IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 March 2024, accepted 11 May 2024, date of publication 15 May 2024, date of current version 23 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3401454

RESEARCH ARTICLE

Capacity Optimization of Large Intelligent Surface With Hardware Impairment Based on Meta-Deep Learning

YIFAN MAO, XIAOYU XIAO, AND ZHIRUN HU, (Member, IEEE)
Department of Electrical and Electronic Engineering, The University of Manchester, M13 9PL Manchester, U.K.
Corresponding author: Zhirun Hu (z.hu@manchester.ac.uk)

ABSTRACT This work proposes a sub-optimal method based on a two-layer structured meta-deep reinforcement learning (MDRL) approach to address the hardware impairment (HWI) optimization issue in large intelligent surface (LIS) systems. This method, designed for distributed LIS systems with reflection matrices, effectively enhances the system capacity and performance despite HWIs. Building upon existing techniques of dividing large-area LIS systems into multiple small-area subsystems, the simulated results demonstrate that sub-optimal LIS performance can be achieved with fewer samples in diverse dynamic wireless environments. This innovative approach enhances the adaptability of distributed LIS systems and offers an effective HWI management strategy, paving the way for future LIS system optimization.

INDEX TERMS Large intelligent surface, distributed system, hardware impairment, reflection matrix design, fewer samples, meta-deep reinforcement learning.

I. INTRODUCTION

The Multiple Input Multiple Output (MIMO) system is currently undergoing extensive research as a key catalyst for efficient access in wireless communication systems and the Internet of Things (IoT), where billions of devices are expected to communicate [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. To achieve enhanced throughput and expansive cell coverage while conserving power through high-gain arrays, the adoption of Large Intelligent Surface (LIS) technologies has emerged as a vital development of the traditional MIMO framework. Serving as a cost-effective medium, LIS manipulates electromagnetic waves to facilitate extraordinary energy concentration in three dimensions, potentially enabling vital applications including wireless charging, high-precision remote sensing, and the transmission of significant data volumes [2], [3], [4], [5].

A typical LIS system consists of a planar array with numerous reflective elements. These elements interact with electromagnetic waves, acting as phase shifters to control the direction of reflected signals. Through this mechanism, they not only alter the propagation paths of signals but also directly impact the quality of communication [6], [7]. Consequently, optimizing the LIS reflection matrix is crucial for enhancing communication quality and increasing data throughput, thereby improving overall system performance. There are two primary approaches to LIS reflection matrix design and optimization: The first approach estimates the LIS-assisted channel at the transmitter/receiver through one-by-one training, demanding extensive computational resources due to the complex interactions of numerous elements during signal reflection [8], [9].The second approach involves choosing the matrix from quantized codebooks based on online reflection estimation. This method eliminates the need for explicit channel estimation, but potentially results in diminished performance [10], [11]. Most recent studies indicate that employing the on-off scheme in channel estimation reduces training overhead. Additionally, a three-stage technique for cascaded channel estimation improves efficiency. However, these approaches often underutilize existing knowledge of the shared channel, highlighting opportunities for future advancements [12].

Deep learning has demonstrated promising capabilities in formulating reflection matrices for LIS systems, yet it

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Zaman Chowdhury.

encounters significant challenges when managing extensive training datasets, particularly with respect to high training costs and hardware limitations [11], [12], [13], [14]. These challenges become increasingly noticeable in the context of big data, highlighting the need for more research and optimization. Additionally, the dynamic nature of wireless environments requires frequent retraining, affecting their operational efficiency and adaptability. Meanwhile, optimizing the number and effectiveness of training samples to boost system efficiency and flexibility is a critical research priority. Deep Reinforcement Learning (DRL) methods have demonstrated considerable promise in overcoming challenges associated with deep learning applications in LIS systems. For instance, a DRL-based method for dynamic reflection matrix design in LIS has been introduced [15]. Concurrently, a LIS-assisted multiple-input-single-output (MISO) systems' optimal design has been investigated [16], suggesting a deep transfer learning (DTL)-based low-complexity algorithm for co-designing transmission beams and phase-shift matrices. This approach employs migration learning to reduce base station transmitting power, thereby optimizing the reflected beam and decreasing reliance on labeled data and tackles hardware constraints by incorporating discrete phase shift constraints [16].

Despite DRL methods' effectiveness in optimizing LIS systems, they face limitations due to high computational resource demands and reliance on quality training data in practical applications [15], [16], [17]. This becomes particularly challenging in practical applications, especially when considering hardware impairments (HWIs). The majority of existing LIS research assumes ideal hardware conditions, often overlooking the impact of HWIs on system performance. However, HWIs in LIS systems, including RF interference, quantization errors, amplifier nonlinearities, and time-frequency synchronization errors, present significant challenges [18], [19], [20], [21], [22]. The large number of cells in LIS results in capacity and utility degradation due to HWIs, particularly when these impairments are severe. To address this, distributed LIS strategies have been proposed [20]. They involve dividing a large-area LIS system into smaller subsystems, thus mitigating HWIs' impacts and effectively increasing system capacity. While the distributed LIS strategy enhances capacity, it also raises the synchronization and synthesis costs for each subsystem. Recently active LIS have been found being more effective in scenarios with a smaller number of elements, whereas passive LIS more suitable for environments with a larger element count [23].

However, delays in convergence can impact the synchronization and synthesis of the receiving signals during the adaptation to a specific sub-LIS environment. Conversely, calculating the equivalent noise density and utility for predicting reflected beamforming vectors in a distributed LIS involves extensive sampling and complex computations, making the reduction of training samples critical [11], [13], [15], [24]. To further optimize system performance, this work proposes a novel deep reinforcement learning approach tailored to address reflection matrix design in distributed LIS systems with HWIs. Recent studies identify the significant potential for cooperative passive beamforming across inter-LIS channels in communication systems supported by multiple LISs [25]. This study employs meta-learning to comprehend the multi-LIS environments, focusing particularly on the dynamics of inter-sub-LIS channels. With pre-acquired knowledge of the environment, deep reinforcement learning is then applied to train the sub-LIS channel, effectively reducing both the learning duration and the number of training samples required.

This work offers a detailed exploration of optimizing reflection matrices in distributed LIS systems under HWI conditions. Specifically, it introduces a novel DRL-based method to enhance the performance of the reflected beamforming matrix in a distributed LIS with HWIs. The main contributions of this paper are as follows:

- The study addresses the distributed LIS matrix design optimization with the constraint of HWIs, proposing a structured method to enhance the rate performance and efficiency of reflected beamforming matrices.
- A sub-optimization method utilizing a two-layer meta-deep reinforcement learning (MDRL) architecture is developed, substantially reducing training sample requirements and nearing the distributed LIS system's maximum capacity with HWIs.
- The proposed method's effectiveness is validated, achieving high performance with fewer samples across various HWI scenarios, demonstrating its adaptability to diverse dynamic wireless environments.

The paper is organized as follows: Section II introduces the distributed LIS communication system with HWI's and discusses the optimal reflection matrix design. Section III presents an MDRL-based algorithm designed to minimize training sample requirements for distributed LIS, leveraging the interrelations of various sub-LIS systems. Section IV demonstrates the performance of the two-layer MDRL algorithm, showcasing simulation results across diverse HWI scenarios. Section V provides a comprehensive summary.

