Fully Automated Design Method Based on Reinforcement Learning and Surrogate Modeling for Antenna Array Decoupling

Wei, Zhaohui; Zhou, Zhao; Wang, Peng; Ren, Jian; Yin, Yingzeng; Pedersen, Gert Frølund; Shen, Ming

Published in:
I E E E Transactions on Antennas and Propagation

DOI (link to publication from Publisher):
10.1109/TAP.2022.3221613

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

Fully Automated Design Method Based on Reinforcement Learning and Surrogate Modelling for Antenna Array Decoupling

Zhaohui Wei, Zhao Zhou, Peng Wang, Jian Ren, *Member, IEEE,* Yingzeng Yin, *Member, IEEE,* Gert Frølund Pedersen, *Senior Member, IEEE,* and Ming Shen, *Senior Member, IEEE*

***Abstract*—Modern electromagnetic (EM) device design generally relies on extensive iterative optimizations by designers using simulation software (e.g. CST), which is a very time-consuming and tedious process. To relieve human engineers and boost productivity, we proposed a machine learning framework to solve the problem of automated design for EM tasks. The proposed approach combines advanced reinforcement learning (RL) algorithms and deep neural networks (DNNs) in an attempt to simulate the decision-making process of human designers to realize automation learning. Specifically, the RL-based agent can interact with the EM design software without engaging human designers, allowing for automated design. Besides, the data accumulated during EM software simulation in the early design stage are reused as training data to build a DNN surrogate model to replace the time-consuming EM simulation and further accelerate the training of RL to achieve better optimization of EM design. Two types of antenna array decoupling including 1×2 and 1×4 arrays working at 3.5 GHz are used as test vehicles to validate the proposed method. The decoupling metasurfaces designed by the proposed fully automated method based on RL showed satisfactory results comparable to the results achievable by human designers. This indicates that the proposed method can be used to build powerful tools to boost the design efficiency of EM devices.**

***Index Terms*—design automation, decoupling metasurface, reinforcement learning, deep neural networks**

I. Introduction

EM metasurfaces are recently utilized to minimize the coupling between antenna elements [1–4] due to their outstanding manipulation of EM waves. Traditional design methods [5–9] rely on extensive trial and error by expert engineers, which is a time-consuming and inefficient procedure as the simulated data cannot be constructively reused. Furthermore, to obtain an acceptable decoupling performance, the metasurface and array antenna are generally co-simulated, which increases the structure's complexity and lengthens the time for each EM iteration. As a result, swiftly designing a satisfactory decoupling metasurface is a challenging task.

In recent years, artificial neural networks (ANNs) are rapidly emerging as a powerful tool in EM-based modeling and optimization [10], such as passive circuits and components [11–13], field-effect transistors [14], antennas [15–18], and fault detection and diagnosis [19], [20]. Deep learning (DL) enhanced the ability of ANNs by utilizing deeper networks and more neurons and has become a dominant paradigm in addressing more complex problems [21], [22]. In [23], a hybrid DNN model was proposed to model and simulate microwave filters, leading to a higher accurate result with fewer samples than traditional ANNs. To further improve training efficiency, domain knowledge was employed to assist DNNs with the design of metalens antenna [24], metasurfaces [25–28], frequency selective surface (FSS) [29], [30], mode recognition [31], and reflectarray [32]. Moreover, a multi-branch DNN-assisted strategy [33] was presented to enhance the algorithm robustness by searching for multiple different branches. Easum et al. [34] combined a black-box multiobjective optimization technique with a surrogate model for the design of monopole and vivaldi antennas. On the other hand, recent methodologies developed within the so-called System-by-Design (SbD) framework [35], enable effective and computationally efficient integration of machine learning surrogate models with evolutionary optimizers. By means of the superior prediction capabilities of the surrogate model and solution-exploration abilities of optimizers, the SbD framework has been successfully applied for the design of complex EM devices and systems, including isotropic lenses[36], innovative radomes[37], reflectarray antennas[38], multiband antennas [39], etc.

Another prominent machine learning method is reinforcement learning (RL) [40]. Unlike other ML methods described above, which require the human designer to annotate the training data set, the RL algorithm can automatically explore an unknown environment to collect training data by interacting with it using the trial-and-error method. RL algorithms have been successfully applied in a variety of fields of science and technology, including wireless communication [41], [42], resource allocation [43], video games [44], machine translation [45], and EM component design [46–48], and so on. Furthermore, AlphaGo built by DeepMind based on DNNs and RL, which is the same approaches taken by this work, has defeated the world human champion of Go [49] and its

Manuscript received *** **, ****; revised *** **, ****. This work is sponsored by China Scholarship Council. (Corresponding author: Jian Ren and Ming Shen)

Zhaohui Wei, Zhao Zhou, Peng Wang, Gert Frølund Pedersen, and Ming Shen are with the Department of the Electronic Systems, Aalborg University, 9220 Aalborg, Denmark. (Ming Shen: mish@es.aau.dk).

Jian Ren, and Yingzeng Yin are with the National Key Laboratory of Antennas and Microwave Technology, Xidian University, Xi'an 710071, China. (Jian Ren: renjian@xidian.edu.cn).

Color versions of one or more of the figures in this communication are available online at https://ieeexplore.ieee.org.

Digital Object Identifier:

upgraded version, AlphaGo Zero, has reached a level way beyond human capacity [50]. Moreover, the time needed in chip design floorplanning, by using RL, can be reduced from months to hours as reported by Google [51]. These successful achievements have encouraged us to apply RL and DL to leverage the automated design of antennas and surfaces.

In this work, we are motivated to apply RL and deep learning for the automated design of decoupling metasurfaces (DCMSs), in an attempt to minimize the mutual coupling between array antenna elements. The design of DCMSs is one of the most challenging EM design tasks which usually require at least days of design time by human designers, and we aim at shortening the design time while maintaining the design outcomes. For this purpose, we first design an RL-based framework to transform the array antenna decoupling problem into a Markov Decision Process (MDP) by defining states, actions, and rewards. Subsequently, the RL-based algorithm can independently learn to self-adjust the parameters of DCMS according to the design target. The EM data collected by CST is not only used to make decisions for the virtual agent but also used to build a surrogate model for CST to speed up the EM simulations. It is worth mentioning that these two processes are carried out simultaneously. Compared to existing supervised deep learning design methods that require manual pre-processing and annotation of training data sets, the proposed approach has the advantage of automated annotation, and improved quality of the training data set. This is because it can analyze the correct exploration direction from the prior data, reducing many ineffective explorations and enhancing the speed of convergence of the surrogate model by virtue of the decision-making capability of the RL algorithm. The RL-based virtual agent can interact directly with the surrogate model once it has been trained, drastically lowering the time it takes to locate the goal solution. Finally, the proposed approach is tested by designing a 1×4 array antenna DCMS.

The remainder of this paper is organized as follows. The basic knowledge of RL is given in Section II. Section III presents the details of the proposed method and algorithm implementation. Then, the performance of the proposed method is verified through two numerical experiments in Section IV. Finally, we conclude this paper in Section V.

II. Preliminary

A. Terminologies of RL

As shown in Fig. 1, RL can be demonstrated by terminologies of state, action, and reward. Based on the current state s_k, the agent select an action a_k to perform under the control of policy $\pi\left(a_k \mid s_k\right)$. Then the environment makes a transition from the current state s_k to a new state s_{k+1} and return a reward r_k to the agent. The reward is required to be previously defined and can be adjusted flexibly in terms of different problems. Also, it is an indicator that reflects how good or how bad action is. According to the rewards from the environment, the agent tends to select more good actions to perform while bad actions are excluded.

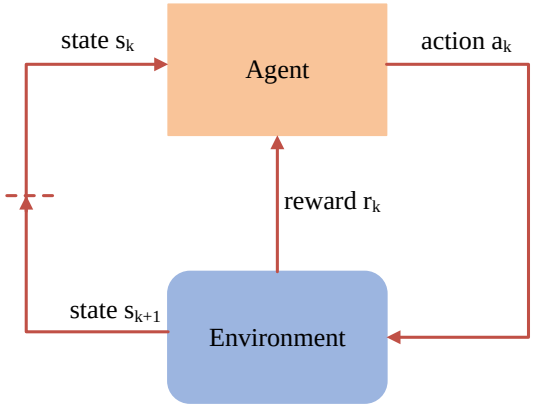

Fig. 1. The agent-environment interaction in reinforcement learning.

