Learning-Based Adaptive IRS Control with Limited Feedback Codebooks

Junghoon Kim, *Student Member, IEEE,* Seyyedali Hosseinalipour, *Member, IEEE,*
Andrew C. Marcum, *Member, IEEE,* Taejoon Kim, *Senior Member, IEEE,*
David J. Love, *Fellow, IEEE,* and Christopher G. Brinton, *Senior Member, IEEE*

***Abstract*—Intelligent reflecting surfaces (IRS) consist of configurable meta-atoms, which can change the wireless propagation environment through design of their reflection coefficients. We consider a practical setting where (i) the IRS reflection coefficients are configured by adjusting *tunable elements* embedded in the meta-atoms, (ii) the IRS reflection coefficients are affected by the *incident angles* of the incoming signals, (iii) the IRS is deployed in multi-path, time-varying channels, and (iv) the feedback link from the base station to the IRS has a low data rate. Conventional optimization-based IRS control protocols, which rely on channel estimation and conveying the optimized variables to the IRS, are not applicable in this setting due to the difficulty of channel estimation and the low feedback rate. Therefore, we develop a novel adaptive codebook-based limited feedback protocol where only a codeword index is transferred to the IRS. We propose two solutions for adaptive codebook design, *random adjacency (RA)* and *deep neural network policy-based IRS control (DPIC)*, both of which only require the end-to-end compound channels. We further develop several augmented schemes based on RA and DPIC. Numerical evaluations show that the data rate and average data rate over one coherence time are improved substantially by our schemes.**

***Index Terms*—Intelligent reflecting surface (IRS), reconfigurable intelligent surface (RIS), software-controlled meta-surface, limited feedback, adaptive codebook, deep reinforcement learning**

I. Introduction

The intelligent reflecting surface (IRS) is one of the innovative technologies being discussed for 6G-and-beyond [1], [2]. Other terminologies have been employed in literature, such as reconfigurable intelligent surface (RIS) [3] and software-controlled meta-surface [4], to represent such tunable surfaces. An IRS is composed of configurable meta-atoms with tunable reflection behavior. By fine-tuning the meta-atoms, an IRS can reflect the incident signals to a desired direction to enhance the communication system performance in terms of power savings, throughput, and other metrics. Compared to a traditional antenna array with one or more radio frequency (RF) chains for active relaying/beamforming, the IRS is built of low cost meta-surfaces with low energy consumption requirements for adaptive tuning [4]. These benefits have motivated active research on utilizing an IRS in communications/signal processing literature [4], [5].

Most communication and signal processing works have focused on designing the *reflection coefficients* of the IRS meta-atoms considering different performance metrics of interest [3], [6], [7], e.g., sum-rate, power saving, secrecy rate, etc. However, these works have not given consideration to the practical reflection behavior of the IRS meta-atoms or the nature of the channels under which the IRS is deployed. In this paper, we focus on the problem of controlling the IRS adaptively considering the practical reflection behavior and realistic channel environment. We take the first step towards this direction by focusing on a *point-to-point communication model*, commonly used in the IRS literature, e.g., [3], [6], [7], which we follow to construct a fundamental system model encompassing practical considerations.

A. Related Work and Shortcomings of Current Methods

1) Practical Reflection Behavior of IRS: Much of the prior works on IRS reflection coefficient design for communications have focused on controlling either (i) only the phase shift with full/lossless signal reflection (i.e., assuming no signal attenuation upon reflecting from the IRS), or (ii) both the phase shift and attenuation of reflection, which are independently controlled from one other. However, it is practically difficult to implement either of these approaches. First, the full/lossless signal reflection cannot be realized in practice due to the inevitable energy loss caused by the dielectric loss, metallic loss, and ohmic loss [8]. Second, the reflection phase shift and attenuation cannot be controlled independently because the reflection behavior is determined by adjusting the *tunable elements* inside the meta-atoms. This fact implies that the IRS reflection phase shift and attenuation are *interdependent* as revealed in the physics literature [9], [10]. This interdependency has only been considered in a few works in the communications area [11], [12].

Another aspect overlooked in prior works is the dependency between the IRS reflection behavior and the *incident angles* of the incoming electromagnetic (EM) waves. This fact was first revealed in recent work [13], which demonstrates that the IRS reflection coefficient is sensitive to the incident angles

An abridged version of this work appeared in the 2022 IEEE International Conference on Communications (ICC). J. Kim, C. G. Brinton, and D. J. Love were supported in part by the National Spectrum Consortium (NSC) under Grant W15QKN-15-9-1004 and Office of Naval Research (ONR) under Grant N00014-21-1-2472. T. Kim was supported in part by the National Science Foundation (NSF) under Grants CNS1955561.

J. Kim, S. Hosseinalipour, D. J. Love, and C. G. Brinton are with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907 USA (e-mail: kim3220@purdue.edu; hosseina@purdue.edu; djlove@purdue.edu; cgb@purdue.edu).

A. C. Marcum is with Raytheon BBN Technologies, Cambridge, MA, 02138 USA (email: andrew.marcum@raytheon.com).

T. Kim is with the Department of Electrical Engineering and Computer Science, University of Kansas, KS, 66045 USA (email: taejoonkim@ku.edu).

of the incoming EM waves. Motivated by this observation, the authors of [14] propose an angle-dependent reflection coefficient model for each IRS meta-atom using an equivalent circuit model. In parallel, the authors in [15], [16] also demonstrate the reflection response varies with the incident angle of the EM wave. To the best of our knowledge, the angle-dependent property of the IRS reflection coefficient has not been incorporated into uplink/downlink signal transmission models for wireless communication systems.

2) Communication Overhead for IRS Control under Realistic Channel Environment: To develop solutions for IRS reflection design, existing works either assume perfect knowledge of the channel state information (CSI) [3], [6], [7], [11] or estimate the CSI before IRS reflection design [17], [18]. In both cases, for adaptive IRS control under time-varying channels, a successive channel estimation at the base station (BS) and feedback of information from the BS to the IRS should be conducted. This successive procedure incurs communication time overhead. The work [19] takes into account the communication time overhead required for channel estimation and feedback for the IRS phase shift design and shows that the *average* data rate over a channel coherence time is decreased by the overhead. Nevertheless, in [19], the practical IRS reflection behavior and successive IRS control under time-varying channels have not been considered.

To reduce the overhead for IRS control, some recent works consider a low overhead *feedback* link from the BS to the IRS by either employing codebook structures [15], [20]–[22] or one-bit feedback [23]. The feedback link typically has a low data rate because the CSI of the feedback link is unknown at the BS [5]. In general, codebooks are known to provide high performance gains in limited feedback systems [24], and thus are widely used in wireless communications standards, such as Long-Term Evolution (LTE), LTE-Adv, LTE-Adv Pro, and 5G New Radio (NR) [25]. In IRS-assisted communications, a codebook refers to a set of IRS reflection coefficients, which are shared at both the BS and the IRS [15], [20]–[22]. The work [20] considers the codebook construction for uniform linear arrays (ULA). In [15] and [21], the codebook is constructed via discrete Fourier transform (DFT) quantization [26] and random vector quantization (RVQ) [27], [28], respectively. In [22], the codebook is designed based on RVQ and maximizing the Euclidean distance among the codewords. In these works, the BS feeds back a specific codeword index to the IRS, which the IRS uses to recover the desired reflection coefficients from the codebook. The work [23] adapts the random perturbation-based method with one-bit feedback for IRS control, previously proposed in traditional wireless communications [29]. All of these works, however, directly design the IRS *reflection coefficients* without considering the practical IRS reflection behavior. Furthermore, the codebook approaches [15], [20]–[22] have not considered an adaptive design of the codebook for time-varying channels.

B. Our Methodology and Summary of Contributions

In this paper, we consider adaptive IRS control in the practical setting where (i) the IRS reflection coefficients are achieved by adjusting tunable elements embedded in the meta-atoms, i.e., their controllable capacitance, (ii) the IRS reflection coefficients are affected by the incident angles of the incoming EM wave, (iii) the IRS is deployed in an environment with multi-path, time-varying channels, and (iv) the feedback link from the BS to the IRS has a low data rate.

The joint consideration of the practical IRS reflection behavior and realistic channel environment makes the contemporary optimization-based methods used for IRS control [3], [6], [7], [11], [17], [19], which rely on channel estimation, inefficient. This is because channel estimation in turn requires known IRS reflection coefficients, which cannot be obtained in a real-world system since (i) incoming signals in a multi-path channel have different angles of arrival to the IRS, and thus experience different reflection responses caused by angle-dependent reflection behavior of the meta-atoms, and (ii) it is difficult to measure the incident angles of multiple incoming signals at the IRS since the IRS is typically a passive device without active sensors.

For effective IRS control in this practical setting, we propose a novel *adaptive codebook-based limited feedback protocol*. There are several novelties in our proposed protocol. First, we directly design the meta-atom *capacitance values* for IRS configuration, different from the current methods that design IRS reflection coefficients, some of which may be not feasible for implementation. Second, we adopt a codebook structure, where the codebook is a set of capacitance values for IRS configuration and employed at the IRS. Third, we develop two *adaptive* codebook design methods, where the codebook is updated to account for time-varying channels. These methods are (i) *random adjacency (RA)*, which utilizes the correlation across the channel instances, and (ii) *deep neural network (DNN) policy-based IRS control (DPIC)*, which is a deep reinforcement learning-based method. Both of these approaches only require the end-to-end compound channels from the user equipment (UE) to the BS (in (7)), which can be readily obtained at the BS. Thus our IRS control methodology does not require any expensive estimation/tracking processes for the channels, UE location, and incident angles.

The contributions of this paper are summarized as follows:

- We introduce a novel signal model that considers the practical IRS reflection behavior for the IRS-assisted uplink communications system, where the reflection coefficient of each IRS meta-atom is a function of the incident angle of the EM wave and its controllable capacitance.
- We formulate the data rate maximization problem and discuss the challenges associated with solving the problem in the practical setting. Motivated by the existence of a low-rate feedback channel between the IRS and BS and the requirement of successive IRS control under time-varying channels, we propose a novel adaptive codebook-based limited feedback protocol.
- We propose two algorithms for adaptive codebook design: RA and DPIC. In DPIC, we tailor an actor-critic network for the DNN policy learning to make it compatible with the limited feedback protocol. We incorporate the RVQ process into the behavior policy, which allows low-overhead feedback. Further, we develop several augmented strategies based on DPIC, which incorporate multi-agent learning and

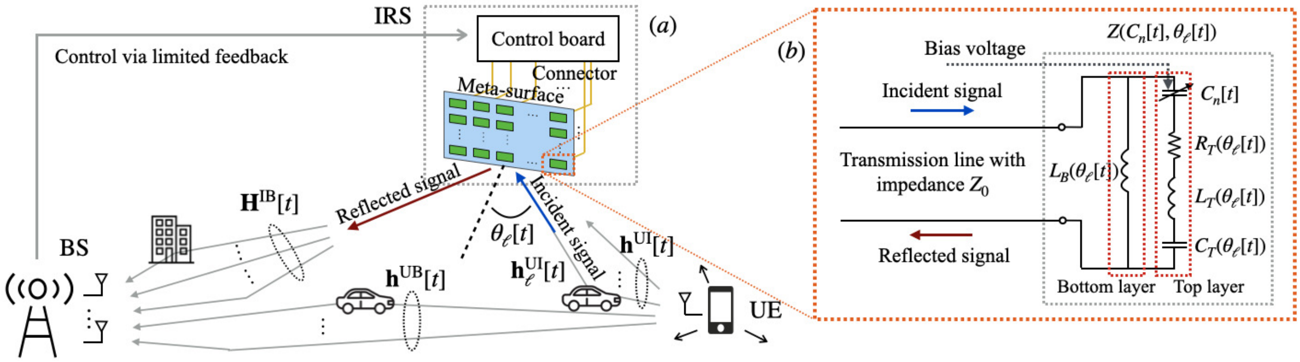


Fig. 1: The system model consisting of a UE, IRS, and BS in uplink point-to-point communication, where the IRS is controlled by the BS via a limited feedback link. (a) Depiction of the IRS as two interconnected systems: meta-surface and control board. (b) Equivalent circuit model of the signal response at each IRS meta-atom.

a hybrid of the RA and DPIC approaches. We analyze the computational complexity and the total time overhead of the proposed approaches.

- For simulations, we consider two practical scenarios in multi-path fading channels: (i) indoor UE with no line-of-sight (LoS) link to the IRS, and (ii) outdoor UE with a LoS link. For both scenarios, we evaluate the performances of the data rate and average data rate over one coherence time, and demonstrate that RA and DPIC outperform the baseline. Our simulations also show the superior performance of our augmented strategies compared to their counterparts.

II. System Model for IRS-assisted Uplink Communications

We begin by formalizing IRS meta-atom reflection behavior in Sec. II-A. Then, we describe the signal model of IRS-assisted uplink communications in Sec. II-B.

A. Reflection Behavior of IRS Meta-atoms

An IRS consists of two interconnected systems shown in Fig. 1(a): a meta-surface and control board. A meta-surface is an ultra-thin sheet composed of periodic sub-wavelength metal/dielectric structures, i.e., meta-atoms. The size of each meta-atom is typically from $\lambda/10$ to $\lambda/2$ [4], where λ denotes the wavelength of the EM wave. Each meta-atom generally contains a semiconductor device as a tunable element, e.g., positive-intrinsic-negative (PIN) diode, variable capacitor (varactor), metal-oxide-semiconductor field-effect transistor (MOSFET) [10]. By adjusting the bias voltages applied to these tunable elements, we can change the impedances over the meta-surface to have a desired functionality, e.g., perfect absorption, anomalous reflection, and polarization of the incoming signal. We focus on the *reflection* behavior, where the reflected signals from the meta-surface become constructive at a desired angle/direction, i.e., beamforming.

A control board that is connected to the meta-surface enables flexible configuration of the tunable elements. A field programmable gate array (FPGA)-based control board is generally considered for IRS control due to the flexible implementation of different logic functions [30]. A control board adjusts the bias voltage applied to the semiconductor in each meta-atom and changes the capacitance of the semiconductor, i.e., the tunable element. To adapt to dynamic channels, the control board can flexibly tune the capacitance over time. Given a range of potential bias voltage values, the capacitance $C_n[t]$ at meta-atom n at time t satisfies

$$C_{\min} \leq C_n[t] \leq C_{\max}, \tag{1}$$

where $C_{\min}$ and $C_{\max}$ may vary for different types of semiconductor devices.