II. LIS SYSTEM MODEL

Research has demonstrated that the design of the LIS reflection matrix and its interaction with wireless environments can be modeled as a Markov Decision Process (MDP) [16], [17]. This study focuses on designing the interaction process between individual sub-LIS in a distributed system and the environments, as illustrated in Fig.1. It is assumed that the direct link between the transmitter and receiver is obstructed, with either the transmitter or receiver equipped with a single antenna.

A. SYSTEM MODEL FOR HWI OF A DISTRIBUTED LIS

Let the power at the transmitting point be P, and the signal wavelength λ, according to [20], the uplink signal received at

the LIS surface point is

$$m(x,y) = \sqrt{P}h_t(x,y)g + \sqrt{P}u(r)h_t(x,y)g + n(x,y), \quad (1)$$

where g represents the transmitted signal. $h_t(x,y)$ denotes the uplink channel gain from the transmitter point $t(x_0,y_0,z_0)$ to the LIS' position point (x,y). Here, $n(x,y)$ signifies the system's Gaussian noise,with a power spectral density of N_0. And $u(r)$ represents the influence of HWIs on the received signal, which follows a Gaussian process.

Given the environmental correlation where each sub-LIS is deployed, each exhibits a Markovian dependency [26], [27]. This implies that the actions of current and previous sub-LISs significantly influence the action strategies of subsequent sub-LISs. The dependency within the distributed LIS system can thus be modeled using a Markov model, rendering its optimization process can be seen as a MDP [26].

B. ANALYSIS OF MDP MODEL FOR ELEMENT FOR LIS

Initially, the interaction between the LIS and its environments is modeled as a MDP to maximize the user rate in LIS communication. The optimization variable in focus is the LIS' reflection matrix, with our design based on reinforcement learning techniques. In Fig.1, the LIS is portrayed as an agent autonomously interacting with the environments, striving to achieve the objectives of the reflection matrix design. The LIS wireless environments, comprising the transmitter, receiver, and wireless channel, represents the agent's interaction domain. The agent observes the environments' current state s and adopts a random action corresponding to state s. The environments, in response to action A, generates a reward R and transitions to a new state s_0 for the agent.The LIS interacts with the wireless environments of the communication system, and the agent gradually learns the optimal reflection matrix strategy π_0 to maximise the expected reward.

To reduce the complexity of the system, the entire LIS surface is designed to include two types of sensing elements [11]: 1) a large number of passive reflecting elements; 2) a few active channel sensing elements. In this study, active channel sensing elements are employed to gather environmental descriptor parameters. Despite their limited number, the active channel elements are highly representative of the entire LIS, owing to the smaller area of the distributed sub-LISs [3], [28], [29].

State Space:

1) ACTIVE CASCADED CHANNEL

$$h_t(x,y) = \frac{1}{2}\sqrt{\frac{z_0}{\pi}}\eta_t^{-\frac{3}{4}} exp(-\frac{2\pi j\sqrt{\eta_t}}{\lambda}), \quad (2)$$

$$\eta_t = z_0^2 + (y-y_0)^2 + (x-x_0)^2, \quad (3)$$

where η_t denotes the distance between the transmitting point and the LIS' position point. The channel gain $h_r(x,y)$, from the LIS position point (x,y) to the receiver point $r_j(x_j,y_j,z_j)$ can be similarly defined [1]. Additionally, η_j represents the distance between the receiver point and the LIS position point.

$$h_r(x,y) = \frac{1}{2}\sqrt{\frac{z_j}{\pi}}\eta_j^{-\frac{3}{4}} exp(-\frac{2\pi j\sqrt{\eta_j}}{\lambda}), \quad (4)$$

$$\eta_j = z_j^2 + (y-y_j)^2 + (x-x_j)^2, \quad (5)$$

The received signal at the receiver can be expressed as,

$$\begin{aligned} y(x,y) &= \sqrt{P}h_r^T(x,y)\Psi h_t(x,y)g + w(z,y), \\ &= \sqrt{P}(h_r(x,y)\odot h_t(x,y))^T\psi g + w(x,y), \end{aligned} \quad (6)$$

where Ψ denotes the LIS reflection beamforming vector, with its diagonal matrix representation $\Psi = diag(\psi)$, and $w(x,y)$ represents the HWI noise. The cascaded channel can be derived as follows,

$$h(x,y) = h_r(x,y)\odot h_t(x,y), \quad (7)$$

Consider a distributed LIS comprising M sub-LISs, and let's define the active cascade channel for the m^{th} sub-LIS as,

$$h^m(x,y) = h_r^m(x,y)\odot h_t^m(x,y), \quad (8)$$

2) HWI NOISE

The HWI noise is defined as [20],

$$w(x,y) = \sqrt{P}u(r)h(x,y)g + n(x,y), \quad (9)$$

where $u(r)$ is a zero-mean Gaussian random variable which has a non-negative variance as a function of $f(r)$ [20],

$$f(r) = \alpha * r^{2\beta}, \quad (10)$$

$$r = \sqrt{x^2+y^2}, \quad (11)$$

where α represents the overall HWI impact factor, β indicates the LIS area size impact factor, and r denotes the distance from the LIS position point to its centre. The surface area of the square LIS is defined as S, with its side length being l. The power spectral density of the HWI noise is calculated as [20],

$$\tilde{N} \approx N_0 + \frac{4^{\beta-1}P\alpha l^{2\beta}}{(\beta+1)z_0^2\pi^{\beta+1}}, \quad (12)$$

Consider a distributed LIS composed of M sub-LISs.For the m^{th} sub-LIS, the power spectral density of the HWI noise can be described as,

$$\tilde{N}^m \triangleq N_0^m + \frac{4^{\beta-1}P\alpha\left(\frac{l}{M}\right)^{2\beta}}{(\beta+1)z_0^2\pi^{\beta+1}}, \quad (13)$$

where N_0^m represents the power spectral density, characterized by a Gaussian distribution.

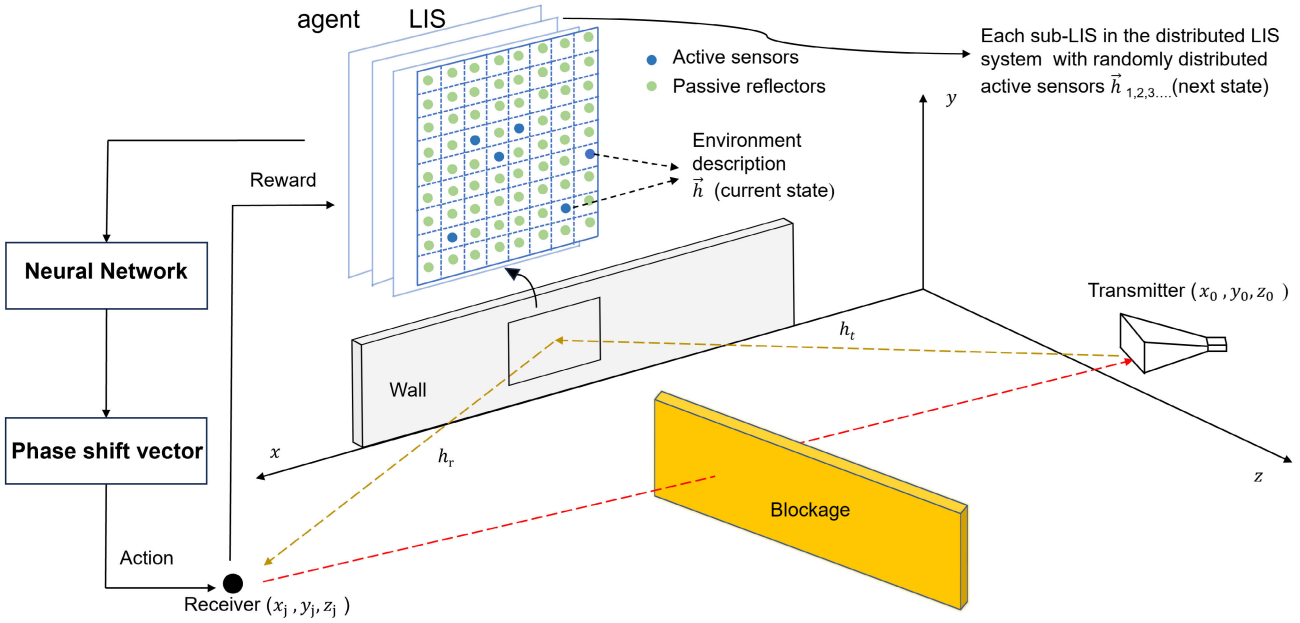

FIGURE 1. A structure of the distributed LIS system configuration enabled by deep reinforcement learning.

