A High-Quality Data Acquisition Method for Machine-Learning-Based Design and Analysis of Electromagnetic Structures

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

Published in:
IEEE Transactions on Microwave Theory and Techniques

DOI (link to publication from Publisher):
10.1109/TMTT.2023.3259477

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Zhou, Z., Wei, Z., Tahir, A., Ren, J., Yin, Y., Pedersen, G. F., & Shen, M. (2023). A High-Quality Data Acquisition Method for Machine-Learning-Based Design and Analysis of Electromagnetic Structures. *IEEE Transactions on Microwave Theory and Techniques*, *71*(10), 4295-4306. https://doi.org/10.1109/TMTT.2023.3259477

A High-Quality Data Acquisition Method for Machine Learning Based Design and Analysis of Electromagnetic Structures

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

***Abstract*—Electromagnetic structures play a significant role in wireless communication, radar detection, medical imaging, etc. Machine learning has been increasingly applied to facilitate the design and analysis of electromagnetic structures. Data acquisition is a major bottleneck. Conventional methods blindly sweep geometric parameters on a uniform grid and acquire corresponding responses via simulation. Acquired data have unstable quality due to inconsistent informativeness of responses, leading to a low ratio of model performance to data amount. This paper proposes a high-quality data acquisition method to increase the ratio of model performance to data amount. It anticipates and generates high-quality data by analyzing distribution of existing data iteratively. Comparative analysis of four implementations proves, the proposed method reduces required data amount by around $40\,\%$ for the same model performance, hence saves around $40\,\%$ simulation and computing resources. The proposed method benefits machine learning applications of metasurfaces, antennas, and many other microwave structures.**

***Index Terms*—Data acquisition, electromagnetic structures, full-wave EM simulation, machine learning, metasurface.**

I. INTRODUCTION

WIRELESS communication is substantially impacted by electromagnetic (EM) structures. EM structures are deliberately designed to satisfy EM constraints in a practical scenario. EM constraints refer to requirements for the structures, such as size, operating frequency, gain, axial ratio (AR), reflection, transmission, scattering coefficient, etc. Accordingly, EM response describes how a structure affects wireless signals within the concerning frequency range. Conventional design of EM structures relies on experienced human engineers. Firstly, engineers determine a topology and initialize its geometric parameters to form a draft structure for the topology according to the EM constraints and their experience. Afterward, they evaluate its EM response through full-wave simulation via EM simulation software (for example, Computer Simulation Technology®(CST)) and tune its geometric parameters based on their understandings of the correlation between parameters and EM responses iteratively. The number of iterations needed varies, depending on the designer's experience depth. Therefore, machine learning (ML) is increasingly being studied and applied in EM applications to improve the current EM solutions. However, data acquisition is a significant barrier for ML [1], especially for EM-related ML applications.

Most EM-related ML applications require [2]–[4] labeled data set consists of geometric parameters and corresponding EM responses. EM-related ML applications can be roughly divided into three categories, forward synthesis, inverse design, and generative method [5]–[14]. Forward synthesis establishes a model to imitate the projection from geometric parameters to EM responses [15]–[22]. The forward model utilizes supervised learning because both geometric parameters and corresponding EM responses are needed as the training data set. After training, the model is used to replace EM simulation software and is often integrated with optimization algorithms to reach an optimal structure design. To start with, a draft structure is initialized with arbitrary geometric parameters, and the forward model synthesizes its responses. Afterward, the optimization algorithm updates the geometric parameters based on the difference between current responses and constraints. Inverse design develops a model to directly determine geometric parameters for given EM constraints, which also uses supervised learning [23]–[27]. The well-trained model acts as a dictionary that records the projection from EM responses to suitable geometric parameters. Generative method utilizes autoencoder (VAE) or generative adversarial network (GAN) to learn characteristics of real geometric parameters [28]. After training, the generative model referred to as generator can generate new structures with similar geometric parameters. However, the generated new structure still requires full-wave simulation to evaluate its EM responses. To reduce the need for full-wave simulation, the generator is sometimes integrated with a forward synthesis model in the real design

Manuscript received 21 December 2022; revised 13 February 2023; accepted 11 March 2023. Date of publication * * 2023; date of current version * * 2023. This work was supported in part by the China Scholarship Council under Grant 202006960011; in part by the National Natural Science Foundation of China under Grant No. 62071351; in part by the 111 Project of China; and in part by the Fundamental Research Funds for the Central Universities and Innovation Fund of Xidian University. (*Corresponding authors: Jian Ren; Ming Shen.*)

Zhao Zhou, Zhaohui Wei, Gert Frølund Pedersen, and Ming Shen are with the Department of 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).

Abdullah Tahir is with the Department of the Electronic Systems, Aalborg University, 9220 Aalborg, Denmark and also with the Mechatronics Department of University of Engineering and Technology Lahore, Faisalabad Campus (e-mail: atah@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.

process. To start with, a noise vector is randomly initialized as the first input for the generator, and the generator forms an initial structure accordingly. The EM responses of the structure are then evaluated using a forward synthesis model. By comparing its responses and constraints, the optimization algorithm updates the noise vector accordingly. The updated vector is taken as a new input for the generator to generate an updated structure, and it is then evaluated and updated again and again. The iteration stops when the updated structure fulfills the EM constraints.

There are two ways to acquire labeled training data for EM-related ML applications. The first way is to fabricate a prototype with respect to each setting of geometric parameters and then measure its EM responses to form each training data sample. Measuring EM responses require specific measurement devices, such as Vector Network Analyzers for reflection or transmission coefficients and anechoic chamber for gain or AR. To collect N training data samples, the designer need to adjust geometric parameters of the prototype and measure responses for N times if the prototype is adjustable, otherwise the designer has to fabricate N prototypes and measure their responses individually. Since acquiring EM data through measurement is costly and restricted, designers normally use full-wave simulation for collecting EM training data. Supported by simulation software such as CST, designers can build a virtual prototype and simulate its responses by using a computer. Full-wave simulation does not require fabrication or measurement devices, but relies on computation resources.

ML applications in EM are often criticized because it needs a large number of full-wave simulation cycles to generate sufficient training data, hence occupies many computation resources. Most works define all settings of geometric parameters first and then simulate corresponding EM responses individually to collect the training data set. The defined geometric parameters usually distribute on a uniform grid within the parameter space, because designers cannot anticipate distribution of responses and the best policy is to uniformly cover the whole parameter space. However, EM responses are extremely sensitive to geometric parameters and do not distribute uniformly. With respect to different areas within the parameter space, the changes of EM responses may be slightly or significantly. Samples in parameter areas where the EM responses change significantly can greatly affect the model performance. These samples are referred to as high-quality data. By contrast, those in parameter areas where EM responses change slightly contribute little to the model performance, which are referred to as low-quality data. Low-quality samples occupy a number of unnecessary simulation cycles. However, designers cannot recognize low-quality data before simulation and avoid unnecessary simulation cycles.

An intelligent high-quality data acquisition method is demanded for EM-related ML. However, most data acquisition methods [29]–[33] are not suitable for EM-related ML applications. There have been many great works [34]–[42] that attempted to identify the most promising region of the parameter space and further tune the design by means of local routines. They improved the global optimization of expensive EM simulation models significantly. They focused on fast convergence of the promising region and the optimal design, instead of generation of a high-quality dataset for the ML model. A high-quality dataset should be informative and representative for the whole parameter space, so that the ML model can learn the intelligence of the whole parameter space. A high-quality data acquisition method is expected to generate a more informative and representative dataset with the smallest amount of samples. It helps ML models obtain the same performance by using a smaller amount of training data, resulting in the reduction of burden on simulation.

This paper proposes a high-quality data acquisition method. The objective is to improve the quality of data and reduce the need of simulation for ML-based design of EM structures. Quality of data for ML is measured based on the performance of the ML model. High-quality data can improve the performance of the ML model significantly. To start with, a small amount of training data samples are initialized, which are defined on a uniform grid within the parameter space. Existing data samples are analyzed with respect to the distribution of parameters and responses to recognize a parameter area where the EM responses change significantly. Afterward, a new data sample is generated by defining its geometric parameters through swarm operation in the selected area, and its EM responses are simulated through simulation. The new data sample is considered of high quality, because it is likely to improve the performance of the ML model significantly. The new high-quality data sample is then added into the existing data samples, and a new round of analysis and generation begins. The existing data samples are iteratively analyzed and expanded, and an increasing number of high-quality data samples are generated. The iteration stops when sufficient data samples have been collected. Unlike conventional data acquisition methods that uniformly sweep geometric parameters on a constant grid of the parameter space, the proposed method adjusts parameter definition dynamically according to the quality of parameter area by analyzing the distribution of existing samples. The proposed method can maximize the quality of training data set with a reduced amount of simulation cycles. Based on the comparative results in four implementations, the proposed method significantly reduces the amount of training data samples required to reach the same model accuracy, hence a significant amount of simulation cycles are saved, and computation resources are greatly released.

The remaining content is arranged as follows: Section II introduces the algorithm of the proposed method; Section III validates the proposed method in four implementations; Section IV gives the conclusion.

