IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 26, 2021, accepted February 26, 2021, date of publication March 4, 2021, date of current version March 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064073

Enabling Large Intelligent Surfaces With Compressive Sensing and Deep Learning

ABDELRAHMAN TAHA, MUHAMMAD ALRABEIAH, AND AHMED ALKHATEEB
School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA

Corresponding author: Ahmed Alkhateeb (aalkhateeb@asu.edu)

ABSTRACT Employing large intelligent surfaces (LISs) is a promising solution for improving the coverage and rate of future wireless systems. These surfaces comprise massive numbers of nearly-passive elements that interact with the incident signals, for example by reflecting them, in a smart way that improves the wireless system performance. Prior work focused on the design of the LIS reflection matrices assuming full channel knowledge. Estimating these channels at the LIS, however, is a key challenging problem. With the massive number of LIS elements, channel estimation or reflection beam training will be associated with (i) huge training overhead if all the LIS elements are passive (not connected to a baseband) or with (ii) prohibitive hardware complexity and power consumption if all the elements are connected to the baseband through a fully-digital or hybrid analog/digital architecture. This paper proposes efficient solutions for these problems by leveraging tools from compressive sensing and deep learning. First, a novel LIS architecture based on *sparse channel sensors* is proposed. In this architecture, all the LIS elements are passive except for a few elements that are active (connected to the baseband). We then develop two solutions that design the LIS reflection matrices with negligible training overhead. In the first approach, we leverage compressive sensing tools to construct the channels at all the LIS elements from the channels seen only at the active elements. In the second approach, we develop a deep-learning based solution where the LIS learns how to interact with the incident signal given the channels at the active elements, which represent the state of the environment and transmitter/receiver locations. We show that the achievable rates of the proposed solutions approach the upper bound, which assumes perfect channel knowledge, with negligible training overhead and with only a few active elements, making them promising for future LIS systems.

INDEX TERMS Large intelligent surface, intelligent reflecting surfaces, reconfigurable intelligent surface, smart reflect-array, beamforming, millimeter wave, compressive sensing, deep learning.

I. INTRODUCTION

Large Intelligent Surfaces (LISs) have been envisioned as integral constituents of beyond-5G wireless systems [2]–[17]. From a conceptual design perspective, by stacking a huge number of sensing or radiating elements, the LIS ideally aims to effectuate a continuous electromagnetically active surface. These LIS elements are expected to interact in a smart way with the incident signals in order to enhance the spectral efficiency and coverage of wireless systems [2], [3]. What adds to the appeal of such surfaces is that their function could be performed with energy-efficient implementations, e.g., using nearly-passive elements such as analog phase shifters [4]–[6]. Prior work focused on designing the LIS interaction matrices and evaluating their spectral efficiencies and coverage gains *while assuming the availability of global channel knowledge*. **But how can these extremely large-dimensional channels be estimated if the LIS is implemented using only reflecting elements?** Obtaining this channel knowledge may require huge—and possibly prohibitive—training overhead, which represents the main challenge for the LIS system operation. To overcome that, this paper proposes a novel LIS hardware architecture along with two solutions based on compressive sensing and deep learning. These solutions utilize the novel architecture of the surface and design the interaction matrix with very negligible training overhead.

A. PRIOR WORK

Under various names such as reconfigurable intelligent surfaces, intelligent reflecting surfaces, and smart reflect-arrays,

The associate editor coordinating the review of this manuscript and approving it for publication was Nafees Mansoor.

LIS-assisted wireless communications have been drawing increasing interest in recent years. From an implementation perspective, LIS can be built using nearly-passive elements with reconfigurable parameters [5]. Various LIS designs have been proposed in the literature with more prominence given to software-defined metamaterials [7], [8] and conventional reflect-arrays [4], [6] among others. For all those designs, different signal processing solutions have been proposed for optimizing the design of the LIS interaction matrices. An LIS-assisted downlink multiuser setup was considered in [5] with single-antenna users. computational low-complexity algorithms were then proposed for optimizing the design of the LIS interaction matrices, using quantized phase shifters/reflectors for modeling the LIS elements. In [9], an LIS-assisted downlink scenario was considered, where both the LIS interaction matrix and the base station precoder matrix were designed, assuming the case where a line-of-sight (LOS) may exist between the base station and the LIS. In [10], a new transmission strategy combining LIS with index modulation was proposed to improve the system spectral efficiency.

In terms of the overall system performance, an uplink multiuser scenario was considered in [11] and the data rates were formulated for the case where channel estimation errors exist in the available channel knowledge. A downlink LIS-assisted multiple-input multiple-output (MIMO) non-orthogonal multiple access (NOMA) framework is proposed in [16] for achieving higher system spectrum efficiency gains. The LIS can be leveraged for wireless localization purposes as well; in [17], an LIS-assisted downlink millimeter wave (mmWave) positioning problem was analyzed from the Fisher Information perspective. Based on this analysis, an algorithm was developed for improving the positioning quality.

Deep learning solutions have been proposed in the literature for addressing design challenges in mmWave and massive MIMO systems [18]–[20]. In [18], a deep learning based beam prediction solution was proposed for distributed mmWave MIMO systems to serve highly mobile users with negligible training overhead and high data rate gains, compared to coordinated beamforming strategies that do not leverage machine learning. In [19], a deep learning based blockage prediction solution was proposed to address the reliability and latency challenges of sudden blockage of the line-of-sight link in mmWave MIMO systems. A channel covariance prediction solution using generative adversarial networks was proposed in [20] for mmWave Massive MIMO systems to reduce the training overhead associated with acquiring the channel knowledge.

The Critical Challenge: All the prior work in [5], [6], [9]–[11], [16] assumed that the knowledge about the channels between the LIS and the transmitters/receivers is available at the base station, either perfectly or with some error. Obtaining this channel knowledge, however, is one of the most crucial challenges for LIS systems because of the massive number of antennas (LIS elements) and the hardware constraints on these elements. More specifically, if the LIS elements are implemented using phase shifters that just reflect the incident signals, then there are two main approaches for designing the LIS reflection matrix. The first approach is to estimate the LIS-assisted channels at the transmitter/receiver by training all the LIS elements, normally one by one, and then use the estimated channels to design the reflection matrix. This yields a massive channel training overhead because of the very large number of elements at the LIS. Instead of the explicit channel estimation, the LIS reflection matrix can be selected from quantized codebooks via online beam/reflection training. This is similar to the common beam training techniques in mmWave systems that employ similar phase shifter architectures [21], [22]. To sufficiently quantize the space, however, the size of the reflection codebooks needs normally to be in the order of the number of antennas, which leads to huge training overhead. To avoid this training overhead, a trivial solution is to employ fully-digital or hybrid analog/digital architectures at the LIS, where every antenna element is connected somehow to the baseband where channel estimation strategies can be used to obtain the channels [23]–[25]. This solution, however, leads to high hardware complexity and power consumption because of the massive number of LIS elements.

B. CONTRIBUTION

In this paper, we consider an LIS-assisted wireless communication system and propose a novel LIS architecture as well as compressive sensing and deep learning based solutions that design the LIS reflection matrix with negligible training overhead. More specifically, the contributions of this paper can be summarized as follows.

- *Novel LIS hardware architecture:* We introduce a new LIS architecture where all the elements are passive except a few randomly distributed active channel sensors. Only those few active sensors are connected to the baseband of the LIS controller and are used to enable the efficient design of the LIS reflection matrices with low training overhead.
- *Compressive sensing based LIS reflection matrix design:* Given the new LIS architecture with randomly distributed active elements, we develop a compressive sensing based solution to recover the full channels between the LIS and the transmitters/receivers from the *sampled* channels sensed at the few active elements. Using the constructed channels, we then design the LIS reflection matrices with no training overhead. We show that the proposed solution can efficiently design the LIS reflection matrices when only a small fraction of the LIS elements are active, yielding a promising solution for LIS systems from both energy efficiency and training overhead perspectives.
- *Deep learning based LIS reflection matrix design:* By leveraging deep learning tools, we propose a solution that learns the direct mapping from the sampled channels seen at the active LIS elements and the optimal LIS

reflection matrices that maximize the system achievable rate. Essentially, the proposed approach teaches the LIS system how to interact with the incident signal given the knowledge of the sampled channel vectors, that we call *environment descriptors*. The LIS learns that when it observes these environment descriptors, it should reflect the incident signal using this reflection matrix. Different from the compressive sensing solution, the deep learning approach leverages the prior observations at the LIS and does not require any knowledge of the array structure. It is worth mentioning that a conference version of this work is presented in [1].

The proposed solutions are extensively evaluated using the accurate ray-tracing based DeepMIMO dataset [26]. The results show that the developed compressive sensing and deep learning solutions can both approach the optimal upper bound, which assumes perfect channel knowledge, when only a few LIS elements are active and with almost no training overhead. We have investigated the use of deep learning approaches (supervised and reinforcement learning) to design the reflection matrices in some of our prior conference work [1], [27]. Therefore, in this work, we provide an in-depth analysis of the performance of the supervised deep learning solution. This is done through two avenues; the first avenue is a comparative study that pits the deep learning and compressive sensing solutions against each other for spectral and energy efficiencies. The second one is a detailed study of the impact of various system parameters on the deep learning performance. These two studies shed some light on how much adaptability and robustness an LIS-assisted system could gain from using deep learning.

The rest of the paper is organized as follows. Section II presents the system and channel models adopted in this paper. Section III presents the formal description of the main problem in this paper, the design of the LIS interaction matrix. Section IV proposes and discusses the novel sparse LIS architecture. Sections V and VI present, respectively, the proposed compressive sensing and deep learning solutions to the problem of designing the interaction matrix. Section VII puts the proposed architecture and solutions to test by investigating the performance of each solution and the effect of various design parameters. Finally, Section VIII concludes this paper with a summary of the findings and a few concluding remarks.

Notation: We use the following notation throughout this paper: $\mathbf{A}$ is a matrix, $\mathbf{a}$ is a vector, a is a scalar, $\mathcal{A}$ is a set of scalars, and $\boldsymbol{\mathcal{A}}$ is a set of vectors. $\|\mathbf{a}\|_p$ is the p-norm of $\mathbf{a}$. $|\mathbf{A}|$ is the determinant of $\mathbf{A}$, whereas $\mathbf{A}^T$, $\mathbf{A}^H$, $\mathbf{A}^*$, $\mathbf{A}^{-1}$, $\mathbf{A}^{\dagger}$ are its transpose, Hermitian (conjugate transpose), conjugate, inverse, and pseudo-inverse respectively. $[\mathbf{A}]_{r,:}$ and $[\mathbf{A}]_{:,c}$ are the r^{th} row and c^{th} column of the matrix $\mathbf{A}$ respectively. diag($\mathbf{a}$) is a diagonal matrix with the entries of $\mathbf{a}$ on its diagonal. $\mathbf{I}$ is the identity matrix. $\mathbf{1}_N$ and $\mathbf{0}_N$ are the N-dimensional all-ones and all-zeros vector respectively. $\mathbf{A} \odot \mathbf{B}$ and $\mathbf{A} \otimes \mathbf{B}$ are the Hadamard and Kronecker products of $\mathbf{A}$ and $\mathbf{B}$, respectively. $\mathcal{N}(\mathbf{m}, \mathbf{R})$ is a complex Gaussian random vector with mean $\mathbf{m}$ and covariance $\mathbf{R}$. $\mathbb{E}\left[\cdot\right]$ is used to denote expectation. $\mathbf{vec}(\mathbf{A})$ is a vector whose elements are the stacked columns of matrix $\mathbf{A}$.

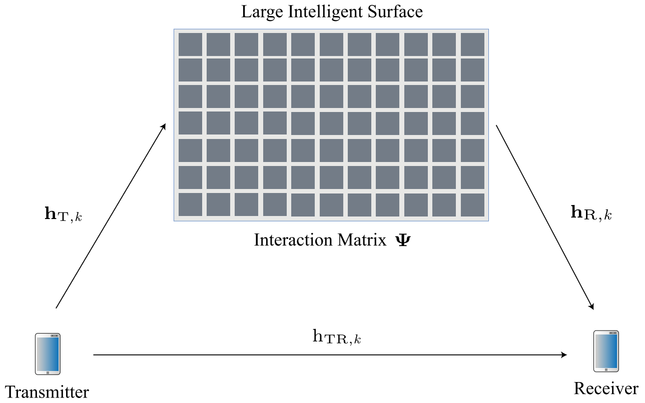

Large Intelligent Surface
$\mathbf{h}_{\mathrm{T},k}$
$\mathbf{h}_{\mathrm{R},k}$
Interaction Matrix $\boldsymbol{\Psi}$
$\mathrm{h}_{\mathrm{TR},k}$
Transmitter
Receiver

FIGURE 1. A block diagram of the adopted system model where the transmitter-receiver communication is assisted by a large intelligent surface (LIS). The LIS is interacting with the incident signal through an interaction matrix $\boldsymbol{\Psi}$.

II. SYSTEM AND CHANNEL MODELS

The adopted system and channel models for large intelligent surfaces (LISs) are described in this section.

A. SYSTEM MODEL

Consider a communication system where a transmitter is communicating with a receiver, and this communication is aided by a large intelligent surface (LIS), as depicted in Fig. 1. These transmitters/receivers can represent either base stations or user equipment. Let the LIS be equipped with M reconfigurable elements and assume that both the transmitter and receiver have a single-antenna. It is worth noting here that such an assumption is only adopted for simplicity of exposition and the proposed solutions and the results in this paper can be readily extended to multi-antenna transceivers. To put that description in formal terms, we adopt an OFDM-based system with K subcarriers. We define $h_{\mathrm{TR},k} \in \mathbb{C}$ as the direct channel between the transmitter and receiver at the k^{th} subcarrier, $\mathbf{h}_{\mathrm{T},k}, \mathbf{h}_{\mathrm{R},k} \in \mathbb{C}^{M\times 1}$ as the $M \times 1$ uplink channels from the transmitter and receiver to the LIS at the k^{th} subcarrier, and by reciprocity, $\mathbf{h}_{\mathrm{T},k}^T$, $\mathbf{h}_{\mathrm{R},k}^T$ as the downlink channels. The received signal at the receiver side could be expressed as

$$y_k = \underbrace{\mathbf{h}_{\mathrm{R},k}^T \boldsymbol{\Psi}_k \mathbf{h}_{\mathrm{T},k} s_k}_{\text{LIS-assisted link}} + \underbrace{h_{\mathrm{TR},k} s_k}_{\text{Direct link}} + n_k, \tag{1}$$

where the matrix $\boldsymbol{\Psi}_k \in \mathbb{C}^{M\times M}$, that we call the LIS interaction matrix, characterizes the *interaction* of the LIS with the incident (impinging) signal from the transmitter. s_k represents the transmitted signal over the k^{th} subcarrier, and satisfies the per-subcarrier power constraint $\mathbb{E}[|s_k|^2] = \frac{P_{\mathrm{T}}}{K}$, with P_{T} being the total transmit power. The receive noise is denoted by $n_k \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_n^2)$.

