Surrogate Modeling With Complex-Valued Neural Nets for Signal Integrity Applications

Oluwaseyi Akinwandee, *Graduate Student Member, IEEE,* Serhat ErdoganG, *Graduate Student Member, IEEE,* Rahul Kumar, and Madhavan Swaminathane, *Fellow, IEEE*

Abstract-Neural networks **(NNs)** are quite attractive **in** creating surrogate models for many signal integrity (SD applications. NN-based surrogate models offer the benefits of reducing the design cycle time and providing the designer with a quick prototype that can efficiently analyze the performance of the SI task. This article, therefore, proposes a new end-to-end learning approach for surrogate modeling using complex-valued NNs, incorporating higher functionality and better representation. This approach introduces a deep complex dense network (CDNet), which is built with complex dense blocks to support complex operations using complex-valued weights, and a physically consistent layer to enforce passivity and causality constraints. We also present a robust inverse multiobjective optimization method to minimize the modeling error and optimize the design space parameters. The results show that our model outperforms state-of-the-art deep surrogate models when tasked with fonvard and inverse learning for a relatively small amount of data. The effectiveness of the proposed approach is demonstrated through two SI design applications, where the model is used to predict broadband S-parameters and obtain optimal design space parameters given the desired target specifications.

Index Terms-Design space exploration, inverse design, neural networks (NNs), packaging, passivity, signal integrity (SD, surrogate modeling.

I. INTRODUCTION

IN A high-speed circuit where data are transmitted and received at peak data rates, proper signal integrity (SI) is essential to satisfy the design specifications. Ridding the high-speed circuits of susceptibilities to SI problems is crucial for the design and development of modern electronic devices, including high-speed interconnects, printed circuit boards (PCBs), and packaging components. This has been a major drive for circuit designers and manufacturers. Designers subject the circuits to SI analysis to account for mismatch, Joss, crosstalk, and reference offset. However, analyzing the SI behavior of electronic systems is challenging, due to the design complexity. To address these challenges, engineers frequently employ both numerical solutions and experimental measurements as a means of assessing the performance of electronic systems in terms of SI. Nevertheless, these techniques can prove to be cumbersome and demanding with respect to computational and time resources. In addition, it may not always be feasible to derive an analytical expression that accurately depicts the SI behavior of the system. In these scenarios, utilizing surrogate models can be a viable option.

A surrogate model is designed to imitate the behavior of the system of interest. Surrogate models are utilized when the desired result cannot be readily obtained through direct measurement or calculation. They act as a substitute, offering a straightforward, fast, and computationally efficient representation of the actual model. Surrogate model development has made extensive use of machine learning (ML) methods, such as artificial neural networks (ANNs) [l], [2], [3], [4], [5], support vector regression (SVR) [6], [7], and Gaussian process regression (GPR) [8], [9], [10]. NNs are quite attractive due to their universal function approximation capability and they can be trained to perform different tasks. Designers can analyze the SI performance of their designs more quickly and improve the system performance by using NNs for surrogate modeling in SI applications. NNs can learn complex and nonlinear relationships between the input parameters and the output response. They can also leverage parallel processing techniques, which can significantly reduce the computation time.

Surrogate models come in all shapes and sizes, but they specifically comprise either/both: I) the forward model [6], [I **l],** [12], [13] or/and 2) the inverse model [14], [15], [16], [17]. Consider a design space X of a parameterized system and the corresponding output response Y, and the forward mapping can be represented as

$$!!7: \boldsymbol{X} \text{ 1-+ } \boldsymbol{Y}. \tag{1}$$

The forward model, which could be a fine or coarse model, is particularly advantageous due to its ability to rapidly evaluate the performance of the system for a given set of design parameters. The use of a forward model enables us to perform a large number of evaluations in a relatively short amount of time. This opens up the possibility of design space exploration, system identification, and sensitivity analysis. In [18], a summary of some recent advancements in ML-based methods for

Manuscript received 18 May 2023; revised IS August 2023; accepted IS September 2023. Date of publication 10 October 2023; date of current version 10 January 2024. This work was supported in part by the National Science Foundation through the Center for Advanced Electronics through Machine Learning (CAEML) under Grant CNS 2137259. *(Corresponding author: Oluwaseyi Akinwande.)*

Oluwaseyi Akinwande and Serhat Erdogan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: oakinwande3@gatech.edu;serhat.erdogan@gatech.edu).

Rahul Kumar and Madhavan Swaminathan are with the School of Electrical Engineering and Computer Science, Pennsylvania State University, State College, PA 16801 USA (e-mail: fmrS28l@psu.edu; mvs7249@psu.edu).

Color versions of one or more figures in this article are available at https://doi.org/l0.1109/fMTI.2023.3319835.

Digital Object Identifier IO.I109/fMTT.2023.3319835

SI and power integrity (PD problems is discussed, providing several examples such as modeling the output waveform from a driver circuit with preemphasis, predicting broadband S-parameters from differential vias, and minimizing the clock skew for 3-D integrated circuits (ICs) among others. Another advantage of the forward model is its ability to provide detailed information about the system's behavior. For example, we can use the forward model to predict the response of the system under various variations and conditions such as process, voltage, and temperature (PVT). This information can be used to improve the design of the system as well as to understand its behavior and performance.

Conversely, the inverse model, also known as the inverse design, is expedient for finding the optimal design parameters that meet the specified requirements. This is achieved by obtaining the inverse mapping

$$g^{-1} : y \rightarrow X. \quad (2)$$

For example, in [19], an inverse design method of obtaining the design parameters of a high-speed link that result in an open eye at the receiver end with maximum eye-height and eye-width characteristics is demonstrated. Furthermore, in [20], the inverse design problem of obtaining the sensitive circuitry design parameters of an active mixer of the RF front end that results in a maximum gain and minimum noise figure is addressed. The inverse design enables us to find optimal design parameters, saving time and resources compared to a trial-and-error approach. Another advantage of the inverse design is its ability to perform tradeoff analyses. Oftentimes, designers have to choose between sets of parameters that have conflicting output specifications. Inverse design serves to facilitate informed decision-making and to pinpoint the optimal balance between conflicting demands. Moreover, it can be used in design space exploration to identify the most promising regions of the design space that satisfy the desired specifications and in sensitivity analysis to observe how the optimal design parameters change when tJ1e system is perturbed. Fig. I offers a broad conceptual overview of a surrogate model.

ln the foregoing, we leverage the benefits to build a surrogate model that can be applied to both forward and inverse modeling. Typically, in early stage prototyping, a designer comes up with an initial design with several parameters in the design space. If the design does not satisfy the target specifications, the designer has to do another iteration and gauge the output response with the target specifications. Usually, the designer explores several parameters and laboriously goes through multiple iterations to satisfy the target specifications. This is the forward modeling challenge. Inverse modeling, on the other hand, starts from the target specifications and generates the circuit parameters that satisfy the target. The main contributions of this article can be summarized as follows.

I) We propose a new end-to-end learning framework with complex-valued data targeted at broadband S-parameter modeling.
2) We propose a deep complex dense network (CDNet) by introducing complex dense blocks built with fully connected layers that support complex operations.

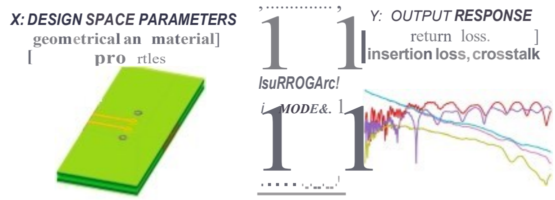

Fig. I. Surrogate model that offers a custom solution.

3) We further propose an inverse multiobjective optimization that minin1izes the modeling error while optimizing the design space parameters that achieve the target specifications.
4) We model physically consistent responses by introducing layers that comply with passivity and causality conditions.
5) We provide a systematic comparison with state-of-the-art deep surrogate models for SI applications.

We demonstrate the effectiveness of the proposed approach for two SI design applications, where the objective is twofold: I) to predict broadband S-pararneters for a given set of design space parameters and 2) to predict the optimal design space parameters given the desired S-parameters and a target band. S-parameter signals are inherently complex-valued due to their amplitude and phase components. Previous approaches of modeling S-parameters using NNs have a less prudent data representation by using only the magnitude component or stacking the real and imaginary parts and handling them as real-valued data [21]. Another approach is to invoke the Kramers-Kronig relations with the Hilbert transform to extrapolate the imaginary component from the real component of the S-parameters [5]. Complex-valued NNs can handle complex numbers directly, allowing them to preserve the amplitude and phase information of the S-parameters. This is crucial in SI applications, where the phase of signals is significant and can impact system behavior. Complex-valued NN increases the expressiveness of the NN model by allowing it to capture interactions between amplitude and phase components, as opposed to its real-valued counterpart. This article builds on the work presented in [22] by providing extensive details of the surrogate methodology.

