Intelligent Interactive Beam Training for Millimeter Wave Communications

Jianjun Zhang, Yongming Huang, *Senior Member, IEEE*, Jiaheng Wang, *Senior Member, IEEE*, Xiaohu You, *Fellow, IEEE*, and Christos Masouros, *Senior Member, IEEE*

***Abstract*—Millimeter wave communications, equipped with large-scale antenna arrays, are able to provide Gbps data by exploring abundant spectrum resources. However, the use of a large number of antennas along with narrow beams causes a large overhead in obtaining channel state information (CSI) via beam training, especially for fast-changing channels. To reduce beam training overhead, in this paper we develop an interactive learning design paradigm (ILDP) that makes full use of domain knowledge of wireless communications (WCs) and adaptive learning ability of machine learning (ML). Specifically, the ILDP is fulfilled via deep reinforcement learning (DRL), which yields DRL-ILDP, and consists of communication model (CM) module and adaptive learning (AL) module, which work in an interactive manner. Then, we exploit the DRL-ILDP to design efficient beam training algorithms for both multi-user and user-centric cooperative communications. The proposed DRL-ILDP based algorithms enjoy three folds of advantages. Firstly, ILDP takes full advantages of the existing WC models and methods. Secondly, ILDP integrates powerful ML elements, which facilitates extracting interested statistical and probabilistic information from environments. Thirdly, via the interaction between the CM and AL modules, the algorithms are able to collect samples and extract information in real-time and sufficiently adapt to the ever-changing environments. Simulation results demonstrate the effectiveness and superiority of the designed algorithms.**

***Index Terms*—Intelligent beam training, interactive learning design paradigm, environment sensing, beam image, deep reinforcement learning, millimeter wave communication.**

I. Introduction

To meet the demands of explosive growth system capacity and data rates, millimeter wave (mmwave) communications arise as an appealing solution and attract considerable attention for large available bandwidth [1]–[4]. However, it is far from easy to reap the benefits of mmwave communications because of the channel features of mmwave propagation. Typically, the path-loss of mmwave is much larger than that of microwave. To overcome this obstacle, a large-scale antenna array has to be used to provide a large array gain. Fortunately, the small wave-length of mmwave makes it possible to pack a large number of antennas into a compact space. Nevertheless, the large-scale antennas pose great difficulties in obtaining channel state information (CSI), especially in mobile applications or dynamic environments.

In WCs, obtaining CSI is crucial in realizing high data rates. In view of the sparsity or double-sparsity (i.e., beam-space sparsity and delay-domain sparsity [5]) and the large dimension of mmwave channels, an effective method to obtain CSI is beam training, based on which equivalent channel matrices can be estimated [6]–[8]. In beam training, candidate beams at the transmitter and/or receiver are directly trained via exhaustive or hierarchical search by selecting the ones that optimize some performance metric, e.g., signal-to-noise ratio (SNR) [6]–[10]. However, these beam search schemes are mainly suitable for single-user single-stream transmission, and the training overhead is still very high in systems with large-scale antenna arrays. Efficient multi-user beam training with low overhead is required in mmwave communications [1], but efficient solutions are still unavailable.

To improve beam training efficiency, one common method is to exploit channel prior knowledge via dynamical channel modelling [11]–[18]. Typically, most of these algorithms explicitly exploit the correlations between angles of arrival and/or departure (AoAs/AoDs) via Markov process or partially observable Markov process modeling. For example, by exploiting temporal variation of angles of departure, the authors in [11] proposed a codebook-based beam tracking strategy. Under the assumption that the mobile user moves along a straight line, an optimization algorithm based on partially observable Markov decision process is proposed in [17] for mmwave vehicular networks. Note that the existing dynamic channel models, e.g., those used in [11]–[18], are generally simplified approximations of real mmwave channels, where many stringent assumptions and simplifications are made [19], thus limiting their applicab-ilities.

Instead of explicitly exploiting prior knowledge (e.g., channel correlations via channel modeling), one promising way to reduce the training overhead is to endow the beam training algorithms with certain intelligence, enabling them to automatically extract and exploit useful information from the training history of the environment, so as to reduce the beam search space for future training. Fortunately, such an idea is facilitated by the fast development of ML and leads to ML based beam training algorithms [19]–[26]. Roughly speaking, the existing

 (Corresponding author: Yongming Huang.)

This work was supported by the National Key R&D Program of China under Grant 2018YFB1800801, the National Natural Science Foundation of China under Grants 61720106003, 61971130, 61720106003, the Research Project of Jiangsu Province under Grant BE2018121, and the Engineering and Physical Sciences Research Council, UK under project EP/S028455/1.

J. Zhang, Y. Huang, J. Wang, and X. You are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China. They are also with the Purple Mountain Laboratories, Nanjing 211111, China. (Email:{jianjunzhang, huangym, jhwang, xhyu}@seu.edu.cn). C. Masouros is with the Department of Electronic & Electrical Engineering, University College London, London WC1E7JE, U.K. (E-mail: c.masouros@ucl.ac.uk).

ML based beam training methods fall into two categories, i.e., supervised learning (SL) and non-supervised learning (NSL) methods.

As a non-interactive learning paradigm, the SL methodology includes most of the existing ML based beam training algorithms [19]–[24]. To achieve satisfying performance, the SL algorithms require a large number of training samples in advance. For example, the algorithm in [20] relies on a multi-path fingerprints database. However, collecting training samples is often costly, especially in WCs, and they have to be renewed if the environment changes. Hence, ML based algorithms that interactively collect training samples from the environment and adapt to the environment are more appealing. The existing NSL methods are mainly based on multi-armed bandit (MAB) [25], [26], a lightweight reinforcement learning method. However, due to the simplicity of the MAB, its ability to extract and exploit contextual information is very limited. Particularly, MAB is difficult to discover useful patterns, make complex decisions, and utilize existing domain knowledge. Hence, more efficient intelligent algorithms (in particular, a general design paradigm) which can better explore and exploit environment information and merge accumulated domain knowledge are desired [27].

To enjoy the benefit of powerful ML techniques and make full use of the existing WC domain knowledge simultaneously, in this paper we propose an ILDP. We fulfill the ILDP via DRL, leading to DRL-ILDP, based on which efficient beam training algorithm are developed. The DRL-ILDP based beam training designs sufficiently enjoy the advantages from the existing WC models and methods as well as the powerful ML techniques. Moreover, the interactive feature facilitates extracting useful information in real-time, thus making the designed algorithm adapt to ever-changing environments. The main contributions of this paper are summarized as follows:

- To incorporate both WC domain knowledge and ML techniques, an ILDP is customized for WCs. The ILDP consists of communication model module and adaptive learning module, which work in an interactive manner.
- We implement the ILDP via DRL, leading to DRL-ILDP. By elaborately designing the MDP state space and action space, the problem of beam training is formulated as an MDP .
- To sense the spatial distribution of effective channel paths in the beam domain, we construct beam images (BIs) based on the equivalent channel coefficients obtained via beam sweeping, and further use them to construct the MDP states.
- Based on the DRL-ILDP, an efficient beam training algorithm is proposed for the multi-user communication (MUC) by integrating deep Q-network (DQN) into the beam training procedure.
- We further extend the DRL-ILDP beam training algorithm to the user-centric cooperative communication (UCCC) by exploiting a dual relationship between UCCC and MUC.
- Comprehensive simulation results are provided to demonstrate the effectiveness and superiority of the proposed algorithms. It is shown that the proposed algorithms can capture dynamic spatial patterns and adjust beam training strategy intelligently, without knowing priori information about dynamic channel modeling.

The remainder of this paper is organized as follows. Section II describes the system model and beam training problem of mmwave multi-user communication. The principle of ILDP is introduced in Section III. In Section IV, an intelligent beam training algorithm is proposed for the mmwave multi-user communication. Section V further extends the multi-user design to the user-centric cooperative case. Simulation results and conclusions are given in Section VI and Section VII, respectively. A brief introduction to MDP and two efficient algorithms for the (communication) model problems are provided in the appendices.

Notations: Bold uppercase $\mathbf{A}$ and bold lowercase $\mathbf{a}$ denote matrices and column vectors, respectively. Non-bold letters A and a denote scalars. Caligraphic letters $\mathcal{A}$ represent sets. $\|\mathbf{x}\|$ and $\mathbf{x}(i)$ represent the L_2-norm and the i-th element of the vector $\mathbf{x}$, respectively. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ denote transpose and Hermitian operators, respectively. $\mathbb{I}_{\mathcal{A}}$ and $\mathrm{card}(\mathcal{A})$ denote the indicator function and cardinality of $\mathcal{A}$, respectively. $\mathcal{CN}(m,R)$ denotes a complex Gaussian random variable with mean m and variance R.

II. System Model

Consider a mmwave multi-user communication system, which consists of one base station (BS) and U mobile users (MUs). [1] The BS is equipped with N transmit antennas, which are controlled by T RF chains. For simplicity, it is assumed that $T = U$ and each MU has a single antenna. A hybrid analog and digital precoding design is considered, where the BS analog precoding matrix $\mathbf{A} \in \mathbb{C}^{N\times T}$ is realized by a phase shifter network, i.e., each element of $\mathbf{A}$ takes the form $\mathbf{A}(m,n) = e^{jx_{m,n}}$ $(x_{m,n} \in [0, 2\pi])$.

In practice, the analog precoder is often implemented via a predefined codebook [28]–[30], i.e., each column of $\mathbf{A}$ is selected from a codebook with finite phases (e.g., 3 to 4 bits quantization). The codebook of size M can be represented by a matrix $\mathbf{F} \in \mathbb{C}^{N\times M}$ with each column denoting a codeword. Let the i-th column of $\mathbf{F}$, i.e., $\mathbf{F}(:,i)$, be denoted by $\mathbf{f}_i$. The analog codebook can be represented by $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_M\}$. Let $\mathbf{v}_u \in \mathbb{C}^{T\times 1}$ and $s_u \sim \mathcal{CN}(0,1)$ be the digital precoding vector and the data stream of user $u \in \mathcal{U} = \{1, 2, \cdots, U\}$, respectively. Then, the received signal at MU $u \in \mathcal{U}$ can be expressed as

$$y_u = \bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{A}\mathbf{v}_u s_u + \sum_{v\neq u} \bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{A}\mathbf{v}_v s_v + w_u, \tag{1}$$

where $w_u \sim \mathcal{CN}(0,\sigma^2)$ is the received noise random variable, and $\bar{\mathbf{h}}_u$ is the channel vector between MU u and the BS.

The conventional hybrid precoding designs [29], [31], [32] generally rely on CSI, i.e., $\{\bar{\mathbf{h}}_u\}$, which, however, is difficult to obtain in mmwave systems with large antenna arrays. To overcome this difficulty, exploiting equivalent CSI obtained via beam sweeping was proposed in [28], instead of directly

[1] Later on, we will extend our design to user-centric cooperative communication, where multiple BSs cooperatively serve one MU.

estimating the physical channel vectors $\{\bar{\mathbf{h}}_u\}$. Specifically, the BS sends training signals from each direction (i.e., codeword) defined in the analog codebook $\mathcal{F}$. For each codeword in $\mathcal{F}$, MUs measure the strength of the received signals and estimate equivalent channel coefficients $\{\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{f}_1, \cdots, \bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{f}_M\}$. Under the beam sweeping framework, the BS focuses on the equivalent channel vector between the BS and each MU u of length M, i.e., $\mathbf{h}_u = \mathbf{F}^{\mathrm{H}}\bar{\mathbf{h}}_u$.

