldsymbol{w}_k|^2 + \sigma^2}, \; n \in \{1, \cdots, N_U\} \tag{2}$$

where $\boldsymbol{h}_n^H = (\boldsymbol{h}_{r,n}^H \boldsymbol{\Theta} \boldsymbol{H}_t + \boldsymbol{h}_{d,n}^H)$. In general, $\boldsymbol{a}_n$ is the $n$th column of matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N]$.

### B. Classical Optimization Benchmark

The authors in [10] study the following transmit power minimization problem to jointly design $\boldsymbol{W}$ and $\boldsymbol{\Theta}$ subject to SINR quality-of-service (QoS) constraints:

$$\min_{\boldsymbol{W}, \boldsymbol{\Theta}} \quad \mathrm{tr}\{\boldsymbol{W}^H \boldsymbol{W}\} \tag{3}$$

$$\text{s.t.} \quad \mathrm{SINR}_n \geq \gamma_n, \; \forall n \in \{1, \cdots, N_U\}, \tag{4}$$

$$0 \leq \theta_m \leq 2\pi, \; \forall m \in \{1, \cdots, M\}, \tag{5}$$

where $\gamma_n$ is a minimum QoS threshold on the SINR for the $n$th user, while $\theta_m$ is the phase shift at the $m$th IRS element. In [10], $\boldsymbol{\Theta}$ is assumed to have the following form, in terms of the reflection amplitudes $\{\beta_m\}_{k=1}^M$ and phase shifts $\{\theta_m\}_{k=1}^M$:

$$\boldsymbol{\Theta} = \mathrm{diag}(\beta_1 e^{j\theta_1}, \cdots, \beta_m e^{j\theta_m}), \tag{6}$$

also assuming $\beta_m = 1, \forall m \in 1, \cdots, M$.

### C. Summary of Alternating Optimization Benchmark in [10]

The optimization variables in problem (3) are coupled in the SINR constraint, and hence, the problem is non-convex. As proposed in [10], an iterative optimization scheme is considered where either $\boldsymbol{W}$ or $\boldsymbol{\Theta}$ is optimized at a time, with the other kept fixed. First, when $\boldsymbol{\Theta}$ is given, the power minimization problem under SINR constraints can be cast as an SOCP in $\boldsymbol{W}$ that can be efficiently solved using standard tools like CVX [18], [19]. On the other hand, when $\boldsymbol{W}$ is fixed, the problem boils down to a feasibility-check problem. It can be also put on an SDP form with relaxing the rank-one condition on a vector outer product. The transformed problem can be efficiently solved using CVX, followed by a Gaussian randomization step to obtain a rank-one feasible vector.

*1) Optimization of Active Beamformer $\boldsymbol{W}$:* Specifically, selecting an arbitrary initial $\boldsymbol{\Theta}$ that satisfies (5), (6), and $\mathrm{rank}(\boldsymbol{H}) = N_U$ to guarantee feasibility, we first solve,

$$\min_{\boldsymbol{W}, P} \quad P \tag{7}$$

$$\text{s.t.} \quad \mathrm{SINR}_n \geq \gamma_n, \forall n \in \{1, \cdots, N_U\}, \tag{8}$$

$$\mathrm{tr}\{\boldsymbol{W}^H \boldsymbol{W}\} \leq P, \tag{9}$$

which is equivalent to the following SOCP form:

$$\min_{\boldsymbol{W}, P} \quad P \tag{10}$$

$$\text{s.t.} \quad \begin{bmatrix} \sqrt{\frac{1}{\gamma_n}} \boldsymbol{w}_n^H \boldsymbol{h}_n \\ \boldsymbol{W}_{(\backslash n)}^H \boldsymbol{h}_n \\ \sigma \end{bmatrix} \succcurlyeq 0, \forall n, \tag{11}$$

$$\|\boldsymbol{W}(:)\| \leq \sqrt{P}, \tag{12}$$

where $\boldsymbol{W}(:) = \left[\boldsymbol{w}_1^H, \boldsymbol{w}_2^H, \cdots, \boldsymbol{w}_{N_U}^H\right]^H \in \mathbb{C}^{N_{BS} N_U \times 1}$ represents a stacked vector form of $\boldsymbol{W}$, while $\boldsymbol{W}_{(\backslash n)}$ is the $\boldsymbol{W}$ matrix with the $n$th column omitted. The optimal $\boldsymbol{W}^*$ satisfies the SINR constraint with equality.