The rest of this article is organized as follows. Section II provides the motivation and background on complex-valued NNs. Section I11 presents the proposed surrogate model for forward and inverse learning. Section IV explores the application of the proposed model to five-layer vias in package. Section V presents the application to 12-layer vias in package. Finally, we draw a conclusion in Section Vl.

II. COMPLEX BUILDING BLOCKS

In this section, we introduce the building blocks for complex-valued NNs that support complex operations with complex-valued weights and activations, as opposed to its real-valued counterpart. They provide a higher functionality by incorporating the phase information since learnable weights do not just change amplitude as in the real-valued case but can be rotated too in the complex-valued case. We employ

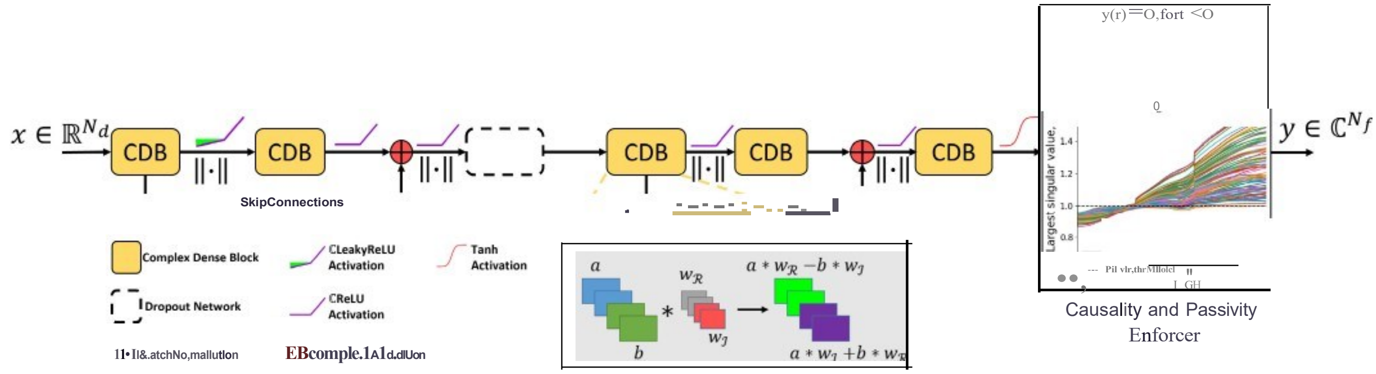

Fig. 2. Proposed deep ICDNet. In inverse modeling, we backpropagate the gradients of the trained ICDNet to update its input parameters and minimize the cost function of the measure of performance. The set of design parameters that minimizes the cost function is the inverse solution. x: design space parameters. y: target specifications.

some innovative approaches from existing literature to form the building blocks needed for a complex-valued NN similar to its real-valued counterpart. In the following discourse, let $z = a + jb$ represent a complex-valued input, where $j = \sqrt{-1}$.

A. Complex Dense Block

The complex dense block introduces feedforward connections from one layer to the next. This encourages feature reusability and strengthens information propagation through the network [23]. Let weight $w = \Re(w) + j\Im(w)$, and the complex dense operator performs the complex operations

$$w * z = (a * \Re(w) - b * \Im(w)) + j(a * \Im(w) + b * \Re(w)) \tag{3}$$

as shown in Fig. 2. In these notations, $*$ denotes the complex dense operator, and $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts of a complex-valued entity, respectively.

B. Complex Activations

As with their real-valued counterparts, complex-valued activations are used to achieve nonlinearity. Designing a complex activation is challenging due to the constraints postulated in Liouville's theorem [24].[1] We employ fully complex activations and split activations where the nonlinearity is applied separately on the real and imaginary parts. Examples include

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{4}$$

$$\mathrm{ICReLU}(z) = \mathrm{ReLU}(a) + j\mathrm{ReLU}(b) \tag{5}$$

$$\mathrm{ICLeakyReLU}(z) = \mathrm{LeakyReLU}(a) + j\mathrm{LeakyReLU}(b). \tag{6}$$

Equations (4)-(6) are the complex hyperbolic tangent, split-complex rectified linear unit (ICReLU), and split-complex leaky rectified linear unit, respectively.

[1] Liouville's theorem states that a complex-valued function that is bounded and analytic everywhere (i.e., a function that is differentiable at every point) is constant. In other words, a function that is bounded and analytic everywhere in the complex plane is not a suitable complex activation function [21].

C. Complex Residual Blocks

The complex residual blocks enable skip connections, which, as the name suggests, skip some of the layers in the NN and add the original input back to the output feature map obtained by passing the input through one or more complex dense layers. This is relevant for preserving contextual information. Furthermore, skip connections prevent the vanishing gradient problem, by directly propagating gradients between layers. Consider a complex NN block that provides a mapping $T(z)$ from the input layer to the output layer. The residual is

$$R(z) = T(z) - z. \tag{7}$$

Equation (7) can be rearranged to form

$$T(z) = R(z) + z \tag{8}$$

which first applies an identity mapping to z and then performs elementwise complex addition.

III. PROPOSED SURROGATE MODEL FOR SI APPLICATIONS

In this section, we present the proposed surrogate model that employs complex-valued NNs for predicting complex and physically consistent representations of S-parameters and performs a multiobjective inverse optimization to achieve the best design space parameters.

A. Forward Model

S-parameters, also known as scattering parameters, are widely used in the design and characterization of high-frequency components and systems. In the context of S-parameters, NNs can be used to learn the mapping from a set of design parameters to the S-parameters of a device.

The proposed model, called deep ICDNet, is shown in Fig. 2. The ICDNet is constructed using complex dense blocks (see Section II) with skip connections between them for preserving contextual information, similar to [25]. Each complex dense block contains one hidden layer of neurons. We apply dropout regularization after the skip connections. ICReLU activation functions and batch normalization layers are used between the complex dense blocks except for the first complex dense block that only has a CLeakyReLU activation. During training, the end-to-end neural net takes the input design parameters x and propagates the information through the network to generate y in the complex domain. We train with a robust ℓ_1- and ℓ_2-supervised loss given by

$$\mathcal{L} = \mathbb{E}_{x,y}\big[\|\Re(y) - \Re(\hat{y})\|_1 + \lambda\|\Re(y) - \Re(\hat{y})\|_2 + \|\Im(y) - \Im(\hat{y})\|_1 + \lambda\|\Im(y) - \Im(\hat{y})\|_2\big] \quad (9)$$

where $\hat{y}$ is the frequency response prediction and $\lambda > 0$ is a "tuning" parameter that provides a tradeoff between the importance of minimizing the fitting error and that of penalizing a large error norm of y. The complex-valued neural net is trained for 250 epochs with an Adam optimizer [26] and a learning rate of 2×10^{-4}.

B. Passivity and Causality Enforcement of Complex S-Parameters in Multiport Networks

Passivity is a fundamental property of linear, time-invariant systems that describes the relationship between the input and output signals. An n-port network is said to be passive if it does not generate any energy on its own and can only dissipate or transfer energy from the input to the output. In the frequency domain, passivity is equivalent to the property that the S-parameters satisfy the following inequality within the frequency band B:

$$S^*(f)S(f) \preceq I \quad \forall f \in B \quad (10)$$

where $(\cdot)^*$ is the Hermitian transpose operator. To verify (10), we apply the singular value decomposition (SVD) to $S(f)$ and obtain its singular values $\sigma_1(f) \geq \sigma_2(f) \geq \cdots \geq \sigma_n(f)$. Therefore, (10) can be rewritten as

$$\sigma_1(f) \leq 1 \quad \forall f \in B. \quad (11)$$

There are several techniques used to ensure that S-parameters are passive in the literature [5], [27], [28], [29]. In this work, we use the method described in [5] where a minimum-phase filter is implemented with minimal computational overhead to enforce the largest singular value of the predicted S-parameter matrix at each frequency point to be less than or equal to 1. This minimum-phase filter is added as a nonlearnable layer to the CDNet.

The minimum-phase filter doubles as a causality enforcer. One important property of a minimum-phase filter is that it has a causal impulse response, which means that the output of the filter depends only on the past and present values of the input signal. This property makes minimum-phase filters suitable for real-time signal processing applications such as the physically consistent layer for S-parameters where a delay in the output signal is undesirable. Having all the poles and zeros of a minimum-phase filter inside the unit circle in the complex plane is a fundamental requirement for ensuring the stability and causality of the filter. A detailed procedure is provided in Algorithm 1.