The aforementioned beam sweeping can partially circumvent the difficulty of acquiring CSI in mmwave systems. However, sweeping the entire beam space in each time-slot is still time-consuming, and is even infeasible in some cases, e.g., when the number of antennas is large and/or the transmitter and receiver are both equipped with large antenna arrays. Typically, in mmwave systems, a BS is equipped with 256 or more antennas, which makes the overhead of beam training or sweeping a heavy burden. The problem of beam training is even harder, when it comes to dynamic environments with fast-varying channels. Next, we will address this problem by developing a novel design paradigm.

III. Interactive Learning Design Paradigm

ML can automatically discover meaningful patterns from data, which is very appealing to WCs. Typically, a communication system is considered as a black box and trained in an end-to-end manner. Such methodology is however data-hungry and brings heavy burdens to WCs, where the environments are dynamic and the data are costly. To alleviate this issue, a model-driven design paradigm that exploits physical mechanisms and domain knowledge was proposed in [33], [34]. However, the underlying neural networks (NNs) are trained offline and the weights keep fixed after training, which makes the designed algorithms mainly suitable for static environments [34].

In view that WCs are the process that constantly interacts with ever-changing environments and to make full use of the domain knowledge and established models of WCs over the past several decades, we consider an ILDP for WCs. ILDP includes two modules, i.e., the communication model (CM) module and the adaptive learning (AL) module. The CM module can take full advantage of existing models and methods in WCs (e.g., sophisticated optimization techniques), and meanwhile the AL module utilizing the powerful ML techniques is in charge of extracting interested information from the environments which is difficult to obtain by conventional methods. The two modules work in an INTERACTIVE manner.

The proposed ILDP is implemented via DRL, which is a ML methodology that learns interactively from the environments. Moreover, DRL can extract and exploit interested information from historical experiences and gradually improve the performance [35]. These features make DRL particularly suitable for WCs. The mathematical foundation of DRL is Markov decision process (MDP). For completeness, a brief introduction of MDP and the deep Q-network (i.e., DQN, a powerful DRL algorithm) is provided in Appendix A. The key to exploit the proposed DRL-ILDP principle in WCs is to formulate the problem at hand as an MDP via carefully defining the states, the actions and the rewards. In particular, the domain knowledge, prior information and other system features can be embedded or encoded into the states, while the optimization variables and objectives are transformed into the actions and rewards, respectively.

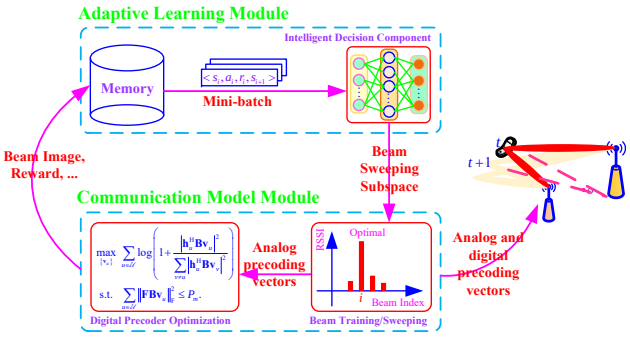

Fig. 1. A sketch of DRL-ILDP based beam training design principle.

In this paper, we exploit the DRL-ILDP to design the beam training in the afore-introduced mmwave communication system, whose sketch is shown in Fig. 1. The main task in each time-slot is to obtain the analog and digital precoding matrix/vectors. The individual roles of the CM and AL modules are as follows:

- The CM module consists of two components, i.e., digital precoder optimization and beam sweeping, whose roles are to determine the analog and digital precoders. The analog precoder (i.e., matrix $\mathbf{A}$) is determined via local beam sweeping, and the digital precoder is obtained via optimization techniques.
- The AL module is in charge of determining the beam sweeping subspace, which is influenced by the ever-changing physical environment caused by various factors (e.g., the mobility of UE). The AL method is used to extract interested statistical and probabilistic information from environments.

To better fulfill each individual goal, mutual interactions between CM and AL are of great importance. In each time-slot, CM feeds necessary experiences to AL, which are stored in the memory. The experiences include: (1) beam spatial pattern of the environment that reflects the change of the physical environment [2]; and (2) feedback (e.g., effective achievable rate) that measures the quality of the decisions made by AL. The AL module extracts statistical information from the collected experiences and makes efficient decisions. More exactly, AL provides CM with a beam sweeping subspace, which reflects the change of the exterior physical environment. In particular, the size of the beam subspace measures the variance of the change of the environment. As more experiences are collected, the decisions made are more intelligent and yield a better performance.

Compared to the existing data-driven or model-driven design paradigms, ILDP is interactive and collaboratively driven

[2]Beam spatial pattern includes beam directions and signal strength on each beam direction, spatial distribution, rate of change of the environment, and so on, which can be obtained by beam subspace sweeping and beam image construction.

by model and data. In DRL-ILDP, the underlying NN is trained online with the training data obtained from the real-time environment, which guarantees that the ILDP based algorithms can sufficiently adapt to the environment. The resulting transmission strategy is the consequence of the interaction of the CM and AL modules, thus bearing the advantages from both CM and AL. It should be pointed out that although the advantages of DRL are apparent, a possible drawback of DRL is that the convergence rate may be slow. In this paper, we speed up the convergence by explicitly extracting features via the BI technique. In addition to DRL, another important approach that can incorporate domain knowledge (e.g., WC models) in WC designs is (deep) transfer learning [36], [37]. Typically, (deep) transfer learning is suitable for low-mobility scenarios (e.g., pedestrian cellular networks), without worrying about the convergence issue in this case.

IV. Intelligent Beam Training Design for Multi-user Communications

In this section, we investigate an intelligent beam training design for the multi-user communications under the guide of DRL-ILDP. For this purpose, we first formulate the problem of beam training an MDP.

A. MDP Modeling

In this subsection, we formulate the beam training problem as an MDP. Before proceeding to details, we introduce the basic principle first. Note that since mmwave channels are sparse in the beam domain, most elements of the equivalent channel vectors $\{\mathbf{h}_u\}$ are near zero (i.e., $\{\mathbf{h}_u\}$ are also sparse) and it is sufficient to estimate the large non-zero elements of $\{\mathbf{h}_u\}$. Hence, if we can estimate these large non-zero elements by intelligently training the corresponding beams, which are often a small subset of the analog codebook $\mathcal{F}$, the overhead of beam training can be effectively reduced. Next, we formulate the problem of beam training as an MDP, i.e., define all the elements in the tuple $\mathcal{E} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, T\}$. [3]

1) Action Space: Due to the discrete nature of the training codebook $\mathcal{C}$, it is intuitive to define each beam in $\mathcal{C}$ as an action [25], [26]. However, because the change of the environment is often stochastic, it is difficult to accurately predict the beam in the next time-slot. To improve the robustness of designed algorithms, instead of defining each action as a single beam, each action is defined as a (beam) subset of $\mathcal{C}$ in this paper, as shown in Fig. 2. For convenience, the indices of the beams in the subset are assumed to be continuous. Then, the subset can be described by a pair of integers.

Definition 1. *(1) For a single UE, an action is defined by a pair of integers (a, b), where a and b denote the start beam index and the size of the subset, respectively. The beam subset corresponding to (a, b) is $\{\mathbf{f}_a, \mathbf{f}_{a+1}, \cdots, \mathbf{f}_{a+b-1}\}$.*
(2) Let the action space for MU u be denoted by

$$\mathcal{A}_u = \{(a_1^u, b_1^u), (a_2^u, b_2^u), \cdots, (a_L^u, b_L^u)\}, \tag{2}$$

where L is the size of the action space. The action space for all U MUs is the product of $\{\mathcal{A}_u\}$, i.e., $\mathcal{A} = \prod_{u=1}^{U} \mathcal{A}_u$.

[3]Note that there is no need to explicitly define the transition probabilities $\mathcal{P}$ in DRL, which is an advantage of DRL algorithms. As for the decision epoch, due to beam sweeping, a time-slot (or coherence block) is divided into three phases, i.e., beam sweeping, hybrid precoding and data transmission. A decision epoch corresponds to the beginning of a time-slot.

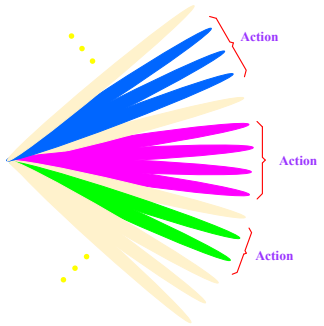

Fig. 2. Each action is defined as a subset of the codebook. Note that different subsets may overlap.

Remark 4.1 For a chosen action (a, b), the second component b characterizes the variance of the rate of the change of the environment. Intuitively, if the environment changes more irregularly, more beams (i.e., a larger b) should be swept in order to avoid misalignment.

A drawback of the previous method (according to Definition 1) to construct an action space is that it is unable to sense the rate of the change of the environment (e.g., due to movements of the MUs). Moreover, the action space may be very large, especially when the codebook $\mathcal{F}$ is large. To tackle the issue, we adopt beam index difference (BID) technique. In particular, the first component a in an action (a, b) now denotes the difference (or offset) of the indices of the (two) optimal beams in two adjacent time-slots, rather than an absolute beam index.

Remark 4.2 The BID technique is critical to the designed algorithms, typically, enabling sensing the rate of the change of the environment and shrinking the action space (and thus reducing the complexity). As an example, we explain how the BID technique shrinks the action space, since the offset (or difference) characterizes the rate of the change of the environment, which is unlikely to be large, and the two components of each action jointly define a training beam subset, the action space constructed via BID is much smaller than the original action space.

Now, given an action (a_i^u, b_i^u) of MU u in time-slot t, the beams used by the BS to sweep the beam space in time-slot t are

$$\mathcal{F}_{t,u}(a_i^u, b_i^u) = \{\mathbf{f}_{u_t + a_i^u}, \mathbf{f}_{u_t + a_i^u + 1}, \cdots, \mathbf{f}_{u_t + a_i^u + b_i^u - 1}\}, \tag{3}$$

where $\mathbf{f}_{u_t}$ denotes the optimal beam of MU u in the previous time-slot $t-1$. The beams used to sweep the beam space at the beam sweeping phase of time-slot t are

$$\mathcal{F}_t = \bigcup_{u=1}^{U} \mathcal{F}_{t,u}. \tag{4}$$

Due to the sparsity of mmwave channels, $\mathcal{F}_t$ is a subset of $\mathcal{F}$, whose size is much smaller than that of $\mathcal{F}$.

