# Representation learning-driven fully-automated inverse design of metasurfaces

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

*Published in:*
I E E E Transactions on Microwave Theory and Techniques

*DOI (link to publication from Publisher):*
https://doi.org/10.1109/TMTT.2023.3235066

*Publication date:*
2023

*Document Version*
Accepted author manuscript, peer reviewed version

# Representation Learning-driven Fully-automated Framework for the Inverse Design of Frequency Selective Surfaces

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

***Abstract*—Frequency selective surfaces (FSSs) refer to planar structures that behave with specific electromagnetic (EM) responses within a frequency range and are widely applied in wireless propagation systems. Given the fact that different EM responses correspond to distinguished topologies, conventional inverse design methods of FSSs are usually labor-intensive, as they rely on experienced human engineers to determine the topology and then rationally tune its structures. There have been great attempts using optimization algorithms (e.g. genetic algorithm) or machine learning to automate the second tuning stage after the initial EM topologies are determined by human engineers. However, the first topology selection stage still require engagements with experienced engineers. This paper proposes a fully-automated framework for the inverse design of FSSs. We achieved a fully-automated inverse design by establishing a machine-friendly mapping flow. The mapping flow derives its continuity and compactness from representation learning, which enables both auto-selection of the topology and auto-evolution of the unit cell based on the topology. The auto-selection stage automatically determines the appropriate topology by compressing the EM constraints through the principal component analysis (PCA) and classifying the topology using the support vector machine (SVM). Afterward, the auto-evolution system can efficiently evolve until it yields an optimal unit cell. We developed a self-monitor strategy to control the evolution and maximize the evolution efficiency by adaptively tuning the three modules within the auto-evolution system. We validated the presented framework with four FSS designs. The results proved its potential as a highly efficient fully-automated tool for the inverse design of FSSs.**

***Index Terms*—Auto-evolution, auto-selection, frequency selective surface, fully-automated, inverse design, representation learning.**

## I. Introduction

Metasurfaces refer to a periodical combination of overlapped planar metallic and dielectric layers in sub-wavelength size, which have unique abilities to manipulate microwave signals at desired frequencies. They possess capabilities of power-conserving transformations [1], [2], cloaking [3], absorber [4], spatial filtering [5], polarizer [6], radar cross-section reduction [7], to name just a few. Among them, the frequency selective surfaces (FSS) focus on the transmission or reflection capability. The typical inverse design of FSSs follows a performance-oriented process and takes two stages, topology selection and geometry optimization, as demonstrated in Fig. 1. Conventional inverse design methods require experienced designers with domain knowledge to select the topology in advance. Afterward, the designers need to manually tune the parameters and check the performance through repetitive full-wave electromagnetic (EM) simulations, which is time-consuming and resource-demanding. The design efficiency depends on the designers' experience level and the computation capability. There have been many published methods that automate the second optimization stage based on optimization algorithms or machine learning. However, most of them are not fully-automated and still require experienced engineers in the topology selection stage. A fully-automated inverse design method is in tremendous demand to reduce the need for experienced engineers for the inverse design of FSSs.

There have been many publications [8]–[22] utilizing machine learning to automatize the inverse design process to different degrees. Machine learning-based methods are popular due to their unique merits compared to optimization algorithm-based methods, such as genetic algorithm-based methods (GA) [23]–[25]. A number of full-wave simulations are inevitable for optimization methods [26]. By contrast, machine learning-based methods accumulate intelligence from historical data to form a surrogate model. The surrogate model can reduce or replace the need for full-wave simulation and facilitate the design for new constraints significantly. According to the automation level, we roughly divide them into semi-automated [9]–[19] and fully-automated methods [20]–[22]. Considering the strategy, there are two categories: some works [9]–[16], [21], [22] developed a forward mapping surrogate model and performed iterative optimization to find the optimal design; the others [17]–[20] directly established an inverse mapping surrogate model or utilized generative networks to produce the satisfying design.

Most of the optimization algorithm-based methods and ML-based methods [9]–[19] are semi-automated, because they mainly focused on automatic optimization in the second stage of the inverse design. Experienced designers are required to decide the topology in the first topology selection stage. For

Manuscript received * * 2022; revised * * 2022; accepted * * 2022. Date of publication * * 2022; date of current version * * 2022. This work was supported in part by the China Scholarship Council. (*Corresponding authors: Jian Ren; Ming Shen.*)

Zhao Zhou, Zhaohui Wei, Gert Frølund Pedersen, and Ming Shen are with the Department of the Electronic Systems, Aalborg University, 9220 Aalborg, Denmark (e-mail: zhazho@es.aau.dk; zw@es.aau.dk; gfp@es.aau.dk; 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 (e-mail: renjianroy@gmail.com; yzyin@mail.xidian.edu.cn).

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

Digital Object Identifier: 000000.

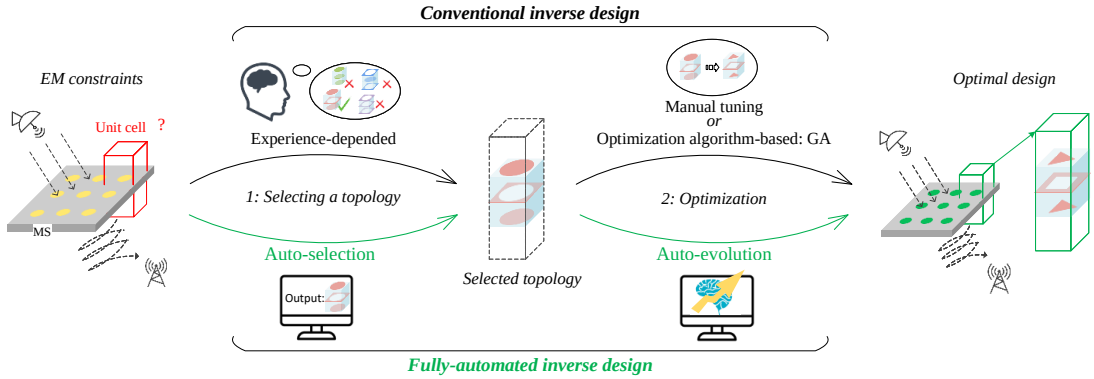


Fig. 1. Inverse design of FSSs: conventional vs. fully-automated.

instance, in [9], an artificial neural network model was developed to determine the reflective phase for a given Minkowski reflectarray element, and then the model was integrated with an optimization algorithm to generate the optimal element for the desired phase. Prado et al. [10], [11] utilized support vector machines (SVMs) to substitute full-wave EM simulation tools to accelerate the inverse design of reflectarrays. The SVMs were trained with a sufficient number of simulation results in advance. Afterward, they were used to analyze reflectarray elements at a low time cost and were integrated with iterative optimization processes to determine the reflectarray element for a given phase shift. Abdullah et al. [12] applied a data-driven supervised-learning technique integrated with the sequential-search strategy on a crusader cross topology to maximize the radar cross-section (RCS) reduction of the coding FSSs. Hodge et al. [14] proposed deep convolutional generative adversarial networks (DC-GANs) to realize the inverse design of metasurface elements. The DC-GANs were trained with elements selected from published literature. After training, a Generator, a Discriminator, and a Simulator were obtained. A three-fold process was taken to achieve given EM constraints: a) a random latent noise vector was initialized; b) generating an element by inputting the vector into the Generator and analyzing its EM response using the Simulator; c) adjusting the vector and repeating steps (b-c) until the EM response meets the EM constraints. Xiao et al. [17] proposed an inverse learning system to predict geometry variables of selected metasurfaces for a desired EM behavior without further optimization, which consists of a data classification technique and a cascade of two inverse learning machines for enhanced learning performance. Zhu et al. [18] artificially built up a modified Jerusalem Cross structure at first; afterward, they utilized a three-layer back-propagation neural network to project the reflection phase to the length of the metal arm. Likewise, Wei et al. [19] targeted the appropriate topology in advance through analysis of filter equivalent circuit before applying the genetic algorithm (GA) to reach an optimal design.

The works in [20]–[22] realized full automation. The work in [20] realized automated design by integrating a variational autoencoder, a predictor, and an optimizer. After training, they represented a metasurface into the latent variables using the encoder and optimized the variables by employing particle swarm optimizations integrated with the predictor. The decoder would decode the optimized variables into a final metasurface. In the implementations of dual-layer and three-layer metasurfaces, they simulated 17.5 thousand and 16.5 thousand structures to collect data for training the autoencoder and predictor. Naseri et al. [21] achieved a fully-automated inverse design of nonuniform bianisotropic metasurfaces in two steps. In the first step, they determined the surface parameters within several candidates using the alternating direction method of multipliers based on the two-dimensional method of moments. In the second step, they trained the machine learning surrogate model with 70 thousand samples and then utilized particle swarm optimization to optimize the surface parameters. Nadell et al. [22] utilized a twelve-layer deep neural network to model the forward mapping of an all-dielectric metasurface, which took 21 thousand simulation data for training. Afterward, they developed a novel fast-forward-dictionary-search method to solve the inverse modeling problem: they collected all combinations of geometric parameters on $13^8$ spectra to generate a dictionary. Given the constraints, they searched through the dictionary to list all combinations that approximately satisfy the constraints and gave the best ones. One shared limitation among these works is that they require tens of thousands of simulation data to train the surrogate model, which is a commonly criticized disadvantage of most machine-learning approaches.

In this paper, we manage to realize the fully-automated inverse design without establishing a dictionary mapping through tremendous simulation data in advance. To achieve this, we need to develop a machine-readable mapping of continuity and compactness. The continuity of mapping determines the degree of automation. The works in [12], [13], [18], [19] achieved only semi-automation because their mappings did not start from the constraints, which is the initial state of inverse design. The compactness of the mapping impacts the difficulty to establish and the overall efficiency. The works in [20]–[22] managed to establish mappings from all the candidate geometries to their EM behaviors, which suffered from high complexity and required a huge amount of simulation data.