3) OBSERVABLE STATE VECTORS

The m^{th} sub-LIS' active cascaded channel and HWI noise constitute the observable state vector o^m. These vectors, in turn, create the distributed LIS system's state vector sequence $\{o^1, o^2, \ldots, o^m\}$.

The m^{th} observable state vector and its elements are,

$$o^m = f(h^m, \tilde{N}^m), \tag{14}$$

$$o_i^m = (h_t^m(i) + n_t^m(i)) * (h_r^m(i) + n_r^m(i)), \tag{15}$$

For each active acquisition element i, $n_t^m(i) = \sqrt{\tilde{N}_i^m} * (e + k * j)$, where e, k are normally distributed random variables. Similarly, $n_r^m(i)$ follows this pattern. Refer to (15), for the m^{th} sub-LIS in a distributed LIS system, the active cascaded channel and HWI noise collectively generate a sequence of state vectors $\{o^1, o^2, \ldots, o^m\}$.

Policy and Action Space: Defining the m^{th} sub-LIS of the reflection beamforming vector to the selection strategy π_0^m, the LIS distributed system forms a strategy sequence $\{\pi_0^1, \pi_0^2, \ldots, \pi_0^m\}$, and all the selection strategies constitute the strategy space π. Each sub-LIS, guided by its selection strategy π^m, sequentially outputs the sequence action a^m, where $m = 1, 2, \ldots, M$. In the LIS, each individual element is implemented solely with phase shifters, under the assumption that every reflection vector represents a phase shift, i.e., $\psi^m = e^{j\phi_m}$. The interaction vector is called the reflection beamforming vector, and each reflection beamforming vector is controlled by a phase shifter. However, due to hardware limitations, these phase shifters cannot shift signal by exact phase required and typically have a set of quantized angles [11], [30]. To satisfy the phase shifter constraints, the reflection beamforming vector ψ can only be selected from a predefined codebook B. The assumption is that each codeword in the codebook B, representing a potential reflection beam, utilizes a quantized phase shift to meet the constraints of the phase shifters [11], [30].

$$\psi \in B, \tag{16}$$

The whole system generates a codebook of the same size as the number of LIS elements. In this codebook B, each column represents a reflection beamforming vector, totalling M candidate vectors.

$$B = \begin{bmatrix} \psi_1^1 & \cdots & \psi_1^M \\ \vdots & \ddots & \vdots \\ \psi_M^1 & \cdots & \psi_M^M \end{bmatrix}, \tag{17}$$

The primary objective of the LIS system is to interact with incoming signals to optimize utility, with each sub-LIS aiming to maximize its data rate. Consequently, selecting the optimal beamforming vector through specific strategies is essential to achieve the highest possible gain. Define the set of actions $A = \{a^m\}$, where $a^m \in \{1, \ldots, M\}, m = 1, 2, \ldots, M$, each action is defined as an index of codewords in the codebook B, with each action bringing current rewards referring to the rate and long-term rewards referring to the utility.

4) STATE EVALUATION VALUE

The state of the system is related to the active channel and noise of the environments, and the value of the state that the system is in determines the high or low starting point for future state transfer, which is an evaluation that seeks to maximize long-term goals. The utility of the LIS distributed

system serves to assess the system's current state value. The channel's utility is defined as [20],

$$\gamma = \left|\frac{\partial C}{\partial S}\right| = \frac{P}{8l(\tilde{N}+\rho P)} \cdot \frac{\rho}{\tilde{N}} \cdot \frac{\partial \tilde{N}}{\partial l}, \tag{18}$$

where C is the capacity of the LIS. In the presence of HWI noise, the channel's utility is,

$$\gamma = \frac{P}{8l(\tilde{N}+\rho P)} \left(\frac{2\tau}{\pi z_0 \sqrt{2\tau^2+1}(\tau^2+1)} - \frac{\rho\beta 4^{\beta-\frac{1}{2}} P\alpha l^{2\beta-1}}{(\beta+1)z_0^2 \pi^{\beta+1} N_0 + 4^{\beta-1} P\alpha l^{2\beta}} \right), \tag{19}$$

The channel's utility upper bound can be expressed as,

$$\gamma_0 = \frac{1}{4z_0^2} \cdot \frac{1}{\arctan\left(\frac{\tau^2}{\sqrt{2\tau^2+1}}\right)} \cdot \frac{1}{\sqrt{2\tau^2+1}(\tau^2+1)}, \tag{20}$$

where $\tau = \frac{l^m}{z_0}$ is the normalized length of m^{th} sub-LIS.

Greedy algorithm [26] is used to achieve the maximum rate of a sub-LIS. This approach however could lead to a new state in the entire distributed system that either diminishes future rewards or prolongs the convergence time. To overcome this shortcoming, this study utilizes utility to improve state transitions in the distributed LIS system. During network training, the utility evaluates each sub-LIS' transitional state, and the utility's upper bound constrains future state transitions.

5) ACTION EVALUATION VALUE

In contrast to the distributed LIS system, which utilizes utility to assess state value, the sub-LIS aims for the maximum LIS rate, employing this rate to determine the value of actions. The state value estimates the value of the current action, which is a kind of evaluation that pursues the local maximum.

Define the action evaluation value of the sub-LIS as the actual rate of the system R^m, and the capacity as the upper bound rate. The capacity of the sub-LIS is,

$$C^m = \log\left(1 + \frac{\rho \frac{P}{M}}{\tilde{N}^m}\right), \tag{21}$$

where $\rho = \frac{1}{\pi}\arctan\left(\frac{\tau^2}{\sqrt{2\tau^2+1}},\right)$. Based on the active cascaded channel, the reflection beamforming vector ψ^m of the passive element on the m^{th} sub-LIS is related to the corresponding rate R^m as,

$$\psi_*^m = \max_{\psi \in B} \log_2\left(1 + \mathrm{SNR}^m \cdot |(h_t^m \odot h_r^m)^T \psi|^2\right), \tag{22}$$

where SNR represents the signal-to-noise ratio, and ψ_*^m is obtained by searching over the entire codebook B. Based on the system model previously outlined, the maximum rate for the m^{th} sub-LIS is defined as follows,

$$R_*^m = \max_{\psi \in B} \log_2\left(1 + \mathrm{SNR}^m \cdot |(h_t^m \odot h_r^m)^T \psi|^2\right), \tag{23}$$

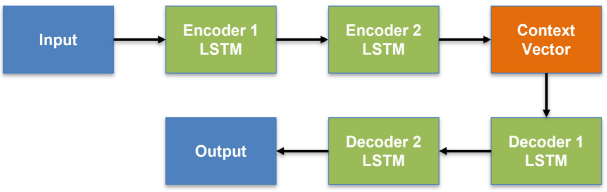

FIGURE 2. LSTM neural network structure.