The overall objective of the LIS is then to interact with the incident signal (via adjusting $\boldsymbol{\Psi}_k$) in a way that optimizes a certain performance metric such as the system achievable rate or the network coverage. To simplify the design and analysis of the algorithms in this paper, we will focus on the case where the direct link does not exist. This represents the scenarios where the direct link is either blocked or has negligible receive power compared to that received through the LIS-assisted link. With this assumption, the receive signal can be expressed as

$$y_k = \mathbf{h}_{\mathrm{R},k}^T \boldsymbol{\Psi}_k \mathbf{h}_{\mathrm{T},k} s_k + n_k, \tag{2}$$

$$\overset{(a)}{=} \left(\mathbf{h}_{\mathrm{R},k} \odot \mathbf{h}_{\mathrm{T},k}\right)^T \boldsymbol{\psi}_k s_k + n_k, \tag{3}$$

where (a) follows from the diagonal structure of the interaction matrix $\boldsymbol{\Psi}_k$, whose diagonal entries could be stacked in a vector $\boldsymbol{\psi}_k \in \mathbb{C}^{M\times 1}$ such that $\boldsymbol{\Psi}_k = \operatorname{diag}\left(\boldsymbol{\psi}_k\right)$. This diagonal structure results from the LIS operation where every element $m, m \in \{1, 2, \ldots, M\}$, reflects only its incident signal after multiplying it with an interaction factor $\left[\boldsymbol{\psi}_k\right]_m$. Now, we make two important notes on these interaction vectors. First, while the interaction factors, $\left[\boldsymbol{\psi}_k\right]_m, \forall m, k$, can generally have different magnitudes (amplifying/attenuation gains), it is more practical to assume that the LIS elements are implemented using only phase shifters. Second, since the implementation of the phase shifters is done in the analog domain (using RF circuits), the same phase shift will be applied to the signals on all subcarriers, i.e., $\boldsymbol{\psi}_k = \boldsymbol{\psi}, \forall k$. Accounting for these practical considerations, we assume that every interaction factor is just a phase shifter, i.e., $[\boldsymbol{\psi}]_m = e^{j\phi_m}$. Further, we will call the interaction vector $\boldsymbol{\psi}$ in this case the *reflection beamforming* vector.

B. CHANNEL MODEL

In this paper, we adopt a wideband geometric channel model for the channels $\mathbf{h}_{\mathrm{T},k}, \mathbf{h}_{\mathrm{R},k}$ between the transmitter/receiver and the LIS [1], [18], [27]. Consider an uplink transmitter-LIS channel, $\mathbf{h}_{\mathrm{T},k} \in \mathbb{C}^{M\times 1}$, consisting of L clusters, each of which (i.e., ℓ^{th} cluster) contributes a single ray with a time delay $\tau_\ell \in \mathbb{R}$; azimuth/elevation angles of arrival, $\phi_\ell \in [0, 2\pi), \theta_\ell \in [0, \pi)$; an uplink path loss ρ_{T}; and a complex coefficient $\alpha_\ell \in \mathbb{C}$. Let $p(\tau)$ denotes the pulse shaping function for T_S-spaced signaling evaluated at τ seconds. Let the array response vector of the LIS at the angles of arrival, ϕ_ℓ, θ_ℓ, be defined as $\mathbf{a}(\phi_\ell, \theta_\ell) \in \mathbb{C}^{M\times 1}$. The delay-$d$ channel vector, $\boldsymbol{h}_{\mathrm{T},d} \in \mathbb{C}^{M\times 1}$, between the transmitter and the LIS can then be formulated as

$$\boldsymbol{h}_{\mathrm{T},d} = \sqrt{\frac{M}{\rho_{\mathrm{T}}}} \sum_{\ell=1}^{L} \alpha_\ell \, p(dT_S - \tau_\ell) \, \mathbf{a}\left(\theta_\ell, \phi_\ell\right), \tag{4}$$

Given this delay-d channel, the channel vector at subcarrier k, $\mathbf{h}_{\mathrm{T},k}$, can be defined in the frequency domain as

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

where D is the channel tap length. The downlink LIS-receiver channel $\mathbf{h}_{\mathrm{R},k}$ can be defined similarly. The channel vectors, $\left\{\mathbf{h}_{\mathrm{T},k}\right\}_{k=1}^{K}$ and $\left\{\mathbf{h}_{\mathrm{R},k}\right\}_{k=1}^{K}$, are assumed constant within the period of one coherence time, T_C, which mainly depends on the dynamics of the environment and the user mobility. It is worth noting that the number of channel paths L depends highly on the operational frequency band and the propagation environment. For example, mmWave channels normally consist of a few channel paths, ∼3-5 paths, [28]–[30], while sub-6 GHz signal propagation generally experiences rich scattering resulting in channels with more multi-path components.

III. PROBLEM FORMULATION

Given the system and channel models in Section II, our objective is to design the LIS interaction vector (reflection beamforming vector), $\boldsymbol{\psi} \in \mathbb{C}^{M\times 1}$, in order to maximize the achievable rate at the receiver, which can be formulated as

$$R = \frac{1}{K}\sum_{k=1}^{K} \log_2\left(1 + \mathsf{SNR}\left|\mathbf{h}_{\mathrm{R},k}^T \boldsymbol{\Psi} \mathbf{h}_{\mathrm{T},k}\right|^2\right), \tag{6}$$

$$= \frac{1}{K}\sum_{k=1}^{K} \log_2\left(1 + \mathsf{SNR}\left|\left(\mathbf{h}_{\mathrm{T},k} \odot \mathbf{h}_{\mathrm{R},k}\right)^T \boldsymbol{\psi}\right|^2\right), \tag{7}$$

where $\mathsf{SNR} = P_{\mathrm{T}}/\left(K\sigma_n^2\right)$ represents the signal-to-noise ratio. As mentioned in Section II-A, every element in the LIS reflection beamforming vector, $\boldsymbol{\psi}$, is implemented using an RF phase shifter. These phase shifters, however, normally have a quantized set of angles and cannot shift the signal with any phase. To capture this constraint, we assume that the reflection beamforming vector $\boldsymbol{\psi}$ can only be picked from a pre-defined codebook $\boldsymbol{\mathcal{P}}$. Every candidate reflection beamforming codeword in $\boldsymbol{\mathcal{P}}$ is assumed to be implemented using quantized phase shifters. With this assumption, our objective is then to find the optimal reflection beamforming vector $\boldsymbol{\psi}^\star$ that solves

$$\boldsymbol{\psi}^\star = \underset{\boldsymbol{\psi}\in\boldsymbol{\mathcal{P}}}{\arg\max} \sum_{k=1}^{K} \log_2\left(1 + \mathsf{SNR}\left|\left(\mathbf{h}_{\mathrm{T},k} \odot \mathbf{h}_{\mathrm{R},k}\right)^T \boldsymbol{\psi}\right|^2\right), \tag{8}$$

to result in the optimal rate $R^\star$ defined as

$$R^\star = \max_{\boldsymbol{\psi}\in\boldsymbol{\mathcal{P}}} \frac{1}{K}\sum_{k=1}^{K} \log_2\left(1 + \mathsf{SNR}\left|\left(\mathbf{h}_{\mathrm{T},k} \odot \mathbf{h}_{\mathrm{R},k}\right)^T \boldsymbol{\psi}\right|^2\right). \tag{9}$$

The optimization problem in (8), unfortunately, has no close-form solution. This is a consequence of (a) the time-domain implementation of the reflection beamforming vector, i.e., using only one vector $\boldsymbol{\psi}$ for all subcarriers, and (b) the quantized codebook constraint.

The main challenge: As characterized in (8), finding the optimal LIS interaction vector $\boldsymbol{\psi}^\star$ and achieving the optimal rate $R^\star$ requires an exhaustive search over the codebook $\boldsymbol{\mathcal{P}}$. Note that the codebook size should normally be in the same

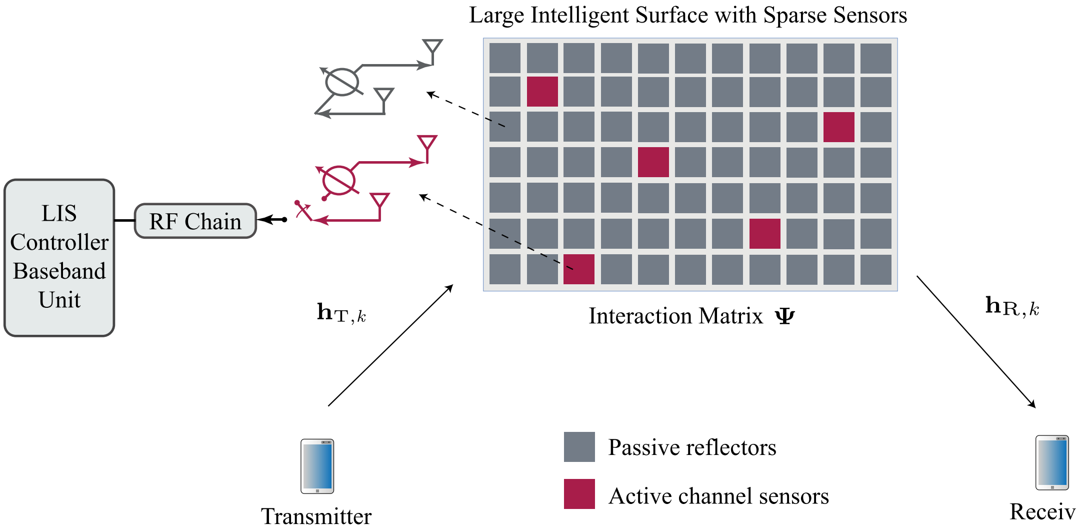

Large Intelligent Surface with Sparse Sensors
LIS Controller Baseband Unit
RF Chain
$\mathbf{h}_{\mathrm{T},k}$
Interaction Matrix $\mathbf{\Psi}$
$\mathbf{h}_{\mathrm{R},k}$
Passive reflectors
Active channel sensors
Transmitter
Receiver

FIGURE 2. This figure illustrates the proposed LIS architecture where $\overline{M}$ active channel sensors are randomly distributed over the LIS. These active elements have two modes of operation: (i) a channel sensing mode where it is connected to the baseband and is used to estimate the channels and (ii) a reflection mode where it just reflects the incident signal by applying a phase shift. The rest of the LIS elements are passive reflectors and are not connected to the baseband.

order as the number of antennas to make use of these antennas. This means that a reasonable reflection beamforming codebook for LIS systems will probably have thousands of candidate codewords. With such huge codebooks, solving the exhaustive search in (8) is very challenging. More specifically, there are two main approaches for performing the search in (8).

- **Full channel estimation with offline exhaustive search:** In this approach, we need to estimate the full channels between the LIS and the transmitter/receiver, $\mathbf{h}_{\mathrm{T},k}$, $\mathbf{h}_{\mathrm{R},k}$ and use it to find the best reflection beamforming vector by the offline calculation of (8). Estimating these channel vectors, however, requires the LIS to employ a complex hardware architecture that connects all the antenna elements to a baseband processing unit either through a fully-digital or hybrid analog/digital architectures [23], [24]. Given the massive numbers of antennas at large intelligent surfaces, this approach can yield **prohibitive hardware complexity** in terms of routing and power consumption among others. If the LIS is operated and controlled via a base station or an access point [5], then this channel estimation process can be done at these communication ends. This, however, assumes an orthogonal training over the LIS antennas, for example by activating one LIS antenna at a time, which leads to **prohibitive training overhead** given the number of antennas at the LIS.
- **Online exhaustive beam training:** Instead of the explicit channel estimation, the best LIS beam reflection vector $\boldsymbol{\psi}^\star$ can be found through an over-the-air beam training process. This process essentially solves the exhaustive search in (8) by testing the candidate interaction vectors $\boldsymbol{\psi} \in \mathcal{P}$ one by one. This exhaustive beam training process, however, incurs again **very large training overhead** at the LIS systems.

Our objective in this paper is to enable large intelligent surfaces by addressing this main challenge. More specifically, our objective is to enable LIS systems to approach the optimal achievable rate in (9) by adopting **low-complexity hardware architectures** and requiring **low training overhead**. For this objective, we first propose a novel energy-efficient LIS transceiver architecture in Section IV. Then, we show in Sections V-VI how to employ this LIS architecture to achieve near-optimal achievable rates with negligible training overhead via leveraging tools from compressive sensing and deep learning.

IV. LARGE INTELLIGENT SURFACES WITH SPARSE SENSORS: A NOVEL ARCHITECTURE

As discussed in Section III, a main challenge for the LIS system operation lies in the high hardware complexity and training overhead associated with designing the LIS interaction (reflection beamforming) vector, $\boldsymbol{\psi}$. To overcome this challenge and enable LIS systems in practice, we adopt a novel LIS architecture that relies on sparsely embedded active sensors. To further illustrate this architecture, consider the LIS depicted in Fig. 2, which consists of (i) a set of M passive reflecting elements and (ii) another set of $\overline{M}$ *active* channel sensors such that $\overline{M} \ll M$. The M passive elements are all implemented using RF phase shifters, and they are not connected to the baseband unit. On the other hand, the $\overline{M}$ active sensors are assumed to be selected from the passive sensors in the LIS. In particular, those sensors are designed to have two modes of operation (as shown in Fig. 2): (i) A channel sensing mode where they work as receivers with full RF chains and baseband processing, and (ii) a reflection mode where they act just like the rest of the passive elements that reflect the incident signal.