Representation learning [27], [28] may offer us inspiration to facilitate the fully-automated inverse design. Boulanger-

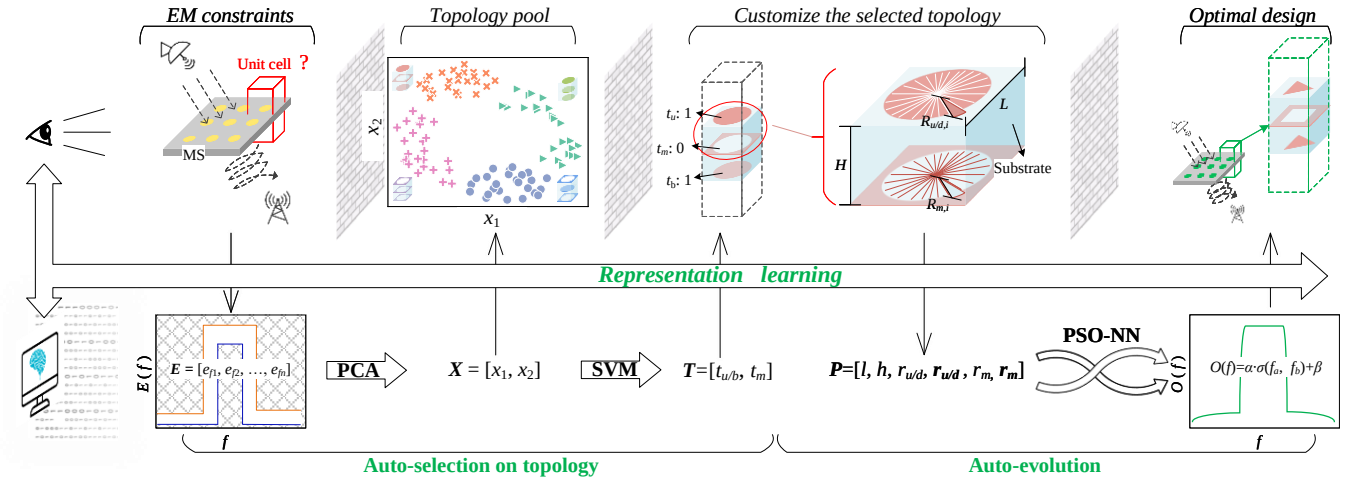


Fig. 2. The proposed fully-automated inverse design framework based on representation learning.

Lewandowski et al. [28] have proven that representation learning outperforms many state-of-art models aiming at the transcription of polyphonic music. Representation learning, also known as feature learning, allows the machine to automatically represent machine-unfriendly physical features into machine-friendly features and then smoothly project from the source space into the target space.

In this paper, a highly efficient fully-automated inverse design framework for FSSs is developed. Based on representation learning, we facilitate fully-automated inverse design by establishing a continuous and compact horizontal projection integrated with vertical representations of physical features. The horizontal projection consists of auto-selection and auto-evolution. The vertical representations include vectorization of the constraints and fan-based modeling. We have three main contributions:

1) compared with [12], [13], [18], [19], we establish a mapping from EM constraints to the topology candidates to enable auto-selection of the suitable topology, which can be accomplished without experienced designers;
2) one advantage over [20]–[22] is that we establish a cross-evolution system to customize the topology and evolve it into an optimal design instead of building a dictionary mapping in advance through tedious training, hence we take only hundreds of simulation cycles while they took tens of thousands of simulation cycles to collect data;
3) the other advantage is, unlike [20]–[22] that obtained the optimal design by optimizing several candidate geometries, the auto-evolution system can evolve the auto-selected topology into new geometries that satisfy various constraints, thanks to the proposed fan-based representation defined by only tens of parameters instead of the pixelated metal layer defined by hundreds of parameters [20].

We validate the proposed framework in four cases where different EM constraints were provided. It has been proven that, with only the EM constraints provided, the framework can automatically select the appropriate topology and yield the optimal designs in different scenarios.

The main content is arranged as follows. Section II demonstrates the workflow and components of the proposed fully-automated inverse design framework. After that, Section III describes the implementations we performed in different scenarios. The corresponding results are discussed and compared with the existing methods to clarify our advantages in Section IV. Section V lists our conclusions.

## II. Fully-Automated Inverse Design Framework

The inverse design seeks a projection from EM constraints to an optimal design. Only a few states during the projection are physically visible to human engineers, and these states belong to various isolated spaces. Thus, conventional methods rely on experienced engineers to deal with these separate spaces. To enable fully-automated inverse design, we built up a continuous mapping flow to connect these states. As illustrated in Fig. 2, this mapping flow derives from representation learning: the physical features in isolated states are represented in machine-readable features; the represented machine-readable features are projected from the current state to its posterior state. The vertical representation and horizontal projection follow adaptive rules in different states. The proposed framework can be roughly divided into two stages: auto-selection of the topology and auto-evolution of the selected topology.

### A. Auto-selection

The goal in the first stage is to select a suitable topology for given EM constraints. The EM constraints define how the FSS should behave. The EM constraints and possible topology are physically visible to human engineers. However, the EM constraints and possible topology are not machine-readable and are in isolated spaces. Thus, to enable auto-selection, we need to represent the EM constraints and possible topology into machine-readable features and establish a mapping that connects the EM constraints to the proper topology.

A single EM constraint can be represented as a curve ($E(f)$) of the coefficient (such as transmission, reflection, axial ratio,

etc.) of interest versus the frequency within the desired band. We can discretely sample the *E(f)* and generate a vector $\boldsymbol{E}$ of a constant length that represents the constraint. Sometimes, there are multiple constraints on one FSS and consequently multiple *E(f)s*. In this case, we can discretely sample each constraint *E(f)s* and calculate their normalized weighted mean to form a final vector $\boldsymbol{E}$. The weights of multiple constraints can be identical if they are equally important. Otherwise, the weights can be unequally assigned according to their importance level. This final vector $\boldsymbol{E}$ represents all the constraints. As shown in Fig. 2, $\boldsymbol{E}$ is defined as a one-dimensional vector:

$$\boldsymbol{E} = [e_{f1}, e_{f2}, ..., e_{fn}], \tag{1}$$

where each element ($e_{fi}$) of the vector $\boldsymbol{E}$ represents the constraints at a single frequency point. The size ($n$) of $\boldsymbol{E}$ equals the number of frequency points of interest. Generally, as $n$ increases, $\boldsymbol{E}$ contains more detailed information. $\boldsymbol{E}$ would be taken as input for the proposed framework to impose the expected constraints on the inverse design process.

As for the topology of the unit cell, we took a combination of three overlapped metal layers and two substrate layers between every two metal layers, as seen in Fig. 2. The up and bottom metal layers are identical as well as the two substrate layers, because the equivalent filtering circuit is symmetrical. We classified the topology according to whether each metal layer should be a patch layer or a slotted layer. The topology (T) can then be defined as:

$$\boldsymbol{T} = [t_{u/b}, t_m], \qquad t_* = \begin{cases} 1, & if \ * \ is \ patch \\ 0, & if \ * \ is \ slotted, \end{cases} \tag{2}$$

where *u/m/b* means up, middle, and bottom metal layers, respectively.

By far, we had the EM constraints and possible topology represented as machine-readable vectors, $\boldsymbol{E}$ and $\boldsymbol{T}$. The next step is to connect them by projecting $\boldsymbol{E}$ to $\boldsymbol{T}$. In machine view, it is a classification task with $\boldsymbol{E}$s as the input features and $\boldsymbol{T}$s as the classes. After training, the machine would output the class of the highest possibility that satisfies the constraints.

To facilitate the classification, we performed the principal component analysis (PCA) [29] on $\boldsymbol{E}$ to compress the input features. The PCA can help reduce the column dimension $n$ of $\boldsymbol{E}$ by projecting a high dimensional space into a low dimensional space. Therefore, it indeed projected $\boldsymbol{E}$ of high column dimension to $\boldsymbol{X}$ of low column dimension. Decreasing the low dimension can reduce the complexity of classification but may lower its accuracy. After tuning and comparison, we fixed the low dimension as 2 to balance the training complexity and classification accuracy. Here is how we performed it:

1) pre-defined $N$ sets of constraints within a broad range and represented them as $\{E(f)\text{s}, \boldsymbol{T}_s\}$, where $N = 100$;
2) discretely sampled *E(f)*s (or the weighted mean in case of multiple constraints) into $N$ sets of $\boldsymbol{E}$ and formed the data sets $\{\boldsymbol{E}_s, \boldsymbol{T}_s\}$;
3) acquired the eigenvectors $\boldsymbol{W}_s$ of $\boldsymbol{E}_s$ by solving the equation $\boldsymbol{E}_s$=$\boldsymbol{W}_s \cdot \Lambda \cdot \boldsymbol{W}_s^{-1}$;
4) picked out two eigenvectors corresponding to the two largest eigenvalues and formed $\boldsymbol{W}_{s2}$;

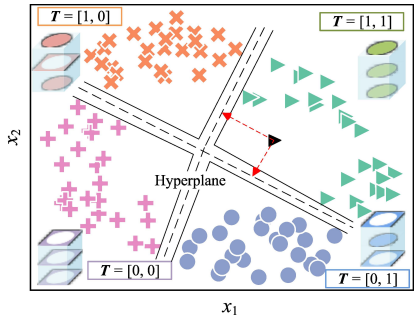


Fig. 3. The new classification space defined by $\boldsymbol{X}$ projected through PCA and the illustration of the support vector classification.

5) projected $\boldsymbol{E}$ to $\boldsymbol{X}$ by transformation, $\boldsymbol{X}$=$\boldsymbol{W}_{s2}{}^T \cdot (\boldsymbol{E} - m)$, here $m$ is the mean of $\boldsymbol{E}_s$;
6) the data set $\{\boldsymbol{E}_s, \boldsymbol{T}_s\}$ was transformed into $\{\boldsymbol{X}_s, \boldsymbol{T}_s\}$.

