Received 15 November 2020; revised 17 December 2020; accepted 5 January 2021. Date of publication 8 January 2021; date of current version 12 February 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3050119

A Deep Learning Based Modeling of Reconfigurable Intelligent Surface Assisted Wireless Communications for Phase Shift Configuration

BAOLING SHEEN [1], JIN YANG [1], XIANGLONG FENG [2], AND MD MOIN UDDIN CHOWDHURY [3]

[1] Radio Access Standards Department, Futurewei Technologies, Bridgewater, NJ 08807, USA

[2] Electrical and Computer Engineering Department, Rutgers University, New Brunswick, NJ 08901, USA

[3] Electrical and Computer Engineering Department, North Carolina State University, Raleigh, NC 27695, USA

CORRESPONDING AUTHOR: B. SHEEN (e-mail: baoling.sheen@futurewei.com)

This work was supported by Futurewei internal research.

ABSTRACT Reconfigurable Intelligent Surface (RIS) has emerged as a promising technology in wireless networks to achieve high spectrum and energy efficiency. RIS typically comprises a large number of low-cost nearly passive elements that can smartly interact with the impinging electromagnetic waves for performance enhancement. However, optimally configuring massive number of RIS elements remains a challenge. In this article, we present a machine learning (ML) based modeling approach that learns the interactions between the phase shifts of the RIS elements and receiver (Rx) location attributes and uses them to predict the achievable rate directly without using channel state information (CSI) or received pilots. Once learned, our model can be used to predict optimal RIS configuration for any new receiver locations in the same wireless network. We leverage deep learning (DL) techniques to build our model and study its performance and robustness. Simulation results demonstrate that the proposed DL model can recommend near-optimal RIS configurations for test receiver locations which achieved close to an upper bound performance that assumes perfect channel knowledge. Our DL model was trained using less than 2% of the total receiver locations. This promising result represents great potential in developing a practical solution for the optimal phase shifts of RIS elements without requesting CSI from the wireless network infrastructure.

INDEX TERMS Reconfigurable intelligent surface, large intelligent surface, deep learning, channel estimation.

I. INTRODUCTION

RIS (Reconfigurable Intelligent Surfaces) has been envisioned as a promising technology to reduce the energy consumption and improve the communication performance by artificially reconfiguring the propagation environment of electromagnetic (EM) waves. As such, RISs have the huge potential to revolutionize the design of wireless networks to realize smart radio environments [1], particularly when combined and integrated together with other candidate technologies for the next generation networks, such as terahertz communications and artificial intelligence (AI)-empowered wireless networks. Some fundamental characteristics that make RIS different from current available technologies as pointed out in [2], [3] include the unique design constraints associated with the nearly passive nature of RIS elements which cannot perform channel estimation directly, the opportunities offered by RIS for redefining the traditional notion of communication without producing new EM signals but by recycling existing radio waves, and the choice of using cost-effective material in realizing RIS to promote more sustainable wireless by design. These characteristics offer new opportunities for customizing the wireless

environment, more efficient use of radio waves, coverage extension, power transfer, positioning, and increasing the spatial capacity density [2], [3], [4] while also improving energy efficiency [5]. At the same time, these characteristics also pose new challenges in designing RIS-assisted networks/systems (e.g., communication, sensing, wireless charging, etc.), such as information transfer within the RIS-embedded environment, RIS configuration optimization with limited information, resource allocation and network optimization in such communication systems as discussed in [6], [7].

In this article, we consider the challenge of RIS configuration optimization which mainly originates from the large number of parameters to be optimized based on the contextual information. Traditional or analytics-based solutions build upon communication theories, mathematical models, and optimization algorithms. While solid and very successful, they face challenges in supporting RIS-assisted systems as discussed in [6]. Firstly, most analytics-based recent works on the optimization of RIS parameters assume relevant channels are available and develop algorithms to configure the RIS given the channel knowledge [8], [9]. However, given the nearly-passive nature of the RIS (i.e., elements on the RIS are not equipped with on-board sensing or signal processing capability while only passively reflecting the incidents according to the configuration) and the massive number of parameters to be estimated, these approaches would encounter more challenges in RIS-assisted wireless systems as discussed in [6], [7]. On top of that, the channel model of an RIS-assisted MIMO system has not yet been well understood. Secondly, such "sensed" channel status is mostly "after fact" and the system can only react to what already happened, which makes it extremely hard to combat any unfavorable or sudden change of propagation conditions, especially at high frequency bands. Lastly, from the deployment perspective, such an approach would require the communication nodes and devices to support the necessary signal processing functions and low layer protocols of the communication systems. Such requirements impose constraints on candidate application scenarios of RIS systems.

Most of the challenges are illustrations of the current wireless network design choice and assumptions that the improvements only operate on the end-points of the communication environment while regarding the wireless propagation environment between the communicating devices as uncontrollable as pointed out by [2], [6]. It leaves the job of tuning and adapting to the environment to field engineers or optimization tools. In this article, we argue that such design choice is not sufficient anymore, given the increasingly challenging requirements for future applications and new technologies/tools at our hands now.

Due to recent advances in machine learning (ML) technology, especially in deep learning (DL), applying ML-based approach in wireless communication has shown promising results for many applications as discussed in [10]. In the area of RIS-assisted wireless communication, ML-based approach has great potential given its strength in handling high-dimensional and non-convex optimization problems [11], [12]. In the supervised learning setting: [13] presented a deep neural network (DNN) model that uses the sampled channel knowledge from a few active RIS elements as input to train the proposed DNN model offline to predict the optimal RIS reflection beamforming matrix; [14] and [15] presented methods that use received pilots as input to train the proposed DNNs to predict the optimal RIS phase shifts and beamforming vector at the base station (BS) while bypassing the intermediate step of channel estimation; the authors in [16] proposed a DNN model that is trained offline to learn the implicit relationship between the measured Rx coordinates and the optimal RIS configuration. To avoid the overhead of collecting labelled data in the supervised learning setting, the authors in [17] leveraged unsupervised learning technique and designed an RIS beamforming neural network (RISBFNN) architecture to predict the optimal phase shift configuration using estimated channels at BS as input and the negated transmission rate as the loss function. Another learning alternative, deep reinforcement learning (DRL), which uses the data collected online to train the model, has gained momentum in various wireless network scenarios, especially for optimization problems [18]. For RIS-assisted wireless communication: the authors in [19] introduced an actor-critic DRL approach to study the joint design of transmit beamforming matrix at the BS and the RIS phase shift configuration for multiuser multiple input single output (MISO) system; in [20], the authors presented a potential standalone RIS solution, also based on DRL, to determine the optimal RIS beamforming vector; [21] presented a DRL-based framework to maximize the average energy efficiency by enabling a BS to determine the proper transmit power and best RIS configuration.

The above-mentioned ML approaches either take estimated channel information or observed pilots at Tx / BS to predict the optimal RIS beamforming matrix or learn the optimal RIS configuration using Rx location coordinates alone without explicitly considering the propagation environment between the Tx and Rx. In this article, we propose a new DL-based approach which aims to learn the characteristics of the channel variations in the propagation environment between the communication devices. Once learned, the DL model will capture the relationship(s) across the RIS panel and configurations, to the devices (and potentially other environmental characteristics). Such model can be used for optimal control of RIS elements and other RIS-aided applications without relying on explicit channel estimation. In addition, we believe there is a need to combine the strength of communication knowledge and ML technique, thus we design a communication theory-inspired feature map that enables the proposed DL model to learn the underlying relationships and environmental characteristics more efficiently.

