Refined-Deep Reinforcement Learning for MIMO Bistatic Backscatter Resource Allocation

Shayan Zargari, Diluka Galappaththige, *Member, IEEE*, and Chintha Tellambura, *Fellow, IEEE*

***Abstract*—Bistatic backscatter communication facilitates ubiquitous, massive connectivity of passive tags for future Internet-of-Things (IoT) networks. The tags communicate with readers by reflecting carrier emitter (CE) signals. This work addresses the joint design of the transmit/receive beamformers at the CE/reader and the reflection coefficient of the tag. A throughput maximization problem is formulated to satisfy the tag requirements. A joint design is developed through a series of trial-and-error interactions within the environment, driven by a predefined reward system in a continuous state and action context. By leveraging recent advances in deep reinforcement learning (DRL), the underlying optimization problem is addressed. Capitalizing on deep deterministic policy gradient (DDPG) and soft actor-critic (SAC), we proposed two new algorithms, namely refined-DDPG for MIMO BiBC (RDMB) and refined-SAC for MIMO BiBC (RSMB). Simulation results show that the proposed algorithms can effectively learn from the environment and progressively improve their performance. They achieve results comparable to two leading benchmarks: alternating optimization (AO) and several DRL methods, including deep Q-network (DQN), double deep Q-network (DDQN), and dueling DQN (DuelDQN). For a system with twelve antennas, RSMB leads with a $26.76\,\%$ gain over DQN, followed by AO and RSMB at $23.02\,\%$ and $19.16\,\%$, respectively.**

***Index Terms*—Bistatic backscatter communication, deep reinforcement learning, resource allocation, multiple-input multiple-output, Internet of Things, 6G.**

I. Introduction

Bistatic backscatter communication (BiBC) networks, which may enable energy harvesting (EH)-based Internet-of-Things (IoT) networks, are of significant interest as the third-generation partnership project (3GPP) has initiated a new study item [1]. Applications of BiBC-enabled IoT networks include logistics, inventory management, warehousing, manufacturing, energy industry, healthcare, agriculture, aerospace and defense, farming, retail, sports, and many more [2]–[4].

A typical BiBC network consists of a dedicated carrier emitter (CE), a reader, and a backscatter device(s) or tag(s) [2]–[4]. The key idea is that tags are passive devices that reflect radio frequency (RF) signals from the CE. The reflections enable tags to communicate with the reader [5]. BiBC with a standalone RF source located separately from the reader offers lower round-trip Pathloss, predictability, reduced interference, system control, and knowledge of source signal parameters over other backscatter configurations[1]. Consequently, this study employs a BiBC system.

S. Zargari, D. Galappaththige, and C. Tellambura with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 1H9, Canada (e-mail: {zargari, diluka.lg, ct4}@ualberta.ca).

[1]Backscatter fundamentals and configurations, i.e., monostatic, bistatic, and ambient, can be found in [2]–[4], [6].

A. Related work

In BiBC systems, CE beamforming, tag reflection coefficients, and receive combiner at the reader must all be optimized to improve communication performance. The optimization is subject to the constraints on transmit power at the CE, minimum activation power targets for the tags, normalized combiner vectors, and unit-interval constraints for the reflection coefficients. Consequently, this is a non-convex and NP-hard problem.

Consequently, related optimization studies can be classified into two categories: (1) classical methods without using machine learning (ML) [7]–[10] and (2) ML-based methods [11]–[14] methods.

Category one often uses alternating optimization (AO) [15] as its bedrock. AO is particularly effective for non-convex problems that can be decomposed into smaller, more manageable subproblems. Each subproblem is typically easier to solve than the overall problem and may be suitable for convex methods, closed-form solutions, or other optimization techniques. Applied to the current problem, AO can be used to iteratively optimize CE beamforming, tag reflection coefficients, and reader combiner until convergence is achieved. For example, [7] uses successive convex approximation (SCA) to optimize CE beamforming, tag reflection coefficients, and reader combiners in a non-orthogonal multiple access (NOMA)-assisted BiBC system. References [8], [9] study an intelligent reflecting surface (IRS)-assisted BiBC system to minimize the CE transmit power. In [8], two algorithms, i.e., minorization-maximization and AO, are proposed for single-tag and multi-tag scenarios, respectively, utilizing successive refinement and semidefinite relaxation (SDR) approaches to improve CE beamforming, tag reflection coefficients, reader combiners, and IRS phase-shifts. In [9], an AO approach is presented for a single tag and a single-antenna reader system to construct CE transmit beamforming, tag reflection coefficient, and IRS phase shifts. Reference [10] offers an algorithm for designing beamforming at distributed CEs, tag reflection coefficients, and reader reception combiners utilizing AO, fractional programming (FP), and Rayleigh quotient techniques to maximize the tag sum rate while proposing a channel estimation protocol.

However, such methods may converge to a local rather than a global optimum, especially in this non-convex problem. The process can be slow if the number of variables is large or if each subproblem is still computationally intensive. They also struggle with network complexity [16]. These issues lead to Category two.

Deep reinforcement learning (DRL), a fusion of reinforcement learning (RL) and deep learning (DL) has emerged as a remedy [11]–[14]. Nevertheless, previous DRL works

[11]–[14] only examine ambient backscatter networks and not BiBC. Moreover, these studies focus on single antenna RF sources and single tag scenarios, except for [12]. Although [12] supports multiple tags, it is limited to a single-antenna source and single-antenna reader.

Reference [11] employs a double deep Q-network (DDQN) to allow the reader to learn the best policy based on primary channel utilization for coordinating the transmission of multiple ambient backscatter tags, including backscattering time, energy harvesting time, and active transmission time. In [12], a deep deterministic policy gradient (DDPG) algorithm is used to optimize the backscatter relaying policies, i.e., reflection coefficients, to maximize network throughput in a backscatter-assisted relaying network. Reference [13] presents two algorithms: (1) constellation learning with labeled signals and (2) constellation learning with labeled and unlabeled signals, using ML-based modulation-constrained expectation maximization algorithms. Study [14] develops a deep transfer learning (DTL) detection framework that includes offline learning, transfer learning, and online detection of tag signals. It implicitly extracts channel features and recovers tag symbols, eliminating the need for channel estimation.

DRL has an innate ability to intelligently and adaptively navigate complex optimization landscapes [17], [18]. The most recent advancements of DRL are DDPG and soft actor-critic (SAC). DDPG is a model-free off-policy algorithm that learns continuous actions and incorporates concepts from deterministic policy gradient (DPG) and deep Q-network (DQN). In particular, it employs experience replay and slow-learning target networks from DQN, and it is built on DPG, which can function in continuous action spaces [19] – Section VI. Conversely, the SAC extends the DDPG by employing a stochastic policy capable of expressing multi-modal optimum policies and entropy regularization based on the stochastic policy's entropy [20]. It acts as a built-in, state-dependent exploration heuristic for the agent rather than depending on non-correlated noise processes, as in DDPG. Additionally, It incorporates two soft Q-networks to address the overestimation bias issue in Q-network-based approaches – Section VII.

B. Motivation and contributions

No prior studies have explored ML techniques tailored explicitly for multiple-input multiple-output (MIMO) BiBC systems, involving a multi-antenna CE and a multi-antenna reader interacting with multiple tags (Fig. 1). Importantly, our advanced DRL paradigm is also adaptable to other configurations, i.e., ambient and monostatic, offering a generalized framework that builds upon previous research [11]–[14]. Therefore, this research represents a pioneering study introducing DRL-based resource allocation methods specifically designed to optimize MIMO BiBC networks.

With IoT networks expanding and 6G wireless technology emerging, there is a pressing need for optimized BiBC systems. Traditional methods falter in handling BiBC's complex and dynamic state and action spaces. Also, ensuring the quality of service (QoS) for tags while maximizing long-term throughput is a significant challenge. This optimization problem is non-convex due to multi-tag interference, and its optimal solution is not amenable to widely available convex optimization tools. To address this, we develop two algorithms based on cutting-edge DRL techniques, i.e., DDPG and SAC, to offer a practical solution without relying on complex mathematical models or numerical optimization methods. The key contributions of this paper can be summarized as follows:

- This paper represents the initial effort to develop a framework that integrates advanced DRL techniques, i.e., DDPG and SAC, into the optimal design of BiBC systems to tackle high-dimensional optimization challenges.
- The focus is on maximizing the tag sum rate while satisfying the EH requirements of the tags. The CE transmit beamforming, tag reflection coefficients, and reader reception combiners are optimized to achieve this.
- Capitalizing on DDPG and SAC, we proposed two new algorithms, namely refined-DDPG for MIMO BiBC (RDMB) and refined-SAC for MIMO BiBC (RSMB). Compared to the existing methods, our algorithms can intuitively explore and decipher the optimization problem through trial-and-error interactions with the environment. Also, they accommodate continuous state and action spaces, conferring substantial advantages over the conventional methods.
- As previously mentioned, the AO technique optimizes the transmit beamforming, reflection coefficient, and reception combiners separately, as shown in [7]–[10], [21], [22]. In contrast, the proposed RDMB and RSMB algorithms jointly optimize these variables while progressively maximizing the sum rate by observing the reward and iteratively adjusting the algorithm parameters. However, given the ubiquity of the AO approach, we implement an AO-based benchmark for extensive comparison.
- The RDMB and RSMB algorithms feature a standard formulation and low implementation complexity, requiring no explicit model of the wireless environment or specific mathematical formulations. This makes it easily scalable to various system settings. Unlike DL-based algorithms that depend on sample labels derived from mathematically formulated algorithms, these algorithms can independently learn about and adapt to the environment.
- Particularly noteworthy is the RSMB algorithm, which outperforms the RDMB algorithm in several aspects. RSMB, using SAC, introduces an entropy regularization term into the objective function, encouraging more exploratory policy and thereby offering improved robustness and stability compared to RDMB.
- The proposed algorithms are also compared against existing ML-based methods. For this purpose, we implement DQN, DDQN, and dueling DQN (DuelDQN) algorithms [17]. Comprehensive simulations indicate that our algorithms learn incrementally from the environment and improve performance. For example, for a twelve-antenna network, RSMB leads with a $26.76\,\%$ gain over DQN, followed by AO and RSMB at $23.02\,\%$ and $19.16\,\%$,

TABLE I: Table of Notations.

Notation	Description
a	Action
a_{NL}	Nonlinear EH parameter
α_k	Reflection coefficient of tag T_k
β	Learning rate
b_{NL}	Nonlinear EH parameter
D	Experience replay buffer capacity
δ_d	Direct signal interference cancellation quality
E	Number of episodes
η	Power conversion efficiency
f_c	Carrier frequency
$\mathbf{F}_0$	Channel matrix between CE and reader
$\mathbf{F}_k$	Effective backscatter channel through tag T_k
γ	Discount factor
γ_k	SINR of tag T_k
$\mathbf{g}_{b,k}$	Channel between tag T_k and reader
$\mathbf{g}_{f,k}$	Channel between CE and tag T_k
K	Number of single-antenna tags
L	Number of experiences in mini-batch
λ_a	Actor network training decay rate
λ_c	Critic network training decay rate
M	Number of antennas at the CE
M_{NL}	Maximum harvested power
μ_a	Actor network training learning rate
μ_c	Critic network training learning rate
N	Number of antennas at the reader
$\Phi(\cdot)$	Nonlinear EH function
π	Policy
p_b	Activation threshold
p'_b	Inverse of nonlinear EH function
p_k^{h}	Harvested power at tag T_k
P_{EH}	Harvested power
P_{th}	Threshold power
P_k^{in}	Incident power at tag T_k
P_s	Maximum transmit power
P_t	Total transmit power of the CE
Q_π	State-action value function
r	Reward
s	State
σ^2	Noise power
T	Number of steps in each episode
τ_a	Target actor network update learning rate
τ_c	Target critic network update learning rate
$\mathbf{u}_k$	Receive combiner vector for tag T_k
$\mathbf{w}$	Beamforming vector
ξ	Entropy regularization term
ζ	Path loss exponent

respectively.

Notations: For matrix $\mathbf{A}$, $\mathbf{A}^H$ and $\mathbf{A}^T$ represent the Hermitian conjugate transpose and transpose, respectively. $\mathbb{R}^{M\times N}$ represents $M \times N$ dimensional real matrices, and $\mathbb{C}^{M\times N}$ refers to $M \times N$ dimensional complex matrices. Euclidean norm, absolute value, and Frobenius norm are represented by $\|\cdot\|$, $|\cdot|$, and $\|\cdot\|_F$, respectively. The expectation operator is denoted by $\mathbb{E}[\cdot]$. A circularly symmetric complex Gaussian (CSCG) random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$ is represented as $\sim \mathcal{CM}(\boldsymbol{\mu}, \mathbf{C})$. The symbol $\mathbf{I}_M$ represents an identity matrix of dimensions $M\times M$. The operators $(\cdot)^{-1}$ and $\otimes$ denote the inversion of a matrix and the Kronecker product. $\mathcal{O}$ expresses the big-O notation. Finally, $\mathcal{K} \triangleq \{1,\ldots,K\}$, $\mathcal{K}_0 \triangleq \{0,1,\ldots,K\}$, and $\mathcal{K}_k \triangleq \mathcal{K}\setminus\{k\}$.