Through tuning the capacitance of the meta-atoms, their impedance can be adjusted. However, the impedance is also dependent on the incident angle of the incoming EM wave [16]. Both of these factors should be considered in IRS reflection behavior design. To explicitly describe the reflection behavior of the meta-surface, we will next investigate the impedance and reflection coefficient at the meta-atom level.[1] As an example, we provide the impedance and reflection coefficient of a meta-atom equipped with a *varactor* using its equivalent circuit model depicted in Fig. 1(b). Denote $\theta_\ell[t]$ as the incident angle of the ℓ-th channel path to the IRS.[2] Under a far-field assumption where $\theta_\ell[t]$ is the same across all the meta-atoms, the work [14] has verified experimentally that the impedance of meta-atom n in this equivalent circuit can be described as

$$Z(C_n[t], \theta_\ell[t]) = \frac{j\omega L_B(\theta_\ell[t])\big(R_T(\theta_\ell[t]) + j\omega L_T(\theta_\ell[t]) + \frac{1}{j\omega C_T(\theta_\ell[t])} + \frac{1}{j\omega C_n[t]}\big)}{j\omega L_B(\theta_\ell[t]) + \big(R_T(\theta_\ell[t]) + j\omega L_T(\theta_\ell[t]) + \frac{1}{j\omega C_T(\theta_\ell[t])} + \frac{1}{j\omega C_n[t]}\big)}, \tag{2}$$

where $L_T(\theta_\ell[t])$, $C_T(\theta_\ell[t])$, and $R_T(\theta_\ell[t])$ are the inductance, capacitance, and loss resistance of the top layer, respectively, $L_B(\theta_\ell[t])$ is the bottom layer inductance, $C_n[t]$ is the variable capacitance, and $\omega = 2\pi f$ is the angular frequency of the incident EM waves. Except $C_n[t]$, all of these parameters are dependent on the incident angle $\theta_\ell[t]$, which makes the reflection behavior of meta-atom angle-dependent. This phenomenon is also observed in [13], [15], [16].

[1] Since the physical size of the meta-atom is usually smaller than the wavelength of the incident signal, the signal response of the meta-atom can be described by an equivalent circuit model [31].

[2] In this section, we discuss the angle-dependent impedance (in (2)) and reflection coefficient model (in (3)) provided in [14] where only azimuth coordinates of the incident angle are considered. Our proposed signal model and methodologies can be readily extended to the case including the elevation angle.

Considering the impedance discontinuity between the free space impedance $Z_0 \approx 376.73\,\Omega$ and the meta-atom impedance $Z(C_n[t], \theta_\ell[t])$, the reflection coefficient[3] of meta-atom n is [32]

$$\Gamma(C_n[t], \theta_\ell[t]) = \frac{Z(C_n[t], \theta_\ell[t]) - Z_0}{Z(C_n[t], \theta_\ell[t]) + Z_0}. \tag{3}$$

The expressions in (2)&(3) reveal two practical considerations for tuning the meta-atoms.

- ***Consideration 1.*** *Dependency between amplitude/attenuation and phase shift.* The amplitude/attenuation $|\Gamma(C_n[t], \theta_\ell[t])|$ and phase shift $\angle\Gamma(C_n[t], \theta_\ell[t])$ of the reflection are jointly controlled by the semiconductor with capacitance $C_n[t]$. In other words, the amplitude and the phase shift at a meta-atom cannot be controlled independently, which is also reported in [11]. Thus, it is beneficial to design the variable capacitance instead of the reflection coefficient since some combinations of attenuation and phase shifts may not be feasible.
- ***Consideration 2.*** *Dependency between reflection coefficient and incident angle.* The reflection coefficient is a function of the incident angle $\theta_\ell[t]$. This will pose new challenges for applications of IRS in practical wireless systems with multi-path and time-varying channels, which will be discussed in detail in Sec. III-A. This dependency is observed and explained in [13]–[16], but not yet incorporated in the canonical signal model for IRS-assisted communications.

We incorporate the above practical considerations into our signal model and methodology.

B. Signal Model for IRS-assisted Uplink Communications

We consider the IRS-assisted uplink communications with a UE, a BS, and an IRS, depicted in Fig. 1. The UE is equipped with a single antenna, while the BS possesses N_{BS} antennas. We assume a block fading channel model with time index $t = 0, 1, ...,$ where channels are constant during each block. Let N_{IRS} denote the number of the IRS meta-atoms. We define the *capacitance vector* across the IRS meta-atoms at time t as

$$\mathbf{c}[t] = \big[C_1[t], ..., C_{N_{\text{IRS}}}[t]\big] \in \mathbb{R}^{N_{\text{IRS}}}, \tag{4}$$

where $C_n[t]$ is the capacitance of the semiconductor in meta-atom n. We also formulate the *reflection coefficient matrix* across the IRS meta-atoms, $\mathbf{\Phi}(\mathbf{c}[t], \theta_\ell[t]) \in \mathbb{C}^{N_{\text{IRS}} \times N_{\text{IRS}}}$, as

$$\mathbf{\Phi}(\mathbf{c}[t], \theta_\ell[t]) = \operatorname{diag}\big(\Gamma(C_1[t], \theta_\ell[t]), ..., \Gamma(C_{N_{\text{IRS}}}[t], \theta_\ell[t])\big), \tag{5}$$

where the n-th diagonal entry $\Gamma(C_n[t], \theta_\ell[t])$ is the reflection coefficient at meta-atom $n \in \{1, ..., N_{\text{IRS}}\}$ given the incident angle $\theta_\ell[t]$. The reflection coefficient matrix enables us to incorporate the practical IRS reflection behavior into the signal model for IRS-assisted communications.

We consider multi-path channels and adopt a geometric channel model representation [33]. We represent the channel from the UE to the IRS (i.e., UE-IRS channel) as $\mathbf{h}^{\text{UI}}[t] = \sum_{\ell=1}^{L[t]} \mathbf{h}_\ell^{\text{UI}}(\theta_\ell[t], t) \in \mathbb{C}^{N_{\text{IRS}} \times 1}$, in which $\mathbf{h}_\ell^{\text{UI}}(\theta_\ell[t], t)$ is the ℓ-th path channel with the incident angle $\theta_\ell[t]$ and $L[t]$ is the number of paths. We assume a narrowband system, where $\theta_\ell[t]$, $\forall \ell$, is the same across the utilized frequency band and consider a single tap channel model. Subsequently, the received signal at the BS at time t is given by

$$\mathbf{y}[t] = \left(\mathbf{h}^{\text{UB}}[t] + \mathbf{H}^{\text{IB}}[t] \sum_{\ell=1}^{L[t]} \mathbf{\Phi}(\mathbf{c}[t], \theta_\ell[t]) \mathbf{h}_\ell^{\text{UI}}(\theta_\ell[t], t)\right) \times \sqrt{P} x[t] + \mathbf{n}[t] \in \mathbb{C}^{N_{\text{BS}} \times 1}, \tag{6}$$

where $P \in \mathbb{R}^+$ denotes the transmit power and $x[t] \in \mathbb{C}$ denotes the transmit symbol of the UE, where $\mathbb{E}[|x[t]|^2] = 1$. The noise vector $\mathbf{n}[t] \in \mathbb{C}^{N_{\text{BS}} \times 1}$ follows the complex Gaussian distribution $\mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I})$, where $\mathbf{I}$ denotes the identity matrix and σ^2 is the noise variance. $\mathbf{h}^{\text{UB}}[t] \in \mathbb{C}^{N_{\text{BS}} \times 1}$ is the direct channel from the UE to the BS (i.e., UE-BS channel) and $\mathbf{H}^{\text{IB}}[t] \in \mathbb{C}^{N_{\text{BS}} \times N_{\text{IRS}}}$ is the channel from the IRS to the BS (i.e., IRS-BS channel). We define the end-to-end compound channel in (6) as the *effective channel* $\mathbf{h}_{\text{eff}}(\mathbf{c}[t], t) \in \mathbb{C}^{N_{\text{BS}} \times 1}$ given by

$$\mathbf{h}_{\text{eff}}(\mathbf{c}[t], t) \triangleq \mathbf{h}^{\text{UB}}[t] + \mathbf{H}^{\text{IB}}[t] \sum_{\ell=1}^{L[t]} \mathbf{\Phi}(\mathbf{c}[t], \theta_\ell[t]) \mathbf{h}_\ell^{\text{UI}}(\theta_\ell[t], t), \tag{7}$$

which encapsulates all the channels (i.e., $\mathbf{h}^{\text{UB}}[t]$, $\mathbf{H}^{\text{IB}}[t]$, and $\mathbf{h}^{\text{UI}}[t]$) and the specific IRS configuration (i.e., $\mathbf{c}[t]$).

III. Problem Formulation, Challenges, and Limited Feedback Protocols

We first formulate the data rate maximization problem for IRS control and discuss the challenges associated with solving it in Sec. III-A. To address the challenges, we propose a novel adaptive codebook-based limited feedback protocol for IRS control in Sec. III-B. Finally, we discuss how the IRS codebook differs from traditional precoding codebooks in Sec. III-C.

A. Problem Formulation and Challenges

We aim to maximize the capacity of the channel as a performance metric. Therefore, we formulate the achievable data rate maximization problem at time t as

$$\underset{\mathbf{c}[t]}{\text{maximize}} \quad R(\mathbf{c}[t], t) = \log_2\left(1 + \frac{P \|\mathbf{h}_{\text{eff}}(\mathbf{c}[t], t)\|_2^2}{\sigma^2}\right) \tag{8}$$

$$\text{subject to} \quad C_{\min} \leq C_n[t] \leq C_{\max},\ n = 1, ..., N_{\text{IRS}}. \tag{9}$$

The constraint (9) states that each capacitance $C_n[t]$, $n = 1, ..., N_{\text{IRS}}$, should reside in the allowed region discussed in (1).[4] The objective is to adapt $\mathbf{c}[t]$ based on the time-varying channels.

Operationally, we aim for the optimization (8)-(9) to be solved at the BS since (i) the BS can obtain measurements and exploit them in deriving the solution for IRS control while the IRS has no sensing capability, and (ii) the BS usually has abundant computing resources while the IRS is often not equipped with powerful processing units. The BS would then

[3] In fact, the impedance in (2) and the reflection coefficient in (3) are dependent on the frequency f. However, since we consider a fixed frequency with a narrowband of a few tens of MHz bandwidth, we can approximate the IRS reflection coefficients as constant across f [14], [15] and, thus, do not consider the dependency of f.

[4] A discrete capacitance control is sometimes preferred to reduce the hardware utilization cost. In this paper, we focus on developing methodologies for continuous control as a general case. Our methodologies can be readily applied to the discrete control case by quantizing the interval $[C_{\min}, C_{\max}]$ and mapping the continuous capacitance to the closest discrete value.

generate feedback information for the IRS, used to reconfigure the capacitance at the meta-atoms via the IRS control board. However, solving (8)-(9) and tuning the IRS meta-atoms via the feedback link face the following challenges.

(C1) *CSI knowledge requirements of optimization-based methods.* Conventional optimization-based methods to solve (8)-(9) require the BS to estimate all the channels, $\mathbf{h}^{\mathrm{UB}}[t]$, $\mathbf{H}^{\mathrm{IB}}[t]$, and $\mathbf{h}^{\mathrm{UI}}[t]$, and incident angles $\{\theta_\ell[t]\}_\ell$ in real-time, which are encapsulated in $\mathbf{h}_{\text{eff}}(\mathbf{c}[t],t)$. However, channel estimation techniques require known IRS reflection coefficients, which cannot be obtained in a real-world system due to the multi-path nature of the channels and angle-dependent behavior of meta-atoms (refer to Sec. I-B).

(C2) *Dynamic channels and overhead requirements.* Adaptive control of $\mathbf{c}[t]$ is necessary to have an efficient IRS operation in time-varying channels. Such control requires periodic information acquisition from the BS. The time overhead of information acquisition should be a small fraction of the channel coherence time to ensure a reasonable data transmission time.

(C3) *Low data rate of feedback link.* A feedback link refers to the data link from the BS to the control board of the IRS [15]. Typically, the feedback link has low data rate because the channel state information (CSI) of the feedback link is unknown at the BS [5]. Therefore, the BS must feed back only small amount of necessary information to the IRS.

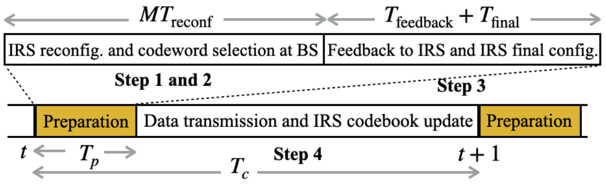
These challenges render the existing IRS control protocols ineffective since they mostly rely on either full CSI or channel estimates, neglect the overhead of information acquisition from the BS, and overlook the behavior of meta-atoms and the characteristics of the feedback channel. The main contribution of our work is developing a methodology to jointly address these challenges.

B. Adaptive Codebook-based Limited Feedback Protocols for IRS-assisted Communication

Motivated by the low overhead feedback requirement (see (C3)), we propose to exploit a *codebook* structure for IRS control, where the BS sends only a quantized codeword index to the IRS. Further, we consider adaptive design of this codebook based on channel variations. We denote the adaptive codebook as $\mathcal{C}[t]=\{\mathbf{q}_m[t]\}_{m=1}^M$, where $\mathbf{q}_m[t]\in\mathbb{R}^{N_{\mathrm{IRS}}}$ is the m-th codeword (capacitance vector) in the codebook and M is the codebook size. The codebook is stored and its updates are conducted at the IRS through its control board [15] (e.g., see Fig. 1(a)). We propose a novel *limited feedback protocol* consisting of four steps conducted per each coherence time block t depicted in Fig. 2:

Step 1 ***IRS channel sounding and reconfiguration***. While the UE transmits pilot symbols, the IRS explores all the M capacitance vectors, i.e., $\mathbf{q}_m[t]$, in $\mathcal{C}[t]$, $m=1,...,M$.

Step 2 ***Codeword selection at BS.*** The BS measures the effective channel $\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t],t)$ and calculates the data rate $R(\mathbf{q}_m[t],t)$ in (8) as the IRS applies $\mathbf{q}_m[t]$, $m=1,...,M$. The BS obtains the codeword index $m^\star[t]=\underset{m\in\{1,...,M\}}{\arg\max}\ R(\mathbf{q}_m[t],t)$.



MT_{reconf}
$T_{\text{feedback}}+T_{\text{final}}$
IRS reconfig. and codeword selection at BS
Feedback to IRS and IRS final config.
Step 1 and 2
Step 3
Preparation
Data transmission and IRS codebook update
Preparation
t
T_p
T_c
Step 4
$t+1$

Fig. 2: Time frame structure of the proposed limited feedback protocol for IRS-assisted communication.

Step 3 ***Feedback to IRS and IRS final configuration.*** The BS feeds back the index $m^\star[t]\in\{1,...,M\}$ to the IRS. Then, the IRS tunes the meta-atoms with $\mathbf{q}_\star[t]=\mathbf{q}_{m^\star[t]}[t]\in\mathcal{C}[t]$.

Step 4 ***Data transmission and IRS codebook update.*** The data transmission is conducted during the rest of the coherence time. During this time period, the IRS obtains the next codebook $\mathcal{C}[t+1]$ either locally or with assistance from the BS.

The benefits of our protocol include its (i) simple procedure for IRS configuration in limited coherence times, (ii) low-overhead feedback, and (iii) adaptation to dynamic channels. In regards to (i), our protocol does not require the complicated processes required for the estimation/tracking of the channels, incident angles, and UE location (see (C1) in Sec. III-A). In particular, it only requires partial CSI (i.e., the compound channel in (7)) for the IRS control. As for (ii), we consider digital feedback, rather than feeding back continuous vectors or matrices, which reduces the feedback time overhead (see (C3) in Sec. III-A). In regards to (iii), we consider adaptive codebook updates to adapt to the dynamic channels found in real systems (see (C2) in Sec. III-A).