*2) Optimization of Reflection Matrix $\boldsymbol{\Theta}$:* With $\boldsymbol{W}$ in hand, we find $\boldsymbol{\Theta}$ via the following relaxed SDP (for detailed explanation, refer to [10]) followed by a Gaussian randomization:

$$\max_{\boldsymbol{V}, \{\alpha\}_{n=1}^{N_U}} \quad \sum_{n=1}^{N_U} \alpha_n \tag{13}$$

$$\text{s.t.} \quad c_{n,n} \geq \gamma_n \sum_{k \neq n}^{N_U} c_{n,k} + \alpha_n, \forall n, \tag{14}$$

$$\alpha_n \geq 0, \forall n, \tag{15}$$

$$\boldsymbol{V} \succcurlyeq 0, v_{m,m} = 1, \forall m \in \{1, \cdots, M+1\}, \tag{16}$$

with

$$c_{n,k} = \mathrm{tr}(\boldsymbol{R}_{n,k} \boldsymbol{V}) + |b_{n,k}|^2, \tag{17}$$

$$a_{n,k} = \mathrm{diag}(\boldsymbol{h}_{r,n}^H) \boldsymbol{H}_t \boldsymbol{w}_k, \tag{18}$$

$$b_{n,k} = \boldsymbol{h}_{d,n}^H \boldsymbol{w}_k, \tag{19}$$

**Algorithm AltOpt** Alternating Optimization in [10]

**Input:** $\boldsymbol{H}_t$, $\boldsymbol{H}_r$, $\boldsymbol{H}_d$, $\{\sigma_n, \gamma_n\}_{n=1}^{N_U}$.
**Output:** $\boldsymbol{W}$, $\boldsymbol{\Theta}$.
1: INITIALIZE:
- $\boldsymbol{\Theta}$ satisfying (5), (6), and $rank(\boldsymbol{H}) = N_U$.
- error = $10^6$ (large value), error threshold $\epsilon = 1e-2$.

2: **while** error $\geq \epsilon$ **do**
3: SOLVE P1 in (10)
4: SOLVE P2 in 13
5: **end while**

$$\boldsymbol{R}_{n,k} = \begin{bmatrix} a_{n,k}a_{n,k}^H & a_{n,k}b_{n,k}^H \\ a_{n,k}^H b_{n,k} & 0 \end{bmatrix}. \quad (20)$$

The Gaussian randomization (to obtain $\boldsymbol{\Theta}$ from $\boldsymbol{V}$):

1) Apply the eigenvalue decomposition $\boldsymbol{V} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^H$.
2) Generate a sufficiently large number of realizations for a circularly-symmetric complex Gaussian random vector $\boldsymbol{r} \in \mathbb{C}^{M+1\times 1}$, with $\boldsymbol{r} \sim \mathcal{CN}\{\boldsymbol{0}, \boldsymbol{I}_{M+1}\}$.
3) Generate $\tilde{\boldsymbol{v}} = \boldsymbol{Q}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{r}$.
4) Select $\tilde{\boldsymbol{v}}$ in which $\tilde{\boldsymbol{V}} = \tilde{\boldsymbol{v}}\tilde{\boldsymbol{v}}^H$ satisfies the constraint in (14) while achieving maximum $\sum_n \alpha_n$ value (maximizes the objective function/the aggregate surplus of desired signal power over that of interference plus noise scaled by $\gamma_n$), with $\alpha_n \geq 0$.

## III. Proposed Reinforcement-Learning-Driven Solution for Beamforming Matrices Design

We present our proposed solution to the problem of designing energy-efficient transmit beamforming and phase-shift matrices subject to per-user QoS constraints. Our solution is based on DRL techniques [20]–[22]. We first recast the problem as a Markov Decision Process (MDP), which is the formal mathematical framework for dealing with RL problems [20]. This implies specifying suitable state and action spaces corresponding to our wireless communication model in addition to a reward function that reflects the energy efficiency and QoS requirements. Next, we describe the basic building block of our DRL-based solution, namely the TD3 agent. Finally, we present a novel selection combining agent (SCA) to avoid QoS violations *without being exceedingly conservative in exploring the space* of beamforming matrices as we will show next.

### A. Reinforcement Learning (RL) Framework

In this section, we introduce the RL framework, which we will use next. The formal mathematical abstraction of the RL is obtained by describing the underlying MDP [20]. In RL (and by extension DRL), at discrete time instants $t = 0, 1, 2\cdots$, there exists a decision-making node (a.k.a., the *agent*) operating in an *environment*. The agent picks an *action* $A(t)$ belonging to an action space $\mathcal{A}$. The agent perceives the environment through a pre-specified compact representation (a.k.a., the *state*), which is denoted by $S(t)$. $S(t)$ belongs to a state space $\mathcal{S}$. After applying $A(t)$, the state of the environment changes to $S(t+1)$ and the agent receives a *reward*, $R(t)$, which reflects the agent's objective and constraints. The RL agent aims at finding the optimal *policy*, i.e., the optimal sequence of actions $\Omega^* = \{A^*(t)\}_{t=0}^{\infty}$ that maximizes the long-term average return, i.e., the sum discounted reward,

$$\Omega^* = \arg\max_{\Omega} \lim_{T\to\infty} \quad \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R(t)\right] \quad (21)$$

where $\gamma$ is the discount factor, and $T$ is the total number of steps in an episode[1]. In our beamformer design, the environment is the wireless system including the base station, users, and the IRS. To completely specify the underlying MDP, we need to characterize $\mathcal{S}$, $\mathcal{A}$, and $R(t)$ as we will do next.