After the PCA, $\boldsymbol{E}$ of size $n$ was projected to $\boldsymbol{X} = [x_1, x_2]$ of size 2. To visualize the performance of the PCA, $\boldsymbol{X}$ is plotted as a two-dimensional scattering plot in Fig. 3. It shows that four types of topology denoted by four different symbols are separated in the $\boldsymbol{X}$ space transformed through PCA.

Now the classification problem was simplified to a projection problem from $\boldsymbol{X}$ to $\boldsymbol{T}$. We then automatized this projection by using SVMs [30] for classification. Here is the guideline to compute the SVM classifier:

1) performed the PCA on $\boldsymbol{E}_s$ from $\{\boldsymbol{E}_s, \boldsymbol{T}_s\}$ and formed training data $\{\boldsymbol{X}_s, \boldsymbol{T}_s\}$;
2) built an optimum hyperplane that separates the classes, its vector is as follows:

$$\boldsymbol{w} = \sum_{i=1}^{n} c_i \cdot \boldsymbol{T}_{si} \cdot \Phi(\boldsymbol{X}_{si}), \tag{3}$$

where $\boldsymbol{X}_{si}$ and $\boldsymbol{T}_{si}$ are the input and label of each training data sample, $\Phi$ represents the projection from input to output;
3) the classification vector $\boldsymbol{w}$ was obtained by solving the linear kernel-based function:

$$\max_{w,b} \ f(c_i) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{T}_{si} \cdot c_i \cdot K(\boldsymbol{X}_{si}, \boldsymbol{X}_{sj}) \cdot \boldsymbol{T}_{si} \cdot c_i,$$

$$subject \ to \ \forall i, \ \sum_{j=1}^{n} c_i \cdot \boldsymbol{T}_{si} = 0, \ 0 \le c_i \le \frac{1}{2n\lambda}; \tag{4}$$

where the linear kernel refers to $K(\boldsymbol{X}_{si}, \boldsymbol{X}_{sj}) = \sum_i^N \boldsymbol{X}_{si} \cdot \boldsymbol{X}_s$, and $\lambda > 0$ determines the trade-off between increasing margins between classes and ensuring samples being classified correctly;
4) the offset $b$ is obtained by

$$b = \boldsymbol{w}^T \cdot \Phi(\boldsymbol{X}_{si}) - \boldsymbol{T}_{si}; \tag{5}$$

5) after training, a soft margin was obtained, and the well-trained SVMs can be utilized to project a vector $\boldsymbol{X}$ to $\boldsymbol{T}$:

$$\boldsymbol{T} = \boldsymbol{w}^T \cdot \Phi(\boldsymbol{X}) - b. \tag{6}$$

In real design, an $\boldsymbol{E}$ vector of size $n$ was projected to $\boldsymbol{X} = [x_1, x_2]$ of size 2 through PCA, which is represented as a

black triangle in Fig. 3. Afterward, $\boldsymbol{X}$ was projected to $\boldsymbol{T}$ using the well-trained SVM classifier. As shown in Fig. 3, the black triangle was classified to $\boldsymbol{T}_s = [1,\ 1]$ according to hyperplanes defined by the well-trained SVM classifier. Through PCA and SVM, the vector $\boldsymbol{E}$ that represents the constraints was projected to $\boldsymbol{T}$ that represents the selected topology. The auto-selection of topology was accomplished.

### B. Auto-evolution

Auto-evolution aims to search for an optimal design of the selected topology that satisfies the EM constraints. In machine view, it is to find a mapping from the design of the unit cell to its EM satisfaction level. The EM satisfaction level ($S(\boldsymbol{O})$) of a design indicates how much its EM behavior fits the EM constraints ($\boldsymbol{E}$). It is evaluated by measuring how much the design's EM behavior fits the EM constraints. At first, we need to represent the design and its EM satisfaction level in machine-readable features.

The design of the unit cell related to any topology is a combination of three overlapped metal layers with two substrate layers clamped in the middle, as exhibited in Fig. 4. The metal layers define the topology type. According to the result $\boldsymbol{T}$ output from auto-selection in the first stage, the three metal layers can be either patch layer ($t_*$ = 1) or slotted layer ($t_*$ = 0). To generalize the instances of the topology, we divide the patch or slot into a fixed number of fans, as shown in Fig. 4. Each fan shares an angle of $360°/N_f$ and there are $N_f$ fans in total ($N_f = 72$). The length of each fan is adjustable to produce changeable capacitance and inductance. Multiple fans act as tunable resonant networks. The number of fans $N_f$ decides the complexity and flexibility of the geometry of each metal layer. Simpler geometries can be obtained by reducing $N_f$ to a smaller value, while more complex geometries can be obtained by increasing $N_f$ to a larger value. The geometrical flexibility of each metal layer decreases or increases accordingly. By adjusting the length ($r_{u/m/b,i}$) of each fan, we can tune the capacitance and inductance to form various resonant networks. The metal layer can be customized into various planar geometries. Fan-based representation reaches a balance between increasing possibilities and reducing complexity. Unlike the traditional method [18] that swept the parameters of a specific geometry, fan-based representation creates new geometries and provides more possibilities and functionalities. Compared with [19] that pixelated the metal layer and involved hundreds of parameters, fan-based representation creates sufficient structures by utilizing only tens of parameters, hence training complexity and the amount of training data are reduced significantly. To further reduce complexity, we divide the metal layer into several sectors according to filtering requirements in different directions. The fans within each sector are identical to the fans within any other sector. Suppose that the number of sectors is $S$, the number of fans to adjust will be reduced to $N_f/S$ because all the sectors are identical. $N_f/S$ is equivalent to the number of fans within each sector. The number of sectors ($S$) depends on the EM constraints: $S = 8$ fits symmetrical constraints; $S = 4$ is suitable when there are different constraints for orthogonally polarized signals; $S = 1$ or 2 is recommended for asymmetrical constraints in beam management scenarios. Figure 4 exhibits three examples of customized metal layers in case of $\boldsymbol{T} = [1, 0], N_f = 72, S = 8/4/2$. Now we represented each instance of the unit cell as a normalized vector $\boldsymbol{P}$:

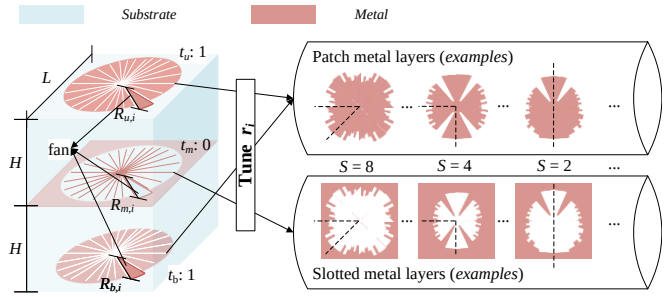


Fig. 4. Customization of the unit cell (An example when $\boldsymbol{T} = [1, 0]$).

$$\boldsymbol{P} = [l, h, r_{u/b}, \boldsymbol{r}_{u/b}, r_m, \boldsymbol{r}_m]. \tag{7}$$

The representation rules between the geometric parameters (mm) of the unit cell and its normalized $\boldsymbol{P}$ can be expressed as follows:

$$L = (4 \cdot f_c \cdot l/c - 0.8)/0.4; \tag{8}$$

$l$ is the normalized length of the unit cell, $L$ is the real length, $f_c$ is the center frequency, and $c$ is the speed of light in vacuum.

$$H = \begin{cases} 0.305, & 0 < h < 0.25 \\ 0.508, & 0.25 \leq h < 0.5 \\ 0.813, & 0.5 \leq h < 0.75 \\ 1.524, & 0.75 \leq h < 1 \end{cases} \tag{9}$$

($H$ is the standard thickness of $Rogers4003C$ substrate from Rogers Corporation [31]);

$$R_{u/b/m} = H/18 \cdot (r_{u/b/m} + 8); \tag{10}$$

$r_{u/b/m}$ is the normalized radius of the metal layer, and $R_{u/b/m}$ is the real radius.

$$R_{u/b/m,i} = R_{u/b/m} \cdot r_{u/b/m,i}. \tag{11}$$

$r_{u/b/m,i}$ represents each element of the vector $\boldsymbol{r}_{u/b/m}$ and indicates the normalized length of each fan. $R_{u/b/m,i}$ represents each element of the vector $\boldsymbol{R}_{u/b/m}$ and indicates the real length of each fan. The size of vectors $\boldsymbol{r}_{u/b/m}$ and $\boldsymbol{R}_{u/b/m}$ equals to $N_f/S$.

The EM satisfaction level ($S(\boldsymbol{O})$) of a design indicates how much its EM behavior fits the EM constraints ($\boldsymbol{E}$), and it is evaluated by measuring how much the design's EM behavior fits the EM constraints. To evaluate the satisfaction level $S(\boldsymbol{O})$ of a design, we represented its EM behavior as

$$\boldsymbol{O} = [o_{f1}, o_{f2}, ..., o_{fn}]. \tag{12}$$

Then, the design's satisfaction level $S(\boldsymbol{O})$ is evaluated by measuring how much $\boldsymbol{O}[o_{fi}]$ fits $\boldsymbol{E}[e_{fi}]$. In practice, we

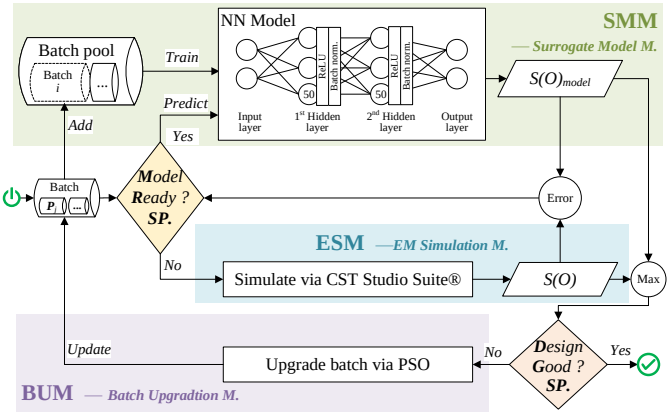


Fig. 5. The auto-evolution workflow based on PSO-NN.

decomposed $\boldsymbol{E}$ into an upper boundary, $\boldsymbol{E}_u[e_{fi,u}]$ and a lower boundary, $\boldsymbol{E}_d[e_{fi,d}]$. $S(\boldsymbol{O})$ can then be defined as (13)

$$S(\boldsymbol{O}) = \frac{1}{|\boldsymbol{L}(\boldsymbol{O})|}, \quad \boldsymbol{L}(\boldsymbol{O}) = \begin{cases} 0, & e_{fi,d} < o_{fi} < e_{fi,u} \\ \beta \cdot e_{fi,u} + (1-\beta) \cdot e_{fi,d}, & else \end{cases} \tag{13}$$

Here, $\beta$ determines the trade-off between emphasizing the upper boundary and emphasizing the lower boundary.

