A Generative Model-Based Method for Inverse Design of Microstrip Filters

Haipeng Wang [1,2], Chenchen Nie [1], Zhongfang Ren [1] and Yunbo Li [2,*]

Supplementary Material

S1. Detailed information of the geometric training datasets

S2. Detailed information of the network architecture

S3. Detailed information of the genetic algorithm optimizer

S4. Additional samples of inverse design test of single-band filters under the target curves in the dataset.

S5. Additional samples of inverse design test of dual-band filters under the target curves in the dataset.

S6. Additional samples of inverse design test of single-band filters under the self-defined objective.

S7. Additional samples of inverse design test of dual-band filters under the self-defined objective.

S8. Detailed information of the fabrication and measurement

S1. Detailed information of the geometric training datasets

The microstrip filters designed in this study employ a square patch resonator structure, where each resonator's physical configuration corresponds to a 32×32 binary pixel image. In this image, "0" indicates the presence of a copper patch, while "1" indicates no metal coverage. To enrich the geometric dataset of the filters, twenty different geometric shapes are introduced as slot perturbations for the resonators. These shapes include arcs, H-shapes, L-shapes, ellipses, rectangles, sectors, crosses, plus signs, crosses with ellipses, crosses with rectangles, and various rotated angles of these shapes, totaling forty-seven geometric variations as illustrated in Figure S1. Using a binarization approach and appropriate steps in MATLAB, a series of binary pixel images corresponding to square patches are generated. Each image represents a different filter structure with distinct resonant frequencies and bandwidths, forming the geometric training dataset for the GAN network, comprising 12,590 samples.

Using ANSYS HFSS 2019 software, batch electromagnetic simulations were conducted on the filters corresponding to binary pixel images. The simulation results include collected transmission coefficients (S_{21}) and reflection coefficients (S_{11}). Single-band and dual-band filter models were simulated within the frequency ranges of 3-8GHz and 1-6 GHz respectively. Each model's simulation yielded 64 data points for transmission coefficients and 64 data points for reflection coefficients, establishing an overall electromagnetic parameter database for inverse design algorithms.Ultimately, a predictor training dataset consisting of 12,590 samples was constructed. This dataset includes geometric training data represented by 32×32 binary pixel images as input, with electromagnetic simulation parameters of corresponding models as output data. The specific definitions of the single layer pattern are illustrated in Fig. S4.

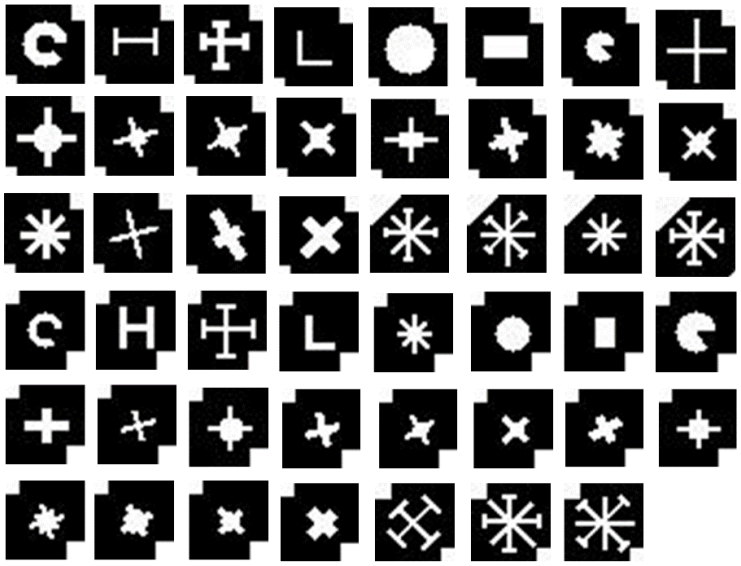


Figure S1. A 32×32 binary pixel image corresponding to the square patch resonator structure, where "0" indicates the presence of copper metallization and "1" indicates the absence of patch coverage.

S2. Detailed information of the network architecture

Figure S2 illustrates the detailed neural network architecture based on Generative Adversarial Networks (GANs), comprising a generator (*G*) and a discriminator (*D*), both constructed using convolutional neural networks (CNNs). The *G* and *D* form the compositional pattern-producing network (CPPN) GAN network.

S2.1 Generator *G*

The generator *G* takes a 128×1 latent vector input, sampled from a Gaussian distribution ($z \sim N(0,1)$), as depicted in Figure S2(a). It employs three Conv2DTranspose operations to upsample and generate new image information. This Conv2DTranspose operation is the transpose of the matrix representation in the Conv2D operation. Consequently, after applying three Conv2DTranspose layers, the resulting tensor has dimensions (1, 32, 32), representing a single-channel image of size 32 × 32 pixels. In the generator, except for the final layer which uses a "Sigmoid" activation function, each convolutional layer employs a "ReLU" activation function.