*1) State Space:* The state $S(t) \in \mathcal{S}$ at time step $t$ is composed of the channels $\boldsymbol{H}_{\mathrm{i}}$, $\mathrm{i} \in \{\mathrm{t}, \mathrm{r}, \mathrm{d}\}$, beamforming matrix $\boldsymbol{W}$, and reflection matrix $\boldsymbol{\Theta}$. Since neural networks cannot take a complex number as an input, the real and imaginary parts are separated as independent inputs. The state can be hence formally expressed as:

$$S(t) = \begin{bmatrix} \boldsymbol{h}_{t,vec}^{\mathrm{T}} & \boldsymbol{h}_{r,vec}^{\mathrm{T}} & \boldsymbol{h}_{d,vec}^{\mathrm{T}} & \boldsymbol{w}_{vec}^{\mathrm{T}} & \boldsymbol{\theta}_{vec}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \quad (22)$$

where $\boldsymbol{x}_{vec}$, with $\boldsymbol{x} \in \{\boldsymbol{h}, \boldsymbol{w}, \boldsymbol{\theta}\}$, is a column vector form of matrix $\boldsymbol{X}$, stacked column-by-column, with real and imaginary parts separately stacked. The input length is hence twice the total number of the original complex elements; that is $2MN_{BS}$ for $\boldsymbol{H}_t$, $2MN_U$ for $\boldsymbol{H}_r$, $2N_{BS}N_U$ for the beamforming matrix, and $2N$ for the diagonal vector of the IRS phases.

*2) Action Space:* The action space $\mathcal{A}$ is the space of the transmit beamforming matrices $\boldsymbol{W} \in \mathbb{C}^{N_{BS}\times N_U}$ concatenated with the space of the phase shift matrices $\boldsymbol{\theta} \in \mathbb{C}^{M\times M}$. Consequently, the action at time instant $t$ is given by:

$$A(t) = \begin{bmatrix} \boldsymbol{w}_{vec}^{\mathrm{T}} & \boldsymbol{\theta}_{vec}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \quad (23)$$

Similarly, the real and imaginary components of elements of $A(t)$ are dealt with separately in the agent's implementation.

*3) Reward Function:* The reward function needs to assess the quality of the agent's action in the sense of fulfilling its objective subject to problem constraints. In this work, the objective of the agent is to minimize power consumption. This can be represented by $R_1(t) = -\mathrm{tr}\{\boldsymbol{W}^H\boldsymbol{W}\}$, where the negative sign signifies that the agent is actually minimizing the power consumption. The agent also needs to further ensure that the agent satisfies the QoS constraints, i.e., the rate of the $k$th user, $\rho_k$, satisfies $\rho_k \geq \rho_{\mathrm{th}}$ for all $k = 1, 2, \cdots, N_U$. Hence, the $k$th user QoS constraint is represented by $R_{2,k}(t) = \min\{\rho_k - \rho_{\mathrm{th}}, 0\}$, to impose a penalty only upon constraint violation. A third positive part in the reward definition, $f$, is added after $N$ episodes when $\rho = \min\{\rho_k\}_{k=1}^{N_U}$ falls within $\epsilon$ portion of the target $\rho_{\mathrm{th}}$; $f = Ku(t-N)u\left(\epsilon - \left|\frac{\rho-\rho_{th}}{\rho_{th}}\right|\right)$, where $u(\cdot)$ is the unit step function. This term is added to encourage convergence. Moreover, to avoid having an agent picking the target QoS constraints with high variance, we choose $f$ to be zero until specific episode $N$. We combine all terms in the following function:

$$R(t) = -\mathrm{tr}\{\boldsymbol{W}^H\boldsymbol{W}\} + \alpha\frac{t}{T}\sum_{k=1}^{N_U}\min\{\rho_k - \rho_{\mathrm{th}}, 0\} + f \quad (24)$$

[1]In theory, $T \to \infty$. However, in this work, we adopt episodic training, where $T$ denotes a pre-specified finite number.

where $\alpha$ is a hyper-parameter that signifies how much the agent cares about adhering to the QoS constraints rather than minimizing the consumed power, and $T$ is the total number of steps per episode. Our reward function penalizes the QoS violations more near the end of the episode. This is to disrupt the agent's convergence if it opts to violate the QoS constraints at the end of the training by increasing the penalty.

### B. Basic Block of Our Proposed RL Technique: TD3 Agent

The basic building block of our SCA algorithm is the TD3 [23], an actor-critic technique. In actor-critic techniques, two separate deep neural networks (DNNs) are employed. The *actor* network picks a proper action $A(t) \in \mathcal{A}$ based on the current state $S(t)$. The *critic*, on the other hand, assesses the actor choice by evaluating the Q-value corresponding to the pair $(S(t), A(t))$ from an independently trained DNN.

TD3 is an extension of the deep deterministic policy gradient (DDPG) actor-critic technique [24] with three major changes. First, TD3 uses two independently trained critic functions. The predicted Q-value is the minimum Q-value of both critics. This avoids overestimation errors and provides a more stable approximation. Second, TD3 employs delayed updates for the target DNNs. Specifically, the target actor and critic DNNs are updated every $T_u$ time steps (in contrast to updating every time step in DDPG). This mitigates agent divergence. Finally, TD3 adds clipped noise to the target policy before updating the weights. This ensures the validity of the target prediction in the neighborhood of the actual action stored in the replay buffer, providing Q-value estimation continuity.