Before proceeding further, we need to emphasize two important points. First, while we describe the M phase-shifting elements as passive elements, they are normally implemented using reconfigurable active RF circuits [4], [31]. We just adopt that terminology to differentiate them from the active channel sensors, i.e., they are passive in the sense that they do not provide any sensing information to the baseband. Second, our proposed architecture is different from the one proposed in [32], where an all-passive LIS assists the multiuser communication systems enabled by an all-active access point. Next, we define the channels from the transmitter/receiver to the active channel sensors of the LIS, and then we discuss how to leverage this energy-efficient LIS architecture for designing the LIS interaction vector $\boldsymbol{\psi}$.

Sampled channel vectors: We define the $\overline{M} \times 1$ uplink *sampled* channel vector, $\overline{\mathbf{h}}_{\mathrm{T},k} \in \mathbb{C}^{\overline{M}\times 1}$, as the channel vector from the transmitter to the $\overline{M}$ active elements at the LIS. This vector can then be expressed as

$$\overline{\mathbf{h}}_{\mathrm{T},k} = \mathbf{G}_{\mathrm{LIS}}\, \mathbf{h}_{\mathrm{T},k}, \tag{10}$$

where $\mathbf{G}_{\mathrm{LIS}}$ is an $\overline{M} \times M$ selection matrix that selects the entries of the original channel vector, $\mathbf{h}_{\mathrm{T},k}$, that correspond to the active LIS elements. If $\mathcal{A}$ defines the set of indices of the active LIS antenna elements, $|\mathcal{A}| = \overline{M}$, then $\mathbf{G}_{\mathrm{LIS}} = [\mathbf{I}]_{\mathcal{A},:}$, i.e., $\mathbf{G}_{\mathrm{LIS}}$ includes the rows of the $M \times M$ identity matrix, $\mathbf{I}$, that correspond to the indices of the active elements. The sampled channel vector, $\overline{\mathbf{h}}_{\mathrm{R},k} \in \mathbb{C}^{\overline{M}\times 1}$, from the receiver to the $\overline{M}$ active sensors of the LIS is similarly defined. Finally, $\overline{\mathbf{h}}_{k} = \overline{\mathbf{h}}_{\mathrm{T},k} \odot \overline{\mathbf{h}}_{\mathrm{R},k}$ is defined as the overall LIS sampled channel vector at the k^{th} subcarrier.

Designing the LIS interaction vector: In the system model and the proposed LIS architecture in, respectively, Section II-A and Fig. 2, the sampled channel vectors $\overline{\mathbf{h}}_{\mathrm{T},k}, \overline{\mathbf{h}}_{\mathrm{R},k}$ can easily be estimated. This is done by, for example, using an uplink training approach, in which the transmitter can send a single pilot that is simultaneously processed with all active elements to get $\overline{\mathbf{h}}_{\mathrm{T},k}$. The same approach could also be followed to estimate $\overline{\mathbf{h}}_{\mathrm{R},k}$. With the knowledge of these two sampled channels, the critical question now becomes: can we use them to select the optimal reflection beamforming vector $\boldsymbol{\psi}^{\star}$ that solves (9)? The next two sections propose two approaches for addressing this problem leveraging compressive sensing (in Section V) and deep learning (in Section VI).

V. COMPRESSIVE SENSING BASED LIS INTERACTION DESIGN

As shown in Section III, finding the optimal LIS interaction (reflection beamforming) vector $\boldsymbol{\psi}^{\star}$ that maximizes the achievable rate with no beam training overhead requires the availability of the full channel vectors $\mathbf{h}_{\mathrm{T},k}, \mathbf{h}_{\mathrm{R},k}$. Estimating these channel vectors at the LIS, however, normally requires that every LIS antenna gets connected to the baseband processing unit through a fully-digital or hybrid architecture [23], [25], [33]. This can massively increase the hardware complexity with the large number of antennas at the LIS systems. In this section, and adopting the low-complexity LIS architecture proposed in Section IV, we show that it is possible to recover the full channel vectors $\mathbf{h}_{\mathrm{T},k}, \mathbf{h}_{\mathrm{R},k}$ from the sampled channel vectors $\overline{\mathbf{h}}_{\mathrm{T},k}, \overline{\mathbf{h}}_{\mathrm{R},k}$ when the channels experience sparse scattering. This is typically the case in mmWave and LOS-dominant sub-6 GHz systems.

A. RECOVERING FULL CHANNELS FROM SAMPLED CHANNELS

With the proposed LIS architecture in Fig. 2, the LIS can easily estimate the *sampled* channel vectors $\overline{\mathbf{h}}_{\mathrm{T},k}, \overline{\mathbf{h}}_{\mathrm{R},k}$ through uplink training from the transmitter and receiver to the LIS with a few pilots. Next, we explain how to use these sampled channel vectors to estimate the full channel vectors $\mathbf{h}_{\mathrm{T},k}, \mathbf{h}_{\mathrm{R},k}$. First, note that the $\mathbf{h}_{\mathrm{T},k}$ in (4), (5) (and similarly for $\mathbf{h}_{\mathrm{R},k}$) can be written as

$$\mathbf{h}_{\mathrm{T},k} = \sqrt{\frac{M}{\rho_{\mathrm{T}}}} \sum_{d=0}^{D-1} \sum_{\ell=1}^{L} \alpha_\ell \, p(dT_S - \tau_\ell)\, \mathbf{a}\left(\theta_\ell, \phi_\ell\right) e^{-j\frac{2\pi k}{K}d}, \tag{11}$$

$$= \sum_{\ell=1}^{L} \beta_{\ell,k}\, \mathbf{a}\left(\theta_\ell, \phi_\ell\right), \tag{12}$$

where $\beta_{\ell,k} = \sqrt{\frac{M}{\rho_{\mathrm{T}}}} \alpha_\ell \sum_{d=0}^{D-1} p(dT_S - \tau_\ell) e^{-j\frac{2\pi k}{K}d}$. Further, by defining the array response matrix $\mathbf{A}$ and the k^{th} subcarrier path gain vector $\boldsymbol{\beta}_k$ as

$$\mathbf{A} = \left[\mathbf{a}\left(\theta_1, \phi_1\right), \mathbf{a}\left(\theta_2, \phi_2\right) \ldots, \mathbf{a}\left(\theta_L, \phi_L\right)\right], \tag{13}$$

$$\boldsymbol{\beta}_k = \left[\beta_{1,k}, \beta_{2,k}, \ldots, \beta_{L,k}\right]^T, \tag{14}$$

we can write $\mathbf{h}_{\mathrm{T},k}$ in a more compact way as $\mathbf{h}_{\mathrm{T},k} = \mathbf{A}\, \boldsymbol{\beta}_k$. Now, we note that in several important scenarios, such as mmWave and LOS-dominant sub-6 GHz, the channel experiences sparse scattering, which results is a small number of paths L [24], [29]. In order to leverage this sparsity, we follow [25] and define the dictionary of array response vectors $\mathbf{A}_{\mathrm{D}}$, where every column constructs an array response vector in one quantized azimuth and elevation direction. For example, if the LIS adopts a uniform planar array (UPA) structure, then we can define $\mathbf{A}_{\mathrm{D}}$ as

$$\mathbf{A}_{\mathrm{D}} = \mathbf{A}_{\mathrm{D}}^{\mathrm{Az}} \otimes \mathbf{A}_{\mathrm{D}}^{\mathrm{El}} \tag{15}$$

with $\mathbf{A}_{\mathrm{D}}^{\mathrm{Az}}$ and $\mathbf{A}_{\mathrm{D}}^{\mathrm{El}}$ being the dictionaries of the azimuth and elevation array response vectors. Every column in $\mathbf{A}_{\mathrm{D}}^{\mathrm{Az}}$ (and similarly for $\mathbf{A}_{\mathrm{D}}^{\mathrm{El}}$) constructs an azimuth array response in one quantized azimuth (elevation) direction. If the number of grid points in the azimuth and elevation dictionaries is $N_{\mathrm{D}}^{\mathrm{Az}}$ and $N_{\mathrm{D}}^{\mathrm{El}}$, respectively, and the number of horizontal and vertical elements of the UPA is $M_{\mathrm{H}}, M_{\mathrm{V}}$, where $M = M_{\mathrm{H}} M_{\mathrm{V}}$, then $\mathbf{A}_{\mathrm{D}}$ has dimensions $M \times N_{\mathrm{D}}^{\mathrm{Az}} N_{\mathrm{D}}^{\mathrm{El}}$. Now, assuming that size of the grid is large enough such that the azimuth and elevation angles $\theta_\ell, \phi_\ell, \forall \ell$ matches exactly L points in this grid (which is a common assumption in the formulations of the sparse channels estimation approaches [24], [25], [34]), then we can rewrite $\mathbf{h}_{\mathrm{T},k}$ as

$$\mathbf{h}_{\mathrm{T},k} = \mathbf{A}_{\mathrm{D}}\, \mathbf{x}_{\boldsymbol{\beta},k}, \tag{16}$$

where $\mathbf{x}_{\boldsymbol{\beta},k}$ is an $N_{\mathrm{D}}^{\mathrm{Az}} N_{\mathrm{D}}^{\mathrm{El}}$ sparse vector with $L \ll N_{\mathrm{D}}^{\mathrm{Az}} N_{\mathrm{D}}^{\mathrm{El}}$ non-zero entries equal to the elements of $\boldsymbol{\beta}_k$. Further, these non-zero entries are in the positions that correspond to the channel azimuth/elevation angles of arrival. Next, let $\widehat{\overline{\mathbf{h}}}_{\mathrm{T},k}$ denote the noisy sampled channel vectors, then we can write

$$\begin{aligned} \widehat{\overline{\mathbf{h}}}_{\mathrm{T},k} &= \mathbf{G}_{\mathrm{LIS}} \mathbf{h}_{\mathrm{T},k} + \mathbf{v}_k, && (17)\\ &= \mathbf{G}_{\mathrm{LIS}} \mathbf{A}_{\mathrm{D}}\, \mathbf{x}_{\boldsymbol{\beta},k} + \mathbf{v}_k, && (18)\\ &= \boldsymbol{\Phi}\, \mathbf{x}_{\boldsymbol{\beta},k} + \mathbf{v}_k, && (19) \end{aligned}$$

where $\mathbf{v}_k \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, \sigma_n^2 \mathbf{I}\right)$ represent the receive noise vector at the LIS active channel sensors and $\mathbf{G}_{\mathrm{LIS}}$ is the selection matrix defined in (10). Now, given the equivalent sensing matrix, $\boldsymbol{\Phi}$ and the noisy sampled channel vector $\widehat{\overline{\mathbf{h}}}_{\mathrm{T},k}$, the objective is to estimate the sparse vector $\mathbf{x}_{\boldsymbol{\beta},k}$ that solves the non-convex combinatorial problem

$$\min \left\| \mathbf{x}_{\boldsymbol{\beta},k} \right\|_0 \quad \text{s.t.} \quad \left\| \widehat{\overline{\mathbf{h}}}_{\mathrm{T},k} - \boldsymbol{\Phi}\, \mathbf{x}_{\boldsymbol{\beta},k} \right\|_2 \leq \sigma. \tag{20}$$

Given the sparse formulation in (20), several compressive sensing reconstruction algorithms, such as orthogonal matching pursuit (OMP) [35], [36], can be employed to find an approximate solution for $\mathbf{x}_{\boldsymbol{\beta},k}$. With this solution for $\mathbf{x}_{\boldsymbol{\beta},k}$, the full channel vector $\mathbf{h}_{\mathrm{T},k}$ can be constructed according to (16). Finally, the constructed full channel vector can be used to find the best LIS reflection beamforming vector, $\boldsymbol{\psi}_{n^{\mathrm{CS}}} \in \boldsymbol{\mathcal{P}}$, out of the codebook $\boldsymbol{\mathcal{P}}$, via an offline search using (8).

In this paper, we assume for simplicity that the $\overline{M}$ active channel sensors are randomly selected from the M LIS elements, assuming that all the elements are equally likely to be selected. It is important, however, to note that the specific selection of the active elements designs the compressive sensing matrix $\boldsymbol{\Phi}$ and decides its properties. Therefore, it is interesting to explore the optimization of the active element selection, leveraging tools from nested arrays [37], co-prime arrays [38], [39], incoherence frames [40], and difference sets [33], [41].

B. SIMULATION RESULTS AND DISCUSSION

To evaluate the performance of the proposed compressive sensing based solution, we consider a simulation setup at two different carrier frequencies, namely 3.5GHz and 28GHz. The simulation setup consists of one large intelligent surface with a uniform planar array (UPA) in the y-z plane, which reflects the signal coming from one transmitter to another receiver, as depicted in Fig. 6. This UPA consists of 16 × 16 antennas at 3.5GHz and 64 × 64 antennas at 28GHz. We generate the channels using the publicly available ray-tracing based DeepMIMO dataset [26], with the 'O1' scenario that consists of a street and buildings on the sides of the street. Please refer to Section VII-A for a detailed description of the simulation setup and its parameters.

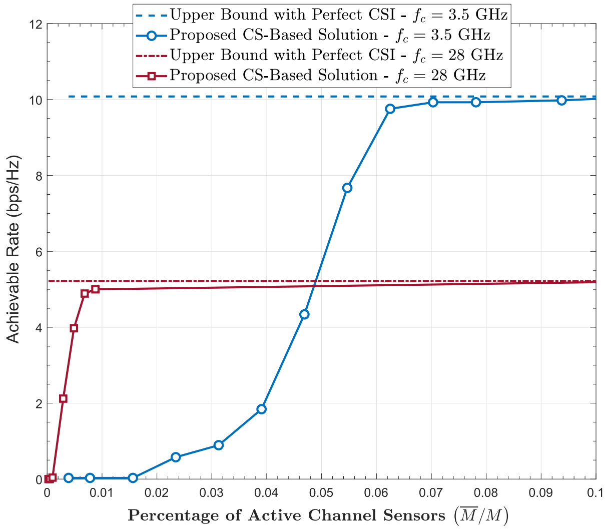


FIGURE 3. This figure plots the achievable rates using the proposed compressive sensing based solution for two scenarios, namely a mmWave 28GHz scenario and a low-frequency 3.5GHz one. These achievable rates are compared to the optimal rate $R^\star$ in (9) that assumes perfect channel knowledge. This figure illustrates the potential of the proposed solutions that approach the upper bound, while requiring only a small fraction of the total LIS elements to be active.