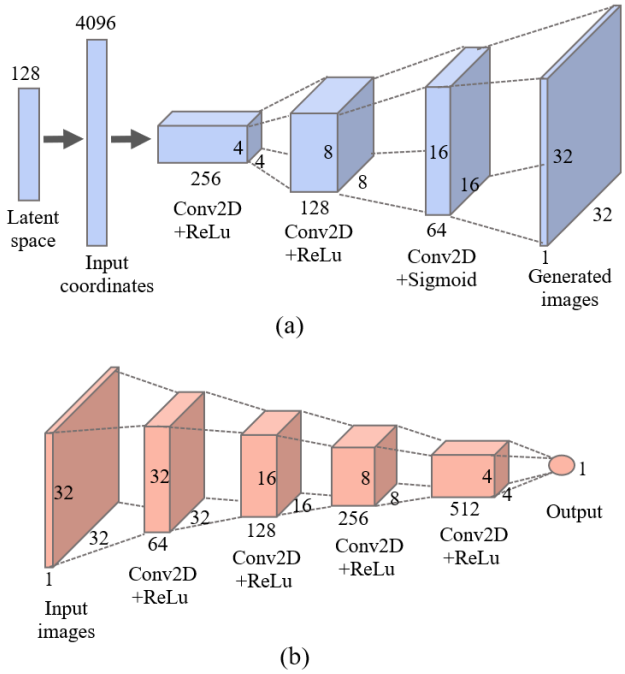


Figure S2. Neural network architectures of the generator and the discriminator. (a) The detailed architecture of the designed generator, which is consists of one linear layer and three transpose convolutional operation (Conv2DTranspose) layers. The generator transforms the input latent vector to the single layer pattern with the dimension of 32×32. (b) The network architecture of the discriminator in the training of compositional pattern-producing network (CPPN) GAN with gradient penalty (CPPN-GAN). The discriminator is constituted of four cascaded convolutional operation (Conv2D) layers. The output of the discriminator is utilized to measure the Euclidean distance between the distribution of the generated patterns and the geometric dataset.

S2.2 Discriminator *D*

The structure of discriminator *D* is illustrated in Figure S2(b), with inputs consisting of 1-channel images of size 32 × 32 pixels from both the generator and geometric dataset,

represented as tensors of dimensions (1, 32, 32). Through four consecutive Conv2D operations, the tensor is transformed into a one-dimensional form for binary classification (real or fake). A ReLU activation function is utilized within each convolutional layer of the discriminator. The discriminator's output is utilized to measure the Euclidean distance between the generated pattern distribution and the geometric dataset. Details of the structural parameters for the generator and discriminator are listed in Table S1.

Table S1. Structures and training details of adopted CPPN -GAN.

Details of CPPN -GAN		
Generator		
Layer Name	Output Feature	Layer Details
Linear 1	128	Input latent vector
Linear 2	4096	Fully connected layer
ReLu	4096	ReLu activation
Linear 3	256×4×4	Reshape
ConvTransposed2d	128×8×8	Transposed convolution layer
ReLu	128×8×8	ReLu activation
ConvTransposed2d	64×16×16	Transposed convolution layer
ReLu	64×16×16	ReLu activation
ConvTransposed2d	1×32×32	Transposed convolution layer
Sigmoid	1×32×32	Sigmoid activation
Final reshape	32×32	Reshape to output size
Discriminator		
Conv2d	64×32×32	Convolution layer
ReLu	64×32×32	ReLu activation
Conv2d	128×16×16	Convolution layer
ReLu	128×16×16	ReLu activation
Conv2d	256×8×8	Convolution layer
ReLu	256×8×8	ReLu activation
Conv2d	512×4×4	Convolution layer
ReLu	512×4×4	ReLu activation
Linear	1	Fully connected layer
Hyperparameters		
Dimension of the latent vector		128
Number of discriminator iterations per generator iteration		5
Batch size		64
Lamda		10
Learning rate of the discriminator		0.00005
Learning rate of the generator		0.00005
Optimizer (Adam)		$ß_1$= 0.5, $ß_2$= 0.9
Training iterations		400000

To assess the effectiveness of the CPPN-GAN network, we employ the Euclidean distance to measure the distance between the fitted distribution of the model and the target distribution. The training results of the CPPN-GAN network are illustrated in Figure S3.

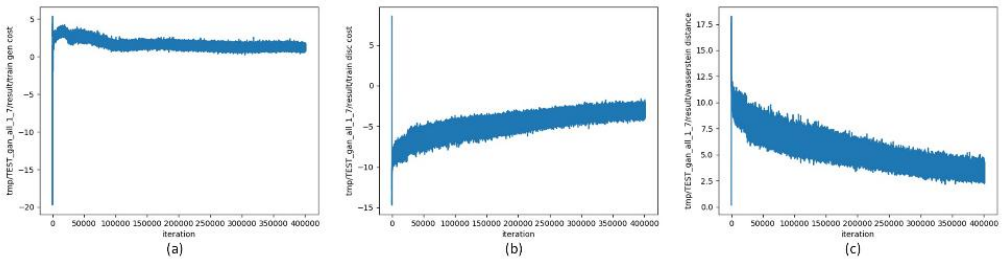


Figure S3. The training results of the CPPN-GAN network. (a) The variation of the generator loss. (b) The discriminator loss. (c) Euclidean distance. During the training of GAN network, the average CPPN-GAN is converged to less than 5 after 400000 iterations.