Careful design of the codebook $\mathcal{C}[t]$ is critical to obtaining high data rates, since the codewords $\{\mathbf{q}_m[t]\}_{m=1}^M$ are the solution candidates and the maximizer ($\mathbf{q}_\star[t]$ in **Step** 3) among them is selected as a solution.[5] Shown in **Step** 4, the IRS obtains the next codebook $\mathcal{C}[t+1]$ at time t, and therefore the codebook update can be regarded as a *prediction and refinement* problem. Since the channels are in practice correlated across consecutive coherence times, the BS exploits the previous effective channels estimated up to time t as available information, denoted by $\mathcal{G}[t]=\{\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t'],t')\}_{1\le m\le M,t'\le t}$, for the codebook update. We develop our methodology under the assumption that the BS obtains the effective channels without any noise. However, for simulations in Sec. V, we consider the case where the BS measures a noisy version of the effective channels (i.e., the received pilot signal that contains noise) to have more realistic results.

We finally reformulate the problem (8)-(9) to be compatible with the proposed protocol and change the design variable to the codebook. Because the channels at time $t+1$ are unknown, we formulate the problem as a stochastic optimization:

$$\underset{\substack{\mathcal{C}[t+1]=\\ \{\mathbf{q}_m[t+1]\}_{m=1}^M}}{\text{maximize}}\ \mathbb{E}_{\mathcal{E}[t+1]}\Big[\max_{\mathbf{q}_m[t+1]\in\mathcal{C}[t+1]} R(\mathbf{q}_m[t+1],t+1)\Big|\mathcal{G}[t]\Big] \tag{10}$$

[5] The data rate performance also depends on the codebook size M. However, M should be limited because of the finite coherence time and non-negligible IRS reconfiguration time. We assume that M is predetermined and fixed in the protocol.

$$\text{subject to} \quad C_{\min} \leq q_{m,n}[t+1] \leq C_{\max}, \quad m = 1, ..., M, \;\; n = 1, ..., N_{\mathrm{IRS}}, \tag{11}$$

where $q_{m,n}[t+1]$ denotes the n-th entry of the m-th codeword $\mathbf{q}_m[t+1] = [q_{m,1}[t+1], ..., q_{m,N_{\mathrm{IRS}}}[t+1]]$ at time $t+1$ and $\mathcal{E}[t+1]$ denotes the channel distribution at time $t+1$, including the statistics of $\mathbf{H}^{\mathrm{IB}}[t+1]$, $\mathbf{h}^{\mathrm{UI}}[t+1]$, and $\mathbf{h}^{\mathrm{UB}}[t+1]$. In the above problem, we aim to obtain the next codebook $\mathcal{C}[t+1]$ with the available information $\mathcal{G}[t]$ such that the codebook contains good codewords to maximize the expectation of the data rate obtained from the *best* codeword over the next channel statistics $\mathcal{E}[t+1]$. Note that the channel distributions are not static over time due to temporal environment variations (e.g., temperature, precipitation, UE mobility, etc.). Also, the channel distribution $\mathcal{E}[t+1]$ is unknown to the BS, which adds another degree of difficulty to solve (10)-(11). We will propose low-overhead adaptive codebook designs to solve (10)-(11).

C. Distinction of the IRS Codebook from the Current Art Precoding Codebooks

In traditional codebook-based wireless communications, an encoding function and distortion measure are defined, and the codebook is designed such that the distortion measure is minimized. When channel vectors are modeled as scaled versions of array response vectors, codebooks are often constructed as a set of particular vector subspaces characterized by an array manifold structure, such as DFT quantization-based codebooks [26], LTE/5G NR codebooks [34], [35], beamforming quantization-based codebooks [36]–[38], codebooks with Grassmannian line packing [39], [40], and codebooks for beam alignment [41]. In [42], [43], adaptive codebook design methods are proposed based on a specific manifold structure of the channels. For such codebook construction, the channels or their statistics are assumed to be known and each codeword is designed on the complex vector space. However, in our problem, we do not assume any knowledge of the channel statistics, and each codeword in the IRS codebook resides in the N_{IRS}-dimensional hypercube where each entry of the codeword ranges in $[C_{\min}, C_{\max}]$.

The RVQ codebook [27], [28] can be exploited in our proposed protocol, and thus we use it as a baseline. Specifically, in RVQ design each codeword is randomly generated such that each entry ranges in $[C_{\min}, C_{\max}]$ at each time. Although this baseline can be operated in the proposed limited feedback protocol, it would not adapt to the varying channels properly. Ideally, it is best to update the codebook by predicting how the optimal solution *changes* according to the next-time channel statistics in (10)-(11). Motivated by this, we next propose two *adaptive* codebook approaches where the codebook is updated with the previous decisions and responses.

IV. Adaptive Codebook Design

For adaptive codebook design, we propose a low-overhead perturbation-based approach in Sec. IV-A and a deep neural network (DNN) policy-based approach in Sec. IV-B. Then, we discuss the computational complexity of the approaches and present a group control strategy in Sec. IV-C. Finally, we quantify the time overhead and the average data rate over one channel coherence block in Sec. IV-D.

Algorithm 1 Random adjacency (RA) codebook design in the limited feedback protocol

```
Input: $N_{\mathrm{timestep}}$ (the duration of the algorithm), $C_{\min}$, and $C_{\max}$.
The IRS randomly generates the initial codebook $\mathcal{C}[0] = \{\mathbf{q}_m[0]\}_{m=1}^{M}$ within the allowed region in (11).
for $t = 0, ..., N_{\mathrm{timestep}} - 1$ do
    Step 1. IRS channel sounding and reconfiguration. The IRS meta-atoms are tuned following $\{\mathbf{q}_m[t]\}_{m=1}^{M}$.
    Step 2. Codeword selection at BS. The BS determines $m^\star[t] = \arg\max_{m \in \{1,...,M\}} R(\mathbf{q}_m[t], t)$.
    Step 3. Feedback to IRS and IRS final configuration. The BS feeds back the index $m^\star[t] \in \{1, ..., M\}$ to the IRS with total $\lceil \log_2 M \rceil$ feedback bits. The IRS tunes the meta-atoms with $\mathbf{q}_\star[t] = \mathbf{q}_{m^\star[t]}[t]$ for data transmission period.
    Step 4. Data transmission and IRS codebook update. The IRS obtains $\mathcal{C}[t+1]$ according to (12).
end for
```

A. Random Adjacency (RA) Approach

One of the natural ways to construct an adaptive codebook is to use random perturbation-based methods used in obtaining the solutions to beamforming design [29], which determine the current solution by adding a random perturbation to the previous solution. We accordingly propose a *random adjacency* (RA) approach, which can be viewed as a random perturbation-based method for codebook design, to solve the optimization (10)-(11). Since the optimization (10)-(11) is conducted successively over time in time-correlated channels, the optimal solutions in adjacent time blocks are expected to be close to one another. The RA approach exploits this intuition by generating multiple solution candidates (for the codebook at time $t+1$) around the previous solution. The codebook resides and is updated at the IRS, which requires no feedback overhead for the codebook update. The feedback is only used to transfer the index of the best codeword deployed for data transmission in each coherence time in **Step** 3 in Sec. III-B.

Formally, the IRS obtains the codebook $\mathcal{C}[t+1] = \{\mathbf{q}_m[t+1]\}_{m=1}^{M}$, where the m-th codeword is updated by adding a random perturbation $\mathbf{z}_m[t] \in \mathbb{R}^{N_{\mathrm{IRS}}}$ to the previous solution $\mathbf{q}_\star[t]$ (obtained in **Step** 4 in Sec. III-B) as

$$\mathbf{q}_m[t+1] = \mathrm{clip}(\mathbf{q}_\star[t] + \mathbf{z}_m[t], [C_{\min}, C_{\max}]), \; m \in \{1, ..., M\}, \tag{12}$$

which we call the *RA update* for the m-th codeword. Here, $\mathrm{clip}(\cdot, [C_{\min}, C_{\max}])$ is an element-wise clip function ensuring constraint (11). Each entry of $\mathbf{z}_m[t]$ is generated from the uniform distribution $\mathcal{U}(-\delta, \delta)$, where δ is the maximum step size for the entry update. The RA approach is summarized in Algorithm 1.

The codebook update by the RA approach incurs a small computation and communication overhead, which will be discussed in Sec. IV-C and IV-D. Intuitively, it becomes more effective as the number of codewords M grows larger since more random points increase the chance of obtaining better codewords. However, M is limited due to the non-negligible IRS reconfiguration time and finite coherence time. This makes the performance of the RA approach restricted due to the nature of the randomness and motivates us to develop the next codebook update algorithm.

B. DNN Policy-based IRS Control (DPIC) Approach

DNNs have been exploited to capture implicit features in the observed data. Motivated by this, we propose a DNN policy-based IRS control (DPIC) approach, aiming to learn *policies* for updating the codebook using the history of observations. In DPIC, the codebook resides both at the IRS and BS, and the IRS updates the codebook via information reception from the BS through the feedback link. We consider that each codeword is updated *independently* based on its prior deployments. Henceforth, without loss of generality, we focus on the updates of the m-th codeword. Note that having individual codeword updates substantially reduces the input/output dimension of the DNN as compared to updating the codebook as a whole (i.e., conducting the learning on the concatenation of all the codewords as a large vector), which enhances learning efficiency. We will first describe the overall procedure for the codeword updates at the IRS and BS in Sec. IV-B1. Since the codeword update is a successive decision making process (Sec. IV-B2), we formulate it as a Markov decision process (MDP) in Sec. IV-B3. We then develop our learning architecture for training (Sec. IV-B4) and utilization (Sec. IV-B5).

1) Low Overhead IRS Control via Direction Codebook: We first introduce a fixed *direction codebook* $\mathcal{D} = \{\mathbf{d}_k\}_{k=1}^{K}$ where $\mathbf{d}_k \in \mathbb{R}^{N_{\text{IRS}}}$, $k = 1, ..., K$, is a *direction vector*, representing the variation of the capacitance vector to be used for the codeword update. The BS only transmits the index of a direction vector in $\mathcal{D}$ to the IRS, which enables low feedback overhead for the codeword update. We assume that $\mathcal{D}$ is generated once at the beginning of the policy learning and shared at both the BS and IRS. For simulations in Sec. V, $\mathcal{D}$ is constructed via RVQ. The BS, as a processing entity, employs a *learning architecture* consisting of a DNN policy and a subsequent quantization process. In the learning architecture, the BS obtains the vector $\boldsymbol{u}_m[t] \in \mathbb{R}^{N_{\text{IRS}}}$ as an output of the DNN policy, and finds the index $k_m[t] \in \{1, ..., K\}$ through the subsequent quantization process, such that $\mathbf{d}_{k_m[t]}$ in $\mathcal{D}$ is the most similar to $\boldsymbol{u}_m[t]$. The BS then feeds back the index $k_m[t]$ to the IRS, and the IRS uses the index to recover $\mathbf{d}_{k_m[t]}$ from $\mathcal{D}$ and then updates the m-th codeword as

$$\mathbf{q}_m[t+1] = \text{clip}(\mathbf{q}_m[t] + \mathbf{d}_{k_m[t]}, [C_{\min}, C_{\max}]), m \in \{1, ..., M\}, \tag{13}$$

which we call the *DPIC update* for the m-th codeword.

2) Successive Decision Making for Codeword Update: Our learning architecture consists of two phases: *training phase* and *utilization phase*. In the training phase, the BS aims to train the DNN policy to have an improved $\boldsymbol{u}_m[t]$ over time, while in the utilization phase the BS exploits the trained DNN policy without additional training. In both phases, the BS first determines $\boldsymbol{u}_m[t]$ with the DNN policy based on the current information (i.e., the codeword $\mathbf{q}_m[t]$ in use and the effective channel $\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t], t)$). Subsequently, the BS quantizes $\boldsymbol{u}_m[t]$ to get the index $k_m[t]$ as described in Sec. IV-B4&IV-B5. The BS then feeds back $k_m[t]$ to the IRS, from which the IRS obtains the next codeword $\mathbf{q}_m[t+1]$ through (13). The next codeword affects the subsequent information at the BS (i.e., $\mathbf{q}_m[t+1]$ and $\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t+1], t+1)$). The codeword update can thus be formulated as a successive decision making process (Sec. IV-B3).

3) Markov Decision Process (MDP) for Codeword Update: We construct an MDP for the codeword update at the BS learning architecture with the following state, action, and reward.

State. The state consists of information pertinent to the environment evolution, defined by

$$\mathbf{s}_m[t] = \{\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t], t), \mathbf{q}_m[t]\} \in \mathcal{S} = \mathbb{R}^{2N_{\text{BS}}+N_{\text{IRS}}}, \quad m \in \{1, ..., M\}, \tag{14}$$

where the real and imaginary parts of $\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t], t)$ are stored as separate state dimensions.

Action. The action is the continuous direction vector $\boldsymbol{u}_m[t]$ described as

$$\mathbf{a}_m[t] = \boldsymbol{u}_m[t] \in \mathcal{A} = [-\delta, \delta]^{N_{\text{IRS}}}, \ m \in \{1, ..., M\}, \tag{15}$$

where each entry of the action is bounded to the maximum step size, i.e., $[-\delta, \delta] \subset \mathbb{R}$. The action $\mathbf{a}_m[t]$ is used to determine the index $k_m[t]$ based on different processes in training (Sec. IV-B4) and utilization (Sec. IV-B5) phases. The next codeword $\mathbf{q}_m[t+1]$ is then obtained from $k_m[t]$ by (13).

Reward. The reward provides an efficacy for desirable policy learning by evaluating an action at a given state. We subsequently define the MDP reward as

$$r_m[t] = R(\mathbf{q}_m[t+1], t+1) - \nu N_{\text{clip},m}[t] \in \mathbb{R}, \ m \in \{1, ..., M\}, \tag{16}$$

where $R(\mathbf{q}_m[t+1], t+1)$ denotes the data rate measured at time $t+1$ using codeword $\mathbf{q}_m[t+1]$ and $N_{\text{clip},m}[t]$ denotes the number of clipped elements/dimensions in the vector $\mathbf{q}_m[t] + \mathbf{d}_{k_m[t]} \in \mathbb{R}^{N_{\text{IRS}}}$ that hit the clipping threshold in (13). In (16), $\nu > 0$ is a weight parameter to match the order-of-magnitude of $R(\mathbf{q}_m[t+1], t+1)$ and $N_{\text{clip},m}[t]$, which we will investigate experimentally in Sec. V-E. $\nu N_{\text{clip},m}[t]$ is added as a penalty to avoid actions that result in the capacitance vectors outside of the allowed region. Note that the reward $r_m[t]$ is obtained at the next time $t+1$ since the data rate $R(\mathbf{q}_m[t+1], t+1)$ is calculated at time $t+1$.

Based on the state, action, and reward, the MDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}_{\mathbf{s}}^{\mathbf{a}}, P_{\mathbf{s},\mathbf{s}'}^{\mathbf{a}}, \gamma)$, where $P_{\mathbf{s},\mathbf{s}'}^{\mathbf{a}} = Pr\big[\mathbf{s}_m[t+1] = \mathbf{s}' \big| \mathbf{s}_m[t] = \mathbf{s}, \mathbf{a}_m[t] = \mathbf{a}\big]$ is the state transition probability for moving from state $\mathbf{s}$ to $\mathbf{s}'$ via action $\mathbf{a}$, $\mathcal{R}_{\mathbf{s}}^{\mathbf{a}} = \mathbb{E}\big[r_m[t] \big| \mathbf{s}_m[t] = \mathbf{s}, \mathbf{a}_m[t] = \mathbf{a}\big]$ is the reward function, and γ is the discount factor used to take into account the rewards for the distant future.