Given this described setup, and adopting the novel LIS architecture in Fig. 2, we apply the proposed compressive-sensing based solution described in Section V-A as follows: (i) We obtain the channel vectors $\mathbf{h}_{\mathrm{T},k}, \mathbf{h}_{\mathrm{R},k}$ using the ray-tracing based DeepMIMO dataset, and add noise with the noise parameters described in Section VII-A. (ii) Adopting the LIS architecture in Fig. 2, we randomly select $\overline{M}$ elements to be active and construct the sampled channel vectors $\widehat{\overline{\mathbf{h}}}_{\mathrm{T},k}, \widehat{\overline{\mathbf{h}}}_{\mathrm{R},k}$. (iii) Using OMP with a grid of size $N_{\mathrm{D}}^{\mathrm{Az}} N_{\mathrm{D}}^{\mathrm{El}}, N_{\mathrm{D}}^{\mathrm{Az}} = 2M_{\mathrm{H}}, N_{\mathrm{D}}^{\mathrm{El}} = 2M_{\mathrm{V}}$, we recover an approximate solution of the full channel vectors and use this to search for the optimal LIS interaction vector using (8). The achievable rate using this proposed compressive sensing based solution is shown in Fig. 3 compared to the upper bound with perfect *full* channel state information (CSI), $\mathbf{h}_{\mathrm{T},k}$ and $\mathbf{h}_{\mathrm{R},k}$, calculated according to (9).

Gains and Limitations: In Fig. 3, we plot the achievable rates of the proposed compressive sensing based solution and upper bound versus the ratio of the active elements to the total number of antennas, i.e., $\overline{M}/M$. As shown in this figure, the proposed novel LIS architecture with the compressive sensing based solution can achieve almost the optimal rate with a small fraction of the LIS antennas being active. This illustrates the significant saving in power consumption that can be achieved using the LIS architecture in Fig. 2 that includes a few active channel sensors. Further, since the LIS reflection beamforming vector $\boldsymbol{\psi}$ is obtained through an offline search with no beam training, the proposed solution approaches the optimal rate with negligible training overhead, ideally with two uplink pilots to estimate $\widehat{\overline{\mathbf{h}}}_{\mathrm{T},k}, \widehat{\overline{\mathbf{h}}}_{\mathrm{R},k}$. This enables the proposed LIS systems to support highly mobile applications such as vehicular communications and wireless virtual/augmented reality.

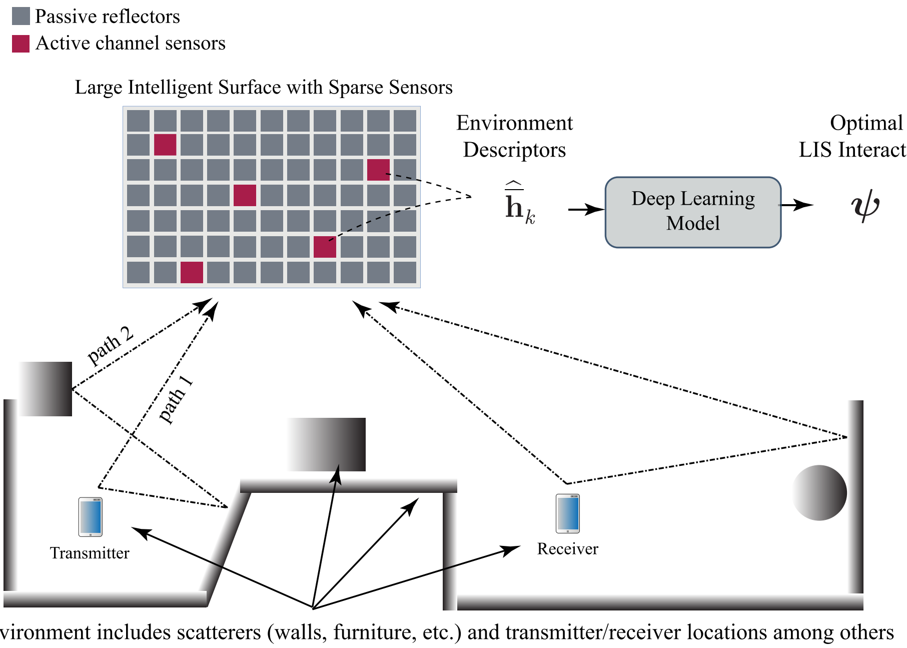


FIGURE 4. This figure summarizes the key idea of the proposed deep learning solution. The sampled channel vectors are considered as environment descriptors as they define, with some resolution, the transmitter/receiver locations and the surrounding environment. The deep learning model learns how to map the observed environment descriptors to the optimal LIS reflection vector.

Despite this interesting gain of the proposed compressive sensing based solution, it has some limitations. First, recovering the full channel vectors from the sampled ones according to Section V-A requires the knowledge of the array geometry and is hard to extend to LIS systems with unknown array structures. Second, the compressive sensing solution relies on the sparsity of the channels and its performance becomes limited in scenarios with rich NLOS scattering. This is shown in Fig. 3 as the compressive sensing based solution requires a higher ratio of the LIS elements to be active to approach the upper bound in the 3.5GHz scenario that has more scattering than the mmWave 28GHz case. Further, the compressive sensing solution does not leverage previous observations to improve the current channel recovery. These limitations motivate the deep learning based solution that we propose in the following section.

VI. DEEP LEARNING BASED LIS INTERACTION DESIGN

In this section, we introduce a *novel* application of deep learning in the reflection beamforming design problem of large intelligent surfaces. The section is organized as follows: First, the key idea of the proposed deep learning (DL) based reflection beamforming design is explained. Then, the system operation and the adopted deep learning model are diligently described. We refer the interested reader to [42] for a brief background on deep learning.

A. THE KEY IDEA

The large intelligent surfaces are envisioned as key components of future networks [5]. These surfaces will interact with the incident signals, for example by reflecting them, in a way that improves the wireless communication performance. To decide on this interaction, however, the LIS systems or their operating base stations and access points need to acquire some knowledge about the channels between the LIS and the transmitter/receiver. As we explained in Section III, the massive number of antennas at these surfaces makes obtaining the required channel knowledge associated with (i) prohibitive training overhead if all the LIS elements are passive or (ii) infeasible hardware complexity/power consumption in the case of fully-digital or hybrid based LIS architectures.

The channel vectors/matrices, however, are intuitively some functions of the various elements of the surrounding environment such as the geometry, scatterer materials, and the transmitter/receiver locations among others. Unfortunately, the nature of this function—its dependency on the various components of the environment—makes its mathematical modeling very hard and infeasible in many cases. This dependence, though, means that the interesting role the LIS is playing could be enabled with some form of awareness about the surrounding environment. With this motivation, and adopting the proposed LIS architecture in Fig. 2, we propose to utilize the sampled channels seen by the few active elements of the LIS as *environment descriptors*. These descriptors are expected to capture some information about the multi-path signature [18]–[20], as shown in Fig. 4. By tapping into the environment-specific information in those descriptors, a prediction on the optimal LIS interaction vector could be made using a deep learning algorithm. The algorithm is simply expected to learn a mapping function that relates the descriptor vector space with that of the LIS interaction vector. **In an abstract sense, this could be seen as teaching the LIS system how to interact with the wireless signal given the knowledge of the environment descriptors.** This is a desirable ability for the LIS to have, especially considering that the sampled channel vectors can be obtained with negligible training overhead as explained in Section IV. Ideally, the algorithm will learn a perfect prediction function that maps an environment descriptor to the optimal interaction vector, which means the LIS can approach the optimal rate in (9) with negligible training overhead and with low-complexity architectures (as only a few elements of the LIS are active).

B. PROPOSED SYSTEM OPERATION

In this section, we describe the system operation of the proposed deep learning based LIS interaction approach. The proposed system operates in two phases, namely (I) the learning phase and (II) the prediction phase.

Learning phase: In this phase, the LIS employs an exhaustive search reflection beamforming approach, as will be explained shortly, while it is collecting the dataset for the deep learning model. Once the dataset is fully acquired, the LIS trains the deep learning model, which in turn will be leveraged in the prediction phase. Let the term "data sample" indicates the data point captured in one coherence block, and define the concatenated *sampled* channel vector as $\bar{\mathbf{h}} = vec\left(\left[\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \ldots, \bar{\mathbf{h}}_K\right]\right)$. Further, let $\bar{\mathbf{h}}(s)$ denotes the concatenated *sampled* channel vector at the s^{th} coherence block, where $s = 1, \ldots, S$ and S is the total number of data samples used to construct the learning dataset. As depicted in Algorithm 1, at every coherence block s, the proposed LIS system operation consists of four steps, namely (1) estimating the sampled channel vector, (2) exhaustive beam training, (3) constructing a new data point for the learning dataset, and (4) data transmission. After collecting the whole dataset with S data samples, the deep learning model is trained. We describe these steps in detail as follows.

1. Sampled channel estimation (lines 1,2): For every channel coherence block s, the transmitter and receiver transmit two orthogonal uplink pilots. The LIS active elements will receive these pilots and estimate the *sampled* channel vectors to construct the multipath signature, which is expressed as

$$\widehat{\bar{\mathbf{h}}}_{\mathrm{T},k}(s) = \bar{\mathbf{h}}_{\mathrm{T},k}(s) + \mathbf{v}_k, \; \widehat{\bar{\mathbf{h}}}_{\mathrm{R},k}(s) = \bar{\mathbf{h}}_{\mathrm{R},k}(s) + \mathbf{w}_k, \tag{21}$$

$$\widehat{\bar{\mathbf{h}}}_k(s) = \widehat{\bar{\mathbf{h}}}_{\mathrm{T},k}(s) \odot \widehat{\bar{\mathbf{h}}}_{\mathrm{R},k}(s), \tag{22}$$

$$\widehat{\bar{\mathbf{h}}}(s) = vec\left(\left[\widehat{\bar{\mathbf{h}}}_1(s), \widehat{\bar{\mathbf{h}}}_2(s), \ldots, \widehat{\bar{\mathbf{h}}}_K(s)\right]\right). \tag{23}$$

where $\mathbf{v}_k, \mathbf{w}_k \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, \sigma_n^2 \mathbf{I}\right)$ are the receive noise vectors at the LIS active channel sensors.

Algorithm 1 Deep Learning Based Reflection Beamforming Prediction

```
Input: Reflection beamforming codebook $\boldsymbol{\mathcal{P}}$.
    Phase I: Learning phase
 1: for $s = 1$ to $S$ do        ▷ For every channel coherence block
 2:     LIS receives two pilots to estimate $\widehat{\bar{\mathbf{h}}}(s)$.
 3:     for $n = 1$ to $|\boldsymbol{\mathcal{P}}|$ do        ▷ Beam training
 4:         LIS reflects using $\boldsymbol{\psi}_n$ beam.
 5:         LIS receives the feedback $R_n(s)$.
 6:     Construct $\mathbf{r}(s) = \left[R_1(s), R_2(s), \ldots, R_{|\boldsymbol{\mathcal{P}}|}(s)\right]^T$.
 7:     Store new entry in the learning dataset,
        $\mathcal{D} \leftarrow \left(\widehat{\bar{\mathbf{h}}}(s), \mathbf{r}(s)\right)$.
 8:     LIS reflects using $\boldsymbol{\psi}_{n^\star}$ beam, $n^\star = \arg\max_n \left[\mathbf{r}(s)\right]_n$.
 9: Train the DL model using the learning dataset $\mathcal{D}$.

    Phase II: Prediction phase
10: while True do        ▷ For every channel coherence block
11:     LIS receives two pilots to estimate $\widehat{\bar{\mathbf{h}}}$.
12:     Predict the rate vector $\widehat{\mathbf{r}}$ using the trained DL model.
13:     LIS reflects using $\boldsymbol{\psi}_{n^{\mathrm{DL}}}$ beam, $n^{\mathrm{DL}} = \arg\max_n \left[\widehat{\mathbf{r}}\right]_n$.
```

2. Exhaustive beam training (lines 3-6): In this step, the LIS performs an exhaustive search over reflection codewords using the reflection codebook $\boldsymbol{\mathcal{P}}$. Specifically, the LIS attempts every candidate reflection beamforming vector, $\boldsymbol{\psi}_n, n = 1, \ldots, |\boldsymbol{\mathcal{P}}|$, and receives a feedback from the receiver indicating the achievable rate attained by using this interaction vector, $R_n(s)$, which is defined as

$$R_n(s) = \frac{1}{K} \sum_{k=1}^{K} \log_2\left(1 + \mathsf{SNR}\left|\left(\mathbf{h}_{\mathrm{T},k}(s) \odot \mathbf{h}_{\mathrm{R},k}(s)\right)^T \boldsymbol{\psi}_n\right|^2\right). \tag{24}$$

Note that, in practice, the computation and feedback of the achievable rate $R_n(s)$ will have some error compared to (24) because of the limitations in the pilot sequence length and feedback channel, which are neglected in this paper. For the rest of this paper, we define the achievable rate vector at the s^{th} coherence block as $\mathbf{r}(s) = \left[R_1(s), R_2(s), \ldots, R_{|\boldsymbol{\mathcal{P}}|}(s)\right]^T$.

3. Learning dataset update (line 7): The new data entry comprised of the sampled channel vector $\widehat{\bar{\mathbf{h}}}(s)$, estimated in step (1), and the corresponding rate vector $\mathbf{r}(s)$, constructed in step (2), is added to the deep learning dataset $\mathcal{D}$, such that $\mathcal{D} \leftarrow (\widehat{\bar{\mathbf{h}}}(s), \mathbf{r}(s))$.