S2.3 Predictor *P*

To efficiently calculate the performance of filters, a predictor that can approximate the transmission/ reflection coefficients without using EM simulations is needed to develop. Due to the excellent performance of adjusting network depth according to the task requirements and the dataset size, the ResNet18 model is chosen to build the EM predictor. The predictor takes one 32 × 32 pixel pattern as input and outputs the coefficients within the target frequency range. Thus, normalized S21 and S11within 0 to 1 are all sampled at 64 points and the output dimension is 2 × 64 = 128. Here, in order to improve the training accuracy, two predictors are separately introduced in the proposed inverse design for the single-band and dual-band filter models, which are simulated and trained within the frequency ranges of 3-8 GHz and 1-6 GHz respectively. The network architecture and training hyperparameters of the predictor are illustrated in Figure S4 and Table S2.

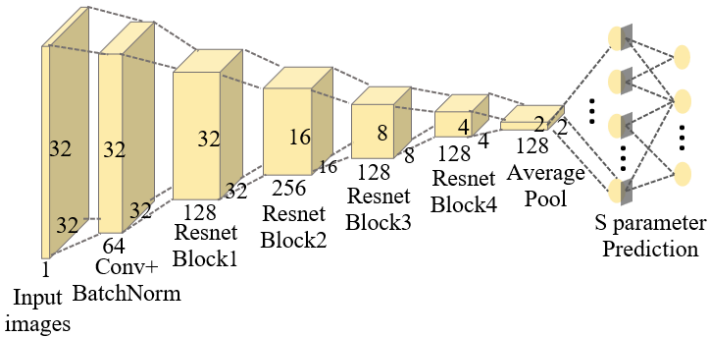


Figure S4. The illustration of network architecture of predictor network based on Resnet18 model.

Table S2. Structures and training details of adopted neural network Predictor.

Details of the Predictor		
ResNet Predictor		
Layer Name	Output Feature	Layer Details
Conv1	64×32×32	With BatchNorm
ResnetBlock1	128×32×32	Stride 2
ResnetBlock2	256×16×16	Stride 2
ResnetBlock3	128×8×8	Stride 2
ResnetBlock4	128×4×4	Stride 2
Average Pool	128×2×2	2×2 average pool
Linear 1	512	Flatten layer
Linear 2	1024	Fully connected layer
ReLu	1024	ReLu activation
Linear 3	512	Fully connected layer
ReLu	512	ReLu activation
Linear 4	256	Fully connected layer
Linear 5	136	Fully connected layer
Hyperparameters		
Dimension of the input vector		1×32×32
Learning rate		0.0003
Batch size		64
Training iterations		1000

S3. Detailed information of the genetic algorithm optimizer

The fundamental principle of Genetic Algorithms (GA) involves simulating the evolutionary process of an artificial population to solve a given problem. Mechanisms such as Selection, Crossover, and Mutation are employed iteratively to maintain a set of candidate solutions. Over successive iterations, the population evolves through several generations, ideally converging to a state of near-optimal fitness. Implementation of genetic algorithms typically includes steps such as encoding, generating initial populations, computing fitness functions, performing selection, crossover, mutation operations, and evaluating termination conditions.

The first step in implementing a genetic algorithm is to determine the encoding method for solving the problem, mapping the solution space to the gene space. Common encoding techniques include real-valued encoding, binary encoding, integer encoding, and data structure encoding. Real-valued encoding directly represents genes as real numbers, which is straightforward and avoids the need for decoding. Therefore, this study adopts real-valued encoding. The GAN network maps geometric shapes corresponding to filters to a 128-dimensional latent vector, encoding each image directly with 128 real numbers. The initial population is generated by randomly creating a certain number of initial solutions. The fitness function evaluates the fitness or quality of each individual, progressively eliminating individuals with lower fitness values while increasing those with higher fitness values through multiple generations of evolution, ultimately yielding the maximum fitness value as the optimal solution to the problem. Selection is based on the fitness values to choose individuals for the next generation, while crossover and mutation operations generate new individuals. Termination conditions determine when the algorithm ceases operation.

In the implementation of genetic algorithms, the fitness function plays a crucial role. This study employs different fitness functions for inverse design of database target curves and custom target curves. For the database target curves, the chosen fitness function is the $L2$ norm between the target S parameters and the generated S parameters from inverse design, representing the Euclidean distance between the distributions of the two datasets, defined as follows:

$$L2 = -\| T_{\text{goal}} - T_{generated} \|_2 = -\sqrt{\frac{\sum_{i=1}^{n} (T_{goal}(f_i) - T_{generated}(f_i))^2}{n}} \quad (1)$$

Here, n denotes the total size of the transmission coefficient data points, and f_{i} represents the i-th frequency data point.