B. Q-learning Framework

Q-learning is an off-policy RL algorithm whose objective is to find an optimal action $a^*(k)$ to take given the current state s_k. The Q in Q-learning means Q-value function, which is introduced to evaluate the impact of the action a_k under the policy $\pi\left(a_k \mid s_k\right)$. The Q-value function also called the discounted cumulative reward, is given by [52]

$$\begin{aligned} Q\left(s_k, a_k\right) &= \boldsymbol{E}\left[U_k \mid s_k, a_k\right] \\ &= \boldsymbol{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{k+t+1} \mid s_k, a_k\right] \\ &= \boldsymbol{E}\left[r_{k+1}+\gamma r_{k+2}+\gamma^2 r_{k+3}+\cdots \mid s_k, a_k\right] \end{aligned} \tag{1}$$

where $\boldsymbol{E}$ denotes the expectation operation and the discount factor $\gamma \in [0,1)$ ensures the sum converges. As a mathematical trick, γ is used to balance immediate and future rewards. γ approaching zero means the agent would mainly consider immediate rewards, while γ approaching one means the future rewards would be considered with greater weight. Moreover, by observing the definition of the Q-value function, we can find that all future rewards are required to calculate the Q-value of each action. In other words, it is only calculated until one episode (each episode is composed of the agent moving from the initial state to the goal state) is finished. However, obtaining the total reward to compute the Q-value function would be difficult and time-consuming, especially for complex scenarios. To solve the problem, the temporal difference algorithm improves the updating rule by predicting the long-term future rewards of each action, which is given by

$$\begin{aligned} Q\left(s_k, a_k\right) =& Q\left(s_k, a_k\right)+\alpha \cdot\left[r\left(s_k, a_k\right)\right. \\ &\left.+\gamma \cdot \max_{a_{k+1} \in \mathrm{A}} Q\left(s_{k+1}, a_{k+1}\right)-Q\left(s_k, a_k\right)\right] \end{aligned} \tag{2}$$

where $\alpha \in (0,1)$ stands for the learning rate. All the state-action pairs and the corresponding Q-value form the Q-value table. The goal of training is to optimize the Q-value table. Once the Q-value table is close enough to convergence, it can be utilized to find the optimal sequences of states by performing the actions with the highest Q-values at each state.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

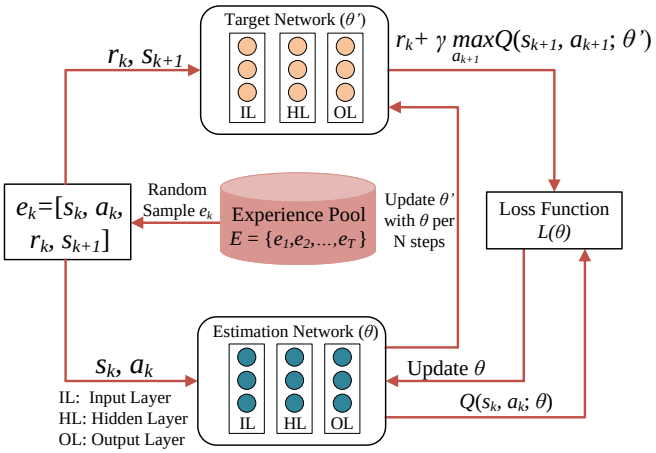

Fig. 2. Deep Q network (DQN) architecture.

C. The Selection Strategy

The exploration and exploitation are pairs of contradictions in the action-selection process. For greedy algorithms, exploitation is mainly considered; that is, the agent always selects the actions with the highest reward to perform. Although the greedy algorithm takes full advantage of the experience from prior exploration, it is easy to trap into the local optimum. On the contrary, the random selection algorithm considers the exploration with greater weight. The local optimum can be avoided by randomly selecting actions. However, the random selection algorithm has a very slow convergence velocity. Therefore, a trade-off has to be made between exploration and exploitation.

In this paper, we use the ϵ-greedy selection strategy. It can be represented by

$$a_k = \begin{cases} \arg\max_{a \in A} Q(a), 1-\epsilon \\ \text{random action}, \epsilon \end{cases} \tag{3}$$

In the ϵ-greedy algorithm, the agent randomly selects an action with the probability of ϵ while selecting the action with the highest Q value with the probability of 1-ϵ. Due to the introduction of ϵ, the agents can achieve a good balance between exploration and exploitation.

D. DQN Algorithm

The design of DCMS is a complex issue because there are many parameters to tune. That would mean that tremendous states and actions need to be stored by the Q-value table, which is impractical for the Q-learning algorithm to implement. DQN has solved this problem by introducing a deep learning technique into Q-learning. Using DNN to approximate the Q-value function instead of the Q-value table enables the Q-learning algorithm to deal with high dimensional input problems.

The DQN architecture is shown in Fig. 2. We can see that the DQN consists of two identical DNNs except for the weights, namely estimation, and target network. At the time k, the agent's experience e_k is defined as a tuple (s_k, a_k, r_k, s_{k+1}). By interacting with the environment, the agent can obtain extensive experiences, which are stored in the experience pool. In the training process, these experiences are randomly sampled from the experience pool to cut off their correlations. This method is called experience memory replay, which has been shown that it can significantly improve and stabilize the DQN training. Moreover, we can also find that the experience retrieved from experience pool are divided into two parts, (s_k, a_k) and (r_k, s_{k+1}) pairs. The former is input into the estimation network, while the latter is input into the target network. As a result, the $Q(s_k, a_k; \theta)$ and $Q(s_{k+1}, a_{k+1}; \theta')$ can be obtained, respectively. Next, the loss function $L(\theta)$ can be calculated by

$$L(\theta) = E\left[r_k + \gamma \max_{a_{k+1}} Q(s_{k+1}, a_{k+1}; \theta') - Q(s_k, a_k; \theta)\right] \tag{4}$$

It is worth mentioning that the target and estimation network is asynchronously updated. In the beginning, the weights of the target network θ' are frozen, and then only the estimation network θ is updated. After updating the N steps, the weights of the target work are updated to the new weights of the estimation network.

III. The Proposed Method and Implementation

A. Problem Statement

Fig. 3 shows the structure of a 1×2 array antenna decoupling design. The height of the antenna element from the ground plane is $H_2 = 4.5$ mm. The length of the antenna element is $l_p = 30.5$ mm. The spacing between the antenna elements is $s_p = 1.5$ mm. Generally, the adjacent array antenna elements are arranged very close to each other to save space (as shown in Fig. 3(b)), which makes the inter-cell coupling become strong and seriously affects the performance of the array antenna, including pattern distortion and impedance mismatch. To solve this problem, a DCMS can be placed above the array antenna (as shown in Fig. 3(a)) in an attempt to reduce the coupling between array antenna elements by manipulating the scattering of EM waves, which has achieved good results in [1–4]. However, the process of designing and adjusting the DCMS can be very complicated and time-consuming. On the one hand, DCMS itself (as shown in Fig. 3(c)) has many parameters to be adjusted, and the parameters are interrelated with each other. On the other hand, the introduction of DCMS will destroy the impedance matching of the array antenna, and the antenna matching has to be re-adjusted. Therefore, how to improve the design efficiency of DCMS according to different array antennas is a big challenge.

