Phase-Only Pattern Synthesis of Domino Tiling Irregular Arrays Based on Deep Reinforcement Learning and Gradient-descent Optimization

Yihan Ma, Yibing Guo, Qi Luo, *Senior Member, IEEE*, Yonggang Zhou, *Member, IEEE*, Guangwei Yang, *Member, IEEE*, and Steven Gao, *Fellow, IEEE*

***Abstract*—The irregular subarray design technique is gaining increasing attention for its excellent compromise between cost-effective design and good beamforming performance. In this article, a novel framework that incorporates a deep reinforcement learning (DRL) technique with a gradient-descent optimization method for phase-only pattern synthesis of domino-shaped irregular arrays is proposed. By employing the deep Q-network (DQN) technique to tile domino-shaped subarrays into an array aperture, the exact phase distribution over the aperture can be derived. Furthermore, to get better performance of the scanning radiation patterns as well as improve the optimization efficiency, a novel phase-only gradient-descent optimization method is integrated with the DRL training procedure to effectively evaluate the pattern synthesis performance of the tiling configuration. Through the proposed framework by combining DQN and phase-only gradient-descent iterative algorithm, the optimal configuration for the target radiation pattern can be efficiently optimized since the subarray tiling and optimization can be realized simultaneously. The good performance of the proposed framework is demonstrated in several case studies of 16×16-dimensional domino-shaped arrays. Moreover, full-wave simulations are used to verify the calculations to demonstrate the practical reliability of the developed method.**

***Index Terms*—Sub-arrayed tiling, sidelobe suppression, pattern synthesis, gradient-descent.**

This work was supported by the Smart Networks and Services Joint Undertaking (SNS JU) project TERRAMETA under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101097101, including top-up funding by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee. (Yihan Ma and Yibing Guo are co-first authors.) (*Corresponding author: Guangwei Yang.*)

Yihan Ma, and Qi Luo are with the School of Physics, Engineering and Computer Science, University of Hertfordshire, Hertfordshire, AL10 9AB, UK (email: nwpu.mayihan@gmail.com; qiluo@ieee.org).

Guangwei Yang is with School of Electronics and Information, Northwestern Polytechnical University, Xi'an Shaanxi,710129, P.R.China.(E-mail:guangwei.yang@nwpu.edu.cn).

Yonggang Zhou is with the School of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: zyg405@nuaa.edu.cn).

Yibing Guo is with School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, UK (e-mail: yibingguo890@gmail.com).

Steven Gao is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong.

I. Introduction

PHASE array antenna (PAA) is widely used in state-of-the-art telecommunication, dynamic holograms, radar technologies, and many other fields due to its excellent reliable and fast response beam manipulation ability. To guarantee effective propagation distance or spital resolution over the air, researchers have shown an increased interest in designing phase arrays with a larger number of individual antenna elements at higher frequencies [1]-[3]. Nevertheless, the densely packed T/R (Transmitter/Receiver) modules can lead to thermal deformation on the array aperture and subsequently deteriorate array transmission efficiency [4], [5], particularly when the dimension of the T/R module approaches or exceeds the size of an individual element. Furthermore, the high implementation cost of T/R modules also constitutes a significant portion of the overall system expenses. Therefore, it is highly desirable to reduce the number of T/R modules while maintaining the beamforming performance in the communication systems [6].

In the last few decades, there has been an increasing interest in the investigation of various unconventional array antenna architectures, including thinned arrays [7]-[11], sparse arrays [8]-[14], and aperiodic tiling arrays [17]-[18], to provide a better trade-off between cost and performance. The thinned array technology enables amplitude modulation [7] or complex amplitude modulation [8], [9] to suppress sidelobes of patterns by selectively nulling the elements in the periodic array through the use of heuristic optimization or convex optimization techniques. However, simultaneously manipulating the amplitude and phase of antenna elements is a very challenging task with computation complexity in some scenarios compared to the phase-only array which is more systematic [10], [11]. Another alternative approach is the sparse array technology, which fills antenna elements with a large spatial distance in an array aperture. This technology is capable of cancelling unnecessary grating lobes as well as reducing the impact of mutual coupling. By tuning the adjacent distances between elements, researchers can realize high-gain beam steering angular ranging from -70° to 70° with peak sidelobe level (SLL) lower than -15 dB [12]. Instead of straightforwardly optimizing inter-element spacing in the sparse array, hyperuniform disordered distribution provides an analytical solution to the

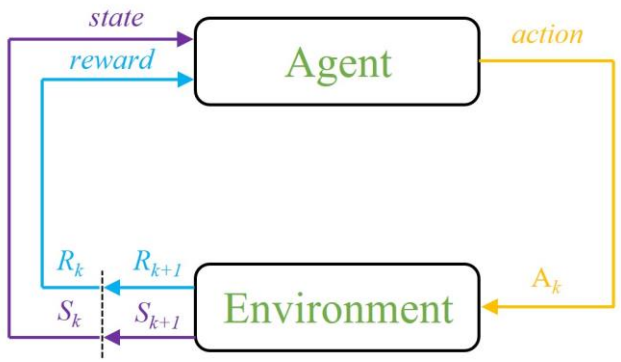

Fig. 1. A generic RL framework with agent-environment interaction.

array design with excellent scanning performance [13], [14]. A reconfigurable reflectarray based on hyperuniform disordered distribution to achieve ±50° beam steering less than -7 dB SLLs is reported [14]. Although the previous sparse array is able to suppress SLL and achieve good beam steering performance, the resulted array antenna requires complicated feeding networks due to the distribution of the array elements [15], which also lead to the reduction in aperture efficiency [16]. In addition to sparse arrays and thinned arrays, aperiodic array methods such as the strip-projection method [17] can reduce the number of elements by generating an aperiodic lattice through Rodrigue's formula while maintaining a low SLL within 30° scanning range. Penrose subarray tiling on aperiodic arrays can ensure a remarkable reduction in T/R modules across a wide scan angle [18]. However, the irregular number of elements in Penrose subarrays also increases the design complexity of the control circuits.

The traditional subarray technology can simplify the fabrication process while maintaining high aperture efficiency. However, the limited angular scan range significantly hinders the application of this technology due to the large array spacing [19]. Alternatively, irregular but exact tilling subarray technology is proposed to strike a balance between array scanning performance and its feeding network complexity [20]-[22]. To fully cover the whole aperture without any gaps or holes, many advanced algorithms, such as Algorithm X [22], convex relaxation iteration programming (ICRP) [23], and weighted L1 norm iterative convex (WL1X) [24], have been implemented to accelerate the tiling speed. Based on the traversed search for tiling configurations, the local optimal solution can then be identified. However, both nonconvex optimization and convex optimization methods significantly increase the computational cost and time consumption when traversal searching for large-scale arrays [25]. Except for the subarray configuration optimization, some advanced antenna pattern synthesis algorithms, such as iterative Fourier transform [26], iterative gradient descent [27][28], and convex optimization [29], are investigated, aiming for the goal of cost-effective and accurate beamforming performance. However, combining pattern synthesis algorithms with heuristic optimization methods to find optimal tiling configurations becomes highly inefficient when dealing with a large number of tiling solutions.

Recently, nonlinear electromagnetic (EM) problems have been successfully addressed thanks to the emergence of deep learning (DL) techniques. Examples include EM inverse scattering [30], the direction of arrival (DoA) estimation [31], remote object recognition [32], and computational electromagnetic (CEM) [33]. As one of the most rapidly developing branches of DL, deep reinforcement learning (DRL) has achieved breakthroughs in many fields such as gaming [34], self-driving [35], and healthcare [36]. Furthermore, more recently, DRL has gradually proven to be a highly effective and versatile framework for optimizing and exploiting phase antenna arrays. For instance, a framework based on deep Q-network (DQN) is employed to control the phase information of a conformal phased antenna array, enabling timely beam steering [37]. Also, the DQN-based agent can interact with numerical solvers to realize automated design for antenna array decoupling [38].

Inspired by previous pioneers' work, we are motivated in this work to incorporate the DRL algorithm and gradient-descent optimization approach to the problem of tiling irregular phased arrays for desired radiation patterns. The proposed work aims to present a highly integrated and cost-effective framework for the pattern synthesis of irregular arrays, aimed to maximize gain in the target region and simultaneously minimize the peak sidelobe level (PSLL). To the best of our knowledge, this is the first time the DRL technique has been exploited in the design of pattern synthesis of exactly tiled irregular pattern arrays.