Algorithm 1 Passivity Enforcement of S-Parameters

Input: S: Predicted complex S-parameter matrix, n: Number of ports, B: Frequency band

Output: Sp: Passive S-parameter matrix

1 Reshape S into a batched matrix form for an n-port network.

2 Transform S into $\bar{S}$ using isomorphism:

$$\bar{S} = \begin{bmatrix} \Re(S) & -\Im(S) \\ \Im(S) & \Re(S) \end{bmatrix}.$$

3 **for** $i : f_i \in B$ **do**

4 Calculate an upper bound for the largest singular value using:

$$\hat{\sigma}_1(f_i) = \sqrt{\frac{P(f_i)}{n} + \sqrt{\frac{n-1}{n}\left(Q(f_i) - \frac{P(f_i)^2}{n}\right)}}$$

where

$$P(f_i) = \sum_{j=1}^{n} |S_{jj}(f_i)|^2 \quad \text{and}$$

$$Q(f_i) = \sum_{j,k=1}^{n} [(S^*(f_i)S(f_i)) \circ (S(f_i)S^*(f_i))]_{jk}$$

5 implement minimum-phase filter as:

$$\Sigma(f_i) = |\Sigma(f_i)|e^{j\phi(f_i)}$$

where

$$|\Sigma(f_i)| = \begin{cases} \frac{1}{\hat{\sigma}_1(f_i)}, & \text{for } \hat{\sigma}_1(f_i) > 1 \\ 1, & \text{for } \hat{\sigma}_1(f_i) \leq 1 \end{cases}$$

$$\phi(f_i) = \mathcal{H}\{\log|\Sigma(f_i)|\}.$$

/* $\mathcal{H}\{\cdot\}$ is the Hilbert transform, operated using a fast Fourier transform approach. */

6 Enforce passivity as:

$$S_p(f_i) = S(f_i)\Sigma(f_i).$$

C. Inverse Optimization

The use of optimization in inverse modeling allows us to adjust the design space parameters, calibrate, and provide the best solutions for our design. By identifying a set of objectives and a measure of the performance of the system, we arrive at optimal solutions for our design. The objective depends on the design parameters we specify. The goal is to find the parameters that optimize the objective. Often, the design parameters are constrained in some way, which we must consider and judiciously optimize to give physically realizable results.

The inverse model relies on the pretrained forward model. After training the forward model, we freeze the ICDNet

Algorithm 2 Inverse Optimization

Input: Initialization $x^{<0>} \in dom(g)$, g: Trained model with the set of all network parameters *0*, B•: Target band, A: learning rate

Output: Estimated x

for $k = 0, 1, 2, \ldots$, *until convergence,* **do**

$y^{(k)} = g(x(k), 0)$

$$\mathscr{C}(\hat{y}^{(k)}) = \sum_{i:f_i \in B^*} \sum_{j=1}^{M} (|\hat{y}^{(k)}_{i,j}| - |y^*_{i,j}|)^2$$

$$\Delta x^{(k)} = -\frac{\partial \mathscr{C}(\hat{y}^{(k)})}{\partial \hat{y}^{(k)}} \frac{\partial \hat{y}^{(k)}}{\partial \theta} \frac{\partial \theta}{\partial x^{(k)}}$$

Update: $x^{(k+1)} \leftarrow x^{(k)} + \lambda \Delta x^{(k)}$

weights and set the design parameters as trainable parameters. We also provide an initial guess of about I00 random sets of design parameters. By backpropagating over the cost function, the CDNet iteratively minimizes the cost function. We can define the cost function as the £2-norm of the difference between the ideal circuit responses (i.e., the desired specifications *y**) and that delivered by the forward model [i.e., y(x)]

$$\mathscr{C}(\hat{y}) = \|\hat{y}(x) - y^*\|_2^2 \tag{L2}$$

for ***M*** frequency responses. Given a target band *B•,* we can rewrite the objective in (12) as

$$C/?(y) = \sum_{i:/;EB^*} \sum_{J=l}^{M} (|\hat{y}_{i,j}(x)| - |y^*_{i,j}|)^2 \tag{13}$$

where *J* 1s the index for the jth frequency response. We minimize the objective in (13) parameterized by the design parameters *x* and the set of all network parameters *0.* We use gradient descent to learn the optimal design parameters that give the minimum cost in (13). The optimized design parameter *x* is the inverse solution. The outline of the inverse optimization method is shown in Algorithm 2. Due to the different nature of NNs, the gradient descent method was a natural choice for the inverse optimization. Also, the CDNet architecture is optimized to have smooth gradients due to the use of complex residual blocks. This leads to having large convex basins around the optimal solution.

D. Optimization Landscape

An optimization landscape is an *N* -dimensional representation of the relationship between an NN's parameters and its performance, where *N* is the dimension of the set of all network parameters *0.* We can use this to visualize and better understand the inverse optimization process of our CDNet. Typically, an optimization landscape can vary greatly depending on the architecture of the network and can be used to identify and diagnose problems with a network's optimization process.

The principal component analysis (PCA) projections method proposed in [30] addresses the problem of analyzing very high-dimensional opttmtzation landscapes. PCA is a dimensionality reduction method used to project the high-dimensional surface of the cost function onto a 2-D plane. PCA obtains the two most important dimensions that influence the optimization landscape and projects the optimization landscape onto these two components.

Consider a set of *N*-dimensional parameters *0* {01, 02,••• , *0N},* which define our NN. The optimization landscape can be represented as a scalar function *C/?(01, 02, ••. , 0N),* where *<,ff* is the cost function. Our objective is to project C/?(01, *0i, ... , 0N)* onto a 2-D plane for visualization. The first step is to compute the covariance matrix :E of the parameters, which is given by

$$:E = \frac{1}{N} \ldots (0; - \mu,)(0; - \mu,)^T \tag{14}$$

where $\mu,$ is the mean of the parameters.

Next, we perform an eigenvalue decomposition on the covariance matrix :E, to find the eigenvectors *(v1, v2,•.• , VN)* and eigenvalues (A-1, .:l2, ..• , *AN).* The PCA method selects the two eigenvectors with the largest eigenvalues as the principal components: *va* and *vp.* As a final step, we obtain a 2-D visualization by projecting the high-dimensional optimization landscape onto these two principal components. The projections are given by

$$u_x = (\theta_\alpha - \mu)^T v_\alpha$$
$$u_y = (\theta_\beta - \mu)^T v_\beta$$

where *Ux* and *Uy* are the coordinates in the 2-D slice of the optimization landscape. In the context of building our inverse surrogate model which is a high-dimensional space, we leverage this concept to see whether the optimizer has a hard time obtaining the optimal solutions.

E. Model Comparison

We perform a comparison with three state-of-the-art methods: I) a generative model called invertible NN; 2) a conditional generative adversarial network; and 3) a deep feedforward NN [with particle swarm optimization (PSO)].

1) Invertible Neural Network: The invertible neural network (INN) is a type of generative model that relies on flow-based generation [31], [32]. It incorporates normalizing flows that leverage the change-of-variable law of probabilities to estimate posterior probability distributions. The INN provides a bijective mapping between the input design space and the output response. With the change-of-variables law of probabilities, we can relate the transformation $Y = f(X)$ as [31], [33]

$$p_X(x) = p_Y(y)\left|\nabla_y f^{-1}(y)\right| \tag{16}$$

where *Px(x)* is the probability density of *x* in the design space ***X,*** *py(y)* is the probability density of *y* in the response space ***Y,*** $f^{-1}(y)$ is the inverse function of the transformation f(X), and !VyJ-1(y)! is the absolute value of the determinant of the Jacobian matrix of the inverse function. More details about the theory and architecture of the INN can be found in [3 1] and [32].

2) Conditional Generative Adversarial Network: We implement a conditional generative adversarial network (cGAN) [34], [35], [36] to achieve inverse modeling. The cGAN comprises two modules called the generator G and discriminator D playing an adversarial game against each other. Both modules are randomly initialized. The training algorithm swaps back and forth between training a generator module (responsible for producing new data) and a discriminator module (responsible for measuring how closely the generator's distribution represents the training dataset). At convergence, the generator learns to produce realistic data, while the discriminator is forced to guess (with probability = 1/2). The cGAN is conditional because we feed the generator module with the desired targets (only) rather than latent noise.[2] The cGAN inherently learns a loss function during training for the inverse mapping $y \to x$ as opposed to hand-tuning the loss function. The penalized cGAN objective can be expressed as [35]

$$\mathcal{L} = \mathbb{E}_{x,y}[\log D(x|y)] + \mathbb{E}_{y}[\log(1 - D(G(y)|y))] + \lambda \mathbb{E}_{x}[\|x - G(y)\|_1]. \quad (17)$$

In (17), the discriminator maximizes the expression, while the generator minimizes the second term. The third term is a supervised loss that makes the generated result $\hat{x} = G(y)$ to be very close to the ground truth x, and λ is a hyperparameter.