Within the context described above, the main contributions of this article can be summarized as follows:

- We consider an RIS-assisted wireless communication network and propose a ML-based approach to model the local propagation environment without relying on explicit channel estimation. The trained model will reflect the important properties of the modeled environment, thus can predict the optimal RIS configuration within a specific context.
- We introduce a novel feature representation design to capture both the correlation between the real part and imaginary part of the RIS reflection beamforming vector, and the interaction between RIS elements and the Rx location attributes. This design enables the customized convolution neural network (CNN) architecture to learn the mapping function more efficiently, thus significantly reduces the overhead of collecting labelled data.
- Simulation results show that the proposed approach can achieve close to an upper bound performance while requiring less data (10x less) in the training phase compared to the benchmark method. This demonstrates that by leveraging both domain knowledge and ML techniques, a practical ML-based RIS configuration control solution can be realized.

The rest of this article is organized as follows. In Section II, we describe the network model and the channel model for an RIS-embedded environment. Section III summarizes the problem space and the mathematic formulation. Note that although our proposed approach is ML-based, we describe the channel model and problem formulation based on traditional communication theory-based mathematic formulas first in Sections II and III; later in this article, we describe how the relationships can be learned without using channel state information. In Section IV, we present our DL-based approach to learn the local propagation environment, its architecture, loss function, and algorithm pseudo-code. Section V shows numerical analysis results for the proposed approach and comparison with the baseline approach. Finally, we present conclusion and share our view for future research topics in Section VI.

Notation: Unless otherwise stated, we use the following notation throughout this article: scalars, vectors, and matrices are denoted by lower/upper case, boldface lower case, and boldface uppercase letters, respectively. For an arbitrary matrix $\mathbf{A}$, $\mathbf{A}^*$ and $\mathbf{A}^T$ are its conjugate and transpose, respectively. diag ($\mathbf{a}$) is a diagonal matrix created by putting the elements of $\mathbf{a}$ in its diagonal positions. vec($\mathbf{A}$) is a vector whose elements are the stacked columns of matrix $\mathbf{A}$. Hadamard product of matrices $\mathbf{A}$ and $\mathbf{B}$ is represented by $\mathbf{A}\odot\mathbf{B}$. $\mathcal{N}(\mu, \mathbf{R})$ is a complex Gaussian random vector with mean μ and covariance matrix $\mathbf{R}$. Imaginary unit of a complex number is denoted by $j = \sqrt{-1}$. Finally, $\mathbb{E}[\cdot]$ denotes expectation.

II. SYSTEM MODEL

In this subsection, we consider an RIS-embedded environment as the target physical world environment and as a use case we study in this article. We describe the models (network model and channel model), and assumptions of such an environment first, followed by problem formulation in Section III. These models and formulas are used to construct/simulate the RIS-environment upon which observations are drawn to train the proposed DL model.

A. NETWORK MODEL

We consider an RIS-assisted downlink communication system, where an RIS composed of N passive reflecting elements is deployed to assist in the communication from the transmitter (Tx) to a receiver (Rx). Tx and Rx are both equipped with single omni-directional antennas each. Our model can be easily extended to a multi-antenna scenario.

As done in [20], [22], we adopt an OFDM-based system of K subcarriers, and let $\boldsymbol{h}_{T,k}, \boldsymbol{h}_{R,k} \in \mathbb{C}^{N\times 1}$ be the $N \times 1$ channels from Tx/Rx to RIS at the k^{th} sub-carrier. In this article, we will focus on the case where the direct link between the Tx and Rx does not exist, i.e., the direct link is either blocked or has negligible received power compared to that received beside the RIS-assisted link and there is no other environmental input to the receiver besides the RIS. The interactions of the RIS elements on the incident signal is modeled by the diagonal matrix

$$\boldsymbol{\Theta}_k = \text{diag}(\alpha_1 \text{exp}(j\theta_1), \ldots, \alpha_{\text{N}} \text{exp}(j\theta_N)), \tag{1}$$

where $\theta_n \in [0, 2\pi)$ and $\alpha_n \in [0, 1]$ represent the phase-shift and the amplitude coefficient for element $n \in \{1, 2, \ldots, N\}$, respectively. According to [23], the amplitude coefficients are dependent on the reflection phase shifts. However, for simplicity, we assume no loss/attenuation on the incident signals, i.e., $\alpha_n = 1$ for all elements in the sequel of the paper. For a multi-carrier system like OFDM, similar time-delay in the analog domain will result into different phase shifts in different sub-carriers and the differences will be dependent on the carrier frequency, f_c. These non-uniform phase shifts over different sub-carriers will cause undesired phase errors that should be compensated by either hardware or signal processing [24]. Such per sub-carrier basis phase shift optimization is done in some recent papers [25], [26]. However, for simplicity, in this article, we assume that the same phase shift will be used for all sub-carriers and we plan to consider per sub-carrier basis phase shift optimization method for future. Hence, we will drop the subscript k from $\boldsymbol{\Theta}_k$ in the rest of this article.

Then the received signal at the receiver can be expressed as:

$$y_k = \boldsymbol{h}_{R,k}^T \boldsymbol{\Theta} \boldsymbol{h}_{T,k} x_k + n_k, \tag{2}$$

$$\overset{(a)}{=} \left(\boldsymbol{h}_{R,k} \odot \boldsymbol{h}_{T,k}\right)^T \boldsymbol{v} x_k + n_k, \tag{3}$$

where $\boldsymbol{v}$ is the RIS reflection beamforming vector, i.e., $\boldsymbol{\Theta} = \text{diag}(\boldsymbol{v})$, x_k denotes the transmitted signal over the

k^{th} subcarrier and satisfies $\mathbb{E}[|x_k|^2] = \frac{P_T}{K}$, with P_T representing the total transmit power and $n_k \sim \mathcal{N}(0, \sigma_n^2)$ is the noise power. The phase of each RIS element can be adjusted through the PIN diodes [24], which are controlled by the RIS-controller over the backhaul link. Here we assume that the backhaul link uses separate frequency resource other than the data communication frequency. We also assume that the signals reflected by RIS two or more times are ignored due to the severe "distance-product" power loss over multiple reflections [24].

B. CHANNEL MODEL

Motivated by [20], [22], here we also adopt the wideband geometric channel model [22] to model the channels $\boldsymbol{h}_{T,k}, \boldsymbol{h}_{R,k}$ between the Tx/Rx and the RIS. Let us consider a Tx-to-RIS channel, $\boldsymbol{h}_{T,k}$, (and similarly for the RIS-to-Rx channel) consisting of M clusters. Each cluster contributes with one ray from the transmitter to the RIS. The ray parameters are: azimuth/elevation angles of arrival, $\theta_m, \phi_m \in [0, 2\pi)$; complex coefficient $g_m \in \mathbb{C}$; time delay $\tau_m \in \mathbb{R}, \forall m \in \{1, 2, \ldots, M\}$. The transmitter-RIS path loss is denoted by L_T. The pulse shaping function, with T_S-spaced signaling, is defined as $p(\tau)$ at τ seconds. The delay-d channel vector $\boldsymbol{h}_{T,d}$, can then be defined as

$$\boldsymbol{h}_{T,d} = \sqrt{\frac{N}{L_T}} \sum_{m=1}^{M} g_m p(dT_s - \tau_m) \mathbf{a}(\theta_m, \phi_m), \tag{4}$$

where $\mathbf{a}(\theta_m, \phi_m) \in \mathbb{C}^{N \times 1}$ denotes the array response vector of the RIS at the angles of arrival (θ_m, ϕ_l). Given this delay-d channel, the frequency domain channel vector at subcarrier k, $\boldsymbol{h}_{T,k}$ can be expressed as

$$\boldsymbol{h}_{T,k} = \sum_{d=0}^{D-1} \boldsymbol{h}_{T,d} e^{-j\frac{2\pi k}{K}d}. \tag{5}$$