The article is organized as follows: Section II provides an overview of the Deep Q-Network (DQN) and describes the procedure for irregular array partitioning using the DQN architecture. Additionally, this section introduces the phase-only gradient-descent optimization algorithm for pattern synthesis with subarray configurations. Then, the performance of the proposed framework is verified through examples of 16×16-dimensional domino-shaped subarray configurations with pencil pattern synthesis and flat-topped pattern synthesis in Section III. Finally, the conclusion is presented in Section IV.

II. Preliminary

A. Overview of Deep Q-Network

Fig. 1 presents the schematic diagram of RL. It is about learning the optimal action in the environment to obtain maximum reward. Similar to how children learn from their surroundings by experimenting and observing the outcomes, this optimal behavior is acquired through engaging with the environment and noting its reactions [39].

At each time k, firstly, the agent receives the current state S_k and reward R_k. Secondly, it then chooses an action A_k forward to the environment according to the optimal decision policy $\pi_k(a|s)$. Finally, the environment updates the next environment state S_{k+1} and rewards R_{k+1} from the current action A_k. This iteration repeats until termination conditions are satisfied. In general, the agent's goal is to maximize the cumulative reward in a loop by carefully selecting the action in each step.

Q-learning is a model-free, value-based, off-policy RL algorithm that will find the optimal series of actions to take

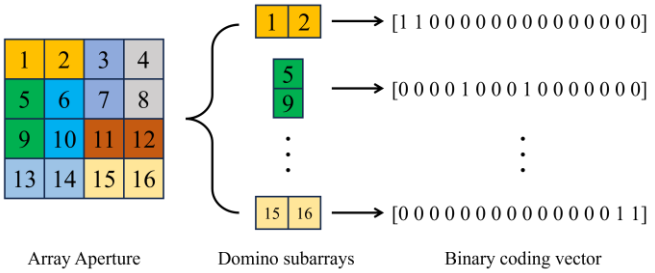

Fig. 2. Example of a 4×4 array with domino subarray tiling layout.

given the current state. It is based on the concept of learning the value of action-state pairs, known as Q-values. It represents the expected utility (total reward) of taking a particular action in a particular state and following the optimal policy thereafter [39]. The Q-value function under decision policy π can be expressed as:

$$\mathrm{Q}^{\pi}(S_k, \mathrm{A}_k) = \mathbf{E}[R_{k+1} + \gamma R_{k+1} + \gamma^2 R_{k+1} + \cdots | S_k, \mathrm{A}_k] \quad (1)$$

where **E** denotes the expectation and γ is the discount factor, which ranges from 0 to 1 to guarantee the sum converges. To update the Q-value efficiently, the Bellman equation is employed to explore the Q-value function as follows:

$$\mathrm{Q}^{new}(S_k, \mathrm{A}_k) = (1-\alpha)\mathrm{Q}^{\pi}(S_k, \mathrm{A}_k) + \alpha(R_{k+1} + \gamma \max_{\mathrm{A}} \mathrm{Q}(S_{k+1}, \mathrm{A})) \quad (2)$$

where $\alpha \mathrm{R}_{k+1}$ is the reward in k+1 step, $(1-\alpha)\mathrm{Q}(\mathrm{S}_k, \mathrm{A}_k)$ represents the current value, $\alpha\gamma \max_{\mathrm{A}} \mathrm{Q}(\mathrm{S}_{k+1}, \mathrm{A})$ defines the maximum discount reward in the state S_{k+1}.

Theoretically, Q-learning can find the optimal actions in a given environment by iteratively updating Q-values for state-action pairs. However, in practice, optimizing the decision policy $\pi_k(\mathrm{a}|\mathrm{s})$ in traditional Q-learning algorithms is extremely challenging, especially in an environment with large state and action spaces. Deep Q-learning overcomes this limitation by employing a deep neural network, known as a Deep Q-Network (DQN), to approximate the action-value function [40]. The DQN takes the state as input and outputs the Q-values for all possible actions in that state, effectively generalizing the learning process across high-dimensional and continuous state spaces. In addition, the experience replay technique employed in DQN breaks the correlation between consecutive samples leading to more stable and efficient learning.

To ensure a good estimate of the optimal Q-value during the training, two neural networks, the estimation network and target network, are used to approximate estimated Q-values $\mathrm{Q}(\mathrm{S}, \mathrm{A}; \theta)$ and target Q-values $\mathrm{Q}(\mathrm{S}, \mathrm{A}; \theta')$. By minimizing the temporal difference error between the $\mathrm{Q}(\mathrm{S}_k, \mathrm{A}_k; \theta)$ and the $\mathrm{Q}(\mathrm{S}_{k+1}, \mathrm{A}_{k+1}; \theta')$, the Q-network is able to learn the expected rewards more accurately over time, aligning the Q-network's predictions with the actual returns received from the environment. The loss function $L(\theta)$ can be expressed as:

$$L(S_k, \mathrm{A}_k, S_{k+1}|\theta) = (\mathrm{r} + \gamma \max_{\mathrm{A}} \mathrm{Q}(S_{k+1}, \mathrm{A}_{k+1}; \theta') - \mathrm{Q}(S_k, \mathrm{A}_k; \theta))^2 \quad (3)$$

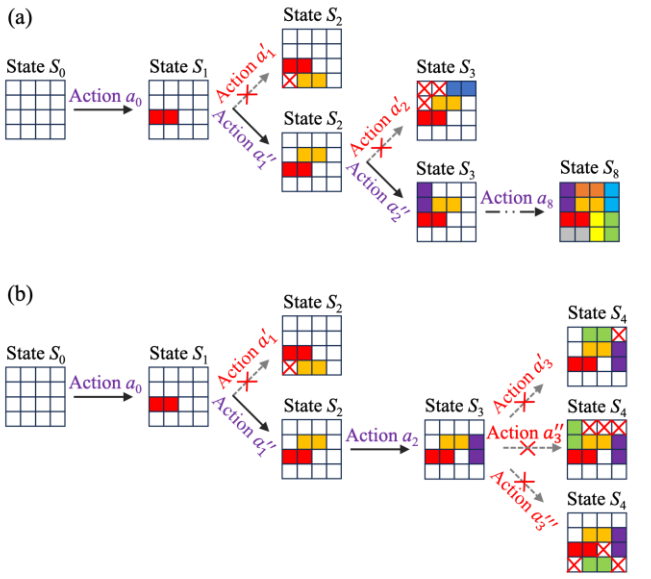

Fig. 3. Examples of action restriction in 4×4 array. (a) Successful exact tiling; (b) Failure tiling.

where r is the reward at step k, and the weights and biases in the target network are updated periodically to allow the Q-network to converge more reliably.

B. Implementation

To create the environment for solving polynomino subarray tiling problems, we first need to streamline the data from the procedure of subarray tiling. In the present problem, successfully tiling the phased array while finding the optimal configuration for the expected radiation pattern is the primary objective. An example of a 4×4-dimensional array with a domino-shaped subarray exact tiling configuration is shown in Fig. 2. Horizontal or vertical domino-shaped subarray configurations are employed to tile the array. Each domino subarray can be represented by a binary coding vector, denoted as $e_{1\times16}$, wherein this vector contains two positions as 1 and the rest as 0. For instance, one of the domino subarrays (5, 9) presented in Fig. 2 is encoded as $e_{1\times16}$ = [0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0], which represents that the fifth and ninth elements in the array are grouped as a subarray. All possible candidate subarrays then can be expressed as $\mathbf{A} = \{\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_{23}, \mathrm{A}_{24}\}$, where a_k is a binary coding vector $e_{1\times16}$. If a subarray layout $\mathrm{L} = \{\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \ldots, \mathrm{A}_8\}_{8\times16}$ of size 8×16, selected from **A**, has exactly one element in each column is 1 and the others are 0, this layout is exactly tiled.