After representation, now the auto-evolution is to find the optimal design $\boldsymbol{P}_{optimal}$ that maximizes the satisfaction level $S(\boldsymbol{O}(\boldsymbol{P}))$, which can be expressed as

$$\boldsymbol{P}_{optimal} = \underset{\boldsymbol{P}}{argmax}\{S(\boldsymbol{O}(\boldsymbol{P}))\} \tag{14}$$

To solve this, we set up a cross-evolution system, as illustrated in Fig. 5. As its name implies, this system iteratively operates across three modules: a surrogate model module (SMM), an EM simulation module (ESM), and a batch upgradation module (BUM).

The SMM consists of a batch pool, a neural network (NN) model, and an output. The batch pool collects all the designs ($\boldsymbol{P}s$) produced by far and provides training data for the NN model. The NN model has two hidden layers, each equipped with 50 neurons and attached with a batch normalization [32]; it utilizes the Adaptive Moment Estimate (Adam) [33] as the optimizer; the Rectified Linear Unit (ReLU) [34] is employed as the activation function. The cost function of the surrogate model is the mean square error between the predicted $S(\boldsymbol{O})$ and the real $S(\boldsymbol{O})$ given by the ESM. The SMM operates in two states, training and predicting. In the training state, it utilizes the historical data from the batch pool to train a surrogate NN model; in the predicting state, it takes a new batch of the designs ($[\boldsymbol{P}_j]$) as input in each iteration and predicts their corresponding $S_i(\boldsymbol{O})$s, the $S(\boldsymbol{O})$s in iteration $i$. The SMM operates in the training state from the beginning, and it shifts to the predicting state when the NN model is well-trained. The NN model is considered well-trained when the predicted $S_i(\boldsymbol{O})$ is close enough to the real $S_i(\boldsymbol{O})$ given by the ESM. The SMM stops running when an optimal design has been obtained.

The ESM is supported by Computer Simulation Technology (CST) Studio Suite®, a full-wave EM modeling and simulation software. This module simulates each design within a batch ($[\boldsymbol{P}_j]$) in each iteration and yields its real $S_i(\boldsymbol{O})$. The real $S_i(\boldsymbol{O})$ is taken as the output label for training the NN model in the SMM. The real $S_i(\boldsymbol{O})$ is compared with the predicted $S_i(\boldsymbol{O})$ to evaluate the performance of the NN model in the SMM. The ESM stops running when the NN model has been well-trained.

At the end of each iteration, the BUM upgrades the batch and produces a new batch for the next iteration. The upgradation follows the rule of the particle swarm optimization (PSO) [35]. Suppose that the batch is $[\boldsymbol{P}_j]_i$ in iteration $i$, then a new batch $[\boldsymbol{P}_j]_{i+1}$ for next iteration $i+1$ can be expressed as:

$$[\boldsymbol{P}_j]_{i+1} = [\boldsymbol{P}_j]_i + \boldsymbol{U}_{i+1}. \tag{15}$$

Here, $\boldsymbol{U}_{i+1}$ depends on the batch-best design ($\boldsymbol{P}_{bb,i}$) and the historical-best design ($\boldsymbol{P}_{hb,i}$). The batch-best design ($\boldsymbol{P}_{bb,i}$) refers to the best design of current batch that maximizes $S_i(\boldsymbol{O}([\boldsymbol{P}_j]))$ in iteration $i$. The historical-best design refers to the best design of all the batches ($\boldsymbol{P}_{hb,i}$) that maximizes $S(\boldsymbol{O}([\boldsymbol{P}]))$. $\boldsymbol{U}_{i+1}$ can be expressed as:

$$[\boldsymbol{U}_j]_{i+1} = w \cdot [\boldsymbol{U}_j]_i + c_1 \cdot r_1 \cdot \boldsymbol{P}_{bb,i} + c_2 \cdot r_2 \cdot \boldsymbol{P}_{hb,i}, \tag{16}$$

where

$$w = w_{max} + \frac{i}{I} \cdot (w_{max} - w_{min}). \tag{17}$$

Here, $w_{max}$, $w_{min}$, $c_1$, and $c_2$ are adjustable hyperparameters; $i$ and $I$ are the current and maximum iterations, respectively; $r_1$ and $r_2$ are random floats within the range [0, 1], as defined in [35]. The BUM keeps running until an optimal design has been reached.

The three modules cooperate. a) The batch pool in the SMM is expanded iteratively by gathering new batches generated by the BUM. b) The ESM provides the real $S_i(\boldsymbol{O})$ as the output label for the SMM to evaluate the performance of the surrogate NN model; the SMM trains the surrogate model to replace the ESM. c) in each iteration, the BUM generates a new batch according to $S_i(\boldsymbol{O})$ obtained from the SMM/ESM, and the new batch is sent to the SMM/ESM in the next iteration.

The running schedule of the three modules is controlled by the model-ready sign producer (MRSP) and the design-good sign producer (DGSP). The MRSP and DGSP monitor the auto-evolution system, and they activate or abort the modules by releasing the model-ready and design-good signs. The model-ready sign indicates that the model has been well-trained. The model-ready sign is released when the mean square error between the SMM's predicted $S_i(\boldsymbol{O})$ and the ESM's real $S_i(\boldsymbol{O})$ reaches a minimum threshold. The mean square error between the SMM's predicted $S_i(\boldsymbol{O})$ and ESM's real $S_i(\boldsymbol{O})$ is the cost function of the MRSP. The model-ready sign shifts the SMM from training to predicting state, and it stops the ESM. The design-good sign indicates that indicates the optimal design has been achieved. The cost function of DGSP is $S_i(\boldsymbol{O})$. The design-good sign is released to stop the whole system when the $S_i(\boldsymbol{O})$ reaches a maximum threshold. In reality, the actual running schedule of the system depends on the EM constraints.

The advantage of the cross-evolution system is to integrate three modules, upgrade the training data dynamically, and adjust working states adaptively. The cross-evolution system

TABLE I
THE PARAMETERS OF THE DESIGNED BAND-PASS UNIT CELL

| Parameter | Value (mm) | Parameter | Value (mm, $i \in [0, 9]$) |
|---|---|---|---|
| $L$ | 12 | $\boldsymbol{R}_{u/b}$ | [8.40, 4.37, 8.51, 8.96, 9.30, 7.95, 4.82, 6.61, 10.64, 7.39] |
| $H$ | 1.524 | $\boldsymbol{R}_m$ | [3.06, 6.43, 4.94, 4.16, 2.98, 4.63, 1.57, 3.37, 4.39, 3.29] |

can maximize the efficiency to evolve the selected topology into an optimal design automatically.

## III. VALIDATION

We validated our framework in four common scenarios: band-pass, dual-band-pass, high-pass, and linear-to-circular polarizer. Note that the operating frequencies were set arbitrarily without any preference for fair validation.

### A. Band-pass

The band-pass behavior is widely required to raise the Signal to Noise (S/N) ratio and improve the sensitivity of wireless transmitters/receivers [36], [37]. A band-pass FSS acts as a spatial filter for incident microwave signals. It allows only the signals between a pre-defined frequency range to pass through and attenuates the remaining.

A band-pass FSS is evaluated by measuring its reflection coefficient ($S_{11/22}$) and transmission coefficient ($S_{21/12}$). As we proceeded our experiment in the ideal loss-free EM environment supported by CST, where the sum square of $S_{11/22}$ and $S_{21/12}$ remains constant and $S_{21}$ equals $S_{21}$, we only utilized $S_{21}$ to evaluate the unit cells.

For validation, we pre-defined the constraints as a band-pass behavior from 6 GHz to 7.5 GHz, an insert loss less than 2 dB, as expressed in (18):

$$S_{21} \begin{cases} < -15\,\text{dB}, & f < 5\,\text{GHz}; \\ > -2\,\text{dB}, & 6\,\text{GHz} < f < 7.5\,\text{GHz}; \\ < -15\,\text{dB}, & f > 9\,\text{GHz}. \end{cases} \tag{18}$$

By discretely sampling 200 points from 2 GHz to 12 GHz, the constraints were represented into two vectors of length 200 that indicate upper and lower boundaries (Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$), as given in (19) and (20). The $\boldsymbol{E}$ vector was defined as the normalized mean of $\boldsymbol{E}_u$ and $\boldsymbol{E}_d$, as given in (21).

$$\boldsymbol{E}_u = [\overbrace{-15, ..., -15}^{length:60}, \overbrace{0, ..., 0}^{length:80}, \overbrace{-15, ..., -15}^{length:60}]; \tag{19}$$

$$\boldsymbol{E}_d = [\underbrace{-30, ..., -30}_{length:80}, \underbrace{-2, ..., -2}_{length:30}, \underbrace{-30, ..., -30}_{length:90}]; \tag{20}$$

$$\boldsymbol{E} = \frac{(\boldsymbol{E}_u + \boldsymbol{E}_d)}{2 \times max(|E_{ui}|, |E_{di}|)}, \quad i = 1, 2, ..., 200. \tag{21}$$

According to $\boldsymbol{E}$, the auto-selection component pointed out the appropriate topology, $\boldsymbol{T}$= [1, 0]. The metal layers of the topology were divided into 8 sectors, and each sector had 10 fans, because the constraints were symmetrical. Once the topology was decided, the auto-evolution system activated and yielded an optimal band-pass unit cell after only 14 iterations, as seen in Fig. 6(a). There were 20 samples in each iteration, including 280 samples in total. The geometry of the designed unit cell is shown in Fig. 7 and Table I listed its geometric parameters. Figure 6(b) exhibits its simulated transmission coefficient ($S_{21}$, the green curve labeled as "Sim. $S_{11}$") of the output unit cell, where red and blue lines mark the upper boundary (Up_bound) and lower boundary (Low_bound). We can observe that the desired band-pass behavior was realized.