We consider a block-fading channel model, $\boldsymbol{h}_{T,k}$ and $\boldsymbol{h}_{R,k}$ are assumed to be constant over the channel coherence time, denoted T_c, which depends on the mobility of the users and the dynamics of the environment. Hence, the reflection coefficient matrix $\boldsymbol{\Theta}$ only needs to be updated after every coherence interval T_c. It is worth noting that for mmWave channels, the value of M can be very low, whereas for sub-6 GHz, signal propagation generally experiences rich scattering resulting in channels with more M.

III. PROBLEM FORMULATION

As stated previously, our main goal is to design the RIS reflection beamforming vector $\boldsymbol{v}$, to maximize the achievable rate at the receiver. Given the system and channel models in Section II, this achievable rate can be written as

$$R = \frac{1}{K} \sum_{k=1}^{K} \log_2\left(1 + \frac{P_T}{K\sigma_n^2} \left|\left(\boldsymbol{h}_{R,k} \odot \boldsymbol{h}_{T,k}\right)^T \boldsymbol{v}\right|^2\right). \tag{6}$$

In this article, we also assume that the RIS elements can only take one of the discrete quantized set of angles due to hardware constraint [23], [24]. Hence, we consider that the reflection beamforming vector $\boldsymbol{v}$ can only be picked from a pre-defined codebook $\mathcal{F}$. Each codeword in $\mathcal{F}$ is assumed to be implemented using quantized phase shifting capability of RIS elements. Hence, our main goal is to find the optimal reflection beamforming vector $\boldsymbol{v}^*$ that satisfies the following

$$\boldsymbol{v}^* = \underset{\boldsymbol{v} \in \mathcal{F}}{\text{argmax}} \sum_{k=1}^{K} \log_2\left(1 + \frac{P_T}{K\sigma_n^2} \left|\left(\boldsymbol{h}_{R,k} \odot \boldsymbol{h}_{T,k}\right)^T \boldsymbol{v}\right|^2\right). \tag{7}$$

The above solution will provide the optimal rate R^*, which can be expressed as

$$R^* = \frac{1}{K} \sum_{k=1}^{K} \log_2\left(1 + \frac{P_T}{K\sigma_n^2} \left|\left(\boldsymbol{h}_{R,k} \odot \boldsymbol{h}_{T,k}\right)^T \boldsymbol{v}^*\right|^2\right). \tag{8}$$

Due to the quantized codebook constraint, there is no closed form solution for the optimization problem in (8). Moreover, such problems are non-convex optimization problems [22], [24], [27], and since the discrete phase shifts are constrained in a finite set $\mathcal{F}$, the optimal solution can be obtained by the exhaustive search (ES) over the codebook $\mathcal{F}$. It is worth noting that the size of the codebook should normally be in the same order of N, which means that for RIS with large N, ES will not be a feasible approach.

A traditional analytical solution would follow similar framework as described above. Instead, in this article we aim to find the optimal RIS reflection beamforming vector efficiently without any explicit channel estimation. Since our problem space is discrete and hence, non-smooth, we cannot directly use gradient search based method as done in [25], [26]. As discussed before, optimal solution of problem (7) will require ES. Moreover, for finding sub-optimal solutions efficiently, authors in [27] have proposed successive refinement algorithm whose complexity is of exponential order with respect to the number of phase quantization bits [27]. On the other hand, one of the goals for our work is to build a realistic model that simulates the RIS-embedded environment's behavior without relying on channel estimation, which allow us to determine the optimal RIS reflection beamforming vector $\boldsymbol{v}^*$ from the model output, and according to the universal approximation theorem [28], DL-based methods can capture complex mappings between input-output pairs where input includes physical objects RIS and user equipment locations and output is the corresponding achievable rates in the scope of our work. These reasons motivated us to exploit DL-based technique to find the optimal RIS configurations given the users' locations as input.

We use (6)-(8) for the construction of observed achievable rates and the ground truth of optimal configurations for different receiver locations. Unlike [20], [22], we assume no presence of any active RIS element. In the next section, we introduce our novel DL-based method to learn a model that maps the environmental attributes for a given location with

any intended RIS reflection beamforming vector to the corresponding achievable rate by exploiting the location attributes of the possible receiver locations in the network.

IV. DEEP LEARNING-BASED APPROACH

In this section, we introduce a ML-based approach that aims to learn the local propagation environment between the communication devices in an RIS-assisted wireless communication network and predict the optimal RIS configuration. We leverage DL technique to build such a model that captures the interaction across the physical objects (RIS and user equipment) and the spatial patterns on the RIS to characterize the RIS-embedded environment and its intended or predicted behavior toward the user equipment, i.e., achievable rate. This approach enables the model to learn the mapping function directly without using CSI as input. First, we explain the key ideas and intuition behind the proposed method. Then, we describe our method in detail, and discuss how we construct the neural network architecture in the proposed method.

A. KEY IDEAS

RIS-assisted wireless networks consist of Tx (e.g., wireless base station), Rx (e.g., user equipment), and RIS panel. From [29], the captured power at the Rx can be represented as some function of gains of RIS elements, wavelength, and Tx/Rx locations. However, with all the elements on the RIS being passive, antenna gain information at the RIS elements is not directly available whereas we understand it depends on RIS phase-shift configuration. With the intuition derived from [29], we design our ML-based approach to directly predict the achievable rate at a given Rx location after applying the RIS reflection beamforming vector $\boldsymbol{v}$ as described in (6) by using its location attributes and the intended $\boldsymbol{v}$ as input, while bypassing channel estimation. Our approach utilizes a DL-based approach to learn the mapping function given its strength in learning complex non-linear relationships. Once the DL model learns the mapping function, the optimal RIS configuration for any new Rx location can be obtained from the model's prediction results. This means the RIS-assisted communication networks can approach optimal achievable rate as described in (8) without requiring channel estimation. Once trained, the model can be deployed as a software component at the controller which can be collocated with the RIS.

Our approach is related to a few prviously proposed methods. In [16], the authors propose a DL-base method to predict the "optimal" RIS configuration directly using only the measured coordinates at a user's location. The major difference between our approach and [16] is that our approach predicts the achievable rate for "any" Rx location and RIS reflection beamforming vector combination, which allows the system to determine proper action based on the prediction result of all or a subset of candidate RIS configurations during online inference phase even if some configurations are not seen during the training phase.