II. BiBC System Model

Fig.1 depicts the MIMO BiBC system consisting of an M-element uniform linear array (ULA) antenna CE, an N-element ULA antenna reader, and K single-antenna tags (T_k denotes the k-th tag for $k \in \mathcal{K}$). The ULA antennas at the CE and the reader are spaced at half-wavelength intervals [23]. The tags employ a power-splitting (PS) protocol for EH and data backscattering; see SectionII-B. This protocol ensures that the tags use part of the received CE signal for EH and reflect the remaining part to transmit data to the reader. The reader recovers the tags' data upon receiving both direct and reflected signals, although the strong direct link interference from the CE affects this process.

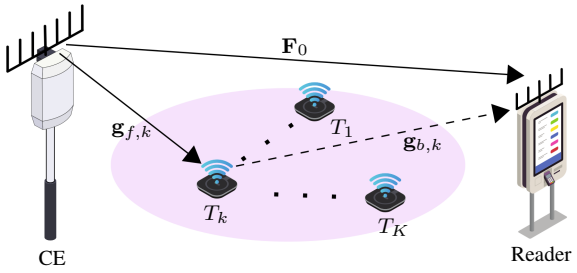

Fig. 1: A generalized BiBC system.

A. Channel Model

Backscatter channels differ from conventional one-way communication channels [2], [3], [24]. The reason is that a backscatter channel is the cascade of the CE-to-tag and tag-to-reader channels. Consequently, it experiences double-Pathloss, leading to severe deep fades. To model small-scale fading, component channels are modeled as zero-mean CSCG random variables. This amounts to the Rayleigh block flat fading model with a predefined coherence time [24], [25].

During each fading block, $\mathbf{F}_0 \in \mathbb{C}^{M\times N}$, $\mathbf{g}_{f,k} \in \mathbb{C}^{M\times 1}$, and $\mathbf{g}_{b,k} \in \mathbb{C}^{N\times 1}$, respectively, represent the channels between the CE and the reader, the CE and T_k, and T_k and the reader. These channels can be represented as $\mathbf{F}_0 = \tilde{\mathbf{F}}_0 \boldsymbol{\zeta}_{F_0}^{1/2}$ and $\mathbf{a} = \zeta_a^{1/2}\tilde{\mathbf{a}}$, where $\mathbf{a} \in \{\mathbf{g}_{f,k}, \mathbf{g}_{b,k}\}$. Here, $\boldsymbol{\zeta}_{F_0}$ (an $N\times N$ diagonal matrix) and ζ_a capture the large-scale Pathloss and shadowing, which stays constant for several coherence intervals. Moreover, $\tilde{\mathbf{F}}_0 \sim \mathcal{CN}(\mathbf{0}_{M\times N}, \mathbf{I}_M \otimes \mathbf{I}_N)$ and $\tilde{\mathbf{a}} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_A)$ account for the small-scale Rayleigh fading, where $A \in \{M, N\}$.

It is assumed that the perfect channel state information (CSI) is available. A central controller, or agent (which is usually associated with the CE), can collect CSI and implement advanced DRL algorithms. The algorithm processes this state information to determine efficient actions, such as transmit beamforming, reflection coefficients, and receive combiners. The CSI can be estimated through emerging techniques such as those detailed in [26]–[28]. These methods include pilot-based, blind, and semi-blind approaches, utilizing algorithms like least squares, minimum mean squared error (MMSE) estimator, expectation maximization, and eigenvalue decomposition for high-precision channel estimation. However, the objective is to determine the optimal transmit beamforming, reflection coefficients, and receiver combiners by maximizing the sum rate, and leveraging recent advancements in DRL techniques for a given CSI.

B. Tag's EH Model

Because passive tags do not generate RF signals, their power consumption is extremely low, allowing them to function without batteries and rely on EH. They reflect (i.e., backscatter) external RF signals to transmit their data. PS of the incident RF signal is a critical operation at the tag [29], [30]. This operation allows the tag to send data and harvest energy simultaneously.

Let the incident RF power at T_k be P_k^{in}. If α_k is its reflection coefficient, Tag T_k uses the PS operation as follows [2], [3]:

1) It reflects $\alpha_k P_k^{\text{in}}$ for data transmission,
2) It uses the reminder, i.e., $\eta(1-\alpha_k)P_k^{\text{in}}$ for EH.

The harvested power, p_k^{h}, can be modeled as a linear or nonlinear function of P_k^{in}. The linear model predicts harvested power at T_k as $p_k^{\text{h}} = \eta(1-\alpha_k)P_k^{\text{in}}$, where $\eta \in (0,1]$ represents the power conversion efficiency. While the linear model is the most often utilized in the literature due to its simplicity, it disregards the nonlinear properties of actual EH circuits, such as saturation and sensitivity [31].

As a result, the parametric nonlinear sigmoid EH model is frequently employed [32]. The harvested power at T_k is calculated as $p_k^{\text{h}} = \Phi((1-\alpha_k)P_k^{\text{in}})$, where $\Phi(\cdot)$ is a function expressing nonlinear effects [32, eq. (4)].

Despite the choice of linear or nonlinear model, the activation threshold, i.e., p_b, is a critical parameter. It is the minimal power required to wake up the EH circuit and is typically $-20\,\text{dBm}$ for commercial passive tags [2]. To activate the tag, the harvested power must surpass the threshold, i.e., $p_k^{\text{h}} \geq p_b$. In particular, $(1-\alpha_k)P_k^{\text{in}} \geq p_b'$, where $p_b' \triangleq \Phi^{-1}(p_b)$ and $\Phi^{-1}(p_b) = b_{\text{NL}} - \frac{1}{a_{\text{NL}}}\ln\left(\frac{M_{\text{NL}}-p_b}{p_b}\right)$. Parameters a_{NL} and b_{NL} model the circuit characteristics like capacitance and resistance. M_{NL} denotes the maximum harvested power when the EH circuit is saturated. Parameters a_{NL}, b_{NL}, and M_{NL} can be derived using a curve fitting tool [32]. Without loss of generality, the nonlinear EH model is adopted for the rest of this paper.

C. Transmission Model

As mentioned before, the tags reflect the RF signal from the CE to send data to the reader. Thus, the CE transmits the signal $\mathbf{x} \in \mathbb{C}^{M\times 1}$ which is given by

$$\mathbf{x} = \mathbf{w}s, \tag{1}$$

where $\mathbf{w} \in \mathbb{C}^{M\times 1}$ is the CE beamforming vector and $s = e^{-j2\pi f_c t}$ is a complex passband equivalent signal with a carrier frequency of f_c and unit power, i.e., $\mathbb{E}\{|s|^2\} = 1$ [33].

Assuming the propagation delay differences for all signals are negligible [34], the reader-received signal is given by

$$\mathbf{y} = \underbrace{\mathbf{F}_0^{\text{H}}\mathbf{w}s}_{\text{Direct signal interference}} + \underbrace{\sum\nolimits_{k\in\mathcal{K}} \sqrt{\alpha_k}\mathbf{F}_k^{\text{H}}\mathbf{w}sc_k}_{\text{Tag signals}} + \underbrace{\mathbf{z}}_{\text{AWGN}}, \tag{2}$$

where $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_N)$ is the additive white Gaussian noise (AWGN) at the reader, $\mathbf{F}_k = \mathbf{g}_{f,k}\mathbf{g}_{b,k}^{\text{H}}$ is the effective backscatter channel through T_k, c_k is T_k's data with $\mathbb{E}\{|c_k|^2\} = 1$, and α_k is the reflection coefficient of T_k.

Remark 1: In BiBC, the reader eliminates the direct link signal from the CE by using a form of successive interference cancellation (SIC) before the tags' data decoding. Specifically, the direct signal in (2) does not carry information bits. Therefore, direct signal interference can be eliminated by estimating and subtracting the mean value of the received signal, $\mathbb{E}\{\mathbf{y}\}$ [33], [35].

The reader then applies the receive combiner, $\mathbf{u}_k \in \mathbb{C}^{N\times 1}$, to the received signal (2) to capture the desired signal of T_k for data decoding. The post-processed signal for decoding T_k's data is thus given as

$$y_k = \underbrace{\sqrt{\delta_d}\mathbf{u}_k^{\text{H}}\mathbf{F}_0^{\text{H}}\mathbf{w}s}_{\text{Direct interference}} + \underbrace{\sqrt{\alpha_k}\mathbf{u}_k^{\text{H}}\mathbf{F}_k^{\text{H}}\mathbf{w}sc_k}_{\text{Desired signal}} + \underbrace{\sum\nolimits_{i\in\mathcal{K}_k}\sqrt{\alpha_i}\mathbf{u}_k^{\text{H}}\mathbf{F}_i^{\text{H}}\mathbf{w}sc_i}_{\text{Muti-tag interference}} + \underbrace{\mathbf{u}_k^{\text{H}}\mathbf{z}}_{\text{AWGN}}, \tag{3}$$

where $\delta_d \in [0,1]$ accounts for the degree of imperfection in the direct signal interference cancellation process.

D. Communication Rates

The reader decodes c_k, considering the other tag signals as interference. From (3), T_k's signal-to-interference-plus-noise ratio (SINR) at the reader can be given as

$$\gamma_k = \frac{\alpha_k|\mathbf{u}_k^{\text{H}}\mathbf{F}_k^{\text{H}}\mathbf{w}|^2}{\delta_d|\mathbf{u}_k^{\text{H}}\mathbf{F}_0^{\text{H}}\mathbf{w}|^2 + \sum_{i\in\mathcal{K}_k}\alpha_i|\mathbf{u}_k^{\text{H}}\mathbf{F}_i^{\text{H}}\mathbf{w}|^2 + \sigma^2\|\mathbf{u}_k\|^2}. \tag{4}$$

Thus, T_k's rate at the reader is approximated as [25]:

$$\mathcal{R}_k \approx \log_2(1+\gamma_k) \quad [\text{bps/Hz}], \tag{5}$$

Here, while the reader can partially cancel interference from the direct-link signal, represented as $\mathbf{F}_0^{\text{H}}\mathbf{w}$, it cannot eliminate interference from tag reflections as they contain unknown tag data, c_k for $k \in \mathcal{K}$.

E. Incident RF Power at the tags

Per Section II-B, the input power at the tags must exceed the activation threshold, typically $-20\,\text{dBm}$ for commercial passive tags [2]. By considering the RF signal from the CE, T_k's input power may be expressed as

$$P_k^{\text{in}} = |\mathbf{g}_{f,k}^{\text{H}}\mathbf{w}|^2. \tag{6}$$

Ensuring that P_k^{in} exceeds the activation threshold is critical for the reliable operation of the tags [2]–[4]. If the input power is below the activation threshold (see section II-B), the tags may fail to activate, leading to a breakdown in data transfer processes.

III. Problem Formulation

The problem is to optimize the BiBC performance in terms of the sum rate. In particular, the objective is to maximize the tag sum rate by jointly optimizing the reader reception combiners, $\{\mathbf{u}_k\}_{k\in\mathcal{K}}$, the CE transmit beamforming, $\mathbf{w}$, and the tag reflection coefficients, $\{\alpha_k\}_{k\in\mathcal{K}}$ for given a particular CSI. Denote the set of these optimization variables as $\mathcal{A} = \{\{\mathbf{u}_k\}_{k\in\mathcal{K}}, \mathbf{w}, \{\alpha_k\}_{k\in\mathcal{K}}\}$. The optimization problem is thus formulated as follows:

$$\mathbf{P}_1 : \max_{\mathcal{A}} \; R_{\text{sum}} = \sum\nolimits_{k\in\mathcal{K}} \mathcal{R}_k, \tag{7a}$$

$$\text{s.t } P_k^{\text{in}} \geq P_{\text{th}}, \ \forall k, \tag{7b}$$
$$P_t \leq P_s, \tag{7c}$$
$$\|\mathbf{u}_k\|^2 = 1, \ \forall k, \tag{7d}$$
$$0 < \alpha_k < 1, \ \forall k, \tag{7e}$$

where $P_{\text{th}} = \Phi^{-1}(p_b)$. In addition, (7b) guarantees the minimum power requirements at the tags for EH, (7d) is the normalization constraint for the reception filter at the reader, and (7c) limits the total transmit power of the CE with P_s maximum allowable transmit power and $P_t = \|\mathbf{w}\|^2$. Finally, (7e) is the reflection coefficient constraint at each tag.

Solving $\mathbf{P}_1$ analytically is intractable, as the problem is non-convex because the objective function in $\mathbf{P}_1$ and constraint (7b) both contain entangled terms involving the product of optimization variables. Although some AO-based approximation methods have been proposed to find suboptimal solutions (e.g., [7]–[10]), they may not work in multi-antenna multi-tag scenarios. This paper thus proposes two frameworks by leveraging recent advancements in DRL techniques, i.e., DDPG and SAC, rather than attempting to solve this highly complex optimization problem through conventional methods. Unlike traditional deep neural networks (DNNs) that require both an offline training phase and an online learning phase, our algorithms utilize each CSI to construct the state and continuously run to obtain $\mathcal{A}$ [17].

Before proceeding to the proposed algorithms, it is helpful to briefly discuss the conventional optimization algorithms for $\mathbf{P}_1$, which serve as a benchmark for our solutions.