2) State Space: Let the equivalent channel vector of MU u in time-slot t be denoted by $\mathbf{h}_{t,u}$. Then, the modulus of all components of $\mathbf{h}_{t,u}$ forms a real vector, denoted by $\mathbf{I}_{t,u}$, i.e.,

$$\mathbf{I}_{t,u}(i) = |\mathbf{h}_{t,u}(i)|. \tag{5}$$

Note that in Eq.(5) and throughout this paper, for a vector, e.g., $\mathbf{x}$, its i-th element is denoted by $\mathbf{x}(i)$. As shown in Fig. 3-(a), by stacking $\mathbf{I}_{t,1}, \mathbf{I}_{t,2}, \cdots, \mathbf{I}_{t,U}$ into a matrix $\mathbf{I}_t$, i.e.,

$$\mathbf{I}_t = [\mathbf{I}_{t,1}, \mathbf{I}_{t,2}, \cdots, \mathbf{I}_{t,U}] \in \mathbb{R}^{M \times U}, \tag{6}$$

we can obtain an "image" $\mathbf{I}_t$, i.e., beam image. Note that $\mathbf{I}_t$ characterizes the distribution and strengths of effective channel paths in the spatial/beam domain in time-slot t.

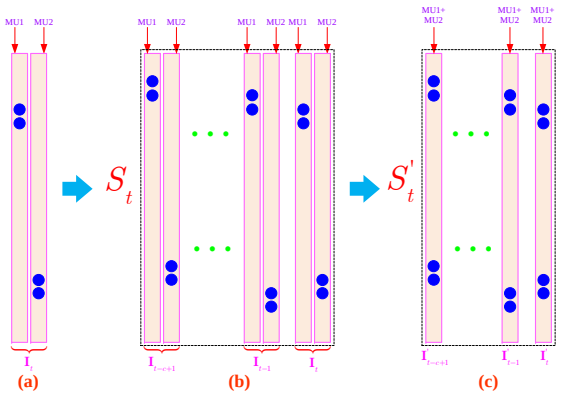

Fig. 3. An illustration of how to define a state. The blue dots denote non-zero channel coefficients. (a) The modulus of equivalent channel vectors forms a sparse image. (b) A state is defined by several successive beam images. (c) A compressed channel image can be obtained by stacking all equivalent channel vectors of different MUs in the same time-slot.

It is intuitive to define $\mathbf{I}_t$ as the MDP state S_t corresponding to time-slot t, which is, however, inappropriate since each single BI is static. As shown in Fig. 3-(b), to capture the change of the environment, we define S_t by stacking several successive BIs, i.e.,

$$S_t = [\mathbf{I}_{t-c+1}, \mathbf{I}_{t-c+2}, \cdots, \mathbf{I}_t], \tag{7}$$

where c is the number of successive images. Experiments show that a good performance can be achieved even for $c = 2$. For convenience, the technique that defines the MDP states via the BIs is referred to as beam image construction (BIC).

BIs have important and interesting properties and play a key role in beam training. Firstly, BIs construct an effective mapping from the spatial domain onto a two-dimensional grid (i.e., images), which provides an efficient and intuitive channel/beam representation. The representation builds a connection between WC and image processing (IP), which facilitates the use of advanced IP techniques (e.g., deep convolution NNs). Secondly, the representation captures the spatial/beam pattern of channel environment and its change, and improves the stability and robustness of the beam training algorithms, e.g., when the path gains vary quickly. Finally, the representation is efficient. In fact, thanks to the sparsity of mmwave channels, $\mathbf{I}_t$ is sparse and most elements of $\mathbf{I}_t$ are near zero. The sparsity of $\{\mathbf{I}_t\}$ further implies compressibility, which helps to save computing and storage resources. As shown in Fig. 3-(c), a compressed BI $\mathbf{I}'_t$ can be obtained by summing $\mathbf{I}_t$ with respect to the row, i.e.,

$$\mathbf{I}'_t = \mathbf{I}_{t,1} + \mathbf{I}_{t,2} + \cdots + \mathbf{I}_{t,U}. \tag{8}$$

Now, the size of S_t is $M \times c$, rather than $M \times Uc$.

3) Reward Function: Having performed the action A_t in time-slot t, i.e., swept the beam space with $\mathcal{F}_t$, we can obtain the equivalent channel vectors $\{\mathbf{h}_{t,u}\}$. With $\{\mathbf{h}_{t,u}\}$ available, we further optimize the analog and digital precoding vectors. Due to the sparsity and directionality of mmwave channels, the codeword that corresponds to the strongest element of $\mathbf{h}_u$ best matches the channel path. Therefore, it is reasonable to select the codeword whose array gain is largest. Without loss of generality, let $|\mathbf{h}_u(k_u)| \geq |\mathbf{h}_u(j)|, (\forall\, j \neq k_u)$. Then, the analog precoding vector chosen for MU u is set to $\mathbf{f}_{k_u}$.

With $\{\mathbf{h}_u\}$ available, we can now consider different design goals, e.g., sum-rate maximization or max-min optimization design. In this paper, we take the goal of maximizing system sum-rate as an example. Note that the analog precoder has been determined via beam sweeping. The remaining task is to optimize the digital precoder. By introducing a binary selection matrix $\mathbf{B} \in \mathbb{R}^{M \times U}$ such that

$$\mathbf{B}(i,j) = \begin{cases} 1 & i = k_u \text{ and } j = u \\ 0 & \text{otherwise,} \end{cases}$$

the design goal can be formulated as

$$\begin{aligned} \max_{\{\mathbf{v}_u\}} \ & \sum_{u \in \mathcal{U}} \log\left(1 + \frac{\left|\mathbf{h}_u^{\mathrm{H}} \mathbf{B} \mathbf{v}_u\right|^2}{\sum_{v \neq u} \left|\mathbf{h}_u^{\mathrm{H}} \mathbf{B} \mathbf{v}_v\right|^2 + \sigma^2}\right) \\ \text{s.t.} \ & \sum_{u \in \mathcal{U}} \|\mathbf{F}\mathbf{B}\mathbf{v}_u\|_{\mathrm{F}}^2 \leq P_{\mathrm{m}}, \end{aligned} \tag{9}$$

where P_{m} denotes the maximal transmit power of the BS. An efficient algorithm is provided in Appendix B to solve optimization problem (9).

Let the optimal objective function value of problem (9) be f_t^{opt}. The immediate reward in time-slot t is defined as the effective achievable sum-rate, which takes the overhead of beam training into consideration. Specifically, the immediate reward in time-slot t is defined by

$$R_t = \left(1 - (|\mathcal{F}_t| t_{\mathrm{S}} + t_{\mathrm{P}})/t_{\mathrm{C}}\right) f_t^{\mathrm{opt}}, \tag{10}$$

where $|\mathcal{F}_t|$ denotes the cardinality/size of $\mathcal{F}_t$, t_{S} denotes the duration of transmitting one beam chosen from $\mathcal{F}_t$, t_{P} denotes the duration of precoding and learning (of DRL algorithms), and t_{C} denotes the duration of one time-slot. Generally speaking, there are two principles for designing the rewards: (1) the rewards should reflect the quality of the chosen actions in terms of the chosen performance metric; and (2) the rewards shall incorporate original design objective and induce the agent to make optimal decisions. Note that $\left(1-(|\mathcal{F}_t| t_{\mathrm{S}} + t_{\mathrm{P}})/t_{\mathrm{C}}\right)$ and f^{opt} both measure the quality of the chosen action, and f^{opt} also reflects the design objective of maximizing the achievable sum-rate.

B. Multi-user Intelligent Beam Training Algorithm

Since the problem of beam training has been formulated as an MDP, efficient beam training algorithms can be obtained by integrating different RL algorithms into the beam training procedure. Because the states are continuous while the actions are discrete, DQN can be used to solve the MDP. For ease of understanding, we choose the basic version of the DQN algorithm. [4] Note that based on other RL algorithms, one can also derive the corresponding beam training algorithms.

For clarity, the DQN based intelligent (environment sensing) beam training algorithm is summarized in Algorithm 1. DQN is initialized in step 1. At the beginning of each episode [5], U initial reference beams of all MUs have to be found out first, based on which a subset $\mathcal{F}_t \subseteq \mathcal{F}$ can be constructed according to (4) and used to sweep the beam space locally. [6] Analog and digital precoding vectors are obtained by performing steps (a)-(1) - (a)-(3). With the analog and digital precoding vectors available, downlink data transmission is performed in step (b). The parameters of DQN are updated in step (c). Note that since the ε-greedy strategy is adopted to explore the environment, it may fail to find out the optimal analog precoding vectors, i.e., misalignment. If misalignment occurs, this episode is ended and the initial reference beams should be updated, typically, via exhaustive or hierarchical search. Similarly, if it is found (e.g., via monitoring the achievable rate) that the current environment becomes worse (e.g., switching from LOS to NLOS), periodical beam sweeping with a larger beam subspace (if required, even the entire beam space) should be conducted subsequently, so as to get away from the bad environment as soon as possible.

The convergence of Algorithm 1 has been demonstrated through a large number of simulation experiments, although a rigorous proof is still unavailable. Recently, it has been shown that DQN (but with a slight simplification) also theoretically converges under appropriate assumptions [40], which partly solved the problem. Algorithm 1 can be directly applied to multi-path mmwave channels, where the beam with the strongest path gain is trained and tracked with a high probability. Algorithm 1 can also be extended to wideband mmwave channels by employing standard multi-carrier techniques such as OFDM and regarding each subcarrier as an independent environment and invoking the algorithm independently. For special channels, e.g., the subchannels are highly-correlated or satisfy group sparsity, the implementation complexity can be further reduced.

In practice, there are two typical schemes to embed Algorithm 1 into a practical system. The first one is online learning and offline prediction, which is mainly suitable for stationary environments. In this scheme, the system operation consists of two phases, i.e., online learning phase and offline prediction phase. In the first phase, the algorithm senses the environment and learns required knowledge from the environment. Then, the algorithm switches to the second phase (i.e., the prediction phase) and makes predictions based on the learned knowledge. If the environment is non-stationary, the algorithm has to keep sensing the environment and continue optimizing its behavior (i.e., update the underlying NN) in each time-slot. This is the second scheme, referred to as continuous online learning scheme. Compared with the first scheme, the application scope of the second scheme is more extensive.

Algorithm 1: Environment Sensing Beam Training for Multi-user Communication System

1 **initialize** DQN: (1) replay memory D; (2) Q-function with random weights θ; (3) target Q-function with weights $\theta' = \theta$

2 **for** each episode do
- (1) **sweep** entire beam space to obtain U initial reference beams
- (2) **let** $t = 1$ and in time-slot t do
  - (a) **obtain** analog and digital precoding vectors
    - (1) choose action a_t according to ε-greedy strategy
    - (2) execute action a_t and observe next state s_{t+1}
    - (3) compute reward r_t and obtain precoding vectors
  - (b) **transmit** data in the remaining of time-slot t
  - (c) **update** parameters θ' and θ of DQN
    - (1) store transition (s_t, a_t, r_t, s_{t+1}) in D
    - (2) sample random mini-batch of transitions from D
    - (3) perform gradient descent step with respect to θ
    - (4) reset network parameters $\theta' = \theta$ every fixed steps
  - (d) **let** $t \leftarrow t + 1$

until current episode ends.

