

A Hybrid Electromagnetic Optimization Method Based on Physics-Informed Machine Learning

Yanan Liu , Hongliang Li , and Jian-Ming Jin , *Fellow, IEEE*

Abstract—In this article, we present an optimization method based on the hybridization of the genetic algorithm (GA) and gradient optimization (grad-opt) and facilitated by a physics-informed machine learning model. In the proposed method, the slow-but-global GA is used as a pre-screening tool to provide good initial values to the fast-but-local grad-opt. We introduce a robust metric to measure the goodness of the designs as starting points and use a set of control parameters to fine tune the optimization dynamics. We utilize the machine learning with analytic extension of eigenvalues (ML w/AEE) model to integrate the two pieces seamlessly and accelerate the optimization process by speeding up forward evaluation in GA and gradient calculation in grad-opt. We employ the divide-and-conquer strategy to further improve modeling efficiency and accelerate the design process and propose the use of a fusion module to allow for end-to-end gradient propagation. Two numerical examples are included to show the robustness and efficiency of the proposed method, compared with traditional approaches.

Index Terms—Design optimization, genetic algorithm, physics-informed machine learning.

I. INTRODUCTION

AS WIRELESS systems increasingly find broader applications, the demand for high performance microwave devices employed by these systems becomes more challenging to fulfill. In order to accommodate a wide variety of functionalities within a compact form factor, modern high-performance electronics are packed with microwave devices with intricate structures and numerous design variables, operating at high speeds and across broad bandwidths. To meet the demands and address the stringent constraints imposed by the commercial deployment of these devices, efficient and robust design optimization methods are highly desirable.

Optimization of microwave devices has been of great interest to researchers in the EM community for decades [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. A wide variety of methods have been proposed, which can be classified into two main categories [19]: search-based and gradient-based methods. The genetic algorithm (GA) [20] is one of the most popular search-based methods. While GA offers the

potential to search for global optima in the presence of a large number of design variables, the method can be prohibitively expensive due to the repeated forward evaluations required.

Gradient-based optimization methods (grad-opt), on the other hand, are based on the analytical evaluation of gradients of the objective function with respect to design variables. Fast convergence is expected when the initial design is in the vicinity of the optimal solution. However, it is a local method whose performance greatly depends on the starting point and has no guarantee of reaching global optima [21].

In this work, we propose a hybrid algorithm for EM optimization which combines the slow-but-global GA with the fast-but-local gradient optimization. In the proposed method, the GA optimizer is used as a pre-screening tool to search for good initial values. Once a good starting point is found, the gradient optimizer is invoked to refine the solution in the hope of reaching an optimum quickly. This is repeated until a given computational budget is exhausted. We introduce a robust metric to measure the goodness of the designs as starting points, and use a set of control parameters to balance exploration versus exploitation and fine tune the optimization process. To further facilitate the proposed optimization scheme, we incorporate a neural network-based surrogate model in both GA and gradient descent algorithm. The neural net model can speed up the forward evaluation in GA by several orders of magnitude, and can provide analytical gradients using standard backpropagation in a very fast and accurate manner. With the introduction of the neural net model, the most computationally intensive part of the optimization is shifted to model pre-training. To alleviate this computational burden, we propose the use of a machine learning model that incorporates the physics knowledge of analytic extension of eigenvalues (AEE) [22], to improve modeling efficiency and reduce training data generation cost. To further improve modeling efficiency and to accelerate the design process, we employ the divide-and-conquer strategy by cutting the device into sub-components and modeling each individual component separately. A fusion module is introduced to cascade the components' responses numerically and allow the gradient to propagate from the overall objective function to the design variables. We apply our method to two numerical examples to demonstrate its capability and efficiency.

II. FORMULATION

A. Genetic Algorithm and Gradient Optimization

GA is a global search algorithm based on the concept of natural evolution [20]. Each possible solution in the design space

Manuscript received 30 November 2023; revised 5 March 2024; accepted 29 March 2024. Date of publication 5 April 2024; date of current version 30 April 2024. (Corresponding author: Jian-Ming Jin.)

Yanan Liu and Jian-Ming Jin are with the Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Champaign, IL 61801 USA (e-mail: yliu258@illinois.edu; j-jin1@illinois.edu).

Hongliang Li is with Resonant, Inc., San Mateo, CA 94402 USA (e-mail: hli@resonant.com).

Digital Object Identifier 10.1109/JMMCT.2024.3385451

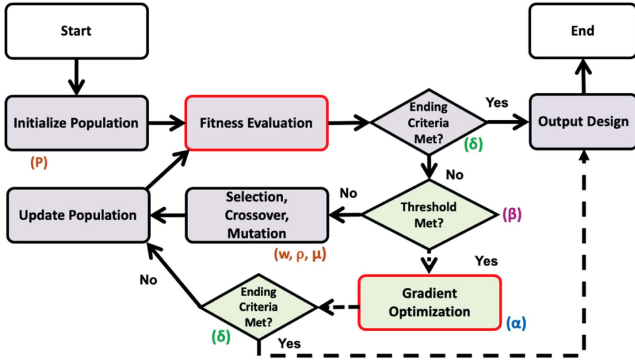


Fig. 1. Flowchart of the proposed hybrid optimization framework.

is represented as a chromosome, and the design objective is cast into a fitness function. There are three main operations upon individual chromosomes, namely selection, crossover, and mutation, which determine the transition from generation to generation. In each generation, the fitness function is evaluated for each individual in the entire population, which is computationally intensive and presents as the main bottleneck of the method. The cost of repeated forward evaluations can be alleviated by using surrogate models, with which the computational burden is shifted to model pre-training, which is conducted out of the optimization loop.

There are four control parameters in a typical GA, namely population size (P), selection probability weight (w), crossover rate (p), and mutation rate (μ). A smaller value of w and larger values of N , p , and μ will lead to more randomness in the GA process, and thus making it more exploration-oriented and more computationally expensive.

Gradient optimization is known for its fast convergence and its proneness to getting stuck in local minima. The basic gradient descent follows the update rule:

$$x_{i+1} = x_i - \alpha \cdot \frac{\partial J_i}{\partial x_i}, \quad (1)$$

where x is the design variable, J is the objective function to be optimized, and α is a parameter that determines the step size in the gradient direction. A large α can provide fast convergence, but can also lead to overshoot and high oscillation. A small α can smooth out the optimization process, but can also make it prohibitively slow.