II. ALGORITHM

Pseudo code for the proposed method is demonstrated in Algorithm 1. The proposed algorithm is established specially according to the requirements of data acquisition in the ML-based design of EM structures. Input features of this ML task in EM are represented as normalized vectors $\boldsymbol{X}$s, whereas, $\boldsymbol{Y}$s represent the output features. Here, each $\boldsymbol{Y}$ is obtained by full-wave EM simulation via CST for a given $\boldsymbol{X}$. The proposed method comprises of two major steps. The first step is to

Algorithm 1 The proposed data acquisition method

Require:

Variables to be fixed:
$\boldsymbol{X}$, $\boldsymbol{Y}$: normalized input, output vector
T: integer, maximum data acquisition iteration
$\boldsymbol{X}_{step}$: vector, minimum step of input features
min_loss: float, expected minimum loss
N_0: integer, number of initial data samples
Built-in variables and functions:
i: integer, index of elements in $\boldsymbol{X}$
j: integer, index of sample within the data set
t: integer, index of data acquisition iteration
$\{(\hat{\boldsymbol{X}}_j, \hat{\boldsymbol{Y}}_j)\}$: reference samples
N_t: integer, number of existing samples in iteration t
k: integer, index of selected sample
c_t: float, between 0 and 1, depend on t
($\boldsymbol{X}_*$, $\boldsymbol{Y}_*$): new sample
model_loss: float, model loss after training
Simulate: full-wave EM simulation via CST
Dist: calculate distance between two vectors

Initialize: ▷ Step 1
$\{(\boldsymbol{X}_j, \boldsymbol{Y}_j)\}_{N_0}, \quad j = 0, 1, ..., N_0$
$\boldsymbol{X}_j = [x_{j,0}, x_{j,1}, ...], \quad x_{j,i} \in \{0, X_{step,i}, 2X_{step,i}, ..., 1\}$
$\boldsymbol{Y}_j \leftarrow Simulate(\boldsymbol{X}_j)$
$c_t = 0$

Acquisition: ▷ Step 2
for $t = N_0$ to $T - 1$ **do**
  for $\hat{j} = 0$ to t **do**
    $\hat{\boldsymbol{X}}_{\hat{j}} = \boldsymbol{X}_j \leftarrow \underset{j, \boldsymbol{X}_j > \boldsymbol{X}_{\hat{j}}}{argmin}\{\{Dist(\boldsymbol{X}_j, \boldsymbol{X}_{\hat{j}})\}_{N_t}\}$ ▷ (a)
  end for
  $\{(\hat{\boldsymbol{X}}_j, \hat{\boldsymbol{Y}}_j)\}_{N_t} = \{(\hat{\boldsymbol{X}}_{\hat{j}}, \hat{\boldsymbol{Y}}_{\hat{j}})\}_{N_t}$ ▷ (b)
  while True **do**
    $\boldsymbol{X}_k, \hat{\boldsymbol{X}}_k = \underset{j}{argmax}\{Dist(\boldsymbol{Y}_j, \hat{\boldsymbol{Y}}_j)\}_{N_t}$ ▷ (c)
    if True in $Dist(\boldsymbol{X}_k, \hat{\boldsymbol{X}}_k) \geq 2\boldsymbol{X}_{step}$ **then** ▷ (d)
      $Break$
    else
      $Dist(\boldsymbol{X}_k, \hat{\boldsymbol{X}}_k) = 0$
    end if
  end while
  Generate new data set:
    $\boldsymbol{X}_* = c_t \cdot \boldsymbol{X}_k + (1 - c_t) \cdot \hat{\boldsymbol{X}}_k$ ▷ (e)
    $c_t = \frac{rand(0,1) + c_t \cdot (t - N_0)}{t - N_0 + 1}$ ▷ (f)
    $\boldsymbol{Y}_* = Simulate(\boldsymbol{X}_*)$ ▷ (g)
    $\{(\boldsymbol{X}_j, \boldsymbol{Y}_j)\}_{N_t}.append((\boldsymbol{X}_*, \boldsymbol{Y}_*))$ ▷ (h)
    $N_t = t + 1$
  $model\_loss \leftarrow$ **Train model** using $\{(\boldsymbol{X}_j, \boldsymbol{Y}_j)\}_{N_t}$
  if $model\_loss \leq min\_loss$ **then**
    Break
  end if ▷ (Optional) Integrated with model training
end for

initialize the initial set of N_0 samples. The input vectors $\boldsymbol{X}$s of the N_0 samples are defined in a uniform manner. The values of $\boldsymbol{X}$ of the first sample are set as the minimum values within the parameter range. The values of $\boldsymbol{X}$s of the following samples gradually increase by a constant increment. The increment is decided by the parameter range and the number of initial samples N_0. The values of $\boldsymbol{X}$ of the last sample are the maximum values within the parameter range. The initial set of samples distribute uniformly within the parameter range. It ensures that the parameter space is represented and covered unbiasedly for avoiding uncertainty caused by initialization. Importantly, N_0 is significantly smaller than the number of samples required in common ML tasks. The second step is to analyze the existing samples and produce samples of high quality, iteratively. This step is integrated with online model training to abort the iteration as soon as adequate samples have been acquired and the model loss for the expected test set reaches the minimum threshold. It is worth noting that the proposed method can also be used independently without being integrated with the model training process. In that case, the iteration stops when a sufficient number of samples are obtained.

For each iteration t in the second step, N_t existing samples $\{(\boldsymbol{X}_j, \boldsymbol{Y}_j)\}_{N_t}$ are analyzed to generate a new sample of high quality. The second step can be further divided into eight sub-steps (a-h) as follows.

(a) Pick a reference input vector $\hat{\boldsymbol{X}}_j$ for each existing input vector $\boldsymbol{X}_j$, by minimizing the distance between $\hat{\boldsymbol{X}}_j$ and $\boldsymbol{X}_j$, while making sure that all the elements of $\hat{\boldsymbol{X}}_j$ are equal to or bigger than those of $\boldsymbol{X}_j$ and at least one element of $\hat{\boldsymbol{X}}_j$ is bigger than that of $\boldsymbol{X}_j$.
(b) Form a list of reference samples $\{(\hat{\boldsymbol{X}}_j, \hat{\boldsymbol{Y}}_j)\}_{N_t}$ with respect to existing samples $\{(\boldsymbol{X}_j, \boldsymbol{Y}_j)\}_{N_t}$.
(c) Pick one input vector $\boldsymbol{X}_k$ by maximizing the distance ($Dist(\boldsymbol{Y}_k, \hat{\boldsymbol{Y}}_k)$) between its output $\boldsymbol{Y}_k$ and its reference output $\hat{\boldsymbol{Y}}_k$. The selected sample ($\boldsymbol{X}_k$, $\boldsymbol{Y}_k$) and its reference sample ($\hat{\boldsymbol{X}}_k$, $\hat{\boldsymbol{Y}}_k$) point at an input space where the sample of high quality exists. The underlying reason is that selected samples have a large distance between their output vectors and a small distance between their input vectors, implying that they confine a parameter space where the output is sensitive to the input. A sensitive parameter space is difficult for prediction. Hence, it is likely that this space contribute a lot to the prediction error of the ML model. On the other hand, adding a new sample in this space can improve the model performance significantly. In this sense, a potential high-quality sample of high informativeness referred to as ($\boldsymbol{X}_*$, $\boldsymbol{Y}_*$) can be generated from the selected sample ($\boldsymbol{X}_k$, $\boldsymbol{Y}_k$) and its reference sample ($\hat{\boldsymbol{X}}_k$, $\hat{\boldsymbol{Y}}_k$).
(d) Examine if the absolute difference between $\boldsymbol{X}_k$ and $\hat{\boldsymbol{X}}_k$ exceeds $2\times\boldsymbol{X}_{step}$. If yes, it guarantees an input space large enough to generate a new input vector; otherwise, then repeat sub-step (c) after excluding these two samples.
(e) Generate a new input vector $\boldsymbol{X}_*$ from the selected input vector $\boldsymbol{X}_k$ and its reference vector $\hat{\boldsymbol{X}}_k$ through formula (1):

$$\boldsymbol{X}_* = c_t \cdot \boldsymbol{X}_k + (1 - c_t) \cdot \hat{\boldsymbol{X}}_k. \tag{1}$$

The input vector of the new sample is obtained through swarm operation, as given in formula (1), on the selected input vector $\boldsymbol{X}_k$ and its reference input vector $\hat{\boldsymbol{X}}_k$. Here,

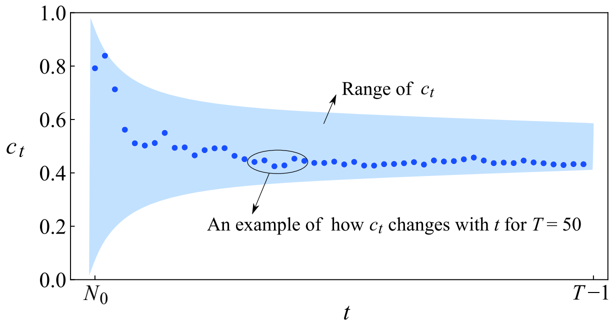


Fig. 1. Values of c_t versus iteration t.

the input vector of the new sample is referred to as $\boldsymbol{X}_*$ and it is set as the weighted sum of the $\boldsymbol{X}_k$ and $\hat{\boldsymbol{X}}_k$. The weights for $\boldsymbol{X}_k$ and $\hat{\boldsymbol{X}}_k$ are c_t and $1 - c_t$, respectively.