V. Extension - Intelligent Beam Training Design for User-Centric Cooperative Communications

By exploiting the dual relationship between the multi-user communication and user-centric cooperative communication, in this section we extend the multi-user communication design to the user-centric cooperative communication case.

A. User-Centric Cooperative Communication

Consider a user-centric cooperative mmwave communication system in a mmwave ultra-dense network, where one MU is cooperatively served by V BSs. Note that the BSs can also

[4] Note that various algorithms (e.g., Dueling DQN [38] and Rainbow [39]) have been proposed to overcome part of the defects of the basic DQN to improve its performance. However, we still choose the most basic DQN due to its simplicity, which can also achieve a good performance.

[5] In RL, agent-environment interaction often breaks into subsequences (i.e., episodes). Each episode ends in a special state called terminal state. As for beam training or tracking, if misalignment occurs, this episode is thought to be ended and the reference beams should be updated. Each episode usually consists of multiple time-slots. As the algorithm learns more knowledge from the environment and can make more intelligent decisions, less misalignments occur and thus each episode will consist of more time-slots.

[6] The reason for this is that the actions are designed based on beam index difference and only beam index offsets are available.

be replaced by other types of access points. For example, in a fog radio access network, the BSs are replaced by enhanced remote radio heads (eRRHs). Each BS is equipped with N antennas and connected to a central processing unit (e.g., base-band processing unit) via an error-free fronthaul link.

Let the channel vector between BS b and the MU be $\bar{\mathbf{h}}_b$. The signal received by the MU can be written as

$$y = \sum_{b=1}^{V} \sqrt{p_b} \bar{\mathbf{h}}_b^{\mathrm{H}} \mathbf{A}_b \mathbf{v}_b s + w,$$

where p_b, $\mathbf{A}_b$, $\mathbf{v}_b$ and $w \sim \mathcal{CN}(0, \sigma^2)$ denote the transmit power (of BS b), the analog precoding matrix, the digital precoding vector and the complex Gaussian random variable, respectively. The transmitted signal for the MU is denoted by s and satisfies $\mathbb{E}[ss^*] = 1$. Each component of $\mathbf{A}_b$ has a unit amplitude, i.e.,

$$|\mathbf{A}_b(i,j)| = 1, \ (1 \le i \le M, 1 \le j \le N_b),$$

where N_b is the number of RF chains that BS b allocates to the MU. The SNR for the MU can be expressed as

$$\gamma = \left| \sum_{b=1}^{V} \bar{\mathbf{h}}_b^{\mathrm{H}} \mathbf{A}_b \mathbf{v}_b \right|^2.$$

A codebook based analog beamforming is also considered for the user-centric cooperative system, i.e., each column of $\mathbf{A}_b$ is selected from an analog codebook $\mathcal{F}$. Then, the design goal is accordingly formulated as

$$\max_{\{\mathbf{A}_b, \mathbf{v}_b\}} \ \log(1+\gamma) \qquad \text{s.t.} \quad \mathbf{A}_b(:,n) \in \mathcal{F}, \quad \|\mathbf{A}_b \mathbf{v}_b\|_{\mathrm{F}}^2 \le P_b. \tag{11}$$

where P_b denotes the maximal transmit power of BS b.

Note that solving problem (11) requires CSI of all BSs, which is difficult to obtain in practice. Next, we will address this problem by formulating it as an MDP and then solving it via DRL. Before proceeding to details, it is necessary to point out the differences between the multi-user communications (MUC) and the user-centric cooperative communications (UCCC). They include action space construction and reward function design and solving, among which the most difficult issue is action space construction. Since multiple BSs serve a single MU and some BSs may use multiple beams simultaneously, the method to construct action space in MUC is inapplicable to UCCC. To tackle this issue, we exploit a dual relationship between MUC and UCCC.

B. MDP Modeling

To solve the beam training problem in UCCC, it is important to exploit the dual relationship between MUC and UCCC. Specifically, the MU in UCCC, which is the focus of a UCCC system, plays a similar role of the BS in MUC. In other words, the MU in UCCC is a virtual "BS". Accordingly, the BSs in UCCC play a similar role of the MUs in MUC, which are virtual "MUs". The essence of formulating the problem of beam training as an MDP is to define the decision epochs, states, actions and rewards of the MDP. A decision epoch of UCCC is similar to that of MUC, which is omitted here. The actions, states and rewards are defined as follows.

1) Action Space: The dual relationship between UCCC and MUC implies that each BS in UCCC has an action space and takes an action in each time-slot. However, a BS may allocate and track multiple beams for the MU, as shown in Fig. 4. The method to construct action space in MUC cannot be used directly. Otherwise, the action space will be too large, and the number of beams required to sweep the beam space in each time-slot will also be very large. To address this issue, each of these beams acts as a virtual MU and corresponds to an action space.

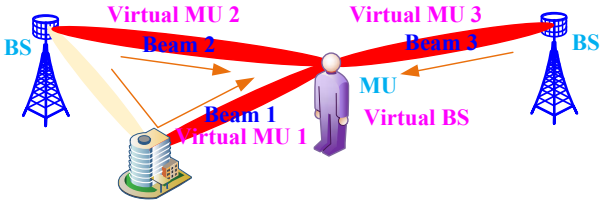

Fig. 4. A BS may align and track multiple beams. In this case, each beam acts as a virtual MU.

To fully utilize the hardware resources, it is assumed that the number of beams that BS b allocates to the MU (i.e., the number of virtual MUs belongs to BS b) is N_b, which may be larger than 1. Then, the action space of BS b is defined as

$$\mathcal{A}^b = \prod_{k \in \mathcal{N}_b} \mathcal{A}_k,$$

where $\mathcal{N}_b = \{1, 2, \cdots, N_b\}$ is the set of virtual MUs of BS b, and $\mathcal{A}_k$ is the action space for virtual MU k. Similar to the MUC case, the action space for virtual MU k is defined as

$$\mathcal{A}_k = \{(a_1^k, b_1^k), (a_2^k, b_2^k), \cdots, (a_L^k, b_L^k)\},$$

where L is the size of the action space. The entire action space for the UCCC system is the product of $\mathcal{A}^b$, i.e.,

$$\mathcal{A} = \prod_{b=1}^{V} \mathcal{A}^b.$$

For a given action $a \in \mathcal{A}$ in time-slot t, a beam subset $\mathcal{F}_t$ can be constructed according to (3) and (4) to sweep the beam space locally.

2) State Space: Similar to the MUC case, the equivalent channel vector of virtual MU u in time-slot t is denoted by $\mathbf{h}_{t,u}$. Then, the modulus of each component of $\mathbf{h}_{t,u}$ also forms a vector, denoted by $\mathbf{I}_{t,u}$, i.e.,

$$\mathbf{I}_{t,u}(i) = |\mathbf{h}_{t,u}(i)|.$$

Let $U' = \sum_{b=1}^{V} N_b$. Stacking $\mathbf{I}_{t,1}, \mathbf{I}_{t,2}, \cdots, \mathbf{I}_{t,U'}$ of all virtual MUs into a matrix yields a beam image, i.e.,

$$\mathbf{I}_t = [\mathbf{I}_{t,1}, \mathbf{I}_{t,2}, \cdots, \mathbf{I}_{t,U'}] \in \mathbb{R}^{M \times U'}.$$

The beam image $\mathbf{I}_t$ characterizes the distribution of effective channel paths in the beam domain, which is often sparse due to the sparsity of mmwave channels.

To capture the change of the environment, the state S_t is similarly defined by several successive beam images, i.e.,

$$S_t = [\mathbf{I}_{t-c+1}, \mathbf{I}_{t-c+2}, \cdots, \mathbf{I}_t],$$

where c denotes the number of successive beam images.

3) Reward Function: The immediate reward R_t in time-slot t is defined as the effective achievable rate, i.e.,

$$R_t = \big(1 - (|\mathcal{F}_t|t_{\mathrm{S}} + t_{\mathrm{P}})/t_{\mathrm{C}}\big) f_t^{\mathrm{opt}}, \tag{12}$$

where the meaning of $\mathcal{F}_t$, t_{S}, t_{P} and t_{C} is similar to that in (9). f_t^{opt} in (12) is the optimal objective function value of problem (11). Let $\mathbf{h}_b = \mathbf{A}_b^{\mathrm{H}} \bar{\mathbf{h}}_b$ with $\mathbf{A}_b$ determined via beam training. Then, problem (11) can be rewritten as

$$\max_{\{\mathbf{v}_b\}} \quad \left| \sum_{b=1}^{V} \mathbf{h}_b^{\mathrm{H}} \mathbf{v}_b \right|^2 \qquad \text{s.t.} \quad \|\mathbf{A}_b \mathbf{v}_b\|_{\mathrm{F}}^2 \le P_b. \tag{13}$$

An iterative algorithm is designed to solve optimization problem (13). Please refer to Appendix C for more details.

C. User-Centric Intelligent Beam Training Algorithm

Since the problem of beam training has been formulated as an MDP, efficient beam training algorithms can be obtained immediately. For simplicity, we still choose the basic DQN algorithm, which yields a DQN based intelligent beam training algorithm. For clarity, it is summarized in Algorithm 2.

Algorithm 2: Environment Sensing Beam Training for User-Centric Cooperative Communication System

1 **initialize** DQN algorithm (Similar to Algorithm 1)

2 **for** each episode do

(1) **find** out U' initial reference beams

(2) **let** $t = 1$ and in time-slot t do

(a) **obtain** analog and digital precoding vectors

(1) choose action a_t according to ε-greedy strategy

(2) execute action a_t and observe next state s_{t+1}

(3) compute reward r_t and obtain precoding vectors by solving problem (24)

(b) **transmit** data in the remaining of time-slot t

(c) **update** parameters θ' and θ of DQN

(d) **let** $t \leftarrow t + 1$

until current episode ends.

In step 1, the parameters of DQN (including the weights of Q-function network and target Q-function network, and replay memory) are initialized. At the beginning of each episode, U' initial reference beams have to be found out, based on which a subset $\mathcal{F}_t \subseteq \mathcal{F}$ constructed according to (4) is used to sweep the beam space. Analog and digital precoding vectors are obtained by performing steps (a)-(1) - (a)-(3). With the analog and digital precoding vectors available, data transmission can be performed next, i.e., in step (b). Finally, the parameters of DQN are updated in step (c), which is similar to Algorithm 2 and omitted. Note that the update of NNs in DRL algorithms, which may require intensive computation, can be fulfilled in the phase of data transmission by the BS with powerful computational resources.

VI. Numerical Results

In this section, we evaluate the performance of the proposed intelligent beam training algorithms via numerical results. The simulation setting is as follows. Uniform linear array (ULA) and the analog DFT codebook are adopted. [7] The size of the DFT codebook $\mathcal{F}$ is $M = N$. For each MU or virtual MU u, the action space $\mathcal{A}_u$ is given by

$$\mathcal{A}_u = \big\{(a, b) \,|\, a = -1, 0, 1, 2; b = 2, 4, 6\big\}.$$

The determination of the action space depends on the scenario or environment where the algorithms will be applied. If the environment changes more irregularly, the actions space should be enlarged. The structure of the NN adopted in the simulation contains one convolution layer, one down-sampling layer and one fully-connected layer. For small-to-medium sized antenna arrays (e.g., typically $N \le 64$), neural networks with only two fully-connected layers also work well.