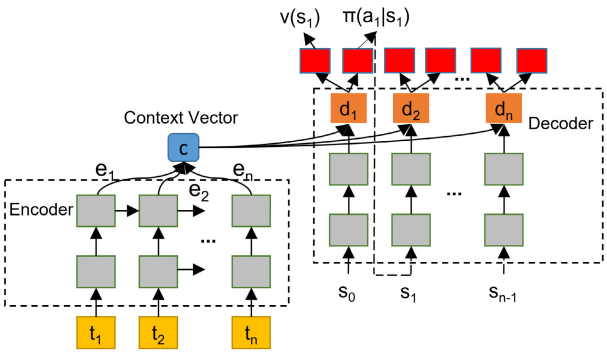

FIGURE 3. Seq2seq neural network structure.

This study treats the normalized capacity of a noise-free channel as the upper bound for the rate,

$$U_*^m = \log_2\left(1 + |(h_t^m \odot h_r^m)^T \cdot \frac{1}{H_{max}}|^2\right), \tag{24}$$

Each row of U_*^m is normalized individually, where $H_{max} = \max\left(\left(h_t^m \odot h_r^m\right)\right)^T$. U_*^m is the upper bound on the rate, which is the sub-LIS capacity C^m.

The sub-LIS outputs action a^m under the guidance of its selection strategy π^m. For the sub-LIS, it is to select the reflection beamforming vector ψ^m, and the environments will give the sub-LIS the corresponding action evaluation value after executing the action.

After each sub-LIS executes its selection of action a^m, the system gives the sub-LIS with corresponding immediate reward r^m. The overall goal of the sub-LIS is to maximize the total rewards $R^m = \sum_i r^i$ after the sub-LIS executes step i. Action value is determined by defining an action evaluation value, which quantifies the immediate reward's magnitude. From a global perspective, the new state s^{m+1} impacts the future rewards. Therefore, the state evaluation value is defined to quantify the extent of long-term rewards.

III. CAPACITY OPTIMIZATION OF THE LIS BASED ON MDRL

A. NEURAL NETWORK EXPRESSION OF THE ALGORITHM

In this study, the MDRL algorithm is applied to address the MDP model, where both input and output are sequential. To accurately represent and adapt to the environments and strategies, a Long Short-Term Memory (LSTM) neural network is utilized, as depicted in Fig. 2.

For managing long data sequences effectively, several LSTM neural networks are linked to form a

sequence-to-sequence (seq2seq) network, depicted in Fig. 3. In this configuration, the encoder and decoder each comprises two LSTM networks. Machine learning is effective for one-to-one mapping problems, yet sequential decision-making integration is crucial when outputs influence subsequent inputs. As illustrated in Fig. 3, Recurrent Neural Networks (RNNs) are well-suited for seq2seq mapping, particularly in encoding and decoding tasks. Among RNNs, LSTM excels in capturing long-term dependencies and retaining essential information due to its unique gating mechanism. This feature enables LSTM to efficiently handle long sequences in distributed LIS applications. Furthermore, layering multiple LSTM units facilitates the construction of more complex and deeper network architectures, thereby enhancing the model's ability to process complex sequence data.

All neural network parameters are denoted as θ, allowing the conditional probability of generating the optimal policy for action at state s to be represented as $\pi_\theta(a|s)$. For each input t_i, the neural network employs its encoder for initial encoding, facilitating learning and memory. Afterwards, the network's memory informs the processing of output d_j through the decoder. This output is then processed by two different activation functions, corresponding to the output state value function $v(s)$ and the probability of the decision sequence $\pi_\theta(a|s)$, respectively.

To constrain the current training, the input sequence from the distributed system is combined with the sub-LIS' decoded subsequence. This combination is transformed into a series of embedded vectors, which are then fed into the LSTM neural network as two-dimensional vectors. Fig. 3 illustrates that for a sub-LIS, the encoder's output vector c effectively captures information from all inputs. Consequently, the encoder's final hidden state serves as the initial hidden state for the decoder. This strategy requires the encoder and decoder to have an equal number of hidden layers and units, ensuring efficient information transfer. In contrast, for an inter-LIS, the decoder's parameters are randomly initialized, reflecting the environmental diversity and granting the model adaptability to varied communication scenarios. Such an initialization strategy imparts the required diversity for the model to address diverse communication challenges effectively.

The encoder input mapping and decoder output mapping are denoted as f_{enc}, f_{dec} respectively, and the encoded output can be activated separately, with the intervening hidden state as,

$$s_i = f_{\text{enc}}(t_i, d_1, \ldots, d_{i-1}), \tag{25}$$

Following multiple mapping layers, the rate is calculated using equation (23) based on the output from the decoding section. This calculation yields:

$$d_i = f_{dec}(s_j), \tag{26}$$

where t_i represents the observable state, while d_j corresponds to the utility during training and the rate during prediction stages, respectively.

The LSTM neural network outputs a vector d, with dimension n corresponding to the total number of LIS samples. Following non-linear activation, this vector generates the probability vector π_θ and the evaluation function $v(s_j)$ in the j^{th} dimension, respectively. The probability vector π_θ corresponds to the probability that the decision action will take a certain a, which sums to 1. At step j, the decision action is determined as $a_j = \arg\max_a(\pi_\theta)$, using the greedy algorithm.

B. STRATEGY FITTING AND PARAMETER OPTIMIZATION

This work adopts a two-layer iterative algorithm architecture with alternating action execution and state transfer to enable parameter updates and policy optimization. In the MDRL algorithm, the outer layer presents an environmental representation of the distributed LIS system, encompassing its contextual information and state transition process. This layer guides the inner layer sub-LISs in adopting specific strategies. The outer layer primarily focuses on the association and synchronization between sub-LISs and pursues the channel's utility of the overall distributed system. Meanwhile, the inner layer is dedicated to maximizing the rate of each individual sub-LIS. The relationship between these two layers is characterized as follows,

1) The input sequence is formed by combining the channels of each active element with the context of the sub-LIS in the outer loop. Merging these with the decoded optimal codebook index results in output sequences. Together, these sequences provide a representation of the distributed LIS system's environmental conditions.
2) Once the inner loop acquires initial environment knowledge, each sub-LIS employs the LSTM deep neural network. The purpose of this network is to ascertain the maximum rate and the appropriate codebook index for each sub-LIS.
3) The inner layer computes the utility of each action. Subsequently, it communicates both the maximum rate and the calculated utility value to the outer layer.
4) The outer layer integrates the action's utility and the maximum rate received from the inner layer with the current observable state, forming a comprehensive new state.
5) The inner and outer layers perform iterative calculations. After stabilization of the neural network parameters, each sub-LIS outputs the optimal strategy based on these parameters.