For the custom target curve, the entire frequency band of the filter is divided into several sub-bands, each with custom thresholds for S_{11} and S_{21}. As shown in Equation (2), the loss function L comprises the reflection coefficient loss LS_{11} and transmission coefficient loss LS_{21}, where γ is a scaling factor adjusting the loss function according to different design objectives. In Equations (3) and (4), S_{11} and S_{21} represent the reflection and transmission coefficients, respectively, simulated by the inverse design pattern. $S_{11\min}$ and $S_{21\min}$ denote the minimum desired values for reflection and transmission coefficients, while $S_{11\max}$ and $S_{21\max}$ denote the maximum desired values. Boldface indicates vectors, and (*)T denotes the transpose operator.

$$L = \gamma \times LS_{11} + (1-\gamma) \times LS_{21} \quad (2)$$

$$LS_{11} = (S_{11} - S_{11\min})(S_{11} - S_{11\max})^{\mathrm{T}} + \left|(S_{11} - S_{11\min})(S_{11} - S_{11\max})^{\mathrm{T}}\right| \quad (3)$$

$$LS_{21} = (S_{21} - S_{21\min})(S_{21} - S_{21\min})^{\mathrm{T}} + \left|(S_{21} - S_{21\min})(S_{21} - S_{21\max})^{\mathrm{T}}\right| \quad (4)$$

The flowchart of the GA optimizer used in the inverse design of the single-band and dual-band microstrip filters is shown in Figure S5.

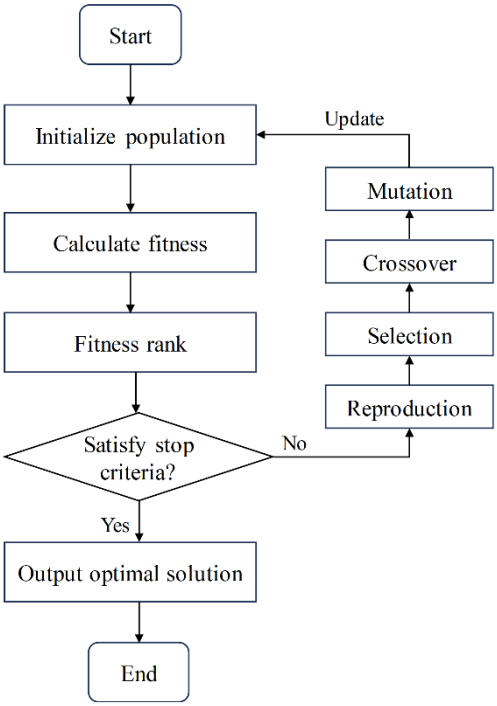


Figure S5. The flowchart of the genetic algorithm (GA) optimizer used in the inverse design of microstrip filters.

S4. Additional samples of inverse design test of single-band filters under the target curves in the dataset.