The most costly part of grad-opt is gradient evaluation, which can be greatly accelerated using neural networks, as proposed in [23]. After the model is fully trained, we fix the weights of the network and treat the inputs as the only tunable parameters. The standard backpropagation algorithm [24] will then provide us with the gradients with respect to design variables, which can be used to perform gradient updates.

B. Hybrid GA and Gradient Optimization Framework

A flowchart of the proposed hybrid optimization method is shown in Fig. 1, which consists of two components: GA (blocks shown in purple), and grad-opt (blocks shown in green).

For the hybrid framework, we introduce two global control parameters, the optimization ending criteria δ , and the threshold β for initiating the gradient optimizer. The optimization process will be terminated when either the maximum number of iterations N_{\max} is reached or its objective function J satisfies the condition:

$$J \leq \delta, \quad (2)$$

where δ should be determined based on the design target. If no satisfactory design is found after N_{\max} iterations, we can start the process again with a new random population.

In order to control when to start gradient descent, we define a metric $|\frac{J}{g}|$ as the magnitude of the cost function J divided by its exponentially weighted moving average of its past gradients g , where g is defined as

$$g_i = \eta \cdot g_{i-1} + (1 - \eta) \cdot \nabla J_i, \quad (3)$$

where the subscript denotes the i -th iteration, and $\eta \in (0, 1)$ is a hyper-parameter that controls the number of past iterations taken into consideration. When the following condition is satisfied, a gradient optimization procedure is invoked with x_i as the starting point:

$$\left| \frac{J_i}{g_i} \right| \leq \beta. \quad (4)$$

The idea is that when (4) is satisfied, the cost function value is small and the accumulated gradient is large, corresponding to regions close to optimality. The use of g instead of ∇J comes from the idea of momentum in optimization [25], which effectively replaces the instantaneous gradient by one that is averaged over multiple past iterations, and allows us to avoid highly oscillating regions where gradient descent can easily get trapped.

The threshold β balances exploration versus exploitation. In the extreme case where $\beta = +\infty$, the hybrid optimization reverts to pure gradient descent, whereas GA plays a greater role as β decreases. In practice, β needs to be picked empirically based on the problem, and may need to be adjusted based on the results from a few test runs. Throughout experiments, we find the following value to be a relatively good starting point:

$$\beta = \frac{2\delta}{\frac{1}{P} \sum_{j=1}^P \left| \nabla J(x_1^j) \right|}, \quad (5)$$

where one iteration of gradient evaluation needs to be performed.

C. ML w/AEE Integration

When neural networks are utilized to speed up fitness evaluation in GA and gradient evaluation in gradient descent, the computational burden is shifted to data generation and model pre-training. To lower this cost, we propose to use the ML w/AEE model [26], which incorporates a fast frequency sweep technique called analytic extension of eigenvalues (AEE) [22], [27] into surrogate modeling. This efficient modeling approach allows us to reconstruct the full frequency response using solutions at only a few sampling frequencies, reducing data generation cost by more than an order of magnitude for most practical applications.

Here we briefly cover the basics of the ML w/AEE model. A more detailed formulation, including full derivative calculation can be found in [28].

The impedance matrix of an N -port device can be represented with its eigenvalues λ_i and eigenvectors v_i . Without loss of generality, let's consider a two-port device, where we have:

$$\mathbf{Z} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} = \sum_{i=1}^2 \frac{\lambda_i \begin{bmatrix} v_{i,1} \\ v_{i,2} \end{bmatrix} [v_{i,1} v_{i,2}]}{v_{i,1}^2 + v_{i,2}^2}. \quad (6)$$

We can use lumped equivalent circuits (LEC) to model the behavior of these eigenvalues ($\lambda_i = \beta_i + j\alpha_i$), with circuit parameters RLGC, which we note as modal equivalent circuit parameters. Using a second-order LEC, the eigenvalues (α and β) can be represented as [28]:

$$\alpha_i = \frac{\omega L_{1,i}}{1 - \omega^2 L_{1,i} C_{1,i}} + \frac{\omega L_{2,i}}{1 - \omega^2 L_{2,i} C_{2,i}} \quad (7)$$

$$\beta_i = \frac{R_{1,i} + \omega^2 L_{1,i}^2 G_{1,i}}{(1 - \omega^2 L_{1,i} C_{1,i})^2} + \frac{R_{2,i} + \omega^2 L_{2,i}^2 G_{2,i}}{(1 - \omega^2 L_{2,i} C_{2,i})^2}. \quad (8)$$

The eigenvectors can also be modeled using RLGC, based on the eigenvector-eigenvalue identity [29]. In particular, we have [28]:

$$v_{i,j} = \pm \sqrt{\frac{\alpha_i - \gamma_j}{\alpha_i - \alpha_{3-i}}} (i = 1, 2; j = 1, 2) \quad (9)$$

$$\gamma_i = \frac{\omega L_{1,i}^s}{1 - \omega^2 L_{1,i}^s C_{1,i}^s} + \frac{\omega L_{2,i}^s}{1 - \omega^2 L_{2,i}^s C_{2,i}^s}, \quad (10)$$

where the superscript in L^s and C^s denotes the corresponding LEC parameters for the sub-circuits.

We can use a neural network to learn the mapping between a device's geometrical parameters \mathbf{x} and its modal LEC parameters. This can be written compactly as:

$$\mathbf{y} = F(\Theta, \mathbf{x}) = f_1(\Theta_1, f_2(\Theta_2, f_3(\dots, f_n(\Theta_n, \mathbf{x}))), \quad (11)$$

where f_1, f_2, \dots, f_n are the layers of the network; Θ denotes the network parameters to be trained; input \mathbf{x} is a multi-dimensional vector containing the geometrical profile of the device; and \mathbf{y} represents the modal LEC parameters RLGC.