Therefore, the environment can be regarded as the tiling state of the 4×4 array grid, and the actions corresponding to filling subarrays from candidate subarray set **A**. At the beginning of each episode, the state matrix S_0 is an empty 4×4 matrix. A random action A_k is then selected to fill the state space and generate the next state S_1. This action-state loop continues until the termination conditions are met. The ε-greedy policy is adopted to explore more situations instead of exploiting what we have learned so far, which can be defined as:

$$A_k = \begin{cases} \text{argmax} Q^{\pi}(S_k, A_k), \text{with probability } \varepsilon \\ \text{random action, with probability } 1-\varepsilon \end{cases} \quad (4)$$

In addition, in contrast to conventional application scenarios of the DQN algorithm, the polyomino exact tiling problem with a given finite set of domino-shaped subarrays involves too many illegal actions, particularly when most of the area is occupied by the subarrays. Therefore, it is necessary to add some action restrictions during the training steps to accelerate convergence and make it easier for the model to learn the polynomial tiling problems. Two action constraint criteria are proposed to improve the training performance of DQN. First, a conventional action mask is employed to filter out impossible or unavailable actions. Specifically, for a given array, instead of selecting an action from all possible actions candidate subarray set **A**, an action mask is applied to exclude illegal actions that cause overlapping with other subarrays in the state matrix. The application of an action mask can guarantee the selection of every time step in the available action space.

Second, the terminal state condition is considered in the action constraint. The strategy for action restriction is that if a closed empty grid or three consecutive closed empty grids appear following an action, that action is retracted and a new action is selected. This process continues until the state matrix no longer exhibits such configurations or until all actions fail to meet the requirements.

For the sake of clarifying the customized action restriction in the subarray tiling process, two examples of action selection in a 4×4-dimensional array are presented in Fig. 3. In the successful exact cover example shown in Fig. 3a, a random action, a_0 is placed on the initial state space S_0. Next, a vertical domino-shaped subarray is placed at the bottom of state S_1, occupying the second column and third column. After placing the yellow subarray, the array has a single empty cell remaining at the left-bottom corner (fourth row, first column). This single empty cell makes it impossible to place any standard domino-shaped subarray to complete the tiling. Therefore, a new action a_1'' replaces the first action a_1' to fill state matrix S_1. Similarly, action a_2'' replaces action a_2' to forward State S_3. After 8 actions, the state matrix is fully tiled as shown in state S_8. It is worth noting that although action constraints can significantly improve sample efficiency and facilitate easier learning of polyomino tiling problems, failures in tiling still occur at the beginning of the training process. Fig. 3b presents an example of failed tiling where all the actions $(a_3', a_3'', a_3''', ...)$ filling on state matrix S_3 cannot tile the state S_4 without leaving a closed empty grid or three consecutive closed empty grids. Therefore, this episode terminates after two steps of action.

The reward function in DQN plays a crucial role in guiding the model to achieve the target results. In our design, shaping the agent's behavior and facilitating policy learning to achieve optimal scanning performance with an exact tiling configuration is the goal.

A tiling reward $\delta_k = 1$ is provided for each step that tiles a subarray to the state matrix S_k to encourage the model to tile more subarrays in the array. Therefore, during training step k, the tiling reward $\delta_k = k$.

TABLE I
PSEUDOCODE OF DQN-BASED ALGORITHM

Algorithm I: DQN-based algorithm for exact tiling array aperture

Input: Initial state space S, target radiation pattern $F_0(\theta, \varphi)$
Output: Exact tiled array configuration and their corresponding phase distribution
Initialization: Learning rate α, discount factor γ, ε-greedy ε, replay memory D, batch size B, estimation network with random weights θ and target network with random weights $\theta' = \theta$, candidate subarray set A.
Process:
For episode = 1, 2, … , ζ **do**
  Initialize state space S_0
  Update ε-greedy $\varepsilon = \varepsilon_{min} + (\varepsilon_{max} - \varepsilon_{min}) \bullet \exp(-\varepsilon_{decay} \bullet t)$
  For k = 1, 2, … , κ **do**
    If k = 1
      Select a random action $A_k \sim$ Uniform{A_1, A_2, A_3… }
    Else
      Reset candidate subarray set A* with action restriction
      With probability ε select a random action A_k from set A*
      With probability 1- ε select $A_k = \text{argmax } Q(S_{k+1}; \theta_k)$
    End
    Execute action A_k and obtain reward R_k
    Store tuple $<S_k, A_k, R_k, S_{k+1}>$ in D
    Sample random mini-batch B from training set D
    If terminate
      $R_{k+1} = R_k$
    Else
      $R_k + \gamma \max_A Q(S_{k+1}, A_{k+1}; \theta'_{k+1})$
    Perform backpropagation step on loss function $L(S_k, A_k, S_{k+1} | \theta_k)$
      Every N steps update $\theta' = \theta$
  End for
End for

TABLE II
HYPER PARAMETERS OF DQN ALGORITHM

DRL parameters	Value
Learning rate α	0.01
Discount factor γ	0.9
ε-greedy ε_{max}	0.9
ε-greedy ε_{min}	0.01
ε-greedy decay rate ε_{decay}	0.001
replay memory D	2000
batch size B	64
Number of episodes ζ	3000
Number of maximum steps in each episode κ	128

In addition to achieving irregular subarray tiling, SLL is one of the most important parameters which is used to evaluate the performance of irregular phase arrays. Thus, we consider the completion of the subarray structures and their corresponding SLL in the target angular range $\Theta = \{(u, v) \mid u_{\text{low}} \le u \le u_{\text{high}}, v_{\text{low}} \le v \le v_{\text{high}}\}$ as the standard for the reward to the agent. $u = \sin(\theta)\cos(\varphi) \in [-1, 1]$, $v = \sin(\theta)\sin(\varphi) \in [-1, 1]$, and (θ, φ) represents the observation direction of far field. Low subscript and high subscript denote the lower and upper boundaries of the target angular range, respectively. In training step k, the reward R_k is composed of tiling reward δ_k and SLL reward ΔSLL by:

$$R_k = k + \Delta\text{SLL} \quad (5)$$

where ΔSLL represents the SLL difference between last training step k-1 and current training step k, which is expressed

as:

$$\Delta SLL = SLL_{k-1} - SLL_k \tag{6}$$

$$\Delta SLL_k = -20\log_{10}\left(\frac{\max\limits_{u,v\in\Theta}|F_k(u,v)|}{\max\limits_{u,v\in\Theta}|F_k(u,v)|}\right) \tag{7}$$

Here, $F_k(u, v)$ is the radiation pattern of an array tiling k subarrays. For a specific single beam scanning problem, it can be defined by:

$$F(u,v) = \sum_{p=1}^{n} f(u,v) \sum_{q=1}^{Q} e^{jk(x_{pq}u + y_{pq}v - x'_p u_0 - y'_p v_0)} \tag{8}$$

where (u_0, v_0) defines the center of scanning angle, $f(u, v)$ is the radiation pattern of the element located at (x_{pq}, y_{pq}). n is the number of tiled subarrays, and Q represents the number of elements in the subarrays. As for the pattern synthesis problems, in training step k, we can first employ (22) - (28) to calculate the phase mask of the state matrix S_n. Subsequently, we can obtain its corresponding ΔSLL according to (6). Thus, the tuple $<S_k, A_k, R_k, S_{k+1}>$ is the format of the training set.

In summary, the proposed framework aims to employ domino-shaped subarrays to tile the array aperture, while also achieving the lowest SLL at the beam steering angle of Θ. The DQN algorithm selects an action A_k to tile the state matrix S_k, resulting in a new state matrix S_{k+1}. The selection of A_k is dependent on the ε-greedy policy. For each episode, the algorithm terminates either when it reaches the maximum tiling steps or when illegal grids appear. After fully training the estimation networks, the agent is capable of quickly exact tiling the array aperture while adjusting the phase distribution of subarrays to scan the beam at the desired angles. The pseudocode of the DQN-based framework is shown in Table I.

In the training process, we train the network structure of our DQN model on the deep learning frameworks TensorFlow and Keras and the computer is configured as a single CPU and a single GPU. As shown in Table II, the learning rate α = 0.01, and the discount factor γ = 0.9. The initial ε_{max} = 0.9. An exponential decay ε_{decay} = 0.9 and minimum ε_{min} = 0.01 is selected in the ε-greedy strategy to balance exploration and exploitation. The replay memory size D is configured to store up to 2000 past experiences, ensuring diverse training samples. A batch size of 64 is used for mini-batch gradient descent, optimizing the training efficiency. In addition, for this 16×16-dimensional subarray tiling problem, the maximum number of steps in one episode is set to 128, since the array achieves exact tiling after 128 steps. Termination occurs either when there is no significant variation in loss and reward or when the maximum number of episodes (ζ = 3000) is reached.