A. Multi-user Intelligent Beam Training

The simulation environment is illustrated in Fig. 5. The channel between the BS and each MU includes one LOS path and three NLOS paths if the LOS path is not blocked. The speed of each MU is stochastic, but obeys a probability law. Accordingly, switching to another beam in the next time-slot is also stochastic and obeys some probability law. For each MU, if the LOS path is not blocked, the probability that the optimal beam switches to the i-th beam of the next S beams is denoted by $p_{S,i}$ $(i = 0, 1, \cdots, S)$, where $p_{S,0}$ represents the probability that the optimal beam in the next time-slot is still the current beam. As an example, two probability distributions are considered, where $\{p_{S,i}\}$ are given by

$$\text{model 1:} \quad p_{S,i} = e^{-\eta i} \left(\sum_{k=0}^{S} e^{-\eta k} \right)^{-1} \tag{14}$$

$$\text{model 2:} \quad p_{S,i} = e^{-\eta (S-i)} \left(\sum_{k=0}^{S} e^{-\eta k} \right)^{-1}. \tag{15}$$

The parameter $\eta > 0$ in (14) and (15) defines the "decay" rate. An instance of the two probability models is provided in Fig. 6. The probability model in (14) indicates that the MU moves relatively slow, while the probability model in (15) indicates that the MU moves relatively fast. Note that the algorithms proposed in this paper are not limited by the considered simulation models.

First, we demonstrate the crucial role of the BIC technique used to design the beam training algorithms. It is well known that as an important feedback from the external environment to the agent in the reinforcement learning, the rewards are important to making decision. According to (10), the rewards depend on both beam training overhead and achievable sum-rate, while the later is further determined by multiple factors, such as transmit power and channel fading, which may affect the performance in terms of robustness and/or stability of the

[7] Note that the proposed beam training algorithms also apply to other types of antenna arrays and codebooks, e.g., a uniform planar array along with a codebook constructed by sampling the elevation-azimuth plane.

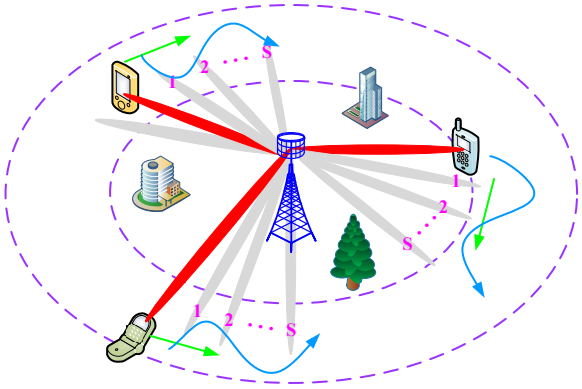

Fig. 5. Simulation environment - U MUs move within an annulus and the BS is located at the center of the annulus.

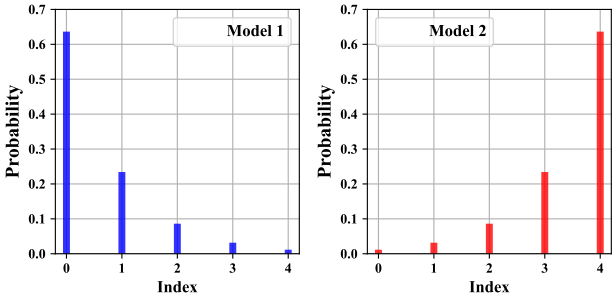

Fig. 6. An illustration of the probability distributions in (14) and (15): $S = 4$ and $\eta = 1$.

designed algorithms. For example, when the transmit power is changed, the reward will also be changed, which may induce the agent to take a wrong action.

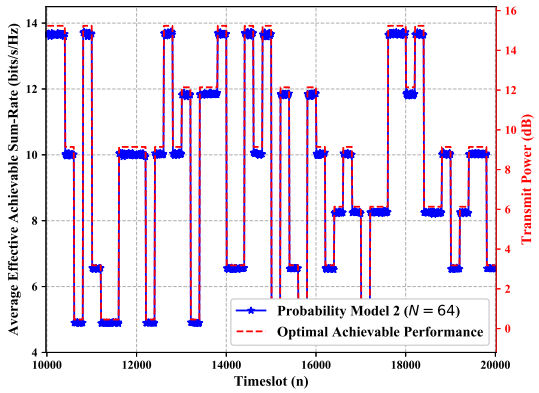

Fig. 7. The AEASR performance of Algorithm 1 (averaged over 300 realizations). The dotted lines represent the optimal performance, i.e., the optimal actions are taken each time.

Fig. 7 demonstrates the stability of Algorithm 1. Average effective achievable sum-rate (AEASR) is, in fact, the average immediate reward calculated according to (10). In particular, in addition to the change of beam directions, the transmit power is also changed randomly within {0dB, 3dB, 6dB, 9dB, 12dB, 15dB} every 200 time-slots. It is observed that a near optimal performance in terms of effective achievable sum-rate can still be achieved at/near the point where the transmit power has just changed, which indicates that Algorithm 1 is robust to the change of transmit power. The reason for this is two-fold. Firstly, the agent takes the current state into account when making decisions [8]. Secondly, and more importantly, the states are designed via BIC, which can extract and encode important information about the environment (e.g., beam spatial pattern, rate of change of the environment, and so on) and helps the agent make wise decisions. Therefore, the designed algorithm is robust to the change of transmit power and achieves a stable performance.

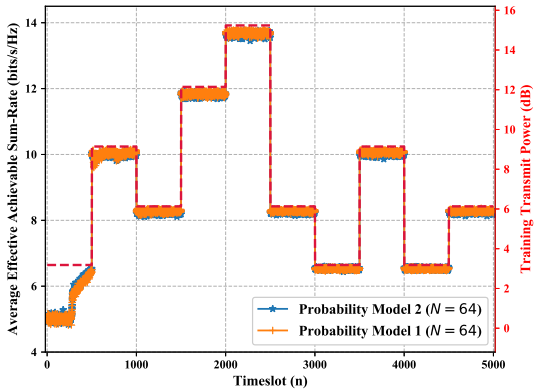

Fig. 8. The learning (or convergence) performance of Algorithm 1, which is obtained by averaging over 300 simulation realizations. The dotted lines represent the optimal performance.

Next, we demonstrate the learning (or convergence) performance of Algorithm 1, as shown in Fig. 8. Similarly, during the learning process, the transmit power is randomly changed every 500 time-slots. It is observed that Algorithm 1 learns fast and converges within about 550 time-slots. Interestingly, it is observed that the learning performance of (probability) model 1 is similar to that of model 2, although the MU in model 2 moves faster than the MU in model 1. This attributes to the use of the beam index difference technique. In fact, the component a in the action (a, b) measures the rate of change of the environment, which takes a larger value in model 2.

We next comprehensively evaluate the AEASR performance of the designed environment sensing beam training (ESBT) algorithm, i.e., Algorithm 1 in this paper. The simulation results of the exhaustive search based beam training (ExSeBT) algorithm, the hierarchical search based beam training (HSBT) algorithm [8] and the stochastic bandit learning based beam training (SBLBT) [41] algorithm are also provided for comparison.

The AEASR performance of different beam training algorithms is shown in Fig. 9. It is seen that ESBT achieves the best performance among the four algorithms, and approaches the oracle aided beam training (OABT) algorithm [9]. The reason for this is that ESBT can sense the change of the environment, and adjusts beam training strategy intelligently, which reduces the training overhead and reserves more time

[8]Note that a policy is defined by a mapping from a state to a probability distribution.

[9]The optimal beams are provided by oracle, causing no training overhead. It is an optimal but ideal algorithm, which is often used as a benchmark.

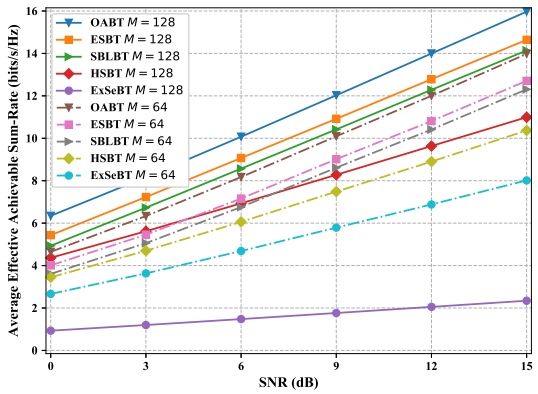

Fig. 9. The AEASR performance of different beam training algorithms - $S=4$ and model 1 in (14).

for data transmission. Although ESBT and ExSeBT can find the optimal beams and thus achieve large array gains, the training overhead is too large, which reduces the AEASR performance. Similar to ESBT, SBLBT also sweeps the beam space locally. However, the change of the beamforming gain (e.g., due to the random locations of the MUs) affects the rewards, which induces the algorithm to take sub-optimal decisions. In contrast, the BIC technique used in ESBT greatly mitigates and even avoids this adverse effect.

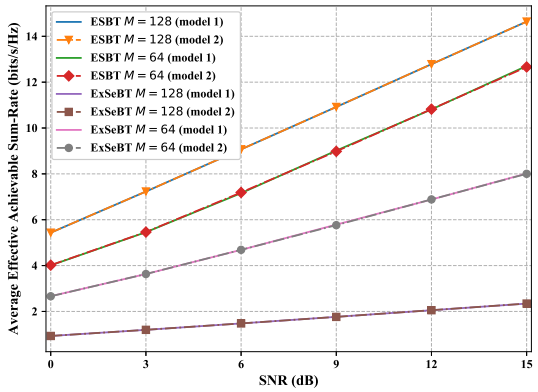

Fig. 10. The AEASR performance of ESBT and ExSeBT for the two probability models - $S=4$.

Fig. 10 demonstrates the AEASR performance of ESBT and ExSeBT for the two probability models. It is not surprising that ExSeBT achieves the same AEASR performance for the two probability models, since its training overhead keeps fixed when the size of the codebook is given. Nevertheless, it is interesting to observe that the AEASR performance corresponding to model 2 achieved by ESBT is almost the same as that corresponding to model 1, although the environment corresponding to model 2 varies faster than that corresponding to model 1. The reason for this is that although the average rate of change of the environment in model 2 is larger than that in model 1, the variances of the rates of the changes of the two models are the same. Hence, the training overheads corresponding to the two models are the same, which leads to almost the same AEASR performance. This is an important and desirable feature of ESBT.

B. User-Centric Cooperative Intelligent Beam Training

The simulation environment is illustrated in Fig. 11. Similarly, the channel between the MU and each BS includes one LOS path and three NLOS paths, and switching to another beam in the next time-slot is stochastic and obeys some probability law. In view of the similar performance achieved for model 1 and model 2, we consider another probability model, i.e.,

$$\text{model 3:} \quad p_{S,i} = \frac{1}{S+1}. \tag{16}$$

It can be verified that the variance of the rate of the change of the model in (16) is larger than that of the model in (14) or (15).