During model training, we tune the model parameters Θ to minimize a error function, such as the mean squared error (MSE), between the model prediction \mathbf{y} and the ground truth $\hat{\mathbf{y}}$ over the training data set of size L . This objective is given as:

$$\min_{\Theta} \frac{1}{L} \sum_{n=1}^L (\hat{\mathbf{y}}^{(n)} - \mathbf{y}^{(n)})^2. \quad (12)$$

Once the model is fully trained, it can be used for fast forward evaluation, where the model is fed with a new input \mathbf{x}^* and will output the corresponding response $\mathbf{y}^* = F(\mathbf{x}^*)$; it can also be used for gradient calculation, where we obtain $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ based on backpropagation, and $\frac{\partial J}{\partial \mathbf{x}}$ based on the chain rule. In particular, we have:

$$\frac{\partial J}{\partial \mathbf{x}} = \frac{\partial J}{\partial S} \cdot \frac{\partial S}{\partial \mathbf{y}} \cdot \frac{\partial \mathbf{y}}{\partial \mathbf{x}}, \quad (13)$$

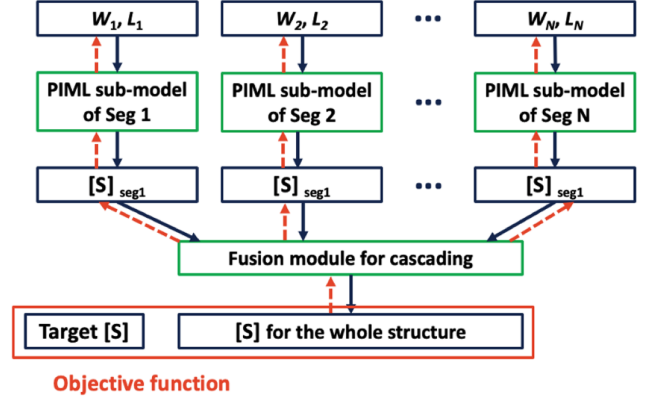


Fig. 2. Overview of the stacked ML w/AEE model, with the fusion module introduced for end-to-end gradient propagation. Blue arrows indicate the forward process, and orange arrows indicate the inverse process.

where S is the S-parameters of the device, which can be obtained analytically from RLGC parameters based on AEE principles and Z-to-S transformation. However, we will use a neural net module to obtain S numerically, as will be illustrated in the next section.

D. Divide-and-Conquer Modeling and the Fusion Module

To fully take advantage of the benefits of the ML w/AEE model, and to further improve modeling efficiency and accelerate the design process, we propose to adopt the divide-and-conquer strategy. The entire structure is divided into segments and modeled individually. The output from each sub-model is cascaded to obtain the overall response. Note that we should divide at locations with smooth and continuous structures such that they incur minimal higher order mode excitation and coupling between segments. Moreover, we should avoid design parameters around the segmentation boundaries, so that the previous condition can be met within the entire design space. The cascading process can be done analytically [30], but that would make it difficult to propagate gradient from the objective function to design variables of each individual segment, hindering its use in the hybrid optimization framework. To address this issue, we propose to use a neural net block called the fusion module that performs the cascading numerically. This makes the overall model end-to-end interconnected and allows for efficient gradient propagation. An illustration of this stacked ML w/AEE model is shown in Fig. 2.

The fusion module takes the RLGC parameters of each individual segment as its input, and generates the S-parameters of the entire device as:

$$S = H(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M), \quad (14)$$

where the subscript $m \in [1, M]$ denotes the p -th segment and $\mathbf{y}_m = F_m(\mathbf{x}_m)$. The fusion module is typically a model of small size (3-layer in our applications) and can be trained efficiently [31] after the sub-models F_1 to F_m are fully trained, requiring no additional cost for data generation.

Algorithm 1: Hybrid GA + Gradient Optimization.

```

Define hyper-parameters  $\alpha, \beta, \eta, \delta, P, N_{max}, w, \rho, \mu$ 
Initialize a population of individuals  $x^1, \dots, x^N$  in the
design space
while  $i \leq N_{max}$  do
  Evaluate fitness for each individual in terms of the
  objective function  $J(x_i^j)$ 
  if  $J(x_i^j) \leq \delta$  then
    End program and output  $x_i^j$ 
  else
    Evaluate gradient for each individual  $\nabla J(x_i^j)$ 
    if  $i=1$  then
       $g_i = \nabla J(x_i)$ 
    else
      Calculate momentum  $g_i = \eta \cdot g_{i-1} +$ 
       $(1 - \eta) \cdot \nabla J_i$ 
    if  $\left| \frac{J_i}{g_i} \right| \leq \beta$  then
      Start gradient descent with initial point  $x_i^j$ 
      if  $J \leq \delta$  during gradient descent then
        End program and output design
      else
        Add gradient optimized design to population
        and continue GA
      end if
    else
      Perform GA operations: selection with
      probability  $w$ , crossover with rate  $\rho$ , mutation
      with rate  $\mu$ 
      Update population
    end if
  end if
end while

```

The overall algorithm for the hybrid optimization method is listed in the box below. Note that within each optimization iteration, we need to evaluate the fitness for and calculate the gradient with respect to every individual in the population. The computational overhead will not be manageable without the use of the stacked ML w/AEE model.

III. NUMERICAL RESULTS

We now consider two specific examples to illustrate the process of the proposed hybrid optimization method and demonstrate its better performance than the traditional optimization methods.

A. Optimization of a Stepped-Impedance Low-Pass Filter

Here we present results on the optimization of a stepped-impedance low-pass filter [27], as shown in Fig. 3. The filter consists of eight microstrip stubs and is divided into four segments. The ports for each segment are fabricated with microstrip lines with two different characteristic impedances, 50 and 120 Ω . There are six design parameters, namely $x = [L_1, L_2, L_3, W_1, W_2, W_3]^T$ mm, where L_{1-3} and W_{1-3} are the

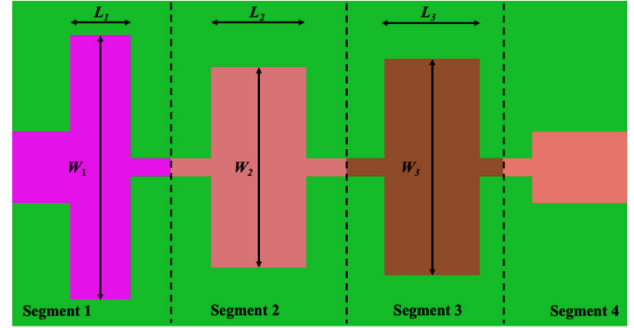


Fig. 3. Schematic of the stepped-impedance low-pass microstrip filter.