(f) Determine the weight c_t through formula (2):

$$c_t = \frac{rand(0,1) + c_t \cdot (t - N_0)}{t - N_0 + 1}. \quad (2)$$

The definition of c_t reaches a balance between exploration and exploitation. Fig. 1 shows that c_t has initial value between 0 and 1 and approaches 0.5 as the iteration t continues. This definition offers freedom to explore the input space in the earlier iterations as c_t can take any value between 0 and 1. With c_t being closer to 0, the new input vector $\boldsymbol{X}_*$ is closer to the selected input vector $\boldsymbol{X}_k$. As if c_t gets closer to 1, the new input vector $\boldsymbol{X}_*$ approaches $\hat{\boldsymbol{X}}_k$. As the iteration continues, c_t gets close to 0.5, and the input vector of the new sample $\boldsymbol{X}_*$ approaches the average of $\boldsymbol{X}_k$ and $\hat{\boldsymbol{X}}_k$. This ensures a minimum distance between the new input vector and the selected input vectors, because the space for exploration between $\boldsymbol{X}_k$ and $\hat{\boldsymbol{X}}_k$ decreases as the iteration continues. A new sample which is too close to an existing sample, lacks informativeness as it is likely to behave approximately the same as that existing sample.

(g) Obtain the output label vector $\boldsymbol{Y}_*$ for this new input vector $\boldsymbol{X}_*$ through full-wave simulation via CST.

(h) Form a new sample ($\boldsymbol{X}_*$, $\boldsymbol{Y}_*$), add this new sample into the existing data set, and repeat sub-steps (a-h) until sufficient data have been acquired.

In Algorithm 1, the proposed data acquisition method is integrated with model training. As the data set is being updated continuously by adding new high-quality samples, a model is trained in the meantime. After training, the model is tested on a fixed test set which is pre-defined according to the practical needs. The whole data acquisition procedure ceases as soon as the test loss reaches the desired minimum loss *min_loss*. Note that the proposed data acquisition method can also be employed without integration with model training. In that case, the iterations in the second step will stop when a sufficient number of samples have been generated.

III. Implementation

The proposed method is validated by comparing the results without and with the proposed method in four implementations. We utilize the proposed data acquisition method and re-implemente the design and analysis of Modified Jerusalem Cross (MJC) reflective surfaces [2] in Section III-A, multi-bit coding metasurface for radar cross-section (RCS) reduction [3] in Section III-B, array radiation synthesis [4] in Section III-C, and larger array radiation synthesis in Section III-D to analyze the performance of our methodology. Comparison of training results in the implementations validates high quality data generation by the proposed method. It is worth noting that we directly use the prior information (e.g., griding space, parameter range, minimum loss, etc.) from the original implementations for fairly comparing with original results. When the proposed method is applied to a new unknown implementation, this prior information can be easily acquired according to domain knowledge and its concrete EM requirements.

TABLE I
Implementation A: Geometric Parameters of the Modified Jerusalem Cross-based Unit Cell [2]

Parameter	l_x	l_y	w_1	w_2	a	u	S	h	t
Value(mm)	[1.9, 4.4]	1.1	0.2	0.2	5	0.1	1	2	0.017

A. Implementation A: MJC Reflective Surface

1) Implementation Description: The authors in [2] proposed a Modified Jerusalem Cross (MJC) reflective surface that offers independent control of orthogonally-polarized signals. The MJC reflective surface was designed to operate at $10\,\mathrm{GHz}$. The structure of its unit cell is given in Fig. 2. The unit cell consists of three overlapped dielectric (F4B) layers, three identical metal MJCs printed on top of each dielectric layer, and a full metal layer as the ground at the bottom. A MJC is composed of two orthogonally-crossed metal bars. The length of each bar (l_x/l_y) can be adjusted independently to tune the reflective phase (φ_x/φ_y) for the corresponding polarization. Implementation A works on adjusting the length of the bar in x direction (l_x) independently for tuning the reflective phase in the x-polarization.

To conveniently design the length l_x for any desired phase φ_x, Zhu. et al. utilized a backpropagation neural network (BPNN) to learn the mapping from the reflective phase φ_x to the length l_x. l_x varies from $1.9\,\mathrm{mm}$ to $4.4\,\mathrm{mm}$, while the rest of the geometric parameters were fixed as given in Table I. l_x is marked in blue color in Fig. 2 and in Table I. The detailed architecture of BPNN is listed in Table II. It can be observed that the BPNN consists of an input layer which takes the phase as input, a hidden layer of 20 neurons with activation function as Tanh [43], and an output layer that outputs l_x. The BPNN uses the Mean Squared Error (MSE) of the predicted and real l_x as its loss function and Levenberg-Marquardt is set as the backpropagation algorithm. 1000 samples were acquired by sweeping the length at a constant step of $0.0025\,\mathrm{mm}$, among which 700 and 150 samples were used for training and validation, and 150 samples for test, respectively. The minimum loss was 5.01×10^{-6}.

2) Re-implementation: For re-implementation, the proposed method is used to generate high-quality data for training the same BPNN. Variables of the proposed method are fixed as

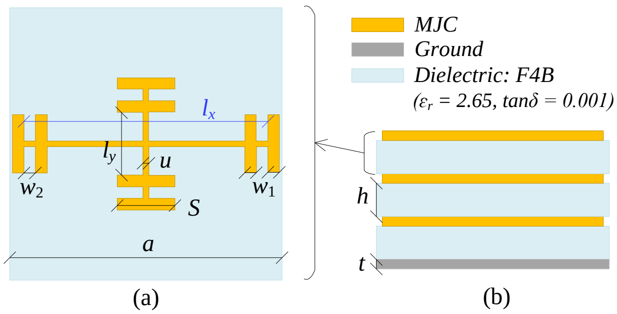


Fig. 2. Implementation A: Structure of Modified Jerusalem Cross-based unit cell: (a) Top view; (b) Side view. [2]

TABLE II
IMPLEMENTATION A: ARCHITECTURE OF THE BACKPROPAGATION NEURAL NETWORK (BPNN) [2]

No.	Layer	Neurons	Function
1	Input layer	1	Input: phase φ_x
2	Hidden layer	20	Fully-connected layer
3	Transfer function	-	Tanh
4	Output layer	1	Output: length l_x
5	Transfer function	-	Linear
-	**Loss function**	-	Mean Squared Error
-	**Algorithm**	-	Levenberg-Marquardt

listed in Table III. Here, the input vector $\boldsymbol{X}$ represents the phase φ_x of size 1 and the output vector $\boldsymbol{Y}$ represents the length l_x of size 1. Maximum number of iteration for data acquisition T is fixed at 700, which is also the number of training samples used in [2]. $\boldsymbol{X}_{step}$ is fixed at $[0.0025\,\mathrm{mm}]$, which is the smallest step of l_x considered in [2] and a common fabrication tolerance. *min_loss* is set as 5.01×10^{-6}, which is also the test loss reported in [2]. The only adjustable variable left is N_0, which is marked in blue color as shown in Table III. Number of training samples used to reach minimum loss *min_loss* of 5.01×10^{-6} (which was actually achieved by [2]), with and without our method are considered for comparison.

The proposed method is performed four times for four different values of N_0s (50, 80, 100, 150). The model uses the architecture introduced in [2], as shown in Table II. The number of required training samples (Ns) and the final losses for 150 validation and 150 test samples (Ls) corresponding to 4 different N_0s are listed in Table IV. The 150 validation and 150 test samples are generated by arbitrarily setting l_x between $1.9\,\mathrm{mm}$ and $4.4\,\mathrm{mm}$ with a step size of $0.0025\,\mathrm{mm}$, in the same manner as introduced in [2]. The results suggest that $N_0 = 80$ requires the lowest number of training samples ($N = 80 + 65 = 145$, $L = 4.81 \times 10^{-6}$) to converge beneath the *min_loss* (marked in blue color in Table IV). When $N_0 = 50$, it fails to converge towards *min_loss* because 50 initialized samples do not provide sufficient information. When N_0 rises up to 100 and 150, more training samples ($N = 100+61 = 161$ and $N = 150+52 = 202$) are required to reach the *min_loss*. The increment of training samples is mainly caused by increasing initialized samples because the amounts of new samples are approximately the same when N_0 is set as 80, 100, or 150. To sum up, N_0 should be large enough to offer adequate beginning data while still being as minimal as feasible to prevent taking up needless simulation cycles. N_0 is set as 80 in Implementation A. The $N_0 = 80$ samples are initialized with l_x being set from $1.9\,\mathrm{mm}$ to $4.4\,\mathrm{mm}$ at a constant step.

TABLE III
RE-IMPLEMENTATION A: VARIABLES OF THE PROPOSED METHOD

Variable	$\boldsymbol{X}$	$\boldsymbol{Y}$	$\boldsymbol{T}$	$\boldsymbol{X}_{step}$	*min_loss*	N_0
Value	$[l_x]$	$[\varphi_x]$	700	$[0.0025\,\mathrm{mm}]$	5.01×10^{-6}	80

TABLE IV
RE-IMPLEMENTATION A: COMPARISON OF RESULTS WITHOUT [2] AND WITH THE PROPOSED DATA ACQUISITION METHOD WITH VARIOUS N_0

	Without	**With the proposed method with N_0 as:**			
	[2]	50	80	100	150
$\boldsymbol{N}$	700	Fail	$80 + 65$	$100+61$	$150+52$
Time	7 h	-	1.45 h	1.61 h	2.02 h
$\boldsymbol{L}(10^{-6})$	5.01	Fail	4.81	4.89	4.90

Note: $\boldsymbol{N}$ is the number of training samples, which is 700 in [2];
Time is the time consumed for full-wave simulation.
$\boldsymbol{L}$ is the final loss, which is 5.01×10^{-6} in [2].