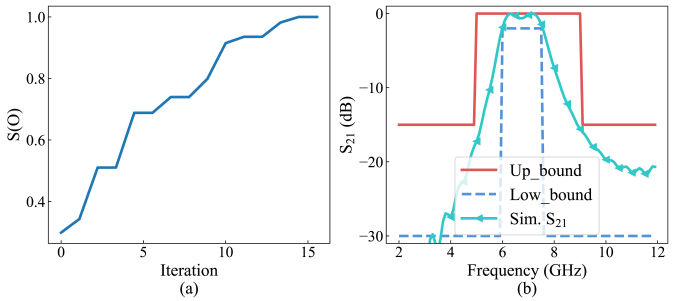


Fig. 6. The evolution result of the band-pass unit cell: (a) The evolution record over the iterations; (b) The boundaries (Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$) and simulated $S_{21}$ of the designed unit cell.

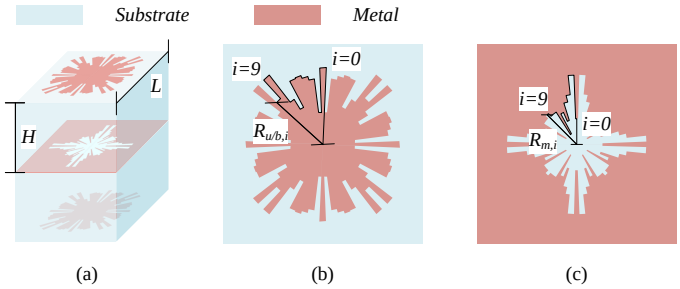


Fig. 7. The final design of the unit cell for the band-pass FSS: (a) Its overall structure; (b) Its up/bottom metal layer; (c) Its middle metal layer.

To validate the simulation results, a prototype of a band-pass surface that consists of 15 by 15 designed unit cells was fabricated and measured in an anechoic chamber. Three metal layers are etched on the surface of two $Rogers4003C$ substrate layers with a thickness of 1.524 mm. The measurement setup is shown in Fig. 8. A transmitter horn and a receiver horn were placed on two sides of the FSS prototype under test, they are opposite to each other and are connected to a network analyzer. The measured $S_{21}$ is compared with the simulated $S_{21}$ in Fig. 9. A good agreement level between measurement and simulation can be observed, which verifies the performance of this band-pass design.

### B. Dual-band-pass

The significant impact of dual-band-pass FSSs [5] attracts both the industry and academia as the $5G/6G$ of wireless communication approaches. Dual-band-pass FSSs can decouple the shared aperture base station antennas and enable their co-existence.

A dual-band-pass FSS should serve as a filter that sieves the approaching microwave signals within two separate frequency ranges. Compared with the single-band-pass case, it

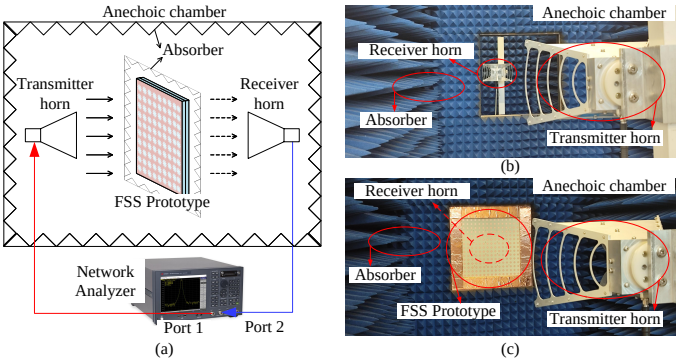


Fig. 8. (a) Illustration of the measurement setup; (b) Photograph of the anechoic chamber; (c) Photograph of the FSS prototype in the anechoic chamber.

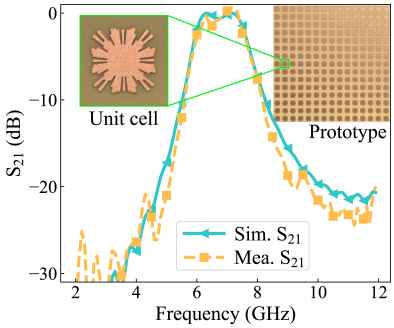


Fig. 9. Measured and simulated $S_{21}$s of the band-pass prototype.

increases the difficulty for human engineers, but it makes no difference in the perspective of the machine equipped with our framework.

Likewise, we pre-defined the constraints as a dual-band-pass behavior at 2.85 GHz and 8 GHz, with bandwidths of 300 MHz and 400 MHz, an insert loss of less than 5 dB, and a roll-off rate of 25 dB/GHz, as expressed in (22):

$$S_{21}\begin{cases} < -15\,\mathrm{dB}, & f < 2.3\,\mathrm{GHz};\\ > -5\,\mathrm{dB}, & 2.7\,\mathrm{GHz} < f < 3\,\mathrm{GHz};\\ < -15\,\mathrm{dB}, & 3.4\,\mathrm{GHz} < f < 7.4\,\mathrm{GHz}.\\ > -5\,\mathrm{dB}, & 7.8\,\mathrm{GHz} < f < 8.2\,\mathrm{GHz};\\ < -15\,\mathrm{dB}, & f > 8.6\,\mathrm{GHz}. \end{cases} \tag{22}$$

The constraints were discretely sampled and then represented into Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$), as given in (23) and (24). The $\boldsymbol{E}$ vector was defined as the normalized mean of $\boldsymbol{E}_u$ and $\boldsymbol{E}_d$), as seen in (25).

$$\boldsymbol{E}_u = [\overbrace{...,-15}^{length:46}, \overbrace{0,...,\ 0}^{length:22}, \overbrace{-15,...,-15}^{length:80}, \overbrace{0,...,\ 0}^{length:24}, \overbrace{-15,...}^{length:28}]; \tag{23}$$

$$\boldsymbol{E}_d = [\underbrace{...,-30}_{length:54}, \underbrace{-5,...,-5}_{length:6}, \underbrace{-30,...,-30}_{length:96}, \underbrace{-5,...,-5}_{length:8}, \underbrace{-30,...}_{length:36}]; \tag{24}$$

$$\boldsymbol{E} = \frac{(\boldsymbol{E}_u + \boldsymbol{E}_d)}{2 \times max(|E_{ui}|, |E_{di}|)}, \qquad i = 1, 2, ..., 200. \tag{25}$$

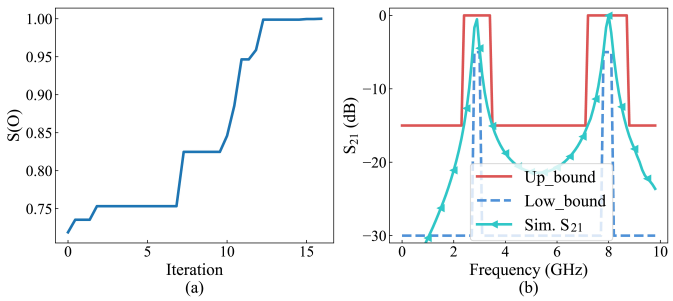


Fig. 10. The evolution result of the dual-band-pass unit cell: (a) The evolution record over the iterations; (b) The boundaries (Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$) and simulated $S_{21}$ of the designed unit cell.

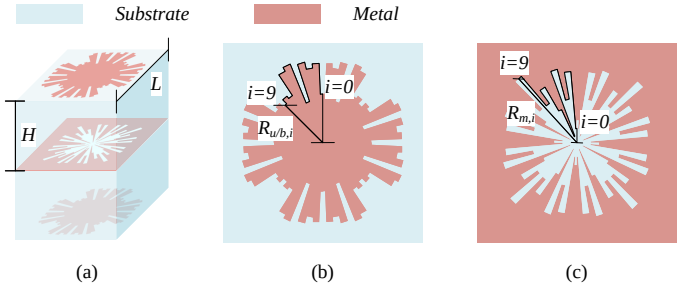


Fig. 11. The designed unit cell for the dual-band-pass FSS: (a) Its overall structure; (b) Its up/bottom metal layer; (c) Its middle metal layer.

TABLE II
THE PARAMETERS OF THE DESIGNED DUAL-BAND-PASS UNIT CELL

| Parameter | Value (mm) | Parameter | Value (mm, $i \in [0, 9]$) |
|---|---|---|---|
| $L$ | 18.98 | $\boldsymbol{R}_{u/b}$ | [3.17, 5.17, 4.90, 5.35, 2.81, 5.53, 5.26, 3.63, 3.81, 3.45] |
| $H$ | 0.813 | $\boldsymbol{R}_m$ | [1.48, 0.93, 4.69, 2.90, 5.00, 0.43, 2.41, 4.19, 3.15, 5.67] |

According to $\boldsymbol{E}_u$ and $\boldsymbol{E}_d$, the topology of $\boldsymbol{T}$= [1, 0] was auto-selected based on the represented dual-band-pass constraints. The metal layers of the topology were divided into 8 sectors, and each sector had 10 fans, because the constraints were symmetrical. Afterward, it went through the auto-evolution system and produced an optimal unit cell to form the required dual-band-pass FSS. The evolutionary history was recorded in Fig. 10(a), indicating that overall, 12 iterations were taken to arrive at the final design. No noticeable enhancement can be observed after 12 iterations. Figure 10(b) depicts the achieved transmission behavior at around 2.85 GHz and 8 GHz. The corresponding unit cell shows its geometry in Fig. 11 and lists its detailed size in Table II.