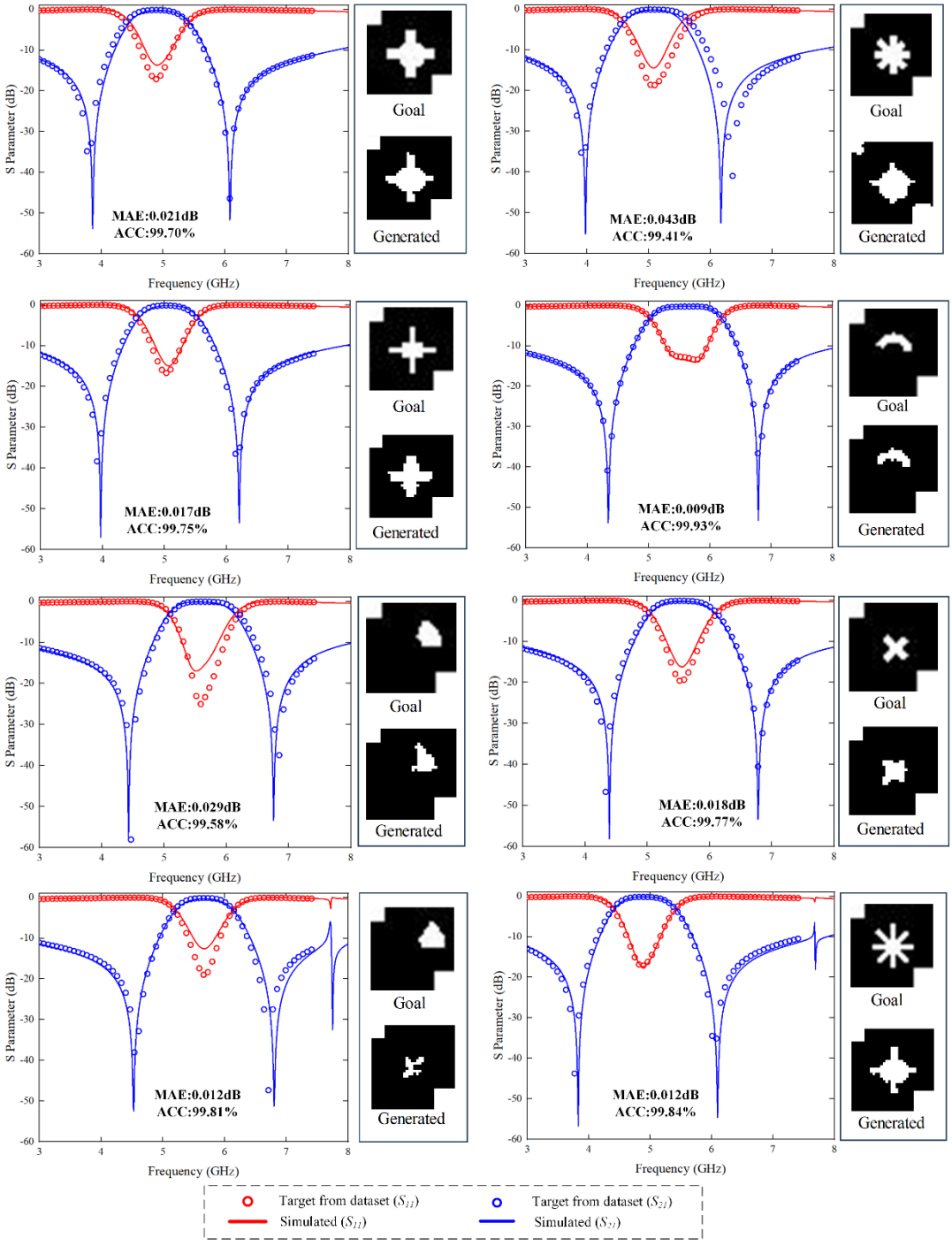


Figure S6. Some additional samples of inverse design test of single-band filters under the target curves in the dataset.

S5. Additional samples of inverse design test of dual-band filters under the target curves in the dataset.

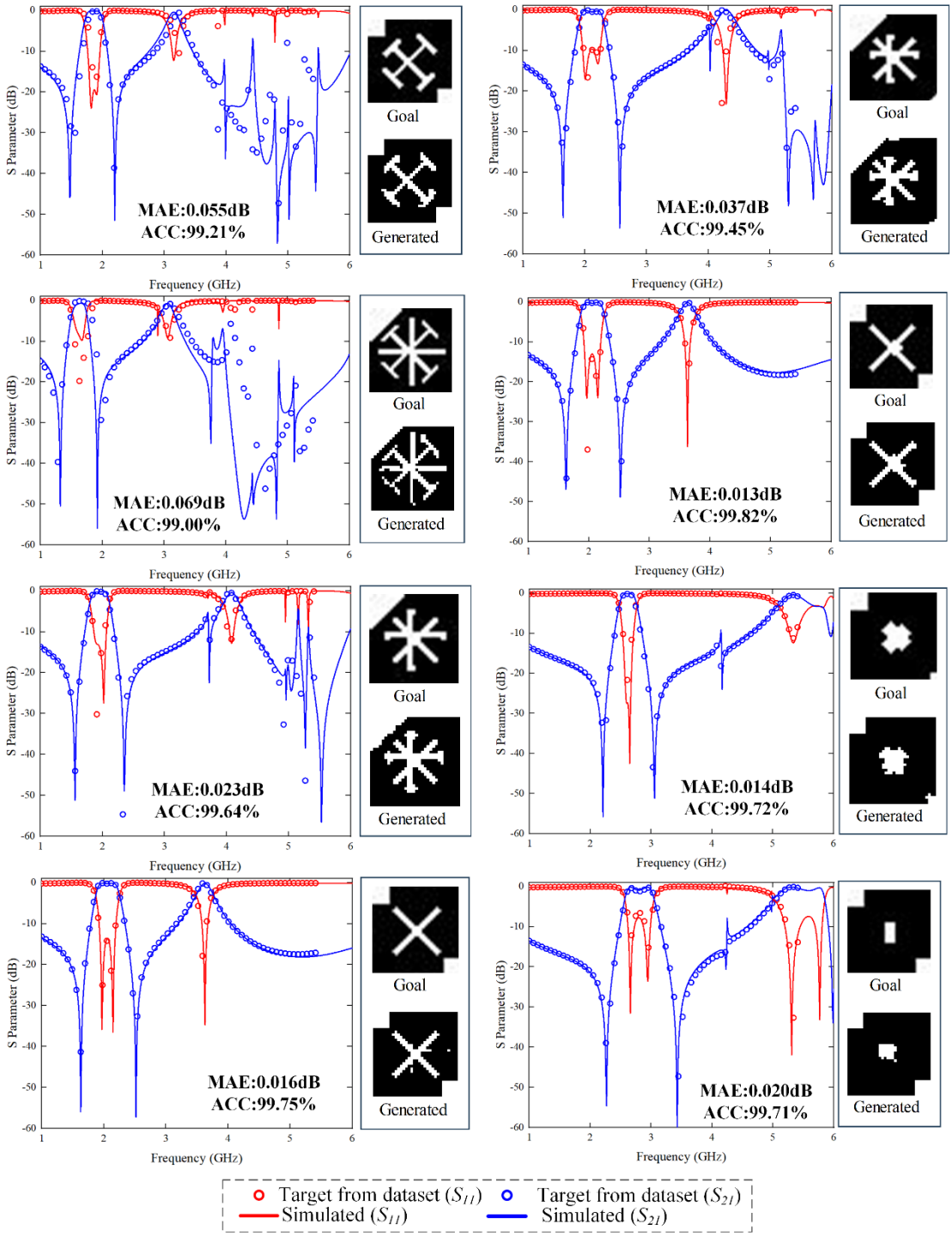


Figure S7. Some additional samples of inverse design test of dual-band filters under the target curves in the dataset.