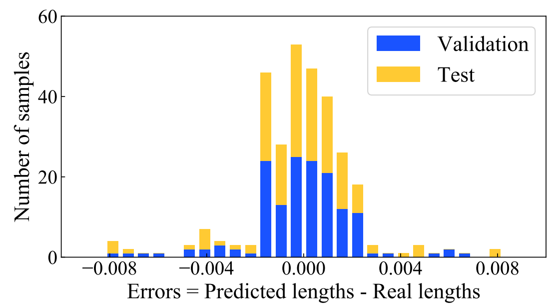


Fig. 3. Re-implementation A: Number of samples versus the errors between predicted lengths and real lengths.

3) Comparison of Training Results: Table IV compares the results obtained without [2] and with using the proposed method. Authors in [2] collected training samples by sweeping l_x at a constant step. 700 training samples were required in [2] to achieve an average loss of 5.01×10^{-6} and an error range of ± 0.008 for 150 validation and 150 test samples. For comparison, the proposed method is integrated with model training to re-implement the work in [2]. The model is also tested for 150 validation and 150 test samples. At iteration $t = 144$, 145 training samples have been generated and are used to train the model. The average loss for 150 validation and 150 test samples is 4.81×10^{-6} and the error range is ± 0.008 as shown in Fig. 3. Note that only validation and test losses are compared for two reasons: the same number of validation (150) and test (150) samples are collected in the same manner as in [2]; the number of training samples are different, which will lead to unfair comparison of the training losses. Numerical results suggest that only $20.71\,\%$ number of training samples are required to realize comparable model performance by using the proposed method. By comparison, the proposed data acquisition method saves $79.29\,\%$ training samples to achieve similar model performance, hence $79.29\,\%$ simulation cycles are saved in implementation A. Each simulation cycle takes around $36\,\mathrm{seconds}$, with a computer equipped with $96\,\mathrm{GB}$

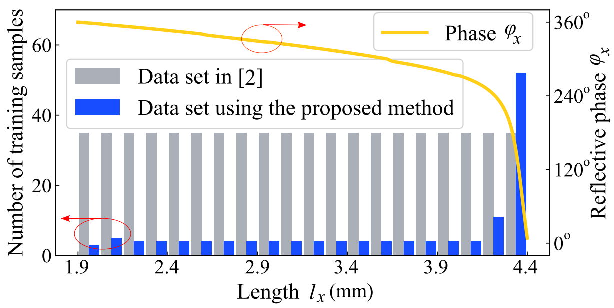


Fig. 4. Re-implementation A: Comparison of distributions of the data set in [2] and the high-quality data set using the proposed method.

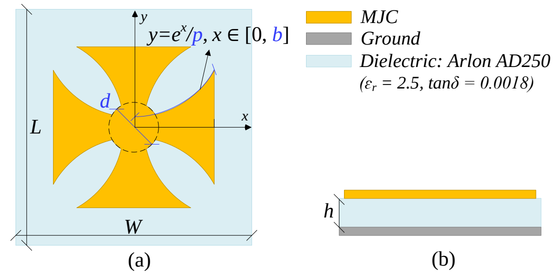


Fig. 5. Implementation B: Structure of Crusader cross-based unit cell: (a) Top view; (b) Side view. [3]

RAM and Intel® Xeon® Silver 4208 CPU @ 2.10 GHz (2 processors). Therefore, 5.55 hours full-wave simulation time is saved in implementation A by using the proposed method.

4) Analysis and Discussion: The distribution of the data set in [2] and the high-quality data acquired by using the proposed method are investigated to explore the underlying reason for saving 79.29 % training samples in Fig. 4. The data set used in [2] was collected by sweeping at a uniform step of 0.0025 mm, hence following a uniform distribution. The high-quality data set is collected adaptively to the output distribution governed by the proposed data acquisition method. Phase φ_x decreases drastically When length l_x varies from 4.1 mm to 4.4 mm, while it shows incremental decrease as length l_x increases from 1.9 mm to 4.1 mm. Therefore, intensive samples are collected with l_x being fixed between 4.1 mm and 4.4 mm, and only a few samples are collected out of this range. This adaptive strategy provides more valuable information with a small number of samples. As a result, 79.29 % training samples are saved, hence 79.29 % simulation cycles are saved by using the proposed method.

B. Implementation B: RCS Reduction Metasurface

1) Implementation Description: The authors in [3] proposed a multi-bit coding metasurface for radar cross section (RCS) reduction. A x-bit metasurface consists of 2^x groups of unit cells. Different groups of unit cells have reflection phases incrementally increasing from 0 at a uniform step of $\frac{2\pi}{2^x}$, while unit cells within each group have identical reflection phases. The topology of unit cells resembles the Crusader cross, as shown in Fig. 5. The structure of a unit cell is determined by three geometric parameters, p, b, and d. By adjusting p, b, and d, its structure can be modified, and its reflection phase can be changed accordingly. The unit cell's overall size and thickness were fixed as constant values, as shown in Table V.

To facilitate the multi-bit metasurface design process, the authors constructed a surrogate model to predict the reflection phase of any given unit cell. The input of the surrogate model was set as $[p, b, d]$ because the unit cell is determined by the three geometric parameters. $[p, b, d]$ was confined within a 3D parameter space defined by an upper bound $[3.5, 0.3, 0.2]$ and a lower bound $[10, 1.6, 2.4]$, as seen in Table V. The 3D space was divided uniformly into $7 \times 12 \times 7$, and 588 uniformly distributed inputs were defined. The 588 samples

TABLE V
IMPLEMENTATION B: GEOMETRIC PARAMETERS OF THE CRUSADER CROSS-BASED UNIT CELL [3]

Parameter	Value	Parameter	Value
p (mm)	[3.5, 10]	W (mm)	6
b (mm)	[0.3, 1.6]	L (mm)	6
d (mm)	[0.2, 2.4]	h (mm)	1.5

TABLE VI
RE-IMPLEMENTATION B: VARIABLES OF THE PROPOSED METHOD

Variable	Value	Variable	Value
$\boldsymbol{X}$	[p, b, d]	T	500
$\boldsymbol{Y}$	[Reflection phase]	*min_loss*	0.86 °
$\boldsymbol{X}_{step}$	[0.5, 0.06, 0.18]	N_0	175

were simulated via full-wave simulation supported by CST. Each output was the reflection phase corresponding to each $[p, b, d]$ within a frequency range from 10 GHz to 35 GHz. Among the 588 samples, 85 % (500) were used for training, and 15 % (88) testing. The surrogate model was developed using kriging interpolation [44]. The loss function was the MSE between the predicted and simulated reflection phases. The average loss for the test samples was 0.86 ° with a standard deviation of 1.7 °.

2) Re-implementation: To re-implement the work in [3], the proposed method is used to generate training samples, and generated training samples are utilized to develop a surrogate model using kriging interpolation. In this re-implementation, the variables for the proposed method are set as shown in Table VI. Here, the input vector $\boldsymbol{X}$ consists of normalized values of p, b, and d, and the output vector $\boldsymbol{Y}$ refers to reflection phases from 10 GHz to 35 GHz. The maximum iteration of data acquisition T is fixed at 500, which is the same as the number of training samples used in [3]. $\boldsymbol{X}_{step}$ is set as [0.5, 0.06, 0.18], which is inversely proportional to the grid density ($7 \times 12 \times 7$) of the geometric parameter space defined in [3]. *min_loss* is set as 0.86 °, which is also the reported test loss in [3]. $N_0 = 175$ samples are initialized. The 175 input vectors distribute on a uniform grid of $5 \times 7 \times 5$. Their corresponding reflection phases from 10 GHz to 35 GHz are collected as output labels through full-wave simulation using CST.

Starting from the 175 initialized samples, new training samples are generated iteratively using the proposed method. At each iteration t, there are $N_t = t + 1$ existing training samples, including 175 initialized samples and $N_t - 175$ new training samples. All the N_t existing samples are used

TABLE VII
RE-IMPLEMENTATION B: COMPARISON OF RESULTS WITHOUT [3] AND WITH THE PROPOSED DATA ACQUISITION METHOD

	Without the proposed method	With the proposed method
N	500	$175 + 122$
Time	3.89 h	2.31 h
L	0.86°	0.82°

Note: N is the number of training samples, which is 500 in [3]; L is the test loss, which is 0.86° in [3].

as training data to develop a surrogate model using kriging interpolation at iteration t. After training, the surrogate model is used to predict reflection phases of 88 test samples. The 88 test samples distribute randomly on a uniform grid of $7\times12\times7$ of the geometric parameter space, which is similar to the 88 test samples used in [3]. The mean square error L for the 88 test samples is compared with *min_loss*. The data acquisition process completes when L is less than or equal to *min_loss*. For re-implementation B, the data acquisition process finishes at iteration $t = 296$, and 297 data samples are collected for training.

3) Comparison of Training Results: The results of implementation B with or without using the proposed method are compared in Table VII. Without the proposed method, 588 data samples were acquired on a uniform grid of $7 \times 12 \times 7$ of the geometric parameter space. The 588 data samples were arbitrarily separated into a training data set of 500 data samples and a test data set of 88 data samples. The surrogate model was trained using the training data set through kriging interpolation. Afterward, the well-trained model was tested on the test data set, and the mean square error on the test data set was $0.86\,°$. With the proposed method, $175 + 122 = 297$ training samples are acquired and utilized to develop the surrogate model using kriging interpolation. A test data set of 88 data samples is formed by arbitrarily choosing 88 points on the uniform grid of $7 \times 12 \times 7$ of the geometric parameter space. The well-trained surrogate model is tested on the test data set, and the MSE for test data is $0.82\,°$. Without the proposed method [3], 500 training samples were required to reach a test loss of $0.86\,°$. With the proposed method, only 297 training samples are required to reach a test loss of $0.82\,°$. By comparison, using the proposed method reduces the number of required training samples by $40.6\,\%$ ($\frac{500-297}{500}$), hence $40.6\,\%$ full-wave simulation cycles are saved. It spends 28 seconds for each simulation cycle supported by a computer equipped with 96 GB RAM and Intel® Xeon® Silver 4208 CPU @ 2.10 GHz (2 processors). Therefore, 1.58 hours full-wave simulation time is saved in re-implementation B by using the proposed method.