During the training stage, the algorithm passes the upper utility limit sequence of the decoded sub-LISs to each sub-LIS, treating it as an environmental factor. The utility upper bound of the decoded LIS as an internal state is combined with the observable state as the system's current state as s^m, which is used as a label for training.

$$s^m = \{o^m; \gamma_0^1; \gamma_0^2; \ldots; \gamma_0^{m-1}\}, \tag{27}$$

During the prediction stage, the action of each decoded sub-LIS, specifically the execution of its beamforming vector, will cause the transfer of the outer state. The optimal codebook index value from the decoded subsequence serves as the internal state, combined with the observable state to form the system's current state as s^m for prediction,

$$s^m = \{o^m; \psi^1; \psi^2; \ldots; \psi^{m-1}\}, \tag{28}$$

In the inner layer, when the m^{th} sub-LIS executes an action by selecting the beamforming vector ψ^m, the reward magnitude of action a^m is evaluated based on the rate. This action also induces a state transition in the outer distributed LIS system, moving from the m^{th} system state to the $(m+1)^{th}$ state, $s^m \rightarrow s^{m+1}$. The subsequent state, s^{m+1} is evaluated using the utility.

The primary reason behind employing a two-layer loop is to provide each sub-LIS with an initial adaptive foundation, tailored to its specific wireless environment, through the distributed LIS system's learning of the overall environments. This approach reduces the need for extensive repetitive training of individual sub-LISs from the ground up, thereby accelerating their learning convergence.

In this study, the Adam algorithm [31] is employed for gradient updating to facilitate neural network parameter adjustments. For the distributed LIS system, the objective function is defined as follows,

$$J = E(R), \tag{29}$$

E represents the expectation, R refers to the rate of LIS. Based on the sequence expression of the neural network, the objective function $J(\theta)$ can be defined as

$$J(\theta) = E_\theta(R(\theta)), \tag{30}$$

θ represents all parameters in the neural network, according to the law of the large numbers [32],

$$J(\theta) \approx \frac{1}{N} \sum_{m=1}^{M} \left(\sum_{i=1}^{N} R_i(\theta^m) \right), \tag{31}$$

where N is the number of samples sampled. Under the constraint of the utility,

$$\begin{cases} J(\theta) \approx \frac{1}{N} \sum_{m=1}^{M} \left(\sum_{i=1}^{N} R_i(\theta^m) \right) \\ \gamma(i) \leq \gamma_0 \end{cases}, \tag{32}$$

Next, based on the iterative transfer of the combined state, the Adam optimizer is used to optimize the network parameters.

$$\nabla_\theta J(\theta) = \frac{1}{M} \sum_{m=1}^{M} \left(\nabla_\theta J(\theta^m) \right), \tag{33}$$

The gradient update formula of the neural network parameters in the inner loop is,

$$\theta_i^m = \theta_{i-1}^m + \alpha_1 \nabla_\theta J(\theta), \quad i = 1, 2, \ldots, M \tag{34}$$

where α_1 represents the training coefficient, which is also the learning rate. When the neural network parameters converge iteratively, the output of the MDRL algorithm is the optimal decision.

C. ALGORITHMS

The following two pseudocode diagrams respectively depict the MDRL algorithm, as well as its associated data sourcing and validation processes.

Algorithm 1 MDRL Algorithm for Distributed LIS

Input: Given the observation space $\rho(\mathcal{T})$ of the distributed LIS
Output: Collect sub-LIS sequence samples from $\rho(\mathcal{T})$ to obtain the sub-LIS sequence sample set $\{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_i, \ldots\}$
Randomly initialize the state and action evaluation values γ^0 and R^0
for $\mathcal{T}_m \in \{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_i, \ldots\}$ **do**
    Use the sampling strategy π^m to collect the state trajectory $D = (\tau_1, \tau_2, \ldots)$ from $\mathcal{T}_m$
    for $\tau_m \in \{\tau_1, \tau_2, \ldots, \}$ **do**
        Train the LSTM neural network on τ_m using Adam algorithm
        Predict the rates corresponding to all beamforming vectors in the codebook, and calculate the corresponding rate and utility
        Execute the greedy algorithm to obtain the maximum rate R_*^m and the corresponding utility, and obtain the index number n of the corresponding beamforming vector ψ_*^m
    end for
    At this point, the system moves to the next state $s^m \rightarrow s^{m+1}$, let $m \leftarrow m+1$
end for

After training, validation data is input into the neural network with optimized parameters to obtain a sub-optimal reflection beamforming vector. The inner loop can rapidly converge with fewer samples and iterations in scenarios where the signal environment changes, leveraging the meta-strategy defined by the outer loop.

IV. RESULTS AND DISCUSSION

The simulation contains various parameters, including the parameters of the MDP model, neural network, DeepMIMO dataset and LIS array. A list of simulation parameters is presented in Table 1.

This study examines the effectiveness of the MDRL algorithm in assessing the performance of large-scale Uniform Planar Array (UPA) LIS systems in the presence of HWI effects. Additionally, it aims to minimize the required training sample sizes. Two deployment strategies, P1 and P2, are applied to a 32 × 32 large-scale LIS. Strategy P1 segments the LIS into sixteen 8 × 8 sub-LISs, whereas P2 divides it into four 16 × 16 sub-LISs. The simulation utilizes training

Algorithm 2 Data Sources and Validation Procedures

Input data:
Outer layer: Active cascade channel with HWI, i.e., observable state vectors o^m
Inner layer: Sub-LIS observable state vectors combined with context constraints equal s^m
Learning strategy:
Outer layer: Number of sub-LIS
Inner layer: Greedy algorithm
Output action:
Outer layer: Polling sub-LIS
Inner layer: Codebook reflection beamforming vector selection
Evaluation data:
Outer layer: State evaluation, long-term rate, cumulative sum of each sub-LIS rate.
Inner layer: Action evaluation, sub-LIS immediate rate R_*^m.
Performance upper bound:
Noise-free channel capacity U_*^m for performance comparison.

TABLE 1. Simulation parameters.

Parameters	Values
Frequency band	28 Ghz
Each sub-LIS elements	8*8, 16*16
Active receivers	36200
Antenna spacing	0.5λ
Active element	6
Transmit power	5
System bandwidth	100MHz
Transmitter gain	3
Receiver gain	3
LIS noise figure	5
Channel process gain	10
Learning rate	0.002
Training samples	22, 220, 2200
Validation samples	62
MDP discounting factor	0.9
LSTM neuron number	256
Dropout	0.5
Hidden unit number	256
Coding layer	2
Decoding layer	2

sample sizes of 22 (small sample), 220 (tenfold the sample size), and 2200 (hundredfold the sample size). The LSTM neural network is chosen as the fundamental basis for the deep learning model to conduct a thorough examination. The objective is to evaluate the predictive accuracy of the MDRL algorithm in estimating the performance rate of LIS systems under the influence of HWIs. Preliminary findings from this work indicate that the MDRL algorithm effectively decreases the necessary amount of training data needed to accurately predict the achievable rates in LIS systems affected by HWIs.

The modelling section of this research utilized the "O1" outdoor ray tracing scenario from the Deep MIMO

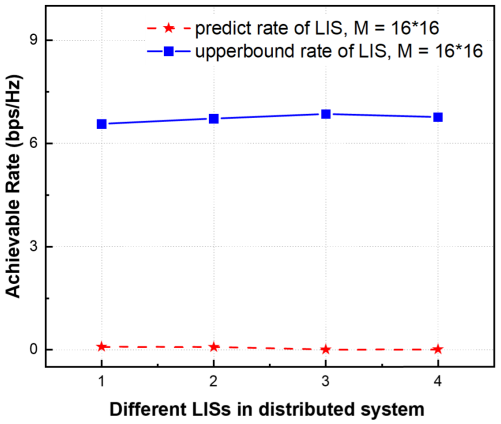

FIGURE 4. The achievable rate of 4 LISs for small samples.