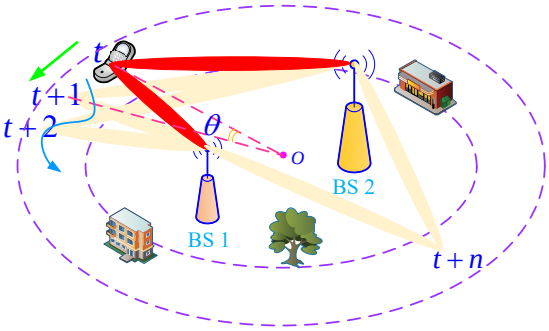

Fig. 11. Simulation environment - the MU moves within an annulus and two BSs are located within the inner circle of the annulus.

In the previous subsection, we have demonstrated the learning performance of Algorithm 1. Note that due to the dual relationship between Algorithm 1 and Algorithm 2, the two algorithms share many characteristics, e.g., the learning performance and beam tracking performance. For this reason, the learning performance of Algorithm 2 and the beam tracking performance of Algorithm 1 will not be provided due to space limitation.

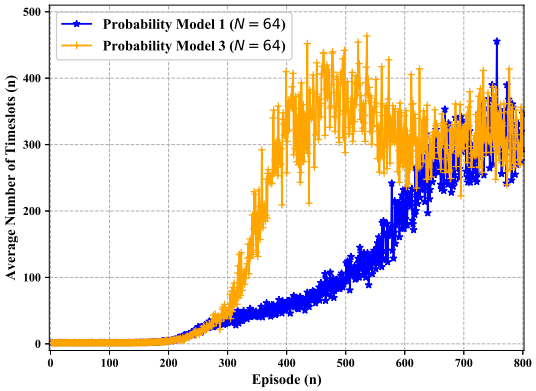

Fig. 12. The beam tracking performance of Algorithm 2 in the training process, which is obtained by averaging over 100 simulation realizations.

We first evaluate the beam tracking performance of Algorithm 2. Since an episode is defined as a process from the initial alignment until failure, the number of time-slots contained

in each episode during the learning process characterizes the beam tracking performance. As shown in Fig. 12, one can observe that more and more time-slots are contained in the episode as the episode index increases, which indicates that a better and better beam tracking performance is achieved. Note that this coincides with our intuition. In fact, this is because more and more knowledge is learned from the environment, which helps to take actions more intelligently. It is also seen that Algorithm 2 performs better in model 3 than in model 1. The reason for this is that the experiences collected in model 3 are more well-balanced.

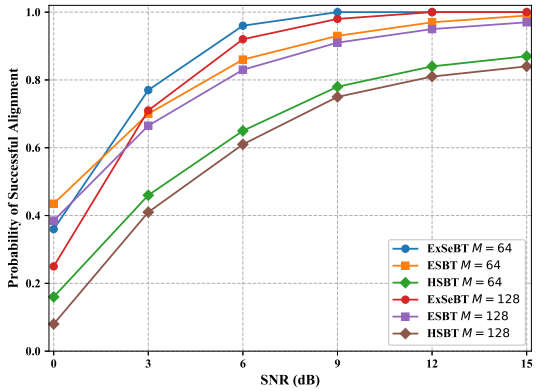

Fig. 13. The probabilities that all beams are successfully aligned by different beam training algorithms - $S = 4$ and probability model 1 in (14).

Fig. 13 shows the probabilities that all beams are successfully aligned by Algorithm 2, ExSeBT and HSBT (by necessary modifications). It can be observed that ExSeBT achieves the highest probability of successful alignment. However, ExSeBT sweeps the entire beam space, which is time-consuming and may be inapplicable if the channel or the environment varies fast. Although the probability of successful alignment of ExSeBT is a bit higher than that of ESBT, the gap between the two algorithms becomes negligible as SNR or the number of antennas increases. It is also observed that HSBT achieves the worst performance, because HSBT is mainly suitable for the situation where different beams are sufficiently separated.

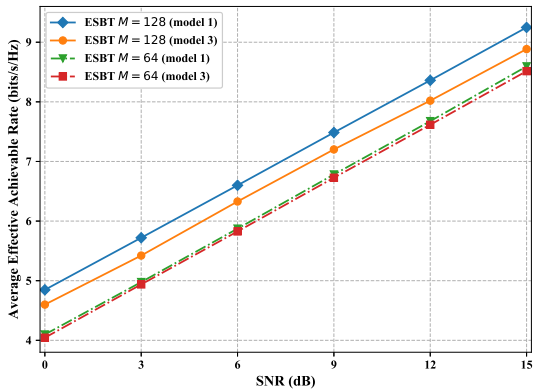

Fig. 14. The AEAR performance achieved by ESBT for the two probability models - $S = 5$.

Fig. 14 shows the average effective achievable rate (AEAR) performance achieved by ESBT for probability models 1 and 3 with $S = 5$. It is observed that the AEAR performance corresponding to model 3 is lower than that corresponding to model 1. The reason is as follows. The optimal action corresponding to model 3 is $(0, 6)$, which can both avoid misalignments and achieve a good AEAR performance. However, this action is not optimal for model 1. In fact, although misalignments can be avoided by taking this action, the resultant training overhead is also large, which, on the contrary, can decrease the AEAR performance. In view of the fact that $P_{S=5,5}$ is very small, the action $(0, 5)$ (maybe $(0, 4)$) may be a better choice. More specifically, although the action may cause some misalignments (but with a very small probability), the overall training overhead has been reduced. The analysis sufficiently indicates that ESBT can intelligently make decisions and take actions.

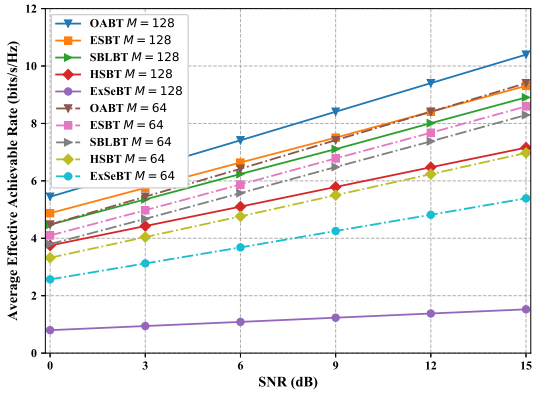

Fig. 15. The AEAR performance of different beam training algorithms - $S = 4$ and probability model 1 in (14).

Fig. 15 demonstrates the AEAR performance of different beam training algorithms, including ExSeBT, HSBT (via necessary modifications) and SBLBT (also with some modifications). Similar to the MU case, it is observed that ESBT achieves the best performance among the four beam training algorithms, and approaches the performance of the ideal OABT algorithm. It is not a surprise since ESBT can adjust the beam training strategy intelligently by firstly sensing the rate of change of the environment. In contrast, SBLBT is affected by the fluctuation of the array gain (e.g., due to the random location of the MU or the transmit power), while the training overhead of ExSeBT and HSBT is large, which finally reduces their AEAR performance.

VII. Conclusion

In this paper, we proposed an efficient beam training design from the perspective of environment sensing. To facilitate the combination of communication domain knowledge and ML techniques, we proposed a WC-suitable interactive learning design paradigm. Then, we proposed an efficient beam training design from the perspective of environment sensing. Specifically, we first formulated the problem of beam training as a MDP. To capture the dynamic spatial patterns of environments, the BIC technique was proposed to define MDP states. Then,

an efficient beam training algorithm was further proposed for multi-user communications by integrating DQN into the beam training procedure. Next, we extended the design to the case of user-centric cooperative communication. Finally, simulation results were provided to demonstrate the effectiveness and superiority of our designs. Particularly, the designed algorithms require no priori knowledge of dynamic channel modeling, and thus can apply to a variety of complicated scenarios.

APPENDIX A
MARKOV DECISION PROCESS AND DEEP REINFORCEMENT LEARNING

In this appendix, we briefly introduce Markov decision process (MDP) and DRL, to simplify the description of DRL based beam training designs.

A. Markov Decision Process

An MDP is defined by a tuple $\mathcal{E} = \{\mathcal{T}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, which consists of five elements, i.e., decision epochs, states, actions, transition probabilities and rewards. Decisions are made at points of time referred to as decision epochs and denoted by $\mathcal{T} = \{1, 2, 3, \cdots, \}$. An element of $\mathcal{T}$ is denoted by t and referred to as "time t". $\mathcal{S}$ and $\mathcal{A}$ are sets called the state space and action space, respectively. $\mathcal{P} = \{p(s'|s,a) \,|\, s', s \in \mathcal{S}, a \in \mathcal{A}\}$ defines the one-step dynamics of the environment. Specifically, $p(s' \,|\, s, a)$ denotes the transition probability that the agent transitions from s to s' when taking action a, i.e.,

$$p(s' \,|\, s, a) = \mathbb{P}(S_{t+1} = s' \,|\, S_t = s, A_t = a),$$

where S_t and A_t are the state and action at time t, respectively. $\mathcal{R} = \{r(s,a) \,|\, s \in \mathcal{S}, a \in \mathcal{A}\}$ is a collection of rewards for all possible state-action pairs. When the agent takes action a in state s, it receives a reward $r(s,a)$.

A policy is used to select actions by the agent. It is usually a (randomized) mapping and denoted by $\pi_\theta : \mathcal{S} \to \mathcal{M}(\mathcal{A})$, where $\mathcal{M}(\mathcal{A})$ is the set of probability measures on $\mathcal{A}$ and θ denotes parameters. Particularly, $\pi_\theta(a_t|s_t)$ is the conditional probability at s_t associated with the policy. Using the policy to interact with the environment gives a trajectory of states, actions and rewards over $\mathcal{S} \times \mathcal{A} \times \mathbb{R}$, which is denoted by

$$\mathcal{H}_{1:T} = \{s_1, a_1, r_1, s_2, a_2, s_2, \cdots, s_T, a_T, r_T\},$$

where T is a positive integer and could be ∞.

The return D_t^γ is defined as cumulative discounted reward from time t onwards, i.e.,

$$D_t^\gamma = \sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}),$$

where $\gamma \in (0,1)$ is the discounted factor. The value of a state s under a policy π, denoted by $v_\pi(s)$, is defined as

$$v_\pi(s) = \mathbb{E}_\pi\big[D_1^\gamma \big| S_1 = s\big], \tag{17}$$

where $\mathbb{E}_\pi$ denotes the expectation on the interaction sequence under the policy π. Similarly, the value of taking action a in state s under the policy π, denoted by $q_\pi(s,a)$ and referred to as Q-function or Q-value, is defined as

$$q_\pi(s,a) = \mathbb{E}_\pi\big[D_1^\gamma \big| S_1 = s, A_1 = a\big]. \tag{18}$$

The goal of reinforcement learning is to find a policy which (approximately) maximizes the cumulative discounted reward from a start state, i.e.,

$$\max \;\; J(\pi) = \mathbb{E}\big[D_1^\gamma \big| \pi\big].$$

B. Deep Reinforcement Learning

If the complete information of the MDP are available, dynamic programming can be used to solve the MDP. However, obtaining complete MDP, especially the transition probabilities $\mathcal{P}$, is often difficult and even impossible. To circumvent this difficulty, the Reinforcement Learning (RL) methodology has emerged [35]. Q-learning is the most well-known and widely used one among various RL algorithms. The key of Q-learning is the following update formula

$$q(s,a) \leftarrow (1-\alpha) q(s,a) + \alpha[r + \gamma \max_{a'} q(s', a')], \tag{19}$$

where α is the learning rate, r is the immediate return, $q(s,a)$ is the Q-value for the current state s and action a pair, and $q(s',a')$ is the Q-value for the state action pair at the resultant state s' after action a was taken at state s.