B. System Framework

We proposed and constructed a scenario for the design of DCMS based on RL and surrogate modeling, as illustrated in Fig. 4. The objective of this scenario is to build a mechanism that automatically designs the DCMS for array antennas. Note that the proposed design system needs to be trained, and a well-trained RL algorithm and surrogate model can accelerate the DCMS designs according to different design targets.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

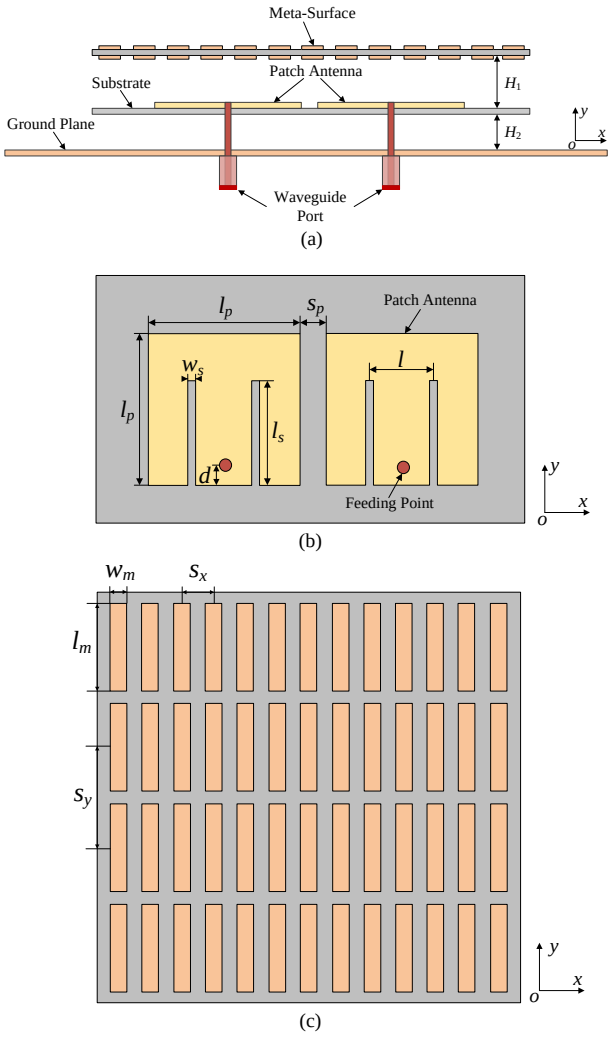

Fig. 3. Geometry of 1×2 array antenna and the corresponding DCMS. (a) Front view of the overall structure. (b) Top view of the 1×2 array antenna. (c) Top view of the DCMS.

There are three stages in the proposed system framework:

1) Early training and data-collection stage: the RL-based agent learns on its own and automates the generation of the training data set by interacting with the environment without human involvement. DCMS works as the manipulatable component whose parameters can be adaptively tuned under the control of the RL algorithm. Full-wave EM simulation software CST is used to evaluate the performance of DCMS and assign rewards accordingly. The collected data is mainly used for the training of the RL algorithm. In addition, it is also re-used to train a surrogate model in stage 2 to speed up the EM simulation.

2) Surrogate training stage: A surrogate model is built and trained to rapidly predict the scattering response of the EM structures. In the training process, the geometric parameters of DCMS are used as the input of the surrogate model, while the scattering parameters are used as the output, and a loss threshold is set to determine whether the surrogate model is sufficiently trained or not. In this paper, this loss threshold

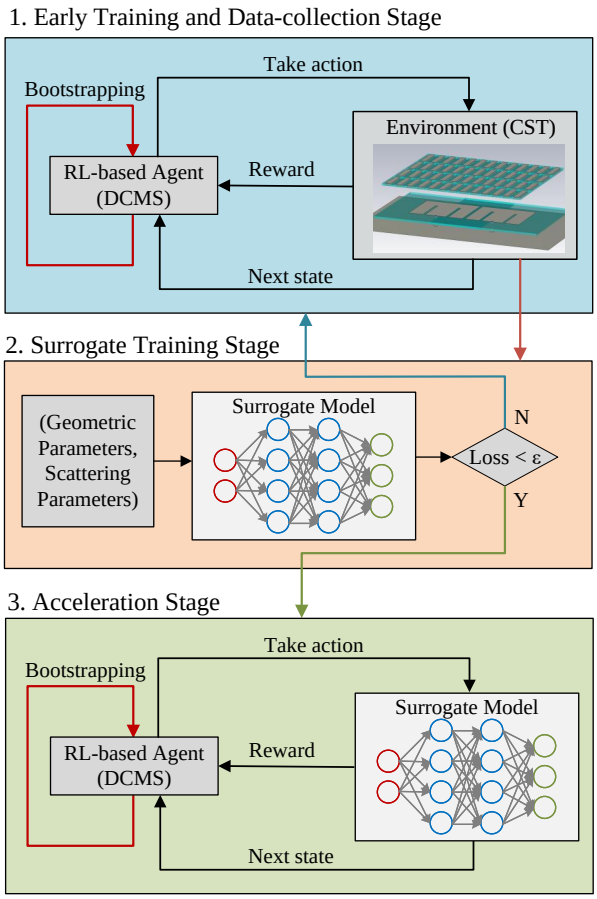

Fig. 4. System framework of the proposed method.

is set to 0.003, which allows for a balance between training accuracy and efficiency. Table I lists the parameters of the surrogate model, which is based on the platform of the Pytorch ML framework using a ThinkStation P920 Workstation computer.

3) Acceleration stage: Once the surrogate model is trained, the RL algorithm will stop interacting with CST but directly with the surrogate model. In such a condition, a large number of training data can be obtained in a short time, which contributes to the fast convergence of the RL-based model.

The RL algorithm can be trained by exploiting the train system shown in Fig. 2. Once the convergence of the RL algorithm is reached, the corresponding parameters of DCMS can be found.

C. Surrogate Model Training

The training of the surrogate model is described in detail in this subsection. This is a typical prediction problem in which the geometric parameters of the EM model are input and the surrogate model outputs its corresponding reflection and transmission coefficients. Since there are 9 geometric parameters to be optimized, the input layer of the neural network has nine neurons. The default dimension of the reflection and transmission coefficients exported from CST is 1001. To reduce the training difficulty and improve training efficiency, we reduce them to 101 dimensions and merge them,

TABLE I
PARAMETERS OF THE SURROGATE MODEL.

Variable	Value
Learning rate	0.0005
Optimizer	Adam
Batch size	32
Episode	2000
No. of neurons in input layer	9
No. of neurons in output layer	202
No. of hidden layers	4
No. of neurons in each layer	{500,400,300,200}

so that the output layer of the network has a dimension of 202. Another trick is to use linear values of the reflection and transmission coefficients for model training, which can obtain a higher accuracy since the linear values range from 0 to 1 and fluctuate less compared to their corresponding logarithmic values. The surrogate model has four hidden layers, each with 500, 400, 300, and 200 neurons. The first three hidden layers are followed by a ReLU as the activation function while the last hidden layer has no activation function. The data acquisition and training of the surrogate model are carried out in multiple sessions. This is because the training data comes from the interaction of RL with the CST. The surrogate model starts training once when RL has collected 100 samples. The total samples are split into training and test datasets in a ratio of 7:3, and the loss threshold is set to 0.003. When the training does not satisfy the requirements, the surrogate model continues to be trained using the data obtained from the interaction between RL and CST until the loss threshold is satisfied.

D. Algorithm Implementation

This section will introduce how to utilize the DQN algorithm to design the DCMS. It is necessary to transform the DCMS design into an RL problem by identifying the system state, action, corresponding reward, and next state.

1) State: Two terminologies are required to be defined: state-space S and state s_k. The state-space S can be represented by the parameters of the DCMS of all steps in one episode, which is given by

$$S = (s_1, s_2, ..., s_k, ...s_N) \tag{5}$$

where N stands for the total number of steps in one episode. We define the state s_k of the DCMS in step k as

$$s_k = (l_{mk}, w_{mk}, s_{xk}, s_{yk}, H_{1k}, l_{sk}, w_{sk}, l_k, d_k) \tag{6}$$

where l_{mk}, w_{mk} represent the length and width of the meta-unit, respectively. s_{xk}, s_{yk} are the meta-unit spacing of along x-axis and y-axis. H_{1k} stands for the height of DCMS with respect to the upper surface of the array antenna. l_{sk}, w_{sk}, l_k and d_k are the parameters used to tune the impedance matching of array antenna, which are the length, width, and spacing of rectangular slots, as well as the location of feeding point.