length and width of the middle microstrip stub in Segments 1-3, respectively. A full-wave solver is used to simulate the frequency response of sample designs and generate training and validation data. The goal is to design the filter to have reflection below -15 dB up to 2 GHz. We define the objective function J as

$$\begin{aligned}
 J = \Sigma_f \left[\mathcal{R} \left(S_{11}^\dagger - \hat{S}_{11} \right)^2 + \mathcal{I} \left(S_{11}^\dagger - \hat{S}_{11} \right)^2 \right. \\
 \left. + \mathcal{R} \left(S_{21}^\dagger - \hat{S}_{21} \right)^2 + \mathcal{I} \left(S_{21}^\dagger - \hat{S}_{21} \right)^2 \right] \quad (15)
 \end{aligned}$$

where \hat{S} and S^\dagger are the design response and target response, respectively. The target response is provided as a square-shaped mask and we are enforcing both the real and imaginary parts simultaneously.

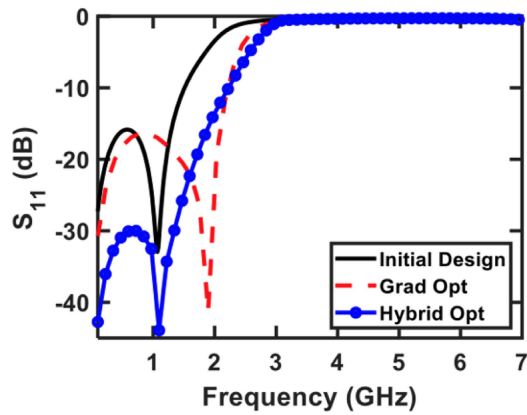
For the GA optimizer used in this work, we use real-valued encoding, with the population size $P = 50$, and the probability of crossover and mutation being $\rho = 0.6$ and $\mu = 0.05$, respectively. The step size in grad-opt is set to $\alpha = 0.01$, the inertia parameter in momentum is set to $\eta = 0.9$, the overall ending criteria is $\delta = 0.25$, and the threshold is $\beta = 1$. The optimization will run for a maximum number of $N_{max} = 50$ iterations.

For the experiments, we have two different initial designs, denoted as configuration #1 and configuration #2, with the design variables listed below.

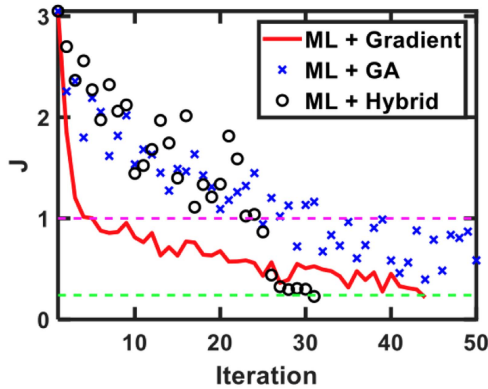
- 1) Configuration #1: $x = [2, 12, 7, 10, 6.5, 10]^T$ mm
- 2) Configuration #2: $x = [1.5, 9, 6.5, 9, 4.5, 9]^T$ mm

We have chosen these two initial designs such that configuration #1 is closer to the target design, whereas configuration #2 is further away. This helps us compare the robustness and efficiency of the the proposed method with GA and grad-opt. The results for these two starting configurations are given in Figs. 4 and 5, respectively.

For configuration #1, both the proposed method and standard grad-opt are able to achieve the design target, as shown in Fig. 4(a), which plots the magnitude of S_{11} for three designs, namely the initial design and the optimized designs obtained with grad-opt and the proposed method. In particular, as shown in Fig. 4(b), it takes GA 50 iterations, 750 forward evaluations, and 340 seconds to reach the optimal design; it takes grad-opt 44 iterations, 88 forward evaluations, and 84 seconds to reach the optimal design; whereas for the hybrid method, it takes 31



(a)

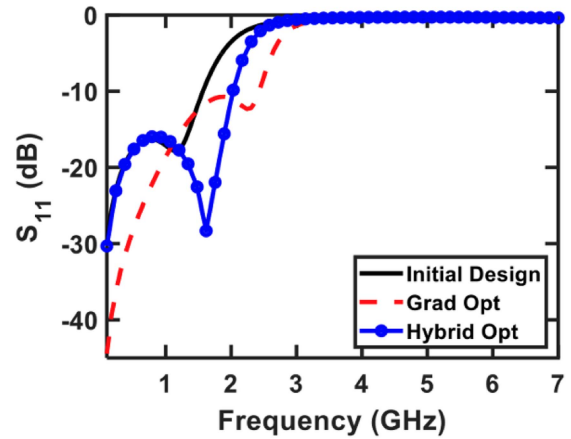


(b)

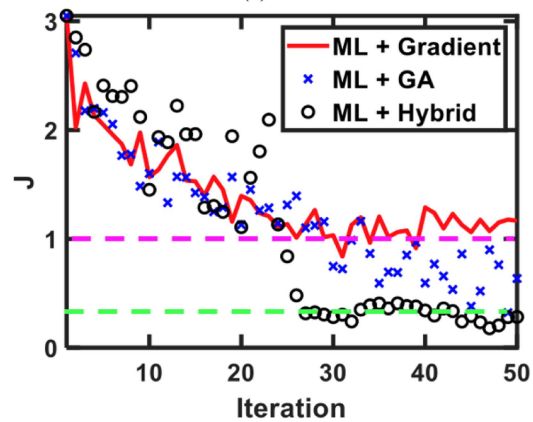
Fig. 4. Optimization results for the stepped-impedance filter example using three different methods: GA, grad-opt, and the proposed method. The initial design is given as configuration #1, where its response is relatively close to the target response. (a) Magnitude of S_{11} for optimized structures; (b) Loss function versus iteration.

iterations, among which 25 are GA iterations and 2 are gradient updates, 637 forward evaluations, and 181 seconds in total. From these results, we can see that the hybrid approach is much more efficient than traditional GA, achieving the design target in half as much time. However, it is more costly than the standard grad-opt. As expected, this is typically the case when the initial design is already close to the design target. Fig. 5 shows the results for configuration #2. In this case, grad-opt improves only slightly from the starting point and fails to achieve the design target, whereas the hybrid method successfully reaches a viable solution. Fig. 5(b) shows the progression of the cost function over iterations. For grad-opt, the cost function decreases in the beginning but quickly saturates. For GA, the cost function heavily fluctuates, but is trending downwards. For hybrid optimization, the cost function behaves like GA in the first few iterations, but drops quickly once good starting points are found and the gradient optimizer is invoked. Further testing shows that for this example, GA is able to reach a solution in around 80 iterations, whereas grad-opt would fail even with smaller step size. This shows the robustness of the proposed method, where it can reach an optimal solution regardless of the initial design.