Using 15 by 15 designed unit cells, a dual-band-pass surface was constructed and a prototype was fabricated and measured to verify the design performance. Three metal layers are etched on the surface of two $Rogers4003C$ substrate layers with a thickness of 0.813 mm. The prototype was measured using the same measurement setup. The measured $S_{21}$ is compared with the simulated $S_{21}$ in Fig. 12. $S_{21}$ was measured from 2 GHz to 10 GHz due to the frequency limitation of the measurement system. Besides, frequencies below 2 GHz are out of the operating band. Within the main operating band, the measured

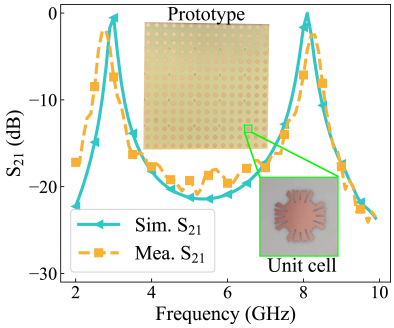


Fig. 12. Measured $S_{21}$ and simulated $S_{21}$ of the dual-band-pass prototype.

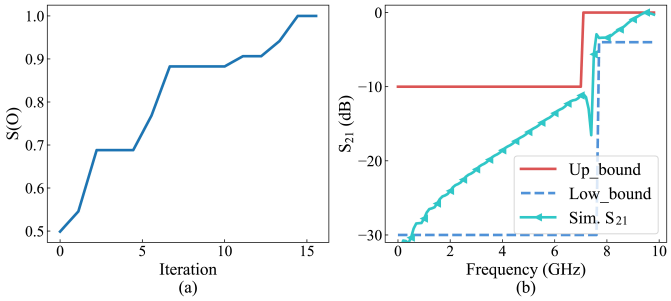


Fig. 13. The evolution result of the high-pass case: (a) The evolution record over the iterations; (b) The boundaries (Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$) and simulated $S_{21}$ of the designed unit cell.

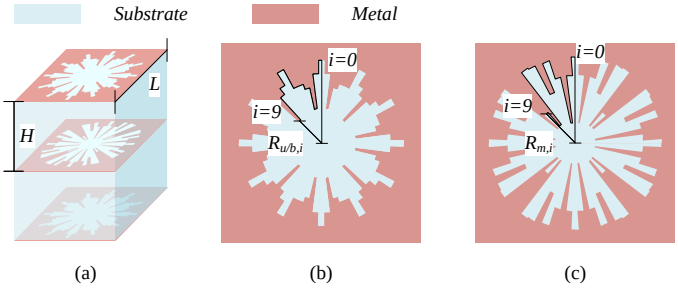


Fig. 14. The designed unit cell for the high-pass FSS: (a) Its overall structure; (b) Its up/bottom metal layer; (c) Its middle metal layer.

$S_{21}$ agrees well with the simulated $S_{21}$, which validates the performance of this dual-band-pass design.

### C. High-pass

The high-pass FSSs prevent signals below a cut-off frequency [38], [39], which is also a widely needed spatial filtering behavior.

Similarly, we pre-defined the constraints as a high-pass behavior, with a cut-off frequency at 7.5 GHz, an insert loss of less than 4 dB, and a roll-off rate of 15 dB/GHz, as expressed in (26):

$$S_{21}\begin{cases} < -10\ dB, & f < 7\ GHz; \\ > -4\ dB, & f > 7.5\ GHz. \end{cases} \tag{26}$$

The constraints were discretely sampled and represented into two vectors that indicate upper and lower boundaries (Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$) as given in (27) and (28).

TABLE III
THE PARAMETERS OF THE DESIGNED HIGH-PASS UNIT CELL

| Parameter | Value (mm) | Parameter | Value (mm, $i \in [0, 9]$) |
|---|---|---|---|
| $L$ | 17.32 | $\boldsymbol{R}_{u/b}$ | [6.59, 5.54, 2.77, 4.48, 4.88, 4.88, 6.59, 5.45, 4.88, 2.52] |
| $H$ | 0.305 | $\boldsymbol{R}_m$ | [5.63, 4.44, 1.33, 4.37, 5.63, 3.56, 5.70, 5.78, 1.48, 2.67] |

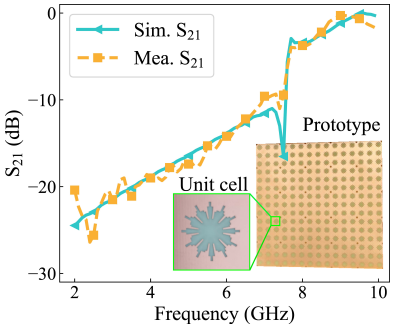


Fig. 15. Measured and simulated $S_{21}$s of the high-pass prototype.

The $\boldsymbol{E}$ vector was fixed as the normalized mean of $\boldsymbol{E}_u$ and $\boldsymbol{E}_d$), as seen in (29).

$$\boldsymbol{E}_u = [\overbrace{-10, ..., -10}^{length:140}, \overbrace{0, ..., 0}^{length:60}]; \tag{27}$$

$$\boldsymbol{E}_d = [\underbrace{-30, ..., -30}_{length:150}, \underbrace{-4, ..., -4}_{length:50}]; \tag{28}$$

$$\boldsymbol{E} = \frac{(\boldsymbol{E}_u + \boldsymbol{E}_d)}{2 \times max(|E_{ui}|, |E_{di}|)}, \qquad i = 1, 2, ..., 200. \tag{29}$$

According to $\boldsymbol{E}_u$ and $\boldsymbol{E}_d$, a unit cell with three slotted metal layers ($\boldsymbol{T}$= $[0, 0]$) was auto-selected given the high-pass filtering constraint. The metal layers of the topology were divided into 8 sectors, and each sector had 10 fans, because the constraints were symmetrical. After a 15-iteration auto-evolution, the unit cell achieved a satisfying high-pass filtering behavior with a cut-off frequency at 6.85 GHz, as shown in Fig. 13(b). The evolution history is recorded in Fig. 13(a). The optimal high-pass design and corresponding parameters are given in Fig. 14 and Table III, respectively.

A high-pass surface prototype was fabricated and measured to verify the simulation results, which consists of 15 by 15 designed high-pass unit cells. Three metal layers are etched on the surface of two $Rogers4003C$ substrate layers with a thickness of 0.305 mm. The prototype was measured using the same measurement setup. The measured and simulated $S_{21}$s are compared in Fig. 15. Due to the frequency limitation of the measuring system, $S_{21}$ was measured from 2 GHz to 10 GHz. Additionally, the operating frequency band does not include frequencies below 2 GHz. It can be observed that measurement and simulation agree well within the operating band, which validates the effectiveness of the high-pass design.

### D. Polarizer

Circular polarization is widely preferred in satellite or point-to-point communication systems due to its immunity to the Faraday rotation effects, polarization mismatch, or multi-path

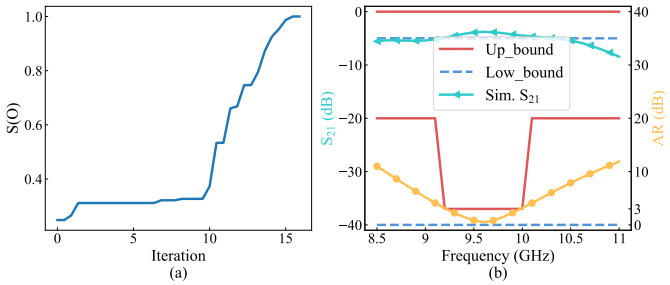


Fig. 16. The evolution result of the polarizer case: (a) The evolution record over the iterations; (b) The boundaries (Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$) and simulated $S_{21}$ and AR of the designed unit cell.

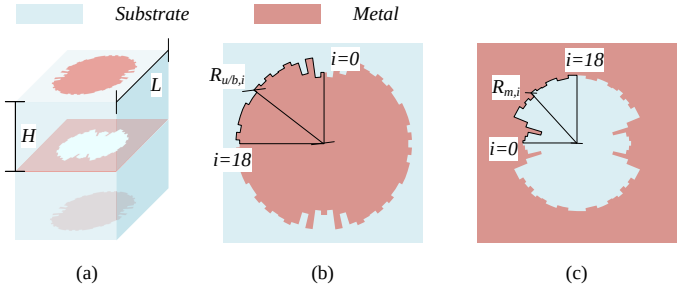


Fig. 17. The designed unit cell for the polarizer FSS: (a) Its overall structure; (b) Its up/bottom metal layer; (c) Its middle metal layer.

fading issues. Mounting a linear-to-circular polarizer [40], [41] on top of a linearly polarized antenna is a valuable way to produce circular polarization.