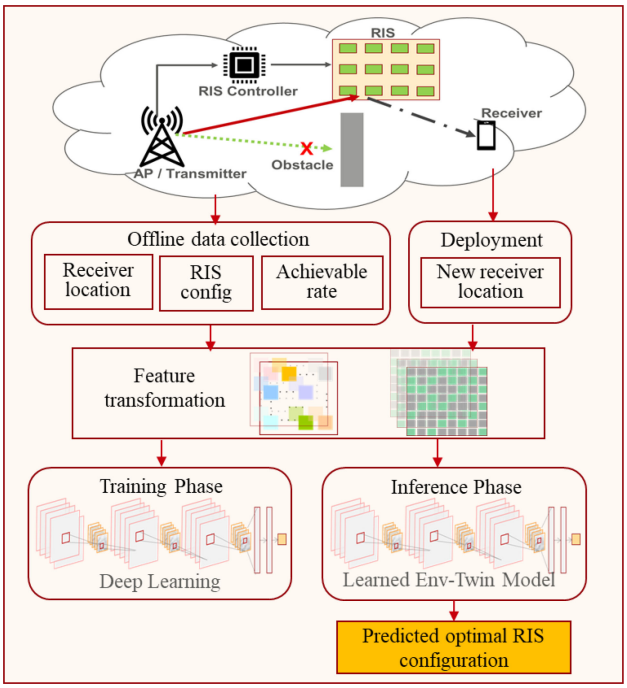

FIGURE 1. Proposed framework for the model learning and inference phases.

Another two related works as discussed in Section I: [17] presented an unsupervised learning approach for passive beamforming in RIS-assisted communication environment and takes estimated channel as input to predict the optimal RIS phase shift configuration; [15] proposed a DL-based approach that learns to configure the RIS phase shifts and the beamforming matrix at the BS to maximize the system sum rate based on the received pilot. The main difference between our approach and [15], [17] is that our approach does not use either estimated channel information or received pilot as input since one of our main goals is to remove the dependency on explicit channel information; instead, our model takes the fused feature maps that are constructed from a set of RIS phase shift configurations and Rx location attributes to predict the corresponding achievable rate at the Rx for a given RIS-assisted communication network.

B. PROPOSED METHOD

The proposed solution framework involves two phases in operation, namely the model offline training phase using training locations and the model inference phase for new Rx locations as depicted in Fig. 1.

1) Model Training Phase: During this phase, the system collects location measurement data offline from a set of sampled Rx locations, a set of RIS phase-shift configurations used in the data transmission for the sampled locations, and the corresponding achievable rates at the Rx locations. Suppose measurements for U Rx locations are collected, each can be denoted as $\boldsymbol{l}_i = (l_{i,x}, l_{i,y}, \boldsymbol{d}_i)$ with $i = 1, 2, \ldots, U$, whereas $l_{i,x}$, $l_{i,y}$ represent Rx i's x coordinate, y coordinate, and $\boldsymbol{d}_i$ represents Rx

i's distance to each of the elements on the RIS panel. Note that the size of d_i is equal to number of elements on the RIS. Part of the input for each training sample is RIS reflection beamforming vector for Rx location i. Each Rx location has N possible RIS reflection beamforming vectors denoted as $\boldsymbol{v}_{i,p}$ with $p = 1, 2, \ldots, N$, where N is the total number of RIS reflection beamforming vectors specified in the predefine codebook $\mathcal{F}$. The default size of N is $N_1 \times N_2$, where (N_1, N_2) is the dimension of RIS, and for convenience of discussion we can set $N_1 = N_2$. Thus, the training input set for location i can be denoted as

$$S_i \triangleq \left\{(\boldsymbol{l}_i, \boldsymbol{v}_{i,1}), (\boldsymbol{l}_i, \boldsymbol{v}_{i,2}), \ldots, (\boldsymbol{l}_i, \boldsymbol{v}_{i,N})\right\}.$$

The corresponding labelled output, represented by $R_{i,p}$ (bps/Hz) is the observed achievable rate at Rx location i after RIS reflects using reflection beamforming vector $\boldsymbol{v}_{i,p}$, thus, there are a total of N achievable rate prediction results for Rx location i, one for each input $\boldsymbol{v}_{i,p}$ as defined in the codebook $\mathcal{F}$. Once the samples are acquired, the DL model is trained using these collected training samples. The training process allows the model to learn how to map an input sample (RIS reflection beamforming vector, $\boldsymbol{v}_{i,p}$, and location attributes $\boldsymbol{l}_i$ for Rx i) to its corresponding output (achievable rate, $R_{i,p}$). The offline data collection and model training procedures can be performed on the RIS controller if it has sufficient capacity, or at a separate computing equipment that has access to the data. The trained DL model can then be uploaded to the RIS controller for the next model inference step.

2) Model Inference Phase: During the model inference phase, the system first receives estimated location attributes from the intended Rx j, namely its x coordinate, y coordinate, and calculates its distance features $\boldsymbol{d}_j$ to each of the RIS elements to form input $\boldsymbol{l}_j = (l_{j,x}, l_{j,y}, \boldsymbol{d}_j)$. Then, the system constructs input samples for location j using all possible RIS reflection beamforming vectors specified in the predefined codebook $\mathcal{F}$. The new input sample set for the intended Rx location j can be denoted as

$$S_j \triangleq \left\{(\boldsymbol{l}_j, \boldsymbol{v}_{j,1}), (\boldsymbol{l}_j, \boldsymbol{v}_{j,2}), \ldots, (\boldsymbol{l}_j, \boldsymbol{v}_{j,N})\right\}.$$

After the new sample set is constructed, the trained DL model takes the input and predicts the achievable rates at location j for each possible RIS reflection beamforming vectors, $\boldsymbol{v}_{j,p}$, which can be expressed as

$$\widehat{\boldsymbol{r}_j} \triangleq \left\{\hat{R}_{j,1}, \hat{R}_{j,2}, \ldots, \hat{R}_{j,N}\right\}.$$

The system performs an exhausted search across the achievable rates from the DL prediction output to determine the optimal RIS reflection beamforming vector for location j. This predicted optimal reflection beamforming vector $\widehat{\boldsymbol{v}}_j$ is then used in data transmission for location j. Note that the inference procedure can be performed on the RIS controller, which is also responsible for acquiring the needed input data, i.e., available RIS beamforming vectors and Rx location attributes, from the communication infrastructure and determining the optimal RIS configuration based on the prediction outcome from the model. The inference procedure will introduce additional delay, mainly due to data acquisition, which needs to be considered when designing the end-to-end solution.

Algorithm 1 Deep Learning based Env-Twin Model

Step 1: Model Training
// Training Input: Q (number of samples), receiver location features $\boldsymbol{\ell}_i \triangleq (x_i, y_i, \boldsymbol{d}_i)$,
// $\boldsymbol{i} = 1, \ldots, U$, RIS reflection beamfoiming vector $\boldsymbol{v}_{l\ldots\mathbf{p}}$, N_1 N_2 (RIS dimension)
// Training Output: achievable rate $\hat{R}$ at receiver location $\boldsymbol{i}$ after RIS reflects using $\boldsymbol{v}_{\mathbf{p}}$
Input_train ← reshape(Input, (Q, N_1, N_2, dim(location attributes) + 2))
Output_train reshape(Output, (Q, 1))
D ← (Input_train, Output_train)
Build CNN model M for learning
Initialize weights, θ for M
for ep = θ to epoch-1 **do**:
  Minibatch training dataset D
  forward propagation to calculate $\hat{R}$
  gradient descent on MSE(R, $\hat{R}$) to obtain new θ
  update model weights using new θ
end
Step 2: Model Inference
// Input: trained CNN model M, location features $\boldsymbol{\ell}_j$ for intended receiver location j,
// predefined RIS reflection beamfoiming vectors $\boldsymbol{v}_{l\ldots\mathbf{p}}$
// Output: predicted best reflection beamfoiming vector for intended receiver location j
Input ← $((\boldsymbol{\ell}_j, \boldsymbol{v}_1), (\boldsymbol{\ell}_j, \boldsymbol{v}_2), \ldots, (\boldsymbol{\ell}_j, \boldsymbol{v}_{\mathbf{p}})$
Input ← reshape(Input, (p, N_1, N_2), dim(location attributes)+2))
$\hat{\boldsymbol{r}} = M.predict$(Input)
opt_index ← argmax($\hat{\boldsymbol{r}}$)
RIS uses $\boldsymbol{v}_{\text{opt\_index}}$ as the reflection beamfoiming vector for the intended receiver location j