2) Action: For the design problem, there are three operations for each parameter: addition, subtraction, and in-variance. For the sake of simplicity, the amplitude of increase or decrease is set as the same, which is represented by amp. Therefore, the variations of each parameter p can be mathematically defined as $p^v \in \{amp, 0, -amp\}$. The setting of amp depends on the sensitivity of each structure parameter, which can be quickly obtained by parameter scanning. We find that small changes of the input variables l, l_m, w_m, l_s, w_s can have a more significant effect on the reflection and transmission coefficients compared to that of other structure parameters, so the amp of l, l_m, w_m, l_s, w_s are set to 0.2 while that of the other variables are set to 0.5. In such a condition, the action a^k at step k can be given by substituting the p in p^v with the parameters of DCMS:

$$a_k = \left(l_{mk}^v, w_{mk}^v, s_{xk}^v, s_{yk}^v, H_{1k}^v, l_{sk}^v, w_{sk}^v, l_k^v, d_k^v\right) \tag{7}$$

3) Reward: In the array antenna decoupling design, the isolation of the array antenna and VSWR are two essential parameters to evaluate the performance of DCMS design. Therefore, these two parameters are selected as the criterion for system evaluation of reward, which is given by

$$r\left(s_k, a_k\right) = \begin{cases} 1, & L_k < L_{k-1} \\ -1, & \text{otherwise} \end{cases} \tag{8}$$

$$L_k = \eta \times LS_{11k} + (1-\eta) \times LS_{21k} \tag{9}$$

$$\begin{aligned} LS_{11k} = &\left(\boldsymbol{S_{11k}} - \boldsymbol{S_{min}}\right)\left(\boldsymbol{S_{11k}} - \boldsymbol{S_{max}}\right)^{\mathrm{T}} \\ &+ \left|\left(\boldsymbol{S_{11k}} - \boldsymbol{S_{min}}\right)\left(\boldsymbol{S_{11k}} - \boldsymbol{S_{max}}\right)^{\mathrm{T}}\right| \end{aligned} \tag{10}$$

$$\begin{aligned} LS_{21k} = &\left(\boldsymbol{S_{21k}} - \boldsymbol{S_{min}}'\right)\left(\boldsymbol{S_{21k}} - \boldsymbol{S_{max}}'\right)^{\mathrm{T}} \\ &+ \left|\left(\boldsymbol{S_{21k}} - \boldsymbol{S_{min}}'\right)\left(\boldsymbol{S_{21k}} - \boldsymbol{S_{max}}'\right)^{\mathrm{T}}\right| \end{aligned} \tag{11}$$

where L_k indicates the difference between the results at the k time step and the target results. The formula for L_k is given in Equations (9) to (11). The η in Equation (9) is used to adjust the ratio between the reflection coefficient and the transmission coefficient. The closer η is to 1, the more attention is paid to the reflection coefficient, and the closer η is to 0, the more attention is paid to the transmission coefficient. In the design of array antenna decoupling, the transmission coefficients are more of interest, so the η is set to 0.2. In Equations (10) and (11), $\boldsymbol{S_{11k}}$, $\boldsymbol{S_{21k}}$ represent the reflection and transmission coefficients of DCMS samples at the k time step. $\boldsymbol{S_{min}}$, $\boldsymbol{S_{min}}'$ are the reflection and transmission coefficient's lower limits of the target DCMS. $\boldsymbol{S_{max}}$, $\boldsymbol{S_{max}}'$ indicate the reflection and transmission coefficient's upper limits of the target DCMS. Also, the boldface is denoted as a row vector, and $(*)^{\mathrm{T}}$ represents the transpose operator.

4) Next state: Based on the definition of state s_k and a_k demonstrated before, the next state s_{k+1} is represented by

$$s_{k+1} = s_k + a_k \tag{12}$$

Also, the termination conditions of the algorithm are given as follows:

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

TABLE II
THE PSEUDO CODE OF DQN ALGORITHM.

Algorithm: DQN-based DCMS Design

Initialization: Learning rate α, discount factor γ
ϵ-greedy ϵ, replay memory D, batch size B
estimation and target network weight θ and θ'
Process:
For episode = 1, 2, ..., E **do**
  Initialize state s_1
  For k = 1, 2, ..., K **do**
    With probability ϵ select a random action a^k
    With probability 1-ϵ select
      $a_k = \max_{a_{k+1}} Q(s_{k+1}, a_{k+1}; \theta_k)$
    Execute action a^k and obtain reward r^k
    Store transition (s_k, a_k, r_k, s_{k+1}) in D
    Sample random batch B of transitions from D
    If Criterion
      $r_{k+1} = r_k$
    else
      $r_{k+1} + \max_{a_{k+1}} Q(s_{k+1}, a_{k+1}; \theta_k)$
    Perform a gradient descent step on
    Every T_{step} steps reset $\theta' = \theta$
  End for
End for

TABLE III
THE DETAILED PARAMETERS OF DQN ALGORITHM.

Variable	Value
Learning rate α	0.001
Discount factor γ	0.9
ϵ-greedy ϵ	0.2
Replay memory D	2000
Batch size B	32
Maximum episode E	100
Maximum steps of each episode K	50

$$Criterion = \begin{cases} \text{Yes}, & L_k = 0 \\ \text{No}, & \text{otherwise} \end{cases} \tag{13}$$

The algorithm will end when it exceeds the allowable episode length or satisfy the termination condition. The pseudo-code of the DQN algorithm is shown in Table II, and the corresponding parameters of the DQN algorithm are listed in Table III.

IV. APPLICATION EXAMPLES

Two array antenna DCMSs are designed in this section to prove the effectiveness of the above-mentioned design concept. To reduce processing costs and deal with restricted computing resources, the array antennas employed in this study are 1×2 and 1×4 arrays, respectively. The array element is a patch antenna that is implemented on an F4B dielectric substrate with a permittivity of 2.65 and a thickness of 0.5 mm. While DCMS has greatly improved the isolation between antenna array elements, the matching of the antenna array deteriorates with the addition of the DCMS. Moreover, it is difficult to re-adjust the antenna array to impedance matching just by changing the size and the feed settings of the antenna array elements. Thus, we introduce two rectangular slots on the antenna array elements to add extra degrees of freedom for impedance tuning.

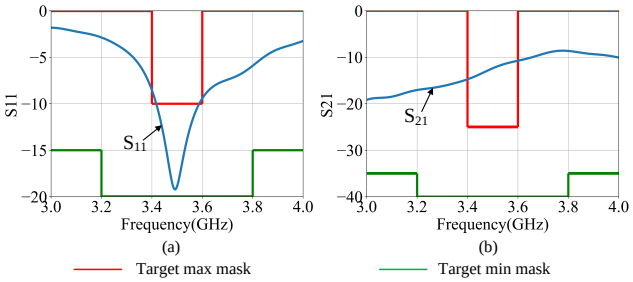

Fig. 5. Simulated reflection and transmission coefficients of the 1×2 array antenna without DCMS. (a) Reflection coefficients (S11). (b) Transmission coefficients (S21).

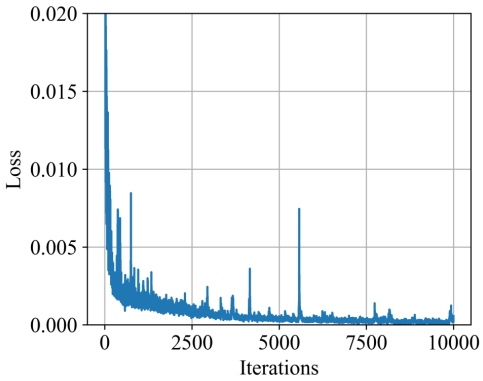

Fig. 6. Loss for the training data set during the training process.

A. 1×2 Array Antenna DCMS

Fig. 3 shows the geometry of 1×2 array antenna DCMS. As shown in Fig. 3(b), the spacing of antenna elements is 1.5 mm (0.017λ), where λ is the free-space wavelength corresponding to the center frequency. Fig. 5 shows the reflection and transmission coefficients of 1×2 array antenna without DCMS. We can find that the isolation between the antenna elements is around 10 dB within the operating frequency band, indicating that the mutual coupling between the antenna elements is serious at such a close distance. To suppress the mutual coupling, the DCMS is introduced over the array antenna to improve the isolation. Using the DQN algorithm, the system can gradually learn how to quickly find the parameters of DCMS to achieve the desired isolation while maintaining a good impedance matching. To accelerate the simulation of CST software, we build a surrogate model and train it to get the EM responses of array antenna DCMSs.

Fig. 6 shows the error between CST and its surrogate model over iterations. For the 1×2 array, we collect a total of 400 simulation samples in the first two stages of the proposed system framework in Fig. 4. The total time spent on data acquisition and model training is 40.3 hours. When the training is completed, the mean square error (MSE) is 1×10^{-3} for the training data set. Fig. 7 shows the interval probability of the test dataset under different MSEs for providing an overview of the performance of the surrogate model. We can find from the figure that the majority (80%) of the tested samples had an MSE of less than 0.004. The largest MSE of the test dataset

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

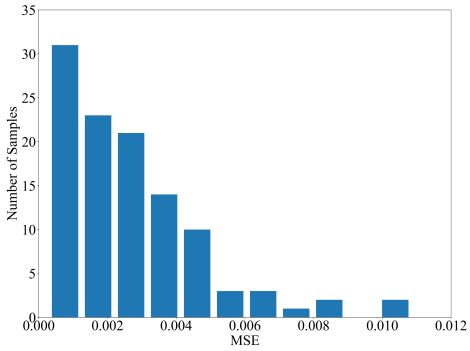

Fig. 7. The interval probability of test dataset under different mean squared errors (MSEs).