4) Analysis and Discussion: Fig. 6 illustrates how the generated training samples distribute in the geometric parameter space to illustrate the underlying reason why the number of training data samples is reduced by $40.6\,\%$. The distributions over p, b, and d are shown in Fig. 6(a), (b), and (c), respectively. In each sub-figure, blue rectangular bars represent the number of training samples generated by using the proposed method, gray rectangular bars represent the number of training samples used in [3], and each orange dot represents the average

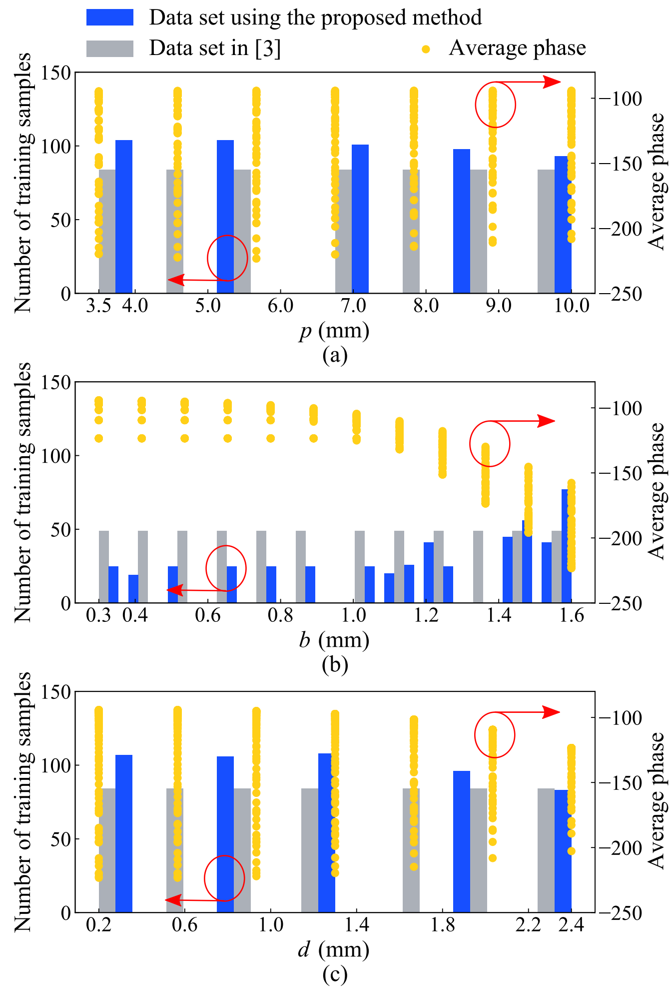


Fig. 6. Implementation B: Comparison of distributions of the data set in [3] and the high-quality data set using the proposed method, while (a) p, (b) b, or (c) d varies. (Each orange dot indicates one data sample's average reflection phase from 10 GHz to 35 GHz.)

reflection phase from 10 GHz to 35 GHz of one data sample.

The data distribution over p and d is shown in Fig. 6(a) and (c). As p increases from 3.5 mm to 10 mm and d increases from 0.2 mm to 2.4 mm, the variance of average reflection phases decreases incrementally, hence adding samples with p and d being fixed at smaller values can improve the model performance. Therefore, the proposed method tends to generate slightly more training samples as p and d decrease.

The data distribution over b is shown in Fig. 6(b). The mean and variance of average reflection phases keep unchanged as b increases from 0.3 mm to 0.9 mm. As b increases from 0.9 mm to 1.6 mm, the mean decreases drastically from $-100\,°$ to $-200\,°$, and the variance increases from $\pm30\,°$ to $\pm80\,°$. Therefore, only a small number of training samples are generated uniformly by the proposed method as $b \in [0.3, 0.9]$, and a significantly increasing number of training samples are generated as b increases from 0.9 mm to 1.6 mm.

In a word, the proposed method tends to add more training samples where the average reflection phases have higher variance or unstable mean. The reason is that the higher variance and unstable mean correspond to the more complex learning area. Adding training samples within this area can significantly improve the model performance. On the other hand, reducing training samples outside this area can save simulation cycles yet does not harm the model performance. It can be observed from Fig. 6 that, the proposed method adjusts the number of training samples adaptively according to the variance and mean of average reflection phases, hence only 297 training samples are required to realize a test loss of $0.82\,^\circ$. By contrast, the training samples were generated uniformly within the whole parameter space in [3], hence 500 training samples are required to realize a test loss of $0.86\,^\circ$. The number of training samples is reduced by $40.6\,\%$, and $40.6\,\%$ simulation cycles and $1.58\,$hours simulation time are saved by using the proposed method.

C. Implementation C: Array Radiation Synthesis

1) Implementation Description: Kim et al. in [4] utilized a deep neural network (DNN) to determine the phases of a 1×4 antenna array for various array radiation patterns. The operating frequency of the antenna array is at $2.4\,$GHz. The antenna array consists of four coaxial-fed patch antennas, as shown in Fig. 7. The amplitudes for four elements were fixed at 1, and the phase for element 1 was set as $0\,^\circ$. When the elements 2, 3, and 4 are fed with signals of different phases (α_2, α_3, and α_4), the combined array radiation pattern varies accordingly. In conventional scenarios, experienced engineers are required to decide the phases (α_2, α_3, and α_4) for a desired array radiation pattern. In [4], a DNN was trained to automatically determine the phases for desired radiation patterns without interference from human engineers.

The architecture of DNN used in [4] is given in Table VIII. There is an input layer, three hidden layers, and an output layer. The input layer and hidden layers have ReLU as activation functions, and the output layer uses the Linear activation function. Three hidden layers have neurons of 150, 100, and 80. Each input data for the input layer is of size 181 that represents a normalized radiation pattern with $\varphi = 0\,^\circ$ and θ ranging from $0\,^\circ$ to $180\,^\circ$ in units of $1\,^\circ$, which is referred to as $[R\{\varphi(0°),\ \theta[0°,\ 180°]\}]$. Each output data for the output layer is of size 6 that represents real and imaginary parts of complex excitation for elements 2, 3, and 4, which is referred to as $[cos\alpha_i,\ sin\alpha_i]$, $i = 2, 3, 4$. MSE is taken as the loss function for the DNN.

The authors in [4] generated 6859 samples for training. The amplitudes for four elements were fixed at 1, and the phase for element 1 was set as $0\,^\circ$. The phases for element 2, 3, and 4 increased from $0\,^\circ$ to $360\,^\circ$ at a step of $20\,^\circ$. The number of states for each element is 19 and the total amount of excitation combinations is 6859. The corresponding radiation patterns for all 6859 excitation combinations were acquired using CST. Similarly, 64 samples were generated as the validation data set. For validation, the phases for element 2, 3, and 4 increased from $10\,^\circ$ to $130\,^\circ$ at a step of $40\,^\circ$.

The DNN was trained for 500 epochs using the 6859 training samples and the batch size was 100. After training, the model was validated on the 64 validation samples. The final training loss is 2.2×10^{-4} and the MSE for validation data is 2.6×10^{-4}.

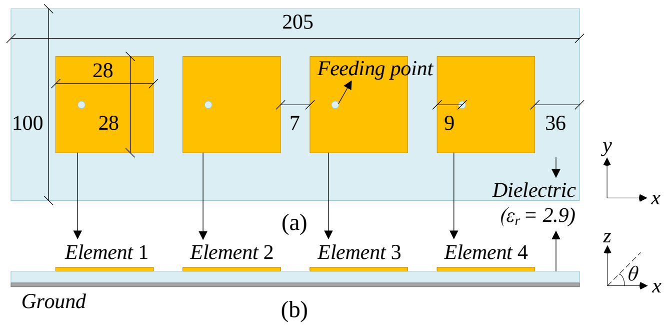


Fig. 7. Implementation C: Array antenna structure: (a) Top view; (b) Side view. [4]

TABLE VIII
IMPLEMENTATION C: ARCHITECTURE OF THE DNN [4]

No.	Layer	Neurons	Function
1	Input layer	181	Input: [radiation]
2	Activation function	-	ReLU
3	Hidden layer	150	Fully-connected layer
4	Activation function	-	ReLU
5	Hidden layer	100	Fully-connected layer
6	Activation function	-	ReLU
7	Hidden layer	80	Fully-connected layer
8	Activation function	-	ReLU
9	Output layer	6	Output: [excitation]
10	Activation function	-	Linear
-	**Loss function**	-	Mean Squared Error

TABLE IX
IMPLEMENTATION C: VARIABLES OF THE PROPOSED METHOD

Variable	Value	Variable	Value
$\boldsymbol{X}$	$[\alpha_2,\ \alpha_3,\ \alpha_4]$ $[cos\alpha_i,\ sin\alpha_i],\ i:2,3,4$	T	6859
$\boldsymbol{Y}$	$[R\{\varphi(0°),\ \theta[0°,\ 180°]\}]$	*min_loss*	2.6×10^{-4}
$\boldsymbol{X}_{step}$	$[10\,^\circ,\ 10\,^\circ,\ 10\,^\circ]$	N_0	2197

2) Re-implementation: To re-implement the work in [4], the variables for the proposed method are determined as given in Table IX. $\boldsymbol{X}$ is set as phases ($[\alpha_2,\ \alpha_3,\ \alpha_4]$) of elements 2, 3, 4 during data acquisition and is converted to the real and imaginary format ($[cos\alpha_i,\ sin\alpha_i]$, $i = 2, 3, 4$) for model training. $\boldsymbol{Y}$ is set as normalized radiation patterns with $\varphi = 0°$ and θ ranging from $0°$ to $180°$ in units of $1°$. The minimum step of phases $\boldsymbol{X}_{step}$ is fixed at $[10\,^\circ, 10\,^\circ, 10\,^\circ]$. The minimum loss *min_loss* is set as 2.6×10^{-4}, which equals the reported validation loss in [4]. The maximum data acquisition iteration is set as the number of training samples used in [4]. The amount of initialized samples N_0 is set as 2197 after comparing results corresponding to different N_0s.