### C. Proposed Scheme: Selection Combining Agent (SCA)

The QoS violation is a consequence that RL techniques aim at optimizing the long-term return (sum discounted reward) in the *expected sense* [20] and ignores less likely violation events. A straightforward solution to this problem is to operate the agent subject to much more strict QoS constraints to minimize the likelihood of a violation event. Nevertheless, this results in a severe transmit power surge. Our SCA algorithm aims to balance the QoS violation likelihood and power efficiency.

To that end, we train $L$ different TD3 agents (with the architecture presented in Section III-B) in terms of the QoS constraint. More specifically, the first agent is trained with $\rho_{\text{th}}^{[1]} = \rho_{th}$, which is the minimum target rate that should attained for every end-user. The $\ell^{\text{th}}$ agent is trained such that the rate constraint in the reward function is given by

$$\rho_{\text{th}}^{[\ell]} = \rho_{\text{th}} + (\ell - 1)\Delta, \quad \ell = 1, \cdots, L \tag{25}$$

where $\Delta$ is the step size of the rate constraint. That is, each agent is a slightly *more conservative* agent in terms of satisfying the QoS (rate) constraint as the likelihood of violating the actual rate threshold $\rho_{\text{th}}$ is a monotonically decreasing function in $\ell$. This presents an interesting tradeoff: as $\ell$ increases, the QoS violation decreases, while the transmitted power increases as a result of the more strict QoS constraint.

We simultaneously run the $L$ agents (in the online phase). The resultant beamforming $\boldsymbol{W}^{[\ell]}$ and phase shift $\boldsymbol{\Theta}^{[\ell]}$ matrices are evaluated in terms of the achievable rate. The pair $(\boldsymbol{W}^{[\ell]}, \boldsymbol{\Theta}^{[\ell]})$ with the least power consumption and at the same time satisfying the original rate constraint is selected. The flow of the algorithm in Algorithm SCA.

**Algorithm SCA** TD3-based Selection Combining Agent

**Input:** $\boldsymbol{H}_t$, $\boldsymbol{H}_r$, $\boldsymbol{H}_d$, $\rho_{th}$, $L$, $\Delta$.
**Output:** $\boldsymbol{W}$, $\boldsymbol{\Theta}$.
TRAINING:
**for** $\ell = 1, \cdots, L$ **do**
  $\rho_{\text{th}}^{[\ell]} = \rho_{\text{th}} + (\ell - 1)\Delta$
  Train TD3 ($\rho_{\text{th}}^{[\ell]}$)
**end for**
ONLINE OPERATION:
**for** $\ell = 1, \cdots, L$ **do**
  Get $\boldsymbol{W}^{[\ell]}$, $\boldsymbol{\Theta}^{[\ell]}$ corresponding to TD3 ($\rho_{\text{th}}^{[\ell]}$)
  Calculate $\text{tr}\{{\boldsymbol{W}^{[\ell]}}^H \boldsymbol{W}^{[\ell]}\}$
  Calculate $\rho_k = \log_2(1 + \text{SINR}_k^{[\ell]}),\ \forall k$
**end for**
SELECTION:
**if** $\rho_k < \rho_{th}$ **then**
  Ignore $\boldsymbol{W}^{[\ell]}$, $\boldsymbol{\Theta}^{[\ell]}$
**else**
  $\ell^* = \arg\min_{\ell} \text{tr}\{{\boldsymbol{W}^{[\ell]}}^H \boldsymbol{W}^{[\ell]}\}$
**end if**
$\boldsymbol{W} = \boldsymbol{W}^{[\ell^*]}$ and $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{[\ell^*]}$

## IV. Numerical Results

We evaluate the performance of our proposed DRL-based algorithm. Our objective is to *jointly* design the beamforming matrices $\boldsymbol{W}$ and $\boldsymbol{\Theta}$ to minimize the transmit power, i.e., $\text{tr}\{\boldsymbol{W}^H \boldsymbol{W}\}$, subject to per-user minimum rate constraints $\{\rho_k\}_{k=1}^{N_U}$. The *online* beamforming design needs to be completed with minimal computational complexity (which corresponds to a minimum processing time) to match (near) real-time requirements of the next-generation cellular networks.

### A. Evaluation Setup

In our evaluations, the channel matrices $\boldsymbol{H}_{\text{i}} \sim \mathcal{CN}\{\boldsymbol{0}, \pi_{\text{i}} \boldsymbol{I}\}$ for $\text{i} \in \{\text{t}, \text{r}\}$, are randomly generated with the channel gains values, $\pi_{\text{i}}$, summarized in Table II, while the direct link is assumed unavailable due to severe path loss, only for ease of exposition and without loss of generality, i.e., $\pi_{\text{d}} = 0$. The direct channel elements are hence omitted from the DRL state $S(t)$. We perform our evaluations for a system with a base station equipped with $N_{BS} = 8$ antennas, and an IRS containing $M = 8$ elements. Our system serves $N_U = 2, 3, \cdots, 7$ users. We assume that all users have a normalized rate (i.e., spectral efficiency) constraint of $\rho_k = 2$ b/s/Hz for all $k$. We rely on Shannon's capacity expression for calculating the achievable spectral efficiency of a specific user. More specifically,

$$\rho_k = \log_2(1 + \text{SINR}_k) \tag{26}$$

where $\text{SINR}_k$ can be evaluated using (2).