4) Training Phase for DNN Policy Learning: We tailor a deep reinforcement learning (DRL) methodology to train the DNN policy with the formulated MDP. We assume that the BS trains $M_{\text{A}} \leq M$ different learning architectures, which are referred to as *agents*. We consider that each agent is trained with a single codeword, where the codewords across the agents are non-overlapping. Thus M_{A} codewords are used during the training phase of DPIC. Let $\mathcal{M}_{\text{A}} \subset \{1, ..., M\}$ denote the indices of the codewords associated with learning agents with $|\mathcal{M}_{\text{A}}| = M_{\text{A}}$. We will use m to denote a codeword and its associated agent interchangeably. We consider that agent $m \in \mathcal{M}_{\text{A}}$ has the DNN policy $\pi(\mathbf{s}_m[t]; \mathbf{w}_{\pi,m})$, which outputs the continuous vector $\boldsymbol{u}_m[t] \in \mathbb{R}^{N_{\text{IRS}}}$ given state $\mathbf{s}_m[t]$, where $\mathbf{w}_{\pi,m}$ is the respective DNN weight parameters.

Behavior policy. We refer to $\pi(\mathbf{s}_m[t];\mathbf{w}_{\pi,m})$ as a *target* policy, which is different from the *behavior* policy that determines the actual action applied to the environment. The actual action of the agent m (i.e., $\mathbf{d}_{k_m[t]}$ in (13)) is determined at the BS via the two following steps. First, the BS adds the random noise vector $\mathbf{v}_m[t]$ to the output of the target policy $\pi(\mathbf{s}_m[t];\mathbf{w}_{\pi,m})$ to have more diverse responses and avoid getting trapped in local optima during training [44], where $\mathbf{v}_m[t] \sim \mathcal{N}(\mathbf{0}, \epsilon[t]\mathbf{I})$ with $\epsilon[t]$ denoting the exploration noise variance. We then use the clip function to confine the output result to the feasible action space. Second, the BS performs the *quantization process*, through which the BS determines the index $k_m[t] \in \{1, ..., K\}$, such that the direction vector $\mathbf{d}_{k_m[t]}$ is closest to the output of the first step in Euclidean distance. In other words, the behavior policy $\mu_{\mathcal{D},\pi}(\mathbf{s}_m[t])$, yielding $k_m[t]$ as an output, is represented as

$$\begin{aligned} k_m[t] = \mu_{\mathcal{D},\pi}(\mathbf{s}_m[t]) = \underset{k\in\{1,...,K\}}{\arg\min} \ & \big\|\text{clip}(\pi(\mathbf{s}_m[t];\mathbf{w}_{\pi,m}) \\ & + \mathbf{v}_m[t], [-\delta,\delta]) - \mathbf{d}_k\big\|_2. \end{aligned} \quad (17)$$

DNN policy learning with actor-critic network. For DNN policy learning, we exploit the actor-critic network using DNNs as function approximators that can learn policies in continuous state and action spaces [45]. The actor-critic network consists of an actor network and a critic network, where the former selects an action using a policy and the later evaluates/criticizes the action to guide the actor network to take better actions over time. First, for a given policy $\pi(\cdot)$, we define the action-value function, called Q-function, with the discount factor γ as

$$Q_m^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_\xi\left[\sum\nolimits_{i=0}^{\infty} \gamma^i r_m[t+i] \big| \mathbf{s}_m[t] = \mathbf{s}, \mathbf{a}_m[t] = \mathbf{a}, \pi\right], \quad (18)$$

where ξ encapsulates the state transition probability $P_{\mathbf{s},\mathbf{s}'}^{\mathbf{a}}$ and reward function $\mathcal{R}_{\mathbf{s}}^{\mathbf{a}}$. Using $Q_m^\pi(\mathbf{s},\mathbf{a})$, we define the performance objective [46] as

$$\begin{aligned} J_m^\mu(\pi) &= \int_{\mathcal{S}} \rho_m^\mu(\mathbf{s}) Q_m^\pi(\mathbf{s}, \pi(\mathbf{s};\mathbf{w}_{\pi,m})) d\mathbf{s} \\ &= \mathbb{E}_{\mathbf{s}\sim\rho_m^\mu}\big[Q_m^\pi(\mathbf{s}, \pi(\mathbf{s};\mathbf{w}_{\pi,m}))\big], \end{aligned} \quad (19)$$

where $J_m^\mu(\pi)$ denotes the expected cumulative discounted reward over all the states when the state trajectory is provided by behavior policy μ. Here, $\rho_m^\mu(\mathbf{s}) = \int_{\mathcal{S}} \sum_{i=1}^{\infty} \gamma^{i-1} p_1(\tilde{\mathbf{s}}) p(\tilde{\mathbf{s}} \to \mathbf{s}, i, \mu) d\tilde{\mathbf{s}}$ is the discounted state distribution where $p(\tilde{\mathbf{s}} \to \mathbf{s}, i, \mu)$ denotes the probability density at state $\mathbf{s}$ after transitioning for i time steps from state $\tilde{\mathbf{s}}$ under behavior policy μ, and $p_1(\tilde{\mathbf{s}})$ is the probability density of the initial state distribution. The objective is to design the target policy π (i.e., the DNN parameters $\mathbf{w}_{\pi,m}$, $m \in \mathcal{M}_{\text{A}}$), such that $J_m^\mu(\pi)$ is maximized. To achieve this, the learning for the DNN parameters is conducted by the gradient-based update as

$$\mathbf{w}_{\pi,m} \leftarrow \mathbf{w}_{\pi,m} - \alpha_\pi \nabla_{\mathbf{w}_{\pi,m}} J_m^\mu(\pi), \quad (20)$$

where $\nabla_{\mathbf{w}_{\pi,m}} J_m^\mu(\pi) \approx \mathbb{E}_{\mathbf{s}\sim\rho_m^\mu}\big[\nabla_{\mathbf{w}_{\pi,m}} \pi(\mathbf{s};\mathbf{w}_{\pi,m}) \nabla_{\mathbf{a}} Q_m^\pi(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s};\mathbf{w}_{\pi,m})}\big]$ denotes the deterministic policy gradient (DPG), the derivation of which is detailed in [46], and α_π is the learning rate.

We define another DNN as a function approximator for the Q-function, i.e., $Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m}) \approx Q_m^\pi(\mathbf{s},\mathbf{a})$, with the parameters $\mathbf{w}_{Q,m}$, which is used to calculate the gradient in (20). We obtain $Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m})$ using Q-learning [44], [46] by minimizing the following loss function

$$\mathcal{L}_m = \mathbb{E}\big[(y - Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m}))^2\big], \quad (21)$$

where the target value y is given by $y = r + \gamma Q(\mathbf{s}', \pi(\mathbf{s}';\mathbf{w}_{\pi,m}); \mathbf{w}_{Q,m})$. Here, $\mathbf{s}'$ and r are the next state and the reward obtained from the environment by taking action $\mathbf{a}$ given state $\mathbf{s}$, respectively. The learning for $\mathbf{w}_{Q,m}$ is followed by the gradient-based update as

$$\mathbf{w}_{Q,m} \leftarrow \mathbf{w}_{Q,m} - \alpha_Q \nabla_{\mathbf{w}_{Q,m}} \mathcal{L}_m, \quad (22)$$

where α_Q is the learning rate and $\nabla_{\mathbf{w}_{Q,m}} \mathcal{L}_m = -\mathbb{E}\big[(y - Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m})) \nabla_{\mathbf{w}_{Q,m}} Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m})\big]$.

Using large and non-linear function approximators, such as DNNs, for reinforcement learning has been known to cause learning instability [44]. We make use of the strategies proposed in [45] to stabilize the learning. First, we use a *replay buffer* $\mathcal{B}_m$ to save the tuple $(\mathbf{s}_m[t], \mathbf{a}_m[t], r_m[t], \mathbf{s}_m[t+1])$ over time, and conduct the training for $\pi(\mathbf{s};\mathbf{w}_{\pi,m})$ and $Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m})$ via mini-batch learning with N_{batch} samples, $(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)$, $i = 1, ..., N_{\text{batch}}$, randomly chosen from $\mathcal{B}_m$. This makes the samples chosen for learning uncorrelated which leads to learning stability. Second, for updating $Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m})$, *soft target* updates are used to improve learning stability by making the target value y in (21) slowly varying. This is enabled by constructing two additional DNNs, called *copied networks*: the copied policy $\pi(\mathbf{s};\mathbf{w}_{\pi,m}^{\text{cp}})$ with parameters $\mathbf{w}_{\pi,m}^{\text{cp}}$, and the copied Q-function $Q(\mathbf{s},\mathbf{a};\mathbf{w}_{Q,m}^{\text{cp}})$ with $\mathbf{w}_{Q,m}^{\text{cp}}$, for $m \in \mathcal{M}_{\text{A}}$. That is, we obtain the target value y_i (corresponding to the i-th sample) as $y_i = r_i + \gamma Q(\mathbf{s}'_i, \pi(\mathbf{s}'_i;\mathbf{w}_{\pi,m}^{\text{cp}}); \mathbf{w}_{Q,m}^{\text{cp}})$. Through the mini-batch learning, the gradients in (20) and (22) are approximated as

$$\begin{aligned} \nabla_{\mathbf{w}_{\pi,m}} J_m^\mu(\pi) \approx \sum\nolimits_{i=1}^{N_{\text{batch}}} & \nabla_{\mathbf{w}_{\pi,m}} \pi(\mathbf{s}_i;\mathbf{w}_{\pi,m}) \\ & \times \nabla_{\mathbf{a}} Q(\mathbf{s}_i, \mathbf{a}; \mathbf{w}_{Q,m})|_{\mathbf{a}=\pi(\mathbf{s}_i;\mathbf{w}_{\pi,m})}, \end{aligned} \quad (23)$$

$$\begin{aligned} \nabla_{\mathbf{w}_{Q,m}} \mathcal{L}_m \approx -\sum\nolimits_{i=1}^{N_{\text{batch}}} & (y_i - Q(\mathbf{s}_i, \mathbf{a}_i); \mathbf{w}_{Q,m}) \\ & \times \nabla_{\mathbf{w}_{Q,m}} Q(\mathbf{s}_i, \mathbf{a}_i; \mathbf{w}_{Q,m}). \end{aligned} \quad (24)$$

The copied networks are then updated with the soft target update parameter τ as

$$\begin{aligned} \mathbf{w}_{\pi,m}^{\text{cp}} &\leftarrow \tau \mathbf{w}_{\pi,m} + (1-\tau)\mathbf{w}_{\pi,m}^{\text{cp}}, \\ \mathbf{w}_{Q,m}^{\text{cp}} &\leftarrow \tau \mathbf{w}_{Q,m} + (1-\tau)\mathbf{w}_{Q,m}^{\text{cp}}. \end{aligned} \quad (25)$$

Fig. 3 depicts the workflow for training each agent m. We have so far focused on the training process of each agent m, $m \in \mathcal{M}_{\text{A}}$. Prior to training, the BS determines the number of agents trained, i.e., $M_{\text{A}} = |\mathcal{M}_{\text{A}}|$. The BS can train as many agents as possible, i.e., $M_{\text{A}} = M$, or $M_{\text{A}} < M$, while the rest of the $M - M_{\text{A}}$ codewords are updated by the RA update, of which indices form the set $\mathcal{M}_{\text{RA}} = \{1, ..., M\} \backslash \mathcal{M}_{\text{A}}$. Training different numbers of agents leads to different performance (see Sec. V) and incurs different computation/communication overhead (see Sec. IV-C&IV-D). The overall algorithm for training M_{A} agents is given in Algorithm 2.

5) Utilization Phase with Trained DNN Policies: In the utilization phase, we utilize the trained agents to conduct the codebook update without additional training of the agents. Among M codewords, we can select M_{DPIC} codewords to be