4. Data transmission (line 8): After the beam training task, given the constructed achievable rate vector $\mathbf{r}(s)$, the best reflection beamforming vector, $\boldsymbol{\psi}_{n^\star}$, that corresponds to the highest achievable rate, where $n^\star = \arg\max_n \left[\mathbf{r}(s)\right]_n$, is used to reflect the transmitted data from the transmitter for the rest of the coherence block.

5. Deep learning model training (line 9): After acquiring the data entries for all S coherence blocks, the deep learning model is trained using the entire dataset $\mathcal{D}$. This model learns how to map an input (the *sampled* channel vector $\widehat{\bar{\mathbf{h}}}$) to an output (predicted achievable rate with every candidate interaction vector $\widehat{\mathbf{r}} = \left[\widehat{R}_1, \widehat{R}_2, \ldots, \widehat{R}_{|\boldsymbol{\mathcal{P}}|}\right]$, as shown in Fig. 5. It is worth mentioning here that while we assume that the system will switch one time to Phase II after the deep learning model

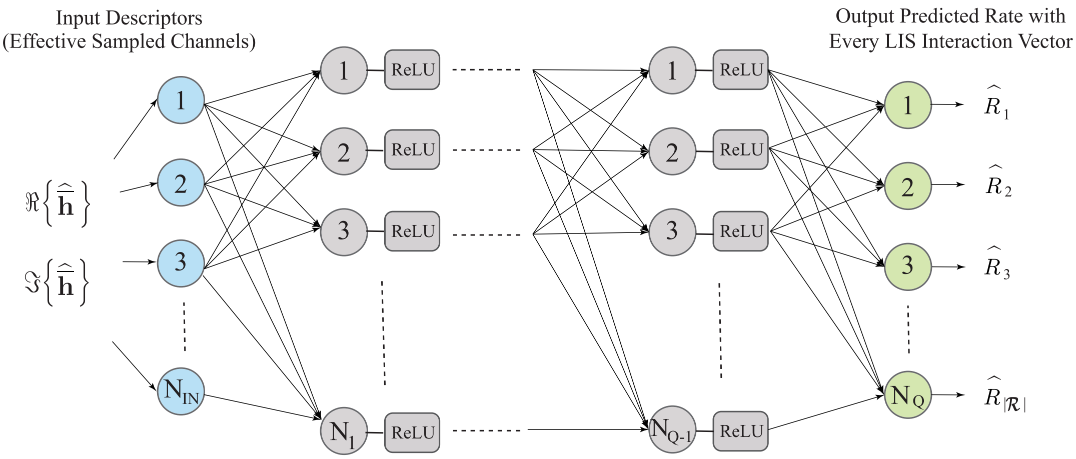

Input Descriptors
(Effective Sampled Channels)
$\Re\{\widehat{\bar{\mathbf{h}}}\}$
$\Im\{\widehat{\bar{\mathbf{h}}}\}$
1
2
3
N_{IN}
1
2
3
N_1
ReLU
1
2
3
N_{Q-1}
Output Predicted Rate with
Every LIS Interaction Vector
1
2
3
N_Q
$\widehat{R}_1$
$\widehat{R}_2$
$\widehat{R}_3$
$\widehat{R}_{|\mathcal{R}|}$

FIGURE 5. The adopted neural network architecture consists of Q fully connected layers. Each layer is followed by a non-linear ReLU activation layer. The deep learning model learns how to map the observed sampled channel vectors to the predicted achievable rate using every LIS interaction vector.

is trained, the system will need to retrain and refine the model frequently to account for the changes in the environment.

Prediction phase: Following the deep learning model training in the learning phase, the LIS leverages the trained model to predict the reflection beamforming vector directly from the estimated *sampled* channel vector, $\widehat{\bar{\mathbf{h}}}$. As shown in Algorithm 1, Phase II performs the following steps repeatedly for every channel coherence block.

1. Sampled channel estimation (line 11): This step is the same as the first step in the learning phase. The active elements of the LIS receive uplink pilots to estimate and construct the concatenated *sampled* channel vector, $\widehat{\bar{\mathbf{h}}}$.

2. Achievable rate prediction (line 12): In this step, the estimated *sampled* channel vector, $\widehat{\bar{\mathbf{h}}}$, is fed into the trained deep learning model. It predicts the achievable rate vector, $\widehat{\mathbf{r}}$, which is used to identify the best DL-based reflection beamforming vector.

3. Data transmission (line 13): In this step, the predicted deep learning reflection beamforming vector, $\boldsymbol{\psi}_{n^{\mathrm{DL}}}$, that corresponds to the highest predicted achievable rate, where $n^{\mathrm{DL}} = \arg\max_n [\widehat{\mathbf{r}}]_n$, is used to reflect the transmitted data from the transmitter for the rest of the coherence block. Note that instead of selecting only the interaction vector with the highest predicted achievable rate, the LIS can generally select the k_{B} beams corresponding to the k_{B} highest predicted achievable rates. It can then refine this set of beams online with the receiver to select the one with the highest achievable rate. In Section VII-F, we evaluate the performance gain if more than one reflection beam, i.e. k_{B} reflection beams, are selected.

C. DEEP LEARNING MODEL

Recent advances in machine learning have proven deep learning to be one of the most successful learning paradigms [43]. With this motivation, a deep neural network is chosen in this work to be the model with which the desired LIS interaction function is learned. In the following, the elements of this model are described.

Input Representation: A single input to the neural network model is defined as a stack of environment descriptors at K sub-carrier frequencies, i.e., the *sampled* channel vector $\widehat{\bar{\mathbf{h}}}$. This sets the dimensionality of a single input vector to $K\bar{M}$. A common practice in machine learning is the normalization of the input data. This guarantees a stable and meaningful learning process [44]. The normalization method of choice here is a simple per-dataset scaling; all samples are normalized by one constant value over the whole input data,

$$\widehat{\bar{\mathbf{h}}}_{\mathrm{norm}}(s) = \frac{\widehat{\bar{\mathbf{h}}}(s)}{\max_s \left\|\widehat{\bar{\mathbf{h}}}(s)\right\|_\infty}, \quad s = 1, \ldots, S. \tag{25}$$

Besides helping the learning process, this normalization choice preserves distance information encoded in the environment descriptors. This way the model learns to become more aware of the surroundings, which is the bedrock for proposing a machine-learning-powered LIS.

The last pre-processing step of input data is to convert them into real-valued vectors without losing the imaginary-part information. This is done by splitting each complex entry into real and imaginary values, doubling the dimensionality of each input vector. The main reason behind this step is the modern implementations of DL models, which mainly use real-valued computations.

Target Representation: The learning approach used in this work is supervised learning. This means the model is trained with input data that are accompanied by their so-called *target responses* [42]. They are the desired responses the model is expected to approximate when it encounters inputs like those in the input training data. Since the target of the training process is to learn a function mapping descriptors to reflection vectors, the model is designed to output a set of predictions on the achievable rates of every possible reflection beamforming vector in the codebook $|\mathcal{P}|$. Hence, the training targets are real-valued vectors, $\mathbf{r}(s), s = 1, \ldots, S$, with the desired rate for each possible reflection vector.

For the same training-efficiency reason expressed for the input representation, the labels are usually normalized. The normalization used in this work is per-sample scaling where every vector of rates $\mathbf{r}(s)$ is normalized using its maximum rate value $\max_n [\mathbf{r}(s)]_n$. The output of the normalization process is denoted by $\widehat{\mathbf{r}}(s)$. The choice of normalizing each vector independently guards the model against being biased towards some strong responses. In terms of our LIS application, it gives the receivers equal importance regardless of how close or far they are from the LIS.

Neural Network Architecture: The DL model is designed as a Multi-Layer Perceptron (MLP) network, sometimes referred to as a feedforward Fully Connected network. It is well-established that MLP networks are universal function approximators [45]. This motivates adopting an MLP network to capture the relation between the environment descriptors and the LIS interaction (reflection beamforming) vectors. As depicted in Fig. 5, the proposed MLP model consists of Q layers. The first $Q - 1$ of them alternate between fully connected and non-linearity layers and the last layer (output layer) is a fully connected layer. For the fully connected layers, each Layer q in the network has a stack of N_q neurons, each of which sees all the outputs of the previous layer. For the non-linearity layers, they all employ Rectified Linear Units (ReLUs) [42].

Training Loss Function: The model training process aims at minimizing a loss function that measures the quality of the model predictions. Given the objective of predicting the best reflection beamforming vector, $\boldsymbol{\psi}_{n^{\mathrm{DL}}}$, having the highest achievable rate estimate, $\max_n \widehat{R}_n$, the model is trained using a regression loss function. At every coherence block, the neural network is trained to make its output, $\widehat{\mathbf{r}}$, as close as possible to the desired output, the normalized achievable rates, $\bar{\mathbf{r}}$. Specifically, the training is guided through minimizing the loss function, $L(\boldsymbol{\theta})$, expressed as

$$L(\boldsymbol{\theta}) = \mathsf{MSE}(\bar{\mathbf{r}}, \widehat{\mathbf{r}}), \tag{26}$$

where $\boldsymbol{\theta}$ represents the set of all the neural network parameters and $\mathsf{MSE}(\bar{\mathbf{r}}, \widehat{\mathbf{r}})$ indicates the mean-squared-error between $\bar{\mathbf{r}}$ and $\widehat{\mathbf{r}}$.

VII. SIMULATION RESULTS

In this section, we evaluate the performance of both deep learning (DL) and compressive sensing (CS) based reflection beamforming approaches. The flow of this section is as follows. First, we describe the adopted experimental setup and datasets. Then, we compare the performance of the deep learning and compressive sensing solutions at both mmWave and sub-6 GHz bands. After that, we investigate the impact of different system and machine learning parameters on the performance of the deep learning solution.

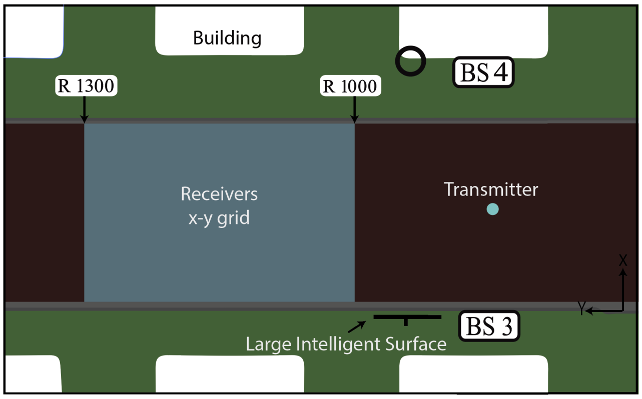


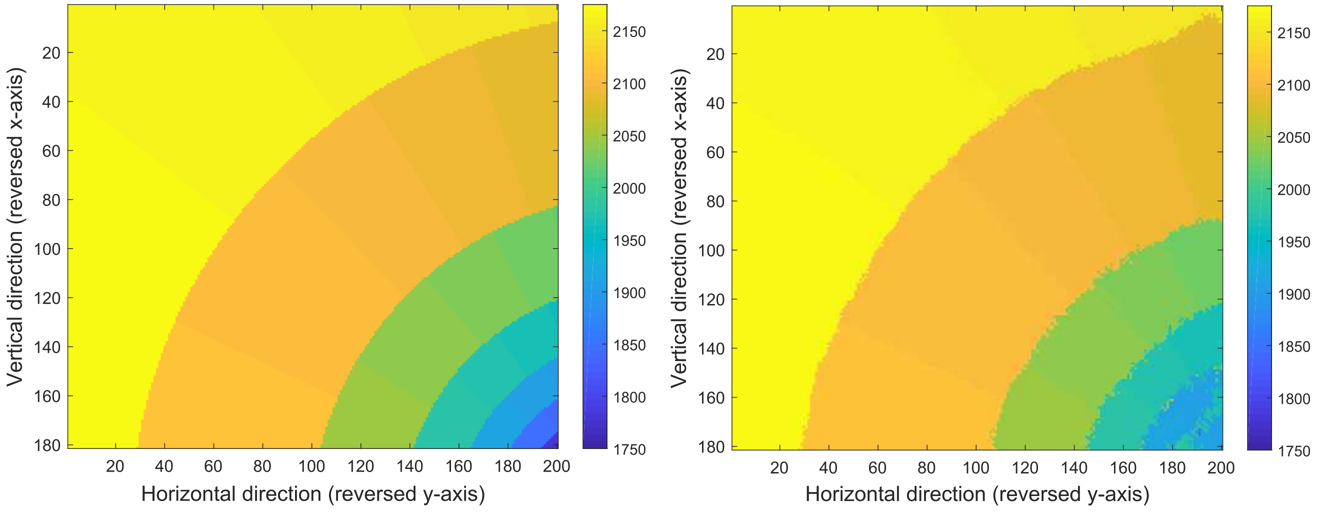
FIGURE 6. This figure illustrates the adopted ray-tracing scenario where an LIS is reflecting the signal received from one fixed transmitter to a receiver. The receiver is selected from an x-y grid of candidate locations. This ray-tracing scenario is generated using Remcom Wireless InSite [46], and is publicly available on the DeepMIMO dataset [26].

A. SIMULATION SETUP

Given the geometric channel model adopted in Section II and the nature of the reflection beamforming optimization problem, with its strong dependence on the environmental geometry, it is critical to evaluate the performance of the proposed solutions based on realistic channels. This motivates using channels generated by ray-tracing to capture the dependence on the key environmental factors such as the environment geometry and materials, the LIS and transmitter/receiver locations, the operating frequency among others. To do that, we adopted the DeepMIMO dataset, described in detail in [26], to generate the channels based on the outdoor ray-tracing scenario ‘O1’ [46], as will be discussed shortly. The DeepMIMO is a parameterized dataset published for deep learning applications in mmWave and massive MIMO systems. The machine learning simulations were executed using the Deep Learning Toolbox of MATLAB R2019a. The source code of this paper is available on [47]. Next, we explain in detail the key components of the simulation setup.