3) Deep Feedforward Neural Network With PSO: The architecture of a typical deep feedforward neural network (DNN) consists of the input layer, the dense (hidden) layer(s), and the output layer. To be able to learn complex representations of data, each neuron in a layer connects to every neuron in its previous layer. A neuron takes in a weighted sum of inputs that passes through a nonlinear activation function to produce the output. In this work, for the forward design, the input layer consists of the design parameters that we want to map to the output response, i.e., we form a mapping

$$y = h_L(\ldots(h_2(h_1(x)))\ldots) = f_\theta(x) \quad (18)$$

where h_i denotes the ith layer, x is the design space tuple, y is the output response, f_θ represents all the composition of the hidden layers, and θ is a set of all network parameters. The hyperparameters of the DNN, such as the number of layers, learning rate, choice of optimizer, and training epochs, are chosen to be comparable to that of the CDNet. For the inverse design, we opt for the widely used PSO with its stopping criterion being when the swarm best objective change is less than 1×10^{-8}.

IV. APPLICATION I: MODELING FIVE-LAYER THROUGH-HOLE VIAS IN PACKAGE

To demonstrate the effectiveness of the proposed method, we consider modeling five-layer through-hole vias in package. The vias are drilled through a five-layer PCB and conformally plated with copper. They are arranged in the ground-signal-signal-ground (GSSG) configuration, connecting a differential microstrip line on the top layer to the differential striplines on the third metal layer. These vias are crucial for providing electrical connectivity between the different layers in a semiconductor package, but their intricate design can make it computationally expensive to accurately model their electrical behavior. Therefore, we employ ML-based surrogate modeling to characterize the vias. The design parameters of the via are shown in Fig. 3(a) and the stack-up of the PCB layers is shown in Fig. 3(b). The ranges for the design parameters are given in Table I. The target characteristics investigated are the S_{11} and the S_{21} responses.

The objective here is to build a fast surrogate model that enables the designers to: 1) simulate their designs to

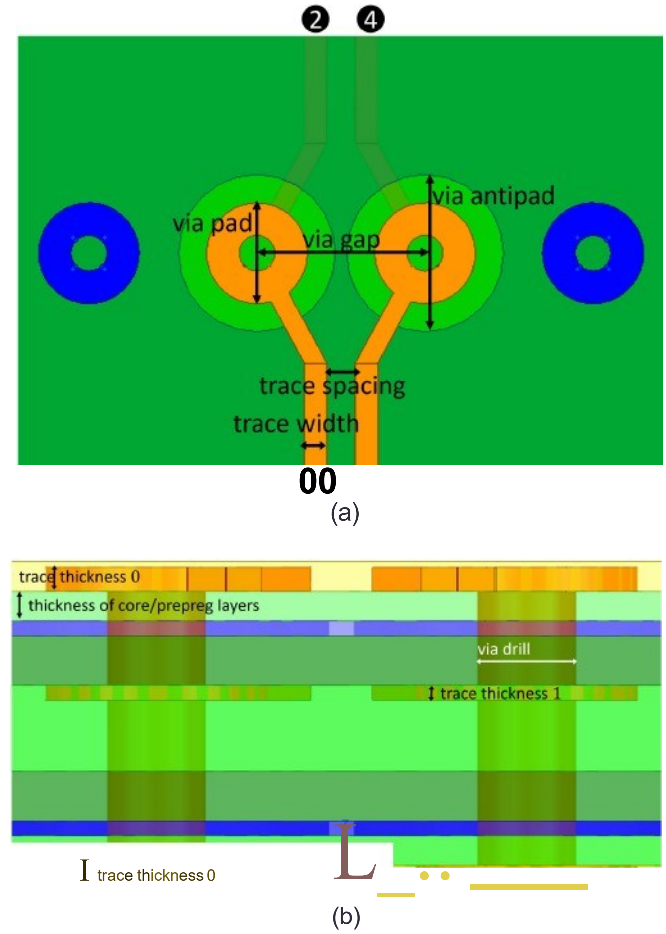

Fig. 3. Geometry of the five-layer through-hole vias. (a) Top view. (b) Front view.

TABLE I
DESIGN PARAMETERS OF FIVE-LAYER THROUGH-HOLE VIAS IN PACKAGE

Parameter		Unit	Min	Max
Trace spacing (first layer)	ts1	mil	3.5	20
Trace thickness (first layer)	tt1	mil	0.6	1.55
Trace width (first layer)	tw1	mil	2.5	12
Trace spacing (third layer)	ts3	mil	3.5	20
Trace thickness (third layer)	tt3	mil	0.6	1.55
Trace width (third layer)	tw3	mil	2.5	12
Anti-pad diameter	da	mil	22	28
Drill diameter	dd	mil	8	10
Pad diameter	dp	mil	16	20

[2]Similar to [35], we also experimented with dropout networks (with probability = 1/2) to achieve stochasticity rather than using latent noise, but this only provided minor stochasticity in the outputs.

ensure that they are withjn a defined threshold of the target specifications of S_{11} and S_{21} and 2) obtain the via design parameters that correspond to a given specification of the S_{11} and S_{21} in the target band. We perform a parametric sweep of the five-layer via design space, with the frequency responses being swept from 0.02 to 20 GHz with steps of 19.98 MHz for each combination of the design parameters. Consequently, each tuple in the design space has the corresponding S_{11} and S_{21} responses with 1001 frequency points. Using Latin hypercube sampling **(LHS)**, we determjne 1000 samples to be analyzed and solved with Ansys HFSS [37], and we extract their S-parameters. Next, the dataset is divided into train and test sets.

A. Forward Model

The goal of the forward model is to train the deep CDNet to learn the forward mapping between the design space x of the five-layer vias and their output response y (i.e., S_{11} and S_{21}). We use six complex dense blocks to trajn the CDNet, each containing one hidden layer of 256 neurons. We apply 40% dropout regularization after the skjp connections. During training, the end-to-end neural net takes in input x with the nine design parameters from the five-layer vias and propagates the information through the network to generate y (i.e., S_{11} and S_{21})-

B. Inverse Optimization

To obtain our inverse surrogate model, we invoke the objective function in (13) and accurately define the ideal S_{11} and S_{21} responses. The purpose of vias is, generally, to act as an interconnect in a vertical direction, as a transmission bne does in a horizontal direction, carrying signals from one layer to another with minimum possible Joss. Therefore, in designjng our five-layer vias, the ideal goal is to allow signals in the target band to pass through with no reflections (i.e., $|S_{11}| = 0$ and $|S_{21}| = 1$). Given a target band B^*, the objective in (12) becomes

$$\mathcal{L}(y) = \sum_{i: f_i \in B^*} |y_{i,1}(x)|^2 + (|y_{i,2}(x)| - 1)^2 \qquad (19)$$

where $y_{i,1}$ and $y_{i,2}$ are the S_{11} and S_{21} responses, respectively, delivered by the forward model at the frequency point f_i. Thjs cost function, parameterized by the design parameters x and the set of all network parameters θ, is then minimized to obtajn the optimal design parameters x^*, whkh is the inverse solution.

C. Results

We present the results from both the forward model training and the inverse optimjzation. We perform inference for the forward model by tabng random samples from the test set. Fig. 4 shows the results obtained from the forward modeling. We compare the real, imaginary, and magnitude components of S_{11} and S_{21} obtained from the forward model with those obtained from the electromagnetic (EM) simulator. We find a close correlation between the output responses from the forward model and the EM simulator.

Next, we display the inverse optimjzation results in Figs. 5 and 6. Given an arbitrary target band $B^* = [0.02, 20]$ GHz, we optimize to find the design parameters of the five-layer vias in package that best achieve this target within the given constraints of the design parameters (see Table I for their respective ranges). Fig. 6 shows that the algorithm was able to deliver a broadband return loss better than 10 dB and an insertion loss better than 0.6 dB

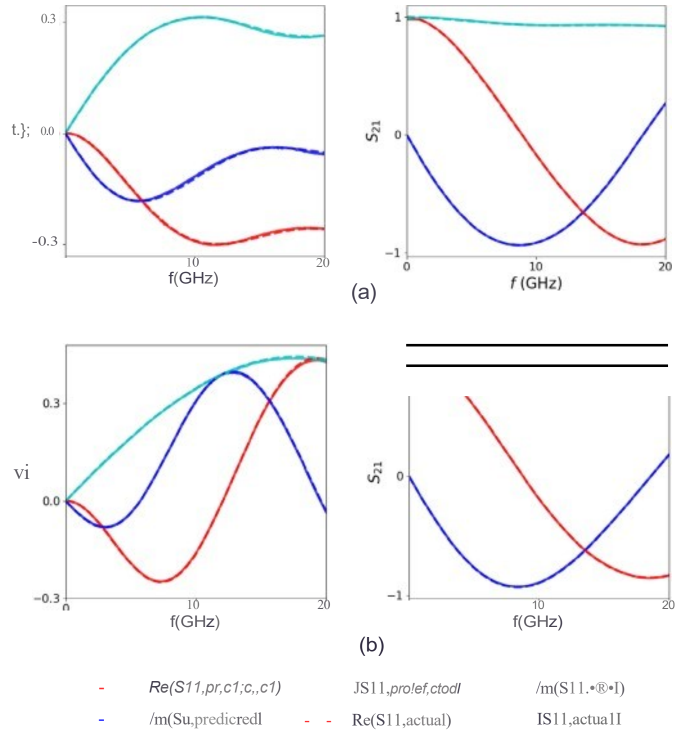

Fig. 4. Forward modeling predictions showing S_{11} and S_{21} for the five-layer vias in package with the trained complex-valued NN model (indicated with solid lines), compared with the EM simulation (indicated with dashed lines) for (a) and (b) two random design tuples in the test set. The real, imaginary, and magnitude components of the frequency responses are shown in red, blue, and cyan, respectively.