IV. Conventional Optimization solutions

$\mathbf{P}_1$ is non-convex due to variable products in the objective and constraints. Traditional methods like gradient descent (GD) can be used to solve $\mathbf{P}_1$ [36]. However, these solutions can be ineffective as they may get stuck in local optima and fail to provide a globally optimum solution. Thus, we resort to the AO technique, which involves splitting $\mathbf{P}_1$ into three simpler convex sub-problems: transmit beamforming, receive combiner, and tags' reflection coefficient optimization [37], [38]. This approach iteratively optimizes a variable or set of variables while keeping the rest of the variables fixed until convergence is achieved [15], [36]. We further assume the availability of perfect CSI.

Sub-problem 1: Reception combiner optimization

Here, we design the reader receive combiners, $\{\mathbf{u}_k\}_{k\in\mathcal{K}}$ for fixed $\{\mathbf{w}, \{\alpha_k\}_{k\in\mathcal{K}}\}$. Since the corresponding SINR of each tag in (7a) observed at the reader depends on its associate combiner vector, we maximize the sum rate by optimizing the SINR of each tag individually [30]. By defining $\mathbf{h}_i = \mathbf{F}_i\mathbf{w}$ for $i \in \mathcal{K}_0$, we can reformulate $\mathbf{P}_1$ into the following problem:

$$\mathbf{P}_{\text{u}} : \max_{\mathbf{u}_k} \frac{\mathbf{u}_k^{\text{H}}\tilde{\mathbf{h}}_k\tilde{\mathbf{h}}_k^{\text{H}}\mathbf{u}_k}{\mathbf{u}_k^{\text{H}}\mathbf{Q}_k\mathbf{u}_k}, \quad \text{s.t} \quad \|\mathbf{u}_k\|^2 = 1, \ \forall k, \tag{8}$$

where $\tilde{\mathbf{h}}_k = \sqrt{\alpha_k}\mathbf{h}_k$ and $\mathbf{Q}_k = \delta_d\mathbf{h}_0\mathbf{h}_0^{\text{H}} + \sum_{i\in\mathcal{K}_k}\alpha_i\mathbf{h}_i\mathbf{h}_i^{\text{H}} + \sigma^2\mathbf{I}_N$. $\mathbf{P}_{\text{u}}$ thus becomes a generalized Rayleigh ratio problem [39], [40] and the optimal received beamforming is given as [39, *Lemma 4.11*]

$$\mathbf{u}_k^* = \frac{\mathbf{Q}_k^{-1}\tilde{\mathbf{h}}_k}{\|\mathbf{Q}_k^{-1}\tilde{\mathbf{h}}_k\|}, \ \forall k, \tag{9}$$

which is a MMSE filter [39], [40].

Sub-problem 2: Transmit beamforming optimization

This sub-problem optimizes $\mathbf{w}$ for given $\{\{\mathbf{u}_k\}_{k\in\mathcal{K}}, \{\alpha_k\}_{k\in\mathcal{K}}\}$. This is a non-convex problem as the objective function in $\mathbf{P}_1$ is a fractional function of $\mathbf{w}$, and constraint (7b) is non-convex. Accordingly, we leverage the SCA and SDR methods to solve it. SDR is widely used for beamforming optimization, enabling powerful semidefinite programming (SDP) solvers to handle large-scale problems efficiently [8], [23], [36]. We define the matrix $\mathbf{W} \triangleq \mathbf{w}\mathbf{w}^{\text{H}}$ with $\mathbf{W} \succeq 0$ and $\text{Rank}(\mathbf{W}) = 1$. By doing so, the quadratic ratio function inside the $\log(\cdot)$ in the objective function is turned into its equivalent linear ratio over $\mathbf{W}$ which ends up with a concave objective function [30]. By dropping rank one constraint, the relaxed convex optimization problem $\mathbf{P}_1$ is given as

$$\mathbf{P}_{\text{w}} : \max_{\mathbf{w}} \ \sum\nolimits_{k\in\mathcal{K}} \log_2(A_k(\mathbf{W})) - B_k(\mathbf{W}, \mathbf{W}^{(t)}), \tag{10a}$$
$$\text{s.t } P_{\text{th}} - (1-\alpha_k)\text{Tr}(\mathbf{g}_{f,k}\mathbf{g}_{f,k}^{\text{H}}\mathbf{W}) \leq 0, \ \forall k, \tag{10b}$$
$$\text{Tr}(\mathbf{W}) \leq P_s, \tag{10c}$$
$$\mathbf{W} \succeq 0, \tag{10d}$$

where $\log_2(A_k(\mathbf{W}))$ and $B_k(\mathbf{W}, \mathbf{W}^{(t)})$ are defined in (11) with $\mathbf{g}_i = \mathbf{F}_i\mathbf{u}_k$ for $i \in \mathcal{K}_0$. Here, per SCA, $B_k(\mathbf{W}, \mathbf{W}^{(t)})$ is linearized using first-order Taylor series approximation near a feasible point $\mathbf{W}^{(t)}$, where $\mathbf{W}^{(t)}$ denotes the previous iteration values of $\mathbf{W}$ [30]. Finally, $\mathbf{P}_{\text{w}}$ is a conventional SDP problem amenable to the CVX tool [36].

Sub-problem 3: Reflection coefficient optimization

This involves optimizing $\{\alpha_k\}_{k\in\mathcal{K}}$ for fixed $\{\mathbf{w}, \{\mathbf{u}_k\}_{k\in\mathcal{K}}\}$. It is non-convex due to the multiple-ratio fractional objective function. To tackle this, we reformulate it by decoupling the objective's numerator and denominator. This FP approach is motivated by Dinkelbach's transform and is referred to as the quadratic transform because it involves quadratic terms [41]. It simplifies the optimization problem by converting a non-linear fractional objective function into a more tractable form and guarantees convergence to a global optimum [41].

The resultant problem is given as

$$\mathbf{P}_{\alpha} : \max_{\alpha_k} \ \sum\nolimits_{k\in\mathcal{K}} \log_2\left(1 + 2\lambda_k\sqrt{V_k} - \lambda_k^2 S_k\right), \tag{12a}$$
$$\text{s.t } (1-\alpha_k)|\mathbf{g}_{f,k}^{\text{H}}\mathbf{w}|^2 \geq P_{\text{th}}, \ \forall k, \tag{12b}$$
$$0 < \alpha_k < 1, \ \forall k, \tag{12c}$$

where V_k and S_k are the numerator and denominator of T_k's SINR as functions of α_k. Besides, $\{\lambda_k\}_{k\in\mathcal{K}}$ is the auxiliary variable introduced by the FP techniques with the optimal value of $\lambda_k^* = \sqrt{V_k}/S_k$ for $k \in \mathcal{K}$ [41]. For given $\{\lambda_k\}_{k\in\mathcal{K}}$, $\mathbf{P}_{\alpha}$ is convex and can be solved via CVX Matlab [36], [41].

$$A_k(\mathbf{W}) = \delta_d \mathrm{Tr}(\mathbf{g}_0\mathbf{g}_0^{\mathrm{H}}\mathbf{W}) + \sum_{i\in\mathcal{K}} \alpha_i \mathrm{Tr}(\mathbf{g}_i\mathbf{g}_i^{\mathrm{H}}\mathbf{W}) + \sigma^2\|\mathbf{u}_k\|^2,$$

$$B_k(\mathbf{W}, \mathbf{W}^{(t)}) = \log_2\left(\delta_d \mathrm{Tr}(\mathbf{g}_0\mathbf{g}_0^{\mathrm{H}}\mathbf{W}^{(t)}) + \sum_{i\in\mathcal{K}_k} \alpha_i \mathrm{Tr}(\mathbf{g}_i\mathbf{g}_i^{\mathrm{H}}\mathbf{W}^{(t)}) + \sigma^2\|\mathbf{u}_k\|^2\right) + \mathrm{Tr}\left(\frac{\delta_d\mathbf{g}_0\mathbf{g}_0^{\mathrm{H}} + \sum_{i\in\mathcal{K}_k}\alpha_i\mathbf{g}_i\mathbf{g}_i^{\mathrm{H}}}{\delta_d \mathrm{Tr}(\mathbf{g}_0\mathbf{g}_0^{\mathrm{H}}\mathbf{W}^{(t)}) + \sum_{i\in\mathcal{K}_k} \alpha_i \mathrm{Tr}(\mathbf{g}_i\mathbf{g}_i^{\mathrm{H}}\mathbf{W}^{(t)}) + \sigma^2\|\mathbf{u}_k\|^2}(\mathbf{W} - \mathbf{W}^{(t)})\right) \quad (11)$$

Algorithm 1 AO Algorithm

1: **Input**: Set the iteration counter $t = 0$, the convergence tolerance $\epsilon > 0$, initial feasible solution $\{\mathbf{w}, \{\alpha_k\}_{k\in\mathcal{K}}\}$. Initialize the objective function value $F^{(0)} = 0$.
2: **while** $\frac{F^{(t+1)} - F^{(t)}}{F^{(t+1)}} \geq \epsilon$ **do**
3: Solve (9) for optimal receive combiner, $\mathbf{u}_k^{(t+1)}$.
4: Solve $\mathbf{P}_{\mathrm{w}}$ to obtain suboptimal transmit beamforming, $\mathbf{w}^{(t+1)}$, by applying Gaussian randomization to recover the rank-one solution.
5: Solve $\mathbf{P}_\alpha$ to obtain the suboptimal power reflection coefficients, $\alpha_k^{(t+1)}$.
6: Calculate the objective function value $F^{(t+1)}$.
7: Set $t \leftarrow t + 1$;
8: **end while**
9: **Output**: Optimal solutions $\mathcal{A}^*$.

Algorithm 1 presents the overall steps to solve $\mathbf{P}_1$ using AO. It begins by initiating random feasible solutions for $\mathbf{w}$ and $\{\alpha_k\}_{k\in\mathcal{K}}$ and refines $\{\{\mathbf{u}_k\}_{k\in\mathcal{K}}, \mathbf{w}, \{\alpha_k\}_{k\in\mathcal{K}}\}$ in each iteration until the normalized improvement in objective is less than $\epsilon = 10^{-3}$.

V. Configuring the DRL Environment for BiBC

In DRL, the agent continually interacts with the environment, executing actions and receiving immediate rewards while observing changes in the environment's state. This helps the agent to identify the best action strategy (Fig. 2). We configure the DRL environment here according to the MIMO BiBC system. This is the foundation for the proposed algorithms [42].

1) **Action space**: This represents an array of possible decisions. At a specific time step t, the agent takes an action $\mathbf{a}^{(t)} \in \mathcal{A}$ based on a policy π. This action results in a transition in the environment's state, moving from the present state $\mathbf{s}^{(t)}$ to the next state $\mathbf{s}^{(t+1)}$. The action space of $\mathbf{P}_1$ is given by

$$\mathcal{A} = \{\mathbf{w}, \alpha_k\}, \quad (13)$$

where the cardinal number of the action space is given by $D_a = 2M + K$ and $\mathbf{w} = \mathrm{Re}\{\mathbf{w}\} + \mathrm{Im}\{\mathbf{w}\}$ is separated into its real and imaginary components. It is worth mentioning that receive combiner, $\{\mathbf{u}_k\}_{k\in\mathcal{K}}$, is not in the action space as it has a closed-form solution based on (9).

2) **State space:** This is a set of observations ($\mathbf{s}^{(t)} \in S$) that describe the environment. To construct the state space for $\mathbf{P}_1$, we integrate relevant BiBC environmental information. Thus, the state space is established based on the actions/reward from $(t-1)$-th time step and all channel links and given by

$$\mathbf{s}^{(t)} = \Big\{\mathbf{a}, r, P_t, P_k^{\mathrm{in}}, \mathrm{Re}\{\mathbf{F}_0\}, \mathrm{Re}\{\mathbf{F}_k\}, \mathrm{Re}\{\mathbf{g}_{f,k}\}, \mathrm{Re}\{\mathbf{g}_{b,k}\}, \mathrm{Im}\{\mathbf{F}_0\}, \mathrm{Im}\{\mathbf{F}_k\}, \mathrm{Im}\{\mathbf{g}_{f,k}\}, \mathrm{Im}\{\mathbf{g}_{b,k}\}\Big\}, \quad (14)$$

where the state space dimension is represented by $D_s = 2MNK + 2MN + 4MK + 2M + 2K + 2$. The states in (14) offer a comprehensive view of the BiBC's dynamics and interactions. The action set ($\mathbf{a}$) and reward function (r) enable efficient decision-making based on the current state while transmit power (P_t) and channel links are included to reflect the BiBC environment's intricacies.

3) **Reward function**: When an action is performed, a reward is given to the agent to evaluate how effectively the goal is achieved. The reward function takes into account both the objective function (R_{sum}) and the constraints of $\mathbf{P}_1$ by incorporating penalty terms, Ω_{Pow} and Ω_{EH}. The reward function is given by

$$r^{(t)} = R_{\text{sum}} - \Omega_{\text{Pow}} - \Omega_{\text{EH}}, \quad (15)$$

where

$$\Omega_{\text{Pow}} = \begin{cases} 0 & \text{if } \|\mathbf{w}\|^2 \leq P_s, \\ 1 & \text{otherwise.} \end{cases} \quad (16)$$

$$\Omega_{\text{EH}} = \sum\nolimits_{k\in\mathcal{K}} \max\left(P_k^{\mathrm{in}} - \Phi^{-1}(p_b), 0\right). \quad (17)$$

Specifically, Ω_{Pow} and Ω_{EH} ensure the transmit power and the tag's harvested power remain within the required limits.