We pre-defined the constraints as a linear-to-circular polarization transformation behavior at 9.6 GHz, with a bandwidth of 800 MHz, an insertion loss of less than 5 dB, and an axial ratio (AR) of less than 3 dB, as listed in (30).

$$\begin{aligned} S_{21} &> -5\,\text{dB}, \quad 9.2\,\text{GHz} < f < 10\,\text{GHz};\\ AR &< 3\,\text{dB}, \quad 9.2\,\text{GHz} < f < 10\,\text{GHz}. \end{aligned} \tag{30}$$

The main difference from the former cases is that now the EM constraints ($\boldsymbol{E}(f)$s) attribute to the transmission coefficient ($S_{21}$) and the axial ratio (AR) as well. Similarly, we discretely sampled 200 points from 8.5 GHz to 11 GHz and generated upper and lower boundaries (Up_bound $\boldsymbol{E}_{uS21}$&$\boldsymbol{E}_{uAR}$ and Low_bound $\boldsymbol{E}_{dS21}$&$\boldsymbol{E}_{dAR}$), as given in (31-34). The $\boldsymbol{E}$ vector was defined as the normalized weighted mean of the four boundaries, as given in (35).

$$\boldsymbol{E}_{uS21} = [\overbrace{0, ..., 0}^{length:200}]; \tag{31}$$

$$\boldsymbol{E}_{dS21} = [\underbrace{-5, ..., -5}_{length:200}]; \tag{32}$$

$$\boldsymbol{E}_{uAR} = [\overbrace{20, ..., 20}^{length:56}, \overbrace{3, ..., 3}^{length:64}, \overbrace{20, ..., 20}^{length:80}]; \tag{33}$$

$$\boldsymbol{E}_{dAR} = [\underbrace{0, ..., 0}_{length:200}]; \tag{34}$$

$$\boldsymbol{E} = \frac{(\boldsymbol{E}_{uS21} + \boldsymbol{E}_{dS21} - \boldsymbol{E}_{uAR} - \boldsymbol{E}_{dAR})}{4 \times max(|E_{uS21i}|, |E_{dS21i}|, |E_{uARi}|, |E_{dARi}|)}, \quad i = 1, 2, ..., 200. \tag{35}$$

The polarizer should generate a 90 ° phase difference between the orthogonally polarized signals to form circular polarization. Therefore, the metal layers of the topology were divided into 4 sectors, and each sector has 19 fans. The proposed framework completed the design process after 16 iterations. As usual, we recorded the satisfaction level ($S(\boldsymbol{O})$) during evolution in Fig. 16. $S_{21}$ and AR of the optimal design are shown in Fig. 16(a) and (b), and its geometry and parameters are shown in Fig. 17 and Table IV. The designed polarizer has an insertion loss less than 5 dB and an AR less than 3 dB from 9.2 GHz to 10 GHz.

TABLE IV
THE PARAMETERS OF THE DESIGNED POLARIZER UNIT CELL

| Parameter | Value (mm) | Parameter | Value (mm, $i \in [0, 18]$) |
|---|---|---|---|
| $L$ | 8.2 | $\boldsymbol{R}_{u/b}$ | [3.20, 3.02, 3.88, 3.24, 3.80, 3.96, 3.84, 3.82, 3.96, 3.86, 3.96, 3.82, 3.80, 3.96, 3.98, 3.96, 3.82, 3.98, 3.80] |
| $H$ | 0.813 | $\boldsymbol{R}_m$ | [1.26, 1.22, 1.16, 0.86, 1.24, 1.58, 1.60, 1.48, 1.54, 1.48, 1.58, 1.54, 1.52, 1.58, 1.60, 1.52, 1.54, 1.60, 1.58] |

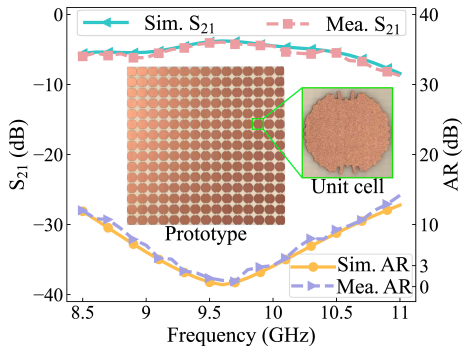


Fig. 18. Measured and simulated $S_{21}$s and $AR$s of the polarizer prototype.

To validate the simulation results, a polarizer prototype that consists of 15 by 15 designed unit cells was fabricated and measured. Three metal layers are etched on the surface of two $Rogers4003C$ substrate layers with a thickness of 0.813 mm. The polarizer prototype was measured using the same measurement setup. Measured $S_{21}$ and $AR$ are compared with simulated ones in Fig. 18(b). The performance of this polarizer design is confirmed by the measurement results, which are in good agreement with the simulation results.

## IV. DISCUSSION

The validation results in Section III proved that the proposed framework could automatically inverse design a unit cell of the FSS to satisfy any given EM constraints in real scenarios. The implementations suggest that the whole inverse design process required no human experience and took only around 15 iterations and 100 + 300 samples on average. Noted that several geometries of the designed unit cell are strange, because $N_f$ is set as 72 to show its potential of producing complex and flexible geometries. Such complex geometries

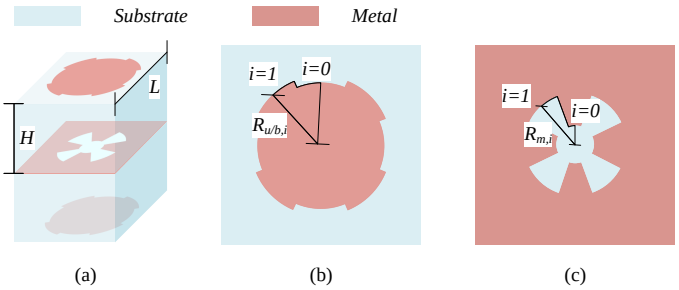


Fig. 19. Simpler band-pass unit cell with $N_f = 16$: (a) Its overall structure; (b) Its up/bottom metal layer; (c) Its middle metal layer.

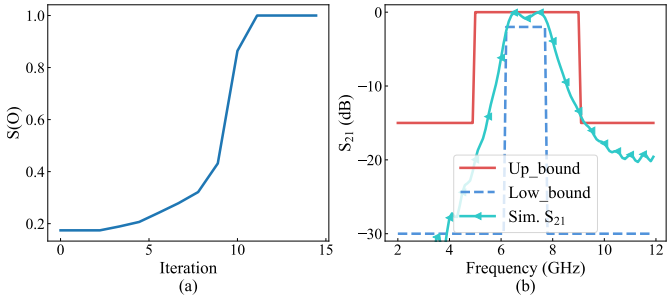


Fig. 20. The evolution result of the simpler band-pass unit cell: (a) The evolution record over the iterations; (b) The boundaries (Up_bound $\boldsymbol{E}_u$ and Low_bound $\boldsymbol{E}_d$) and simulated $S_{21}$ of the designed simpler unit cell.

TABLE V
THE PARAMETERS OF THE DESIGNED SIMPLER BAND-PASS UNIT CELL

| Parameter | Value (mm) | Parameter | Value (mm, $i \in [0, 1]$) |
|---|---|---|---|
| $L$ | 15.23 | $\boldsymbol{R}_{u/b}$ | [6.53, 6.68] |
| $H$ | 1.524 | $\boldsymbol{R}_m$ | [1.31, 3.66] |

may not be suitable for mm-wave or THz frequency bands due to fabrication limitations. For mm-wave or THz frequency bands, $N_f$ can be reduced to generate simpler geometries. The smallest value of $N_f$ is 8, in which case each metal layer would be a circular patch or slot. For example, a simpler band-pass unit cell is designed when $N_f$ is set as 16. The designed simpler unit cell is shown in Fig. 19, and its geometric parameters are listed in Table V. A satisfying band-pass performance can be achieved, as shown in Fig. 20. On the contrary, $N_f$ can also be increased to generate more complex geometries for more complicated constraints in low frequency scenarios.

We compared our framework with existing methods in Table VI. In our work, the amount of training data refers to the sum of the amount used for training the SVM (100) and the amount used for auto-evolution (300). References [12], [18] both required human engineers to determine the topology and involve intensive data pre-processing pipelines. Unlike them, our framework auto-selects the suitable topology.

Furthermore, the proposed method requires the smallest amount of training data to arrive at a satisfying design, whereas [21], [22] required a huge amount of training data. The reason is that existing methods often sweep over the whole parameter space to include all possibilities within the training data. We significantly reduce the number of required data thanks to the elaborately organized auto-evolution system. Instead of blindly sweeping over the solution space, our system optimizes the generation of the training data iteratively. The optimized training data have improved informativeness and quality. As a result, the number of required training data is reduced significantly.

TABLE VI
COMPARISON BETWEEN THE PROPOSED FRAMEWORK AND EXISTING METHODS

| | Fully-automated | The amount of training data |
|---|---|---|
| [12] | No | 588 |
| [18] | No | 1 thousand |
| [21] | Yes | 70 thousand |
| [22] | Yes | 21 thousand |
| The proposed work | Yes | $100 + 300$ |

The proposed fully-automated framework can be adopted by designers without domain knowledge to realize the inverse design of FSSs for given EM constraints. Thanks to the auto-evolution system, it takes only hundreds of simulation cycles to arrive at the final design. It offers inspiration for the fully-automated inverse design of other EM components. In future work, we will further integrate filter theory and equivalent circuit approach into machine learning-based methods to increase the interpretability and to further understand the projection between geometries and responses.

## V. CONCLUSION

We developed a fully-automated framework for the inverse design of FSSs. Unlike conventional inverse design methods that rely on experienced human engineers to determine the topology or involve intensive pre-processing pipelines, our framework can automatically read the physical EM constraints and then decide the appropriate topology. Afterward, the auto-evolution would evolve the selected topology into an optimal design. The auto-evolution system consists of three co-operated modules and two self-monitors to optimize the evolution process and maximize its efficiency. We have validated the effectiveness and efficiency of the presented framework in four inverse design scenarios. The validation results proved that our framework frees human engineers and requires a smaller amount of training data compared with existing inverse design methods. The proposed framework can fully automatize the inverse design of FSSs and offer inspiration for the automation of other EM applications.

## REFERENCES


[1] S. Pearson and S. V. Hum, "Optimization of electromagnetic metasurface parameters satisfying far-field criteria," *IEEE Trans. Antennas Propag.*, vol. 70, no. 5, pp. 3477–3488, May 2022, doi: 10.1109/TAP.2021.3137272.