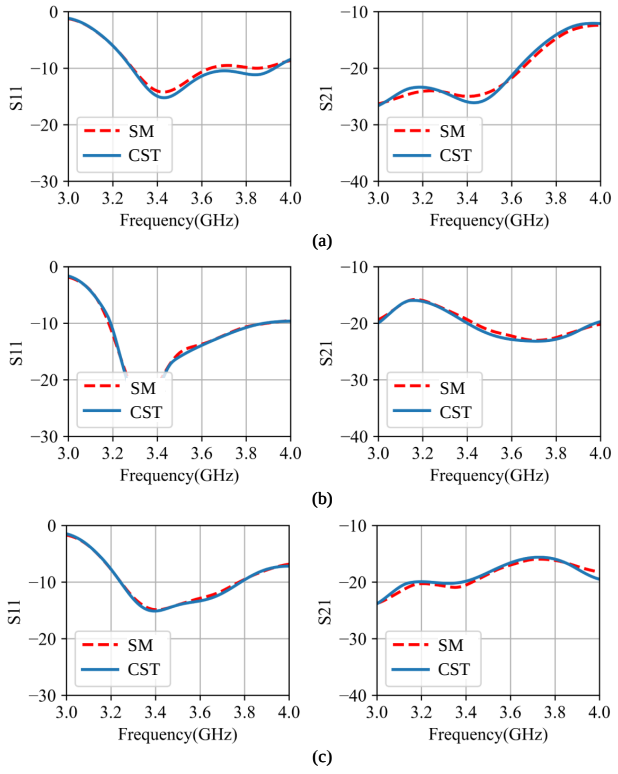

Fig. 8. Three examples are randomly sampled from the test data set. (a) Test example 1. (b) Test example 2. (c) Test example 3. SM: Surrogate Model.

is about 0.011 and its probability is only 2%. Moreover, the average MSE of the total test dataset is 0.003. We demonstrate the accuracy of the well-trained DNN with three examples that are randomly selected from the test data set. As shown in Fig. 8, the predicted scattering coefficients are labeled with the red dotted line, while the accurate results simulated with CST are labeled with the solid blue line. Excellent agreements have been achieved between the predicted and simulated results, which indicates that the surrogate model is well trained and can replace CST to give the EM responses of DCMS.

The returns of the DQN algorithm over episodes are depicted in Fig. 9. The reward value is normalized. It is observed that the return is relatively low at the beginning of the learning process. This is because the agent has no prior knowledge of

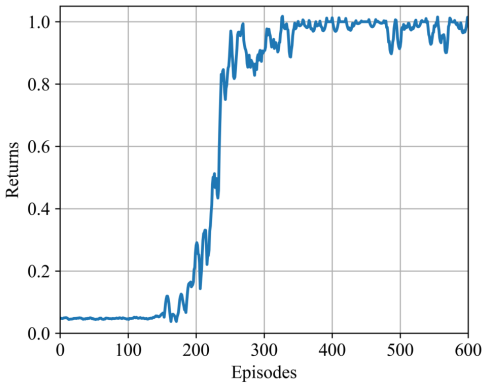

Fig. 9. Normalized returns of the DQN algorithm.

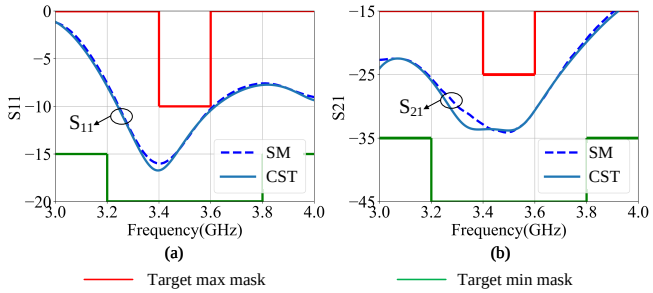

Fig. 10. Comparison between simulated results of CST and predicted results of surrogate model (SM). (a) Reflection coefficients (S11). (b) Transmission coefficients (S21).

the environment (DCMS) and takes action randomly. As the training episodes increase, the agent gradually learns the parameters of the system; therefore, the return increases greatly. We can also find some fluctuations on the curve, which can be explained by the ϵ-greedy selection strategy. Specifically, a small probability of random action should be taken to prevent the system from getting to a local minimum, leading to fluctuations. After about 350 episodes, the return fluctuates gently, indicating that the DQN algorithm has reached a convergence.

To achieve an ideal decoupling performance while maintaining a good impedance matching, it is generally required that the magnitudes (logarithm value) of transmission coefficients are less than -25 dB while return coefficients are less than -10 dB. Based on the constraints explained above, we define two sets of masks for the transmission and reflection coefficients. It is noticed that the upper mask (marked as red lines) plays an important role in controlling the transmission and return coefficients over the band of interest 3.4-3.6 GHz. Therefore, the design objective of transmission and return coefficients should focus more on satisfying the upper mask. The predicted results from the surrogate model and the simulated results from CST are shown in Fig. 10 for validation. It can be found that both results agree well and transmission coefficients of less than -25 dB with return coefficients of less than -10 dB over the working band of 3.4-3.6 GHz are achieved. The parameters of the designed DCMS are listed in Table IV.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

TABLE IV
DIMENSIONS OF THE DESIGNED 1×2 ARRAY ANTENNA DCMS.

Variable	Value	Variable	Value	Variable	Value
H_1	7.5	l_m	18	s_x	9
l_s	21.2	l	13	s_y	21
w_m	6	d	1	w_s	1.4

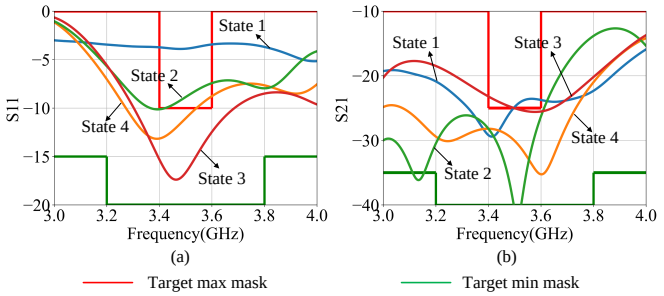

Fig. 11. The intermediate processes of how the RL tuned the structural parameters.

TABLE V
THE STRUCTURAL PARAMETERS OF THE INTERMEDIATE PROCESS FOR 1×2 ARRAY.

Variable	State 1	State 2	State 3	State 4
H_1	8.5	10	9	8
s_x	9.5	9	10	9.5
s_y	20.5	20.5	21	21
l_m	16.6	17	17.8	18.2
w_m	5.4	5.8	6.2	6
l_s	20.2	20.8	21	21.2
w_s	1.2	1.0	1.6	1.4
l	13.4	13.2	12.8	13
d	1	1.8	1.4	1.2

To better illustrate how the trained RL adjusts the structural parameters to achieve array decoupling, we provide four intermediate processes (from state 1 to state 4) from the surrogate model. The reflection and transmission coefficients are presented in Fig. 11, while their corresponding geometric parameters are presented in Table V. As shown in Fig. 11, state 1 is an arbitrarily chosen initial state. It has poor impedance matching but very low coupling between antenna elements. This is because most of the energy is reflected off and not radiated through the antenna, resulting in a very low coupling between the elements. From state 1 to state 2, the impedance matching and isolation are significantly improved. From state 2 to state 3, good impedance matching has been achieved but the coupling between antenna elements has deteriorated. In state 4, a compromise is made between the reflection and transmission parameters, allowing impedance matching and isolation to advance one step further from the target.