4) **Policy:** The policy $\pi(\mathbf{s}^{(t)}, \mathbf{a}^{(t)})$ represents the probability of taking action $\mathbf{a}^{(t)}$ given state $\mathbf{s}^{(t)}$, ensuring that $\sum_{\mathbf{a}^{(t)}\in A} \pi(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) = 1$.
5) **State-action value function:** This represents the value of being in a particular state s and performing an action a, defined as $Q_\pi(\mathbf{s}^{(t)}, \mathbf{a}^{(t)})$. While the reward measures the immediate return from action a in state s, the value function estimates the potential future rewards for action a in state s.
6) **Experience replay buffer:** This stores the agent's experiences at each time step, represented as $(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}, r^{(t+1)}, \mathbf{s}^{(t+1)})$.

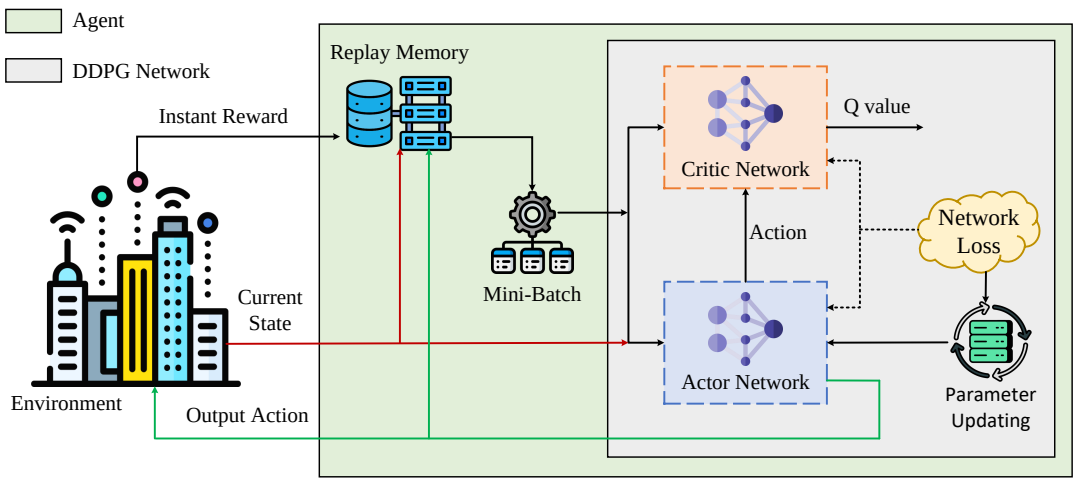

Fig. 2: An illustration of deep Q learning, where a double DNN approximates the optimal state-action value and Q-value function.

For given $(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}, r^{(t)})$ at time step t, the Q-value function is given by [42]:

$$Q_\pi(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) = \mathbb{E}_\pi \left[R^{(t)} \mid \mathbf{s}^{(t)} = \mathbf{s}, \mathbf{a}^{(t)} = \mathbf{a} \right], \tag{18}$$

where $R^{(t)} = \sum_{\tau=0}^{\infty} \gamma^\tau r(t+\tau+1)$ and $\gamma \in (0, 1]$ denotes the discount rate. Specifically, the Q-value function assesses how choosing an action $\mathbf{a}^{(t)}$ impacts the expected future reward under policy impacts the expected future reward under policy π. Based on the Bellman equation, the Q-value function can be represented as [42]:

$$\begin{aligned} Q_\pi(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) = \mathbb{E}_\pi \left[r^{(t+1)} \mid \mathbf{s}^{(t)} = \mathbf{s}, \mathbf{a}^{(t)} = \mathbf{a} \right] \\ + \gamma \sum\nolimits_{\mathbf{s}' \in S} P^{a}_{\mathbf{s}\mathbf{s}'} \sum\nolimits_{\mathbf{a}' \in A} \pi(\mathbf{s}', \mathbf{a}') Q_\pi(\mathbf{s}', \mathbf{a}'), \end{aligned} \tag{19}$$

where $P^{\mathbf{a}}_{\mathbf{s}\mathbf{s}'} = \Pr(\mathbf{s}^{(t+1)} = \mathbf{s}' \mid \mathbf{s}^{(t)} = \mathbf{s}, \mathbf{a}^{(t)} = \mathbf{a})$ indicates the transition probability from state s to state $\mathbf{s}'$ when action a is implemented [42]. The Q-learning algorithm aims to find the optimal policy π^*. Based on (19), the optimal Q-value function associated with the optimal policy can be found as

$$\begin{aligned} Q^*(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) = r^{(t+1)}(\mathbf{s}^{(t)} = \mathbf{s}, \mathbf{a}^{(t)}, \pi = \pi^*) \\ + \gamma \sum\nolimits_{\mathbf{s}' \in S} P^{\mathbf{a}}_{\mathbf{s}\mathbf{s}'} \max_{\mathbf{a}' \in A} Q^*(\mathbf{s}', \mathbf{a}'). \end{aligned} \tag{20}$$

By recursively solving Bellman equation (20), one can determine the optimal $Q^*(\mathbf{s}', \mathbf{a}')$ without needing precise information about the reward model or state transition model [42]. Thus, the updated Q-value function can be expressed as

$$\begin{aligned} Q^*(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) \leftarrow (1-\beta) Q^*(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) \\ + \beta \left(r^{(t+1)} + \gamma \max_{\mathbf{a}'} Q_\pi(\mathbf{s}^{(t+1)}, \mathbf{a}') \right), \end{aligned} \tag{21}$$

where β denotes the learning rate for updating the Q-value function.

Continuously updating $Q(\mathbf{s}^{(t)}, \mathbf{a}^{(t)})$ drives convergence towards the optimal $Q^*(\mathbf{s}^{(t)}, \mathbf{a}^{(t)})$ [43]. Achieving this with vast state and action spaces poses significant challenges. Function approximation techniques are often utilized to address these challenges, such as feature representation, DNNs, and methods that link value functions to state variables.

The DNN employs non-linear functions to approximate the state/action-value functions and policy. Both the Q-value function and action are approximated by the DNN. However, DNN-based approximations lack interpretability, potentially leading DRL algorithms to attain only local optimality due to temporal correlation among states. Experience replay buffer substantially improves DRL performance by updating the DNN from a batch of randomly sampled states in the replay memory, rather than solely updating from the last state.

The Q-value function in DRL is updated as follows:

$$Q(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) = Q(\boldsymbol{\theta} \mid \mathbf{s}^{(t)}, \mathbf{a}^{(t)}). \tag{22}$$

where $\boldsymbol{\theta}$ denotes the weight and bias parameters within the DNN. Instead of directly updating the Q-value function as in (21), the optimal Q-value function is approximated by updating $\boldsymbol{\theta}$ through stochastic optimization algorithms, represented as

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \mu \Delta_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}), \tag{23}$$

where μ indicates the learning rate and $\Delta_{\boldsymbol{\theta}}$ is the gradient of the loss function $\mathcal{L}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

The loss function measures the difference between the predicted value from the DNN and the actual target value. Although DRL involves approximating the optimal Q-value function, the true target value remains unknown. Two DNNs with identical architecture are designed to address this: the training DNN and the target DNN. Their value functions are respectively given by $Q(\boldsymbol{\theta}^{\text{train}} \mid \mathbf{s}^{(t)}, \mathbf{a}^{(t)})$ and $Q(\boldsymbol{\theta}^{\text{target}} \mid \mathbf{s}^{(t)}, \mathbf{a}^{(t)})$. The actual target value is estimated as follows:

$$y^{\text{target}} = r^{(t+1)} + \gamma \max_{\mathbf{a}'} Q(\boldsymbol{\theta}^{\text{target}} \mid \mathbf{s}^{(t+1)}, \mathbf{a}'). \tag{24}$$

The loss function is thus defined as

$$\mathcal{L}(\boldsymbol{\theta}) = \left(y^{\text{target}} - Q(\boldsymbol{\theta}^{\text{train}} \mid \mathbf{s}^{(t)}, \mathbf{a}^{(t)}) \right)^2. \tag{25}$$

VI. Refined-DDPG for MIMO BiBC

The proposed joint design of transmit beamforming and reflection coefficient faces a fundamental challenge. That optimization problem has both continuous state and action spaces. We utilize the DDPG network to meet it, as depicted in

Fig. 2. The DDPG consists of two DNNs: (i) the actor network and (ii) the critic network [20], [44]. The actor network uses the state to produce a continuous action, which is then input into the critic network along with the state. This approach approximates the action, eliminating the need to find the action that optimizes the Q-value function for the next state. The update rule for the critic network training is given by

$$\boldsymbol{\theta}_c^{(t+1)} \leftarrow \boldsymbol{\theta}_c^{(t)} - \mu_c \Delta_{\boldsymbol{\theta}_c^{\text{train}}} \mathcal{L}(\boldsymbol{\theta}_c^{\text{train}}), \tag{26}$$

where μ_c indicates the learning rate for the training critic network. The gradient with respect to the training critic network $\boldsymbol{\theta}_c^{\text{train}}$ is represented by $\Delta_{\boldsymbol{\theta}_c^{\text{train}}} \mathcal{L}(\boldsymbol{\theta}_c^{\text{train}})$. In addition, the loss function is defined by the mean squared error (MSE) as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{L} \left\| \mathbf{r}^{(t)} + \gamma q(\boldsymbol{\theta}_c^{\text{target}} \mid \mathbf{s}^{(t+1)}, \mathbf{a}') - q(\boldsymbol{\theta}_c^{\text{train}} \mid \mathbf{s}^{(t)}, \mathbf{a}^{(t)}) \right\|_2^2, \tag{27}$$

where $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')_{i=1}^{L}$ is the mini-batch of transitions sampled from the experience replay buffer with a size of L and $\mathbf{a}'$ denotes the action output from the target actor network. The target and training critic network parameters are symbolized by $\boldsymbol{\theta}_c^{\text{target}}$ and $\boldsymbol{\theta}_c^{\text{train}}$, with the target network parameters being updated like those of the training network at specific time intervals. The updating of the target network is much slower than that of the training network [20], [44]. The update rule for the actor network training is given by

$$\boldsymbol{\theta}_a^{(t+1)} \leftarrow \boldsymbol{\theta}_a^{(t)} - \mu_a \Delta_{\mathbf{a}} q(\boldsymbol{\theta}_c^{\text{target}} \mid \mathbf{s}^{(t)}, \mathbf{a}) \Delta_{\boldsymbol{\theta}_a^{\text{train}}} \pi(\boldsymbol{\theta}_a^{\text{train}} \mid \mathbf{s}^{(t)}), \tag{28}$$

where μ_a is the learning rate for adjusting the training actor network. $\pi(\boldsymbol{\theta}_a^{\text{train}} \mid \mathbf{s}^{(t)})$ is the training actor network with $\boldsymbol{\theta}_a^{\text{train}}$ as the DNN parameters and $\mathbf{s}^{(t)}$ as the given state input. The gradient of the target critic network over the action is represented by $\Delta_a q(\boldsymbol{\theta}_c^{\text{target}} \mid \mathbf{s}^{(t)}, \mathbf{a})$, whereas $\Delta \boldsymbol{\theta}_a^{\text{train}} \pi(\boldsymbol{\theta}_a^{\text{train}} \mid \mathbf{s}^{(t)})$ represents the gradient of the training actor network over its parameter $\boldsymbol{\theta}_a^{\text{train}}$. As can be inferred from (28), the update process of the training actor network is guided by the target critic network through the gradient of the target critic network over the action. This ensures that the subsequent action selection follows the preferred direction of actions for optimizing the Q-value function [20], [44].

The updates on the target critic network and the target actor network are respectively given by

$$\boldsymbol{\theta}_c^{\text{target}} \leftarrow \tau_c \boldsymbol{\theta}_c^{\text{train}} + (1 - \tau_c) \boldsymbol{\theta}_c^{\text{target}}, \tag{29}$$

$$\boldsymbol{\theta}_a^{\text{target}} \leftarrow \tau_a \boldsymbol{\theta}_a^{\text{train}} + (1 - \tau_a) \boldsymbol{\theta}_a^{\text{target}}, \tag{30}$$

where τ_c and τ_a are the learning rates for updating the target critic network and the target actor network, respectively.

A. Construction of DNN for RDMB

The RDMB architecture consists of fully connected layers for the critic and actor networks, as shown in Fig. 3.

1) **Actor network** determines the optimal action given a particular state. The input layer corresponds to the state space. It comprises two hidden layers. The dimensions of each layer are determined by rounding up to the nearest power of 2 that is equal to or greater than the number of state variables. The output layer corresponds to the action space with dimensions equal to the number of action variables. It is a linear layer followed by a hyperbolic tangent function (tanh) activation function [45]. The output of this function is then scaled to the maximum possible action value.