S6. Additional samples of inverse design test of single-band filters under the self-defined objective.

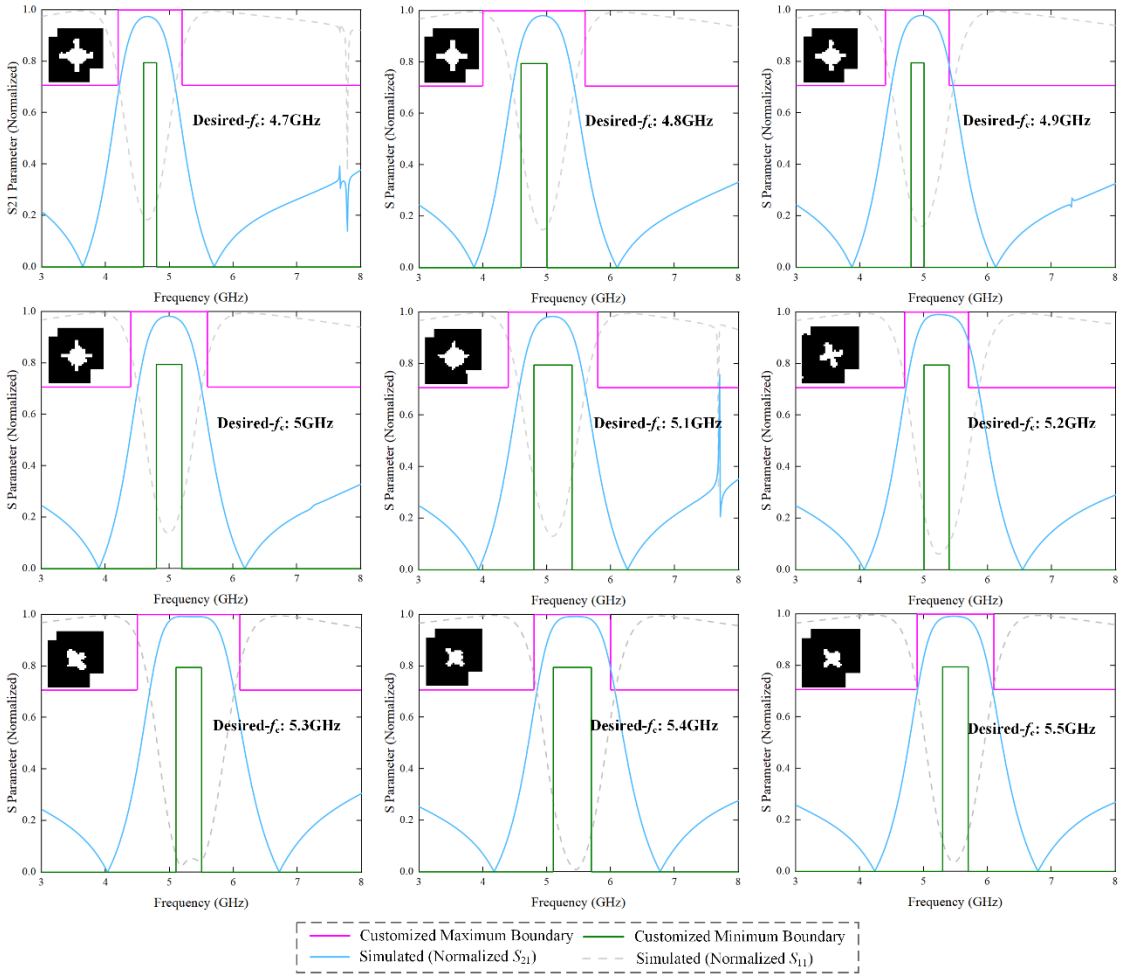


Figure S8. Some additional samples of inverse design test of single-band filters under the self-defined objective.

S7. Additional samples of inverse design test of dual-band filters under the self-defined objective.