C. Gradient-Descent Optimization on Pattern Synthesis

To guarantee reliable and precise pattern synthesis results, various pattern synthesis theories and methods have been investigated in the last decades. Among them, the gradient-descent optimization method updates the model parameters incrementally, which means it can converge to the optimal solution faster than other optimization algorithms.

Suppose a phased array is composed of N elements, and the positions of the antenna elements can be expressed as x_i and y_i along x-axis and y-axis, respectively. The radiation pattern of antenna array $F(\theta, \varphi)$ can be donated by:

$$F(\theta, \varphi) = \boldsymbol{w}^{\mathbf{H}}\, \boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \tag{9}$$

where $\boldsymbol{w} = (e^{j\Psi_1}, e^{j\Psi_2}, \ldots, e^{j\Psi_N})^T$. Ψ_i (i = 1, 2, …, N) is the phase vector of the antenna elements. The superscript H defines the complex conjugate transpose, and the superscript T denotes the transpose. θ is the elevation angle, and φ is the azimuth angle. The element $a_i(\theta, \varphi)$ is given by:

$$a_i(\theta, \varphi) = f(\theta, \varphi)\, e^{-jk_0(x_i \sin\theta \cos\varphi + y_i \sin\theta \sin\varphi)} \tag{10}$$

where $f(\theta, \varphi)$ is the element radiation pattern. $k_0 = 2\pi/\lambda_0$ is the wavenumber, and λ_0 is the wavelength. To achieve pattern synthesis in the target beam region, we aim to minimize the following loss function:

$$J_{MV}(\boldsymbol{w}) = \int_{Q_s} |F(\theta, \varphi) - F_0(\theta, \varphi) e^{j\zeta(\theta, \varphi)} F_{max}|^2 \, d\theta d\varphi \tag{11}$$

where $F_0(\theta, \varphi)$ is the target normalized amplitude pattern, Q_s is the target beam region, and $\zeta(\theta, \varphi)$ is the phase distribution of $F(\theta, \varphi)$. F_{max} is the maximum value of $|F(\theta, \varphi)|$. By tuning the complex-amplitude value of antenna elements using (11), the difference between the array radiation pattern and the target radiation pattern is minimized. This expression can be further simplified in the following manner:

$$J_{MV}(\boldsymbol{w}) = \boldsymbol{w}^{\mathbf{H}} \mathbf{R}_s \boldsymbol{w} - \boldsymbol{w}^{\mathbf{H}} \mathbf{R}_k - \mathbf{R}_k^{\mathbf{H}} \boldsymbol{w} + R_{0k} \tag{12}$$

where $\mathbf{R}_s$, $\mathbf{R}_k$, and R_{0k} can be expressed as:

$$\begin{aligned} \mathbf{R}_s &= \int_{Q_s} \boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi})\, \boldsymbol{a}^{\mathbf{H}}(\boldsymbol{\theta}, \boldsymbol{\varphi})\, d\theta d\varphi \\ \mathbf{R}_k &= F_{max} \int_{Q_s} \boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi})\, F_0(\theta, \varphi) e^{-j\zeta(\theta, \varphi)}\, d\theta d\varphi \\ R_{0k} &= F_{max}^2 \int_{Q_s} F_0^2(\theta, \varphi)\, d\theta d\varphi \end{aligned} \tag{13}$$

More details regarding the process of deduction can be found in [28]. According to the gradient descent algorithm, the complex amplitude vector $\boldsymbol{w}$ is able to update itself to find the local minimum of loss function $J_{MV}(\boldsymbol{w})$. The iterative formula is given as follows:

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k - \eta \nabla_w J_{MV}(\boldsymbol{w}) \tag{14}$$

where η defines the learning rate, and k is an iteration index. To solve a partial complex vector gradient operator presented in (14), from the mathematical point of view, the analytic function $J_{MV}(\boldsymbol{w})$ of a complex variable w $J_{MV}(\boldsymbol{w}) = g(z, z^*) = u + iv$ is complex differentiable anywhere and must satisfy the Cauchy-Riemann conditions. Here, the operates on z^* is the conjugate of z. For a differentiable function, the following partial differential operators can be defined according to the definition of Wirtinger derivatives [41]:

$$\begin{aligned} \partial g/\partial z &= 1/2\, (\partial J_{MV}(\boldsymbol{w})/\partial u - i\, \partial J_{MV}(\boldsymbol{w})/\partial v) \\ \partial g/\partial z^* &= 1/2\, (\partial J_{MV}(\boldsymbol{w})/\partial u + i\, \partial J_{MV}(\boldsymbol{w})/\partial v) \end{aligned} \tag{15}$$

TABLE III
PSEUDOCODE OF PHASE-ONLY SUBARRAY SYNTHESIS ALGORITHM

Algorithm II: Subarray-based synthesis algorithm for phase distribution manipulation

Input: Initial phase vector of array elements $\boldsymbol{w}^{\mathbf{0}}$, target radiation pattern of array $F_0(\theta, \varphi)$
Output: Optimized phase vector of array elements $\boldsymbol{w}^{k+1}$
Initialization: Learning rate η, and weight coefficient $W(\theta, \varphi)$
Process:
While stopping criterion is not satisfied **do**
  Calculate $F(\theta, \varphi)$ according to $\boldsymbol{w}^{k}$
  Update the weight coefficient $W(\theta, \varphi)$ according to (25)
  Calculate $\mathbf{R_s}$, $\mathbf{R}_k$, and R_{0k} according to (27)
  Update the (k+1)-*th* $\boldsymbol{w}^{k+1}$ according to (22)
  Normalize the $\boldsymbol{w}^{k+1}$ according to (28)
End while

It is also easily deduced from (15) that $\partial z/\partial z^* = \partial z^*/\partial z = 0$ and $\partial z/\partial z = \partial z^*/\partial z^* = 1$, which indicates both z and z^* can be treated as two independent variables in the $g(z, z^*)$. Hence, the derivatives $\partial(\boldsymbol{w}^{\mathbf{H}}\mathbf{R_s}\boldsymbol{w})/\partial w_n$, $\partial(\boldsymbol{w}^{\mathbf{H}}\mathbf{R}_k)/\partial w_n$, and $\partial(\mathbf{R}_k^{\mathbf{H}}\boldsymbol{w})/\partial w_n$ with respect to w_n, are obtained through:

$$\partial(\boldsymbol{w}^{\mathbf{H}}\mathbf{R_s}\boldsymbol{w})/\partial w_n = \partial(\sum_m \sum_n w_m^* R_{mn} w_n)/\partial w_n = \sum_n w_m^* R_{mn} \quad (16)$$
$$\partial(\boldsymbol{w}^{\mathbf{H}}\mathbf{R}_k)/\partial w_n = \partial(\sum_n w_n^* R_n)/\partial w_n = 0 \quad (17)$$
$$\partial(\mathbf{R}_k^{\mathbf{H}}\boldsymbol{w})/\partial w_n = \partial(\sum_n R_n^* w_n)/\partial w_n = R_n^* \quad (18)$$

Since $\mathbf{R_s}$ is a Hermitian matrix, $\nabla_w(\boldsymbol{w}^{\mathbf{H}}\mathbf{R_s}\boldsymbol{w})$, $\nabla_w(\boldsymbol{w}^{\mathbf{H}}\mathbf{R}_k)$, and $\nabla_w(\mathbf{R}_k^{\mathbf{H}}\boldsymbol{w})$ can be reformulated as:

$$\nabla_w(\boldsymbol{w}^{\mathbf{H}}\mathbf{R_s}\boldsymbol{w}) = \mathbf{R_S}^{\mathrm{T}}\, \boldsymbol{w}^* \quad (19)$$
$$\nabla_w(\boldsymbol{w}^{\mathbf{H}}\mathbf{R}_k) = \mathbf{0} \quad (20)$$
$$\nabla_w(\mathbf{R}_k^{\mathbf{H}}\boldsymbol{w}) = \mathbf{R}_k^* \quad (21)$$

Substituting (19)-(21) into (14), the following formulation is obtained:

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k - \eta(\mathbf{R_S}^{\mathrm{T}}\, \boldsymbol{w}^{k*} + \mathbf{R}_k^*) \quad (22)$$

which allows for straightforwardly iterative complex-amplitude vector $\boldsymbol{w}$ to minimize the loss function (11). Although the target beam region can be synthesized well according to (22), it is worth noting that large side lobes may occur during the iterations. To suppress the side lobes, the loss function is reformulated as follows:

$$J_{\mathrm{MV}}(\boldsymbol{w}) = \int_{Q_s} |F(\theta, \varphi) - F_0(\theta, \varphi)e^{j\zeta(\theta, \varphi)}F_{\max}|^2 \, d\theta d\varphi + \mathrm{U}\int_{Q_p} |F(\theta, \varphi)|^2 \, W(\theta, \varphi) \, d\theta d\varphi \quad (23)$$

where U is a constant weight coefficient to balance the weight of the side lobes region, $W(\theta, \varphi)$ is the weight coefficient for the angular direction (θ, φ), and Q_p defines the side lobes region. Hence, the $\mathbf{R_s}$ is reformulated as:

$$\mathbf{R_s} = \int_{Q_s} \boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi})\, \boldsymbol{a}^{\mathbf{H}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \, d\theta d\varphi + \mathrm{U}\int_{Q_p} \boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi})\, \boldsymbol{a}^{\mathbf{H}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \, W(\theta, \varphi) \, d\theta d\varphi \quad (24)$$

The weight coefficient is adjusted according to the following formula:

$$W(\theta, \varphi) = |F(\theta, \varphi)| / \sum_{Q_p} |F(\theta, \varphi)| \quad (25)$$

For a given domino subarray consisting of N antenna elements in candidate sets $\mathbf{A} = \{A_1, A_2, \ldots, A_p\}$, there exists N/2 grouped elements, and each domino-shaped element shares the same complex amplitude value. In this way, an exact tiling subarray layout $\mathbf{L} = \{A_2, A_3, A_i, \ldots, A_j\}_{\mathrm{N/2\times N}}$ can be regarded as an (N/2, N)-dimensional matrix, where the row number corresponds to the number of grouped elements and the column number corresponds to the total number of elements in the array. Thus, the complex-amplitude vector $\boldsymbol{w}$ in a domino subarray can be expressed as:

$$\boldsymbol{w} = (\boldsymbol{w}_s^{\mathrm{H}}\mathbf{L})^{\mathrm{H}} \quad (26)$$

where $\boldsymbol{w_s} = (e^{j\Psi_1}, e^{j\Psi_2}, \ldots, e^{j\Psi_{N/2}})^{\mathrm{T}}$ is the complex-amplitude vector for the grouped elements in the domino subarray. Substituting (26) into (11), the $J_{\mathrm{MV}}(\boldsymbol{w})$, $\mathbf{R_s}$, $\mathbf{R}_k$, and R_{0k} can be reformulated as:

$$\begin{aligned} J_{\mathrm{MV}}(\boldsymbol{w}) &= \boldsymbol{w}_s^{\mathrm{H}}\, \mathbf{R_s}\boldsymbol{w} - \boldsymbol{w}_s^{\mathrm{H}}\, \mathbf{R}_k - \mathbf{R}_k^{\mathbf{H}}\boldsymbol{w}_s + R_{0k} \\ \mathbf{R_s} &= \int_{Q_s} (\mathbf{L}\boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi}))\, (\mathbf{L}\boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi}))^{\mathrm{H}} \, d\theta d\varphi + \\ &\quad \mathrm{U}\int_{Q_p} (\mathbf{L}\boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi}))\, (\mathbf{L}\boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi}))^{\mathrm{H}} \, W(\theta, \varphi) \, d\theta d\varphi \\ \mathbf{R}_k &= F_{\max}\int_{Q_s} (\mathbf{L}\boldsymbol{a}(\boldsymbol{\theta}, \boldsymbol{\varphi}))\, F_0(\theta, \varphi) e^{-j\zeta(\theta, \varphi)} \, d\theta d\varphi \\ R_{0k} &= F_{max}^2 \int_{Q_s} F_0^2(\theta, \varphi) \, d\theta d\varphi \end{aligned} \quad (27)$$

In addition, to consider the situations where the simplified and cost-effective phase-only control architecture is preferred, we can divide each element in the complex-amplitude vector $\boldsymbol{w}$ by its magnitude, ensuring that only the phase of the element is updated in each iteration as follows:

$$\boldsymbol{w}^{\boldsymbol{k+1}} = [\frac{w_1^{k+1}}{|w_1^{k+1}|}, \frac{w_2^{k+1}}{|w_2^{k+1}|}, \ldots, \frac{w_N^{k+1}}{|w_N^{k+1}|}]^{\mathrm{T}} \quad (28)$$

In summary, the steps of the proposed phase-only subarray synthesis algorithm are given in Table III.

D. Complexity Analysis of the Proposed Algorithm

Assume that the number of elements in the array is N and that the numbers of sampling points in the main beam and side lobe regions are L_1 and L_2, respectively. For the proposed algorithm, it requires $(L_1N^2)/2$ and $3N^2(L_1+L_2)/4$ complex multiplications to calculate $\mathbf{R}_k$ and $\mathbf{R_s}$, respectively. In (14), it requires $N^2/4$ complex multiplications to calculate $\boldsymbol{w}^{k+1}$. Assume the proposed algorithm requires ξ iterations. The computational complexity of this algorithm is on the order of $O[(5L_1/4 + 3L_2/4 + 1/4)\, \xi\, N^2]$.

The estimation network is designed as a multilayer convolutional neural network (CNN). Assume the estimation network has L layers and the number of input channels and output channels are C_{in}^i and C_{out}^i in the *i-th* layer, respectively. M^i donates the side length of the *i-th* layer, and the side length of each kernel is K^i in the *i-th* layer. Then, according to the

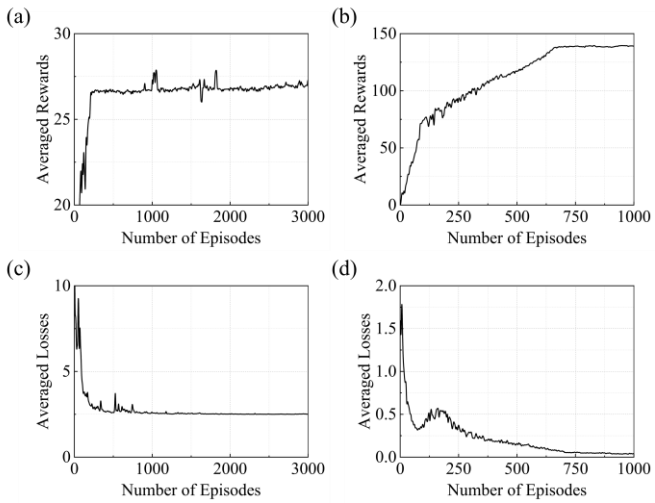

Fig. 4. Performance evaluation of the DQN algorithm applied in the 16×16 domino-shaped subarray configuration when beam steering at (θ = 30°, φ = 0°). Averaged rewards: (a) without adding action constraint; (b) with adding action constraint. Averaged loss: (c) without adding action constraint; (d) with adding action constraint.

convolutional operation theory in the CNN model [42], the computational complexity of *i-th* layer of the CNN model is on the order of $O[C^i_{in}\cdot C^i_{out}\cdot M^i\cdot M^i\cdot K^i\cdot K^i]$. Thus, the computational complexity for the estimation network with L CNN layers and ξ iterations is on the order of $O[\xi \sum_{i=1}^{L} C^i_{in}\cdot C^i_{out}\cdot M^i\cdot M^i\cdot K^i\cdot K^i]$.

III. Results and Analysis

In order to assess the effectiveness of the proposed framework, the far-field radiation performance of several representative numerical examples, including pencil beam pattern synthesis and flat-topped beam pattern synthesis with domino-shaped subarrays, is implemented in the array aperture with dimensions of 16×16.

A. Performance Evaluation of DQN Algorithm

Calculating the loss between these two networks helps to improve predictions and enhance future decision-making. Similarly, the reward value defines the SLL of subarray configurations. The comparison of the performance of averaged rewards and losses without and with action limitations is presented in Fig. 4. According to Fig. 4a, it is clearly observed that the averaged rewards continue to increase until around 350 iterations. After 350 iterations, the rewards stabilize, showing occasional fluctuations but maintaining a value of around 26.5. Meanwhile, the corresponding averaged losses converge to 2.5 after 350 iterations. The low-averaged rewards and high averaged losses indicate that the agent has failed to learn how to achieve exact tiling in subarray configurations.