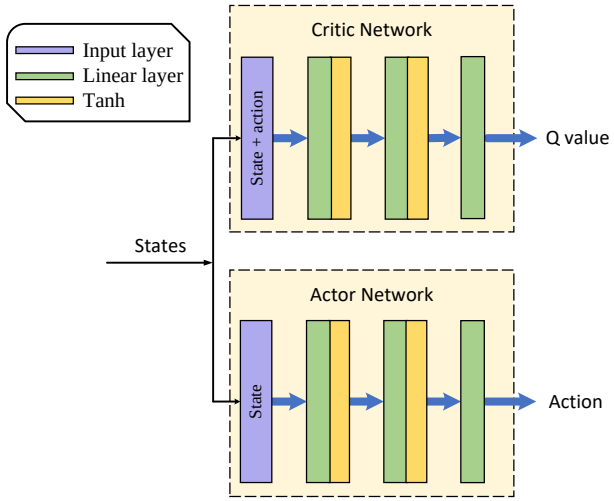

Fig. 3: The DNN structure of the critic network and the actor network for the RDMB method.

2) **Critic network**: This evaluates the expected return, or value, of being in a specific state and taking a particular action. The input layer takes in two types of inputs: the state and the action. The network contains two hidden layers. The first hidden layer is a fully connected linear layer. Its dimension is set to the nearest power of two, equal to or larger than the sum of the number of state variables and action variables. After the first hidden layer processes the state input, the action input is concatenated to its output. The second hidden layer takes this concatenated vector as input. The output layer has dimensions equal to one, representing the expected return of a given state-action pair.

In both networks, the tanh function is used as the activation function for the hidden layers to effectively handle negative inputs [45]. The training process for both the critic and actor networks employs the Adam optimizer, which uses adaptive learning rates [19], [45]. The learning rates are updated at each time step according to $\mu_c^{(t)} = \lambda_c \mu_c^{(t-1)}$ and $\mu_a^{(t)} = \lambda_a \mu_a^{(t-1)}$, where $\mu_c^{(t)}$ and $\mu_a^{(t)}$ represent the learning rates for the critic and actor networks at time step t, respectively. The parameters λ_c and λ_a are the decay rates used to gradually reduce the learning rates over time, promoting convergence during training [20], [42], [44].

A detailed description of the proposed RDMB is presented in Algorithm 2. It assumes a central controller or agent that can instantaneously gather channel information, i.e., $\{\mathbf{F}_0, \mathbf{g}_{f,k}, \mathbf{g}_{b,k}\}$. At each time step t, the agent employs the present channel information and the action executed previously, i.e., $(\mathbf{w}^{(t-1)}, \{\alpha_k\}_{k \in \mathcal{K}}^{(t-1)})$, to formulate the current state $\mathbf{s}^{(t)}$. Furthermore, we initialize $\mathbf{w}$ with the identity matrix and set $\{\alpha_k\}_{k \in \mathcal{K}}$ to a value of 0.5. The algorithm operates over

Algorithm 2 RDMB Algorithm

Input: State dimension, action space, actor and critic learning rates, $\mathbf{h}_{sr}$, $\mathbf{h}_{tr}$, and h_{st}

Initialization: Setup experience replay database of capacity D, initialize training actor network parameters $\boldsymbol{\theta}_a^{(train)}$, set target actor network parameters $\boldsymbol{\theta}_a^{(target)} = \boldsymbol{\theta}_a^{(train)}$, define training critic network parameters $\boldsymbol{\theta}_c^{(train)}$, let target critic network parameters $\boldsymbol{\theta}_c^{(target)} = \boldsymbol{\theta}_c^{(train)}$, generate received beamforming vector $\mathbf{w}$ and reflection coefficient $\{\alpha_k\}_{k\in\mathcal{K}}$

for $e = 0, 1, 2, \ldots, E-1$ **do**
  Collect and preprocess $\mathbf{h}_{sr}^{(p)}$, $\mathbf{h}_{tr}^{(p)}$, $h_{st}^{(p)}$ for the e-th iteration to generate the initial state $s^{(0)}$
  for $t = 0, 1, 2, \ldots, T-1$ **do**
    Derive action $\mathbf{a}(t) = \{\mathbf{w}^{(t)}, \{\alpha_k\}_{k\in\mathcal{K}}^{(t)}\} = \pi(\boldsymbol{\theta}_a^{(train)})$ from the actor network
    Solve (9) for optimal receive combiner, $\{\mathbf{u}\}_{k\in\mathcal{K}}^{(t)}$
      Determine new $\mathbf{s}^{(t+1)}$ using action $\mathbf{a}^{(t)}$
      Record immediate reward $\mathbf{r}^{(t+1)}$
      Preserve the experience $(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}, \mathbf{r}^{(t+1)}, \mathbf{s}^{(t+1)})$ in the replay database
    Extract the Q value function as $Q = q(\boldsymbol{\theta}_c^{(train)} \mid \mathbf{s}^{(t)}, \mathbf{a}^{(t)})$ from the critic network
    Randomly draw mini-batches of experiences of size L from the replay database
    Develop training critic network loss function $\mathcal{L}(\boldsymbol{\theta}_c^{(train)})$ as indicated in (27)
    Apply SGD on training critic network to obtain $\Delta_{\boldsymbol{\theta}_c^{\text{train}}} \mathcal{L}(\boldsymbol{\theta}_c^{\text{train}})$
    Apply SGD on target critic network to obtain $\Delta_a q(\boldsymbol{\theta}_c^{\text{target}} \mid \mathbf{s}^{(t)}, \mathbf{a})$
    Apply SGD on training actor network to obtain $\Delta_{\boldsymbol{\theta}_a^{(train)}} \pi(\boldsymbol{\theta}_a^{(train)} \mid \mathbf{s}^{(t)})$
    Update the training critic parameters $\boldsymbol{\theta}_c^{(train)}$
    Load DNN with input as $\mathbf{s}^{(t+1)}$
  end for
end for
return Optimal action $\mathbf{a} = \{\mathbf{w}^*, \{\alpha_k\}_{k\in\mathcal{K}}^*, \{\mathbf{u}\}_{k\in\mathcal{K}}^*\}$

E episodes, with each episode iterating T steps. The optimal $\mathbf{w}^*$ and $\{\alpha_k\}_{k\in\mathcal{K}}^*$ are determined by the action that yields the highest immediate reward.

VII. Refined-SAC for MIMO BiBC

The SAC method is a state-of-the-art maximum entropy DRL algorithm that optimizes the trade-off between maximizing the cumulative reward and maintaining a high entropy policy to encourage adequate exploration [20]. The proposed RSMB algorithm employs the policy iteration approach, alternating between policy evaluation and policy improvement steps [46]. This technique is off-policy, making it efficient and scalable. It allows learning from experiences stored in a replay buffer. This algorithm primarily involves three networks: two Q-networks and one policy actor network. The two Q-networks, i.e., $Q_{\boldsymbol{\theta}_1}$ and $Q_{\boldsymbol{\theta}_2}$, mitigate the overestimation of

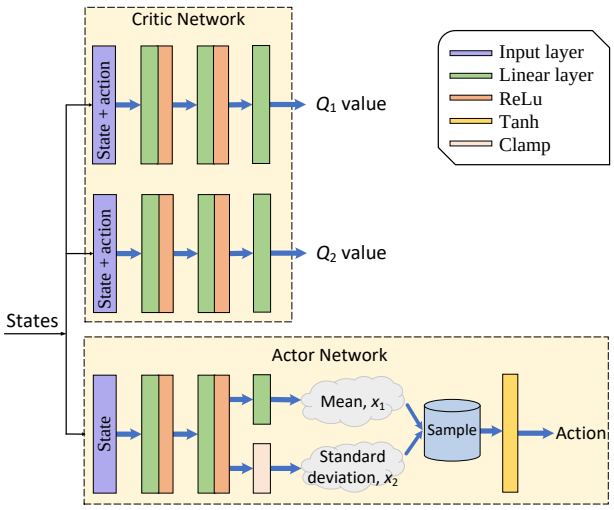

Fig. 4: The DNN structure of the critic network and the actor network for the RSMB method.

Q-value estimates, a common issue in DRL algorithms [20], [42].

The Q-networks utilize states provided by the environment, and actions generated by the actor network to derive Q-value estimates. Denote the actor network as $\pi_{\boldsymbol{\psi}}$, where $\boldsymbol{\psi}$ represents the policy parameters. In the RSMB algorithm, the Q-networks are trained jointly with the following rule:

$$\hat{\mathbf{y}} = \mathbf{r} + \gamma \min_{i=1,2} Q_{\boldsymbol{\theta}_i'}(\mathbf{s}', \mathbf{a}')\big|_{\mathbf{a}'\sim\pi_{\boldsymbol{\psi}}(\cdot|\mathbf{s}')} - \xi \log(\mathbf{a}' \mid \mathbf{s}'), \quad (31)$$

where $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$ indicates a transition obtained from the replay buffer. The $\boldsymbol{\theta}_i$ denotes the parameters of the i-th Q-network, ξ is the entropy regularization term or temperature, and μ is the learning rate. Accordingly, The loss function is defined by the MSE between the target value y^{target} and the current Q-value estimate as follows:

$$\mathcal{L}(\boldsymbol{\theta}_i) = \frac{1}{L} \left\| \hat{\mathbf{y}} - Q_{\boldsymbol{\theta}_i}(\mathbf{s}, \mathbf{a}) \right\|_2^2. \quad (32)$$

The update on the parameters of the i-th Q-network is given by

$$\boldsymbol{\theta}_i^{(t+1)} \leftarrow \boldsymbol{\theta}_i^{(t)} - \mu \nabla_{\boldsymbol{\theta}_i} \mathcal{L}(\boldsymbol{\theta}_i). \quad (33)$$

Similarly, the policy network utilizes state vectors from the environment to produce action vectors. The SAC framework calculates the policy network loss through:

$$\mathcal{L}(\boldsymbol{\psi}) = \frac{1}{L} \sum_{i=1}^{L} \xi \log \pi_{\boldsymbol{\psi}}(\hat{\mathbf{a}}_i \mid \mathbf{s}_i) - \min_{j=1,2} Q_{\boldsymbol{\theta}_j}(\mathbf{s}_i, \hat{\mathbf{a}}_i)\Big|_{\hat{\mathbf{a}}\sim\pi_{\boldsymbol{\psi}}(\cdot|\mathbf{s})}. \quad (34)$$

Subsequently, the policy gradient $\nabla_{\boldsymbol{\psi}} \mathcal{L}(\boldsymbol{\psi})$ is computed via the stochastic policy gradient algorithm and applied to update the parameters using gradient ascent:

$$\boldsymbol{\psi}^{(t+1)} \leftarrow \boldsymbol{\psi}^{(t)} + \mu \nabla_{\boldsymbol{\psi}} \mathcal{L}(\boldsymbol{\psi}). \quad (35)$$

In particular, the degree of exploration within the system is primarily regulated by the entropy regularization term, ξ. Higher values of ξ encourage more exploration. Although a deterministic policy-based DRL algorithm can be utilized, it

Algorithm 3 RSMB Algorithm

Input: State dimension, action space, actor and critic learning rates, ξ, $\mathbf{h}_{sr}$, $\mathbf{h}_{tr}$, and h_{st}

Initialization: Setup experience replay database of capacity D, initialize critic network $Q_{\boldsymbol{\theta}_1}$ and $Q_{\boldsymbol{\theta}_2}$, actor network $\pi_{\boldsymbol{\psi}}$, the parameters for training, generate received beamforming vector $\mathbf{w}$ and reflection coefficient $\{\alpha_k\}_{k\in\mathcal{K}}$

for $e = 0, 1, 2, \ldots, E-1$ **do**
  Preprocess the state and action for the e-th episode
  for $t = 0, 1, 2, \ldots, T-1$ **do**
    Derive action $\mathbf{a}^{(t)}$ from the actor network by sampling from the policy $\pi_{\boldsymbol{\psi}}$
    Solve (9) for optimal receive combiner, $\{\mathbf{u}\}_{k\in\mathcal{K}}^{(t)}$
      Detect new state $\mathbf{s}^{(t+1)}$ using action $\mathbf{a}^{(t)}$
      Compute immediate reward $\mathbf{r}^{(t+1)}$
      Preserve the experience $(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}, \mathbf{r}^{(t+1)}, \mathbf{s}^{(t+1)})$ in the replay buffer
    Sample a mini-batch from the replay buffer
    Compute the target Q-value for each tuple in the mini-batch using (31) and the critic target network
    Update the critic networks by minimizing the loss function $\mathcal{L}(\boldsymbol{\theta}_i)$ as defined in (32)
    Update the i-th Q-network parameter based on (33)
    Compute the policy loss $\mathcal{L}(\boldsymbol{\psi})$ as defined in (34) with the help of the minimum predicted Q-value of the two critic networks
    Optimize the actor by performing gradient ascent as per (35) on the expected return
    Adjust ξ by optimizing it towards maximizing entropy
  end for
end for
return Optimal action $\mathbf{a} = \{\mathbf{w}^*, \{\alpha_k\}_{k\in\mathcal{K}}^*, \{\mathbf{u}\}_{k\in\mathcal{K}}^*\}$

requires additional noise for efficient exploration. Conversely, the SAC approach's entropy regularization feature utilizes the current policy knowledge, enhancing its ability to handle complicated such as received beamforming and reflection coefficient design.