In our evaluation setup, it suffices to use an SCA with two agents. We train the first agent with a spectral efficiency threshold of $\rho_{\text{th}}^{[1]} = 2$ b/s/Hz, while the second agent is trained to have $\rho_{\text{th}}^{[2]} = 2.5$ b/s/Hz (i.e., we choose $\Delta = 0.5$ b/s/Hz as we will explain in Section IV-B). Thus, the second agent is more conservative. The selection mechanism is such that the learned action from agent 1 is adopted whenever it satisfies the rate constraint $\rho_k = 2$, otherwise, the learned action from the second agent is employed. The two agents use TD3 as their DRL technique. The architecture of actor and critic

TABLE I: Proposed TD3 Actor/Critic Network Architecture.

| No. of users | Actor Network | | Critic Network | |
|---|---|---|---|---|
| 2 | 64 | relu | 256 | relu |
| | 64 | relu | 256 | relu |
| | $2N_{BS}M + N_{BS}N_U$ | tanh | 1 | None |
| 3 | 128 | relu | 256 | relu |
| | 128 | relu | 256 | relu |
| | $2N_{BS}M + N_{BS}N_U$ | tanh | 1 | None |
| 4,5,6 | 128 | relu | 512 | relu |
| | 128 | relu | 512 | relu |
| | $2N_{BS}M + N_{BS}N_U$ | tanh | 512 | None |

TABLE II: Optimized hyper-parameters for TD3 agents.

| Parameter | Description | Value |
|---|---|---|
| $\pi_{\text{i}}$ | Channel Gains | 1e-6 |
| $\gamma$ | Discounted rate for future reward | 0.99 |
| $\mu$ | Buffer size for experience replay | 1e4 |
| $B$ | Number of episodes | 1e4 |
| $T$ | Number of steps in each episode | 1e4 |
| $\alpha$ | Scaling factor | 2e6 |
| $N_{\text{policy}}$ | Policy Update Frequency in steps | 1e4 |
| $N_{\text{target}}$ | Target Update Frequency in steps | 1e4 |
| $W$ | Number of the experiences per a mini-batch | 16 |
| $N$ | Start episode for positive reward in $f$ | 9000 |
| $K$ | Positive reward value in $f$ | 1000 |
| $\epsilon$ | Margin threshold in $f$ | 0.1 |

networks for both agents is shown in Table I. Each network consists of 3 layers. The activation function and number of neurons are specified in terms of the number of users $N_U$. Extensive hyper-parameter tuning is performed to optimize the convergence and minimize violations. Our optimized hyper-parameters are summarized in Table II. Tuning the hyper-parameters mainly depends on the nature of the environment. Since our environment is dynamic, the number of steps and episodes should be relatively larger than environments with static attributes (e.g., same channel realizations) among training episodes. In addition, the agent's convergence in such environments requires longer training time (e.g., thousands of training episodes). Furthermore, the target network update frequency should be much less frequent for the same reason.

All aforementioned agents are implemented using the Reinforcement Learning toolbox in MATLAB. We performed our training on 64 cores with 1TB of memory. We compare the performance of our proposed DRL-based SCA algorithm with the alternating optimization algorithm AltOpt [10]. In the sequel, we present and discuss our numerical results.

### B. Convergence and Violations

Fig. 1 shows the convergence behavior of the average reward function versus the number of training episodes for the case of having $N_U = 2$ users. The reward in Fig. 1 is obtained by averaging the instantaneous reward $R(t)$ in (24) over the number of steps per episode $T =$ 1e4 steps. Fig. 1 shows that the average reward starts from extremely low reward $\approx -20\times10^7$ upto the 2000th episode. This is natural as the agent randomly explores the action space at the beginning of the training. This results in choosing beamforming matrices that incur high transmit power and/or unsatisfactory spectral efficiency (below $\rho_{\text{th}}$). In both cases, the returned reward would be a large negative value, heavily penalizing the agent, especially on rate violations. The average reward increases (almost) linearly between episodes 2000 to 9000 as the agent decreases its exploration rate and relies more on exploiting the learned optimal beamformers. The learned beamformers gradually become more and more efficient, i.e., incur less power consumption and fewer rate-constraint violations. The average reward crosses the zero value starting from the 9000th episode (yielding positive cumulative reward for the first time), indicating that the agent learns to use the most power-efficient beamformer without incurring significant violations.