On the contrary, there is a more noticeable and smoother upward trend in the performance of averaged rewards in the situation with action limitations compared to the no action limitation case. The rewards continue to increase until around 700 iterations, reaching a value of around 140. Similarly, the averaged losses consistently decrease until 700 iterations, then stabilize at a much lower value compared to the no action limitation case.

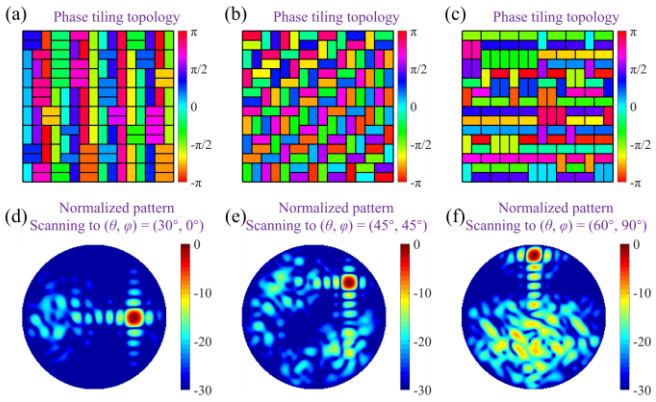

Fig. 5. Calculated results for phase-only optimized irregular arrays with single beam steering: (a), (b), (c) Optimized domino-shaped phase tiling topologies, and (d), (e), (f) their corresponding normalized radiation patterns.

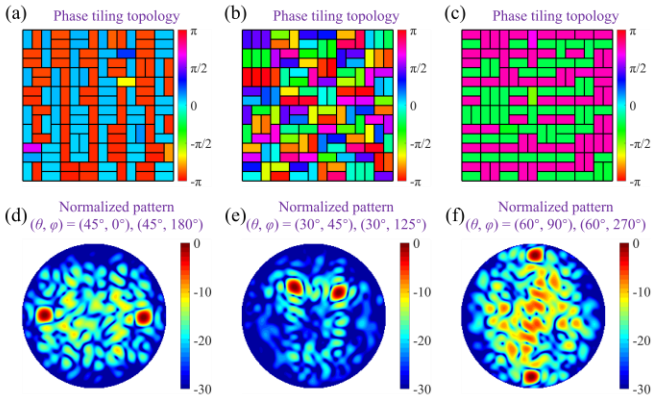

Fig. 6. Calculated results for phase-only optimized irregular arrays with dual-beam steering: (a), (b), (c) Optimized domino-shaped phase tiling topologies, and (d), (e), (f) their corresponding normalized radiation patterns.

TABLE IV
Comparison of Optimal Solution for Single Beam Pattern

(θ, φ)	PSLL (dB)	
	Proposed method	Algorithm in [43]
(30°, 0°)	-13.24	-12.6
(45°, 45°)	-11.92	-11.82
(60°, 90°)	-8.95	Not mentioned

Hence, the action constraint helps guide the learning process more effectively, leading to higher rewards and lower losses, thereby improving the overall performance of the DQN algorithm in the given subarray tiling and optimization tasks.

B. Synthesis of Irregular Arrays for Pencil Beam Pattern

The dimension of the phased array is a 16×16-element array with $0.5\lambda_0$ interelement spacing. The domino-shaped subarrays are employed to tile the array aperture, and the phase distribution of the array is optimized by the proposed gradient-descent approach.

The phase-only optimized tiling configurations of domino-shaped subarrays donated by different colors, as well as their combined normalized radiation pattern of arrays, are shown in Fig. 5 for three different scanning angles: (θ_1 = 30°, φ_1

= 0°), (θ_1 = 45°, φ_1 = 45°), and (θ_1 = 60°, φ_1 = 90°). As can be observed, the scanning angles agree well with the target ones, and the PSLL is successfully suppressed to -13.24 dB, -11.92 dB, and -8.95 dB when array scanning to (θ_1 = 30°, φ_1 = 0°), (θ_1 = 45°, φ_1 = 45°), (θ_1 = 60°, φ_1 = 90°), respectively.

Fig. 6 depicts the phase excitations of arrays and their corresponding far-field radiation pattern. The PSLL is -10.83 dB, -12.17 dB, and -6.96 dB when the beam is respectively steered to (θ_1 = 45°, φ_1 = 0°; θ_2 = 45°, φ_2 = 180°), (θ_1 = 30°, φ_1 = 45°; θ_2 = 30°, φ_2 = 125°), and (θ_1 = 60°, φ_1 = 90°; θ_2 = 60°, φ_2 = 270°). Similarly, the scanning performance of domino-shaped subarrays also maintains at acceptable levels in dual-beam steering cases.

To evaluate the SLL suppression performance of our proposed method, we compare it with one of the state-of-the-art approaches in the literature [43]. As listed in Table IV, our method is still slightly better than the optimal configuration using a differential evolution algorithm (DEA) when scanning both at (θ_1 = 30°, φ_1 = 0°) and (θ_1 = 45°, φ_1 = 45°).

From Fig. 5 and Fig. 6, it can be observed that as the elevation angle θ increases, the scanning performance gradually decreases. The phase mismatch caused by the subarray configurations can be regarded as the main factor for the deterioration of scanning performance [43]. Additionally, the PSLL in dual-beam steering cases is higher than the values observed in single-beam steering cases.

C. Synthesis of Irregular Arrays for Flat-Topped Beam Pattern

A flat-topped beam pattern is crucial for scenarios requiring uniform gain across an angular region without any nulls. The synthesized flat-topped beam patterns with beamwidth of 20°, 40°, and 60° on the 16×16-dimensional phased array with 0.5λ_0 interelement spacing are given in Fig. 7. Figures 7a to 7c represent the optimized domino-shaped phase tiling topologies for arrays with flat-topped radiation patterns with beamwidths of 20°, 40°, and 60°, respectively. The middle layer shows the results comparing the synthesized and desired radiation patterns. The black solid line represents the calculated radiation pattern of the optimized array, while the red dashed line is the target pattern. It can be observed that the PSLL is suppressed to below -10 dB in all three cases. In addition, the fluctuation in the main beam region with a beamwidth of 20° is less than 1 dB, whereas the 40° and 60° wide beam patterns exhibit more significant fluctuations. The enlarged views of the main beam regions further highlight these fluctuations, demonstrating the challenges in maintaining a perfectly flat-topped pattern as the beamwidth broadens.

The optimized configurations according to the proposed framework successfully synthesize radiation patterns that closely match the desired flat-topped beam shapes across various beamwidths. This indicates that the proposed method is capable of achieving high fidelity in beam pattern synthesis, ensuring the designed arrays perform as intended for different beamwidth requirements.

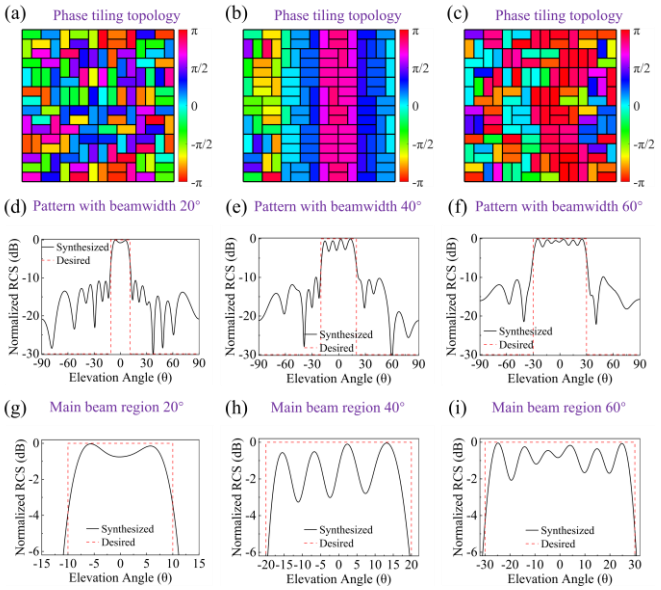

Fig. 7. Calculated results for phase-only optimized irregular arrays with flat-topped beam patterns: Optimized domino-shaped phase tiling topologies with beamwidths of (a) 20°, (b) 40°, and (c) 60°, along with their corresponding normalized radiation patterns with beamwidths of (d) 20°, (e) 40°, and (f) 60°. Enlarged normalized radiation patterns are shown for (g) 20°, (h) 40°, and (i) 60°.