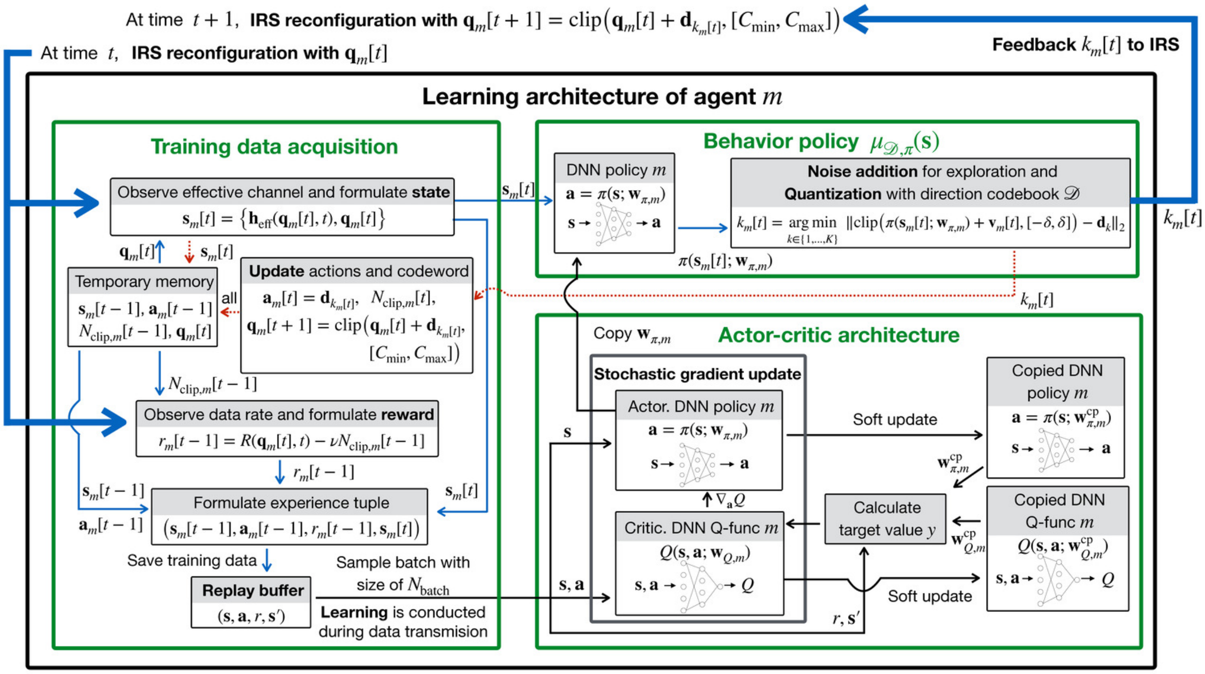

At time $t+1$, IRS reconfiguration with $\mathbf{q}_m[t+1] = \text{clip}(\mathbf{q}_m[t] + \mathbf{d}_{k_m[t]}, [C_{\min}, C_{\max}])$
At time t, IRS reconfiguration with $\mathbf{q}_m[t]$
Feedback $k_m[t]$ to IRS
Learning architecture of agent m
Training data acquisition
Observe effective channel and formulate state
$\mathbf{s}_m[t] = \{\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t], t), \mathbf{q}_m[t]\}$
$\mathbf{q}_m[t]$
$\mathbf{s}_m[t]$
Temporary memory
$\mathbf{s}_m[t-1], \mathbf{a}_m[t-1]$
$N_{\text{clip},m}[t-1], \mathbf{q}_m[t]$
all
Update actions and codeword
$\mathbf{a}_m[t] = \mathbf{d}_{k_m[t]}, \quad N_{\text{clip},m}[t],$
$\mathbf{q}_m[t+1] = \text{clip}(\mathbf{q}_m[t] + \mathbf{d}_{k_m[t]},$
$[C_{\min}, C_{\max}])$
$N_{\text{clip},m}[t-1]$
Observe data rate and formulate reward
$r_m[t-1] = R(\mathbf{q}_m[t], t) - \nu N_{\text{clip},m}[t-1]$
$r_m[t-1]$
$\mathbf{s}_m[t-1]$
$\mathbf{a}_m[t-1]$
$\mathbf{s}_m[t]$
Formulate experience tuple
$(\mathbf{s}_m[t-1], \mathbf{a}_m[t-1], r_m[t-1], \mathbf{s}_m[t])$
Save training data
Replay buffer
$(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')$
Sample batch with size of N_{batch}
Learning is conducted during data transmision
$\mathbf{s}_m[t]$
Behavior policy $\mu_{\mathscr{D},\pi}(\mathbf{s})$
DNN policy m
$\mathbf{a} = \pi(\mathbf{s}; \mathbf{w}_{\pi,m})$
$\mathbf{s} \rightarrow$ $\rightarrow \mathbf{a}$
Noise addition for exploration and Quantization with direction codebook $\mathscr{D}$
$k_m[t] = \underset{k \in \{1,\dots,K\}}{\arg\min} \|\text{clip}(\pi(\mathbf{s}_m[t]; \mathbf{w}_{\pi,m}) + \mathbf{v}_m[t], [-\delta, \delta]) - \mathbf{d}_k\|_2$
$\pi(\mathbf{s}_m[t]; \mathbf{w}_{\pi,m})$
$k_m[t]$
$k_m[t]$
Copy $\mathbf{w}_{\pi,m}$
Actor-critic architecture
Stochastic gradient update
Actor. DNN policy m
$\mathbf{a} = \pi(\mathbf{s}; \mathbf{w}_{\pi,m})$
$\mathbf{s} \rightarrow$ $\rightarrow \mathbf{a}$
$\mathbf{s}$
$\nabla_{\mathbf{a}} Q$
Critic. DNN Q-func m
$Q(\mathbf{s}, \mathbf{a}; \mathbf{w}_{Q,m})$
$\mathbf{s}, \mathbf{a} \rightarrow$ $\rightarrow Q$
$\mathbf{s}, \mathbf{a}$
Soft update
Copied DNN policy m
$\mathbf{a} = \pi(\mathbf{s}; \mathbf{w}^{\text{cp}}_{\pi,m})$
$\mathbf{s} \rightarrow$ $\rightarrow \mathbf{a}$
$\mathbf{w}^{\text{cp}}_{\pi,m}$
Calculate target value y
$\mathbf{w}^{\text{cp}}_{Q,m}$
Copied DNN Q-func m
$Q(\mathbf{s}, \mathbf{a}; \mathbf{w}^{\text{cp}}_{Q,m})$
$\mathbf{s}, \mathbf{a} \rightarrow$ $\rightarrow Q$
Soft update
$r, \mathbf{s}'$

Fig. 3: The workflow for training each agent m in our limited feedback protocol. The agent collects the training data, updates the DNN policy (actor network) and DNN Q-function (critic network) in the actor-critic architecture via mini-batch learning, selects the action via the behavior policy, and feeds back the direction index to the IRS.

Algorithm 2 Training M_{A} agents with actor-critic architecture in the proposed protocol

```
Input. $N_{\text{episode}}$ (the number of learning episodes), $N_{\text{timestep}}$ (the duration of each episode), $\epsilon_0 = (C_{\max} - C_{\min})/5$ (the initial exploration variance), $\epsilon_{\min} = \epsilon_0/300$ (the minimum exploration variance) $C_{\min}$, $C_{\max}$, $\mathcal{M}_{\text{A}}$, and $\mathcal{M}_{\text{RA}}$.
Initialize $\mathbf{w}_{Q,m}$, $\mathbf{w}_{\pi,m}$, $\mathbf{w}^{\text{cp}}_{Q,m}$, and $\mathbf{w}^{\text{cp}}_{\pi,m}$ for the DNN networks. Empty the replay buffer $\mathcal{B}_m$, $m \in \mathcal{M}_{\text{A}}$. The direction codebook $\mathcal{D} = \{\mathbf{d}_k\}_{k=1}^{K}$ is shared at both the BS and the IRS.
for $e = 0, \dots, N_{\text{episode}} - 1$ do
    Randomly generate the codebook $\mathcal{C}[0] = \{\mathbf{q}_m[0]\}_{m=1}^{M}$ satisfying (11). Update $\epsilon_e = \max\{\epsilon_{\min}, 0.99\epsilon_{e-1}\}$, if $e \geq 1$.
    for $t = 0, \dots, N_{\text{timestep}} - 1$ do
        Step 1. IRS channel sounding and reconfiguration. The IRS meta-atoms are tuned following $\{\mathbf{q}_m[t]\}_{m=1}^{M}$.
        Step 2. Codeword selection and inference at BS. The BS determines the index $m^\star[t] = \underset{m \in \{1,\dots,M\}}{\arg\max} R(\mathbf{q}_m[t], t)$. Each agent $m, m \in \mathcal{M}_{\text{A}}$, at the BS forms $\mathbf{s}_m[t] = \{\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t], t), \mathbf{q}_m[t]\}$ and determines $k_m[t] = \mu_{\mathcal{D},\pi}(\mathbf{s}_m[t])$ using (17), where $\mathbf{v}_m[t] \sim \mathcal{CN}(\mathbf{0}, \epsilon_e \mathbf{I})$.
        Step 3. Feedback to IRS and IRS final configuration. The BS feeds back $m^\star[t]$ and $\{k_m[t]\}_{m \in \mathcal{M}_{\text{A}}}$ to the IRS with $\lceil \log_2 M \rceil + M_{\text{A}} \lceil \log_2 K \rceil$ feedback bits. The IRS tunes the meta-atoms with $\mathbf{q}_\star[t] = \mathbf{q}_{m^\star[t]}[t]$ for data transmission.
        Step 4. Data transmission, IRS codebook update, and BS training. The IRS updates $\mathcal{C}[t+1] = \{\mathbf{q}_m[t+1]\}_{m=1}^{M}$, where the DPIC update is conducted by (13) for $m \in \mathcal{M}_{\text{A}}$, and the RA update by (12) for $m \in \mathcal{M}_{\text{RA}}$. Each agent $m \in \mathcal{M}_{\text{A}}$ at the BS computes $r_m[t-1]$ using (16), stores $(\mathbf{s}_m[t-1], \mathbf{a}_m[t-1], r_m[t-1], \mathbf{s}_m[t])$ in $\mathcal{B}_m$, samples $(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)$ from $\mathcal{B}_m$, and updates the DNN networks through (20), (22), (23)-(25).
    end for
end for
```

Algorithm 3 DNN policy-based IRS control (DPIC) approach in the utilization phase.

```
Input. $N_{\text{timestep}}$ (the duration of the algorithm), $C_{\min}$, $C_{\max}$, $\mathcal{M}_{\text{DPIC}}$, and $\mathcal{M}_{\text{RA}} = \{1, \dots, M\} \setminus \mathcal{M}_{\text{DPIC}}$.
The IRS randomly generates the codebook $\mathcal{C}[0] = \{\mathbf{q}_m[0]\}_{m=1}^{M}$ satisfying (11). The BS and IRS share $\mathcal{D} = \{\mathbf{d}_k\}_{k=1}^{K}$.
for $t = 0, \dots, N_{\text{timestep}} - 1$ do
    Step 1. IRS channel sounding and reconfiguration. The IRS meta-atoms are tuned following $\{\mathbf{q}_m[t]\}_{m=1}^{M}$.
    Step 2. Codeword selection and inference at BS. The BS determines $m^\star[t] = \underset{m \in \{1,\dots,M\}}{\arg\max} R(\mathbf{q}_m[t], t)$. Each agent $m \in \mathcal{M}_{\text{DPIC}}$ constructs $\mathbf{s}_m[t] = \{\mathbf{h}_{\text{eff}}(\mathbf{q}_m[t], t), \mathbf{q}_m[t]\}$ and determines $k_m[t] = \mu_{\mathcal{D},\pi}(\mathbf{s}_m[t])$ by (17).
    Step 3. Feedback to IRS and IRS final configuration. The BS feeds back $m^\star[t]$ and $\{k_m[t]\}_{m \in \mathcal{M}_{\text{DPIC}}}$ to the IRS with $\lceil \log_2 M \rceil + M_{\text{DPIC}} \lceil \log_2 K \rceil$ feedback bits. The IRS tunes its meta-atoms with $\mathbf{q}_\star[t] = \mathbf{q}_{m^\star[t]}[t]$ for data transmission.
    Step 4. Data transmission and IRS codebook update. The IRS updates $\mathcal{C}[t+1] = \{\mathbf{q}_m[t+1]\}_{m=1}^{M}$, where the DPIC update is conducted by (13) for $m \in \mathcal{M}_{\text{DPIC}}$, and the RA update by (12) for $m \in \mathcal{M}_{\text{RA}}$. The BS calculates and stores $\{\mathbf{q}_m[t+1]\}_{m \in \mathcal{M}_{\text{DPIC}}}$.
end for
```

updated by the DPIC update in (13) and M_{RA} codewords to be updated by the RA update in (12), where $M = M_{\text{DPIC}} + M_{\text{RA}}$. We develop four different strategies with different selections of M_{DPIC} and M_{RA}:

a) $M_{\text{DPIC}} = M$ and $M_{\text{A}} = 1$: a single agent handles M codeword updates. We call this case as *single-agent DPIC (SDPIC)*.

b) $M_{\text{DPIC}} = M$ and $M_{\text{A}} > 1$: multiple agents handles M codeword updates. We call this case as *multi-agents DPIC (MDPIC)*.

c) $M_{\text{DPIC}} < M$ and $M_{\text{A}} = 1$: a single agent handles M_{DPIC} codeword updates while $M_{\text{RA}} = M - M_{\text{DPIC}}$ codewords are updated by the RA update. We call this case as *RA+SDPIC*.

d) $M_{\text{DPIC}} < M$ and $M_{\text{A}} > 1$: multi-agents handles M_{DPIC} codeword updatse while $M_{\text{RA}} = M - M_{\text{DPIC}}$ codewords are updated by the RA update. We call this case as *RA+MDPIC*.

When multiple codewords are updated with multiple agents in MDPIC and RA+MDPIC, the BS allocates/partitions the codewords among the agents.[6] Let $\mathcal{M}_{\text{DPIC}}$ with $|\mathcal{M}_{\text{DPIC}}| = M_{\text{DPIC}}$ denote the set of indices of the codewords updated by the DPIC update during the utilization phase. We let $j[m] \in \mathcal{M}_{\text{A}}$ denote the agent handling the codeword $m \in \mathcal{M}_{\text{DPIC}}$. We take a simple round-robin strategy to allocate the codewords among the trained agents. If $M_{\text{A}} \leq M_{\text{DPIC}}$, some of the agents may handle multiple codewords. If $M_{\text{A}} > M_{\text{DPIC}}$, some trained agents are not used while each of the rest takes charge of one codeword independently. Utilizing more agents often improves performance due to the ensemble learning principle [47]. During the utilization phase, for the m-th codeword update, the BS determines $k_m[t]$ by using $\mathbf{v}_m[t] = 0$ (no random noise addition) and $\mathbf{w}_{\pi,j[m]}$ (instead of $\mathbf{w}_{\pi,m}$) in (17). While we refer to the aforementioned four cases as *DPIC approaches*, we only refer to MDPIC, RA+SDPIC, and RA+MDPIC as *augmented DPIC approaches* (SDPIC is excluded). The pseudo-code of the DPIC approach is given in Algorithm 3.

C. Computational Complexity and Group Control

We analyze the computational complexity measured in number of elementary operations performed by our algorithms at the BS and IRS in each channel coherence block. We first consider the RA approach (see Algorithm 1). In line 5, the BS calculates the data rate with the measured effective channel over a total of M codewords with $\mathcal{O}(MN_{\text{BS}})$ complexity. In line 7, the IRS updates the codebook with $\mathcal{O}(MN_{\text{IRS}})$ complexity.

We next consider the DPIC approach. The BS employs M_{A} agents each having four DNNs. For each DNN, we consider a fully connected neural network with two hidden layers, which have L_1 and L_2 neurons, respectively. For the DNN policy and copied DNN policy, the sizes of the input and output layer are $2N_{\text{BS}} + N_{\text{IRS}}$ and N_{IRS}, respectively. For the DNN Q-function and copied DNN Q-function, the sizes of the input and output layer are $2N_{\text{BS}} + N_{\text{IRS}}$ and 1, respectively. The actions are included at the second hidden layer. We first consider the training phase in Algorithm 2. In line 7, the BS infers $k_m[t]$ with the agent m, $m \in \mathcal{M}_{\text{A}}$. The computational complexity for total inference with M_{A} agents is thus $\mathcal{O}(M_{\text{A}}((2N_{\text{BS}}+N_{\text{IRS}})L_1 + L_1L_2 + L_2N_{\text{IRS}} + KN_{\text{IRS}}))$, which includes the quantization process for each inference with $\mathcal{O}(KN_{\text{IRS}})$ complexity. In line 9, the complexity to train the M_{A} agents each with mini-batch size N_{batch} is $\mathcal{O}(M_{\text{A}}N_{\text{batch}}((2N_{\text{BS}}+N_{\text{IRS}})L_1 + L_1L_2 + L_2N_{\text{IRS}}))$, which includes the updates for the DNN policy and DNN Q-function conducted via back-propagation and for the copied DNN policy and copied DNN Q-function conducted via soft target update. At the IRS, in line 9, the codebook update has the complexity of $\mathcal{O}(MN_{\text{IRS}})$. The computational complexity at the BS and IRS during the utilization phase is the same as that of the training phase with excluding the process of training the DNNs.

From individual meta-atom control to group control. Since the number of meta-atoms N_{IRS} is typically large, individual control for meta-atoms would incur high computational overhead at the BS and the IRS. To further reduce the computational overhead, we can consider a *group control* [17], where IRS meta-atoms are partitioned into multiple groups and the same capacitance is applied for the meta-atoms belonging to the same group. We denote the number of groups as N_{G}, where $N_{\text{G}} < N_{\text{IRS}}$. We then focus on controlling N_{G} capacitance values to configure N_{IRS} meta-atoms over the meta-surface. This implies that we can reduce the dimension of the design variables, i.e., capacitance vector and codeword, from N_{IRS} to N_{G}. Then, the computational complexity is reduced by replacing N_{IRS} with N_{G} in the complexity formula that we provided above. Furthermore, due to the group control, the DNN policy learning can be stabilized since the DNN policy learning has been successful when the action space size is not prohibitively large [45]. Due to these benefits, we incorporate the group control for our simulations in Sec. V.