3) Comparison of Training Results: The results of re-implementation using the proposed method are compared with results claimed in [4] in Table X. The authors in [4] collected 6859 training data samples by incrementally increasing elements' phases (α_2, α_3, and α_4) from $0\,^\circ$ to $360\,^\circ$ at a constant step of $20\,^\circ$. Similarly, 64 test data samples were collected by incrementally increasing elements' phases (α_2, α_3, and α_4) from $0\,^\circ$ to $360\,^\circ$ at a constant step of $20\,^\circ$.

TABLE X
RE-IMPLEMENTATION C: COMPARISON OF RESULTS WITHOUT [4] AND WITH THE PROPOSED DATA ACQUISITION METHOD

	Without the proposed method	With the proposed method
N	6859	$2197 + 1923$
Time	17.14 h	10.30 h
L	2.6×10^{-4}	2.6×10^{-4}

Note: N is the number of training samples, which is 6859 in [4]; L is the test loss, which is 2.6×10^{-4} in [4].

The DNN was trained using the 6859 training samples and was tested on the 64 test samples. The mean square error over the 64 test samples was 2.6×10^{-4}. For comparison, the proposed method is integrated with model training to re-implement the work in [4]. At iteration 4119, 4120 training samples are generated, and the DNN is trained using the 4120 training samples and is tested on the same 64 test samples. The mean square error equals the test loss reported in [4]. By comparison, the number of required training samples is reduced by 39.93 % by using the proposed method, hence 39.93 % simulation cycles are saved in implementation C. On average, it takes around 9 seconds for each simulation cycle supported by a computer equipped with 96 GB RAM and Intel® Xeon® Silver 4208 CPU @ 2.10 GHz (2 processors). Therefore, the proposed method saves 6.84 hours full-wave simulation time in implementation C.

4) Analysis and Discussion: To get an insight into the reason why 39.93 % training data are saved, the distribution of the high-quality training data set acquired using the proposed method is compared with that of the training data set used in [4]. Fig. 8(a), (b), and (c) exhibits the distribution over α_2, α_3, and α_4, respectively. Here, the high-quality data set is represented as blue rectangular bars, while the training data samples used in [4] are represented as gray rectangular bars. As it is difficult to plot the whole radiation pattern of each data sample, only the peak gain and average gain of the radiation pattern of each data sample are plotted. Each orange dot represents the peak gain of one data sample, while each red diamond-shaped symbol represents the average gain of one data sample.

As can be observed in Fig. 8, the peak gain and average gain fluctuate within a certain range as phases of elements (α_2, α_3, and α_4) change. The peak gain and average gain are just two compressed features of the radiation pattern. The changing tendency of radiation patterns is way more complex than that it can be plotted and observed from Fig. 8. Heavier fluctuations can be expected for the changing tendency of the whole radiation patterns. Therefore, it is difficult to specifically clarify the distribution of the generated data set using the proposed method. An overall observation is that the generated data set using the proposed method distribute adaptively and the number of generated training data samples varies concerning different phases of elements. By contrast, the data set used in [4] distribute uniformly on a constant grid. The benefit and effectiveness of this adaptive sampling strategy are validated through numerical and comparative results. The number of training data samples required for the same model accuracy is significantly reduced by 39.93 % by using the proposed method.

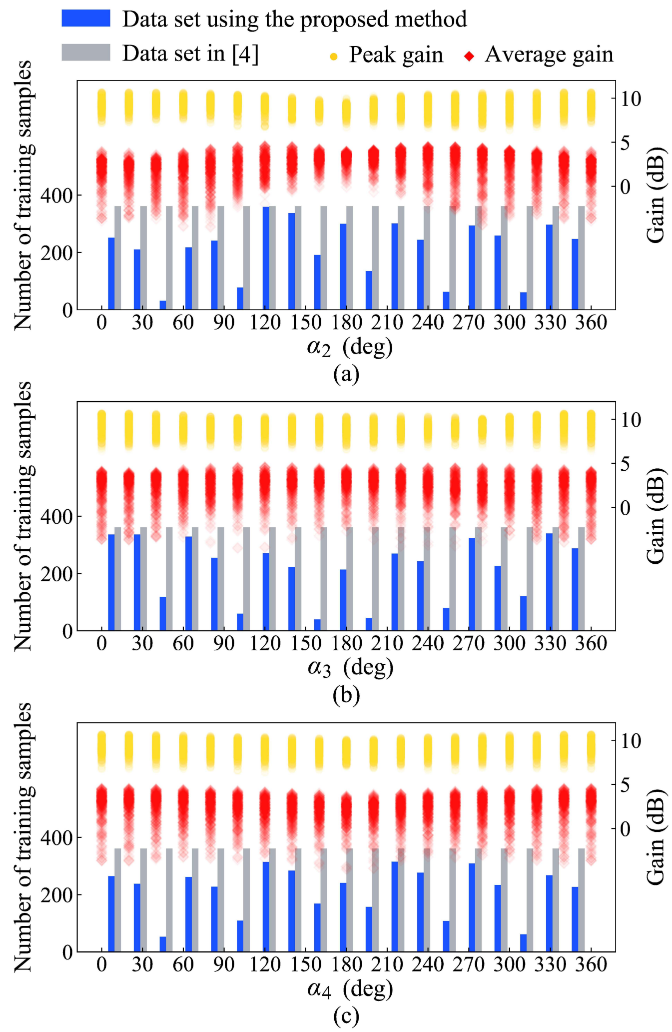


Fig. 8. Implementation C: Comparison of distributions of the data set in [4] and the high-quality data set using the proposed method, while (a) α_2, (b) α_3, or (c) α_4 varies. (Each orange dot indicates one data sample's peak gain; each red diamond indicates one data sample's average gain.)

ML-based radiation synthesis can be applied to a larger array with more than four elements, as long as powerful computation resources are available for simulating such large arrays, manipulating massive data, and training complicated models. The proposed method can also be used for this large array case, and the reduced amount will be proportional to the number of required training data acquired conventionally.

D. Implementation D: Enlarged Array Radiation Synthesis

The proposed method is validated in a three-dimensional parameter space in implementation C. To validate its performance in a higher-dimensional parameter space, we enlarge the four-element linear array in implementation C into an eight-element linear array, which is referred to as implementation D. Except for the number of elements, the array structure is the same. For the enlarged array, we use a new forward

TABLE XI
IMPLEMENTATION D: ARCHITECTURE OF THE DNN

No.	Layer	Neurons	Function
1	Input layer	14	Input: [excitation]
2	Activation function	-	ReLU
3	Hidden layer	300	Fully-connected layer
4	Activation function	-	ReLU
5	Hidden layer	200	Fully-connected layer
6	Activation function	-	ReLU
7	Hidden layer	100	Fully-connected layer
8	Activation function	-	ReLU
9	Output layer	181	Output: [radiation]
10	Activation function	-	Linear
-	**Loss function**	-	Mean Squared Error

DNN for training, as shown in Table XI. Here, phases of its excitation are set as input, and radiation patterns are set as output.

The DNN is first trained without using the proposed method. 78125 samples are collected for training by sweeping the phases of 7 elements. The amplitudes of eight elements are set as 1, and the phase of element 1 is set as $0\,°$. The phases of the rest 7 elements 2-8 vary from $0\,°$ to $180\,°$ at a constant step of $45\,°$. The corresponding 78125 radiation patterns are generated using CST. 128 samples are collected similarly as the validation data set, where the phases of 7 elements vary between $30\,°$ and $150\,°$. The MSE for validation data is around 0.73.

The proposed method is then used to re-implement implementation D for comparison. All the settings are similar to implementation C, and the variables are listed in Table XII. 16384 samples are initialized by sweeping the phases of 7 elements from $0\,°$ to $180\,°$ at a constant step of $60\,°$. The generation of high-quality data stops in iteration $t = 46499$, and 46500 samples are generated in total. The test loss of the DNN for the 128 validation samples reaches $min\_loss = 0.73$ after being trained using the 46500 data acquired using the proposed method.

The results without and with using the proposed method are compared and listed in Table XIII. As can be observed, the number of required training data is reduced by $40.48\,\%$ with using the proposed method, hence $40.48\,\%$ simulation cycles are saved. The time needed for each simulation cycle is around $13\,\mathrm{seconds}$. Thus, $114.2\,\mathrm{hours}$ simulation time is saved in implementation D with using the proposed method.