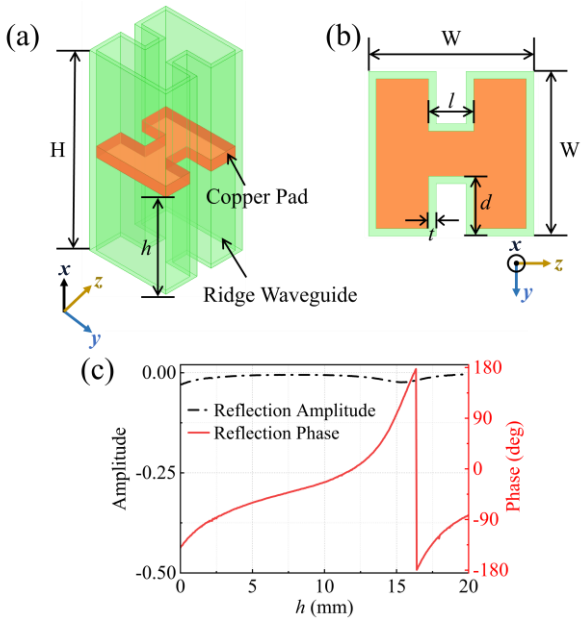

Fig. 8. Geometric design of ridge-waveguide-based unit cell: (a) perspective view and (b) top view. The corresponding geometry parameters of the design are as follows: H = 25 mm, W = 12.5mm, l = 3 mm, t = 0.5 mm, d = 4 mm. (c) Reflection amplitude and phase of unit cell operating at 12 GHz.

D. Verification on Practical Irregular Reflect Array

The practical reliability of theoretical results based on the proposed optimal configurations is verified by the consideration of practical irregular reflect arrays with real unit cells. Thus, the 16×16-dimensional reflect array is considered, where a ridge-waveguide-based structure is employed as a single unit cell. The detailed geometry information of the unit cell is shown in Fig. 8a and 8b. This structure is composed of a

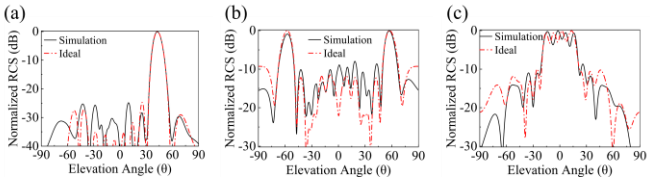

Fig. 9. The comparisons between the simulated patterns and ideal patterns when pattern synthesis in (a) ($\theta_1 = 45°$, $\varphi_1 = 45°$) at $\varphi_1 = 45°$ cutting plane, (b) ($\theta_1 = 60°$, $\varphi_1 = 90°$; $\theta_2 = 60°$, $\varphi_2 = 270°$) at $\varphi = 90°$ cutting plane, (c) flat-topped beam patterns with beamwidth of 20° at $\varphi = 0°$ cutting plane.

ridge waveguide and a copper pad. A full 360° reflection phase variation of the unit cell at 12 GHz is achieved by simply controlling the height of the copper pad within the ridge waveguide structure. Fig. 8c presents the detailed EM responses of the unit cell versus the position of the copper pad. The magnitude of the reflection amplitude is over -0.4 dB, which indicates that this structure can be regarded as a phase-only variation reflecting unit cell.

As shown in Fig. 9, three examples of previous optimal configurations based on the irregular reflect array are simulated in the High-Frequency Structure Simulator (HFSS). The *y*-polarized plane wave illuminates the surface along the -*x* axis, and all *x*, *y*, and *z* directions assume an open space condition. Due to the unavoidable mutual coupling effects between elements as well as the mismatch between the elements' real pattern model and the elements' isotropic pattern model, the normalized pattern in the sidelobe region is slightly different from the ideal pattern. While the normalized pattern in the main lobe region is in reasonable agreement with the ideal one.

IV. Conclusion

In this paper, a novel optimization model is developed for the pattern synthesis of domino-shaped subarray configurations, satisfying constraints on the target beam-scanning radiation patterns with low PSLL. Since the subarray tiling configurations and the pattern synthesis are optimized simultaneously, this optimization problem is efficiently solved by incorporating the proposed gradient-descent algorithm into the DQN model. The calculated results from the case studies of the synthesis of pencil beam patterns and flat-topped beam patterns demonstrate the good performance of the developed method. The practical reliability of the proposed model is verified by performing comparisons between the results of full-wave simulations in HFSS and theoretical calculations. It should be noted that the developed framework is not limited to domino-shaped subarray tiling problems and it applies to other polyomino-shaped configurations.

References

[1] A. Alhamed, G. Gultepe, and G. M. Rebeiz, "64-Element 16–52-GHz Transmit and Receive Phased Arrays for Multiband 5G-NR FR2 Operation," *IEEE Trans. Microwave Theory Techn.*, vol. 71, no. 1, pp. 360–372, Jan. 2023, doi: 10.1109/TMTT.2022.3200415.

[2] C.-N. Chen *et al.*, "38-GHz Phased Array Transmitter and Receiver Based on Scalable Phased Array Modules With Endfire Antenna Arrays for 5G MMW Data Links," *IEEE Trans. Microwave Theory Techn.*, vol. 69, no. 1, pp. 980–999, Jan. 2021, doi: 10.1109/TMTT.2020.3035091.

[3] S. Venkatesh, X. lu, H. Saeidi, and K. Sengupta, "A high-speed programmable and scalable terahertz holographic metasurface based on tiled CMOS chips," *Nature Electronics*, vol. 3, pp. 1–9, Dec. 2020, doi: 10.1038/s41928-020-00497-2.

[4] Y. Wang *et al.*, "Effect of Temperature on Electromagnetic Performance of Active Phased Array Antenna," *Electronics*, vol. 9, no. 8, Art. no. 8, Aug. 2020, doi: 10.3390/electronics9081211.

[5] S. Qian, B. Duan, S. Lou, C. Ge, and W. Wang, "Investigation of the Performance of Antenna Array for Microwave Wireless Power Transmission Considering the Thermal Effect," *Antennas Wirel. Propag. Lett.*, vol. 21, no. 3, pp. 590–594, Mar. 2022, doi: 10.1109/LAWP.2021.3139003.

[6] J. S. Herd and M. D. Conway, "The Evolution to Modern Phased Array Architectures," *Proceedings of the IEEE*, vol. 104, no. 3, pp. 519–529, Mar. 2016, doi: 10.1109/JPROC.2015.2494879.

[7] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 7, pp. 993–999, Jul. 1994, doi: 10.1109/8.299602.

[8] Y. Ma *et al.*, "A Thinned Irregular Array Synthesis Approach Based on Benders Decomposition," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 7, pp. 3875–3885, Jul. 2021, doi: 10.1109/TAP.2020.3044667.

[9] M. A. Elmansouri, G. R. Friedrichs, L. B. Boskovic, and D. S. Filipovic, "An *X* -Band Through *Ka* -Band Thinned All-Metal Vivaldi Phased Array," *IEEE Trans. Antennas Propagat.*, vol. 69, no. 11, pp. 7613–7623, Nov. 2021, doi: 10.1109/TAP.2021.3076680.

[10] Y. Lee, "Adaptive Interference Suppression of Phase-Only Thinned Arrays via Convex Optimization," *IEEE Trans. Antennas Propagat.*, vol. 68, no. 6, pp. 4583–4592, Jun. 2020, doi: 10.1109/TAP.2020.2977732.

[11] T. Hong, X.-P. Shi, and X.-S. Liang, "Synthesis of Sparse Linear Array for Directional Modulation via Convex Optimization," *IEEE Trans. Antennas Propagat.*, vol. 66, no. 8, pp. 3959–3972, Aug. 2018, doi: 10.1109/TAP.2018.2835641.

[12] Y.-F. Cheng, X. Ding, W. Shao, and C. Liao, "A High-Gain Sparse Phased Array With Wide-Angle Scanning Performance and Low Sidelobe Levels," *IEEE Access*, vol. 7, pp. 31151–31158, 2019, doi: 10.1109/ACCESS.2019.2901721.

[13] O. Christogeorgos, H. Zhang, Q. Cheng, and Y. Hao, "Extraordinary Directive Emission and Scanning from an Array of Radiation Sources with Hyperuniform Disorder," *Phys. Rev. Appl.*, vol. 15, no. 1, p. 014062, Jan. 2021, doi: 10.1103/PhysRevApplied.15.014062.