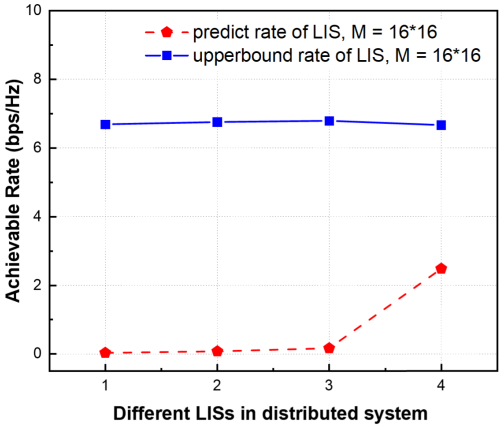

FIGURE 5. The achievable rate of 4 LISs for 10x samples.

dataset [33], as the foundational framework. Within this scenario, Base Station 3 (BS 3) was set up as a UPA LIS system. The transmitter's location was fixed at row 850 and column 90. To simulate receiver conditions, data from rows 1000 to 2000 in the "O1" scenario were used, with each row comprising 181 points, totalling 36,200 reception points. The active elements on the UPA antenna were randomly chosen to reflect real-world operating conditions.

This work investigates the performance of the LIS system under various HWI scenarios, utilizing the MDRL algorithm. The main focus is on evaluating the MDRL algorithm's effectiveness in reducing the dependency on extensive training samples, particularly in comparison to traditional machine learning techniques that often require substantial data volumes for training.

In scenarios with severe HWI, noise levels have been observed to significantly affect the LIS system's performance. Simulations indicate that the LIS system's maximum capacity averages around 7 bps/Hz. Utilizing the MDRL algorithm, the system achieves a prediction rate of 0.2 bps/Hz even with limited samples. By increasing the sample size by factors of 10 and 100, the prediction rates are enhanced to 2.5 bps/Hz and 3 bps/Hz, respectively. This trend emphasizes

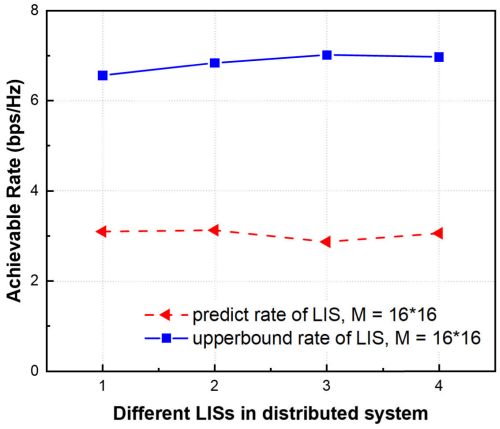

FIGURE 6. The achievable ate of 4 LISs for 100x samples.

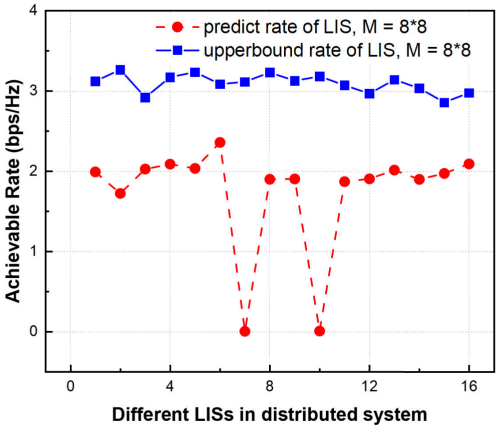

FIGURE 8. The achievable rate of 16 LISs for 10x samples.

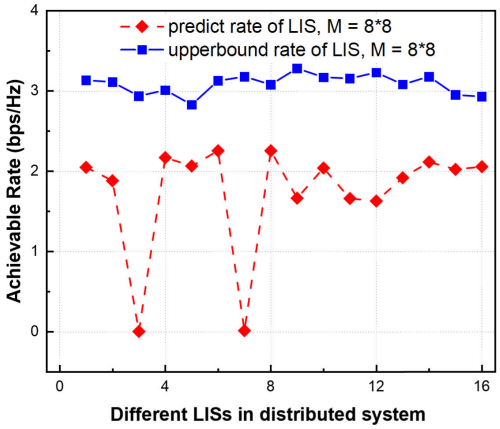

FIGURE 7. The achievable rate of 16 LISs for small samples.

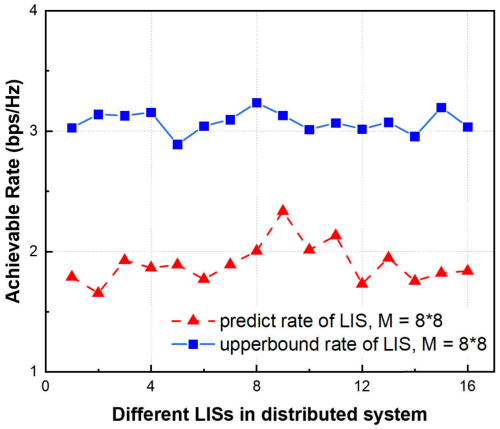

FIGURE 9. The achievable rate of 16 LISs for 100x samples.

the critical role of larger sample sizes in improving the precision of predictive modelling.

The MDRL algorithm is highly effective in reducing the number of training samples needed in scenarios involving minor HWIs. The simulation results, depicted in Fig. 7, 8, and 9, demonstrate that the average prediction rate is 1.9 bps/Hz across various sample sizes. The MDRL algorithm consistently produces accurate predictions, even with smaller sample sets. Comparatively, strategy P1 exhibits less fluctuation in prediction accuracy and the upper capacity limit than strategy P2, highlighting the model's stability in less severe HWI conditions.

Fig. 7 shows that using a limited number of active elements for LIS channel sensing can decrease rate performance in certain situations. Although this method accurately estimates the channel in various cases, it can lead to errors in some estimations. The primary issue arises from the unknown channel characteristics of passive elements, which depend on the detection outcomes from nearby active elements. This dependence can result in estimation inaccuracies, especially with significant discrepancies between adjacent channels, directly impacting rate prediction accuracy. To explore this issue, simulations were conducted by dividing the LIS into 64 sub-LISs and changing the proportion of LIS active elements. Fig. 10 shows subdividing the LIS into additional sub-LIS segments enhances the system's resilience to HWI.

However, employing all active elements in LISs increases power while also exacerbating the negative impacts of HWIs, resulting in a decreased SNR. A distributed strategy that employs sub-LISs interconnected by inter-sub-LIS sequences presents an effective solution. Enhanced by meta-learning, this strategy refines channel sensing accuracy and mitigates HWIs by reducing sub-LIS component counts. Such modifications greatly accelerate the convergence of the model and lower the need for training samples. The simulations presented in Fig. 10 demonstrate that a distributed sub-LIS configuration significantly improves rate performance.

The MDRL approach presented in this study strategically assesses the channel by randomly adjusting active LIS elements within each sub-LIS, thereby acquiring crucial prior environmental knowledge. This dual-layer deep learning strategy exhibits superior adaptability over traditional machine learning techniques [11], [15], [34], [35], resulting in significant reductions in both training time and sample requirements. Achieved via meta-learning optimization of the distributed LIS, this methodology marks a considerable

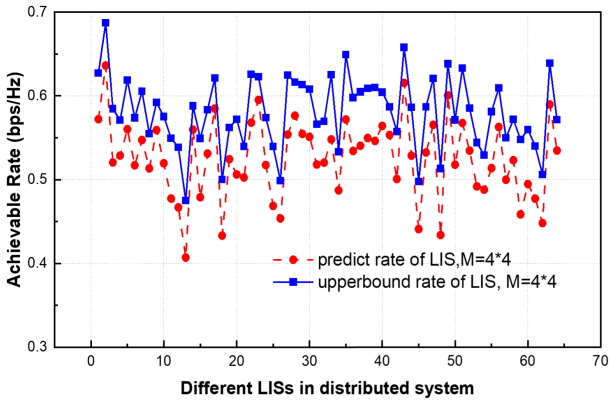

FIGURE 10. The achievable rate of 64 LISs for 10x samples.