B. 1×4 Array Antenna DCMS

Following the verification of the validity and convergence of the DQN algorithm, we move on to the implementation of 1×4 array antenna DCMS. Fig. 12 shows the geometry of 1×4 array antenna DCMS. The spacing of antenna elements

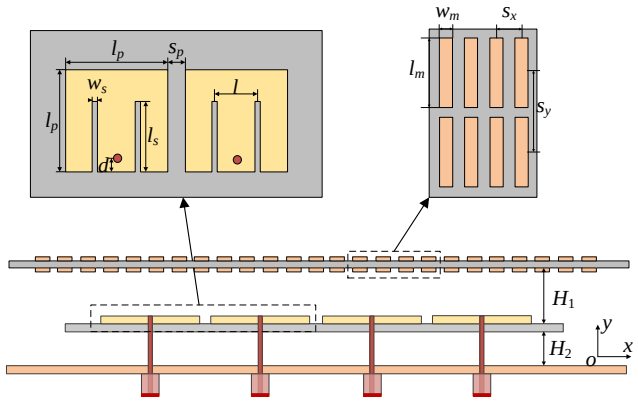

Fig. 12. Geometry of 1×4 array antenna and the corresponding DCMS.

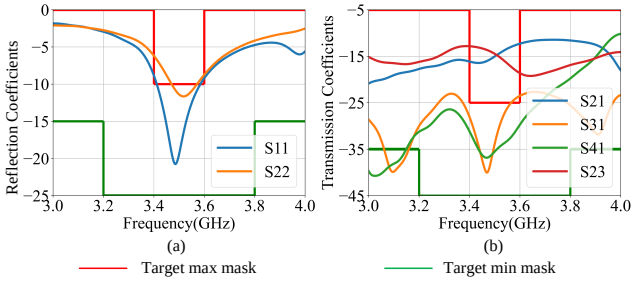

Fig. 13. Simulated reflection and transmission coefficients of the 1×4 array antenna without DCMS. (a) Reflection coefficients (S11, S22). (b) Transmission coefficients (S21, S31, S41, and S23).

remains the same at 0.017 λ. The reflection and transmission coefficients of the 1×4 array antenna elements without DCMS are given in Fig. 13. Considering the symmetry, we only give the scattering coefficients of S11, S22, S21, S31, S41, and S23. We can find that the 1×4 antenna array becomes more complex due to the number of antenna elements increasing compared to the 1×2 array antenna. The mutual coupling and the distance between antenna elements are closely related; the isolation of antenna elements adjacent to each other is around 15 dB while that of antenna elements farther away is more than 25 dB. We can also observe that mutual coupling has a great effect on impedance matching. As shown in Fig. 13(a), the antenna elements located at the edge of the array have a good impedance matching but the impedance matching of antenna elements in the middle of the array deteriorates, which can be explained by that the latter has a more complex coupling environment than the former. Therefore, we introduce two rectangular slots to increase the capacitance effect to adjust the impedance matching of the array antenna.

Fig. 14 shows the error between CST and its surrogate model over iterations. For the 1×4 array, we collect a total of 500 simulation samples in the first two stages of the proposed system framework in Fig. 4. The total time spent on data acquisition and model training is 67 hours. When the training is completed, the MSE is 2×10^{-3} for the training data set. The interval probability of the test dataset under different MSEs is depicted in Fig. 15 for providing an overview of

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

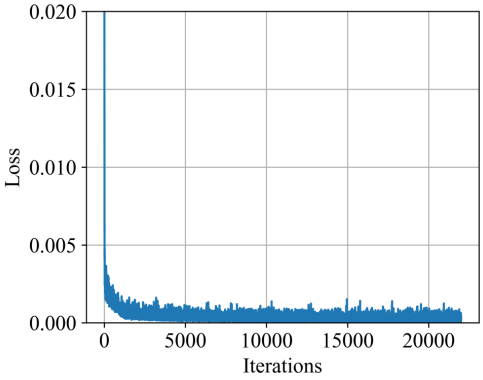

Fig. 14. Loss for the training data set during the training process.

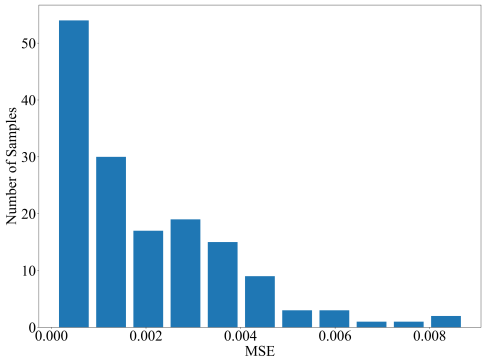

Fig. 15. The interval probability of test dataset under different mean squared errors (MSEs).

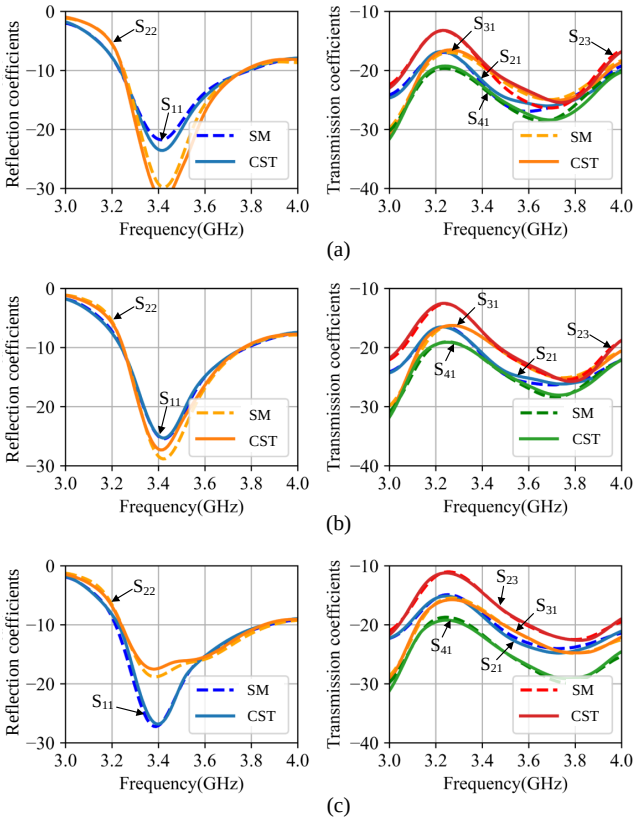

Fig. 16. Three examples are randomly sampled from the test data set. (a) Test example 1. (b) Test Example 2. (3) Test example 3. (Note: the predicted results from the surrogate model (SM) are represented by a dotted line while the real results from CST are represented by the solid line with the same color. Considering the symmetry, reflection coefficients (S11, S22), and transmission coefficients (S21, S32, S41, and S23) are given in the left and right figures, respectively.)

the performance of the surrogate model. It can be found from the figure that the samples with MSE less than 0.004 account for 89% of the overall samples. The largest MSE of the test sample is approximately 0.008 with a probability of 2%. The average MSE of the test dataset is 0.002. We illustrate the accuracy of the well-trained DNN with three examples that are randomly selected from the test data set. As shown in Fig. 16, the predicted scattering coefficients are labeled with the dotted lines, while the corresponding accurate results simulated with CST are labeled with the solid lines in the same color. Good agreements have been achieved between the predicted and simulated results, which indicates the surrogate model can still be well trained and accurately predict the EM response of DCMS even when the array antenna is complex.

The returns of the DQN algorithm over episodes are depicted in Fig. 17. Similarly, the return is relatively low at the beginning of the learning process and increases rapidly as the episodes increase. After about 500 episodes, the return fluctuates slightly, indicating that the DQN algorithm has reached a convergence. We use the well-trained DQN algorithm to search for the desired solution. To verify the rightness of the proposed method, the predicted scattering parameters and the simulated results are depicted in Fig. 18. We can find that an acceptable agreement is reached and transmission coefficients of less than -25 dB and return coefficients of less than -10 dB over the working frequency band of 3.4-3.6 GHz are achieved. The parameters of the designed DCMS are listed in Table VI.

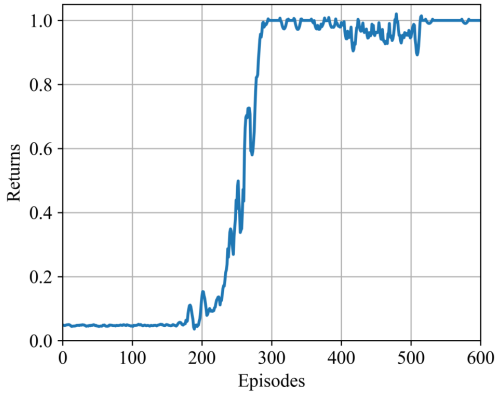

Fig. 17. Normalized returns of the DQN algorithm.