Fig. 2 shows the violation percentage of our proposed DRL-based agent versus the number of users. In this experiment, we use a single agent, which is trained to satisfy the rate constraint $\rho_{\text{th}} \geq 2$ at the end of the training phase. Fig. 2 shows a significant violation percentage ($12\% - 25\%$ in the training phase, and $25\% - 35\%$ in the testing phase). This implies that using a *single* DRL-agent is not sufficient to avoid QoS degradation even with extended training of $1e4$ episodes, each consisting of $1e4$ steps. Furthermore, it can be noticed that violations increase with the number of users. This is expected as the problem becomes more constrained. Hence, the feasible space of beamformers shrinks, and the DRL agent is more likely to pick infeasible beamforming matrices.

We analyze the violation percentage in Fig. 3. Specifically, we plot the histogram of the achievable rate at the $5000^{\text{th}}$ and $10000^{\text{th}}$ episodes. For fair comparison, we normalize the histogram by the number of steps (i.e., we show a discretized probability density function of the achievable rate) and show only the outer (envelope) of the histogram. Fig. 3 shows that the rate distribution becomes narrower as the number of episodes increases. This implies that violation probability decreases as the training is prolonged, which confirms that our agent is successfully converging to a good choice of beamforming matrices[2]. Furthermore, Fig. 3 shows that the least possible achievable rate is approximately 1.5 at the $10000^{\text{th}}$ episode. This justifies our choice of having two agents to constitute our SCA. Consequently, by choosing the conservative agent to have $\rho_{\text{th}} = 2.5$ b/s/Hz (i.e., $\Delta = 0.5$ b/s/Hz) the likelihood of violations diminishes as the rate distribution for the conservative agent would be approximately translated by $\Delta = 0.5$ b/s/Hz.

### C. Power Consumption Results

Fig. 4 shows the achievable power consumption $\text{tr}\{\boldsymbol{W}^H\boldsymbol{W}\}$ in ($\text{dB}_m$) versus the number of users $N_U$. Our first observation is that power consumption increases as the number of users increases. This is expected since the optimization problem becomes more strict as the rate constraints increase. We compare our proposed SCA algorithm with the classical optimization benchmark AltOpt in [10], which is based on alternating optimization (AO). For each algorithm, we show the results when the rate constraint takes the values $\rho_{\text{th}} = 2, 2.5$ b/s/Hz. Our results show that, by ignoring violation events, our DRL-based agent at $\rho_{\text{th}} = 2$ b/s/Hz outperforms its AO counterpart by $\approx$ 1dB (saving

[2]The optimal rate distribution would be a single *impulse* at $\rho_{\text{th}} = 2$ b/s/Hz.

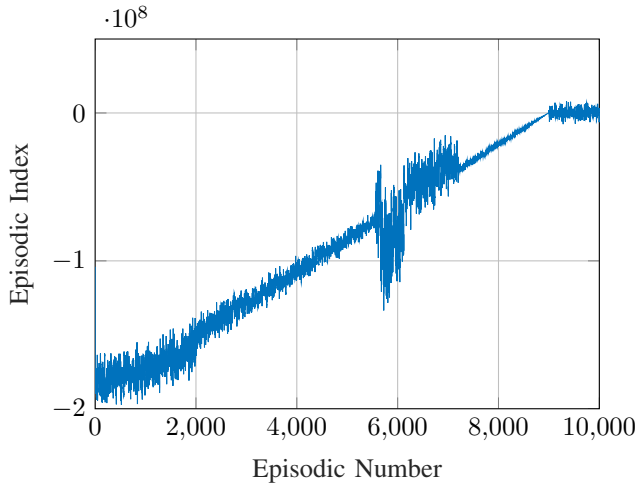


Fig. 1: Reward convergence results for $N_U = 2$ users case. Reward converges starting from episode 9112.

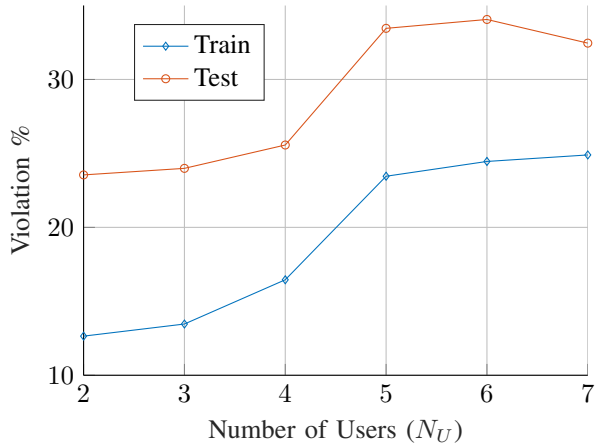


Fig. 2: Agent rate-constraint violation percentage of our DRL-based agent versus the number of user $N_U$ in the training and testing phases. The agent is trained to satisfy $\rho_{\text{th}} \geq 2$.