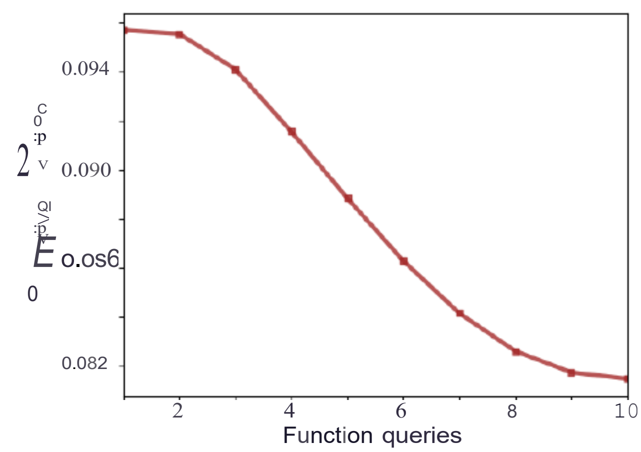

Fig. 5. Inverse optimization results for the five-layer vias in package showing the minimized objective function and function queries.

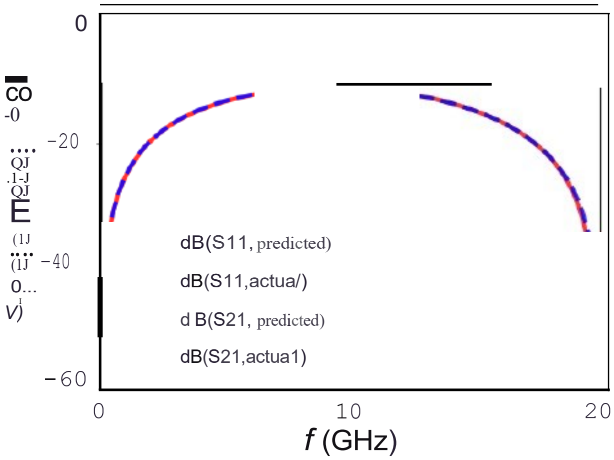

Fig. 6. Inverse optimization results for the five-layer vias in package. The optimization yielded an inverse solution of {19.81, 0.65, 7.37, 5.44, 1.20, 7.59, 24.00, 10.00, 16.00) mil for the design parameters of the five-layer vias in package. For this solution, the model predicts the frequency responses in the solid red and cyan lines, while the EM simulation of the inverse solution gives the frequency responses in the dashed blue and yellow lines.

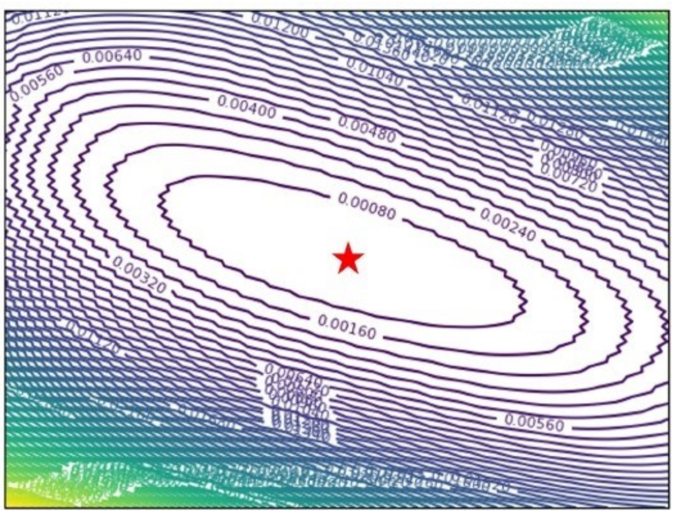

Fig. 7. Contours showing the area around the extremum of a 2-D slice of a very high-dimensional optimization landscape for the surrogate model of the five-layer vias. The large convex basins simplify the optimizer's task of finding the optimal solution (i.e., the red star).

for a prediction of {19.81, 0.65, 7.37, 5.44, 1.20, 7.59, 24.00, 10.00, 16.00} mil corresponding to the design parameters {ts1, t,1, tw1, ts3, t,3, tw3, d_o, dd, dp} of the five-layer vias in package. We emphasize that this solution is not in the test set but was obtained by running the multiobjective inverse optimization. Aside from the evident advantages of preserving amplitude and phase information and achieving physical consistency of complex S-parameters as a fast surrogate model, our proposed method with its inverse optimization (using gradient descent) leverages the differentiable capability of NNs and the smooth gradients of the NN surrogate model. This leads to having large convex basins around the optimal solution, as shown in Fig. 7.

With respect to the passivity enforcement of S-parameters, we achieve physical consistency in the CDNet predictions. Fig. 8 gives the largest singular values σ_1 of the predicted S-parameter matrices for all tuples in the test set. The CDNet

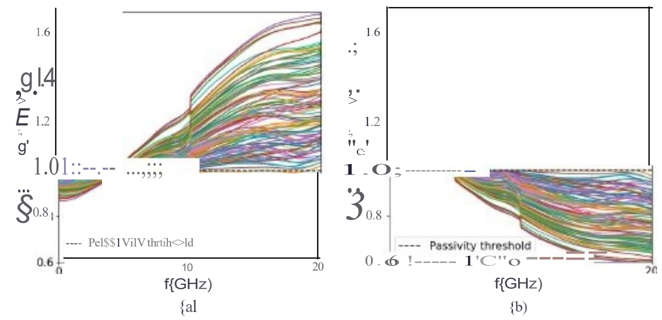

Fig. 8. Achieving physical consistency by enforcing constraints on largest singular values of NN predicted S-parameters. (a) CDNet predictions without passivity enforcement. (b) CDNet predictions with passivity enforcement.

TABLE II

MODEL COMPARISON FOR THE FIVE-LAYER VIAS

Metric		Ours	INN	cGAN	DNN + PSO
Dimensionality of design parameters		9	9	9	9
Frequency points		1001	1001	1001	1001
Number of samples	Train	900	900	900	900
	Test	100	100	100	100
Forward inference error (dB)	S11	0.133	3.639		0.161
	S21	otm.	o.m		0.021
Inverse inference error		0.004	0.055	0.055	0.050
Training time		1.4 min	5.9 min	39.7 s	57.2 s
Forward inference time*		17 ms	16 ms		12 ms
Inverse inference time*		0.4'	15ms	1ms	1.2 s
Forward method		Deterministic	Probabilistic		Deterministic
Inverse method		Iterative	Probabilistic	Probabilistic	Iterative
Model size (network parameters)		6.53×10^5	6.56×10^5	6.59×10^5	6.54×10^5

* For 100 samples.

† All programming is performed with PyTorch [38] on a Windows desktop with Intel® Core™ i7-10700 CPU @ 2.90 GHz and 32 GB RAM.

predictions without passivity enforcement have all $\sigma_1(f) > 1$ with a range of [0.867, 1.703], indicating that passivity is violated, whereas the CDNet predictions with passivity enforcement have all $\sigma_1(f) \leq 1$ with a range of [0.587, 1], indicating that passivity is achieved.