1) SYSTEM MODEL

Following the system model in Section II-A, we adopt an LIS-assisted communication system where one LIS aims to reflect the signal received from a transmitter to a receiver. The transmitter is assumed to be fixed in position while the receiver can take any random position in a specified x-y grid as illustrated in Fig. 6. We implemented this setup using the outdoor ray-tracing scenario ’O1’ of the DeepMIMO dataset that is publicly available at [26]. As shown in Fig. 6, we select BS 3 in the ’O1’ scenario to be the LIS and the user in row R850 and column 90 to be the fixed transmitter. The uniform x-y grid of candidate receiver locations includes 54300 points from row R1000 to R1300 in the ’O1’ scenario where every row consists of 181 points. Unless otherwise stated, the adopted LIS employs a UPA with 64 × 64 ($M = 4096$) antennas at the mmWave 28GHz setup and a UPA with 16 × 16 ($M = 256$) antennas at the 3.5GHz



(a) Original Codebook Beams

(b) Predicted Codebook Beams

FIGURE 7. This figure illustrates the *optimal* and *predicted* index map of the LIS reflection bemforming codebook. Each pixel represents the location of a candidate receiver on the x-y user grid under-study (shown in Fig. 6). The pixel color represents the index of the optimal/predicted reflection beamforming vector for the user at this location. In this scenario with 64 × 64 LIS, the optimum achievable rate, *R*⋆, averaged across all candidate locations, is 5.06 bps/Hz, while the achievable rate of the proposed deep learning based predicted beams is 4.74 bps/Hz.

setup. The active channel sensors described in Section IV are randomly selected from the M UPA antennas. The transmitter and receiver are assumed to have a single antenna each. The antenna elements have a gain of 3dBi and the transmit power is 35dBm. The antenna element spacing is set to half the wavelength, 0.5λ, where λ is the operating wavelength. The rest of the adopted DeepMIMO dataset parameters are summarized in Table 1.

2) CHANNEL GENERATION

The channels between the LIS and the transmitter/receiver, $\mathbf{h}_{\mathrm{T},k}$, $\mathbf{h}_{\mathrm{R},k}$, for all the candidate receiver locations in the x-y grid, are constructed using the DeepMIMO dataset generation code [26] with the parameters in Table 1. With these channels, and given the randomly selected active elements in the proposed LIS architecture, we construct the sampled channel vectors $\overline{\mathbf{h}}_{\mathrm{T},k}$, $\overline{\mathbf{h}}_{\mathrm{R},k}$. The noisy sampled channel vectors $\widehat{\overline{\mathbf{h}}}_{\mathrm{T},k}$, $\widehat{\overline{\mathbf{h}}}_{\mathrm{R},k}$ are then generated by adding noise vectors to $\overline{\mathbf{h}}_{\mathrm{T},k}$, $\overline{\mathbf{h}}_{\mathrm{R},k}$ according to (23), with the noise power calculated based on the bandwidth and other parameters in Table 1, and with receiver noise figure of 5dB. These noisy sampled channels are then used to design the LIS interaction (reflection beamforming) vectors following the proposed compressive sensing and deep learning approaches.

3) LIS INTERACTION (REFLECTION BEAMFORMING) CODEBOOK

We adopt a DFT codebook for the candidate LIS interaction vectors. More specifically, considering the UPA structure, we define the LIS interaction codebook as $DFT_{M_{\mathrm{H}}} \otimes DFT_{M_{\mathrm{V}}}$. The codebook $DFT_{M_{\mathrm{H}}} \in \mathbb{C}^{M_{\mathrm{H}} \times M_{\mathrm{H}}}$ is a DFT codebook for the azimuth (horizontal) dimension where the m_{H}th column, $m_{\mathrm{H}} = 1, 2, \dots, M_{\mathrm{H}}$, is defined as $[1, e^{-j\frac{2\pi}{M_{\mathrm{H}}}m_{\mathrm{H}}}, \dots, e^{-j(M_{\mathrm{H}}-1)\frac{2\pi}{M_{\mathrm{H}}}m_{\mathrm{H}}}]^T$. The codebook $DFT_{M_{\mathrm{V}}}$ is similarly defined for the elevation (vertical) dimension.

TABLE 1. The adopted DeepMIMO dataset parameters.

DeepMIMO Dataset Parameter	Value
Frequency band	3.5GHz or 28GHz
Active BSs	3
Number of BS Antennas	$(M_x, M_y, M_z) \in \{(1, 16, 16)\,;$ $(1, 32, 32)\,; (1, 64, 64)\}$
Active users (receivers)	From row R1000 to row R1300
Active user (transmitter)	row R850 column 90
System bandwidth	100MHz
Number of OFDM subcarriers	512
OFDM sampling factor	1
OFDM limit	64
Number of channel paths	$\{1, 2, 5, 10, 15\}$
Antenna spacing	0.5λ

As an example, Fig. 7 illustrates the *optimal* index map of the LIS reflection bemforming codebook at $f_c = 28$ GHz, $M = 64 \times 64$ antennas, and $L = 1$ channel path. The map orientation and directions are set according to the adopted ray-tracing scenario, previously shown in Fig. 6. The pixel position represents the candidate location of the receiver on the x-y grid under-study. The pixel color represents the index number of the optimal reflection beamforming vector for each candidate location, calculated according to (8), under the assumption of perfect *full* channel knowledge, $\mathbf{h}_{\mathrm{T},k}$ and $\mathbf{h}_{\mathrm{R},k}$, at the LIS. By comparison, Fig. 7(b) depicts the *predicted* index map of the LIS reflection bemforming codebook using the proposed Deep Learning (DL) based reflection beamforming with only $\overline{M} = 8$ active channel sensors.

4) DEEP LEARNING PARAMETERS

We adopt the deep learning model described in Section VI-C. To reduce the neural network complexity, however, we input the normalized sampled channels only at the first $K_{\mathrm{DL}} = 64$ subcarriers, $\widehat{\widehat{\mathbf{h}}}_k$, $k = 1, \ldots, K_{\mathrm{DL}}$ and $K_{\mathrm{DL}} \leq K$, which sets the length of the DL input vector to be $2\overline{M}K_{\mathrm{DL}}$. This is motivated by the fact that the channel is highly correlated in the frequency domain, a consequence of channel sparsity, especially in the mmWave range. The length of the DL output vector is $M = |\mathcal{P}|$, as described in Section VI-C. The neural network architecture consists of four fully connected layers. Unless otherwise mentioned, the number of hidden nodes of the four layers is $(M, 4M, 4M, M)$, where M is the number of LIS antennas. Given the size of the x-y grid of the candidate receiver locations in Fig. 6, the deep learning dataset has 54300 data points. We split this dataset into two sets, namely a training set and a testing set with 85% and 15% of the points, respectively. A dropout layer is added after every ReLU layer. Unless otherwise mentioned, we consider a batch size of 500 samples, a 50% dropout rate, an L_2 regularization factor of 10^{-4}, and 20 epochs of training. The learning rate starts from 0.1 and drops by 50% every 3 epochs.

5) COMPRESSIVE SENSING PARAMETERS

We consider the developed compressive sensing solution in Section V to recover the full LIS-transmitter/receiver channels and design the LIS reflection beamforming vectors. For approximating the solution of (20), we use OMP with a grid of size $N_{\mathrm{D}}^{\mathrm{Az}} N_{\mathrm{D}}^{\mathrm{El}}$ points, where $N_{\mathrm{D}}^{\mathrm{Az}} = 2M_{\mathrm{H}}$, $N_{\mathrm{D}}^{\mathrm{El}} = 2M_{\mathrm{V}}$.

Next, given this described setup, and adopting the novel LIS architecture in Fig. 2 with only $\overline{M}$ active channel sensors, we evaluate the performance of the developed compressive sensing and deep learning solutions.

B. ACHIEVABLE RATES WITH COMPRESSIVE SENSING AND DEEP LEARNING BASED LIS SYSTEMS

In this subsection, we evaluate the achievable rates of the proposed compressive sensing (CS) and deep learning (DL) based reflection beamforming solutions for LIS systems, as previously described in Section V-A and Section VI-B, respectively. These rates are compared to the genie-aided upper bound, $R^\star$, in (9) which assumes perfect knowledge of the *full* channel vectors, $\mathbf{h}_{\mathrm{T},k}$ and $\mathbf{h}_{\mathrm{R},k}$, at the LIS. The average achievable rate used for assessing the performance of these proposed solutions can be formulated as

$$R = \frac{1}{K}\sum_{k=1}^{K} \log_2\left(1 + \mathsf{SNR}\left|\left(\mathbf{h}_{\mathrm{T},k} \odot \mathbf{h}_{\mathrm{R},k}\right)^T \boldsymbol{\psi}\right|^2\right), \qquad (27)$$

where $\boldsymbol{\psi} \in \left\{\boldsymbol{\psi}_{n\mathrm{CS}}, \boldsymbol{\psi}_{n\mathrm{DL}}\right\}$ is the reflection beamforming vector chosen by the CS or DL based reflection beamforming solutions, respectively. To reduce the computational complexity of the performance evaluation, we compute the achievable rate summation over the first subcarrier instead of computing over all the $K = 512$ subcarriers.

In Fig. 8, we consider the simulation setup in Section VII-A at the mmWave 28GHz band with LIS employing a UPA of 64×64 antennas. The channels are constructed to include the strongest $L = 10$ channel paths. Fig. 8 shows that the proposed deep learning solution approaches the optimal upper bound with a very small number of active antennas. For example, with only $\overline{M} = 4$ active antennas (out of $M = 4096$ total antennas), the deep learning solution achieves almost 85% of the optimal achievable rate. This figure also illustrates the performance gain of the deep learning solution compared to the compressive sensing solution, especially when the number of active antennas is very small. Note that the two solutions approach the upper bound with $28 - 36$ active antennas, which represent less than 1% of the total number of antennas ($M = 4096$) in the LIS. This illustrates the high energy efficiency of the proposed LIS architecture and reflection beamforming solutions, as will be demonstrated in the upcoming subsection.

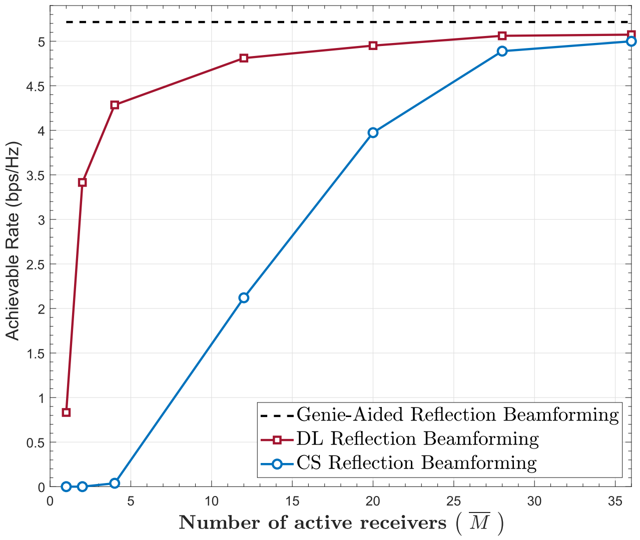


FIGURE 8. The achievable rate of both proposed CS and DL based reflection beamforming solutions are compared to the upper bound $R^\star$, for different numbers of active receivers, $\overline{M}$. The figure is generated at $f_c = 28$ GHz, $M = 64 \times 64$ antennas, and $L = 10$ paths.

Additionally, to evaluate the performance at sub-6 GHz systems, we plot the achievable rates of the proposed deep learning and compressive sensing solutions compared to the optimal rate $R^\star$ as illustrated in Fig. 9. This figure adopts the simulation setup in Section VII-A at a 3.5GHz band. The LIS is assumed to employ a UPA with 16×16 antennas, compared to 64×64 in the 28GHz band, given the path loss difference between the 3.5GHz and 28GHz bands. Each channel incorporates the strongest $L = 15$ paths, compared to $L = 10$ in the 28GHz band, motivated by the fact that the channels are less sparse in the sub-6 GHz systems compared to the mmWave systems.

Fig. 9 shows that the proposed deep learning and compressive sensing solutions are also promising for sub-6 GHz LIS systems. This is captured by the convergence to the upper bound with only 4 active elements in the deep learning case and around 18 elements in the compressive sensing case. This figure also illustrates the gain from employing the deep learning approach over the compressive sensing approach in the sub-6 GHz systems, where the channels are less sparse

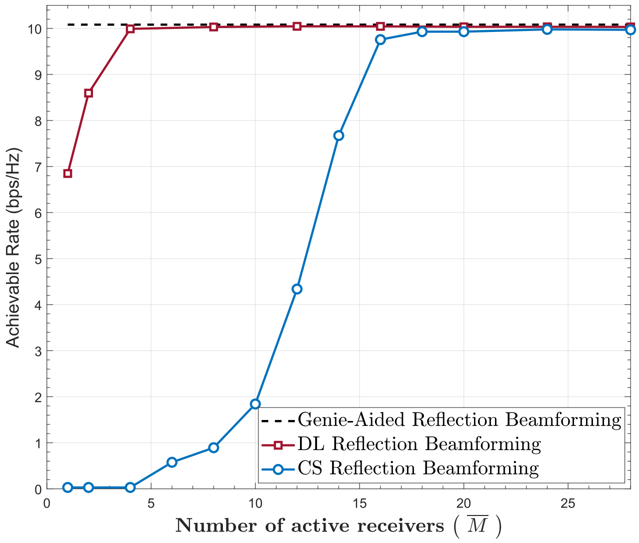


FIGURE 9. **The achievable rate of both proposed CS and DL based reflection beamforming solutions are compared to the upper bound $R^\star$, for different numbers of active receivers, $\overline{M}$. The figure is generated at $f_c = 3.5$ GHz, $M = 16 \times 16$ antennas, and $L = 15$ paths.**

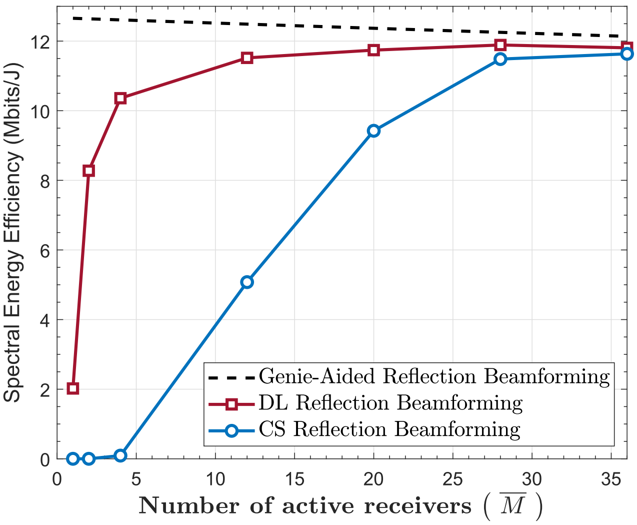


FIGURE 10. **The spectral energy efficiency of both proposed CS and DL based reflection beamforming solutions are compared to the upper bound $R^\star$, for different numbers of active receivers, $\overline{M}$. The figure is generated at $f_c = 28$ GHz, $M = 64 \times 64$ antennas, and $L = 10$ paths.**

than mmWave systems. This gain, however, has the cost of collecting a dataset to train the deep learning model, which is not required in the compressive sensing approach.