It proves that the proposed method is still effective as the dimension of parameter space increases from three to seven. Due to the limitations of computation resources, we cannot keep increasing the dimension and validating the proposed method in higher-dimensional spaces. If more powerful computation resources are available, it is meaningful to further investigate the effectiveness in higher-dimensional spaces, because it gets more complicated and may involve new challenges.

TABLE XII
IMPLEMENTATION D: VARIABLES OF THE PROPOSED METHOD

Variable	Value	Variable	Value
$\boldsymbol{X}$	$[\alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \alpha_6,\ \alpha_7,\ \alpha_8]$ $[cos\alpha_i,\ sin\alpha_i],\ i:2,3,4,5,6,7,8$	T	78125
$\boldsymbol{Y}$	$[R\{\varphi(0°),\ \theta[0°,\ 180°]\}]$	*min_loss*	0.73
$\boldsymbol{X}_{step}$	$[10\,°,10\,°,10\,°,10\,°,10\,°,10\,°,10\,°]$	N_0	16384

TABLE XIII
RE-IMPLEMENTATION D: COMPARISON OF RESULTS WITHOUT AND WITH THE PROPOSED DATA ACQUISITION METHOD

	Without the proposed method	With the proposed method
$\boldsymbol{N}$	78125	$16384 + 30116$
Time	$282.12\,\mathrm{h}$	$167.92\,\mathrm{h}$
$\boldsymbol{L}$	0.73	0.73

Note: N is the number of training samples;
L is the test loss.

IV. DISCUSSION

The proposed solution addresses a common challenge in the ML-based design of electromagnetic structures. The issue is that ML often requires a vast amount of simulations to gather data, which is both time-consuming and computationally expensive. Our approach reduces the number of simulations by evaluating the quality of data before simulation and prioritizing computation resources for high-quality data.

Designers can use the method to understand the sensitivity of the geometric parameters of electromagnetic structures. By analyzing the distribution of acquired high-quality data, as seen in Fig. 4, 6, and 8, designers can identify the most critical parameters and their respective ranges, helping them understand the motivation behind the geometry and how to adjust it to modify the electromagnetic response.

The effectiveness of the proposed method has been validated in low-dimensional implementations, and it remains effective as the dimension increases from three to seven. It is difficult to keep increasing the dimension and validate it in higher-dimensional spaces, due to the limitations of computation resources. Further investigation on higher-dimensional implementations is meaningful, as it gets more complicated and may involve new challenges. In future work, we will attempt to apply the proposed method in higher-dimensional implementations.

V. CONCLUSION

An intelligent high-quality data acquisition method for ML-related EM applications is proposed in this paper. Starting from a small uniformly initialized data set, the proposed method can intelligently generate high-quality data samples based on the analysis of existing data samples. Compared with conventional EM-ML works that acquired training data by blindly sweeping the whole geometric parameter space on a constant and uniform grid, the proposed method adaptively adjusts the sampling density in different geometric parameter areas. The proposed method produces a data set that maximizes informativeness with the least number of simulation cycles. To validate its performance, the proposed method is utilized to re-implement four implementations. The comparative results without and with the proposed method show that the proposed method significantly reduces the number of training data required for the same model accuracy. Around $40\,\%$ training data are saved by using the proposed method, hence a huge number of full-wave simulation cycles and time are saved, and computing resources are significantly released.

References

[1] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data-AI integration perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1328–1347, Oct. 2019, doi: 10.1109/TKDE.2019.2946162.

[2] 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, Jan. 2022, doi: 10.1109/TAP.2022.3140523.

[3] 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, Aug. 2021, doi: 10.1109/TMTT.2021.3105677.

[4] J. H. Kim and S. W. Choi, "A deep learning-based approach for radiation pattern synthesis of an array antenna," *IEEE Access*, vol. 8, pp. 226 059–226 063, Dec. 2020, doi: 10.1109/ACCESS.2020.3045464.

[5] D. Erricolo, P.-Y. Chen, A. Rozhkova, E. Torabi, H. Bagci, A. Shamim, and X. Zhang, "Machine learning in electromagnetics: A review and some perspectives for future research," in *Int. Conf. Electromagn. Adv. Appl. (ICEAA)*, Oct. 2019, pp. 1377–1380.

[6] Y. Kiarashinejad, S. Abdollahramezani, and A. Adibi, "Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures," *Npj Comput. Mater.*, vol. 6, no. 1, pp. 1–12, Feb. 2020, doi: 10.1038/s41524-020-0276-y.

[7] N. A. Alderete, N. Pathak, and H. D. Espinosa, "Machine learning assisted design of shape-programmable 3D kirigami metamaterials," *Npj Comput. Mater.*, vol. 8, no. 1, pp. 1–12, Sep. 2022, doi: 10.1038/s41524-022-00873-w.

[8] M. Sedaghat, R. Trinchero, Z. H. Firouzeh, and F. G. Canavero, "Compressed machine learning-based inverse model for design optimization of microwave components," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 7, pp. 3415–3427, Apr. 2022, doi: 10.1109/TMTT.2022.3166151.

[9] Z. Zhou, Z. Wei, Y. Zhang, P. Wang, J. Ren, Y. Yin, G. F. Pedersen, and M. Shen, "Training of deep neural networks in electromagnetic problems: a case study of antenna array pattern synthesis," in *2021 IEEE MTT-S International Wireless Symposium (IWS)*, Nanjing, China, May. 2021, pp. 1-3, doi: 10.1109/IWS52775.2021.9499638.

[10] N. Kazemi, M. Abdolrazzaghi, and P. Musilek, "Comparative analysis of machine learning techniques for temperature compensation in microwave sensors," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 9, pp. 4223–4236, May 2021, doi: 10.1109/TMTT.2021.3081119.

[11] Z. Wei, Z. Zhou, Y. Zhang, P. Li, J. Ren, Y. Yin, G. F. Pedersen, and M. Shen, "Domain knowledge assisted training dataset generation for metasurface designs," in *2021 IEEE MTT-S International Wireless Symposium (IWS)*, Nanjing, China, May. 2021, pp. 1-3, doi: 10.1109/IWS52775.2021.9499612.

[12] C. Lewis, J. Bryan, N. Schwartz, J. Hale, K. Fanning, and J. S. Colton, "Machine learning to predict quasi TE_{001} mode resonances in double-stacked dielectric cavities," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 4, pp. 2135–2146, Feb. 2022, doi: 10.1109/TMTT.2022.3145357.

[13] Z. Zhou, Z. Wei, J. Ren, Y. Yin, G. F. Pedersen and M. Shen, "Representation learning-driven fully automated framework for the inverse design of frequency-selective surfaces," *IEEE Trans. Microw. Theory Techn.*, early access, 2023, doi: 10.1109/TMTT.2023.3235066.

[14] Z. Wei, Z. Zhou, P. Wang, J. Ren, Y. Yin, G. F. Pedersen, and M. Shen, "Fully automated design method based on reinforcement learning and surrogate modeling for antenna array decoupling," *IEEE Trans. Antennas Propag.*, vol. 71, no. 1, pp. 660-671, Jan. 2023, doi: 10.1109/TAP.2022.3221613.

[15] Z. Zhou, Z. Wei, J. Ren, Y. Yin, G. F. Pedersen, and M. Shen, "Two-order deep learning for generalized synthesis of radiation patterns for antenna arrays," *IEEE Trans. Artif. Intell.*, pp. 1–9, Jul. 2022, doi: 10.1109/TAI.2022.3192505.

[16] 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, Dec. 2018, doi: 10.1109/TAP.2018.2889029.

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

[18] 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.

[19] 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.*, Oct. 2019, pp. 1–6.

[20] 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, Aug. 2019, doi: 10.1109/LAWP.2019.2932902.

[21] 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, Mar. 2021, doi: 10.1109/TMTT.2021.3061128.

[22] 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.*, Mar. 2022, pp. 1–5.

[23] R. Zhu, T. Qiu, J. Wang, S. Sui, C. Hao, T. Liu, Y. Li, M. Feng, A. Zhang, C.-W. Qiu *et al.*, "Phase-to-pattern inverse design paradigm for fast realization of functional metasurfaces via transfer learning," *Nat. Commun.*, vol. 12, no. 1, pp. 1–10, May 2021, doi: 10.1038/s41467-021-23087-y.

[24] 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, Feb. 2021, doi: 10.1109/TAP.2021.3060142.

[25] 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, Jan. 2022, doi: 10.1109/TAP.2022.3140523.

[26] Z. Zhou, Z. Wei, J. Ren, Y. Yin, G. F. Pedersen, and M. Shen, "Transfer learning assisted multi-element calibration for active phased antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 71, no. 2, pp. 1982–1987, Oct. 2022, doi: 10.1109/TAP.2022.3216548.

[27] 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, Apr. 2020, doi: 10.1109/LAWP.2020.2986023.

[28] 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, Feb. 2022, doi: 10.1109/TAP.2022.3152592.

[29] J. Bouvette, H.-F. Liu, X. Du, Y. Zhou, A. P. Sikkema, J. da Fonseca Rezende e Mello, B. P. Klemm, R. Huang, R. M. Schaaper, M. J. Borgnia *et al.*, "Beam image-shift accelerated data acquisition for near-atomic resolution single-particle cryo-electron tomography," *Nat. Commun.*, vol. 12, no. 1, pp. 1–11, Mar. 2021, doi: 10.1038/s41467-021-22251-8.

[30] K. Jeong, M. Babović, V. Gorshkov, J. Kim, O. N. Jensen, and O. Kohlbacher, "FLASHIda enables intelligent data acquisition for top–down proteomics to boost proteoform identification counts," *Nat. Commun.*, vol. 13, no. 1, pp. 1–12, Jul. 2022, doi: 10.1038/s41467-022-31922-z.