[14] H. Zhang, W. Wu, Q. Cheng, Q. Chen, Y.-H. Yu, and D.-G. Fang, "Reconfigurable Reflectarray Antenna Based on Hyperuniform Disordered Distribution," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 9, pp. 7513–7523, Sep. 2022, doi: 10.1109/TAP.2022.3193230.

[15] A. Ramalli, E. Boni, E. Roux, H. Liebgott, and P. Tortoli, "Design, Implementation, and Medical Applications of 2-D Ultrasound Sparse Arrays," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 69, no. 10, pp. 2739–2755, Oct. 2022, doi: 10.1109/TUFFC.2022.3162419.

[16] P. Rocca, N. Anselmi, A. Polo, and A. Massa, "Modular Design of Hexagonal Phased Arrays Through Diamond Tiles," *IEEE Trans. Antennas Propagat.*, vol. 68, no. 5, pp. 3598–3612, May 2020, doi: 10.1109/TAP.2019.2963561.

[17] P. Mevada, S. Gupta, S. Chakrabarty, and M. Mahajan, "A novel approach for design of beam steerable aperiodic planar array antenna with reduced number of elements," *Journal of Electromagnetic Waves and Applications*, vol. 36, no. 6, pp. 748-766, 2021, doi: 10.1080/09205071.2021.1983474.

[18] F. A. Dicandia and S. Genovesi, "Wide-scan and energy-saving phased arrays by exploiting Penrose tiling subarrays," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 9, pp. 7524-7537, Sept. 2022, doi: 10.1109/TAP.2022.3178917.

[19] E. Carrasco, M. Barba, and J. A. Encinar, "X-Band Reflectarray Antenna With Switching-Beam Using PIN Diodes and Gathered Elements," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 12, pp. 5700–5708, Dec. 2012, doi: 10.1109/TAP.2012.2208612.

[20] Y. Ma, S. Yang, Y. Chen, S.-W. Qu, and J. Hu, "Sparsely Excited Tightly Coupled Dipole Arrays Based on Irregular Array Techniques," *IEEE Trans. Antennas Propagat.*, vol. 68, no. 8, pp. 6098–6108, Aug. 2020, doi: 10.1109/TAP.2020.2985383.

[21] Z.-Y. Xiong, Z.-H. Xu, and S.-P. Xiao, "Beamforming properties and design of the phased arrays in terms of irregular subarrays," *IET Microwaves, Antennas & Propagation*, vol. 9, no. 4, pp. 369–379, 2015, doi: 10.1049/iet-map.2014.0308.

[22] Zi-Yuan Xiong, Zhen-Hai Xu, Si-Wei Chen, and Shun-Ping Xiao, "Subarray Partition in Array Antenna Based on the Algorithm X," *Antennas Wirel. Propag. Lett.*, vol. 12, pp. 906–909, 2013, doi: 10.1109/LAWP.2013.2272793.

[23] W. Dong, Z.-H. Xu, X.-H. Liu, L.-S.-B. Wang, and S.-P. Xiao, "Modular Subarrayed Phased-Array Design by Means of Iterative Convex Relaxation Optimization," *Antennas Wirel. Propag. Lett.*, vol. 18, no. 3, pp. 447–451, Mar. 2019, doi: 10.1109/LAWP.2019.2893946.

[24] C. Jiyuan, Z.-H. Xu, and X. Shunping, "Irregular Subarray Design Strategy Based on Weighted L1 Norm Iterative Convex Optimization," *Antennas Wirel. Propag. Lett.*, vol. 21, no. 2, pp. 376–380, Feb. 2022, doi: 10.1109/LAWP.2021.3132001.

[25] Y. Ma, S. Yang, Y. Chen, S.-W. Qu, and J. Hu, "Sparsely Excited Tightly Coupled Dipole Arrays Based on Irregular Array Techniques," *IEEE Trans. Antennas Propagat.*, vol. 68, no. 8, pp. 6098–6108, Aug. 2020, doi: 10.1109/TAP.2020.2985383.

[26] B. K. Daniel and A. L. Anderson, "Phase-Only Beam Broadening of Contiguous Uniform Subarrayed Arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 4001–4013, Oct. 2020, doi: 10.1109/TAES.2020.2987115.

[27] W. Peng, L. Xie, J. Shi, T. Gu, and Z. He, "Beampattern Synthesis Using Quantized Phase Control via Multi-Point Iterative Gradient Descent," *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 647–651, Mar. 2022, doi: 10.1109/LCOMM.2021.3135450.

[28] C. Lu, W. Sheng, Y. Han, and X. Ma, "Phase-only pattern synthesis based on gradient-descent optimization," *Journal of Systems Engineering and Electronics*, vol. 27, no. 2, pp. 297-307, 2016.

[29] T. Hong, X.-P. Shi, and X.-S. Liang, "Synthesis of Sparse Linear Array for Directional Modulation via Convex Optimization," *IEEE Trans. Antennas Propagat.*, vol. 66, no. 8, pp. 3959–3972, Aug. 2018, doi: 10.1109/TAP.2018.2835641.

[30] L. Li, L. G. Wang, F. L. Teixeira, C. Liu, A. Nehorai, and T. J. Cui, "DeepNIS: Deep Neural Network for Nonlinear Electromagnetic Inverse Scattering," *IEEE Trans. Antennas Propagat.*, vol. 67, no. 3, pp. 1819–1825, Mar. 2019, doi: 10.1109/TAP.2018.2885437.

[31] M. Chen, Y. Gong, and X. Mao, "Deep Neural Network for Estimation of Direction of Arrival With Antenna Array," *IEEE Access*, vol. 8, pp. 140688–140698, 2020, doi: 10.1109/ACCESS.2020.3012582.

[32] H. Ren, X. Yu, L. Zou, Y. Zhou, X. Wang, and L. Bruzzone, "Extended convolutional capsule network with application on SAR automatic target recognition," *Signal Processing*, vol. 183, p. 108021, Jun. 2021, doi: 10.1016/j.sigpro.2021.108021.

[33] R. Lupoiu and J. A. Fan, "Machine Learning Advances in Computational Electromagnetics," in *Advances in Electromagnetics Empowered by Artificial Intelligence and Deep Learning*, John Wiley & Sons, Ltd, 2023, pp. 225–252. doi: 10.1002/9781119853923.ch7.

[34] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, Art. no. 7676, Oct. 2017, doi: 10.1038/nature24270.

[35] B. R. Kiran *et al.*, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022, doi: 10.1109/TITS.2021.3054625.

[36] X. Wu, R. Li, Z. He, T. Yu, and C. Cheng, "A value-based deep reinforcement learning model with human expertise in optimal treatment of sepsis," *npj Digit. Med.*, vol. 6, no. 1, Art. no. 1, Feb. 2023, doi: 10.1038/s41746-023-00755-5.

[37] B. Zhang, C. Jin, K. Cao, Q. Lv, and R. Mittra, "Cognitive Conformal Antenna Array Exploiting Deep Reinforcement Learning Method," *IEEE Trans. Antennas Propagat.*, vol. 70, no. 7, pp. 5094–5104, Jul. 2022, doi: 10.1109/TAP.2021.3096994.

[38] Z. Wei *et al.*, "Fully Automated Design Method Based on Reinforcement Learning and Surrogate Modeling for Antenna Array Decoupling," *IEEE Trans. Antennas Propagat.*, vol. 71, no. 1, pp. 660–671, Jan. 2023, doi: 10.1109/TAP.2022.3221613.

[39] L. Graesser and W. L. Keng, Foundations of Deep Reinforcement Learning: Theory and Practice in Python. Addison-Wesley Professional, 2019.

[40] H. Van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016, doi: https://doi.org/10.1609/aaai.v30i1.10295.

[41] A. Hjørungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2740-2746, Jun. 2007.

[42] K. O'Shea and R. Nash, "An introduction to convolutional neural networks, " *arXiv preprint arXiv:1511.08458*, 2015.

[43] H. Jiang, Y. Gong, J. Zhang, and S. Dun, "Irregular Modular Subarrayed Phased Array Tiling by Algorithm X and Differential Evolution Algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 22, no. 7, pp. 1532–1536, Jul. 2023, doi: 10.1109/LAWP.2023.3250260.