To better evaluate the robustness of the proposed method, we conducted an additional experiment, where we perform 100



(a)



(b)

Fig. 5. Optimization results for the stepped-impedance filter example using three different methods: GA, grad-opt, and the proposed method. The initial design is given as configuration #2, where its response is further away from the target response. (a) Magnitude of S_{11} for optimized structures; (b) Loss function versus iteration.

runs of the optimization using different methods with random initial designs. For each run we set the maximum number of iterations to 50, and record the objective function value during the process. In Fig. 6, we plot the means as the solid lines, and one standard deviation confidence intervals as the shaded areas. We can see that gradient optimization has a huge trial-to-trial variation, and can get trapped in local minima resulting in sub-optimal designs. This is one major issue that our proposed method aims to overcome.

Table I shows a comparison of the computational costs using five different approaches, namely HFSS with GA, HFSS with gradient descent, the hybrid method with standard neural-net-based surrogate model, and the hybrid method with ML w/AEE. We include the data generation and model pre-training cost for the two ML-based methods, and calculate the average cost per design for 100 optimization runs. Note that we only run the two HFSS-based methods for one trial to record the time per iteration. The average numbers of iterations for these methods are estimates based on their counterparts that incorporate ML w/AEE. Also note that for HFSS with gradient descent, a significant portion (42 out of 100) of the experiments do not

TABLE I
COMPARISON OF COMPUTATIONAL COST FOR DIFFERENT METHODS FOR THE STEPPED-IMPEDANCE FILTER EXAMPLE

	HFSS (GA)	HFSS (Gradient)	Hybrid + NN	Hybrid + ML w/AEE	GA + ML w/AEE
Data generation and model training	NA	NA	756 (min)	108 (min)	108 (min)
Avg. time per iteration	15 (min)	4 (min)	2.4 (s)	3.2 (s)	3.7 (s)
Avg. #iterations taken	68	38	46	42	68
Avg. cost per design (over 100 runs)	1,020 (min)	152 (min)	9.4 (min)	3.3 (min)	5.27 (min)

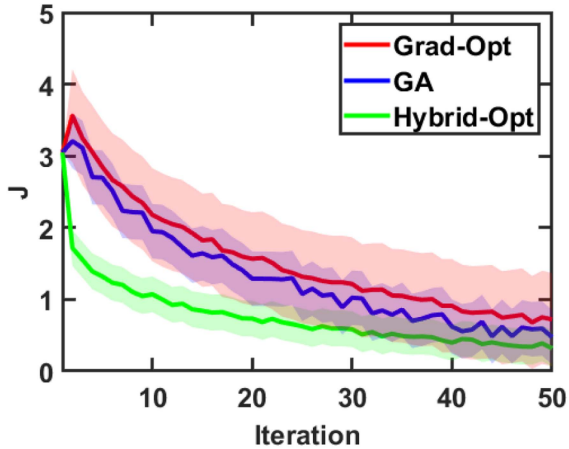


Fig. 6. Comparison of the proposed method with GA and gradient descent in the robustness experiment for the stepped-impedance filter example. We plot means and one standard deviation confidence intervals of the objective function value across 100 trials.

result in a design that meets the design target. The average number of iterations for this method is calculated based on those experiments that successfully converged. HFSS-based methods do not involve any cost for data generation or model pre-training, but they need to incur multiple full-wave simulations at each design iteration, which makes them very expensive. Moreover, for each new design, they have to start from the ground up as there is no information shared from run to run. On the other hand, ML-based methods scale much better in that the model can be re-used and the one-time cost for data generation and model training will be thinned out through multiple runs. This makes them perfect for applications where repeated design is needed, e.g., microwave filters that have the same topology but with different target responses. We also note the two major benefits of using ML w/AEE compared to standard neural net models. 1) The data generation cost for ML w/AEE is greatly reduced compared to the standard NN; 2) Because ML w/AEE models have superior generalization performance, they serve as more robust surrogate models and facilitate a faster convergence when used in optimization. Due to these benefits, the hybrid method with ML w/AEE is overall much more efficient.

B. Optimization of a Hairpin-Line Band-Pass Filter

In the second example, we consider the design optimization of a hairpin-line band-pass filter [32], as shown in Fig. 7. This period-symmetric microstrip hairpin-line filter consists of two components: Component 1 and Component 2, and has seven design variables, i.e., $x = [L_1 \ W_1 \ d_1 \ L_2 \ W_2 \ d_2 \ S]^T$ (mm). Here,

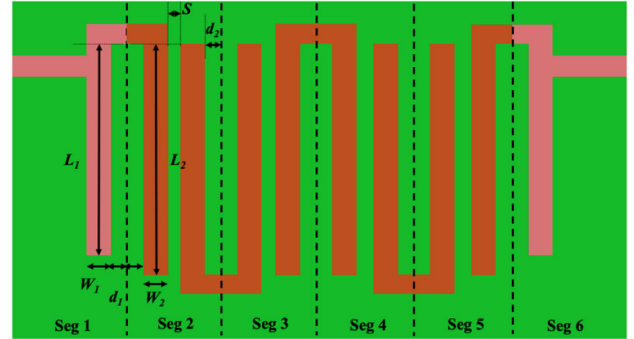


Fig. 7. Schematic of the hairpin-line band-pass filter.