The results from Table II and Fig. 9 compare the performance of the proposed model and state-of-the-art models across several metrics. The metric to assess the numerical accuracy of the forward models is chosen as the mean absolute error (MAE) in decibels, between the actual response y and the predicted response $\hat{y}$, taken over all the frequency points in the response, given by

$$\Delta = \frac{1}{r}\sum_{i=1}^{r}\left|20\log_{10}|y_i| - 20\log_{10}|\hat{y}_i|\right| \quad (20)$$

We find that the CDNet has a combined lower forward inference error by a factor of up to 23 and a lower inverse inference error by a factor of up to 13. It is important to note that this does not generally imply that the INN is a worse model; although its hyperparameters were optimized and its model size is comparable to the other models, its poor performance in this working example can be attributed to a relatively small training set. Being a generative model that learns the underlying probability distribution of the data generating process, the INN naturally excels where the training data

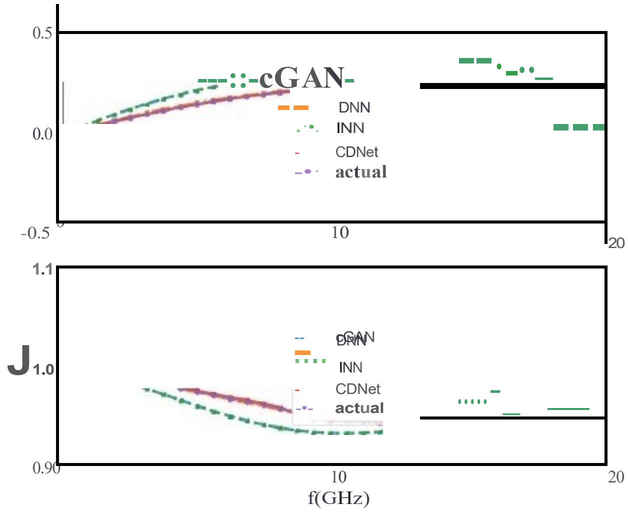

Fig. 9. Comparison of frequency responses simulated from a random test set design tuple obtained from the proposed CDNet model and state-of-the-art models such as INN, cGAN, and DNN for the five-layer vias in package.

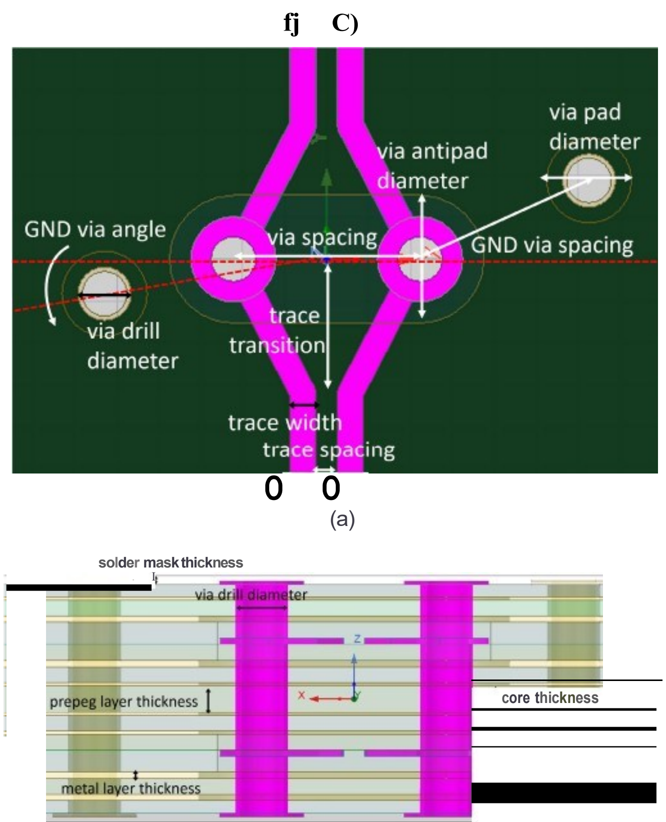

Fig. 10. Geometry of the 12-layer through-hole vias. (a) Top view. (b) Front view.

are abundant In terms of run times, the trained ICDNet takes ~17 ms to generate 100 different broadband S-parameters compared with ~13 h with the EM simulator, for the same number of samples.

V. APPLICATION II: MODELING 12-LAYER THROUGH-HOLE VIAS IN PACKAGE

The second application we consider is modeling 12-layer through-hole vias in package. The vias are drilled through a 12-layer PCB and conformally plated with copper. They are arranged in the GSSG configuration, transitioning from differential striplines on the fourth metal layer to differential striplines on the ninth metal layer. These vias are critical for ensuring that signals are transmitted accurately and efficiently, power is delivered effectively, heat is dissipated efficiently, and the package is able to support high-frequency operations. The design parameters of the via are shown in Fig. I0(a) and the stack-up of the PCB layers is shown in Fig. lO(b). The ranges for the design parameters are provided in Table III. There are a total of 42 design parameters comprising the material and geometrical properties. **In** practice, we only have a set of materials to choose from, but to demonstrate the scalability of the proposed surrogate model, we include the material properties as tunable design parameters and show that the model can still provide an approximate and faster response compared to the EM simulator. The target characteristics predicted are the S_{11} and the S_{21} responses.

Corresponding)y to Section IV, the objective here is to build a fast surrogate model, comprising a forward model that churns out the frequency responses given the design parameters and an inverse model that optimizes the design parameters for a desired response. We determine 2000 samples using LHS and perform a parametric sweep of the 12-layer

TABLE ill

DESIGN PARAMETERS OF 12-LAYER THROUGH-HOLE VIAS IN PACKAGE

Parameter		Unit	Min	Max
Loss tangent (second core)	tan Oc2		0.001	0.05
Relative permittivity (second core)	Er,c2		2	5
Loss tangent (founh core)	tanOc4		0.001	0.05
Relative permittivity (fourth core)	Er,c4		2	5
Loss tangent (sixth core)	tanoc6		0.001	0.05
Relative permittivity (sixth core)	<-r,c6		2	5
Loss tangent (first prepeg layer)	tanop1		0.001	0.05
Relative permittivity (first prepeg layer)	t:r,pl		2	5
Loss tangent (third prepeg layer)	tanop3		0.001	0.05
Relative permittivity (third prepeg layer)	fr,p3		2	5
Loss tangent (fifth prepeg layer)	tan ops		0.001	0.05
Relative permittivity (fifth prepeg layer)	fr,p5		2	5
Loss tangent (solder mask layer)	tan o.o		0.001	0.05
Relative permittivity (solder mask layer)	Er,sO		2	5
Copper thickness (first layer)	ta,1	mil	0.5	1.55
Copper thickness (second layer)	ta,2	mil	0.5	1.55
Copper thickness (third layer)	ta,3	mil	0.5	1.55
Copper thickness (fourth layer)	ta,4	mil	0.5	1.55
Copper thickness (fifth layer)	ta,5	mil	0.5	1.55
Copper thickness (sixth layer)	tcu6	mil	0.5	1.55
Core thickness (second layer)	tc2	mil	2	7
Core thickness (fourth layer)	tc4	mil	2	7
Core thickness (sixth layer)	tc6	mil	2	7
Prepeg thickness (first layer)	tpl	mil	2	7
Prepeg thickness (third layer)	tp3	mil	2	7
Prepeg thickness (fifth layer)	tvs	mil	2	7
Solder mask thickness	t,o	mil	2	7
GND via angle (left)	01	degree	-55	55
GND via spacing (left)	Sg,l	mil	10	50
GND via angle (right)	0r	degree	-55	55
GND via spacing (right)	Sg,r	mil	10	50
Trace spacing (input ports)	St,in	mil	2.5	10
Trace spacing (output ports)	St,oot	mil	2.5	10
Trace transition (input ports)	h;,,	mil	10	30
Trace transition (output ports)	ho.,	mil	10	30
Trace width (input ports)	Wu,	mil	2.5	10
Trace width (output ports)	Wool	mil	2.5	JO
Anti-pad diameter	do	mil	14	32
Drill diameter	dd	mil	4	12
Pad diameter	dv	mil	8	20
Via-plating thickness	tp	mil	0.6	6
Via spacing	s,,	mil	14	50

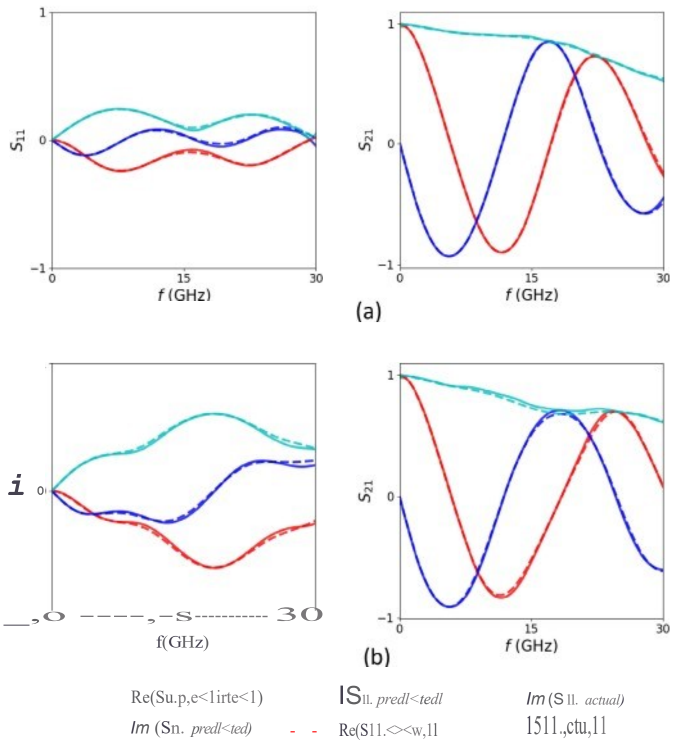

Fig. 11. Fotward modeling predictions showing S11 and S21 for the 12-layer vias in package with the trained complex-valued **NN** model (indicated with solid lines), compared with the EM simulation (indicated with dashed lines) for (a) and (b) two random design tuples in the test set. The real, imaginary, and magnitude components of the frequency responses are shown in red, blue, and cyan, respectively.