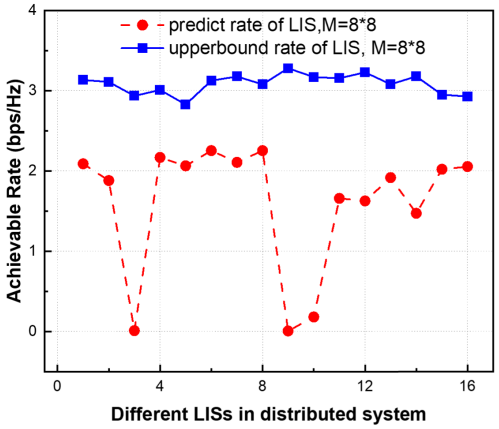

FIGURE 11. The achievable Rate of 16 LISs (6 Active Elements per LIS) for 10x samples.

advancement in channel estimation and LIS optimization in complex environments.

Traditional machine learning approaches typically depend on extensive training data, frequently consisting of hundreds or tens of thousands of samples, to reach comparable levels of accuracy in predicting the reachability rates of LIS systems [11], [15], [34], [35]. Nevertheless, the MDRL method reduces the need for big datasets, maintaining high prediction accuracy even with limited data. This characteristic is particularly beneficial in real-world scenarios where collecting vast amounts of data is expensive or when data availability is restricted.

A comprehensive analysis of the results in Fig. 4, 5, 6, and Fig. 7, 8, 9 demonstrates that the MDRL algorithm significantly reduces the number of required training samples and improves sample efficiency in diverse HWI scenarios. Significantly, the algorithm is capable of achieving commendable prediction rates with only hundreds of data points in scenarios with 4 sub-LISs. Furthermore, the model's capacity to predict stability has been enhanced in the context of 16 sub-LISs, especially when the influence of HWIs is minimised.

For strategy P2, especially under severe HWI conditions, a marked increase in training sample requirements is

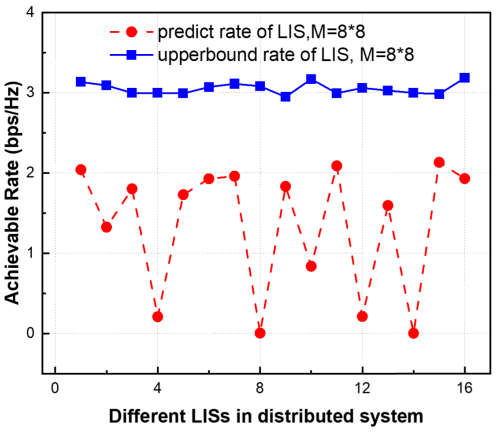

FIGURE 12. The achievable Rate of 16 LISs (12 active elements per LIS) for 10x samples.

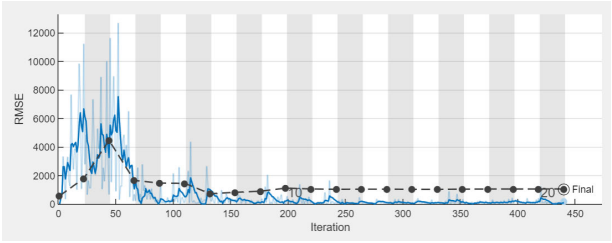

FIGURE 13. Convergence process of MDRL algorithm state transfer (small sample size training).

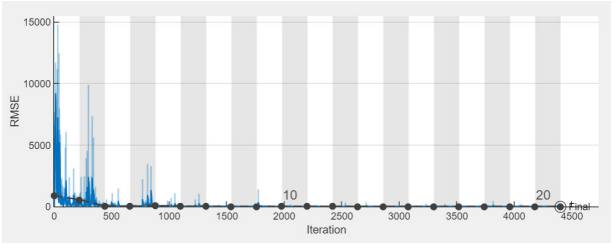

FIGURE 14. Convergence process of MDRL algorithm state transfer (10x sample size training).

observed. This trend is depicted in Fig. 13 and 14, showing the Root Mean Square Error (RMSE) curves' variations for both training and validation datasets across different iterations. The data trends in these figures highlight the difficulties in accurately evaluating channel information in noisy HWI environments, leading to inaccuracies in the model's predictions and deviations from the system's maximum achievable rate. The study indicates that in environments with high noise, an increased number of training samples is crucial for improving prediction accuracy. This highlights the importance of expanding the sample size to address challenges in complex scenarios where noise significantly affects the results.

The study in Fig. 11 and Fig. 12 examine channel characteristics by varying the number of active elements within each distributed sub-LIS. Findings indicate that even a few active elements can effectively capture the environmental

context due to the smaller coverage area of each sub-LIS. Simulations shown in Fig. 11 and 12, conducted with consistent power usage, reveal that six active elements in a sub-LIS yield the most efficient configuration, while increasing to twelve active elements decreases performance. This decrease is mainly due to the fact that more active elements increase the system's power and amplify the effects of HWIs, thus diminishing the SNR.

Comprehensive study in this work shows that entirely passive LIS underperform due to inadequate channel detection and limited environmental comprehension. Within specific power constraints, the strategic addition of active elements is identified as an effective method to reduce HWIs and improve channel detection accuracy. Furthermore, dividing the LIS into more sub-LISs and reducing the number of elements per sub-LIS efficiently mitigates HWI effects. Integrating active elements in this manner not only elevates the signal-to-noise ratio but also substantially minimizes double fading, thus boosting LIS-based system performance.

The proposed two-layer MDRL algorithm significantly improves environmental perception accuracy in this study. This is achieved by strategically deploying a designated number of active elements on sub-LISs within a distributed LIS and utilizing channel data for model input. Importantly, the algorithm effectively addresses the amplification of HWIs that often results from the overuse of active elements, thus substantially enhancing system performance. These outcomes highlight the vital importance of the number and strategic placement of active elements in LIS system design. Furthermore, the MDRL algorithm's success in refining LIS configurations offers valuable perspectives for the development of future communication systems, showcasing its potential to advance network functionalities.

The study findings demonstrate that the MDRL algorithm performs exceptionally well in managing distributed LIS systems across various HWI scenarios. Compared to traditional machine learning methods, the MDRL substantially reduces reliance on large training data sets, improving data efficiency and system flexibility. This research lays the foundation for further optimization of LIS systems in data-limited communication settings, leading to innovative developments in algorithm optimization despite significant HWI obstacles.

V. CONCLUSION

This study tackles the optimization challenges in distributed LIS systems in the face of HWIs by employing a novel learning approach. Demonstrating significant adaptability and robustness in diverse HWI conditions, this method markedly reduces dependency on large training samples while maintaining high predictive performance. It emphasizes the importance of increasing sample sizes in high-noise environments, showcasing the MDRL algorithm's superior data efficiency and adaptability. This study is based on narrowband systems, and suggests that its methods could be extended to multi-carrier broadband systems and multi-antenna configurations. These findings open new avenues for optimizing LIS systems in data-constrained communication environments and set the stage for further research, especially in algorithmic optimization in HWI environments.