L_1 and W_1 are the length and width of the coupled line in Component 1, respectively. d_1 is the distance between the coupled line in Component 1 and the dividing boundary of Component 1 and Component 2. L_2 and W_2 are the length and width of the coupled line in Component 2, respectively. d_2 is the distance between the coupled line in Component 1 and the dividing boundary of Component 2 and Component 2. S is the distance between the two coupled lines in Component 2. W is the line width of the two ports of Component 1 and Component 2, and the corresponding port impedance is designed to be $50 \ \Omega$. We use HFSS for the full-wave EM simulation of each part and generate training and test data. The goal is to design a band-pass filter with below -10 dB reflection loss between 3.5 and 4.5 GHz. The overall model consists of six sub-models, and the objective function is given as (15), where the design target is provided as a square-shaped mask response. All the hyper-parameters have the same value as the previous example, except for $\beta = 2$ and $\delta = 0.5$, indicating that the proposed method is not sensitive to the choice of hyper-parameters.

For the experiments, we again choose two different initial designs, denoted as configuration #1 and configuration #2, with the design variables listed below.

- 1) Configuration #1: $x = [9.2, 1, 1, 9.5, 1.2, 0.9, 0.4]^T$ mm
- 2) Configuration #2: $x = [8.5, 0.9, 0.9, 9, 1, 0.7, 0.3]^T$ mm

These two initial designs are chosen such that configuration #1 is closer to the target design, whereas configuration #2 is further away. The results are given in Fig. 8. We can see that for configuration #1, both the proposed method and standard grad-opt are able to achieve the design target. However, for configuration #2, grad-opt only improves slightly from the starting point and fails to achieve the design target, whereas the hybrid

TABLE II
COMPARISON OF COMPUTATIONAL COST FOR DIFFERENT METHODS FOR THE HAIRPIN-LINE FILTER EXAMPLE

	HFSS (GA)	HFSS (Gradient)	Hybrid + NN	Hybrid + ML w/AEE	GA + ML w/AEE
Data generation and model training	NA	NA	2120 (min)	271 (min)	271 (min)
Avg. time per iteration	26 (min)	7 (min)	3.4 (s)	4.2 (s)	4.9 (s)
Avg. #iterations taken	64	40	49	43	64
Avg. cost per design (over 100 runs)	1,664 (min)	280 (min)	23.9 (min)	5.7 (min)	7.9 (min)

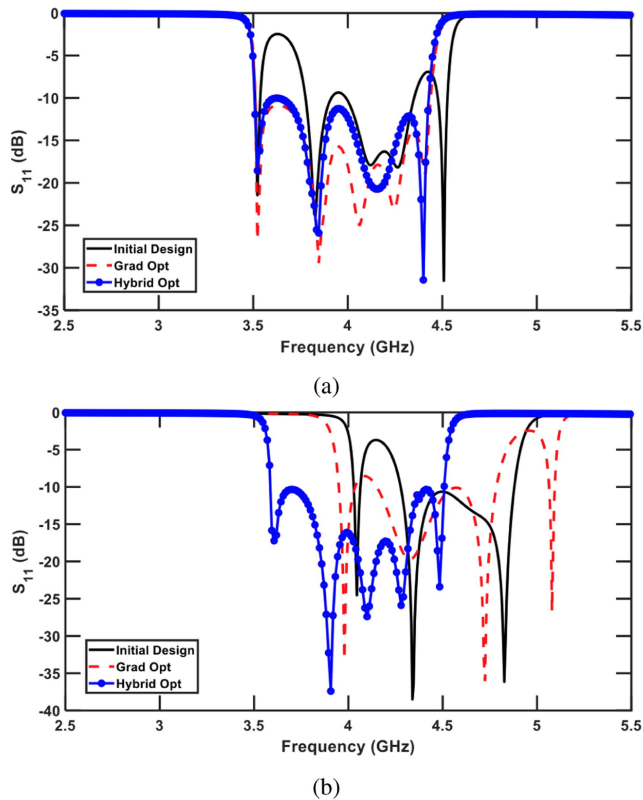


Fig. 8. Optimization results for the hairpin-line filter example using gradient optimization and the proposed method. (a) The initial design is given as configuration # 1, where its response is relatively close to the target response. (b) The initial design is given as configuration # 2, where its response is further away from the target response.

method successfully reaches a viable solution. This again shows the robustness of the proposed method.

We conduct a similar experiment for this example with 100 random runs of the optimization using different methods. The results are shown in Fig. 9, where the means are plotted as the solid lines, and one standard deviation confidence intervals as the shaded areas. We observe a similar pattern where the performance of gradient optimization heavily depends on the starting point, making it very prone to getting trapped in local minima. On the other hand, the proposed method is able to reach the design target much more consistently. Table II lists a comparison of the computational costs using five different approaches for this example. We observe a similar pattern where HFSS-based GA and gradient-opt methods are much more costly when repeated designs are needed. In addition, for this more

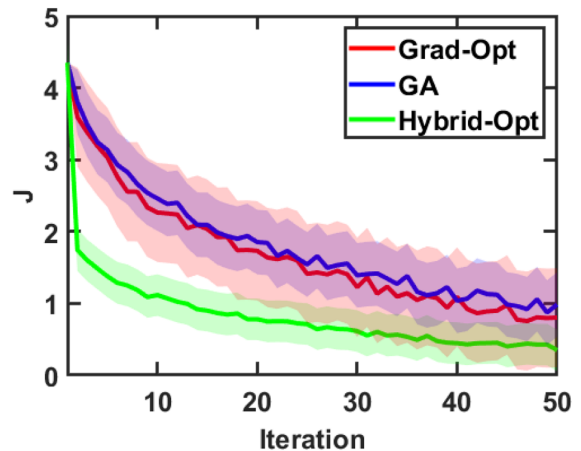


Fig. 9. Comparison of the proposed method with GA and gradient descent in the robustness experiment for the hairpin-line filter example. We plot means and one standard deviation confidence intervals of the objective function value across 100 trials.

complicated example, the benefit of using ML w/AEE compared to standard neural net model is even more significant.