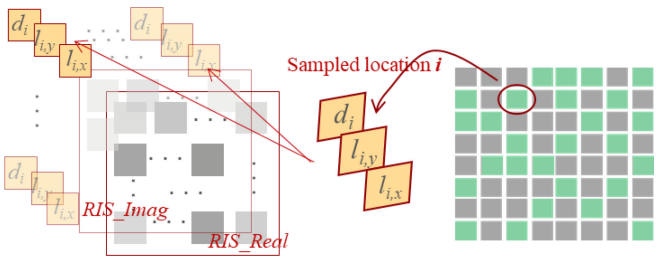

FIGURE 2. This figure shows the representation of input features to the proposed CNN model.

The Proposed DL-Based Operation is Described in Algorithm 1.

C. DEEP NEURAL NETWORK ARCHITECTURE AND PARAMETERS

1) Feature Representation: Each input sample to the DL model comprises of RIS configuration, and Rx's location attributes. Inspired by [29], we construct input features for each sample to the DL model, $(\boldsymbol{l}_i, \boldsymbol{v}_{i,p})$, as separate 2D channels. The first channel is the real part of $\boldsymbol{v}_{i,p}$, the second channel is the imaginary part of $\boldsymbol{v}_{i,p}$, and channels 3 – 5 are the location attributes. This representation is depicted in Fig. 2.
2) Neural Network Architecture: Our proposed DNN architecture for RIS-assisted wireless networks is illustrated in Fig. 3. We choose to leverage CNN-based NN

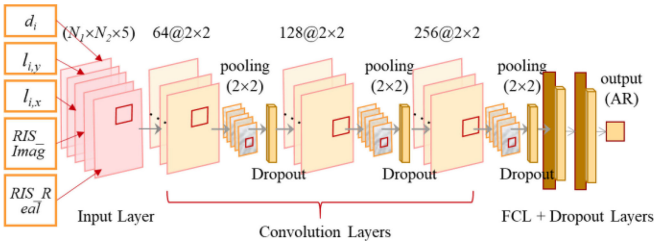

FIGURE 3. The CNN Architecture for the proposed NN model.

architecture because a) we believe there are spatial patterns on the RIS, thus CNN-based approach is a better fit in learning and extracting the spatial relationships / features vs. multilayer perceptron-based NN architecture, and b) the interactions across input features will jointly impact the achievable rate at the Rx location, thus it is natural to represent them as separate channels and use a CNN model to learn their interactions more explicitly. In our proposed CNN-based approach, the input layer is a 5-channel map as described in the Feature Representation subsection, and the dimension of each channel is $N_1 \times N_2$, where we assume $N_1 = N_2$. The first channel comprises of the real part of the input RIS reflection beamforming vector, the second channel comprises of the imaginary part of the RIS reflection beamforming vector, the third channel comprises of the Rx location's x coordinate, the fourth channel is the Rx location's y coordinate, and the fifth and last channel comprises of the distance between the Rx location and each RIS element. Layers 2, 5, and 8 are convolutional layers (CLs), and each is followed by a pooling layer. The last pooling layer is followed by two fully connected layers (FCLs), and there is a dropout layer between each pooling layer and the next CL and after each FCL. The last layer is a regression layer which predicts the scalar achievable rate for the input Rx location. We train the model using Adam optimizer [30] with default beta_1 and beta_2 and use 0.001 for the learning rate. During training, we use batch size to 64 and set epochs to 100, and apply early stopping with patience set to 20, which ceases training procedure when the validation loss does not improve in 20 consecutive epochs. Note that we tried a few learning rate variations and select the final learning rate based on the performance of the validation samples. We also tried a few different batch sizes but we notice that changing batch size doesn't impact performance much. As noisy estimations of location attributes are ubiquitous in real-world dataset, it poses a challenge for training a DNN that can generalize well when encounters unseen noisy input data. There are various regularization mechanisms to improve the robustness of DNN as discussed in [31], [32]. In this study, we add a dropout layer and Gaussian noise after each pooling layer and after each FCL in the proposed DNN architecture to reduce model overfitting on the training dataset. A more thorough study may be conducted when real-world sample data is collected to determine what regularization techniques work best for our scenario.

3) Loss Function: The objective of the model is to accurately learn the mapping function and the output is the achievable rate for a given Rx location i after applying RIS reflection beamforming vector $\boldsymbol{v}_{i,p}$. We train the model using a regression loss function of mean-squared-error (MSE), denoted as MSE ($\hat{R}$, R) for each minibatch, in each epoch in the model training phase to minimize the MSE between the achievable rate predicted by the convolution neural network (CNN) model and the ground-truth label of achievable rate, which is obtained through simulation.

V. NUMERICAL RESULTS

A. EXPERIMENTAL SETUP

In this section, we evaluate our performance by leveraging the public dataset, DeepMIMO [33]. This dataset includes scripts and tools that enables generating different datasets to simulate different scenarios by using specified parameters and settings, which facilitates ML research and development for mmWave/massive MIMO. Some research works [13], [20] have already been published using this open dataset, thanks to its open-source and ease of use. To verify our approach, we also leverage the DeepMIMO ray-tracing scenario to generate sample data for training and testing our proposed model. Considering the use case of RIS and the goal of using RIS to reflect the beamforming, we select BS 3 in the DeepMIMO 'O1' scenario to be the RIS surface and treat the antenna elements on BS 3 as RIS elements. To best simulate the impact of the environmental geometry on the realistic channels, ray-tracing is adopted to capture the dependence on the key environmental factors such as the environment geometry and materials the RIS and Tx/Rx locations, the operating frequency, etc. More specifically in this project, we chose the scenario O1_28 as the simulated real physical environment. The layout of scenario O1_28 is shown in Fig. 4.

In the scenario O1_28, there is one Tx as labelled by a red dot in Fig. 4. The Rx's location area is shown by a gray rectangle starting from Row 1,000 to Row 1,300. For each row in the receiving area, there are 181 Rx locations. In another way, there are a total of $300 \times 181 = 54,300$ possible Rx locations in total in this scenario. Between the Tx and Rx's, BS3 is selected as the RIS for reflecting the beamforming, which is denoted by a red line.