A. Construction of DNN for RSMB

Fig. 4 demonstrates the critic and actor DNN structures used for the RSMB method.

1) **Actor network**: This generates a distribution over actions for a given state. The input layer corresponds to the state space. This network has two fully connected linear layers as its hidden layers. The dimensions of these layers are calculated as the nearest power of two larger than or equal to the number of state variables. The output layer comprises two parts. One part outputs the mean of the action distribution, and the other outputs the log standard deviation of the action distribution. Both are fully connected linear layers with dimensions equal to the number of action variables. To ensure numerical stability, the outputted standard deviation is clamped between predefined minimum and maximum values.
2) **Critic network**: This predicts the Q-value for a given state-action pair. The input layer takes in both the state and the action. This network has two sets of hidden layers, effectively forming two separate sub-networks. Each sub-network has two fully connected linear layers. The dimensions of these layers are determined by the nearest power of 2 that is larger than or equal to the sum of the number of state variables and the number of action variables. Each sub-network has an output layer with a single unit, which outputs the estimated Q-value for the given state-action pair. The critic network's final output is the minimum of these two Q-value estimates to provide a conservative estimate.

All networks utilize the rectified linear unit (ReLU) as the activation function for hidden layers and the Adam optimizer for optimization with adaptive learning rates. Model parameters are updated using experiences sampled from the replay buffer. The critic network minimizes the disparity between its Q-value estimates and target Q-values derived from the actor's policy, while the actor network maximizes the expected Q-value estimated by the critic and policy entropy. The entropy coefficient (ξ) adjusts dynamically during training to balance exploration and exploitation. Critic target network parameters are periodically soft-updated from the critic network. The SAC method is outlined in Algorithm 3.

VIII. Deep Q-Benchmarks

This section presents DQN-based benchmarks, including DQN, DDQN, and DuelDQN, to evaluate the performance of the proposed algorithms.

DQN and its variants are inherently designed for discrete action spaces [17]. Applying DQN to problems with continuous action spaces requires discretizing these spaces, which enables the use of DQN but at the cost of lower sum rate performance. This can result in less effective learning and diminished performance than actor-critic architectures [17].

The trade-off here involves a balance between performance and complexity [42]. DQN methodologies, while more straightforward and easier to implement due to their reliance on discrete action spaces, tend to suffer from lower performance in terms of sum rate and learning efficiency when applied to continuous problems. This is because discretization reduces the granularity with which actions can be taken, leading to potential suboptimal decision-making [47]. On the other hand, actor-critic architectures, although more complex, can directly handle continuous action spaces. This allows for more precise action selection and potentially better overall performance, especially in environments where fine-grained control is crucial. Therefore, the choice between these approaches depends on the specific requirements of the problem, such as the acceptable level of performance and the available computational resources [20], [48].

In DQN methodologies, we need to have a discrete action space [48]. This process is generally outlined as follows:

1) **Transmit beamforming:** The selection of the beamforming vector is executed from a predetermined codebook. This codebook contains an array of potential

TABLE II: Comparison of DRL Algorithms

Algorithm	Key Feature	Application	Advantage
SAC	Entropy-based DRL	Continuous action spaces	Balances exploration and exploitation efficiently
DDPG	Actor-critic method with off-policy learning	High-dimensional, continuous action spaces	Stable and robust to hyper-parameter settings
DQN	Value-based with DL	Discrete action spaces	Simple implementation, good in stable environments
DDQN	Reduces overestimation by decoupling selection and evaluation	Discrete action spaces	Improves over DQN in stability
Dueling DDQN	Separates state-value and advantage-value estimation	Discrete action spaces	Provides finer control over Q-value estimation

Algorithm 4 Unified Deep Q-Learning Algorithm

Input: Environment with states S, actions $\mathcal{A}$, and reward function R **Initialization:** Setup replay memory to capacity D, action-value function Q with random weights $\boldsymbol{\theta}$, target action-value function $\hat{Q}$ with weights $\boldsymbol{\theta}^- = \boldsymbol{\theta}$, exploration rate ϵ, network architecture (DQN, DDQN, Dueling)

for $e = 0, 1, 2, \ldots, E-1$ **do**
  Preprocess the state and action for the e-th episode
  for $t = 0, 1, 2, \ldots, T-1$ **do**
    Select action a_t based on ϵ-greedy policy from $Q(\mathbf{s}_t, \mathbf{a}; \boldsymbol{\theta})$
    Execute action $\mathbf{a}_t$ in emulator and observe reward r_t and next state $\mathbf{s}_{t+1}$
    Store transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ in replay memory
    Sample random minibatch $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ from replay memory
    if DDQN **then**
      Set $\hat{y}_t = r_t + \gamma \hat{Q}(\mathbf{s}_{t+1}, \arg\max_{\mathbf{a}'} Q(\mathbf{s}_{t+1}, \mathbf{a}'; \boldsymbol{\theta}); \boldsymbol{\theta}^-)$
    else if Dueling **then**
      Compute $V(\mathbf{s}_t)$ and $A(\mathbf{s}_t, \mathbf{a}_t)$ using separate streams; combine to form $Q(\mathbf{s}_t, \mathbf{a}_t)$
      Set $\hat{y}_t = r_t + \gamma\big(V(\mathbf{s}_{t+1}) + A(\mathbf{s}_{t+1}, \mathbf{a}') - \frac{1}{|A|}\sum_{\mathbf{a}'} A(\mathbf{s}_{t+1}, \mathbf{a}')\big)$
    else
      Set $\hat{y}_t = r_t + \gamma \max_{\mathbf{a}'} \hat{Q}(\mathbf{s}_{t+1}, \mathbf{a}'; \boldsymbol{\theta}^-)$
    end if
    Perform GD on $(\hat{y}_t - Q(\mathbf{s}_t, \mathbf{a}_t; \boldsymbol{\theta}))^2$ with respect to $\boldsymbol{\theta}$
    Every C step, update $\hat{Q}$ with weights from Q
    Reduce ϵ gradually (e.g., $\epsilon\times$ decay factor)
  end for
  Evaluate policy π
end for

Output: Optimal action $\mathbf{a} = \{\mathbf{w}^*, \{\alpha_k\}^*_{k\in\mathcal{K}}\}$

beamforming configurations [49]. It is assumed that the CE employs analog-only beamforming vectors. The beamforming weights for a vector $\mathbf{w}$ are effectuated utilizing constant-modulus phase shifters. Additionally, it is assumed that the beamforming vector is derived from a beamsteering-based beamforming codebook $\mathcal{W}$ with a cardinality $|\mathcal{W}| := L_{\text{CE}}$. Thus, the l-th element within this codebook is defined as

$$\mathbf{w}_l = \frac{1}{\sqrt{M}}\left[1, e^{jkd\cos(\boldsymbol{\theta}_l)}, \ldots, e^{jkd(M-1)\cos(\boldsymbol{\theta}_l)}\right]^T, \quad (36)$$

where d and k denote the antenna spacing and the wavenumber, respectively, while $\boldsymbol{\theta}_l$ denotes the steering angle.

2) **Transmit power:** The transmit power is divided into $L_P = 5$ equally sized intervals, ranging from a minimum of 0 to a maximum power of P_s. This division creates a set $\mathcal{P}$, which consists of values starting from 0.1 up to P_s, evenly distributed across $L_P = 5$ points.
3) **Reflection coefficients:** The reflection coefficients $\{\alpha_k\}_{k\in\mathcal{K}}$ are quantized into $L_{\text{EH}} = 5$ discrete levels. The set $\mathcal{E}$ is constructed using evenly spaced values between $\frac{1}{L_{\text{EH}}-1}$ and $1 - \frac{1}{L_{\text{EH}}-1}$, resulting in a total of L_{EH} points.

A. Deep Q-Networks

We present an integrated approach combining DL strategies with Q-learning. At the heart of the DQN framework lies a complex DNN architecture [42], [48]. It is crafted to provide precise estimates of the Q function, which is expressed as $Q(\mathbf{s}, \mathbf{a}) = \mathbb{E}[r_{t+1} + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}') \mid \mathbf{s}, \mathbf{a}]$, where it predicts the expected rewards for taking action $\mathbf{a}$ in state $\mathbf{s}$ and following the optimal policy thereafter [48].

The DNN is trained with a variant of the Bellman equation as its loss function, denoted as $L(\boldsymbol{\theta}) = \mathbb{E}[(\hat{y}_t - Q(\mathbf{s}, \mathbf{a}; \boldsymbol{\theta}))^2]$ [42], [48]. Here, $\hat{y}_t = r_{t+1} + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}'; \boldsymbol{\theta})$ denotes the target Q-value, with $\boldsymbol{\theta}$ representing the unified set of parameters for action selection and evaluation. Additionally, the DQN methodology enhances learning efficiency and stability by implementing an experience replay mechanism. This mechanism involves storing and randomly sampling experiences, which minimizes the correlation between consecutive learning samples [50].

B. Double Deep Q-Networks

The DDQN represents an evolution of the DQN model, specifically aimed at mitigating the overestimation of Q-values inherent in the original DQN approach [48]. DDQN achieves this by decoupling the action selection and action evaluation processes [48]. The revised equation for calculating Q-values in DDQN is given by

$$Q(\mathbf{s}, \mathbf{a}) = r_{t+1} + \gamma Q(\mathbf{s}', \underset{\mathbf{a}'}{\operatorname{argmax}}\ Q(\mathbf{s}', \mathbf{a}'; \boldsymbol{\theta}), \boldsymbol{\theta}^-). \quad (37)$$

TABLE III: Simulation Hyper-Parameter

Parameter	Default Value
Maximum transmit power (P_s)	40 dBm
Carrier frequency (f_c)	3 GHz
Minimum required EH (P_{th})	-20 dBm
The number of antennas at the CE/reader (M/N)	12
Experience replay buffer capacity (D)	100000
Number of experiences in the mini-batch (L)	32
Number of steps in each episode (T)	10
Number of episodes (E)	5000
Future reward discount factor (γ)	0.99
Actor network training learning rate (μ_a)	10^{-3}
Critic network training learning rate (μ_c)	10^{-3}
Target actor network update learning rate (τ_a)	10^{-3}
Target critic network update learning rate (τ_c)	10^{-3}
Actor network training decay rate (λ_a)	10^{-5}
Critic network training decay rate (λ_c)	10^{-5}
Initial coefficient of entropy (ξ)	0.2
SAC log standard deviation clipping	$(-20, 2)$
Entropy target	$-$action dimension
Non-linear EH parameters ($\{a_{\text{NL}}, b_{\text{NL}}\}$)	$\{6400, 0.003\}$

In this setup, the decision regarding which action $\mathbf{a}'$ to take is made using the primary network, characterized by $\boldsymbol{\theta}$. However, this action's Q-value is evaluated using a separate target network with parameters $\boldsymbol{\theta}^-$. This bifurcation significantly contributes to a reduction in the overestimation of Q-values [48].

Moreover, DDQN introduces an improved methodology for updating the target network, opting for a gradual or "soft" update mechanism rather than a direct transfer of parameters from the primary to the target network [17]. This method allows for a more gradual alignment of $\boldsymbol{\theta}^-$ with $\boldsymbol{\theta}$, thereby ensuring a smoother update process for the target network and ultimately leading to a more stable learning environment [48].

C. Dueling Deep Q-Networks

DuelDQN introduces a significant modification to traditional DQN by separately evaluating the state value function $V(\mathbf{s})$ and the advantage function $A(\mathbf{s}, \mathbf{a})$ [51]. Advantage functions, $A(\mathbf{s}, \mathbf{a}) = Q(\mathbf{s}, \mathbf{a}) - V(\mathbf{s})$, are designed to distinguish between the value derived from being in a certain state s and the value derived from executing a certain action a in that state, with the formal definition being $A(\mathbf{s}, \mathbf{a}) = Q(\mathbf{s}, \mathbf{a}) - V(\mathbf{s})$ [51]. After the initial feature extraction within the network, DuelDQN proceeds to calculate the Q-value as follows:

$$Q(\mathbf{s}, \mathbf{a}) = V(\mathbf{s}) + \left(A(\mathbf{s}, \mathbf{a}) - \frac{1}{|\mathcal{A}|} \sum\nolimits_{\mathbf{a}'} A(\mathbf{s}, \mathbf{a}') \right). \quad (38)$$

This highlights the architecture's capability to independently learn the intrinsic value of each state value function and the relative advantage of each action within that state. The latter is normalized by subtracting the average advantage, stabilizing the training phase [51].

IX. Computational Complexity Analysis

Table IV compares the total computational complexity of the proposed and benchmark algorithms, where T is the number of time steps, E denotes episodes, B indicates batch size, $|D|$ is the size of the replay buffer, and l_{a_i} and l_{c_i} are layer dimensions for actor and critic networks, respectively.

The computational complexities for the AO algorithm arise from three subproblems [15]:

- The first involving matrix operations with a complexity of $\mathcal{O}(KN^2 + N^3)$;
- The second, a standard SDP problem with $\mathcal{O}(I_1(2KM^2 + M + N)^{4.5})$;
- The third uses the interior point method with the complexity of $\mathcal{O}(I_2 K^{3.5} \log(1/\epsilon))$.