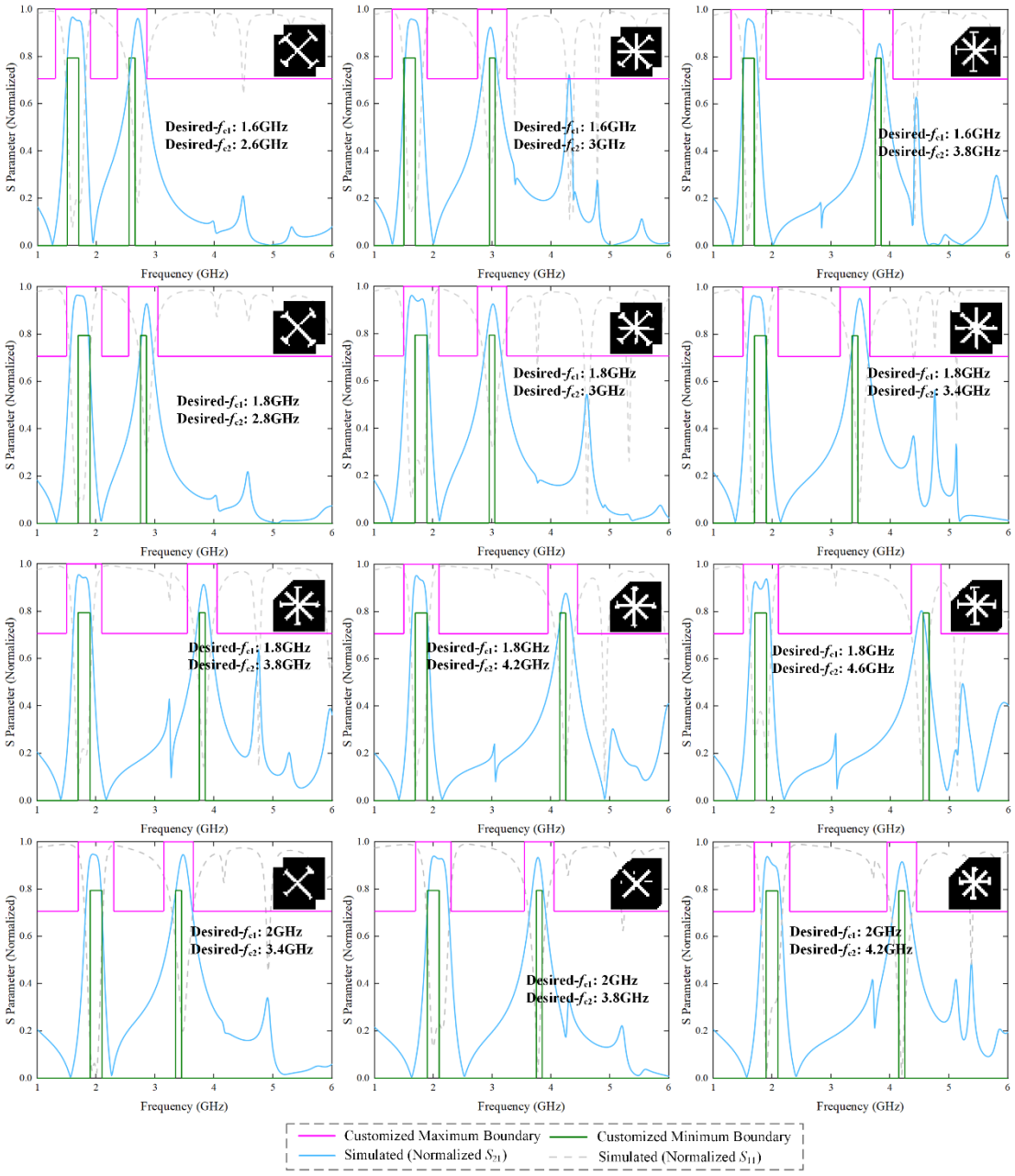
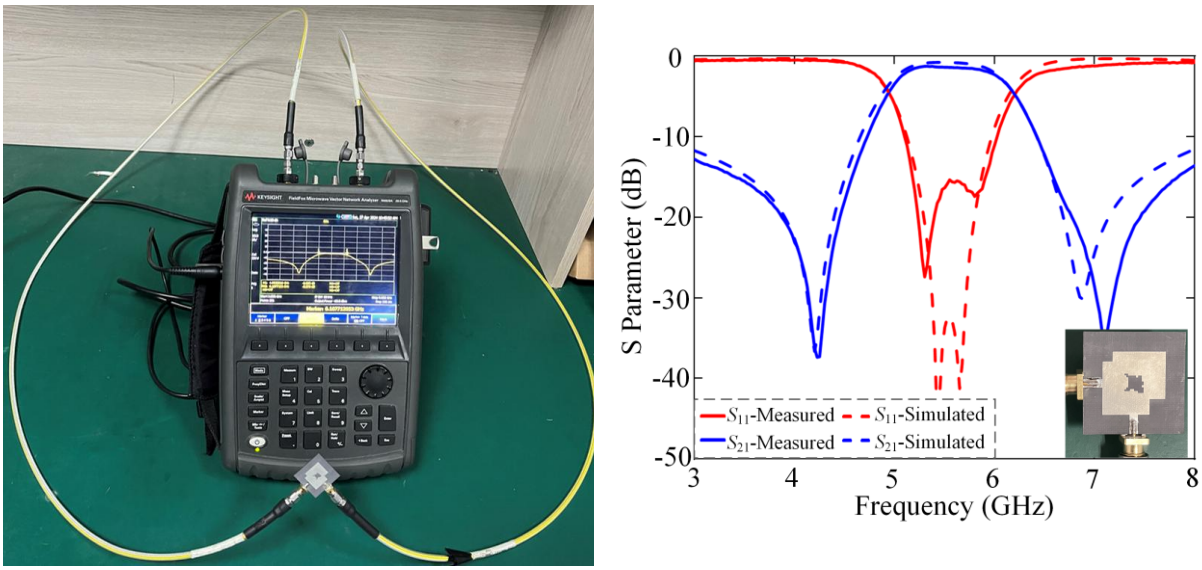


Figure S9. Some additional samples of inverse design test of dual-band filters under the self-defined objective.