D. Time Overhead and Effective Data Rate

The implementation of our methods incurs (i) computation time, (ii) communication time, and (iii) IRS reconfiguration time overheads. For the computations carried out during RA and DPIC, it is reasonable to assume that the BS calculates the data rate over M codewords within the time duration for M IRS reconfiguration by using its high computing power and that the IRS updates the codebook within the data transmission time. Also, in DPIC, we assume that the BS with high computational capabilities can conduct the total inference within M IRS reconfiguration time and the training in each coherence time. We accordingly neglect the computation time overhead and only focus on the communication time and IRS reconfiguration time.

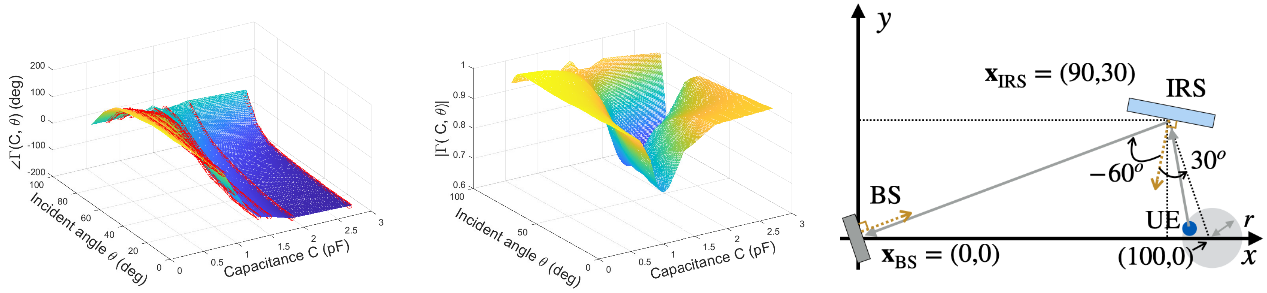
We define the *time overhead* T_p as the total time consumption except for data transmission shown in Fig. 2, given by

$$T_p = MT_{\text{reconf}} + T_{\text{feedback}} + T_{\text{final}}. \tag{26}$$

First, MT_{reconf} denotes the total time for M IRS reconfiguration used in both RA and DPIC approaches, where T_{reconf} is the time for each IRS reconfiguration. Second, $T_{\text{feedback}} = B/R_{\text{feedback}}$ is the time required for the feedback from the BS to the IRS, where B is the number of feedback bits during one coherence time T_c and R_{feedback} (bits/s) is the data rate for the feedback link. Note that $B < R_{\text{feedback}}T_c$ since T_{feedback} should not exceed T_c. For the RA approach, $B = \lceil \log_2 M \rceil$ for the feedback of $m^\star[t] \in \{1, ..., M\}$. For the DPIC approach, during the utilization time, $B = \lceil \log_2 M \rceil + M_{\text{DPIC}} \lceil \log_2 K \rceil$ for the feedback of $m^\star[t]$ and $\{k_m[t]\}_{m \in \mathcal{M}_{\text{DPIC}}}$. During the training period of DPIC, $B = \lceil \log_2 M \rceil + M_{\text{A}} \lceil \log_2 K \rceil$. Lastly, T_{final} denotes the execution time of the final IRS reconfiguration in **Step** 3 in Sec.III-B. If the selected index $m^\star[t]$ coincides with the last configuration in **Step** 1, the IRS does not need to change the configuration, i.e., $T_{\text{final}} = 0$; otherwise $T_{\text{final}} = T_{\text{reconf}}$.

To measure the average data rate during one coherence time T_c under time-varying channels, we introduce a performance metric, called *effective data rate*, according to

[6] Since the agents use identical learning structures, the data rate performance is mostly affected by the number of employed agents rather than different codeword-agent allocations among the same number of agents. We will see in Sec. V that increasing the number of agents improves data rate performance due to ensemble learning.



(a) Phase shift $\angle\Gamma(C,\theta)$

(b) Attenuation $|\Gamma(C,\theta)|$

(c) System configuration (top view)

Fig. 4: The IRS phase shift and attenuation data in (a) and (b), and the system configuration in (c).

$$R_{\text{eff}}[t] = \frac{T_c - T_p}{T_c} \log_2\left(1 + \frac{P\|\mathbf{h}_{\text{eff}}(\mathbf{q}_\star[t], t)\|_2^2}{\sigma^2}\right), \qquad (27)$$

where $T_c - T_p$ is the actual data transmission time and $\mathbf{q}_\star[t] \in \mathcal{C}[t]$ is the selected codeword for the final IRS configuration. The effective data rate captures the tradeoff between the data rate and the time overhead T_p. As M grows large, the data rate may increase due to the larger number of reconfigurations. However, as M increases, T_p also increases and, thus, $R_{\text{eff}}[t]$ may decrease. In Sec. V, we evaluate the data rate and effective data rate under different M.

V. Numerical Evaluation and Discussion

In this section, we describe the simulation setup in Sec. V-A and the channel model in Sec. V-B. We conduct simulations for two scenarios: (i) existence of no LoS link between the UE and IRS in Sec. V-C and (ii) existence of an LoS link between them in Sec. V-D. The former replicates a scenario with an indoor UE, while the later corresponds to an outdoor UE.

A. Simulation Setup

1) System parameters: To emulate realistic IRS reflection behavior, we interpolate/extrapolate the data in Fig. 4 and Table 1 of [14], and obtain the phase shift $\angle\Gamma(C,\theta)$ and attenuation $|\Gamma(C,\theta)|$, which are depicted in Fig. 4(a) and (b), respectively. Then, we obtain $\Gamma(C,\theta) = |\Gamma(C,\theta)| \exp(j\angle\Gamma(C,\theta))$ with the ranges of C and θ as $(C_{\min}, C_{\max}) = (0.4, 2.7)$ pF and $(0^o, 90^o)$, respectively. We follow the same simulation setup as in [14] to utilize the reflection coefficients. We set $f = 5.195$ GHz and consider only azimuth coordinates. We consider $T_c = 5$ ms, $N_{\text{BS}} = 5$ and $N_{\text{IRS}} = 200$, where the number of meta-atoms over the width and height of the IRS are $N_{\text{IRS,w}} = 50$ and $N_{\text{IRS,h}} = 4$, respectively. We consider a group control with $N_{\text{G}} = 10$, where $(N_{\text{IRS,w}}/N_{\text{G}}) \times N_{\text{IRS,h}}$ (i.e., 5×4) meta-atoms are controlled by each common capacitance. The distance between adjacent BS antennas is $d_{\text{BS}} = \lambda/2$, and the distance between adjacent IRS meta-atoms is $d_{\text{IRS}} = \lambda/10$, where $\lambda = c/f$ is the wavelength and c is the speed of light. Fig. 4(c) illustrates the system configuration for our simulations, where the BS, IRS, and UE are assumed to have the same height [3], [6], [7]. The BS and IRS are located at $\mathbf{x}_{\text{BS}} = (0,0)$ m and $\mathbf{x}_{\text{IRS}} = (90,30)$ m, respectively. The initial UE position $\mathbf{x}_{\text{UE}}[0]$ is randomly generated within the circle with radius $r = 5$ m at the center point $(100,0)$ m. The UE is moving with the velocity $v_{\text{UE}} = 3$ km/h and constant azimuth angle η over time, i.e., $\mathbf{x}_{\text{UE}}[t] = \mathbf{x}_{\text{UE}}[t-1] + v_{\text{UE}} T_c[\cos\eta, \sin\eta]^T$, where $\eta \sim \mathcal{U}(0, 2\pi)$ is generated. We set $P = 20$ dBm, $\sigma^2 = -80$ dBm, and $R_{\text{feedback}} = 10^6$ bits/s. To have more realistic results, we consider a noisy version of the effective channels in (6). Also, we set $T_{\text{reconf}} = 100\mu s$ [30], unless otherwise stated. Therefore, we conduct simulations with different T_{reconf} in Figs. 5(d)&6(d).[7]

2) Parameters for the proposed algorithms: For the RA algorithm, we set $\delta = (C_{\max} - C_{\min})/5$. For the DPIC algorithm, we set $\gamma = 0.9$, $\nu = 1$, $N_{\text{batch}} = 32$, and $|\mathcal{B}_m| = 5 \times 10^5$, $m \in \mathcal{M}_{\text{A}}$. We consider $L_1 = 400$ and $L_2 = 300$ for the DNNs with ReLU activation function. We employ the Adam optimizer for training. We consider $\alpha_\pi = 3 \times 10^{-4}$, $\alpha_Q = 3 \times 10^{-3}$, and $\tau = 0.005$. For the DNN policy, the input and output size is $2N_{\text{BS}} + N_{\text{G}} = 20$ and $N_{\text{G}} = 10$, respectively. In the output layer, the tanh function is employed, and the output is subsequently scaled by $\delta = (C_{\max} - C_{\min})/4$ to be bounded within $[-\delta, \delta]$. We set $|\mathcal{D}| = K = 2048$, where each codeword in $\mathcal{D}$ is constructed by RVQ ranging within $[-\delta, \delta]^{N_{\text{G}}}$.[8] For the DNN Q-function, the input and output sizes are 20 and 1, respectively. We normalize the values for the state and action to match with the scale of the values, such that $\mathbf{h}_{\text{eff}}(\cdot) \leftarrow \sqrt{P/(\sigma^2 N_{\text{BS}} N_{\text{G}})}\mathbf{h}_{\text{eff}}(\cdot)$ and $\mathbf{q}_m \leftarrow 10^{12}\mathbf{q}_m$ in (14), and $\mathbf{a}_m \leftarrow 10^{13}\mathbf{a}_m$ in (15).

B. Models for Channels and Their Variations

We adopt a multi-path geometric channel model [33] for the IRS-BS, UE-BS, and UE-IRS channels. In this model, a *vector channel* $\mathbf{h}[t]$ is constructed as the sum of the signals over multiple paths as $\mathbf{h}[t] = \sum_\ell \mathbf{h}_\ell[t]$ where the ℓ-th path channel $\mathbf{h}_\ell[t]$ is constructed with the path gain $g_\ell[t]$ and the angle (angle of arrival (AoA) or angle of departure (AoD)) $\theta_\ell[t]$ as $\mathbf{h}_\ell[t] = g_\ell[t]\text{ARV}(\theta_\ell[t])$ with the array response vector $\text{ARV}(\theta_\ell[t])$. Subsequently, a *matrix channel* is constructed similarly with path gains, AoAs and AoDs of multiple paths. To model channel variations, we consider that $g_\ell[t]$ evolves over time according to a first-order Gauss-Markov process [48], and $\theta_\ell[t]$ varies via random perturbation addition given by

$$g_\ell[t] = \rho g_\ell[t-1] + \sqrt{1-\rho^2}\nu_\ell[t],$$

[7] The reconfiguration time of IRS is determined by the characteristics of the control board and the internal communication between the control board and the meta-surface. The reconfiguration speed is typically known to be a few kHz [30].

[8] Given M, we can determine K experimentally by implementing the agents with different values of K for a short period of time in the beginning of the utilization phase, obtaining their corresponding data rates, calculating the effective data rates using (26)-(27), and choosing the value of K (for the rest of the utilization phase) that has the highest performance.

$$\theta_\ell[t] = \theta_\ell[t-1] + \Delta\theta_\ell[t]. \tag{28}$$

In (28), the time correlation coefficient ρ obeys the Jakes model [48], i.e., $\rho = J_0(2\pi f_d T_c)$, where $J_0(\cdot)$ is the zeroth order Bessel function of the first kind and $f_d = vf/c$ is the maximum Doppler frequency corresponding to the velocity v of the UE or scatterer. For simulations, we set $v = 3$ km/h and accordingly obtain $\rho = 0.95$. Also, $\Delta\theta_\ell[t] \sim \mathcal{U}(-0.1^o, 0.1^o)$, and we set $\nu_\ell[t] \sim \mathcal{CN}(0, \beta[t])$ and $g_\ell[0] \sim \mathcal{CN}(0, \beta[0])$ with $\beta[t]$ denoting the instantaneous large-scale fading factor, which is defined with Euclidean distance $d[t]$ between two network elements as

$$\beta[t] = \beta_0 - 10\alpha \log_{10}(d[t]/d_0) \text{ (dB)}, \tag{29}$$

where $\beta_0 = -30$ dB is the path loss at distance $d_0 = 1$ m [3], [6] and α is the path loss exponent.

1) IRS-BS channel: Since the IRS is deployed to have an LoS path to the BS [4], [5], we model the IRS-BS channel with the Rician channel [33] as $\mathbf{H}^{\text{IB}}[t] = \sqrt{\frac{K^{\text{IB}}}{1+K^{\text{IB}}}}\mathbf{H}_0^{\text{IB}} + \sqrt{\frac{1}{1+K^{\text{IB}}}}\sum_{\ell=1}^{L^{\text{IB}}} \mathbf{H}_\ell^{\text{IB}}[t]$ where $K^{\text{IB}} = 5$. The LoS channel $\mathbf{H}_0^{\text{IB}}$ is characterized by the AoA and AoD, set to 0^o and -60^o, respectively. The number of non-LoS (NLoS) paths is $L^{\text{IB}} = 10$. The variations of the path gain, AoA, and AoD for each NLoS channel $\mathbf{H}_\ell^{\text{IB}}[t]$ are modeled by (28), where the initial AoA and AoD are generated from $\mathcal{U}(-90^o, 90^o)$. The large-scale fading coefficient is modeled by (29) using the fixed distance between the IRS and the BS, and $\alpha = 2$.

2) UE-BS channel: Assuming that there exists a blockage between the UE and the BS, we model the UE-BS channel with only NLoS signals as $\mathbf{h}^{\text{UB}}[t] = \sum_{\ell=1}^{L^{\text{UB}}} \mathbf{h}_\ell^{\text{UB}}[t]$, where $L^{\text{UB}} = 10$. The variations of the path gain and AoA for each channel $\mathbf{h}_\ell^{\text{UB}}[t]$ are modeled by (28), where the initial AoA is generated from $\mathcal{U}(-90^o, 90^o)$. The large-scale fading factor is modeled by the varying distance between the UE and the BS, and $\alpha = 3.75$.

3) UE-IRS channel: We consider two different scenarios. In the first scenario, there exists no LoS link between the UE and the IRS, for which we model the UE-IRS channel with only NLoS signals as $\mathbf{h}^{\text{UI}}[t] = \sum_{\ell=1}^{L} \mathbf{h}_\ell^{\text{UI}}[t]$ where $L = 10$. The variations of path gain and AoA for each channel $\mathbf{h}_\ell^{\text{UI}}[t]$ are modeled by (28), where the initial AoA is generated from $\mathcal{U}(0^o, 90^o)$. The large-scale fading factor is modeled by the varying distance between the UE and the IRS, and $\alpha = 2.2$. In the second scenario, there exists an LoS between the UE and IRS, for which we model the UE-IRS channel with the Rician channel as $\mathbf{h}^{\text{UI}}[t] = \sqrt{\frac{K^{\text{UI}}}{1+K^{\text{UI}}}}\mathbf{h}_0^{\text{UI}}[t] + \sqrt{\frac{1}{1+K^{\text{UI}}}}\sum_{\ell=1}^{L} \mathbf{h}_\ell^{\text{UI}}[t]$, where $K^{\text{UI}} = 1$ and $L = 10$. The LoS channel $\mathbf{h}_0^{\text{UI}}[t]$ is varying according to the UE movement described in Sec.V-A1 modeled via the changing AoA and distance between the UE and the IRS. The NLoS channel $\mathbf{h}_\ell^{\text{UI}}[t]$ is modeled as in the first scenario.