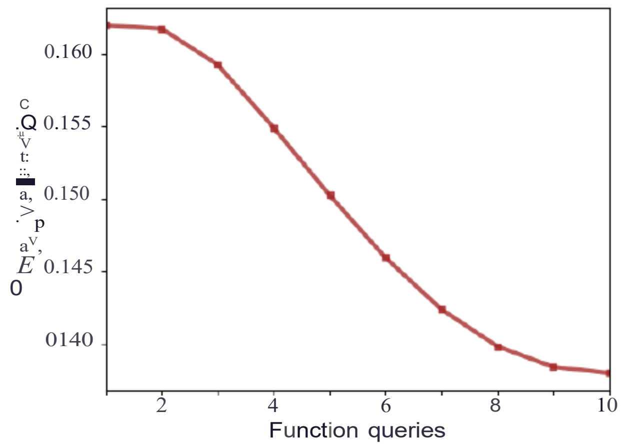

Fig. 12. Inverse optimization results for the 12-layer vias in package showing the minimized objective function and function queries.

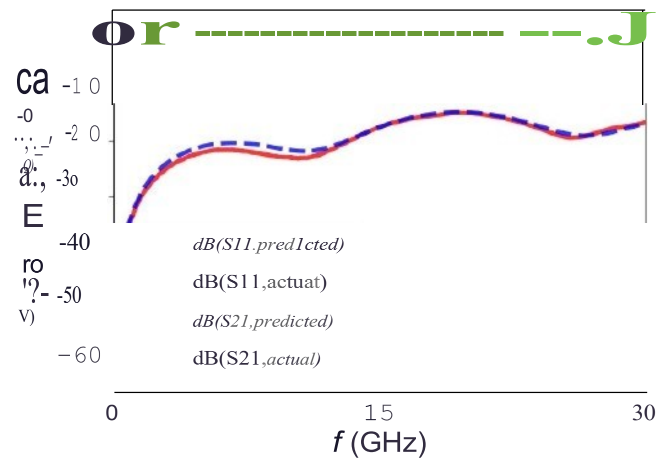

Fig. 13. Inverse optimization results for the 12-layer vias in package. The optimization yielded an inverse solution i for the design parameters of the 12-layer vias in package. For this solution, the model predicts the frequency responses in the solid red and cyan lines, while the EM simulation of the inverse solution results in the frequency responses in the dashed blue and yellow lines.

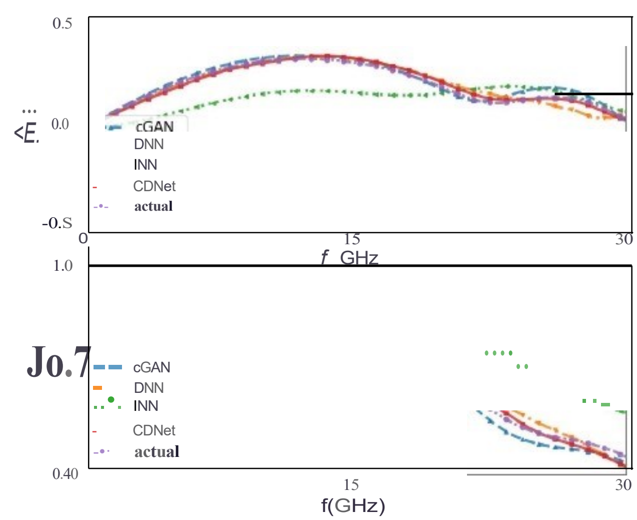

Fig. 14. Comparison of frequency responses simulated from a random test set design tuple obtained from the proposed CDNet model and state-of-the-art models such as INN, cGAN, and DNN for the 12-layer vias in package.

via design space. We extract their S-parameters with Ansys HFSS [37] from de to 30 GHz with steps of 30 MHz, giving 1001 frequency points. Subsequently, we split the dataset into train and test sets. The simulation setups for the forward model and inverse optimization are similar to the setups described in Sections IV-A and IV-B.

A. Results

We show the outcomes of forward learning and inverse optimization. We perform inference for the forward model by taking random samples from the test set. Fig. 11 shows the results obtained from the forward modeling. We compare the real, imaginary, and magnitude components of the S_{11} and S_{21} obtained from the forward model with those obtained from the EM simulator. We find that there is a close correlation between the output responses from the forward model and the EM simulator.

Next, we display the inverse optimization results in Figs. 12 and 13. Given an arbitrary target band $B^* = [0, 30]$ GHz, we optimize the design parameters of the 12-layer vias in package that best achieves this target within the given constraints of the design parameters (see Table III for their respective bounds). Fig. 13 shows that the algorithm was able to deliver a broadband return loss better than 15 dB and an insertion loss better than 4 dB for an inverse solution i of the 12-layer vias in package.

Regarding physical consistency of the CDNet predictions by enforcing passivity constraints on S-parameters, the CDNet predictions with passivity enforcement have all $\sigma_1(f) \leq 1$ in

TABLE IV

MODEL COMPARISON FOR THE 12-LAYER VIAS

Metric		Oun	INN	cGAN	DNN+ PSO
Dimensionality of design parameleers		42	42	42	42
Frequency points		1001	1001	1001	1001
Number of **samples**	Train	1900	1900	1900	1900
	Test	100	100	100	100
Forward inference error'(dB)	*S11*	1.097	5.460		1.177
	"821	0.208	1.004		0.244
Inverse infc.rencc error		O.oJ8	0.074	O.o35	0.130
Training time		3.5 min	7.6 min	38.5 s	2.2 min
Forward inference time		**22 ms**	0.2 s		20 ms
Inverse inference. time		0.4 s	**13 ms**	**2 ms**	2.4 s
Forward method		**Deterministic**	Probabilistic		**Detemtinistjc**
Inverse method		**Iterative**	Probabilistic	Probabilistic	**Iterative**
Model size, (network parameters)		J.58 x 10^6	J.59 x 10^6	1.63 x 10^6	1.60 x 10^6

• **For I00 samples.**
t All programming is performed with PyTorcb [381 on a Windows desktop with Intel® Core™ i7-10700 CPU @ 2.90 GHz and 32 GB RAM.

the test set with a range of [0.611, 1]. However, without passivity enforcement, we find that the range of o-1(f) is [0.768, 1.636], indicating that passivity is violated.

The performance of the proposed model and state-of-the-art models is compared in Fig. 14 and Table IV using several metrics. The results indicate that the ICDNet outperforms the others with a forward inference error reduction by a factor of up to 5 and an inverse inference error reduction by a factor of up to 4. In addition, the ICDNet requires significantly less time to generate 100 broadband S-parameters, taking approximately 17 ms compared to the EM simulator's 13 h.

VI. CONCLUSION

We present both forward and inverse modeling of SI applications by using complex-valued NNs. The forward model takes the input parameters of the actual model, propagates the information through a series of building blocks with complex operations, and generates the physically consistent complex-valued output response. However, in inverse modeling, we propose a well-defined objective as a measure of performance of the SI application and optimize this objective using gradient descent to obtain the optimal input parameters. This surrogate model has the capability of reducing design cycle time and it gives the designer a quick prototype.

REFERENCES

[IJ Q.-J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," / EEE Trans. Mierow. Theory Techn, vol. 51, no. 4, pp. 1339-1350, Apr. 2003.

[2) S. Chen, J. Chen, T. Zhang, and S. Wei, "Semi-supervised learning based on hybrid neural network for the signal integrity analysis," /EEE Trans. Circuits Syst II, Exp. Briefs, vol. ol, no. 10. pp. 1934-1938, Oct 2020.

[3) T. Nguyen et al., "Transient simulation for high-speed channels with recurrent neural network," in Proc. IEEE 27th Conf Electr. Perform. Electron Packag. Syst. (EPEPS), Oct. 2018, pp. 303-305.

[4) H. Yu, M. Swaminathan, C. Ji, and D. White, "A nonlinear behavioral modeling approach for voltage-controlled oscillators using augmented neural networks," in IEEE M1T-S Im. Mierow. Symp. Dig., Jun. 2018, pp. 551-554.

[SJ H. M. Torun, A. C. Durgun, K. Ayglin, and M. Swaminathan, "Causal and passive parameterization of S-parameters using neural networks," IEEE Trans. Mierow. Theory Teclm., vol. 68, no. 10, pp. 4290-4304, Oct. 2020.