[2] C. Wang, J. Zhang, S. Bai, X. Zhu, and Z. Zheng, "A harmonic suppression energy collection metasurface insensitive to load and input power for microwave power transmission," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 8, pp. 4036–4044, Aug. 2022, doi: 10.1109/TMTT.2022.3182238.

[3] C. Y. Tay and Z. N. Chen, "Azimuthally inhomogeneous metasurface cloak for cylindrical objects," *IEEE Trans. Antennas Propag.*, vol. 69, no. 1, pp. 254–262, Jan. 2021, doi: 10.1109/TAP.2020.3016187.

[4] Y. Wei, J. Duan, H. Jing, Z. Lyu, J. Hao, Z. Qu, J. Wang, and B. Zhang, "A multiband, polarization-controlled metasurface absorber for electromagnetic energy harvesting and wireless power transfer," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 5, pp. 2861–2871, May 2022, doi: 10.1109/TMTT.2022.3155718.

[5] J. Zhang, S. Zhang, and G. F. Pedersen, "Dual-band structure reused antenna based on quasi-elliptic bandpass frequency selective surface for 5G application," *IEEE Trans. Antennas Propag.*, vol. 68, no. 11, pp. 7612–7617, Nov. 2020, doi: 10.1109/TAP.2020.2990225.

[6] W. Tang, S. Mercader-Pellicer, G. Goussetis, H. Legay, and N. J. Fonseca, "Low-profile compact dual-band unit cell for polarizing surfaces operating in orthogonal polarizations," *IEEE Trans. Antennas Propag.*, vol. 65, no. 3, pp. 1472–1477, Mar. 2017, doi: 10.1109/TAP.2016.2647691.

[7] H. Shan, K. Jiang, J. Xing, and T. Jiang, "BPSO and staggered triangle layout optimization for wideband rcs reduction of pixelate checkerboard metasurface," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 7, pp. 3406–3414, Jul. 2022, doi: 10.1109/TMTT.2022.3171519.

[8] K. Yao, R. Unni, and Y. Zheng, "Intelligent nanophotonics: Merging photonics and artificial intelligence at the nanoscale," *Nanophotonics*, vol. 8, no. 3, pp. 339–366, Jan. 2019, doi: 10.1515/nanoph-2018-0183.

[9] F. Güneş, S. Nesil, and S. Demirel, "Design and analysis of minkowski reflectarray antenna using 3-D CST microwave studio-based neural network model with particle swarm optimization," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 23, no. 2, pp. 272–284, Mar. 2013, doi: 10.1002/mmce.20711.

[10] D. R. Prado, J. A. López-Fernández, M. Arrebola, and G. Goussetis, "Support vector regression to accelerate design and crosspolar optimization of shaped-beam reflectarray antennas for space applications," *IEEE Trans. Antennas Propag.*, vol. 67, no. 3, pp. 1659–1668, Mar. 2019, doi: 10.1109/TAP.2018.2889029.

[11] D. R. Prado, J. A. López-Fernández, M. Arrebola, M. R. Pino, and G. Goussetis, "Wideband shaped-beam reflectarray design using support vector regression analysis," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2287–2291, Nov. 2019, doi: 10.1109/LAWP.2019.2932902.

[12] M. Abdullah and S. Koziel, "Supervised-learning-based development of multibit RCS-reduced coding metasurfaces," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 1, pp. 264–274, Jan. 2022, doi: 10.1109/TMTT.2021.3105677.

[13] S. Koziel and M. Abdullah, "Machine-learning-powered EM-based framework for efficient and reliable design of low scattering metasurfaces," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 4, pp. 2028–2041, Apr. 2021, doi: 10.1109/TMTT.2021.3061128.

[14] J. A. Hodge, K. V. Mishra, and A. I. Zaghloul, "RF metasurface array design using deep convolutional generative adversarial networks," in *IEEE Int. Symp. Phased Array Syst. Technol.*, 2019, pp. 1–6.

[15] J. A. Fan, "Generating high performance, topologically-complex metasurfaces with neural networks," in *Conf. Lasers Electro-Opt.*, 2019, pp. 1–2.

[16] P. Naseri, G. Goussetis, N. J. Fonseca, and S. V. Hum, "Inverse design of a dual-band reflective polarizing surface using generative machine learning," in *Eur. Conf. Antennas Propag.*, 2022, pp. 1–5.

[17] L.-Y. Xiao, F.-L. Jin, B.-Z. Wang, Q. H. Liu, and W. Shao, "Efficient inverse extreme learning machine for parametric design of metasurfaces," *IEEE Antennas Wireless Propag. Lett.*, vol. 19, no. 6, pp. 992–996, Jun. 2020, doi: 10.1109/LAWP.2020.2986023.

[18] R. Zhu, J. Wang, Y. Han, S. Sui, T. Qiu, Y. Jia, M. Feng, X. Wang, L. Zheng, and S. Qu, "Design of aperture-multiplexing metasurfaces via back-propagation neural network: Independent control of orthogonally-polarized waves," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 4569–4575, Jun. 2022, doi: 10.1109/TAP.2022.3140523.

[19] 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 Trans. Antennas Propag.*, vol. 70, no. 7, pp. 5120–5129, Jul. 2022, doi: 10.1109/TAP.2022.3152592.

[20] P. Naseri and S. V. Hum, "A generative machine learning-based approach for inverse design of multilayer metasurfaces," *IEEE Trans. Antennas Propag.*, vol. 69, no. 9, pp. 5725–5739, Sep. 2021, doi: 10.1109/TAP.2021.3060142.

[21] P. Naseri, S. Pearson, Z. Wang, and S. V. Hum, "A combined machine-learning/optimization-based approach for inverse design of nonuniform bianisotropic metasurfaces," *IEEE Trans. Antennas Propag.*, vol. 70, no. 7, pp. 5105–5119, Jul. 2022, doi: 10.1109/TAP.2021.3137496.

[22] C. C. Nadell, B. Huang, J. M. Malof, and W. J. Padilla, "Deep learning for accelerated all-dielectric metasurface design," *Optics Express*, vol. 27, no. 20, pp. 27 523–27 535, Sep. 2019, doi: 10.1364/OE.27.027523.

[23] M. Ohira, H. Deguchi, M. Tsuji, and H. Shigesawa, "Multiband single-layer frequency selective surface optimized by genetic algorithm with geometry-refinement technique," in *IEEE Antennas Propag. Soc. Int. Symp.*, 2003, pp. 833–836.

[24] M. Gingrich and D. H. Werner, "Synthesis of zero index of refraction metamaterials via frequency-selective surfaces using genetic algorithms," in *IEEE Antennas Propag. Soc. Int. Symp.*, 2005, pp. 713–716.

[25] L. Lanuzza, A. Monorchio, and G. Manara, "Synthesis of zero index of refraction metamaterials via frequency-selective surfaces using genetic algorithms," in *IEEE Antennas Propag. Soc. Int. Symp.*, 2002, pp. 364–367.

[26] M. O. Akinsolu, K. K. Mistry, B. Liu, P. I. Lazaridis, and P. Excell, "Machine learning-assisted antenna design optimization: A review and the state-of-the-art," in *Eur. Conf. Antennas Propag.*, 2020, pp. 1–5.

[27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.

[28] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Int. Conf. Mach. Learn.*, 2012, pp. 1881–1888.

[29] R. Bro and A. K. Smilde, "Principal component analysis," *Anal. Methods*, vol. 6, no. 9, pp. 2812–2831, Mar. 2014, doi: 10.1039/C3AY41907J.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Lang.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1023/A:1022627411411.

[31] *High Frequency Electronics Product Selector Guide*, Rogers Corporation, 2021, pp. 14–15.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Repr.*, Y. Bengio and Y. LeCun, Eds., 2015, pp. 1–15.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Int. Conf. Mach. Learn.*, 2010, pp. 1–8.

[35] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE Int. Conf. Evol. Comput. Proc.*, 1998, pp. 69–73.

[36] L. Zhou and Z. Shen, "Hybrid frequency-selective rasorber with low-frequency diffusion and high-frequency absorption," *IEEE Trans. Antennas Propag.*, vol. 69, no. 3, pp. 1469–1476, Mar. 2021, doi: 10.1109/TAP.2020.3018537.

[37] A. Lalbakhsh, M. U. Afzal, K. P. Esselle, and S. L. Smith, "All-metal wideband frequency-selective surface bandpass filter for te and tm polarizations," *IEEE Trans. Antennas Propag.*, vol. 70, no. 4, pp. 2790–2800, Apr. 2022, doi: 10.1109/TAP.2021.3138256.

[38] Z.-C. Hao and J.-S. Hong, "UWB bandpass filter using cascaded miniature high-pass and low-pass filters with multilayer liquid crystal polymer technology," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 4, pp. 941–948, Apr. 2010, doi: 10.1109/TMTT.2010.2042632.

[39] Y.-J. Huang, C.-H. Hsieh, and Z.-M. Tsai, "A compact, high-selectivity, and wide passband semi-lumped 70 MHz high-pass filter," in *2017 IEEE Asia Pac. Microw. Conf.*, 2017, pp. 730–733.

[40] S. M. A. M. H. Abadi and N. Behdad, "Wideband linear-to-circular polarization converters based on miniaturized-element frequency selective surfaces," *IEEE Trans. Antennas Propag.*, vol. 64, no. 2, pp. 525–534, Feb. 2016, doi: 10.1109/TAP.2015.2504999.

[41] S. V. Samsonov, A. A. Bogdashov, and I. G. Gachev, "Waveguide linear-to-circular polarization converter with cross polarization below $-40$dB within 16% band," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 4, pp. 2108–2114, Apr. 2022, doi: 10.1109/TMTT.2022.3145062.