Despite its popularity, Q-learning is a table-based algorithm and mainly suitable for small-scale discrete problems due to the limitation of memory and computational capacity. To solve MDPs with large or continuous state space, deep Q-network (DQN) was proposed in [42], where deep NNs (DNNs) are used as function approximators for the Q-values. Specifically, DQN uses a DNN with weights θ to parameterize the Q-value $q(s,a)$, which yields $q(s,a;\theta)$.

To train the DNN, DQN starts with some random initialization of the Q-value $q(s,a;\theta_0)$, where θ_0 denotes the initial weights. Then, an approximation $q(s,a;\theta_k)$ of the Q-value at the k-th iteration is updated towards the target

$$y_k^{\text{tar}} = r + \gamma \max_{a' \in \mathcal{A}} q(s', a'; \theta_k). \tag{20}$$

The weights θ_k are updated via stochastic gradient descent by minimizing the square loss

$$L_{\text{DQN}} = \big(q(s,a;\theta_k) - y_k^{\text{tar}}\big)^2,$$

which amounts in updating the weights as follows

$$\theta_{k+1} = \theta_k + \alpha\big(y_k^{\text{tar}} - q(s,a;\theta_k)\big) \triangledown_{\theta_k} q(s,a;\theta_k). \tag{21}$$

When updating the weights θ, the target y_k^{tar} changes as well, which makes the training instable. To address this issue, two important features are added to DQN, i.e., set target network to compute target Q-values and use experience replay. First, the target Q-value in (20) is replaced by $q(s', a'; \theta_k^-)$, where its weights θ_k^- are updated only every $C \in \mathbb{Z}^+$ iterations with the assignment: $\theta_k^- = \theta_k$. The second feature added to DQN is experience replay. The key idea is that the agent can store its experiences and use them in batches to train the DNN. Storing the experiences allows the agent to randomly draw batches and helps the network to learn with approximate independent samples. Each of these experiences are stored in the form of *state*, *action*, *reward* and *next state*, i.e., $< s, a, r, s' >$. A sketch of the DQN algorithm is provided in Fig. 16. More details about DQN can be found in [42].

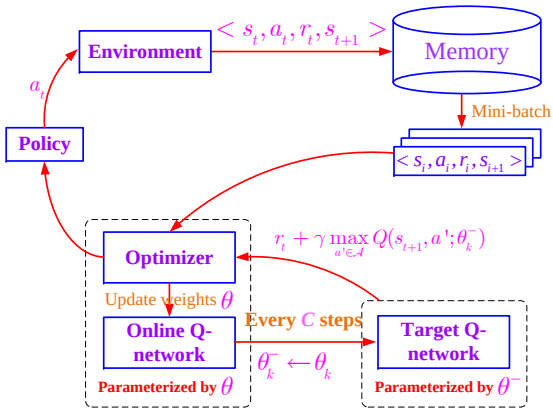

Fig. 16. A sketch of the DQN algorithm.

Appendix B
Algorithm to Solve Problem (9)

Problem (9) can be solved via successive convex approximation (SCA) [43] or inner approximation (IA) [44], either of which has been widely developed for wireless network optimization [45]–[47]. The algorithm designed in this paper follows the IA approach. To solve problem (9), we introduce $2U$ auxiliary variables $\{p_u, q_u\}$ and equivalently write problem (9) into

$$\begin{aligned} \max_{\{\mathbf{v}_u, p_u, q_u\}} \ & \sum_{u\in\mathcal{U}} \log(1+p_u) \\ \text{s.t.} \quad & \sigma^2 + \sum_{v\neq u} \left|\mathbf{h}_u^{\mathrm{H}}\mathbf{B}\mathbf{v}_v\right|^2 \le q_u \\ & \frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{B}\mathbf{v}_u\right|^2}{q_u} \ge p_u, \quad \sum_{u\in\mathcal{U}} \|\mathbf{F}\mathbf{B}\mathbf{v}_u\|_{\mathrm{F}}^2 \le P_{\mathrm{m}}. \end{aligned} \tag{22}$$

An iterative algorithm can be designed to solve the problem in (22). Specifically, in the $(n+1)$-th iteration, we shall solve the following convex optimization problem

$$\begin{aligned} \max_{\{\mathbf{v}_u, p_u, q_u\}} \ & \sum_{u\in\mathcal{U}} \log(1+p_u) \\ \text{s.t.} \quad & \sigma^2 + \sum_{v\neq u} \left|\mathbf{h}_u^{\mathrm{H}}\mathbf{B}\mathbf{v}_v\right|^2 \le q_u, (u\in\mathcal{U}) \\ & 2\mathrm{Re}\big(\mathbf{v}_{u,n}^{\mathrm{H}}\mathbf{B}^{\mathrm{H}}\mathbf{h}_u\mathbf{h}_u^{\mathrm{H}}\mathbf{B}\mathbf{v}_u\big)q_{u,n}^{-1} - \\ & \qquad q_{u,n}^{-2}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{B}\mathbf{v}_{u,n}\right|^2 q_u \ge p_u \\ & \sum_{u\in\mathcal{U}} \|\mathbf{F}\mathbf{B}\mathbf{v}_u\|_{\mathrm{F}}^2 \le P_{\mathrm{m}}, \end{aligned} \tag{23}$$

where $\mathrm{Re}(\cdot)$ represents the real part of a complex, $\mathbf{v}_{u,n}$ is the n-th iteration of $\mathbf{v}_u$, and the notations are defined similarly for other variables.

For clarity, the algorithm to solve problem (9) is summarized in Algorithm 3. Note that each constraint $\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{B}\mathbf{v}_u\right|^2/q_u \ge p_u$ in (22) has been replaced by its first order approximation in (23), which satisfies the properties required by IA for convergence. Hence, Theorem 1 in [44] asserts that the limit point of the sequence generated by Algorithm 3 is a stationary point.

Algorithm 3: Reward Function Solving (MU Case)

1: **input:** equivalent channel vectors of U MUs
2: **initialize:** digital precoding vectors $\{\mathbf{v}_u\}$ randomly
3: **repeat**
(1) construct and solve problem (23)
(2) check whether convergence criterion is met
until convergence criterion is satisfied
4: **output:** digital precoding vectors $\{\mathbf{v}_u^*\}$.

Appendix C
Algorithm to Solve Problem (13)

Note that the problem in (13) is non-convex and an iterative algorithm can be designed to solve this problem. For the first step, we equivalently write problem (13) as

$$\begin{aligned} \min_{\{\mathbf{v}_b\}, q} \ & -q \\ \text{s.t.} \quad & \left|\sum_{b=1}^{V} \mathbf{h}_b^{\mathrm{H}}\mathbf{v}_b\right|^2 \ge q, \quad \|\mathbf{A}_b\mathbf{v}_b\|_{\mathrm{F}}^2 \le P_b. \end{aligned} \tag{24}$$

For convenience, let $\mathbf{H} = \mathrm{diag}(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_V)$ and $\mathbf{V}^{\mathrm{H}} = [\mathbf{v}_1^{\mathrm{H}}, \mathbf{v}_2^{\mathrm{H}}, \cdots, \mathbf{v}_V^{\mathrm{H}}]$. Then, problem (24) can be rewritten as

$$\begin{aligned} \min_{\{\mathbf{v}_b\}, q} \ & -q \\ \text{s.t.} \quad & \mathbf{V}^{\mathrm{H}}\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{V} \ge q, \quad \|\mathbf{A}_b\mathbf{v}_b\|_{\mathrm{F}}^2 \le P_b. \end{aligned} \tag{25}$$

To seek a stationary point of problem (25), we resort to the successive convex approximation (SCA). Let $\mathbf{V}_n$ denote the n-th iteration of $\mathbf{V}$. In the $(n+1)$-th iteration, we need to solve the following problem

$$\begin{aligned} \min_{\{\mathbf{v}_b\}, q} \ & -q \\ \text{s.t.} \quad & 2\mathrm{Re}(\mathbf{V}_n^{\mathrm{H}}\mathbf{H}\mathbf{H}^{\mathrm{H}}(\mathbf{V}-\mathbf{V}_n)) + \mathbf{V}_n^{\mathrm{H}}\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{V}_n \ge q \\ & \|\mathbf{A}_b\mathbf{v}_b\|_{\mathrm{F}}^2 \le P_b. \end{aligned} \tag{26}$$

Note that problem (26) is convex, which can be efficiently solved. For clarity, the algorithm to solve problem (13) is summarized in Algorithm 4.

Algorithm 4: Reward Function Solving (UCCC Case)

1: **input:** equivalent channel vectors of V BSs
2: **initialize:** digital precoding vectors $\{\mathbf{v}_b\}$ randomly
3: **repeat**
(1) construct and solve problem (26)
(2) check whether convergence criterion is met
until convergence criterion is satisfied
4: **output:** digital precoding vectors $\{\mathbf{v}_b^*\}$.

References

[1] M. Xiao, S. Mumtaz, Y. Huang, L. Dai, Y. Li, M. Matthaiou, G. K. Karagiannidis, E. Björnson, K. Yang, C. L. I, and A. Ghosh, "Millimeter wave communications for future mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1909–1935, Sept 2017.

[2] R. Baldemair, T. Irnich, K. Balachandran, E. Dahlman, G. Mildh, Y. Seln, S. Parkvall, M. Meyer, and A. Osseiran, "Ultra-dense networks in millimeter-wave frequencies," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 202–208, January 2015.

[3] Y. Huang, C. Xu, C. Zhang, M. Hua, and Z. Zhang, "An overview of intelligent wireless communications using deep reinforcement learning," *Journal of Communications and Information Networks*, vol. 4, no. 2, pp. 15–29, 2019.

[4] C. Han, J. Wang, J. Wang, and L. Bai, "Hybrid beamforming design for uplink mmwave systems with a predefined low-resolution codebook," *Journal of Communications and Information Networks*, vol. 4, no. 3, pp. 1–8, 2019.

[5] S. Gao, X. Cheng, and L. Yang, "Estimating doubly-selective channels for hybrid mmwave massive mimo systems: A doubly-sparse approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5703–5715, 2020.

[6] X. Gao, L. Dai, Z. Chen, Z. Wang, and Z. Zhang, "Near-optimal beam selection for beamspace mmwave massive mimo systems," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1054–1057, May 2016.

[7] Z. Xiao, T. He, P. Xia, and X. G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.

[8] J. Zhang, Y. Huang, Q. Shi, J. Wang, and L. Yang, "Codebook design for beam alignment in millimeter wave communication systems," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4980–4995, Nov 2017.