For DRL methods:

- The online application phase primarily involves the actor network's execution in the proposed RDMB and RSMB algorithms, contributing to a moderate complexity [20], [44].
- The training phase for RDMB and RSMB escalates the computational load due to the simultaneous training of actor and critic networks [20], [44].
- For DQN, the complexity in both the online and training phases arises from executing a single network [42].
- DDQN introduces a second network during the training phase to reduce overestimation by decoupling the selection from the action evaluation, doubling the training complexity compared to DQN [48].
- DuelDQN modifies the network architecture to separately estimate the state value and the advantages for each action, which adds complexity due to the additional computations for these separate components [51].

The overall computational complexity for training is structured around the number of neurons in each network layer, the number of episodes, steps per episode, batch size, and the capacity of the replay memory [20], [42], [44].

In particular, RDMB and RSMB, suitable for continuous action spaces, balance complexity with performance, with RSMB offering better robustness due to entropy regularization. DQN, practical for discrete action spaces, has lower computational complexity at the cost of a lower sum rate, with DDQN and DuelDQN enhancing stability and learning efficiency by addressing overestimation bias and separating state value from the advantage function, respectively. These complexities highlight the trade-off between computational demands and performance.

X. Simulation Results

To evaluate the proposed resource allocation methods, we consider a MIMO BiBC network where the CE and the reader are positioned at coordinates $(3, 0, 0)$m and $(0, 8, 0)$m, respectively. We assume that the tags are uniformly distributed within a circle of radius $r = 2$ m, centered at $(3, 8, 0)$m. The Pathloss is given by $L(d) = C_0 \left(\frac{d}{d_0}\right)^{-\zeta}$, where $C_0 = -30$ dB represents the Pathloss at the reference distance $d_0 = 1$ m, and d denotes the distance between the transceivers. We set $\zeta = 2$ and use the Rayleigh fading model for channels. Unless otherwise specified, all simulation hyperparameters are provided in Table III. The noise power is given by $\sigma_u^2 = -147 + 10\log_{10}(\text{BW}) + \text{NF}$, where bandwidth (BW)

TABLE IV: Overall computational complexity of different methods.

Algorithm	Complexity
AO	$\mathcal{O}\left(\mathcal{O}(KN^2+N^3)+T\left(\mathcal{O}(I_1(2KM^2+M+N)^{4.5})+\mathcal{O}(I_2K^{3.5}\log(1/\epsilon))\right)\right)$
RDMB	$\mathcal{O}\left(ETB\left(2\sum_{i=0}^{2}l_{a_i}l_{a_{i+1}}+2\sum_{i=0}^{2}l_{c_i}l_{c_{i+1}}\right)+\lvert D\rvert\sum_{i=0}^{2}l_{a_i}l_{a_{i+1}}\right)$
RSMB	$\mathcal{O}\left(ETB\left(2\sum_{i=0}^{2}l_{a_i}l_{a_{i+1}}+4\sum_{i=0}^{2}l_{c_i}l_{c_{i+1}}\right)+\lvert D\rvert\sum_{i=0}^{2}l_{a_i}l_{a_{i+1}}\right)$
DQN	$\mathcal{O}\left(ETB\left(2\sum_{i=0}^{2}l_il_{i+1}\right)+\lvert D\rvert\left(\sum_{i=0}^{2}l_il_{i+1}\right)\right)$
DDQN	$\mathcal{O}\left(ETB\left(4\sum_{i=1}^{2}l_il_{i+1}\right)+\lvert D\rvert\left(\sum_{i=0}^{2}l_il_{i+1}\right)\right)$
DuelDQN	$\mathcal{O}\left(ETB\left(2\left(\sum_{i=0}^{1}l_il_{i+1}+l_v+l_aa\right)\right)+\lvert D\rvert\left(2\left(\sum_{i=0}^{1}l_il_{i+1}+l_v+l_aa\right)\right)\right)$

and noise figure (NF) are 10^6Hz and 10 dB, respectively. The training process encompasses a maximum of 5000 episodes, with each episode comprising 10 time slots, resulting in a cumulative total of 50 000 training steps.

The proposed RSMB and RDMB algorithms are compared against two benchmarks: 1) the AO method and 2) deep Q-network methods of DQN, DDQN, and DuelDQN. The minimum exploration rate for the agent, the initial exploration rate for the agent, the exploration decay rate for the agent, and number of steps between target network updates for deep Q-networks are set to 0.01, 1.0, 0.995, and 1000, respectively.

Fig. 5 illustrates the RSMB algorithm's enhanced capability in achieving higher average sum rates over epochs compared to the RDMB approach. The smoother convergence pattern observed with RSMB can be attributed to SAC's entropy-regularization feature, which promotes exploring a broad range of actions through a stochastic policy framework. This feature is especially beneficial in complex scenarios characterized by large, high-dimensional state spaces, such as those encountered in MIMO systems. Conversely, the RDMB algorithm exhibits more variability in its learning curve, likely due to its actor-critic structure's proneness to accumulating errors in the Q-function estimation process. Additionally, RDMB's performance is sensitive to the tuning of hyperparameters, including learning rates and the size of the replay buffer, which can lead to fluctuations in its effectiveness. Unlike RSMB's stochastic policy approach, RDMB's reliance on deterministic policies may limit its ability to explore the action space thoroughly. Despite these challenges, RSMB maintains a high average sum rate, demonstrating its superior ability to balance exploration and exploitation of well-known profitable strategies, making it a highly suitable algorithm for real-world applications where environments are neither stationary nor entirely predictable.

Fig. 6 assesses the performance of various algorithms when applied to the MIMO BiBC network, focusing on their adaptability to continuous, high-dimensional action spaces. Among the algorithms evaluated, the RSMB stands out for its ability to effectively navigate the intricate optimization landscape of MIMO systems. RSMB's design combines off-policy learning with an entropy-driven exploration mechanism, enabling it to converge toward the AO method. Assuming perfect CSI, the AO benchmark provides a valuable metric for assessing the practical utility of DRL algorithms in resource allocation scenarios. Also, RSMB's implementation of twin-delayed updates can reduce the overestimation bias often observed in the value network, allowing for a more accurate approximation of the AO benchmark. This feature contrasts

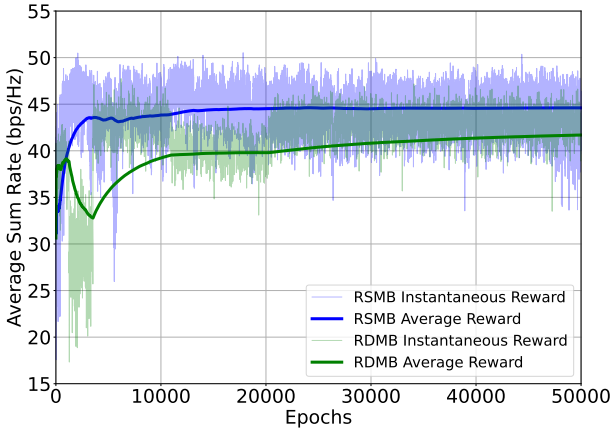

Fig. 5: Average and instantaneous sum rate versus epoch for proposed schemes when $N = M = 10$ and $P_s = 40$ dBm.

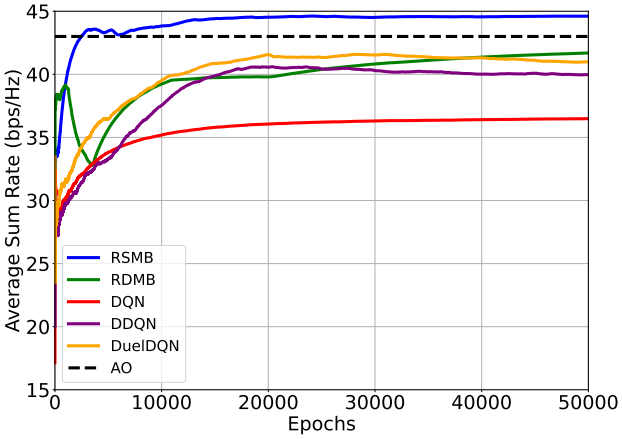

Fig. 6: Average sum rate versus epoch for different schemes when $N = M = 10$, $P_s = 40$ dBm.

with RDMB, which is prone to overestimating value functions through its single Q-value estimation mechanism, potentially leading to the adoption of suboptimal policies.

In comparison, value-based DRL algorithms such as DQN, DDQN, and DuelDQN, typically more suited to environments with discrete action spaces, fall short in this setting. Their slower convergence is mainly due to the challenges associated with discretizing the continuous action space and managing the curse of dimensionality inherent in MIMO. Conversely, policy gradient techniques like SAC and DDPG are naturally more compatible with the continuous nature of the action spaces in MIMO networks, as they learn policies directly.

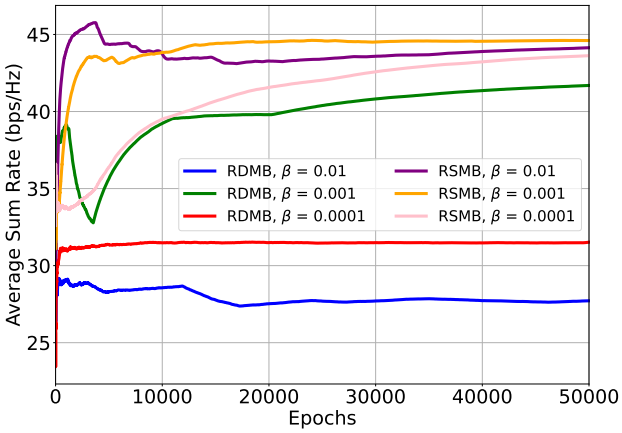

Fig. 7: Average sum rate versus epoch for proposed schemes under different learning rates when $N = M = 10$ and $P_s = 40$ dBm.

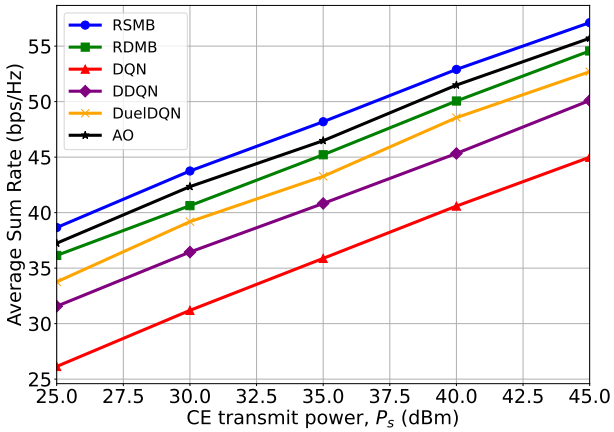

Fig. 9: Average sum rate versus CE transmit power for different schemes when $N = M = 20$.

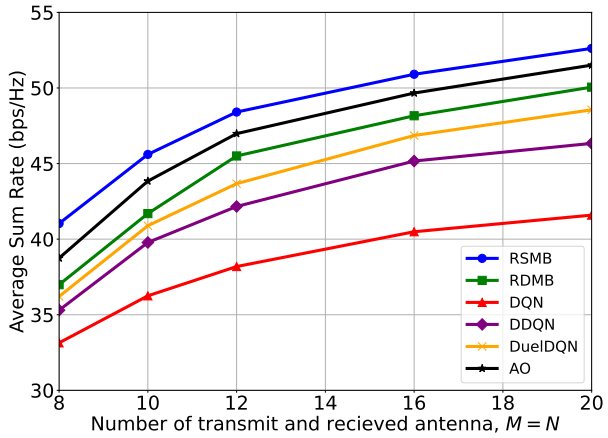

Fig. 8: Average sum rate versus number of antennae at the reader for different schemes when $N = M = 10$ and $P_s = 40$ dBm.

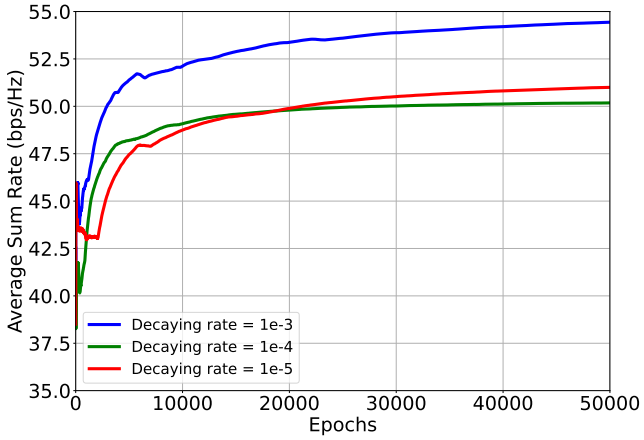

Fig. 10: Average sum rate versus epochs for DDPG under different decaying rates when $N = M = 20$ and $P_s = 40$ dBm.

Fig. 7 delves into how variations in the learning rate influence the performance of the RSMB and RDMB algorithms. The analysis reveals RDMB's acute responsiveness to changes in the learning rate, indicating that its policy updates might become overly cautious or precipitate training instability at different rates. A reduced learning rate could stabilize the training by moderating the pace of updates; however, this adjustment risks decelerating the overall learning trajectory. Conversely, RSMB exhibits a remarkable resilience to fluctuations in learning rate, a characteristic attributed to its dual-network structure. This relative insensitivity of RSMB to learning rate modifications is further enhanced by its entropy regularization feature, which systematically encourages a thorough investigation of the action space.