S8. Detailed information of the fabrication and measurement

The passive filter was fabricated on an F4B substrate with a relative permittivity of 2.65, loss tangent of 0.0013, and thickness of 0.8 mm. The top layer features copper metallization traces, while the bottom layer is entirely coated with copper serving as the reference ground plane. During fabrication, a tin immersion surface treatment was employed to prevent oxidation of the copper surface in ambient air and to provide enhanced corrosion resistance.

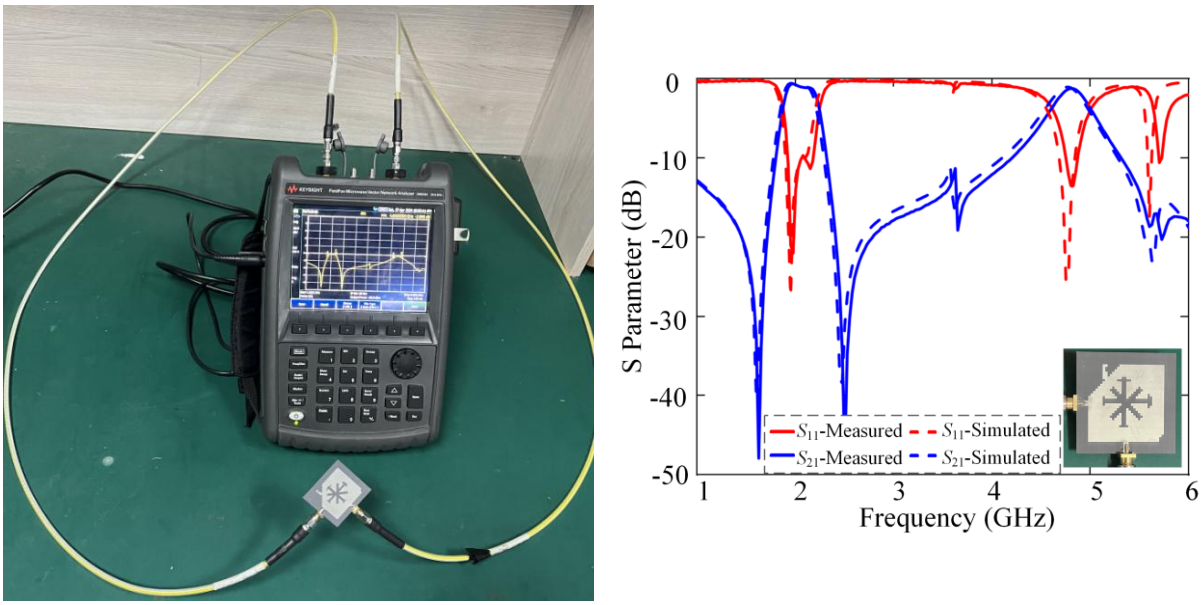
The filter was characterized using a FieldFox portable vector network analyzer (VNA) from Keysight Technologies. Prior to measurements, the VNA was calibrated to eliminate systematic errors. The filter's input and output ports were connected to Port 1 and Port 2 of the VNA via SMA connectors, respectively. The frequency sweep range was set according to the designed operating bandwidth of the filter. Scattering parameters S11 and S21 were measured sequentially, and the obtained data were saved in CSV format for further analysis. The experimental setup is illustrated in Figure S10 and Figure S11.



(a) **(b)**

Figure S10. (a) Measurement setup for the single-band filter; (b) Measured response of the single-band filter.

As illustrated in Figure S10, the single-band filter was characterized over a frequency range of 3–8 GHz. The measured center frequency was 5.56 GHz with a 3-dB bandwidth of 1.06 GHz, while the simulated center frequency was 5.53 GHz with a 3-dB bandwidth of 1.12 GHz, demonstrating good agreement between simulated and measured *S*-parameters. Experimental results indicate an insertion loss of less than 1.23 dB and a return loss better than 15.3 dB within the passband. Two transmission zeros were observed on each side of the passband, contributing to enhanced selectivity. The out-of-band rejection reached 38.5 dB, and the shape factor was 2.83, confirming excellent frequency selectivity.



(**a**) (**b**)

Figure S11. (a) Measurement setup for the dual-band filter; (b) Measured response of the dual-band filter.

Figure S11 presents a comparative analysis of simulated and measured results across the 1-6 GHz frequency range, showing excellent agreement in *S*-parameters. The lower passband exhibited measured/simulated center frequencies of 2.05/2.03 GHz with 360/340 MHz 3-dB bandwidths, while the upper passband demonstrated 4.8/4.77 GHz center frequencies with 320/360 MHz bandwidths. Both passbands maintained insertion losses below 1.19 dB and return losses exceeding 10.1 dB, featuring two transmission zeros on each side. Notably, the lower passband achieved superior performance with 47.8 dB out-of-band rejection and a 2.5 shape factor, compared to the upper passband's 23 dB rejection and 5.5 shape factor, clearly indicating enhanced frequency selectivity in the lower frequency range.