IV. CONCLUSION

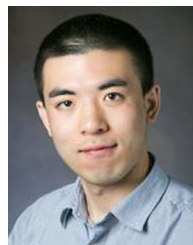
In this article, we presented an efficient and robust hybrid optimization method that combines GA and gradient descent optimization. In this approach, GA is used to start the optimization with random search and heuristics-based evolution. During the GA process, individual designs that satisfy a certain goodness criteria are seeded as the starting point for gradient-based updates, where an optimal solution can be reached quickly within only a few iterations. To facilitate the optimization process, we proposed the use of a neural network model that can speed up fitness evaluation in GA and gradient calculation in grad-opt. To further improve modeling efficiency and accelerate the design process when dealing with large number of design variables, we adopted the divide-and-conquer strategy, which is fully compatible with the ML w/AEE model. We introduced a special neural network block called the fusion module to perform component cascading numerically, allowing the gradients to be passed from the objective function to design variables.

By leveraging the advantages of both global and local methods, our proposed approach has been found to be robust and efficient, achieving optimal solutions more consistently than gradient-based methods, and in fewer iterations than traditional GA. The incorporation of ML w/AEE model greatly alleviates

the computational burden, both during the iterative design loop, and during model pre-training. The hybrid method can be used as an off-the-shelf optimizer for various design optimization problems. We have made available a Python implementation of the proposed method at https://github.com/yngliu/Hybrid_Optimization_with_MLAEE.

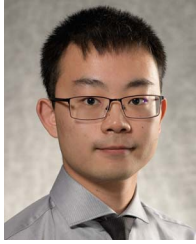
REFERENCES

- [1] B. Liu, H. Aliakbarian, Z. Ma, G. A. E. Vandenbosch, G. Gielen, and P. Excell, "An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques," *IEEE Trans. Antennas Propag.*, vol. 62, no. 1, pp. 7–18, Jan. 2014.
- [2] B. Liu et al., "An efficient method for complex antenna design based on a self adaptive surrogate model-assisted optimization technique," *IEEE Trans. Antennas Propag.*, vol. 69, no. 4, pp. 2302–2315, Apr. 2021.
- [3] J. E. Rayas-Sánchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 420–435, Jan. 2004.
- [4] J. E. Rayas-Sánchez and V. Gutierrez-Ayala, "EM-based Monte Carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 12, pp. 4528–4537, Dec. 2006.
- [5] J. E. Rayas-Sánchez, S. Koziel, and J. W. Bandler, "Advanced RF and microwave design optimization: A journey and a vision of future trends," *IEEE J. Microw.*, vol. 1, no. 1, pp. 481–493, Jan. 2021.
- [6] S. Lei, Y. Yang, H. Hu, Z. Zhao, B. Chen, and X. Qiu, "Power gain optimization method for wide-beam array antenna via convex optimization," *IEEE Trans. Antennas Propag.*, vol. 67, no. 3, pp. 1620–1629, Mar. 2019.
- [7] H. M. Torun and M. Swaminathan, "High-dimensional global optimization method for high-frequency electronic design," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 6, pp. 2128–2142, Jun. 2019.
- [8] S. Koziel and S. Ogurtsov, "Multi-objective design of antennas using variable-fidelity simulations and surrogate models," *IEEE Trans. Antennas Propag.*, vol. 61, no. 12, pp. 5931–5939, Dec. 2013.
- [9] S. Koziel, S. Ogurtsov, W. Zieniutycz, and A. Bekasiewicz, "Design of a planar UWB dipole antenna with an integrated Balun using surrogate-based optimization," *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 366–369, 2014.
- [10] F. Feng, W. Na, J. Jin, W. Zhang, and Q.-J. Zhang, "ANNs for fast parameterized EM modeling: The state of the art in machine learning for design automation of passive microwave structures," *IEEE Microw. Mag.*, vol. 22, no. 10, pp. 37–50, Oct. 2021.
- [11] D. E. Root, "Future device modeling trends," *IEEE Microw. Mag.*, vol. 13, no. 7, pp. 45–59, Nov./Dec. 2012.
- [12] F. Mkadem and S. Boumaiza, "Physically inspired neural network model for RF power amplifier behavioral modeling and digital predistortion," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 4, pp. 913–923, Apr. 2011.
- [13] M. B. Steer, J. W. Bandler, and C. M. Snowden, "Computer-aided design of RF and microwave circuits and systems," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 3, pp. 996–1005, Mar. 2002.
- [14] A.-D. Huang, Z. Zhong, W. Wu, and Y.-X. Guo, "An artificial neural network-based electrothermal model for GaN HEMTs with dynamic trapping effects consideration," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 8, pp. 2519–2528, Aug. 2016.
- [15] H. Yu, H. Chalamalasetty, and M. Swaminathan, "Modeling of voltage-controlled oscillators including I/O behavior using augmented neural networks," *IEEE Access*, vol. 7, pp. 38973–38982, 2019.
- [16] S. Liao, H. Kabir, Y. Cao, J. Xu, Q.-J. Zhang, and J.-G. Ma, "Neural-network modeling for 3-D substructures based on spatial EM-field coupling in finite-element method," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 1, pp. 21–38, Jan. 2011.
- [17] S. A. Sadrossadat, Y. Cao, and Q.-J. Zhang, "Parametric modeling of microwave passive components using sensitivity-analysis-based adjoint neural-network technique," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 5, pp. 1733–1747, May 2013.
- [18] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q.-J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4140–4155, Oct. 2019.
- [19] M. Kovaleva, D. Bulger, and K. P. Esselle, "Comparative study of optimization algorithms on the design of broadband antennas," *IEEE J. Multiscale Multiphys. Comput. Techn.*, vol. 5, pp. 89–98, 2020.
- [20] J. Johnson and V. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas Propag. Mag.*, vol. 39, no. 4, pp. 7–21, Aug. 1997.
- [21] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [22] H. Li, J.-M. Jin, D. R. Jachowski, and R. B. Hammond, "Fast frequency sweep analysis of passive miniature RF circuits based on analytic extension of eigenvalues," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 1, pp. 4–14, Jan. 2021.
- [23] J. Peurifoy et al., "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.*, vol. 4, no. 6, 2018, Art. no. eaar4206.
- [24] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 65–93.
- [25] Y. Nesterov et al. *Lectures on Convex Optimization*. Berlin, Germany: Springer, 2018, vol. 137.
- [26] Y. Liu, H. Li, and J.-M. Jin, "Physics-embedded machine learning for efficient modeling of high-frequency circuits," in *Proc. IEEE Int. Symp. Antennas Propag. USNC-URSI Radio Sci. Meeting, 2022*, pp. 834–835.
- [27] H. Li, J.-M. Jin, D. R. Jachowski, and R. B. Hammond, "Second-order analytic extension of eigenvalues for fast frequency sweep analysis of RF circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 4, pp. 2078–2087, Apr. 2021.
- [28] W. Liu et al., "Electromagnetic parametric modeling using combined neural networks and RLGC-based eigenfunctions for two-port microstrip structures," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4666–4682, Nov. 2022.
- [29] P. Denton, S. Parke, T. Tao, and X. Zhang, "Eigenvectors from eigenvalues: A survey of a basic identity in linear algebra," *Bull. Amer. Math. Soc.*, vol. 59, no. 1, pp. 31–58, 2022.
- [30] K. Zhang, C.-F. Wang, and J.-M. Jin, "A hybrid FETD-GSM algorithm for broadband full-wave modeling of resonant waveguide devices," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 9, pp. 3147–3158, Sep. 2017.
- [31] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.
- [32] T. Singh, J. Chacko, N. Sebastian, R. Thoppilan, A. Kotrashetti, and S. Mande, "Design and optimization of microstrip hairpin-line bandpass filter using DOE methodology," in *Proc. IEEE Int. Conf. Commun., Inf. Comput. Technol.*, 2012, pp. 1–6.