V. Results and Discussion

A. Results

To demonstrate the advantages of the proposed approach, Table VII compares the three DCMS design methodologies. The GPU is used only for neural networks training not used for CST in this study and has already achieved good performance. Superior performance will be obtainable if the GPU can be used for CST with some firmware setting. From

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2022.3221613

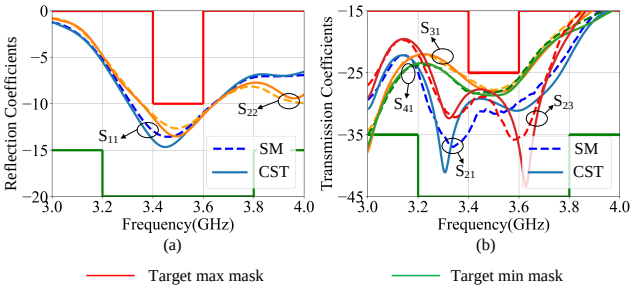

Fig. 18. The comparison between simulated results of CST and predicted results of surrogate model (SM). (a) Reflection coefficients (S11, S22). (b) Transmission coefficients (S21, S31, S41, and S23).

the perspective of data acquisition, our proposed method (RL + Surrogate Model) can achieve satisfactory training results using fewer samples than the second method (GA + Surrogate Model), which stems from the different training mechanisms of the two methods. Specifically, the second method requires data collection in advance to train the surrogate model. The brute data acquisition method of parameter scanning used in the second method may lead to a large number of invalid samples in the training dataset. Our proposed approach, on the other hand, alternates between collecting data and training two neural networks (surrogate model and RL algorithm). In other words, the RL algorithm first interacts with the environment to collect a portion of EM simulation data and then uses this portion of data to train both RL and the surrogate model, with the aim of improving the decision-making ability of RL and the prediction ability of the surrogate model. Next, the trained RL continues to interact with the environment to collect data, and the cycle continues. In such a mechanism, the quality (relevance to the target) of the collected data will be higher and good performance can be obtained using less data. Moreover, the training time is longer since both the RL algorithm and the surrogate model are needed to train, but the optimization time is shorter owing to the excellent decision-making capability obtained by the trained RL.

Compared to the trial-and-error method, the design efficiency of the latter two methods is higher, which mainly stems from two aspects: on the one hand, the well-trained DNN surrogate model allows for the rapid derivation of the scattering response of EM structures, which greatly improves CST simulation speed. On the other hand, the use of intelligent algorithms (GA, DQN) considerably improves the efficiency of exploring the optimal solution. In addition, the proposed design method achieves a fully automatic design of the DCMS. The trial-and-error method relies entirely on the designer's experience and a large number of simulation iterations, while the second method (GA + DNN) also requires the designer to artificially provide data for the DNN training. In contrast, the proposed method takes advantage of the decision-making capability of the RL algorithm to obtain increasingly better training data without human involvement. Therefore, the proposed method is an efficient and fully automated design method.

TABLE VI
DIMENSIONS OF THE DESIGNED 1×4 ARRAY ANTENNA DCMS.

Variable	Value	Variable	Value	Variable	Value
H_1	7.5	w_m	3	s_x	6
l_s	21.2	l	12.6	s_y	19
l_m	16.8	d	1	w_s	1.8

TABLE VII
PERFORMANCE COMPARISON OF THREE METHODS FOR 1 × 2 ARRAY DECOUPLING.

Methods	Trail-and Error	GA + Surrogate Model	**This work**
Computational Resources	CPU	CPU, GPU	**CPU, GPU**
Data Acquisition	/	600 samples, 60 h	**400 samples, 40 h**
Model Training	/	3 mins	**20 mins**
Optimization	168 h	0.5 h	**5 mins**
Total Time	168 h	60.5 h	**40.4 h**
Automation Level	Manual	Semi-Automation	**Full-Automation**

B. Discussion

It should be noted, that the advantage of the fully automated process allows easy implementation of the proposed methods to different EM designs without time-consuming simulation and manual annotation for big amount of straining data used in most existing supervised deep learning design methods.

Some limitations of the current method are pointed out. Since the DQN algorithm has to discretize the action space, it is computationally intensive when solving high-dimensional or continuous action space problems, and the discretization may lose some important action information. This problem can be alleviated by using more advanced RL algorithms, such as deep deterministic policy gradient (DDPG) [53], proximal policy optimization (PPO) [54], etc. In addition, the neural network structure of the surrogate model used in this method is fixed. Such an architecture may not yield optimal performance when dealing with different problems and different sizes of data. It may be a promising solution by using evolutionary algorithms to optimize the structure of the surrogate model [55] for improving its generalization ability.

C. Future Work

In this paper, we have chosen a double-layer rectangular patch as a metamaterial unit, which is not "invented" by the machine to decouple the microstrip patch antenna array. For some other forms of array antennas, this metamaterial unit may not be a suitable choice. But there is plenty of proven metamaterial unit forms available and a unit library can be built for the RL agent to choose from using a simple classification network currently under development. In future work, we will upgrade

the RL algorithm to explore unknown metamaterial units for decoupling different array antenna types. To achieve this goal, we plan to include a VAE network, based on our previous work to generate different metamaterial units [29], and then combine the proposed method to achieve the exploration of metamaterial units to further improve the design efficiency of DCMS.

VI. Conclusion

In this paper, we proposed an RL-based automation design method for decoupling array antennas. Through the interaction of the RL algorithm and the surrogate models, the proposed method can find satisfactory solutions efficiently. Compared with classic trial-and-error and supervised learning methods, the proposed method makes full use of the decision-making capability of RL derived from data analysis, which considerably enhances the quality of the training data set and accelerates the convergence of the surrogate model. Meanwhile, it also eliminates the manual training data preparation needed by supervised learning, achieving the automation design of array antenna decoupling. Two array decoupling metasurface examples have been given to demonstrate the feasibility of the proposed method. This design concept paves the way for the fully automated design of EM components and systems.

References

[1] K.-L. Wu, C. Wei, X. Mei, and Z.-Y. Zhang, "Array-antenna decoupling surface," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6728–6738, 2017.

[2] F. Liu, J. Guo, L. Zhao, G.-L. Huang, Y. Li, and Y. Yin, "Dual-band metasurface-based decoupling method for two closely packed dual-band antennas," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 1, pp. 552–557, 2019.

[3] Y. Zhu, Y. Chen, and S. Yang, "Decoupling and low-profile design of dual-band dual-polarized base station antennas using frequency-selective surface," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 8, pp. 5272–5281, 2019.

[4] M. Li, B. G. Zhong, and S. Cheung, "Isolation enhancement for mimo patch antennas using near-field resonators as coupling-mode transducers," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 2, pp. 755–764, 2018.

[5] T. Weiland, M. Timm, and I. Munteanu, "A practical guide to 3-d simulation," *IEEE Microwave Magazine*, vol. 9, no. 6, pp. 62–75, 2008.

[6] W.-L. Guo, G.-M. Wang, W.-Y. Ji, Y.-L. Zheng, K. Chen, and Y. Feng, "Broadband spin-decoupled metasurface for dual-circularly polarized reflector antenna design," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 5, pp. 3534–3543, 2020.

[7] H. M. Bernety, A. B. Yakovlev, H. G. Skinner, S.-Y. Suh, and A. Alù, "Decoupling and cloaking of interleaved phased antenna arrays using elliptical metasurfaces," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 6, pp. 4997–5002, 2019.

[8] A. Hurshkainen, M. S. M. Mollaei, M. Dubois, S. Kurdjumov, R. Abdeddaim, S. Enoch, S. Glybovski, and C. Simovski, "Decoupling of closely spaced dipole antennas for ultrahigh field mri with metasurfaces," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 2, pp. 1094–1106, 2020.

[9] J. Guo, F. Liu, L. Zhao, G.-L. Huang, W. Lin, and Y. Yin, "Partial reflective decoupling superstrate for dual-polarized antennas application considering power combining effects," *IEEE Transactions on Antennas and Propagation*, 2022.

[10] P. Burrascano, S. Fiori, and M. Mongiardo, "A review of artificial neural networks applications in microwave computer-aided design (invited article)," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 9, no. 3, pp. 158–174, 1999.