In Fig. 8 and Fig. 9, the number of active sensors ($\overline{M}$) is a design parameter that controls the size of the input of the neural network. As that number varies, the relation between the input vector and the output target vector also varies. This suggests that the neural network architecture needs to be designed carefully to capture that relation. In Fig. 8, the neural network architecture used for $\overline{M} = \{1, 2, 4, 12, 20\}$ has the following number of nodes $(M, 4M, 4M, M)$. This architecture changes to $(3M, 4M, 4M, M)$ to account for the change in the input-output relation as the number of sensors increases to $\overline{M} = \{28, 36\}$. For the results in Fig. 9, we have found that the architecture with $\left(4\overline{M}K_{\mathrm{DL}}, 16384, 16384, M\right)$ performs consistently well across all choices of $\overline{M}$.

C. ENERGY EFFICIENCY

In this subsection, we evaluate the energy efficiency of both proposed CS and DL based reflection beamforming approaches, compared to the upper bound on spectral energy efficiency, which assumes perfect *full* channel knowledge at the LIS. Starting with a formulation of a generic power consumption model for the proposed LIS architecture, we can then evaluate the energy efficiency, as formulated in [5], [48]. Consider the proposed LIS architecture shown in Fig. 2 and described in Section IV, with $\overline{M}$ active elements connected to the baseband through fully-digital architecture of b-bit ADCs. Let $P_{\mathrm{BB}}, P_{\mathrm{RFchain}}, P_{\mathrm{ADC}}, P_{\mathrm{PS}}, P_{\mathrm{LNA}}$ denote the power consumption in the baseband processor, RF chains, ADC, phase shifter (passive reflector), and LNA, respectively. The LIS power consumption model, P_{c}, can be generally formulated as [48]

$$P_{\mathrm{c}} = MP_{\mathrm{PS}} + \overline{M}\left(P_{\mathrm{LNA}} + P_{\mathrm{RFchain}} + 2P_{\mathrm{ADC}}\right) + P_{\mathrm{BB}}. \quad (28)$$

The power consumption of the ADC, P_{ADC}, can be further calculated as

$$P_{\mathrm{ADC}} = FOM_{\mathrm{W}} \times f_{\mathrm{S}} \times 2^b, \quad (29)$$

where b is the number of bits, f_{S} is the Nyquist sampling frequency, and FOM_{W} is the Walden's figure-of-merit for power efficiency ranking of the ADCs [49], [50]. Finally, the energy efficiency can be formulated as

$$\eta_{\mathrm{EE}} = \frac{R \times W}{P_{\mathrm{c}}} bits/Joule, \quad (30)$$

where W is the transmission bandwidth and R is the achievable rate.

Next, using (28)-(30), we evaluate the energy efficiency of both proposed CS and DL based reflection beamforming solutions compared to the upper bound, as depicted in Fig. 10. The various power consumption variables are assumed to be $P_{\mathrm{BB}} = 200$ mW, $P_{\mathrm{RF}} = 40$ mW, $P_{\mathrm{PS}} = 10$ mW, $P_{\mathrm{LNA}} = 20$ mW, and $W = 100$MHz [48]. Assume $b = 4$ bits according to the trade-off figure between the achievable rate and power consumption for fully-digital architecture, illustrated in [48]. Also, assume $FOM_{\mathrm{W}} = 46.1$ fJ/conversion-step at 100 MHz bandwidth according to the architecture in [50], [51]. In Fig. 10, The energy efficiency values across different numbers of active channel sensors are calculated from the achievable rate values of Fig. 8.

Fig. 10 shows the high energy efficiency gained from employing the proposed LIS architecture with few active channel sensors. This figure also illustrates that both proposed CS and DL based beamforming solutions can approach the upper bound with only $28 - 36$ active antennas. The DL solution achieves more energy efficiency gains when compared to the CS solution. Also, according to (28)-(30), since the upper bound is a monotonically decreasing bound when the number of active elements increases, it's safe to state that the optimal operating point for the DL based reflection beamforming approach is at $\overline{M} = 28$ active antenna elements, with an optimal energy efficiency of $\sim$ 12Mbits/J, for the described scenario only.

D. HOW MUCH TRAINING IS NEEDED FOR THE DEEP LEARNING MODEL?

The data samples in the deep learning dataset are captured when the receiver is randomly sampling the x-y grid. In Fig. 11, we study the performance of the developed deep learning approach for designing the LIS interaction vectors for different dataset sizes. This illustrates the improvement in the machine learning prediction quality as it sees more data samples. For Fig. 11, we adopt the simulation setup in Section VII-A with an LIS of 64 × 64 UPA and a number of active channel sensors $\overline{M} = 2, 4$, and 8. The setup considers a mmWave 28GHz scenario and the channels are constructed with only the strongest path, i.e., $L = 1$. Fig. 11 shows that with only 8 active antennas, the proposed deep learning solution can achieve almost 90% of the optimal rate in (9) when the model is trained on 14 thousand data points (out of the 54300 points) in the x-y grid. Further, this figure highlights the performance gain of the deep learning approach compared to the compressive sensing solution. This gain increases with larger dataset sizes as the compressive sensing solution does not leverage the prior channel estimation/LIS interaction observations and its performance does not depend on the size of the dataset.

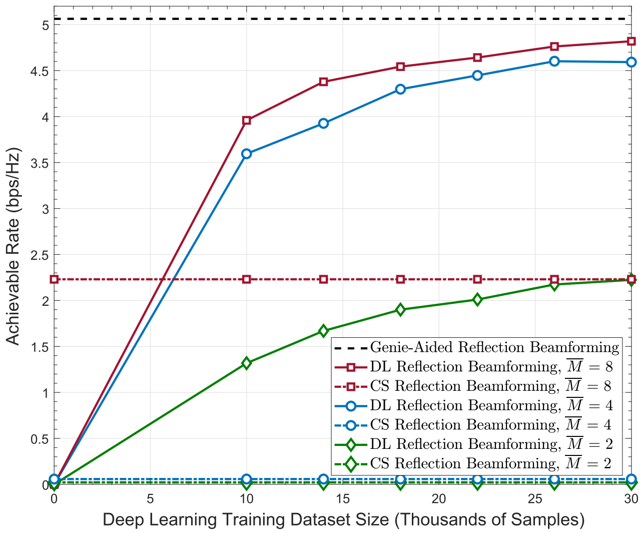


FIGURE 11. The achievable rate of the proposed DL based reflection beamforming approach is compared to the upper bound $R^\star$ and the CS beamforming approach, for different numbers of active receivers, $\overline{M}$. The adopted setup considers an LIS with 64 × 64 UPA, at 28GHz with $L = 1$ channel path. This figure highlights the promising gain of the proposed deep learning solution that approaches the upper bound using only 8 active elements (less than 1% of the total number of antennas). This performance requires collecting a dataset of around 20-25 thousand data points (user locations).

E. IMPACT OF IMPORTANT SYSTEM AND CHANNEL PARAMETERS

In this subsection, we evaluate the impact of the key system and channel parameters on the performance of the developed deep learning solution.

1) NUMBER OF LIS ANTENNAS

Fig. 12 examines the achievable rate performance of the developed solutions for designing the LIS interaction vectors when the LIS employs either a 32 × 32 or a 64 × 64 UPA. This figure adopts the same mmWave scenario considered in Fig. 11. As illustrated, with only $\overline{M} = 8$ active receivers, the proposed deep learning solution approaches the optimal rate in (9) that assumes perfect channel knowledge for different LIS sizes. This shows the potential of the proposed LIS architecture and deep learning solution in enabling large intelligent surfaces with large numbers of antennas. **Note that the proposed solution does not require any beam training overhead (as it relies on the deep learning prediction of the best beam) and needs only 8 active receivers to realize this near-optimal performance in Fig. 12**.

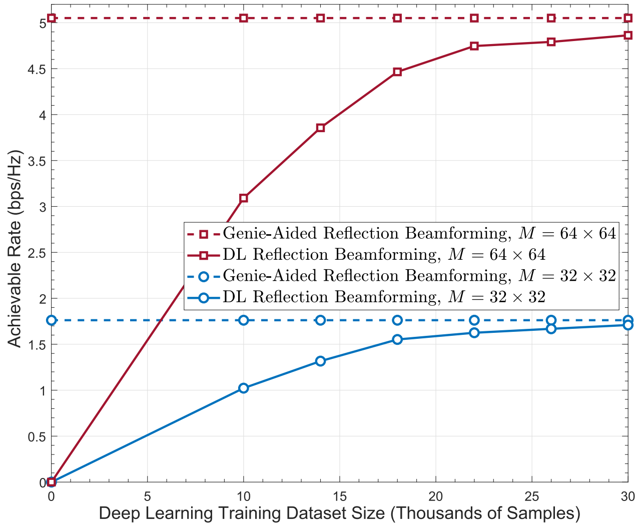


FIGURE 12. The achievable rate of the proposed DL based reflection beamforming approach is compared to the upper bound $R^\star$ for different sizes of intelligent surfaces, namely with LIS of 32 × 32 and 64 × 64 UPAs. The number of active elements (channel sensors) equals $\overline{M} = 8$. This figure is generated at 28GHz with $L = 1$ channel path.

2) TRANSMIT POWER

In Fig. 13, we study the impact of the transmit power (and *receive* SNR) on the achievable rate performance of the developed deep learning solution. This is important in order to evaluate the robustness of the learning and prediction quality, as we input the noisy sampled channel vectors to the deep learning model. In Fig. 13, we plot the achievable rates of the proposed deep learning solution as well as the upper bound in (9) for three values of the transmit power, $P_\mathrm{T} = -5, 0, 5$ dBW. These transmit powers map to *receive* SNR values of $-3.8, 6.2, 16.2$ dB, respectively, including the LIS beamforming gain of the 4096 antennas. The rest of the setup parameters are the same as those adopted in Fig. 11.

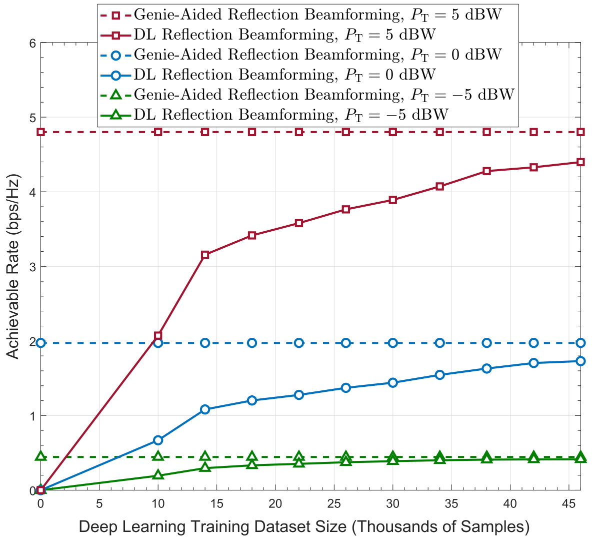


FIGURE 13. The achievable rate of the proposed deep learning based reflection beamforming approach is compared to the upper bound $R^\star$, for different values of user transmit power, P_T. The figure is generated for an LIS with $M = 64 \times 64$ UPA and $\overline{M} = 8$ active elements, at 28GHz with $L = 1$ channel path. This figure shows that the proposed DL solution is capable of learning and approaching the optimal achievable rate even with a relatively small transmit power.

Fig. 13 illustrates that the proposed deep learning solution can perform well even with relatively small transmit powers and low SNR regimes.

3) NUMBER OF CHANNEL PATHS

In Fig. 14, we investigate the impact of the number of channel paths on the performance of the developed deep learning solution. In other words, we examine the robustness of the proposed deep learning model with multi-path channels. For this figure, we adopt the same simulation setup of Fig. 11 with an LIS employing 64 × 64 UPA. The channels are constructed considering the strongest $L = 1, 2,$ or 5 channel paths. As shown in Fig. 14, with the increase in the number of channel paths, the achievable rate by the proposed deep learning solution converges slower to the upper bound. This shows that the proposed deep learning model can learn from multi-path channels if a large enough dataset is available.

F. REFINING THE DEEP LEARNING PREDICTION

In Fig. 8-Fig. 14, we considered the proposed deep learning solution where the deep learning model uses the sampled channel vectors to predict the best beam and this beam is directly used to reflect the transmitted data. Relying completely on the deep learning model to determine the reflection beamforming vector has the clear advantage of eliminating the beam training overhead and enabling highly mobile applications. The achievable rates using this approach, however, may be sensitive to small changes in the environment. A candidate approach for enhancing the reliability of the system is to use the machine learning model to predict the most promising k_B beams. These beams are then refined through beam training with the receiver to select the final beam reflection vector. Note that the most promising k_B beams refer to the k_B beams with the highest predicted rates from the deep learning model. To study the performance using this approach, we plot the achievable rate of the deep learning solution in Fig. 15, for different values of k_B. As this figure shows, refining the most promising k_B yields higher achievable rates compared to the case when the LIS relies completely on the deep learning model to predict the best beam, i.e., with k_B. The gain in Fig. 15 is expected to increase with a more time-varying and dynamic environment, which is an interesting extension in future work.