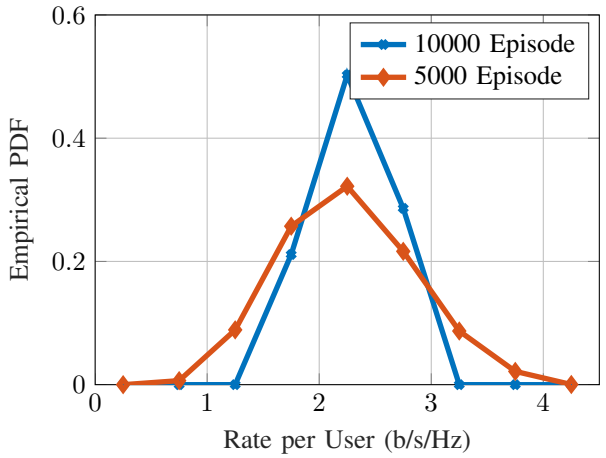


Fig. 3: Envelope of the histogram of the achievable rate for 5000 and 10000 episodes. The rate becomes more confined as training extends. The minimum achievable rate $\approx$ 1.5b/s/Hz.

$\approx$ 25% of the power consumption). The same observation can be drawn for the solutions of the $\rho_{\text{th}} = 2.5$ b/s/Hz. Nevertheless, our combined SCA agent is slightly worse (by

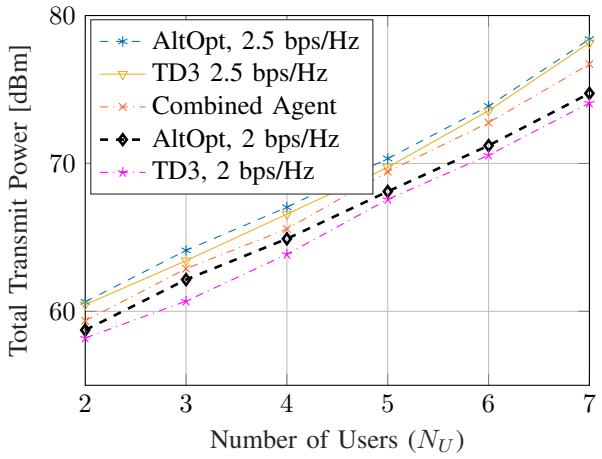


Fig. 4: Power consumption ($\text{dB}_m$) versus the number of users $N_U$. Our agent SCA (combined agent) is slightly worse than the AltOpt with significantly less computational complexity.

$\approx$ 1.5dB) than the AO-based optimization (with $\rho_{\text{th}} = 2$ b/s/Hz). This is predictable as our algorithm opts to be extremely conservative to avoid rate-constraint violations by switching to higher rate threshold[3]. Our SCA still outperforms the AO if we operated it in a conservative manner with $\rho_{\text{th}} = 2.5$ b/s/Hz by $\approx$ 1.5dB as well (our SCA is almost midway between the AO agents at $\rho_{\text{th}} = 2$ and 2.5). This shows the efficacy of our proposed technique. The slight power loss in the SCA agent is the cost of avoiding violations. Specifically, using the SCA agent incurs in return 239 violation events out of $10^6$ channel realizations.

### *D. Computational Complexity Results*

Finally, we assess the computational complexity of our proposed SCA algorithm by measuring the processing time needed to find the optimal beamforming matrices. We compare the processing times of our proposed scheme and the AO benchmark on the *same computing machine*. Fig. 5 shows the processing time (in seconds) of the optimization scheme versus the number of users. Fig. 5 shows that for both algorithms, the processing time increases as the number of users increases. Nevertheless, the processing time of our scheme increases linearly with a miniature slope ($\approx$ 0.4 s increase for every additional user). This is due to the increase in the complexity of the actor and critic networks as $N_U$ increases. This in turn increases the required calculations in the neural networks. On the other hand, the processing time of the AO optimization follows a piecewise linear pattern. When $N_U \leq 4$, the processing time increases by 5 s for every additional user. When $N_U > 4$, the slope reduces to 2 s/user. This implies that at $N_U = 7$ users, our proposed algorithm reduces the computational complexity by $\approx$ 60% as the processing time reduces from 26 s in AO to 10 s using our proposed algorithm. This difference in computational complexity is fundamental as the DRL-based approaches rely on neural networks, which merely perform

[3]Our results can be enhanced in terms of the power consumption and violations by either extending the training episodes, decreasing the rate increment $\Delta$, and/or increasing the number of agents of the SCA. Most notably, satisfying the main objective of this work, the obtained results show an extreme reduction of the processing time (as will be shown in Section IV-D) at the expense of a slight trade-off in the power efficiency.

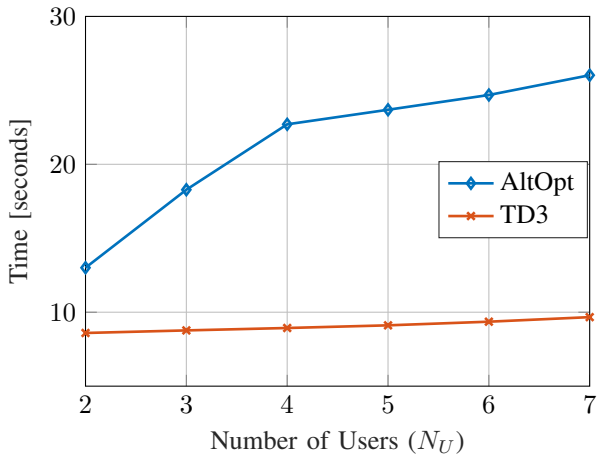


Fig. 5: Processing time versus the number of users. Our proposed scheme reduces the processing time needed of obtaining the optimal beamformer by $\approx 60\%$ at $N_U = 7$ users.

simple matrix multiplications (simple deterministic function evaluation) in the testing phase. These operations are done in a single shot and do not require any iterations. The AO, however, is actually solving a hard optimization problem (SOCP or SDP) once it acquires new channel matrices. Solving these optimization problems requires iterating between solving for $\boldsymbol{W}$ and $\boldsymbol{\Theta}$. I.e., the DRL-based approaches trade *online* computational complexity by *offline* computational complexity.