Yanan Liu received the B.S. degree in electromagnetics and wireless technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2014, and the M.S. degree in electrical engineering in 2017 from the University of Illinois Urbana-Champaign, Champaign, IL, USA, where he is currently working toward the Ph.D. degree. Since 2014, he has been a Research and Teaching Assistant with the Center for Computational Electromagnetics, Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign.

His research interests include modeling and design optimization of electromagnetic devices using machine learning techniques. He is a Reviewer for multiple journals, including IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, ANTENNA AND PROPAGATION, IEEE ACCESS, *Light: Science and Applications*, and *International Journal of Numerical Modeling: Electronic Networks, Devices and Fields*.



Hongliang Li was born in Luohe, China, in 1994. He received the B.S. degree in electrical engineering and information science from the University of Science and Technology of China, Hefei, China, in 2017, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois Urbana-Champaign, Champaign, IL, USA, in 2020 and 2022, respectively. From 2017 to 2022, he was a Research Assistant with the Center for Computational Electromagnetics, University of Illinois Urbana-Champaign.

He is currently a Computational Scientist with Resonant Inc. - A Murata Company, San Mateo, CA, USA. His research interests include multiphysics simulation, time- and frequency-domain finite element methods, domain decomposition methods, numerical modeling of acoustic wave devices, electrical machines, and RF circuits, and the development of fast algorithms. Dr. Li is a Reviewer for multiple journals, including IEEE TRANSACTIONS ON ULTRASONICS, FERROELECTRICS, AND FREQUENCY CONTROL, IEEE TRANSACTIONS ON POWER ELECTRONICS, IEEE JOURNAL ON MULTISCALE AND MULTIPHYSICS COMPUTATIONAL TECHNIQUES, and *International Journal of Numerical Modeling: Electronic Networks, Devices and Fields*. He was the recipient of the Paul D. Coleman Outstanding Research Award and the John Bardeen Memorial Graduate Research Award from the Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, in 2021 and 2022, respectively.



Jian-Ming Jin (Fellow, IEEE) is currently the Y. T. Lo Chair Professor with the Department of Electrical and Computer Engineering and the Director with the Electromagnetics Laboratory and Center for Computational Electromagnetics, University of Illinois Urbana-Champaign, Champaign, IL, USA. He has authored and coauthored more than 280 articles in refereed journals and 22 book chapters. He has also authored *The Finite Element Method in Electromagnetics* (Wiley, First Edition, 1993, Second Edition, 2002, and Third Edition, 2014), *Electromagnetic Analysis*

and Design in Magnetic Resonance Imaging (CRC, 1998), *Theory and Computation of Electromagnetic Fields* (Wiley, First Edition, 2010, and Second Edition, 2015), and coauthored *Computation of Special Functions* (Wiley, 1996), *Fast and Efficient Algorithms in Computational Electromagnetics* (Artech, 2001), and *Finite Element Analysis of Antennas and Arrays* (Wiley, 2008). His research interests include computational electromagnetics, multiphysics modeling, scattering and antenna analysis, electromagnetic compatibility, high-frequency circuit modeling and analysis, bioelectromagnetics, and magnetic resonance imaging. Dr. Jin was elected by the ISI as one of the world's most cited authors in 2002 and is also a Fellow of the Optical Society of America, Electromagnetics Academy, and Applied Computational Electromagnetics Society. He was an Associate Editor and the Guest Editor of IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, *Radio Science*, *Electromagnetics*, *Microwave and Optical Technology Letters*, and *Medical Physics*, and is currently the Editor-in-Chief for *International Journal of Numerical Modeling: Electronic Networks, Devices and Fields*. He was the recipient of the 1994 National Science Foundation Young Investigator Award, 1995 Office of Naval Research Young Investigator Award, 1999 ACES Valued Service Award, 2014 ACES Technical Achievement Award, 2015 IEEE APS Chen-To Tai Distinguished Educator Award, 2015 and 2020 IEEE Edward E. Altschuler AP-S Magazine Prize Paper Awards, 2016 ACES Computational Electromagnetics Award, 2017 IEEE APS Harrington-Mitra Computational Electromagnetics Award, and 2020 ECE Distinguished Educator Award from the University of Michigan, Ann Arbor, MI, USA. He was awarded Adjunct, Visiting, Guest, or Chair Professorship by 14 institutions around the world and appointed an IEEE AP-S Distinguished Lecturer in 2015. He was also the recipient of the 1997 and 2000 Xerox Research Awards and was appointed as the first Henry Magnuski Outstanding Young Scholar in 1998 and later as a Sony Scholar in 2005 at the University of Illinois Urbana-Champaign. His name appeared 24 times in the University's List of Excellent Instructors. His students have received the Best Paper Awards in IEEE 16th Topical Meeting on Electrical Performance of Electronic Packaging and 25th, 27th, 31st, and 32nd Annual Review of Progress in Applied Computational Electromagnetics.