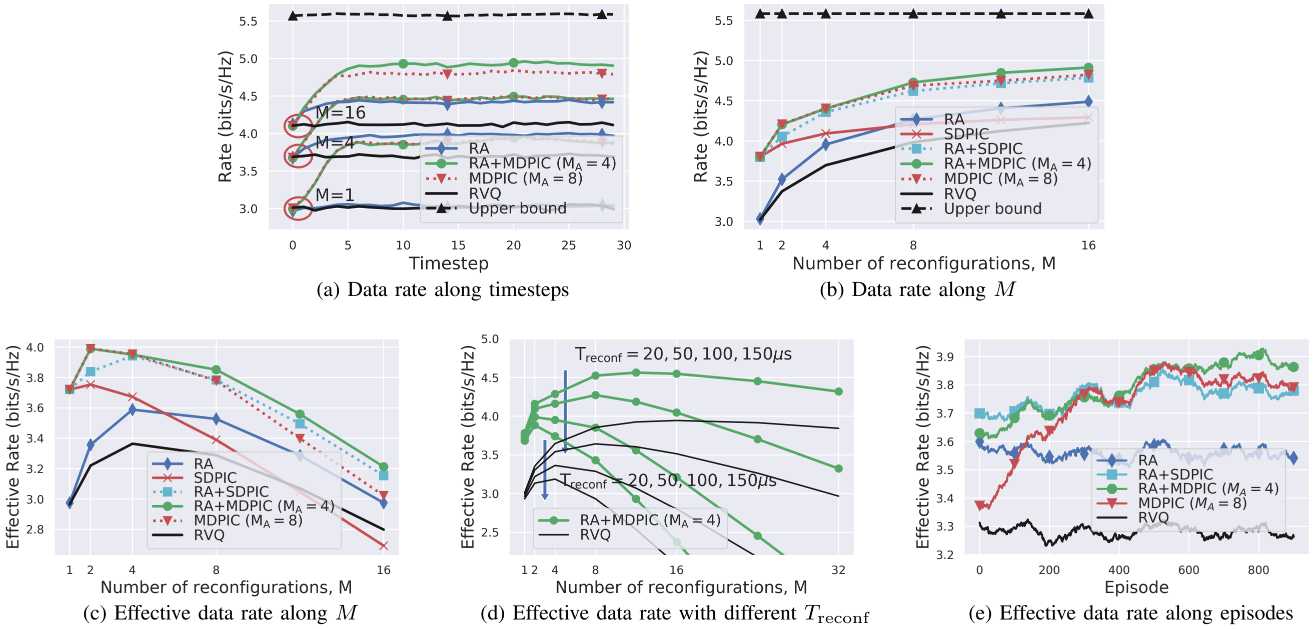
C. Scenario 1. No LoS Link between the UE and the IRS

Scenario 1 represents an *indoor UE*, which does not have a LoS link to the IRS. We first evaluate the performance of the proposed algorithms in the utilization phase with 2000 episodes, where each episode contains 30 timesteps (coherence blocks). Each episode has different realizations of the UE-IRS channel, UE-BS channel, IRS-BS channel, UE initial location, and UE moving direction. Our baseline method is the RVQ codebook described in Sec. III-C, denoted by "RVQ" in the figures. For benchmarking purposes, we consider an experimental upper bound obtained by conducting 10^4 random searches over the solution space $[C_{\min}, C_{\max}]^{N_{\text{G}}}$ and selecting the solution maximizing the data rate. The specific configuration of the proposed schemes is $M_{\text{A}} = 1$ for SDPIC, $M_{\text{A}} = 8$ for MDPIC, $M_{\text{A}} = 1$, $M_{\text{DPIC}} = 1$, and $M_{\text{RA}} = M - 1$ for RA+SDPIC, and $M_{\text{A}} = 4$, $M_{\text{DPIC}} = \min\{M, M_{\text{A}}\}$, and $M_{\text{RA}} = M - M_{\text{DPIC}}$ for RA+MDPIC.

Fig. 5(a) shows the average data rate along the timesteps over 2000 episodes. The proposed schemes – RA, RA+MDPIC, and MDPIC – update the codebook adaptively at every timestep t by using current observations (i.e., previously used codewords, effective channels, and data rates) to improve the data rate for the next timestep $t+1$. The performances of the proposed schemes are improved over time and converge only within 4-5 timesteps. Overall, as the number of IRS reconfigurations M increases, a higher data rate is achieved. Interestingly, RA+MDPIC yields better data rate compared to that of MDPIC and RA because the multiple trained agents give good update directions, and RA further improves the performance via random exploration around the previous solution. Fig. 5(b) shows the data rate along M, where each data point is averaged over 2000 episodes and 30 timesteps. MDPIC yields a better data rate than that of SDPIC, due to the advantage of using multiple agents. Among the methods, RA+MDPIC yields the best performance as the same in Fig. 5(a). We see that RA+MDPIC obtains within 10% of the performance of the experimental upper bound as the number of timesteps and reconfigurations increases, indicating strong performance.

Fig. 5(c) shows the effective data rate along M. The effective data rate in (27) captures the tradeoff between the data rate and the time overhead. As M gets large, the data rate may increase due to having larger number of reconfigurations; however, at the same time, a larger M increases the total time overhead. RA+MDPIC shows the best performance in effective data rate for any M. We obtain the highest effective data rate when $M^\star = 2$ or $M^\star = 4$ depending on the method. As M grows larger than 2 or 4, the increased time overhead outweighs the improvement of the data rate, leading to the decrease of the effective data rate. This finding agrees with recent results from [19], where the performance of the overhead-aware metric is degraded as the overhead for the channel sounding and feedback increases. Fig. 5(d) shows the effective data rate of RA+MDPIC along M with different T_{reconf}. For $T_{\text{reconf}} = 20, 50, 100, 150\mu s$, the best M yielding the highest effective data rate is $M^\star = 12, 8, 2, 2$, respectively. For larger T_{reconf}, smaller M is preferred since a large T_{reconf} implies a large time overhead for each IRS reconfiguration. For smaller T_{reconf}, larger M is preferred since more IRS reconfiguration increases the data rate. Although finding optimal $M^\star$ in advance is challenging due to the difficulty of the analysis on agents' inferences, we could set a proper range of M empirically from the value of T_{reconf}.

We next focus on the training phase with 1000 episodes



(a) Data rate along timesteps

(b) Data rate along M

(c) Effective data rate along M

(d) Effective data rate with different T_{reconf}

(e) Effective data rate along episodes

Fig. 5: Performance evaluation of our methodology in Scenario 1. The plots in (a)-(d) correspond to the utilization phase, while the plot in (e) describes the training phase.

each containing 500 timesteps and $M = 8$. Fig. 5(e) shows the effective data rate averaged over 500 timesteps for each episode. Each data point is a moving average over the previous 100 episodes. For the training, the BS determines the number of agents being trained, M_{A}. If $M_{\text{A}} = M = 8$, all codewords are dedicated to training the agents, denoted by MDPIC ($M_{\text{A}} = 8$). At the beginning of the training, the performance of MDPIC is similar to RVQ but is improved over time. If $M_{\text{A}} < M$, M_{A} codewords are dedicated to training M_{A} agents while $M - M_{\text{A}}$ codewords are updated by the RA update, which are the cases of RA+SDPIC and RA+MDPIC ($M_{\text{A}} = 4$). These *hybrid* approaches use random exploration that supplements the low initial data rate of MDPIC. The performance of RA+MDPIC is better than that of MDPIC despite of using less agents, even after the completion of the training, which agrees with the result in the utilization phase in Fig. 5(c).

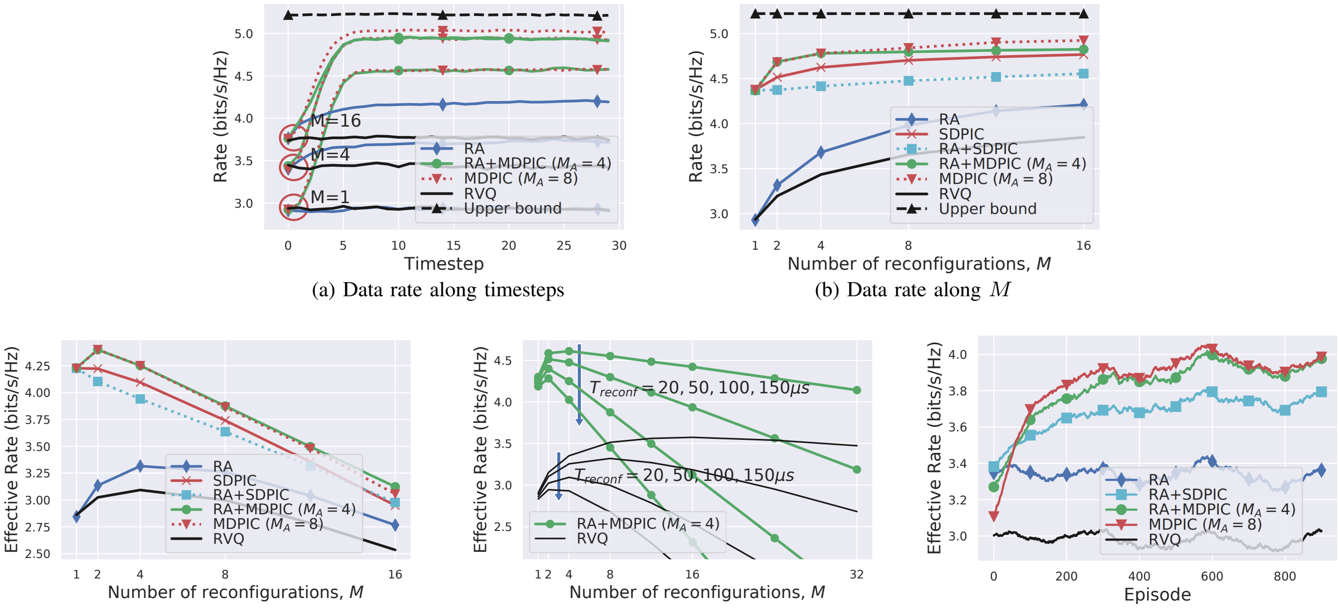
D. Scenario 2. LoS Link between the UE and the IRS

Scenario 2 represents an *outdoor UE* with an LoS link to the IRS, for which we follow the same order of the simulations and the same configuration for the proposed schemes as in the previous scenario. While the UE-IRS channel in Scenario 2 is subject to less variations than in Scenario 1, it still corresponds to a dynamic environment which is challenging to address, because the LoS channel between the UE and IRS is different per episode due to the different UE position and varies over timesteps due to the UE mobility.

We consider 2000 episodes each containing 30 timesteps for the utilization phase. Fig. 6(a) shows the average data rate along the timesteps over 2000 episodes. The performances of the proposed schemes are improved and converge only within 4-5 timesteps. RA+MDPIC and MDPIC yield better performances than other methods by a large margin due to the exploitation of the multiple trained agents. Fig. 6(b) shows the data rate along the number of IRS reconfiguration M, where each data is averaged over 2000 episodes and 30 timesteps. MDPIC is slightly better than RA+MDPIC, while the opposite is the case in Scenario 1. One possible explanation for this is that the agents can learn better in less complex environments; the channel variations in Scenario 2 are less complex than in Scenario 1, because the channel is mainly dominated by the LoS path signal. Thus, relying on more agents, i.e., MDPIC, may yield better performance. Also, due to this fact, the data rate performances in Scenario 2 are closer to the upper bound than in Scenario 1, with MDPIC reaching within 5%.

In Fig. 6(c), MDPIC and RA+MDPIC yield better effective data rate performances than those of RA and RA+SDPIC, although an additional feedback overhead is required due to the DPIC updates. However, RA+MDPIC is slightly better than MDPIC because less agents in RA+MDPIC require less feedback overhead for the codebook update. At $M^{\star} = 2$ or $M^{\star} = 4$ depending on the method, the effective data rate is the highest, which means that, as M grows larger than 2 or 4, the increased overhead outweighs the improvement of the data rate. Fig. 6(d) shows the effective data rate along M with different T_{reconf}. With the same reason in Fig. 5(d), $M^{\star}$ would decrease as T_{reconf} increases, where $M^{\star} = 4, 2, 2, 2$ for $T_{\text{reconf}} = 20, 50, 100, 150\mu s$, respectively. We next focus on the training phase with 1000 episodes each consisting of 500 timesteps and $M = 8$. Fig. 6(e) shows the effective data rate averaged over 500 timesteps per episode. Each data point is a moving average of the previous 100 episodes. Similar to Scenario 1, RA+MDPIC yields the best performance during training.

Discussion on comprehensive strategy. Utilizing our learning-based method, i.e., the DPIC algorithm, is preferred for both the NLoS and LoS scenarios. In practice, the BS



(a) Data rate along timesteps

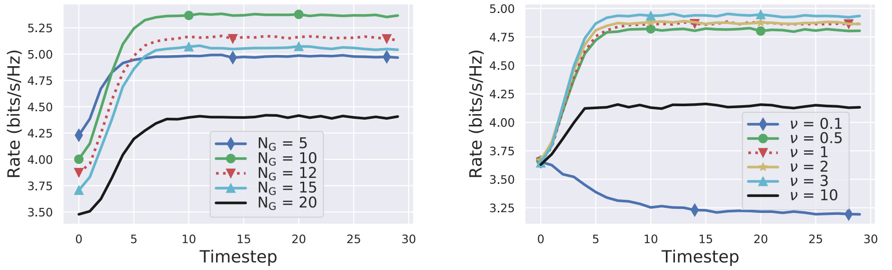
(b) Data rate along M

(c) Effective data rate along M

(d) Effective data rate with different T_{reconf}

(e) Effective data rate along episodes

Fig. 6: Performance evaluation of our methodology in Scenario 2. The plots in (a)-(d) correspond to the utilization phase, while the plot in (e) to the training phase.



(a) Group control with N_{G} groups

(b) Penalty scaled by ν

Fig. 7: Comparison of the data rate performance with different learning configuration parameters.

may not be aware of whether there exists an LoS link from the UE to the IRS. The BS can thus select RA+MDPIC as a comprehensive strategy because it yields a satisfactory performance in either of the scenarios. RA+MDPIC also has advantages during the training phase since (i) it only requires just a few agents to be trained, leading to less burden for training, and (ii) it exhibits a high performance during the training phase.

E. Impact of Learning Configuration Parameters

Finally, we investigate the impact of different learning configuration parameters in the DPIC scheme on the data rate performance. The simulation parameters/setup are the same as before, unless otherwise stated. For performance comparisons, we employ MDPIC and consider Scenario 2 with $M = 8$. Fig. 7(a) shows the data rate under different choices of the number of groups N_{G} with $N_{\mathrm{IRS}} = 240$. With a very large value of N_{G} (i.e., $N_{\mathrm{G}} = 20$), the learning process does not result in high performance due to the large action space [45]. With a very small value of N_{G} (i.e., $N_{\mathrm{G}} = 5$), a high data rate is not achieved either due to the low degree of freedom for IRS control. The performance is the highest at $N_{\mathrm{G}} = 10$, which balances the learning effectiveness with the degree of freedom for the IRS control. Fig. 7 (b) shows the effect of ν in (16) on the data rate performance. Considering the data rate ranges from 1 to 10 bits/s/Hz, $\nu = 0.5, 1, 2, 3$ result in the same order-of-magnitude between the data rate and the penalty value, for which we find the data rate is notably improved. When trained with $\nu = 0.1$, the agents tend to output the largest action values so that the updated capacitance stays at the boundary of the feasible capacitance region over time. When trained with $\nu = 10$, the agents are strongly discouraged from outputting the actions that result in the updated capacitance close to the boundary of the feasible capacitance region. Thus, with either very small or very large ν, the training for the agents is ineffective due to limited state and action exploration. In practice, we can select N_{G} and ν experimentally by investigating the learning effectiveness, i.e., the data rate performance, under different choices of N_{G} and ν during the training phase of our methodology.