[9] S. Hur, T. Kim, D. Love, J. Krogmeier, T. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4391–4403, October 2013.

[10] J. Singh and S. Ramakrishna, "On the feasibility of codebook-based beamforming in millimeter wave systems with multiple antenna arrays," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2670–2683, May 2015.

[11] D. Zhang, A. Li, M. Shirvanimoghaddam, P. Cheng, Y. Li, and B. Vucetic, "Codebook-based training beam sequence design for millimeter-wave tracking systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5333–5349, Nov 2019.

[12] D. Zhang, A. Li, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Exploring aoa/aod dynamics in beam alignment of mobile millimeter wave mimo systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6172–6176, June 2019.

[13] M. Gao, B. Ai, Y. Niu, Z. Zhong, Y. Liu, G. Ma, Z. Zhang, and D. Li, "Dynamic mmwave beam tracking for high speed railway communications," in *2018 IEEE WCNCW*, April 2018, pp. 278–283.

[14] D. Zhang, H. Chen, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Training beam sequence optimization for millimeter wave mimo tracking systems," in *2018 IEEE ICC*, May 2018, pp. 1–6.

[15] V. Va, H. Vikalo, and R. W. Heath, "Beam tracking for mobile millimeter wave communication systems," in *2016 IEEE GlobalSIP*, Dec 2016, pp. 743–747.

[16] N. Michelusi and M. Hussain, "Optimal beam-sweeping and communication in mobile millimeter-wave networks," in *2018 IEEE ICC*, May 2018, pp. 1–6.

[17] M. Scalabrin, N. Michelusi, and M. Rossi, "Beam training and data transmission optimization in millimeter-wave vehicular networks," in *2018 IEEE GLOBECOM*, Dec 2018, pp. 1–7.

[18] J. Zhao, J. Liu, Y. Nie, and S. Ni, "Location-assisted beam alignment for train-to-train communication in urban rail transit system," *IEEE Access*, vol. 7, pp. 80 133–80 145, 2019.

[19] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[20] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, "Inverse multipath fingerprinting for millimeter wave v2i beam alignment," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4042–4058, May 2018.

[21] J. C. Aviles and A. Kouki, "Position-aided mm-wave beam training under nlos conditions," *IEEE Access*, vol. 4, pp. 8703–8714, 2016.

[22] Z. Wei, Y. Zhao, X. Liu, and Z. Feng, "Doa-lf: A location fingerprint positioning algorithm with millimeter-wave," *IEEE Access*, vol. 5, pp. 22 678–22 688, 2017.

[23] K. Satyanarayana, M. El-Hajjar, A. A. M. Mourad, and L. Hanzo, "Deep learning aided fingerprint-based beam alignment for mmwave vehicular communication," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10 858–10 871, Nov 2019.

[24] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Online learning for position-aided millimeter wave beam training," *IEEE Access*, vol. 7, pp. 30 507–30 526, 2019.

[25] M. Hashemi, A. Sabharwal, C. E. Koksal, and N. B. Shroff, "Efficient beam alignment in millimeter wave systems using contextual bandits," in *IEEE INFOCOM 2018*, April 2018, pp. 2393–2401.

[26] M. Cheng, J. Wang, J. Wang, M. Lin, Y. Wu, and H. Zhu, "A fast beam searching scheme in mmwave communications for high-speed trains," in *2019 IEEE ICC*, May 2019, pp. 1–6.

[27] J. Zhang, Y. Huang, J. Wang, and X. You, "Intelligent beam training for millimeter-wave communications via deep reinforcement learning," in *2019 IEEE GLOBECOM*, Dec 2019, pp. 1–7.

[28] S. He, J. Wang, Y. Huang, B. Ottersten, and W. Hong, "Codebook-based hybrid precoding for millimeter wave multiuser systems," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5289–5304, Oct 2017.

[29] O. El Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. Heath, "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499–1513, March 2014.

[30] A. Alkhateeb, G. Leus, and R. Heath, "Limited feedback hybrid precoding for multi-user millimeter wave systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6481–6494, Nov 2015.

[31] T. E. Bogale, L. B. Le, and X. Wang, "Hybrid analog-digital channel estimation and beamforming: Training-throughput tradeoff," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 5235–5249, Dec 2015.

[32] X. Yu, J. Shen, J. Zhang, and K. B. Letaief, "Alternating minimization algorithms for hybrid precoding in millimeter wave mimo systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 485–500, April 2016.

[33] Z. Xu and J. Sun, "Model-driven deep-learning," *National Science Review*, vol. 5, no. 1, pp. 22–24, 2018.

[34] H. He, S. Jin, C. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 77–83, October 2019.

[35] R. S. Sutton, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2018.

[36] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-Based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, 2019.

[37] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 60–69, 2019.

[38] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2015.

[39] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," 2017.

[40] Z. Yang, Y. Xie, and Z. Wang, "A theoretical analysis of deep q-learning," *CoRR*, vol. abs/1901.00137, 2019. [Online]. Available: http://arxiv.org/abs/1901.00137

[41] J. Zhang, Y. Huang, Y. Zhou, and X. You, "Beam alignment and tracking for millimeter wave communications via bandit learning," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5519–5533, 2020.

[42] M. Volodymyr, K. Koray, S. David, and et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb 2015.

[43] G. Venkatraman, A. Tölli, M. Juntti, and L. Tran, "Multigroup multicast beamformer design for MISO-OFDM with antenna selection," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5832–5847, 2017.

[44] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, 1978.

[45] D. Nguyen, L. Tran, P. Pirinen, and M. Latva-aho, "Precoding for full duplex multiuser mimo systems: Spectral and energy efficiency maximization," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4038–4050, 2013.

[46] A. Zappone, E. Björnson, L. Sanguinetti, and E. Jorswieck, "Globally optimal energy-efficient power control and receiver design in wireless networks," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2844–2859, 2017.

[47] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization - Part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, 2017.

Jianjun Zhang (S'16-M'18) received the M.S. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014, and the Ph.D. degree from Southeast University, Nanjing, China, in 2018. He is currently a Research Fellow of the electrical and electronics engineering with University College London (UCL), U.K. He was the recipient of the Best Paper Award in the IEEE Globecom 2019. His current research interests include machine learning and optimization, intelligent communications, and probability theory and its applications.

Yongming Huang (M'10-SM'17) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, in 2007.

Since March 2007 he has been a faculty in the School of Information Science and Engineering, Southeast University, China, where he is currently a full professor. During 2008-2009, Dr. Huang was visiting the Signal Processing Lab, Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden. His current research interests include intelliegent 5G/6G mobile communcations and millimeter wave wireless communications. He has published over 200 peer-reviewed papers, hold over 80 invention patents. He submitted around 20 technical contributions to IEEE standards, and was awarded a certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj.

He has served as an Associate Editor for the IEEE Transactions on Signal Processing, and a Guest Editor for the IEEE Journal Selected Areas in Communications, and is serving as an Editor-at-Large for the IEEE Open Journal of the Communications Society, and an Associate Editor for the IEEE Wireless Communications Letters.

Jiaheng Wang (M'10-SM'14) received the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2010, and the B.E. and M.S. degrees from the Southeast University, Nanjing, China, in 2001 and 2006, respectively.

He is currently a Full Professor at the National Mobile Communications Research Laboratory (N-CRL), Southeast University, Nanjing, China. From 2010 to 2011, he was with the Signal Processing Laboratory, KTH Royal Institute of Technology, Stockholm, Sweden. He also held visiting positions at the Friedrich Alexander University Erlangen-Nrnberg, Nürnberg, Germany, and the University of Macau, Macau. His research interests include optimization in signal processing and wireless communications.

Dr. Wang has published more than 130 articles on international journals and conferences. From 2014 to 2018, he served as an Associate Editor for the IEEE Signal Processing Letters. From 2018, he serves as a Senior Area Editor for the IEEE Signal Processing Letters. He was a recipient of the Humboldt Fellowship for Experienced Researchers and the best paper awards of IEEE GLOBECOM 2019, ADHOCNETS 2019, and WCSP 2014.

Xiaohu You (M'89-SM'11-F'12) received B.S., M.S. and Ph.D. degrees in electrical engineering from Nanjing Institute of Technology, Nanjing, China, in 1982, 1985, and 1989, respectively. From 1987 to 1989, he was with Nanjing Institute of Technology as a Lecturer. From 1990 to the present time, he has been with Southeast University, first as an Associate Professor and later as a Professor. His research interests include mobile communications, adaptive signal processing, and artificial neural networks with applications to communications and biomedical engineering. He has contributed over 40 IEEE journal papers and 2 books in the areas of adaptive signal processing, neural networks and their applications to communication systems. He was the Premier Foundation Investigator of the China National Science Foundation. From 1999 to 2002, he was the Principal Expert of the C3G Project, responsible for organizing China's 3G Mobile Communications R&D Activities. From 2001-2006, he was the Principal Expert of the national 863 FuTURE Project. He received the excellent paper award from the China Institute of Communications in 1987 and Elite Outstanding Young Teacher Awards from Southeast University in 1990, 1991, and 1993. He was a recipient of the 1989 Young Teacher Award of Fok Ying Tung Education Foundation, State Education Commission of China. Dr. You now is the Chairman of IEEE Nanjing Section. He was selected as IEEE Fellow for his contributions to development of mobile communications in China in 2012.

Christos Masouros (M'06-SM'14) received the Diploma degree in Electrical and Computer Engineering from the University of Patras, Greece, in 2004, and MSc by research and PhD in Electrical and Electronic Engineering from the University of Manchester, UK in 2006 and 2009 respectively. In 2008 he was a research intern at Philips Research Labs, UK. Between 2009-2010 he was a Research Associate in the University of Manchester and between 2010-2012 a Research Fellow in Queen's University Belfast. In 2012 he joined University College London as a Lecturer. He has held a Royal Academy of Engineering Research Fellowship between 2011-2016.

He is currently a Full Professor in the Information and Communication Engineering research group, Department of Electrical and Electronic Engineering, and affiliated with the Institute for Communications and Connected Systems, University College London. His research interests lie in the field of wireless communications and signal processing with particular focus on Green Communications, Large Scale Antenna Systems, Communications and Radar Co-existence, interference mitigation techniques for MIMO and multi-carrier communications. He was the recipient of the Best Paper Awards in the IEEE Globecom 2015 and IEEE WCNC 2019 conferences, and has been recognized as an Exemplary Editor for the IEEE Communications Letters, and as an Exemplary Reviewer for the IEEE Transactions on Communications. He is an Editor for IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, the IEEE Open Journal of Signal Processing, and Editor-at-Large for IEEE Open Journal of the Communications Society. He has been an Associate Editor for IEEE Communications Letters, and a Guest Editor for IEEE Journal on Selected Topics in Signal Processing issues "Exploiting Interference towards Energy Efficient and Secure Wireless Communications", "Hybrid Analog / Digital Signal Processing for Hardware-Efficient Large Scale Antenna Arrays" and "Joint Communication and Radar Sensing for Emerging Applications". He is currently an elected member of the EURASIP SAT Committee on Signal Processing for Communications and Networking.