[11] J. E. Rayas-Sánchez, "Em-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 1, pp. 420–435, 2004.

[12] V. Rizzoli, A. Costanzo, D. Masotti, A. Lipparini, and F. Mastri, "Computer-aided optimization of nonlinear microwave circuits with the aid of electromagnetic simulation," *IEEE transactions on microwave theory and techniques*, vol. 52, no. 1, pp. 362–377, 2004.

[13] V. K. Devabhaktuni, C. Xi, F. Wang, and Q.-J. Zhang, "Robust training of microwave neural models," *International Journal of RF and Microwave Computer-Aided Engineering: Co-sponsored by the Center for Advanced Manufacturing and Packaging of Microwave, Optical, and Digital Electronics (CAMPmode) at the University of Colorado at Boulder*, vol. 12, no. 1, pp. 109–124, 2002.

[14] V. K. Devabhaktuni, M. C. Yagoub, and Q.-J. Zhang, "A robust algorithm for automatic development of neural-network models for microwave applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 12, pp. 2282–2291, 2001.

[15] L.-Y. Xiao, W. Shao, F.-L. Jin, and B.-Z. Wang, "Multiparameter modeling with ann for antenna design," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 7, pp. 3718–3723, 2018.

[16] H. Aliakbari, A. Abdipour, A. Costanzo, D. Masotti, R. Mirzavand, and P. Mousavi, "Ann-based design of a versatile millimetre-wave slotted patch multi-antenna configuration for 5g scenarios," *IET Microwaves, Antennas & Propagation*, vol. 11, no. 9, pp. 1288–1295, 2017.

[17] Y. Sharma, X. Chen, J. Wu, Q. Zhou, H. H. Zhang, and H. Xin, "Machine learning methods-based modeling and optimization of 3-d-printed dielectrics around monopole antenna," *IEEE Transactions on Antennas and Propagation*, 2022.

[18] Y.-f. Liu, L. Peng, and W. Shao, "An efficient knowledge-based artificial neural network for the design of circularly polarized 3d-printed lens antenna," *IEEE Transactions on Antennas and Propagation*, 2022.

[19] A. Patnaik, B. Choudhury, P. Pradhan, R. Mishra, and C. Christodoulou, "An ann application for fault finding in antenna arrays," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 775–777, 2007.

[20] J. Huang, W. Li, Y. He, L. Zhang, and S.-W. Wong, "Optimization of antenna design using the artificial neural network and the simulated annealing algorithm," in *2021 Computing, Communications and IoT Applications (ComComAp)*. IEEE, 2021, pp. 119–122.

[21] S. D. Campbell, R. P. Jenkins, P. J. O'Connor, and D. Werner, "The explosion of artificial intelligence in antennas and propagation: How deep learning is advancing our state of the art," *IEEE Antennas and Propagation Magazine*, vol. 63, no. 3, pp. 16–27, 2020.

[22] A. Massa, D. Marcantonio, X. Chen, M. Li, and M. Salucci, "Dnns as applied to electromagnetics, antennas, and propagation—a review," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 11, pp. 2225–2229, 2019.

[23] Y. Zhou, J. Xie, Q. Ren, H. H. Zhang, and Q. H. Liu, "Fast multi-physics simulation of microwave filters via deep hybrid neural network," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 5165–5178, 2022.

[24] P. Liu, L. Chen, and Z. N. Chen, "Prior-knowledge-guided deep-learning-enabled synthesis for broadband and large phase shift range metacells in metalens antenna," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 5024–5034, 2022.

[25] L. Li, H. Ruan, C. Liu, Y. Li, Y. Shuang, A. Alù, C.-W. Qiu, and T. J. Cui, "Machine-learning reprogrammable metasurface imager," *Nature communications*, vol. 10, no. 1, pp. 1–8, 2019.

[26] X. Shi, T. Qiu, J. Wang, X. Zhao, and S. Qu, "Metasurface inverse design using machine learning approaches," *Journal of Physics D: Applied Physics*, vol. 53, no. 27, p. 275105, 2020.

[27] S. An, B. Zheng, M. Y. Shalaginov, H. Tang, H. Li, L. Zhou, J. Ding, A. M. Agarwal, C. Rivero-Baleine, M. Kang *et al.*, "Deep learning modeling approach for metasurfaces with high degrees of freedom," *Optics Express*, vol. 28, no. 21, pp. 31 932–31 942, 2020.

[28] F. Ghorbani, S. Beyraghi, J. Shabanpour, H. Oraizi, H. Soleimani, and M. Soleimani, "Deep neural network-based automatic metasurface design with a wide frequency range," *Scientific Reports*, vol. 11, no. 1, pp. 1–8, 2021.

[29] Z. Wei, Z. Zhou, P. Wang, J. Ren, Y. Yin, G. F. Pedersen, and M. Shen, "Equivalent circuit theory-assisted deep learning for accelerated generative design of metasurfaces," *IEEE Transactions on Antennas and Propagation*, 2022.

[30] E. Zhu, Z. Wei, X. Xu, and W.-Y. Yin, "Fourier subspace-based deep learning method for inverse design of frequency selective surface," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 5130–5143, 2021.

[31] Y. Xiao, K. W. Leung, K. Lu, and C.-S. Leung, "Mode recognition of rectangular dielectric resonator antenna using artificial neural network," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 5209–5216, 2022.

[32] G. Oliveri, M. Salucci, and A. Massa, "Towards efficient reflectarray digital twins-an em-driven machine learning perspective," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 5078–5093, 2022.

[33] W. Chen, Q. Wu, C. Yu, H. Wang, and W. Hong, "Multibranch machine learning-assisted optimization and its application to antenna design," *IEEE Transactions on Antennas and Propagation*, 2022.

[34] J. A. Easum, N. Jogender, L. W. Pingjuan, and H. W. Douglas, "Efficient multiobjective antenna optimization with tolerance analysis through the use of surrogate models," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 6706–6715, 2018.

[35] A. Massa and M. Salucci, "On the design of complex em devices and systems through the system-by-design paradigm: A framework for dealing with the computational complexity," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 2, pp. 1328–1343, 2021.

[36] M. Salucci, L. Tenuti, G. Gottardi, A. Hannan, and A. Massa, "System-by-design method for efficient linear array miniaturisation through low-complexity isotropic lenses," *Electronics Letters*, vol. 55, no. 8, pp. 433–434, 2019.

[37] M. Salucci, G. Oliveri, M. A. Hannan, and A. Massa, "System-by-design paradigm-based synthesis of complex systems: The case of spline-contoured 3d radomes," *IEEE Antennas and Propagation Magazine*, vol. 64, no. 1, pp. 72–83, 2021.

[38] G. Oliveri, A. Gelmini, A. Polo, N. Anselmi, and A. Massa, "System-by-design multiscale synthesis of task-oriented reflectarrays," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 4, pp. 2867–2882, 2019.

[39] M. Salucci, N. Anselmi, S. Goudos, and A. Massa, "Fast design of multiband fractal antennas through a system-by-design approach for nb-iot applications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–15, 2019.

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

[41] 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 Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, 2020.

[42] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2018.

[43] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019.

[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[45] L. Wu, F. Tian, T. Qin, J. Lai, and T.-Y. Liu, "A study of reinforcement learning for neural machine translation," *arXiv preprint arXiv:1808.08866*, 2018.

[46] B. Zhang, C. Jin, K. Cao, Q. Lv, and R. Mittra, "Cognitive conformal antenna array exploiting deep reinforcement learning method," *IEEE Transactions on Antennas and Propagation*, 2021.

[47] I. Sajedian, H. Lee, and J. Rho, "Double-deep q-learning to increase the efficiency of metasurface holograms," *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[48] J. Hu, H. Zhang, K. Bian, M. Di Renzo, Z. Han, and L. Song, "Metasensing: Intelligent metasurface assisted rf 3d sensing by deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2182–2197, 2021.

[49] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[50] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[51] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi *et al.*, "A graph placement methodology for fast chip design," *Nature*, vol. 594, no. 7862, pp. 207–212, 2021.

[52] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 287–308, 2000.

[53] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*. PMLR, 2014, pp. 387–395.

[54] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[55] Y. Tsukamoto and A. Namatame, "Evolving neural network models," in *Proceedings of IEEE International Conference on Evolutionary Computation*. IEEE, 1996, pp. 689–693.