## V. Conclusion

In this paper, we considered the problem of designing energy-efficient transmit beamforming and IRS phase shift matrices for a MISO setting under per-user QoS constraints. Our objective is to design a lightweight responsive optimization designing framework to satisfy adaptive (near) real-time requirements in 5G and beyond. We proposed a DRL solution to the aforementioned problem based on the TD3 technique. The rationale is to trade extensive offline computational complexity in RL training (which is generally affordable) by simplified online computations. Furthermore, we proposed a novel SCA technique to avoid QoS violations without being exceedingly conservative in exploring the action space. We showed from extensive numerical evaluations that our SCA scheme achieves similar energy efficiency as the state-of-the-art AO methods (within 1dB-gap), with radically reduced online computations (up to $60\%$ reduction), and negligible QoS violations (unlike the direct application of DRL techniques).

## References


[1] M. ElMossallamy, H. Zhang, L. Song, K. Seddik, Z. Han, and G. Li, "Reconfigurable intelligent surfaces for wireless communications: Principles, challenges, and opportunities," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 990–1002, 2020.

[2] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M. Alouini, and R. Zhang, "Wireless communications through reconfigurable intelligent surfaces," *IEEE Access*, vol. 7, pp. 116 753–116 773, 2019.

[3] M. Di Renzo, M. Debbah, D.-T. Phan-Huy, A. Zappone, M.-S. Alouini, C. Yuen, V. Sciancalepore, G. C. Alexandropoulos, J. Hoydis, H. Gacanin *et al.*, "Smart radio environments empowered by reconfigurable AI meta-surfaces: an idea whose time has come," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 129, 2019.

[4] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *arXiv preprint arXiv:1902.10265*, 2019.

[5] N. Yu, P. Genevet, M. A. Kats, F. Aieta, J.-P. Tetienne, F. Capasso, and Z. Gaburro, "Light propagation with phase discontinuities: Generalized laws of reflection and refraction," *Science*, vol. 334, no. 6054, pp. 333–337, 2011. [Online]. Available: https://science.sciencemag.org/content/334/6054/333

[6] N. Kaina, M. Dupré, G. Lerosey, and M. Fink, "Shaping complex microwave fields in reverberating media with binary tunable metasurfaces," *Nature Scientific reports*, vol. 4, p. 6693, 2014.

[7] L. Zhang, X. Q. Chen, S. Liu, Q. Zhang, J. Zhao, J. Y. Dai, G. D. Bai, X. Wan, Q. Cheng, G. Castaldi *et al.*, "Space-time-coding digital metasurfaces," *Nature communications*, vol. 9, no. 1, p. 4334, 2018.

[8] M. M. Shanei, D. Fathi, F. Ghasemifard, and O. Quevedo-Teruel, "All-silicon reconfigurable metasurfaces for multifunction and tunable performance at optical frequencies based on glide symmetry," *Nature Scientific reports*, vol. 9, no. 1, pp. 1–11, 2019.

[9] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network: Joint active and passive beamforming design," in *Proc. IEEE GLOBECOM'18*, Dec. 2018, pp. 1–6.

[10] ——, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5394–5409, Nov. 2019.

[11] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2019.

[12] Q. Wu and R. Zhang, "Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts," *IEEE Trans. Commun.*, pp. 1–1, 2019.

[13] C. Huang, G. C. Alexandropoulos, C. Yuen, and M. Debbah, "Indoor signal focusing with deep learning designed reconfigurable intelligent surfaces," *CoRR*, vol. abs/1905.07726, 2019. [Online]. Available: http://arxiv.org/abs/1905.07726

[14] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling large intelligent surfaces with compressive sensing and deep learning," *CoRR*, vol. abs/1904.10136, 2019. [Online]. Available: http://arxiv.org/abs/1904.10136

[15] K. Feng, Q. Wang, X. Li, and C. Wen, "Deep reinforcement learning based intelligent reflecting surface optimization for MISO communication systems," *IEEE Wireless Commun. Lett.*, pp. 1–1, 2020.

[16] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," 2020.

[17] H. Yang, Z. Xiong, J. Zhao, D. Niyato, and L. Xiao, "Deep reinforcement learning based intelligent reflecting surface for secure wireless communications," *arXiv preprint arXiv:2002.12271*, 2020.

[18] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[19] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[21] K. Arulkumaran, M. Deisenroth, M. Brundage, and A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

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

[23] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1587–1596. [Online]. Available: https://proceedings.mlr.press/v80/fujimoto18a.html

[24] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.