The configuration of the RIS and the communication parameters in O1_28 are shown in Table 1. The RIS employs a uniform planar array (UPA) with 16×16 (256) antennas at the 28GHz setup. The Tx and Rx are assumed to have a single antenna each. Based on the above configuration parameters, the DeepMIMO simulation tool generates the RIS reflection beamforming codebook. In

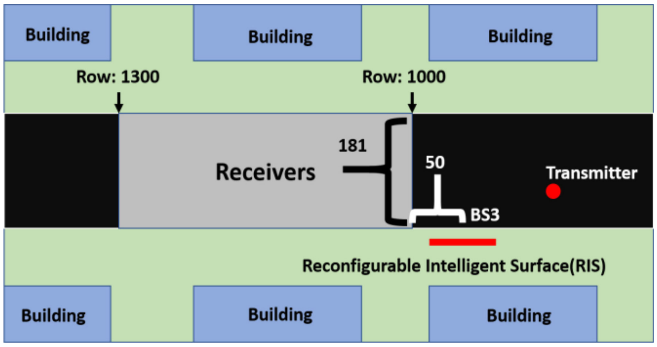

FIGURE 4. This figure illustrates the DeepMIMO simulation scenario 'O1' used to generate the dataset for performance study of the proposed approach. The Tx is fixed at R950 and column 90. Candidate receiver is located between R1000 and R1300.

TABLE 1. The adopted DeepMIMO dataset configuration parameters.

PARAMETER	Value
Number of RIS Antennas	(*Nx;Ny;Nz*) = (1; 16; 16)
Antenna spacing	0.5
System bandwidth	100 MHz
Number of OFDM subcarriers	512
OFDM sampling factor	1
OFDM limit	16
Number of paths	5
Transmit Power P_T	5 dB

this scenario, it adopts a discrete Fourier transform (DFT) codebook [34], [35] for the candidate RIS reflection beamforming vectors, which are also part of the input to the neural network of our method. Next, the DeepMIMO dataset used the ray-tracing simulator, Remcom Wireless InSite [36], to calculate the achievable bit rate for each location of the Rx based on the obtained DFT codebook and parameters for scenario O1_28. Given the above RIS configurations and parameters, there are a total of 256 reflection beamforming vectors in the codebook by default. The DeepMIMO simulator calculates the achievable bit rate for each RIS reflection beamforming vectors for each location according to (6), which is treated as the ground truth of our neural network training process. Note that the DeepMIMO dataset doesn't contain separate location coordinates for each element on the RIS / BS3, thus, we use BS3's coordinates as feature input in our model.

B. EVALUATION METRICS

This subsection discusses the metrics we use for evaluating the performance of our work. Previous research [13] using the DeepMIMO dataset has demonstrated promising results using ML-based approach, thus we choose this approach as our baseline method. For comparable comparison, we use the same parameters to run the baseline method [13] and our proposed approach. We also use the same set of training locations and testing locations in all our comparisons. Note that [13] requires some active elements on the RIS panel and we set that to 8 ($\overline{M} = 8$) in our experiment to generate the baseline result. In [13], sampled channel vectors are used as input to predict achievable rates for all the RIS reflection beamforming vectors specified in the codebook at the same time. Our approach uses the Rx's location attributes with any intended RIS reflection beamforming vector as defined in the codebook $\mathcal{F}$ as input to predict the corresponding achievable rate. For performance evaluation, we use the following metrics:

- Top 1 prediction accuracy: This metric calculates the percentage of testing locations with correct prediction for the optimal RIS reflection beamforming vector, i.e., the predicted matches the ground truth according to the DeepMIMO dataset. The top 1 prediction accuracy, denoted as $Top1_{acc}$ can be expressed as

$$Top1_{acc} = \frac{1}{U}\sum_{i=1}^{U}\left(\widehat{Opt}(i) == Top_1{}^*(i)\right), \tag{9}$$

  where U is total number of testing Rx locations, $Top_1{}^*(i)$ is the true optimal RIS reflection beamforming vector for location i according to the DeepMIMO dataset, and $\widehat{Opt(i)}$ is the best RIS reflection beamforming vector for location i according to the prediction results from the model.
- Top 3 prediction accuracy: This metric measures the percentage of testing locations whose predicted optimal RIS reflection beamforming vectors from the model are among the top 3 RIS reflection beamforming vectors according to the DeepMIMO dataset. Note that RIS reflection beamforming vectors for each testing location are ranked in descending order according to the resulted achievable rates based on the DeepMIMO ray-tracing output which assumes perfect channel knowledge. The top 3 prediction accuracy, denoted as $Top3_{acc}$ can be expressed as

$$Top3_{acc} = \frac{1}{U}\sum_{i=1}^{U}\left(\widehat{Opt}(i) \in \left\{Top_1{}^*(i), Top_2{}^*(i), Top_3{}^*(i)\right\}\right), \tag{10}$$

  where U is total number of testing Rx locations, $Top_1{}^*(i)$ is the true optimal RIS reflection beamforming vector for location i, and $Top_2{}^*(i)$ and $Top_3{}^*(i)$ are the true second and third best RIS reflection beamforming vectors for location i. $Top_1{}^*(i)$, $Top_2{}^*(i)$, and $Top_3{}^*(i)$ are obtained from the DeepMIMO dataset.
- Recovered achievable rate percentage: For each testing location, we calculate the achievable rate reached when RIS applied the best reflection beamforming vector as predicted by the model. This predicted achievable rate is divided by the optimal achievable rate for the location based on the DeepMIMO dataset. We then average the results across all the testing locations to get the average recovered achievable rate percentage. The average recovered achievable rate percentage, denoted as

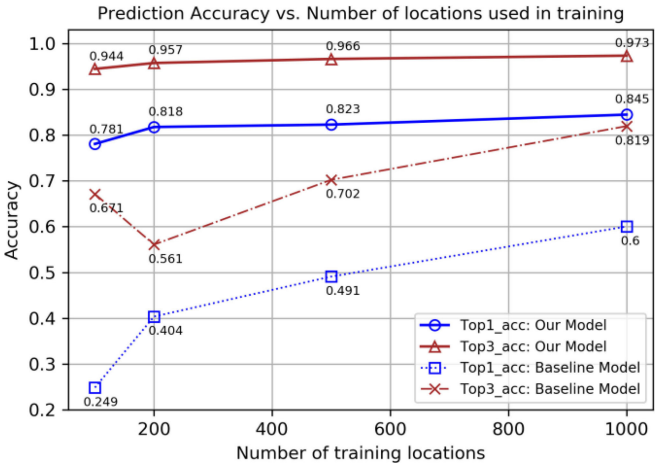

FIGURE 5. This chart illustrates top 1 and top 3 prediction accuracy comparison between our approach and the baseline approach [13]. Results from each model are averages from 10 runs.

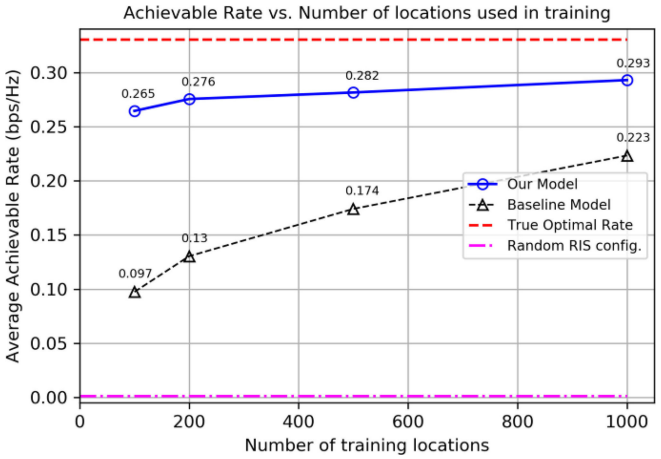

FIGURE 6. The average achievable rate of our approach is compared with the baseline approach proposed in [13] and random RIS configuration. The upper bound is generated from the DeepMIMO dataset, which assumes perfect channel knowledge. Results for each approach are the averages from 10 runs.