REFERENCES

[1] 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.

[2] M. T. Mamaghani and Y. Hong, "Aerial intelligent reflecting surface-enabled terahertz covert communications in beyond-5G Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19012–19033, Oct. 2022.

[3] S. Hu, K. Chitti, F. Rusek, and O. Edfors, "User assignment with distributed large intelligent surface (LIS) systems," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1–6.

[4] 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.

[5] J. Qiao, C. Zhang, A. Dong, J. Bian, and M.-S. Alouini, "Securing intelligent reflecting surface assisted terahertz systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8519–8533, Aug. 2022.

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

[7] X. Shao, C. You, W. Ma, X. Chen, and R. Zhang, "Target sensing with intelligent reflecting surface: Architecture and performance," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2070–2084, Jul. 2022.

[8] C. You, B. Zheng, and R. Zhang, "Channel estimation and passive beamforming for intelligent reflecting surface: Discrete phase shift and progressive refinement," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2604–2620, Nov. 2020.

[9] B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface assisted multi-user OFDMA: Channel estimation and training design," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8315–8329, Dec. 2020.

[10] T. Jiang, H. V. Cheng, and W. Yu, "Learning to reflect and to beamform for intelligent reflecting surface with implicit channel estimation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1931–1945, Jul. 2021.

[11] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling large intelligent surfaces with compressive sensing and deep learning," *IEEE Access*, vol. 9, pp. 44304–44321, 2021.

[12] C. Pan, G. Zhou, K. Zhi, S. Hong, T. Wu, Y. Pan, H. Ren, M. D. Renzo, A. Lee Swindlehurst, R. Zhang, and A. Y. Zhang, "An overview of signal processing techniques for RIS/IRS-aided wireless systems," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 5, pp. 883–917, Aug. 2022.

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

[14] W. Mei, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless networks: From single-reflection to multireflection design and optimization," *Proc. IEEE*, vol. 110, no. 9, pp. 1380–1400, Sep. 2022.

[15] Y. Ge and J. Fan, "Beamforming optimization for intelligent reflecting surface assisted MISO: A deep transfer learning approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3902–3907, Apr. 2021.

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

[17] C. Huang, G. Chen, Y. Gong, M. Wen, and J. A. Chambers, "Deep reinforcement learning-based relay selection in intelligent reflecting surface assisted cooperative networks," *IEEE Wireless Commun. Lett.*, vol. 10, no. 5, pp. 1036–1040, May 2021.

[18] S. Zarei, W. H. Gerstacker, J. Aulin, and R. Schober, "Multi-cell massive MIMO systems with hardware impairments: Uplink-downlink duality and downlink precoding," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5115–5130, Aug. 2017.

[19] C. Shan, Y. Zhang, L. Chen, X. Chen, and W. Wang, "Performance analysis of large scale antenna system with carrier frequency offset, quasi-static mismatch and channel estimation error," *IEEE Access*, vol. 5, pp. 26135–26145, 2017.

[20] S. Hu, F. Rusek, and O. Edfors, "Capacity degradation with modeling hardware impairment in large intelligent surface," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[21] Z. Xing, R. Wang, J. Wu, and E. Liu, "Achievable rate analysis and phase shift optimization on intelligent reflecting surface with hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5514–5530, Sep. 2021.
[22] C. Chen, M. Wang, B. Xia, Y. Guo, and J. Wang, "Performance analysis and optimization of IRS-aided covert communication with hardware impairments," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5463–5467, Apr. 2023.
[23] K. Zhi, C. Pan, H. Ren, K. K. Chai, and M. Elkashlan, "Active ris versus passive ris: Which is superior with the same power budget?" *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1150–1154, Jun. 2022.
[24] S. Kim, H. Lee, J. Cha, S.-J. Kim, J. Park, and J. Choi, "Practical channel estimation and phase shift design for intelligent reflecting surface empowered MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 6226–6241, Aug. 2022.
[25] B. Zheng, C. You, W. Mei, and R. Zhang, "A survey on channel estimation and practical passive beamforming design for intelligent reflecting surface aided wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1035–1071, 2nd Quart., 2022.
[26] J. Zhang, J. Li, Y. Zhang, Q. Wu, X. Wu, F. Shu, S. Jin, and W. Chen, "Collaborative intelligent reflecting surface networks with multi-agent reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 532–545, Apr. 2022.
[27] S. Ahmad, S. Khan, K. S. Khan, F. Naeem, and M. Tariq, "Resource allocation for IRS-assisted networks: A deep reinforcement learning approach," *IEEE Commun. Standards Mag.*, vol. 7, no. 3, pp. 48–55, Aug. 2023.
[28] R. Alghamdi, R. Alhadrami, D. Alhothali, H. Almorad, A. Faisal, S. Helal, R. Shalabi, R. Asfour, N. Hammad, A. Shams, N. Saeed, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "Intelligent surfaces for 6G wireless networks: A survey of optimization and performance analysis techniques," *IEEE Access*, vol. 8, pp. 202795–202818, 2020.
[29] S. Hu, F. Rusek, and O. Edfors, "Beyond massive MIMO: The potential of positioning with large intelligent surfaces," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1761–1774, Apr. 2018.
[30] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 375–388, Jan. 2021.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[32] K. Zhi, C. Pan, H. Ren, and K. Wang, "Uplink achievable rate of intelligent reflecting surface-aided millimeter-wave communications with low-resolution ADC and phase noise," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 654–658, Mar. 2021.
[33] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, Feb. 2019, pp. 1–8. [Online]. Available: https://www.deepmimo.net/
[34] S. Dinh-Van, T. M. Hoang, R. Trestian, and H. X. Nguyen, "Unsupervised deep-learning-based reconfigurable intelligent surface-aided broadcasting communications in industrial IoTs," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19515–19528, Oct. 2022.
[35] A. M. Elbir, A. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "Deep channel learning for large intelligent surfaces aided mm-wave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1447–1451, Sep. 2020.

YIFAN MAO received the B.Eng. degree in communication engineering from Hangzhou Dianzi University, in 2019, and the M.Sc. degree (Hons.) from the Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K., in 2020, where he is currently pursuing the Ph.D. degree in wireless communication and intelligent reflecting surface technology with the Department of Electrical and Electronic Engineering. His research interests include large intelligent surfaces, array signal processing, and sixth-generation wireless communications.

XIAOYU XIAO received the B.Eng. degree in electronics from Chengdu University of Information Technology, Chengdu, China, and the M.Sc. degree (Hons.) from the Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K., in 2020, where he is currently pursuing the Ph.D. degree in wireless communication and intelligent reflecting surface technology. His research interests include non-volatile RF switches, graphene/2-D materials RF, and fifth-generation wireless communications.

ZHIRUN HU (Member, IEEE) received the B.Eng. degree in communication engineering from Nanjing, China, and the M.B.A. and Ph.D. degrees in electrical and electronic engineering from Queens' University Belfast, U.K. He is currently a Professor of RF and microwave electronics. He serves as the Head of the Materials, Devices and Systems Division, Department of Electrical and Electronic Engineering, The University of Manchester, England, U.K. He has published more than 300 peer-reviewed journals and conference papers. His current research interests include data communications and networks, machine learning for communication systems and device-level designs, graphene/2-D materials enabled reconfigurable devices, circuits, and metasurfaces for wireless sensing and communications, and the IoT applications, RF energy harvesting, and transfer.