Fig. 8 explores the impact of increasing the number of antennas on the performance of various algorithms. This observation corroborates the theoretical expectation that additional antennas contribute to greater spatial diversity and multiplexing gains. Notably, the RSMB algorithm exhibits a more pronounced performance improvement with the antenna array's expansion, indicating its superior capability to exploit the augmented spatial opportunities for optimizing MIMO BiBC systems. This advantage is likely rooted in SAC's proficiency in managing high-dimensional continuous action spaces. In comparison, other algorithms, such as the RDMB, also benefit from adding antennas, but to a lesser extent. This figure suggests that RSMB is well-positioned to take advantage of such advancements, making it a compelling choice for adaptive and scalable resource allocation in MIMO BiBC.

When $N = M = 12$, it is observed that the RSMB scheme exhibits the highest performance gain of $26.76\,\%$, compared to DQN. AO and RDMB follow this with gains of $23.02\,\%$ and $19.16\,\%$, respectively. In contrast, DDQN and DuelDQN report more modest improvements of $10.40\,\%$ and $14.36\,\%$ when compared to DQN. Further analysis, with RDMB serving as the baseline, reveals RSMB's superiority with a $6.38\,\%$ increase in performance, while AO demonstrates a $3.24\,\%$ improvement. This detailed comparison underscores the effectiveness of RSMB in enhancing the MIMO BiBC system.

Fig. 9 compares the performance of various algorithms with the CE transmit power (P_s). In particular, RSMB stands out for its exceptional performance, showcasing an advanced approach to power management. This adaptability is essential for developing wireless devices that are power-efficient, cost-effective, and long-lasting. By reducing energy use, RSMB

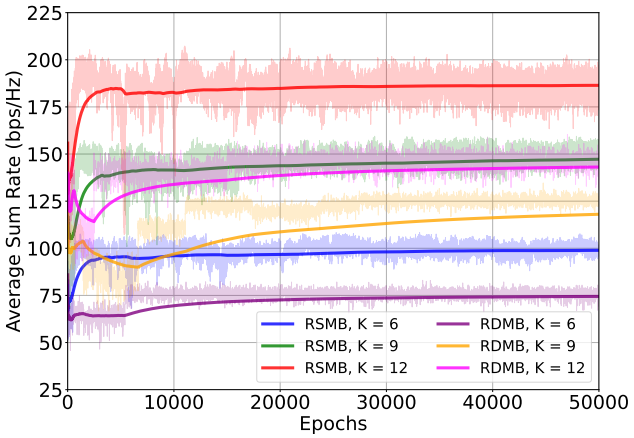

Fig. 11: Average sum rate versus epochs for proposed schemes for different numbers of users when $N = M = 20$ and $P_s = 40$ dBm.

enhances network efficiency and supports many applications and services that demand high performance.

Fig. 10 delves into how different decay rates affect the learning trajectory of the RDMB algorithm. The decay rate dictates how the learning rate decreases over time. A gradual decay rate, depicted by the $1e^{-5}$ curve, enables the algorithm to refine its policy further after initial convergence. This gradual approach is advantageous in complex scenarios, where finding the optimal policy requires navigating through a landscape fraught with local optima. Conversely, a decay rate of $1e^{-3}$ leads to a rapid early performance boost but soon reaches a plateau. This early plateau suggests the algorithm may converge prematurely to suboptimal policies.

In Fig. 11, we observe the effect of varying the number of backscatter devices on the performance of RSMB and RDMB algorithms. A key observation is that both algorithms benefit from the increased number of devices, which likely corresponds to more opportunities for constructive multi-user interference and potential capacity enhancement. The RSMB algorithm, however, seems to leverage these opportunities more efficiently, as indicated by its superior performance at higher K values. RSMB's ability to maintain a stable performance despite the increased state space suggests that its policy updates are robust to the curse of dimensionality.

XI. Conclusion

This paper addressed the intricate challenge of maximizing throughput in MIMO BiBC systems through the joint optimization of CE transmit beamforming, tag reflection coefficients, and reader reception combiners while adhering to the EH requirements of the tag. Two advanced DRL techniques, DDPG and SAC, designed for continuous state and action spaces, are exploited to propose two new algorithms (i.e., RDMB and RSMB). These algorithms employ iterative trial-and-error interactions with the environment to refine their strategies. RSMB incorporates an entropy regularization term that encourages exploration, enhancing its robustness and stability over RDMB. Our comprehensive simulation results demonstrate that RSMB outperforms AO and DQN methodologies, and in turn, AO shows superior performance over RDMB. These findings highlight the effectiveness of the advanced SAC method, as it incrementally improves and surpasses established benchmark techniques. Looking ahead, our research will pivot towards exploring multi-agent DRL frameworks to manage scenarios involving multiple readers and tags.

References

[1] "3GPP TR 38.848, Study on ambient IoT (Internet of Things) in RAN, V18.0.0 Rel. 18," Sept. 2023. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4146

[2] D. Galappaththige, F. Rezaei, C. Tellambura, and S. Herath, "Link budget analysis for backscatter-based passive IoT," *IEEE Access*, vol. 10, pp. 128 890–128 922, Dec. 2022.

[3] F. Rezaei, D. Galappaththige, C. Tellambura, and S. Herath, "Coding techniques for backscatter communications - A contemporary survey," *IEEE Commun. Surveys Tuts.*, pp. 1020–1058, 2th Quart. 2023.

[4] J.-P. Niu and G. Y. Li, "An overview on backscatter communications," *J. Commun. Inf. Netw.*, vol. 4, no. 2, pp. 1–14, Jun. 2019.

[5] S. Zargari, A. Hakimi, F. Rezaei, C. Tellambura, and A. Maaref, "Signal detection in ambient backscatter systems: Fundamentals, methods, and trends," *IEEE Access*, vol. 11, pp. 140 287–140 324, Dec. 2023.

[6] F. Rezaei, C. Tellambura, and S. P. Herath, "Large-scale wireless-powered networks with backscatter communications - A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1100–1130, Aug. 2020.

[7] G. Sacarelo, M. Awais, and Y. H. Kim, "Bistatic backscatter NOMA with transmit and receive beamforming," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 851–853.

[8] X. Jia, X. Zhou, D. Niyato, and J. Zhao, "Intelligent reflecting surface-assisted bistatic backscatter networks: Joint beamforming and reflection design," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 799–814, Jun. 2022.

[9] H. Zhao, M. Peng, Y. Ni, L. Li, and M. Zhu, "Joint beamforming and reflection design for RIS assisted bistatic backscatter networks with practical phase shift and amplitude response," *IEEE Commun. Lett.*, vol. 28, no. 6, pp. 1362–1366, Jun. 2024.

[10] D. Galappaththige, F. Rezaei, C. Tellambura, and A. Maaref, "Cell-free bistatic backscatter communication: Channel estimation, optimization, and performance analysis," *IEEE Trans. Commun.*, pp. 1–1, 2024.

[11] T. T. Anh, N. C. Luong, D. Niyato, Y.-C. Liang, and D. I. Kim, "Deep reinforcement learning for time scheduling in RF-powered backscatter cognitive radio networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7.

[12] T. T. Anh, N. C. Luong, and D. Niyato, "A deep reinforcement learning approach for backscatter-assisted relay communications," *IEEE Wireless Commun. Lett.*, vol. 10, no. 1, pp. 166–169, Jan. 2021.

[13] Q. Zhang, H. Guo, Y.-C. Liang, and X. Yuan, "Constellation learning-based signal detection for ambient backscatter communication systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 452–463, Feb. 2019.

[14] C. Liu, Z. Wei, D. W. K. Ng, J. Yuan, and Y.-C. Liang, "Deep transfer learning for signal detection in ambient backscatter communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1624–1638, Mar. 2021.

[15] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural, Parallel Sci. Comput.*, vol. 11, no. 4, pp. 351–368, Dec. 2003.

[16] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 957–975, Jul. 2020.

[17] Y. Li, "Deep reinforcement learning: An overview," *ArXiv*, vol. abs/1701.07274, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:17540505

[18] G. Dulac-Arnold *et al.*, "Deep reinforcement learning in large discrete action spaces," *ArXiv*, vol. abs/1512.07679, 2016.

[19] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *ArXiv*, vol. abs/1509.02971, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:16326763

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *ArXiv*, vol. abs/1801.01290, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:28202810

[21] F. Rezaei, D. Galappaththige, C. Tellambura, and S. Herath, "NOMA-assisted symbiotic backscatter: Novel beamforming designs under imperfect SIC," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 6829–6844, May 2024.

[22] D. Galappaththige, F. Rezaei, C. Tellambura, and S. Herath, "Beamforming designs for enabling symbiotic backcom multiple access under imperfect CSI," *IEEE Access*, vol. 11, pp. 89 986–90 005, Aug. 2023.

[23] Z. He, W. Xu, H. Shen, D. W. K. Ng, Y. C. Eldar, and X. You, "Full-duplex communication for ISAC: Joint beamforming and power optimization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 9, pp. 2920–2936, Sept. 2023.

[24] S. Zargari, A. Hakimi, C. Tellambura, and A. Maaref, "Enhancing AmBC systems with deep learning for joint channel estimation and signal detection," *IEEE Trans. Commun.*, pp. 1–1, Nov. 2023.

[25] R. Long, Y.-C. Liang, H. Guo, G. Yang, and R. Zhang, "Symbiotic radio: A new communication paradigm for passive internet of things," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1350–1363, Feb. 2020.

[26] F. Rezaei, D. Galappaththige, C. Tellambura, and A. Maaref, "Time-spread pilot-based channel estimation for backscatter networks," *IEEE Trans. Commun.*, vol. 72, no. 1, pp. 434–449, Jan. 2024.

[27] S. Abdallah, Z. Verboven, M. Saad, and M. A. Albreem, "Channel estimation for full-duplex multi-antenna ambient backscatter communication systems," *IEEE Trans. Commun.*, pp. 1–1, Mar. 2023.

[28] S. Ma, G. Wang, R. Fan, and C. Tellambura, "Blind channel estimation for ambient backscatter communication systems," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1296–1299, Jun. 2018.

[29] R. Zhang and C. K. Ho, "MIMO broadcasting for simultaneous wireless information and power transfer," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 1989–2001, May 2013.

[30] A. Hakimi, S. Zargari, C. Tellambura, and S. Herath, "Sum rate maximization of MIMO monostatic backscatter networks by suppressing residual self-interference," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 512–526, Jan. 2023.

[31] D. Wang, F. Rezaei, and C. Tellambura, "Performance analysis and resource allocations for a WPCN with a new nonlinear energy harvester model," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1403–1424, Sept. 2020.

[32] E. Boshkovska, D. W. K. Ng, N. Zlatanov, and R. Schober, "Practical non-linear energy harvesting model and resource allocation for SWIPT systems," *IEEE Commun. Lett.*, vol. 19, no. 12, pp. 2082–2085, Dec. 2015.

[33] N. Fasarakis-Hilliard, P. N. Alevizos, and A. Bletsas, "Coherent detection and channel coding for bistatic scatter radio sensor networking," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1798–1810, May 2015.

[34] Y. Liao, G. Yang, and Y.-C. Liang, "Resource allocation in NOMA-enhanced full-duplex symbiotic radio networks," *IEEE Access*, vol. 8, pp. 22 709–22 720, Jan. 2020.

[35] R. Biswas, M. U. Sheikh, H. Yigitler, J. Lempiaainen, and R. Jantti, "Direct path interference suppression requirements for bistatic backscatter communication system," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTCSpring)*, Apr. 2021.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2004.

[37] A. Hakimi, S. Zargari, C. Tellambura, and S. Herath, "IRS-enabled backscattering in a downlink non-orthogonal multiple access system," *IEEE Commun. Lett.*, vol. 26, no. 12, pp. 2984–2988, Dec. 2022.

[38] S. Zargari, C. Tellambura, and S. Herath, "Energy-efficient hybrid offloading for backscatter-assisted wirelessly powered MEC with reconfigurable intelligent surfaces," *IEEE Trans. Mob. Comput.*, vol. 22, no. 9, pp. 5262–5279, Sept. 2023.

[39] S. Stanczak, *Fundamentals of Resource Allocation in Wireless Networks Theory and Algorithms*, 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[40] W. Wan, X. Wang, J. Yang, and B. Zhao, "Joint linear pre-coder and combiner optimization for distributed antenna systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[41] K. Shen and W. Yu, "Fractional programming for communication systems—part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.

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

[43] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, May 1992.

[44] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1839–1850, Aug. 2020.

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[46] B. Saglam, D. Gurgunoglu, and S. S. Kozat, "Deep reinforcement learning based joint downlink beamforming and RIS configuration in RIS-aided MU-MISO systems under hardware impairments and imperfect CSI," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2023, pp. 66–72.

[47] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[48] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, ser. AAAI'16, 2016, p. 2094–2100.

[49] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1581–1592, Mar. 2020.

[50] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Mach. Learn.*, vol. 8, no. 3–4, p. 293–321, May 1992.

[51] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, ser. ICML'16, Jun. 2016, p. 1995–2003.