Recov_AR_avg, can be expressed as

$$Recov\_AR\_avg = \frac{1}{U}\sum_{i=1}^{U}\frac{\widehat{AR}(i)}{AR^*(i)}, \tag{11}$$

where U is total number of testing Rx locations, $\widehat{AR}(i)$ is the achievable rate after applying the best RIS reflection beamforming vector for location i based on the prediction result of the model, and $AR^*(i)$ is the achievable rate for location i after applying the true optimal RIS reflection beamforming vector based on the DeepMIMO ray-tracing simulation scenario.

C. SIMULATION RESULTS

In this subsection, we discuss the performance evaluation results. We train our model based on the generated DeepMIMO dataset for scenario O1_28 and evaluate the results using the metrics of average achievable rate and the Top P accuracy, where $P \in \{1,\ 3\}$ as defined in the Evaluation Metrics subsection.

Fig. 5 illustrates the optimal RIS reflection beamforming vector prediction performance of the proposed model in terms of $Top1_{acc}$ and $Top3_{acc}$ as defined in (9) and (10), respectively. As shown, prediction accuracy improves when the number of training locations increases, and our proposed model achieved decent performance when using samples from only 100 Rx locations. We compare the performance with the approach proposed in [13], which also achieved decent $Top3_{acc}$ when using 1,000 training locations, whereas the proposed model showed significantly better performance in $Top1_{acc}$ when fewer training locations were used in training the model. Note that the result of each approach is the average of ten independent runs of the corresponding method. We notice performance degradation in the baseline method when number of training locations is increased from 100 to 200. This could be due to randomly initialized neural network weights.

TABLE 2. Recovered achievable rate for 1,000 testing locations (results are the averages from 10 runs).

Training Rx locations	**100%** RIS samples		**30%** RIS samples		True avg. optimal AR (bps/Hz)
	Avg. AR (Pred.)	*Recov_AR_avg*	Avg. AR (Pred.)	*Recov_AR_avg*	
100	0.265	0.861	0.264	0.861	0.330
200	0.276	0.889	0.270	0.879	
500	0.282	0.895	0.278	0.887	
1000	0.293	0.913	0.290	0.909	

To understand the achievable rate recovered after applying the best RIS reflection beamforming vector predicted by our model, we study model performance by using different numbers of Rx locations in the training phase. We calculate *Recov_AR_avg* as defined in (11) and the results are described in the second and third columns of Table 2. We compare the recovered achievable rate between our model, the baseline approach [13] and random RIS configuration, and the results are illustrated in Fig. 6. As shown in the second and third columns of Table 2 and Fig. 6, *Recov_AR_avg* increases when the number of training locations increases. Our model can achieve ~0.89 in *Recov_AR_avg* when using only 200 locations (less than 0.5% of the total 54,300 Rx locations) in training, which is significantly higher than both the baseline approach and random RIS configurations. With 1,000 training Rx locations, our model achieved 0.91 in Recov_*AR_avg*.

To evaluate the robustness of the proposed CNN-based approach, we perform two analyses. The first analysis is to evaluate whether our proposed approach can generalize to unseen Rx locations. We first train our model using 1,000 Rx locations (less than 2% of 54,300 locations), then we use the trained model to predict the achievable rates for various numbers of testing Rx locations and the results are illustrated in Table 3. Note that due to memory constraint on the

TABLE 3. Achievable rates reached using predicted RIS configuration vs. true optimal achievable rate from the DeepMIMO dataset (1,000 locations were used in training the model).

Number of Testing Locations	Average Achievable Rate (using **Predicted** best config.)	Average Optimal Achievable Rate (using **True** optimal config.)	*Recov_AR_avg*
200	0.26415	0.31084	0.889
500	0.27722	0.31064	0.905
1000	0.27443	0.30865	0.896
6000	0.27612	0.31429	0.893
11000	0.28114	0.31662	0.901
16000	0.28375	0.32122	0.898
21000	0.28351	0.32106	0.899
26000	0.28029	0.31809	0.896
31000	0.28120	0.31881	0.897

machine, we ran predictions up to 31,000 testing receiver locations. As shown, the *Recov_AR_avg* stays stale between 0.89 and 0.91 across all the testing Rx location ranges. The second analysis is to evaluate whether the proposed approach can generalize well for unseen RIS reflection beamforming vectors. In this analysis, we first randomly sampled 30% of the available RIS reflection beamforming vectors from the DeepMIMO dataset for each training Rx location, then used only these 30% of RIS samples in the model training phase. Note that the random sampling was performed for each location independently to resemble the real-world deployment scenario. The results are described in the fourth and fifth columns of Table 2. As shown, the prediction performance for the optimal RIS configuration degraded slightly when using less RIS samples in the training phase comparing with the original model that used 100% of the RIS reflection beamforming vectors. However, we regard the performance is still decent. These two analyses suggest that our CNN model is very sampling efficient, which enables practical realization of RIS control as it requires fewer number of training locations as well as fewer number of RIS reflection beamforming vector samples.

VI. CONCLUSION

In this article, we introduced a ML-based approach to predict the optimal RIS phase shifts by building a DL model that aims to capture the interactions and characteristics between the surrounding environment and the communication network. Our model learns the mapping between the RIS-embedded environment and the achievable rates at Rx locations and recommends the optimal RIS phase shifts without relying on any explicit channel estimation effort. Simulation results showed that our DL model can converge to near-optimal data rates using less than 2% of the total number of Rx locations, and it can generalize well to unseen RIS reflection beamforming vectors. Even though RIS-assisted wireless communications is still in its initial research stage, we believe this technology has great potential which makes the research work of developing techniques to optimize the RIS configuration important and rewarding. For future research, we plan to extend our research for scenarios like multi-user, multiple BS and multiple antennas at the Tx and Rx and to study the practical application challenges of the proposed method using data collected from the real-world including non-RF sensing type of data to capture additional environmental characteristics. Meanwhile, how to combine ML-based approach with traditional analytical method is a potential research topic to leverage the strengths of both.

REFERENCES

[1] M. D. Renzo *et al.*, "Smart radio environments empowered by reconfigurable AI meta-surfaces: An idea whose time has come," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, p. 129, May 2019, doi: 10.1186/s13638-019-1438-9.

[2] M. D. Renzo *et al.*, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, Nov. 2020, doi: 10.1109/JSAC.2020.3007211.

[3] C. Huang *et al.*, "Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 118–125, Oct. 2020, doi: 10.1109/MWC.001.1900534.

[4] D. Dardari, "Communicating with large intelligent surfaces: Fundamental limits and models," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2526–2537, Nov. 2020, doi: 10.1109/JSAC.2020.3007036.

[5] 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, Aug. 2019, doi: 10.1109/TWC.2019.2922609.

[6] M. A. ElMossallamy, H. Zhang, L. Song, K. G. Seddik, Z. Han, and G. Y. Li, "Reconfigurable intelligent surfaces for wireless communications: Principles, challenges, and opportunities," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 990–1002, Sep. 2020, doi: 10.1109/TCCN.2020.2992604.

[7] X. Yuan, Y.-J. A. Zhang, Y. Shi, W. Yan, and H. Liu, "Reconfigurable-intelligent-surface empowered wireless communications: Challenges and opportunities," Aug. 2020. Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/2001.00364.