[31] H. Garcia-Molina, M. Joglekar, A. Marcus, A. Parameswaran, and V. Verroios, "Challenges in data crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 901–911, Jan. 2016, doi: 10.1109/TKDE.2016.2518669.

[32] X. Zhu and X. Wu, "Cost-constrained data acquisition for intelligent data preparation," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1542–1556, Sep. 2005, doi: 10.1109/TKDE.2005.176.

[33] Y.-C. Lin, D.-N. Yang, H.-H. Shuai, and M.-S. Chen, "Data acquisition for probabilistic nearest-neighbor query," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 410–427, Feb. 2014, doi: 10.1109/TKDE.2013.2297916.

[34] A. Pietrenko-Dabrowska, S. Koziel, and U. Ullah, "Reduced-cost two-level surrogate antenna modeling using domain confinement and response features," *Sci. Rep.*, vol. 12, no. 1, Art. no. 1, Mar. 2022, doi: 10.1038/s41598-022-08710-2.

[35] S. Koziel and A. Pietrenko-Dabrowska, "Expedited variable-resolution surrogate modeling of miniaturized microwave passives in confined domains," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4740–4750, Nov. 2022, doi: 10.1109/TMTT.2022.3191327.

[36] A. Pietrenko-Dabrowska, S. Koziel, and L. Golunski, "Two-stage variable-fidelity modeling of antennas with domain confinement," *Sci. Rep.*, vol. 12, no. 1, Art. no. 1, Oct. 2022, doi: 10.1038/s41598-022-20495-y.

[37] S. Koziel, N. Çalık, P. Mahouti, and M. A. Belen, "Reliable computationally efficient behavioral modeling of microwave passives using deep learning surrogates in confined domains," *IEEE Trans. Microw. Theory Techn.*, pp. 1–13, 2022, doi: 10.1109/TMTT.2022.3218024.

[38] J. A. Tomasson, A. Pietrenko-Dabrowska, and S. Koziel, "Expedited globalized antenna optimization by principal components and variable-fidelity EM simulations: application to microstrip antenna design," *Electron.*, vol. 9, no. 4, Art. no. 4, Apr. 2020, doi: 10.3390/electronics9040673.

[39] S. Koziel and A. Pietrenko-Dabrowska, "Expedited feature-based quasi-global optimization of multi-band antenna input characteristics with Jacobian variability tracking," *IEEE Access*, vol. 8, pp. 83907–83915, 2020, doi: 10.1109/ACCESS.2020.2992134.

[40] A. Pietrenko-Dabrowska and S. Koziel, "Globalized parametric optimization of microwave components by means of response features and inverse metamodels," *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, Dec. 2021, doi: 10.1038/s41598-021-03095-0.

[41] S. Koziel, A. Pietrenko-Dabrowska, and M. Mahrokh, "Globalized simulation-driven miniaturization of microwave circuits by means of dimensionality-reduced constrained surrogates," *Sci. Rep.*, vol. 12, no. 1, Art. no. 1, Sep. 2022, doi: 10.1038/s41598-022-20728-0.

[42] S. Koziel and A. Pietrenko-Dabrowska, "Low-cost quasi-global optimization of expensive electromagnetic simulation models by inverse surrogates and response features," *Sci. Rep.*, vol. 12, no. 1, Art. no. 1, Nov. 2022, doi: 10.1038/s41598-022-24250-1.

[43] B. L. Kalman and S. C. Kwasny, "Why tanh: choosing a sigmoidal function," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 4, Jun. 1992, pp. 578–581.

[44] T. W. Simpson, J. Poplinski, P. N. Koch, and J. K. Allen, "Metamodels for computer-based engineering design: survey and recommendations," *Eng. Comput.*, vol. 17, no. 2, pp. 129–150, Jul. 2001, doi: 10.1007/PL00007198.

Zhao Zhou was born in Shaanxi, China. He received the B.Sc. and M.Eng. degrees in electronic engineering from Xidian University, Xi'an, China, in 2017 and 2020, respectively. He is currently pursuing a Ph.D. degree in the Antennas, Propagation, and Millimeter-Wave Systems Section, Department of Electronic Systems, Aalborg University, Aalborg, Denmark.

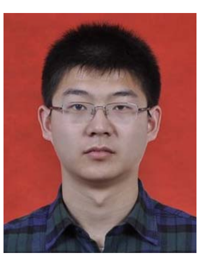
His research interests include dual-polarized antennas and machine learning-based design and optimization of electromagnetic components.

Zhaohui Wei was born in Shandong, China. He received the B.Sc. and M.Eng. degrees in electronic engineering from Xidian University, Xi'an, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the Department of Electronic Systems, Antennas, Propagation and Millimeter-Wave Systems Section, Aalborg University, Aalborg, Denmark.

His research interests include filtering antenna, frequency selective surface, deep learning-based methods for design and analysis of antenna systems.

Abdullah Tahir was born in Lahore, Pakistan. He received B.Sc. and M.Sc. degrees in Mechatronics from UET Lahore and GIST, South Korea in 2013 and 2017, respectively. He is currently a Ph.D. scholar in the Department of Electronic Systems, Aalborg University, Denmark.

His research interests include machine learning and human-robot-interaction.



Jian Ren (Member, IEEE) was born in Shandong, China. He received the B.Sc. and M.Eng. degrees in electronic engineering from Xidian University, Xi'an, China, in 2012 and 2015, respectively, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2018.

In 2019, he joined the National Key Laboratory of Antennas and Microwave Technology, Xidian University, where he is currently an Associate Professor. Since 2015, he has been a Research Assistant with the City University of Hong Kong. He has authored or co-authored over 40 refereed journal articles. His current research interests include dielectric resonator antenna, millimeter-wave antennas, and metamaterials. Dr. Ren received the Honorable Mention at the Student Best Paper Competition at the 2018 IEEE 7th Asia–Pacific Conference on Antennas and Propagation (APCAP). He has served as a reviewer for different peer-reviewed journals, including the IEEE Transactions on Antennas and Propagation, the IEEE Antennas and Wireless Propagation Letters, the Journal of Physics D: Applied Physics, and the IET Microwaves, Antennas & Propagation.

Yingzeng Yin (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electromagnetic wave and microwave technology from Xidian University, Xi'an, China, in 1987, 1990, and 2002, respectively.

From 1990 to 1992, he was a Research Assistant and an Instructor with the Institute of Antennas and Electromagnetic Scattering, Xidian University, where he was an Associate Professor with the Department of Electromagnetic Engineering, from 1992 to 1996, and has been a Professor, since 2004. His current research interests include the design of microstrip antennas, feeds for parabolic reflectors, artificial magnetic conductors, phased array antennas, base-station antennas, and computer-aided design for antennas.

Gert Pedersen Frølund (Senior Member, IEEE) was born in 1965. He received the B.Sc. degree (Hons.) in electrical engineering from the College of Technology, Dublin, Ireland, in 1991, and the M.Sc. degree in electrical engineering and the Ph.D. degree from Aalborg University, Aalborg, Denmark, in 1993 and 2003, respectively.

Since 1993, he has been with Aalborg University, where he is currently a Full Professor heading the Antenna, Propagation and Networking Laboratory with 36 researchers. He is also the Head of the Doctoral School on Wireless Communication with some 100 Ph.D. students enrolled. He has also involved as a consultant for developments of more than 100 antennas for mobile terminals, including the first internal antenna for mobile phones in 1994 with lowest specific absorption rate (SAR), first internal triple-band antennas in 1998 with low SAR and high TRP and TIS, and lately various multiantenna systems rated as the most efficient on the market. He has involved most of the time with joint university and industry projects and has received more than U.S. $12 M in direct research funding. Latest, he is the Project Leader of the SAFE Project with a total budget of U.S. $8 M investigating tunable front end, including tunable antennas for the future multiband mobile phones. He has been one of the pioneers in establishing over-the-air (OTA) measurement systems. The measurement technique is now well established for mobile terminals with single antennas and he was chairing the various COST groups (swg2.2 of COST 259, 273, 2100, and now ICT1004) with liaison to 3GPP for OTA test of MIMO terminals. He is also deeply involved in MIMO OTA measurement. He has published over 175 peer-reviewed papers and holds 28 patents. His research has focused on radio communication for mobile terminals, especially small antennas, diversity systems, and propagation and biological effects.

Ming Shen (Senior Member, IEEE) was born in Yuxi, China. He received the M.Sc. degree in electrical engineering from the University of Chinese Academy of Sciences, Beijing, China in 2005 and the Ph.D. degree in wireless communications, with the spar nord annual best thesis nomination, from Aalborg University, Aalborg, Denmark, in 2010.,He is currently an Associate Professor in RF and mm-wave circuits and systems with the Department of Electronic Systems, Aalborg University, Denmark.

He has 20 years experience in RF and millimeter wave circuits and systems, including 12 years experience in CMOS RF/mixed-signal IC design. His research interests include circuits and antennas for 5G and satellite communications, low power CMOS RF and millimeter wave circuits and systems, circuits and systems for biomedical imaging, and artificial intelligence.,Dr. Shen is the Grant Holder and PI of two Danish national research projects, and the management committee member substitute from Denmark in the EU COST Action IC1301 with the aim to gather the international efforts and address efficient wireless power transmission technologies. He isthe TPC member of IEEE NORCAS, serves as a Reviewer for IEEE and Kluwer.