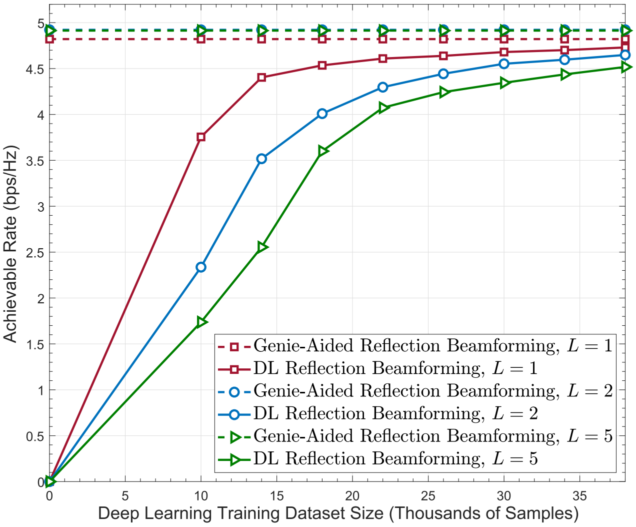


FIGURE 14. The achievable rate of the proposed DL based reflection beamforming approach is compared to the upper bound $R^\star$, for different numbers of channel paths, L. The figure is generated for an LIS with 64 × 64 UPA and $\overline{M} = 4$ active elements, at 28GHz. As the number of channel paths increases, the achievable rate achieved by the proposed DL solution converges slower to the upper bound. Hence, using more training data can help learn multi-path signatures.

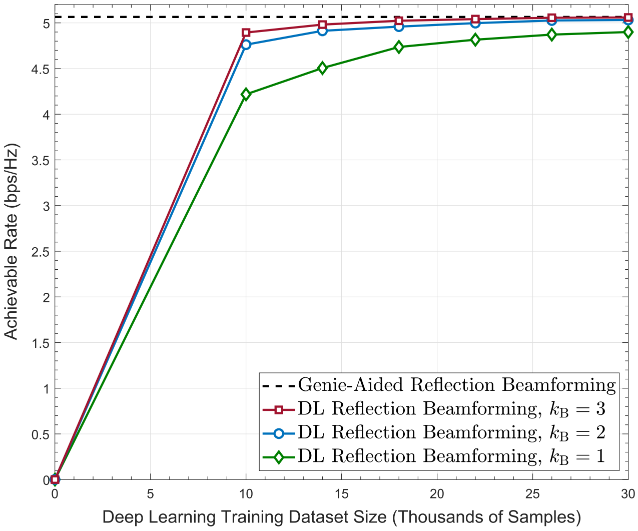


FIGURE 15. The achievable rate of the proposed DL based reflection beamforming approach is compared to the upper bound $R^\star$. The simulation considers an LIS with 64 × 64 UPA and $\overline{M} = 4$ active channel sensors, at 28GHz with $L = 1$ channel path. The figure illustrates the achievable rate gain when the beams selected by the deep learning model are further refined through beam training over k_B beams.

VIII. CONCLUSION

In this paper, we considered LIS-assisted wireless communication systems and developed efficient solutions that design the LIS interaction (reflection) matrices with negligible training overhead. We first introduced a novel LIS architecture where only a small number of the LIS elements are active (connected to the baseband). Then, we developed two solutions that design the LIS reflection matrices for this new architecture with almost no training overhead. The first solution leverages compressive sensing tools to construct the channels at all the antenna elements from the *sampled* channels seen only at the active elements. The second approach exploits deep learning tools to learn how to predict the optimal LIS reflection matrices directly from the sampled channel knowledge, which represents what we call *environment descriptors*. Extensive simulation results based on accurate ray-tracing showed that the two proposed solutions can achieve near-optimal data rates with negligible training overhead and with a few active elements. Compared to the compressive sensing solution, the deep learning approach requires a smaller number of active elements to approach the optimal rate, thanks to leveraging its prior observations. Further, the deep learning approach does not require any knowledge of the LIS array geometry and does not assume sparse channels. To achieve these gains, however, the deep learning model needs to collect enough dataset, which is not needed in the compressive sensing solution. There are several interesting extensions for the compressive sensing solution, including the optimization of the sparse distribution of the active sensors leveraging tools from nested and coprime arrays. For future work, it is interesting to investigate solutions for developing fully standalone LIS architectures, where the LIS is not controlled by the infrastructure but rather operating on its own while interacting with the environment.

ACKNOWLEDGMENT

A conference version of this paper has been published in [1].

REFERENCES

[1] 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.

[2] A. Puglielli, N. Narevsky, P. Lu, T. Courtade, G. Wright, B. Nikolic, and E. Alon, "A scalable massive MIMO array architecture based on common modules," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1310–1315.

[3] S. Hu, F. Rusek, and O. Edfors, "Beyond massive MIMO: The potential of data transmission with large intelligent surfaces," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2746–2758, May 2018.

[4] S. V. Hum and J. Perruisseau-Carrier, "Reconfigurable reflectarrays and array lenses for dynamic antenna beam control: A review," *IEEE Trans. Antennas Propag.*, vol. 62, no. 1, pp. 183–198, Jan. 2014.

[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.

[6] X. Tan, Z. Sun, D. Koutsonikolas, and J. M. Jornet, "Enabling indoor mobile millimeter-wave networks based on smart reflect-arrays," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 270–278.

[7] Y. Zhang, J. Zhang, Y. Wang, Z. Yu, and B. Zhang, "A 4-bit programmable metamaterial based on VO2 mediums," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2018, pp. 984–986.

[8] 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.

[9] Q.-U.-U. Nadeem, A. Kammoun, A. Chaaban, M. Debbah, and M.-S. Alouini, "Asymptotic max-min SINR analysis of reconfigurable intelligent surface assisted MISO systems," 2019, *arXiv:1903.08127*. [Online]. Available: http://arxiv.org/abs/1903.08127

[10] E. Basar, "Reconfigurable intelligent surface-based index modulation: A new beyond MIMO paradigm for 6G," 2019, *arXiv:1904.06704*. [Online]. Available: http://arxiv.org/abs/1904.06704

[11] M. Jung, W. Saad, Y. Jang, G. Kong, and S. Choi, "Performance analysis of large intelligent surfaces (LISs): Asymptotic data rate and channel hardening effects," 2018, *arXiv:1810.05667*. [Online]. Available: http://arxiv.org/abs/1810.05667

[12] A. Faisal, H. Sarieddeen, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "Ultra-massive MIMO systems at terahertz bands: Prospects and challenges," 2019, *arXiv:1902.11090*. [Online]. Available: http://arxiv.org/abs/1902.11090

[13] E. De Carvalho, A. Ali, A. Amiri, M. Angjelichinoski, and R. W. Heath Jr, "Non-stationarities in extra-large scale massive MIMO," 2019, *arXiv:1903.03085*. [Online]. Available: http://arxiv.org/abs/1903.03085

[14] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality—What is next? Five promising research directions for antenna arrays," 2019, *arXiv:1902.07678*. [Online]. Available: http://arxiv.org/abs/1902.07678

[15] L. Sanguinetti, E. Björnson, and J. Hoydis, "Towards massive MIMO 2.0: Understanding spatial correlation, interference suppression, and pilot contamination," 2019, *arXiv:1904.03406*. [Online]. Available: http://arxiv.org/abs/1904.03406

[16] X. Mu, Y. Liu, L. Guo, J. Lin, and N. Al-Dhahir, "Exploiting intelligent reflecting surfaces in NOMA networks: Joint beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6884–6898, Oct. 2020.

[17] H. Wymeersch and B. Denis, "Beyond 5G wireless localization with reconfigurable intelligent surfaces," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[18] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, 2018.

[19] A. Alkhateeb, I. Beltagy, and S. Alex, "Machine learning for reliable mmWave systems: Blockage prediction and proactive handoff," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 1055–1059.

[20] X. Li, A. Alkhateeb, and C. Tepedelenlioglu, "Generative adversarial estimation of channel covariance in vehicular millimeter wave systems," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 1572–1576.

[21] J. Wang, Z. Lan, C.-w. Pyo, T. Baykas, C.-s. Sum, M. A. Rahman, J. Gao, R. Funada, F. Kojima, H. Harada, and S. Kato, "Beam codebook based beamforming protocol for multi-gbps millimeter-wave WPAN systems," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1390–1399, Oct. 2009.

[22] 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, Oct. 2013.

[23] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic, and R. W. Heath, "MIMO precoding and combining solutions for millimeter-wave systems," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 122–131, Dec. 2014.

[24] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.

[25] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[26] 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. [Online]. Available: https://www.deepmimo.net/

[27] 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.

[28] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao, and J. I. Tamir, "Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications," *IEEE Trans. Antennas Propag.*, vol. 61, no. 4, pp. 1850–1859, Apr. 2013.

[29] T. S. Rappaport, R. W. Heath Jr, R. C. Daniels, and J. N. Murdock, *Millimeter Wave Wireless Communications*. London, U.K.: Pearson, 2014.

[30] M. K. Samimi and T. S. Rappaport, "Ultra-wideband statistical channel model for non line of sight millimeter-wave urban channels," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 3483–3489.

[31] S. Foo, "Liquid-crystal reconfigurable metasurface reflectors," in *Proc. IEEE Int. Symp. Antennas Propag. USNC/URSI Nat. Radio Sci. Meeting*, Jul. 2017, pp. 2069–2070.

[32] 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.

[33] R. Mendez-Rial, C. Rusu, N. Gonzalez-Prelcic, A. Alkhateeb, and R. W. Heath, "Hybrid MIMO architectures for millimeter wave communications: Phase shifters or switches?" *IEEE Access*, vol. 4, pp. 247–267, 2016.

[34] J. Lee, G.-T. Gil, and Y. H. Lee, "Exploiting spatial sparsity for estimating channels of hybrid MIMO systems in millimeter wave communications," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 3326–3331.

[35] T. Cai, W. Liu, and X. Luo, "A constrained ℓ_1 minimization approach to sparse precision matrix estimation," *J. Amer. Stat. Assoc.*, vol. 106, no. 494, pp. 594–607, 2011.

[36] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[37] P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4167–4181, Aug. 2010.

[38] P. P. Vaidyanathan and P. Pal, "Sparse sensing with co-prime samplers and arrays," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 573–586, Feb. 2011.

[39] Z. Tan, Y. C. Eldar, and A. Nehorai, "Direction of arrival estimation using co-prime arrays: A super resolution viewpoint," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5565–5576, Nov. 2014.

[40] C. Rusu, N. González-Prelcic, and R. W. Heath, "Algorithms for the construction of incoherent frames under various design constraints," *Signal Process.*, vol. 152, pp. 363–372, Nov. 2018.

[41] P. Xia, S. Zhou, and G. B. Giannakis, "Achieving the welch bound with difference sets," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1900–1907, May 2005.

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/

[43] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, Jun. 2014.

[44] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer-Verlag, 2012, pp. 9–48.

[45] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[46] Remcom. *Wireless InSite*. Accessed: Mar. 4, 2021. [Online]. Available: http://www.remcom.com/wireless-insite

[47] *LIS Deep Learning Simulation Code*. Accessed: Mar. 4, 2021. [Online]. Available: https://github.com/Abdelrahman-Taha/LIS-DeepLearning

[48] J. Mo, A. Alkhateeb, S. Abu-Surra, and R. W. Heath, "Hybrid architectures with few-bit ADC receivers: Achievable rates and energy-rate tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2274–2287, Apr. 2017.

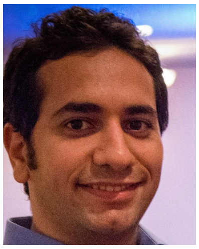
[49] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[50] B. Murmann. (2019). *ADC Performance Survey 1997–2019*. [Online]. Available: https://web.stanford.edu/~murmann/adcsurvey.html

[51] Y. Lim and M. P. Flynn, "A 100 MS/s, 10.5 bit, 2.46 mW comparator-less pipeline ADC using self-biased ring amplifiers," *IEEE J. Solid-State Circuits*, vol. 50, no. 10, pp. 2331–2341, Oct. 2015.

ABDELRAHMAN TAHA received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, in 2013 and 2018, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical, Computer and Energy Engineering, Arizona State University (ASU), with a specialization in signal processing systems and communications. His research interests include massive MIMO systems and signal processing for wireless communications. He received the Egyptian Government Excellence Award in 2013, for being ranked first among all engineering programs, B.S.E.E. Valedictorian, Faculty of Engineering, Ain Shams University.

MUHAMMAD ALRABEIAH received the Bachelor of Engineering (B.Eng.) degree in electrical engineering from King Saud University (KSU), Riyadh, Saudi Arabia, in 2010, and the Master of Applied Sciences (M.A.Sc.) degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 2015. He is currently pursuing the Ph.D. degree with the School of Electrical, Computer, and Energy Engineering (ECEE), Arizona State University (ASU), Tempe, AZ, USA. He holds a Lecturer position with the Electrical Engineering Department (EE Dept), KSU. His research interests include machine learning, computer vision, and wireless communications and sensing. He was a recipient of KSU's Scholarship for Graduate-Studies Abroad.



AHMED ALKHATEEB received the B.S. degree (Hons.) and the M.S. degree in electrical engineering from Cairo University, Egypt, in 2008 and 2012, respectively, and the Ph.D. degree in electrical engineering from The University of Texas at Austin, USA, in August 2016. From September 2016 to December 2017, he was a Wireless Communications Researcher with the Connectivity Laboratory, Facebook, Menlo Park, CA, USA. He joined Arizona State University (ASU), in 2018, where he is currently an Assistant Professor with the School of Electrical, Computer and Energy Engineering. He has held Research and Development internships with FutureWei Technologies (Huawei), Chicago, IL, USA, and Samsung Research America (SRA), Dallas, TX, USA. His research interests include broad areas of wireless communications, communication theory, signal processing, machine learning, and applied mathematics. He was a recipient of the 2012 MCD Fellowship from The University of Texas at Austin and the 2016 IEEE Signal Processing Society Young Author Best Paper Award for his work on hybrid precoding and channel estimation in millimeter-wave communication systems.