[8] Q. Wu and R. Zhang, "Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1838–1851, Mar. 2020, doi: 10.1109/TCOMM.2019.2958916.

[9] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5394–5409, Nov. 2019, doi: 10.1109/TWC.2019.2936025.

[10] A. Zappone, M. D. Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019, doi: 10.1109/TCOMM.2019.2924010.

[11] T. Lin and Y. Zhu, "Beamforming design for large-scale antenna arrays using deep learning," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 103–107, Jan. 2020, doi: 10.1109/LWC.2019.2943466.

[12] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–6, doi: 10.1109/SPAWC.2017.8227766.

[13] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Deep learning for large intelligent surfaces in millimeter wave and massive MIMO systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6, doi: 10.1109/GLOBECOM38437.2019.9013256.

[14] Ö. Özdogan and E. Björnson, "Deep learning-based phase reconfiguration for intelligent reflecting surfaces," presented at the Asilomar Conf. Signals Syst. Comput., Sep. 2020. Accessed: Oct. 22, 2020. [Online]. Available: http://arxiv.org/abs/2009.13988

[15] T. Jiang, H. V. Cheng, and W. Yu, "Learning to beamform for intelligent reflecting surface with implicit channel estimate," presented at the IEEE Global Commun. Conf. (GLOBECOM), Dec. 2020. Accessed: Dec. 10, 2020. [Online]. Available: http://arxiv.org/abs/2009.14404

[16] C. Huang, G. C. Alexandropoulos, C. Yuen, and M. Debbah, "Indoor signal focusing with deep learning designed reconfigurable intelligent surfaces," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5, doi: 10.1109/SPAWC.2019.8815412.

[17] J. Gao, C. Zhong, X. Chen, H. Lin, and Z. Zhang, "Unsupervised learning for passive beamforming," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1052–1056, May 2020, doi: 10.1109/LCOMM.2020.2965532.

[18] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019, doi: 10.1109/COMST.2019.2926625.

[19] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1839–1850, Aug. 2020, doi: 10.1109/JSAC.2020.3000835.

[20] A. Taha, Y. Zhang, F. B. Mismar, and A. Alkhateeb, "Deep reinforcement learning for intelligent reflecting surfaces: Towards standalone operation," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5, doi: 10.1109/SPAWC48557.2020.9154301.

[21] G. Lee, M. Jung, A. T. Z. Kasgari, W. Saad, and M. Bennis, "Deep reinforcement learning for energy-efficient networking with reconfigurable intelligent surfaces," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149380.

[22] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling large intelligent surfaces with compressive sensing and deep learning," Apr. 2019. Accessed: Aug. 14, 2020. [Online]. Available: http://arxiv.org/abs/1904.10136.

[23] S. Abeywickrama, R. Zhang, and C. Yuen, "Intelligent reflecting surface: Practical phase shift model and beamforming optimization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148961.

[24] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface aided wireless communications: A tutorial," Jul. 2020. Accessed: Aug. 14, 2020. [Online]. Available: http://arxiv.org/abs/2007.02759.

[25] 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, Jul. 2020, doi: 10.1109/TCOMM.2020.2981458.

[26] B. Zheng and R. Zhang, "Intelligent reflecting surface-enhanced OFDM: Channel estimation and reflection optimization," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 518–522, Apr. 2020, doi: 10.1109/LWC.2019.2961357.

[27] C. You, B. Zheng, and R. Zhang, "Intelligent reflecting surface with discrete phase shifts: Channel estimation and passive beamforming," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149292.

[28] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Trans. Neural Netw.*, vol. 6, no. 4, pp. 911–917, Jul. 1995, doi: 10.1109/72.392253.

[29] E. Basar and I. Yildirim, "SimRIS channel simulator for reconfigurable intelligent surface-empowered communication systems," May 2020. Accessed: Aug. 12, 2020. [Online]. Available: http://arxiv.org/abs/2006.00468.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2017, Accessed: Aug. 19, 2020. [Online]. Available: http://arxiv.org/abs/1412.6980.

[31] D. Arpit *et al.*, "A closer look at memorization in deep networks," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 233–242. Accessed: Oct. 13, 2020. [Online]. Available: http://proceedings.mlr.press/v70/arpit17a.html

[32] L. Jiang, D. Huang, M. Liu, and W. Yang, "Beyond synthetic noise: Deep learning on controlled noisy labels," in *Proc. Int. Conf. Mach. Learn.*, vol. 1, 2020, pp. 4804–4815. Accessed: Oct. 13, 2020. [Online]. Available: https://proceedings.icml.cc/paper/2020/hash/40cb228987243c91b2dd0b7c9c4a0856

[33] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. Inf. Theory Appl. Workshop (ITA)*, San Diego, CA, USA, Feb. 2019, pp. 1–8.

[34] Freescale Semiconductor, Inc., *Results on Zero-Forcing MU-MIMO*, document TSG RAN WG1, R1-071511, 3GPP, Gothenburg, Sweden, 2007. [Online]. Available: http://www.3gpp.com

[35] 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, doi: 10.1109/TCOMM.2003.814195.

[36] REMCOM. *Wireless EM Propagation Software—Wireless InSite*. Accessed: Aug. 14, 2020. [Online]. Available: https://www.remcom.com/wireless-insite-em-propagation-software

BAOLING SHEEN received the M.S. and Ph.D. degrees in computer science from the Illinois Institute of Technologies, Chicago, IL, USA, in 1991 and 1996, respectively.

From 1997 to 2006, she was a Member of Technical Staff with Lucent Technologies. From 2006 to 2011, she was a System Engineer with Alcatel-Lucent Technologies, working on performance management for 3G and 4G LTE systems. She joined Futurewei Technologies, Chicago, IL, USA, in 2011, and she is currently a Principal Engineer with Applied Wireless ML/AI Research Group. Her research interests include V2X, network optimization, predictive analytics for network maintenance, traffic forecast, and machine learning.

JIN YANG received the B.S. and M.S. degrees in computer science and engineering from Tsinghua University, Beijing, China, in 1987 and 1989, respectively, and the Ph.D. degree in computer science from Imperial College, London, U.K., in 1997.

From 1997 to 1998, he was a Research Fellow with University College London. Since 1998, he carried out various technology research and leadership roles in the telecommunication industry, working for Lucent Technologies and Motorola, in U.K. and China. He joined Futurewei Technologies, Bridgewater, NJ, USA, in 2011, and is currently the Technical VP leading a research group exploring and applying machine learning and AI technologies in wireless network operation, optimization and design space, toward an AI enabled 5G and future networks. He has authored more than 20 articles and inventor of more than 30 patents. His other research interests include V2X, collaborative driving, and Intelligent Infrastructure.

XIANGLONG FENG received the B.S. degree in communication engineering from China University of Petroleum, China, in 2012, and the M.S. degree in computer application technology from the Chinese Academy of Sciences, China, in 2015. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Rutgers University, New Brunswick, NJ, USA.

His works have been published in ACM MM, Ubicomp, IEEE VR, and ICCAD. He has a wide research interest including hardware security in heterogeneous platform, wireless communication, and video streaming optimization, the AR/VR application, and using artificial intelligence to solve related problems.

MD MOIN UDDIN CHOWDHURY received the B.S. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology in 2014. He is currently pursuing the Ph.D. degree in electrical and computer engineering with North Carolina State University.