VI. Conclusion

In this paper, we introduced a novel signal model that takes into account the practical IRS reflection behavior. To address the design challenges associated with the (i) practical IRS reflection behavior, (ii) multi-path time-varying channels, and (iii) low-overhead feedback requirement, we proposed an adaptive codebook-based limited feedback protocol for the IRS-assisted communication. We proposed two adaptive codebook design approaches: random adjacency (RA) and deep neural network policy-based IRS control (DPIC). Then, we discussed the computational complexity of RA and DPIC. Further, we developed several augmented schemes based on DPIC. Throughout the simulations, we showed that the data rate performance is improved by the proposed schemes. In addition, we demonstrated that the average data rate over one coherence time is degraded when the time overhead for the IRS reconfiguration and feedback increases.

References

[1] J. Zhang, E. Björnson, M. Matthaiou, D. W. K. Ng, H. Yang, and D. J. Love, “Prospective multiple antenna technologies for beyond 5G,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1637–1660, Jun. 2020.

[2] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, “From federated to fog learning: Distributed machine learning over heterogeneous wireless networks,” *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 41–47, Dec. 2020.

[3] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, “Reconfigurable intelligent surfaces for energy efficiency in wireless communication,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, Jun. 2019.

[4] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, “A new wireless communication paradigm through software-controlled metasurfaces,” *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 162–169, Sep. 2018.

[5] Q. Wu and R. Zhang, “Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network,” *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, Nov. 2019.

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

[7] M. Cui, G. Zhang, and R. Zhang, “Secure wireless communication via intelligent reflecting surface,” *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1410–1414, May 2019.

[8] H. Rajagopalan and Y. Rahmat-Samii, “Loss quantification for microstrip reflectarray: Issue of high fields and currents,” in *IEEE Antennas and Propag. Society Int. Symp.*, Jul. 2008, pp. 1–4.

[9] B. O. Zhu, J. Zhao, and Y. Feng, “Active impedance metasurface with full 360 reflection phase tuning,” *Sci. Rep.*, vol. 3, p. 3059, Oct. 2013.

[10] L. Shao and W. Zhu, “Electrically reconfigurable microwave metasurfaces with active lumped elements: A mini review,” *Front. Mater.*, vol. 8, p. 212, Jun. 2021.

[11] S. Abeywickrama, R. Zhang, Q. Wu, and C. Yuen, “Intelligent reflecting surface: Practical phase shift model and beamforming optimization,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5849–5863, Sep. 2020.

[12] W. Tang, J. Y. Dai, M. Z. Chen, K.-K. Wong, X. Li, X. Zhao, S. Jin, Q. Cheng, and T. J. Cui, “MIMO transmission through reconfigurable intelligent surface: System design, analysis, and implementation,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2683–2699, Jul. 2020.

[13] W. Tang, M. Z. Chen, X. Chen, J. Y. Dai, Y. Han, M. Di Renzo, Y. Zeng, S. Jin, Q. Cheng, and T. J. Cui, “Wireless communications with reconfigurable intelligent surface: Path loss modeling and experimental measurement,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 421–439, Sep. 2020.

[14] W. Chen, L. Bai, W. Tang, S. Jin, W. X. Jiang, and T. J. Cui, “Angle-dependent phase shifter model for reconfigurable intelligent surfaces: Does the angle-reciprocity hold?” *IEEE Commun. Lett.*, May 2020.

[15] X. Pei, H. Yin, L. Tan, L. Cao, Z. Li, K. Wang, K. Zhang, and E. Björnson, “RIS-aided wireless communications: Prototyping, adaptive beamforming, and indoor/outdoor field trials,” *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8627–8640, Sep. 2021.

[16] F. Costa and M. Borgese, “Electromagnetic model of reflective intelligent surfaces,” *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1577–1589, Jun. 2021.

[17] Y. Yang, B. Zheng, S. Zhang, and R. Zhang, “Intelligent reflecting surface meets OFDM: Protocol design and rate maximization,” *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4522–4535, Mar. 2020.

[18] L. Wei, C. Huang, G. C. Alexandropoulos, C. Yuen, Z. Zhang, and M. Debbah, “Channel estimation for RIS-empowered multi-user MISO wireless communications,” *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 4144–4157, Mar. 2021.

[19] A. Zappone, M. Di Renzo, F. Shams, X. Qian, and M. Debbah, “Overhead-aware design of reconfigurable intelligent surfaces in smart radio environments,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 126–141, Sep. 2020.

[20] J. He, H. Wymeersch, T. Sanguanpuak, O. Silvén, and M. Juntti, “Adaptive beamforming design for mmWave RIS-aided joint localization and communication,” in *IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2020, pp. 1–6.

[21] J. Kim, S. Hosseinalipour, T. Kim, D. J. Love, and C. G. Brinton, “Multi-IRS-assisted multi-cell uplink MIMO communications under imperfect CSI: A deep reinforcement learning approach,” in *IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–7.

[22] J. An, C. Xu, L. Gan, and L. Hanzo, “Low-complexity channel estimation and passive beamforming for RIS-assisted MIMO systems relying on discrete phase shifts,” *IEEE Trans. Commun.*, Nov. 2021.

[23] C. Psomas and I. Krikidis, “Low-complexity random rotation-based schemes for intelligent reflecting surfaces,” *IEEE Trans. Wireless Commun.*, Mar. 2021.

[24] D. J. Love, R. W. Heath, V. K. Lau, D. Gesbert, B. D. Rao, and M. Andrews, “An overview of limited feedback in wireless communication systems,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, pp. 1341–1365, Oct. 2008.

[25] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology*. Academic Press, Sep. 2020.

[26] D. J. Love and R. W. Heath, “Equal gain transmission in multiple-input multiple-output wireless systems,” *IEEE Trans. Commun.*, vol. 51, no. 7, pp. 1102–1110, Jul. 2003.

[27] C. K. Au-Yeung and D. J. Love, “On the performance of random vector quantization limited feedback beamforming in a MISO system,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 458–462, Feb. 2007.

[28] W. Santipach and M. L. Honig, “Capacity of a multiple-antenna fading channel with a quantized precoding matrix,” *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1218–1234, Feb. 2009.

[29] R. Mudumbai, G. Barriac, and U. Madhow, “On the feasibility of distributed beamforming in wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1754–1763, May 2007.

[30] S. Abadal, T.-J. Cui, T. Low, and J. Georgiou, “Programmable metamaterials for software-defined electromagnetic control: Circuits, systems, and architectures,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 1, pp. 6–19, Feb. 2020.

[31] S. Koziel and L. Leifsson, *Surrogate-based modeling and optimization*. Springer, Jun. 2013.

[32] D. M. Pozar, *Microwave engineering*. John wiley & sons, Nov. 2011.

[33] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, May 2005.

[34] 3GPP TS 36.211, “LTE: Evolved universal terrestrial radio access (E-UTRA): Physical channels and modulation,” vol. V14.2.0 Release 14, Mar. 2017.

[35] A. Ghosh, “5G new radio (NR): physical layer overview and performance,” in *Proc. IEEE Commun. Theory Workshop*, May 2018, pp. 1–38.

[36] V. Lau, Y. Liu, and T.-A. Chen, “On the design of MIMO block-fading channels with feedback-link capacity constraint,” *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 62–70, Mar. 2004.

[37] A. Narula, M. J. Lopez, M. D. Trott, and G. W. Wornell, “Efficient use of side information in multiple-antenna data transmission over fading channels,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1423–1436, Oct. 1998.

[38] J. C. Roh and B. D. Rao, “Transmit beamforming in multiple-antenna systems with finite rate feedback: A VQ-based approach,” *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1101–1112, Mar. 2006.

[39] D. J. Love, R. W. Heath, and T. Strohmer, “Grassmannian beamforming for multiple-input multiple-output wireless systems,” *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2735–2747, Oct. 2003.

[40] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, “On beamforming with finite rate feedback in multiple-antenna systems,” *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2562–2579, Oct. 2003.

[41] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. 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, Sep. 2013.

[42] T. Kim, D. J. Love, and B. Clerckx, "MIMO systems with limited rate differential feedback in slowly varying channels," *IEEE Trans. Commun.*, vol. 59, no. 4, pp. 1175–1189, Mar. 2011.

[43] B. Mondal and R. W. Heath, "Channel adaptive quantization for limited feedback MIMO beamforming systems," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4717–4729, Nov. 2006.

[44] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, Nov. 2018.

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

[46] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Int. Conf. on Mach. Learn. (ICML)*, Jun. 2014, pp. 387–395.

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, Nov. 2016.

[48] B. Sklar *et al.*, *Digital communications: fundamentals and applications*, 2001.

Junghoon Kim (S'19) received the B.S. degree from Yonsei University, Seoul, South Korea, in 2014, and M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016, both in electrical engineering. Starting from fall 2019, he has worked towards the Ph.D. degree at Purdue University. He was a research intern at the Korea Institute of Science and Technology (KIST), Seoul, South Korea, from 2017 to 2018, and a research assistant at KAIST from 2018 to 2019, respectively. He was the recipient of the Best Application Paper Award in the 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) in 2017.

Seyyedali Hosseinalipour (S'17, M'20) (hosseina@purdue.edu) received the B.S. degree in electrical engineering from Amirkabir University of Technology, Tehran, Iran, in 2015, and the M.S. and Ph.D. degrees in electrical engineering from North Carolina State University, NC, USA, in 2017 and 2020, respectively. He is currently a postdoctoral researcher at Purdue University, IN, USA. He was the recipient of the ECE Doctoral Scholar of the Year Award (2020) and ECE Distinguished Dissertation Award (2021) at North Carolina State University. He has served as the TPC Co-Chair of "The First International Workshop on Distributed Machine Learning and Fog Networks (FogML)", held in conjunction with IEEE INFOCOM 2021, TPC Co-Chair of "2nd Workshop on Edge Learning Over 5G Mobile Networks and Beyond (EL5GMNB)", held in conjunction with IEEE GLOBECOM 2021, TPC member of "Edge Machine Learning for 5G Mobile Networks and Beyond (EdgeLearn5G)", held in conjunction with IEEE ICC 2021, and Program Committee Member of "Edge Computing and IoT", IEEE MSN. His research interests include the analysis of modern wireless networks and communication systems, distributed machine learning, and network optimization.

Andrew C. Marcum (M'16) received the B.S. degree in electrical engineering from the Rose–Hulman Institute of Technology, Terre Haute, IN, USA, in 2006, the M.S. degree in engineering with focus in systems engineering and electrical engineering from the Purdue University, Fort Wayne, IN, USA, in 2010, and the Ph.D. degree in electrical engineering with focus in communications and signal processing from the Purdue University, West Lafayette, IN, USA, in 2015. Currently, Dr. Marcum is a Lead Scientist and Engineering Fellow at Raytheon BBN Technologies (BBN), Arlington, VA, USA. At BBN, Dr. Marcum serves as the technical director for BBN's emerging RF technologies (ERFT) line of business. In this role, Dr. Marcum has served (and continues to serve) as the principle investigator for numerous US Government sponsored research and development programs. From 2006 to 2016, he served as a Systems Engineer at Raytheon Company, Fort Wayne, IN, USA. Dr. Marcum's research interests center around the application and development of statistical and cyclostationary signal processing techniques for distributed coherent RF systems, passive RF systems, wireless communications, radar, intelligent control systems, and electronic warfare systems.

Taejoon Kim (S'8, M'11, SM'21) received the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN. He is currently an Assistant Professor and Chair's Council Faculty at the University of Kansas (KU). Prior to joining KU, he was a Senior Researcher at the Nokia Bell Labs, Berkeley, CA, a Postdoctoral Researcher at KTH, Stockholm, Sweden, and an Assistant Professor at City University of Hong Kong. His research interest includes 5G-and-beyond wireless systems, multiple-input multiple-output (MIMO) communications, optimization theory, machine learning, and security. He was an Associate Editor for the IEEE Transactions on Communications and previously served as a Guest Editor of IEEE Transactions on Industrial Informatics. He holds 29 issued U.S. patents. Dr. Kim was the recipient of the Harry Talley Excellence in Teaching Award (2022) and Miller Faculty Award (2022 and 2021) from KU. He received The President's Award (2017 from City University of Hong Kong). Along with the coauthors, he won the best paper award of IEEE Transactions on Communications (2016 Stephen O. Rice Prize) and IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) Best Paper award (2012).

David J. Love (S'98 - M'05 - SM'09 - F'15) received the B.S. (with highest honors), M.S.E., and Ph.D. degrees in electrical engineering from the University of Texas at Austin in 2000, 2002, and 2004, respectively. Since 2004, he has been with the Elmore Family School of Electrical and Computer Engineering at Purdue University, where he is now the Nick Trbovich Professor of Electrical and Computer Engineering. He served as a Senior Editor for IEEE Signal Processing Magazine, Editor for the IEEE Transactions on Communications, Associate Editor for the IEEE Transactions on Signal Processing, and guest editor for special issues of the IEEE Journal on Selected Areas in Communications and the EURASIP Journal on Wireless Communications and Networking. He was a member of the Executive Committee for the National Spectrum Consortium. He holds 32 issued U.S. patents. His research interests are in the design and analysis of broadband wireless communication systems, beyond-5G wireless systems, multiple-input multiple-output (MIMO) communications, millimeter wave wireless, software defined radios and wireless networks, coding theory, and MIMO array processing.

Dr. Love was named a Thomson Reuters Highly Cited Researcher (2014 and 2015), is a Fellow of the Royal Statistical Society, and has been inducted into Tau Beta Pi and Eta Kappa Nu. Along with his co-authors, he won best paper awards from the IEEE Communications Society (2016 Stephen O. Rice Prize and 2020 Fred W. Ellersick Prize), the IEEE Signal Processing Society (2015 IEEE Signal Processing Society Best Paper Award), and the IEEE Vehicular Technology Society (2010 Jack Neubauer Memorial Award).

Christopher G. Brinton (S'08, M'16, SM'20) is an Assistant Professor in the School of Electrical and Computer Engineering at Purdue University. His research interest is at the intersection of networked systems and machine learning, specifically in distributed machine learning, fog/edge network intelligence, and data-driven network optimization. Dr. Brinton was the recipient of the 2022 NSF CAREER Award, 2022 ONR Young Investigator Program (YIP) Award, and 2022 DARPA Young Faculty Award (YFA). He currently serves as an Associate Editor for IEEE Transactions on Wireless Communications, in the ML and AI for wireless area. Prior to joining Purdue, Dr. Brinton was the Associate Director of the EDGE Lab and a Lecturer of Electrical Engineering at Princeton University. Dr. Brinton received the PhD (with honors) and MS Degrees from Princeton in 2016 and 2013, respectively, both in Electrical Engineering.