[6) T. Lu, J. Sun, K. Wu, and Z. Yang, "High-speed channel modeling with machine learning methods for signal integrity analysis," IEEE Trans. Electromagn. Compat., vol. 60, no. 6, pp. 1957-1964, Dec. 2018.

[7) R. Trinchero and F. G. Canavero, "Modeling of eye diagram height in high-speed links via support vector machine," in Proc. IEEE 22nd Workshop Signal Power Imegrity (SP/), May 2018, pp. 1-4.

[8) H. M. Torun and M. Swaminathan, "High-dimensional global optimization method for high-frequency electronic design," IEEE Trans. Mierow. Theory Techn, vol. 67, no. 6, pp. 2128-2142, Jun. 2019.

[9) Z. Kiguradre et al., "Bayesian optimization for stack-up design," in Proc. IEEE Int. Symp. Electromagn. Compat., Signal Power Integrity (EMC+SIPI), Jul. 2019, pp. 629-634.

[IO) M. A. Dolatsara and M. Swaminathan, "Determining worst-case eye height in low BER channels using Bayesian optimization," in Proc. IEEE I Ith Latin Amer. Symp. Circuits Syst. (LASCAS), Feb. 2020, pp. 1-4.

[I I J Q.-J. Zhang and L. Zhang, "Neural network techniques for high-speed electronic component modeling," in Proc. Im. Mierow. Workshop Ser. Signal Integrity High-Speed Interconnects, Feb. 2009, pp. 69-72.

[12) H. Ma, E.-P. Li, J. Schutt-Aine, and A. C. Cangellaris, "Deep learning method for prediction of planar radiating sources from near-field intensity data," in Proc. IEEE Int. Symp. Electromagn Compal., Signal Power Imegrity (EMC+SIPI), Jul. 2019, pp. 610-olS.

[13) C. M. Schierholz, K. Scharff, and C. Schuster, "Evaluation of neural networks to predict target impedance violations of power delivery networks," in Proc. IEEE 28th Conf Electr. Perform. Electron Packag. Syst. (EPEPS), Oct. 2019, pp. 1-3.

[14) H. Kabir, Y. Wang, M. Yu. and Q.-J. Zhang, "Neural network inverse modeling and applications to microwave filter design," IEEE Trans. Mierow. Theory Teclm., vol. 56, no. 4, pp. 8o7-fil9, Apr. 2008.

[15) M. M. Vai, S. Wu, B. Li, and S. Prasad, "Reverse modeling of microwave circuits with bidirectional neural network models," IEEE Trans. Mierow. Theory Techn, vol. 46, no. 10, pp. 1492-1494, Oct. 1998.

[16) A. Pietrenko-Dabrowska, S. Koziel, and J. W. Bandier, "Rapid microwave optimization using a design database and inverse/forward metamodels," in IEEE M1T-S Int. Mierow. Symp. Dig., Aug. 2020, pp. 868-871.

[17) C. Zhang et al., "Multivalued neural network inverse modeling and applications to microwave filters," IEEE Trans. Mierow. Theory Techn, vol. 66, no. 8, pp. 3781-3797, Aug. 2018.

[18) M. Swaminathan et al.• "Demystifying machine learning for signal and power integrity problems in packaging," IEEE Trans. Compon., Packag., Manuf Tee/mo/., vol. 10, no. 8, pp. 1276-1295, Aug. 2020.

[19) 0. W. Bhatti et al., "Comparison of invertible architectures for high speed channel design," in Proc. IEEE Electr. Design Adv. Packag. Syst (EDAPS), Dec. 2021, pp. 1-3.

[20) M. Swaminathan et al., "Bayesian learning for uncertainty quantification, optimization, and inverse design," IEEE Trans. Mierow. Theory Techn, vol. 70, no. 11, pp. 4620-4634, Nov. 2022.

[21) C. Lee, H. Hasegawa, and S. Gao, "Complex-valued neural networks: A comprehensive survey," IEEE/CAA J. Auto11L Sinica, vol. 9, no. 8, pp. 1406-1426, Aug. 2022.

[22) 0. Akinwande, 0. Waqar Bhatti, K.-Q. Huang, X. Li, and M. Swaminathan, "Surrogate modeling with complex-valued neural nets and its application to design of sub-THz patch antenna-in-package," in IEEE M1T-S Int. Mierow. Symp. Dig., Jun. 2023, pp. 1-6.

[23) M. A. Dedmari, S. Conjeti, S. Estrada, P. Ehses, T. Stocker, and M. Reuter, "Complex fully convolutional neural networks for MR image reconstruction," 2018, arXiv:IB(Jl.03343.

[24) J. Liouville, "Lecons sur les fonctions doublement periodiques faites en 1847. premiere partie. theorie generale," J. Fur Die Reine Und Angew. Math, vol. 88, pp. 277-310, Jan. 1879. [Online]. Available: http://eudml.org/doc/148443

[25) G. Zhang, H. He, and D. Katabi, "Circuit-GNN: Graph neural networks for distributed circuit design," in Proc. 36th Im. Conf. Mach Learn (ICML), vol. 'fl. Long Beach, CA, USA, Jun. 2019, pp. 7364-7373. [Online]. Available: httpsJ/proceedings.mlr.press/v'f7/zhang19e.html

[26) D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf Learn Represent., Dec. 2014.

[27) K. Doshi, A. Sureka, and P. J. Pupalaikis, "Fast and optimal algorithms for enforcing reciprocity, passivity and causality in s-parameters," in Proc. DesignCon, 2012. p. 21.

[28) F. M. Tesche, "On the use of the Hilbert transform for processing measured CW data," IEEE Trans. Electromag11. Compal, vol. 34, no. 3, pp. 259-266, Aug. 1992.

[29] P. Triverio, "Robust causality check for sampled scattering parameters via a filtered Fourier transform," *IEEE Mierow. Wireless Compon Lett,* vol. 24, no. 2, pp. 72--74, Feb. 2014.

[30] M. Gallagher and T. Downs, "Visualization of learning in multilayer perceptron networks using principal component analysis," *IEEE Trans. SysL Man Cybern., B, Cybern,* vol. 33, no. l, pp. 2&-34, Feb. 2003.

[3l] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," 2016. *arXiv:1605.08803*.

[32] L. Ardizzone et al., "Analyzing inverse problems with invertible neural networks," 2018, *arXiv:1808.04730*.

[33] J. A. Gubner, *Random Processes for Electrical and Computer Engineers.* Cambridge, U.K.: Cambridge Univ. Press, 2006.

[34] M. Mina and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[35] P. Isola, J.-Y. Zhu,T. Zhou, and A. A. Efros, "Image-to-image translation with conclitional adversarial networks," 2016, *arXiv:1611.07004*.

[36] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural hif. Process. Syst.,* 2014, pp. 2672-2680.

[37] Ansys Inc. (2022). *Ansys HFSS.* [Online]. Available: https://www.ansys.com/

[38] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.

Ralml Kumar received the bachelor's degree in electronics engineering from Nagpur University, Nagpur, India, in 2012, the M.Tech. degree in very large-scale integration (VLSI) design and automation techniques from the National Institute of Technology Hamirpur, Hamirpur, India, in 2016, and the Ph.D. degree in electrical engineering from IIT Ropar, Bara Phool, Inclia, in December 2021.

He is currently a Post-Doctoral Fellow with the Department of Electrical Engineering and Computer Science, Pennsylvania State University, State College, PA, USA. Prior to that, he was a Post-Doctoral Fellow with the 3D System Packaging Research Center (PRC). School of Electrical and Computer Engineering (ECE), Georgia Institute of Technology, Atlanta, GA, USA. His research area is machine learning applications in signal integrity/electromagnetic compatibility (EMC) analysis and solving complex packaging problems.

Dr. Kumar was a recipient of the Visvesvaraya Fellowship Scheme for his doctoral studies. He delivered a tutorial in the 2020 IEEE Electrical Design of Advanced Packaging and Systems (EDAPS), on the topic "Implications of Thermal Aspects on Interconnect and Packaging Technology-An Electro-Thermal Co-Design Perspective," held in Shenzhen, China.

Oluwaseyi Akinwande (Graduate Student Member, IEEE) received the bachelor's degree in electrical and electronics engineering from the University of Ibadan, Ibadan, Nigeria, in 2018, and the master's degree in electrical engineering from Auburn University, Auburn, AL, USA, in 2021. He is currently pursuing the Ph.D. degree in electrical engineering at the Georgia Institute of Technology, Atlanta, GA, USA, advised by Prof. Madhavan Swaminathan.

He specializes in surrogate modeling and is interested in creating machine learning algorithms to derive models used for electronic design automation with applications in high-speed channels and microwave electronics.

Serhat Erdogan (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the Georgia Institute of Technology, Atlanta, GA, USA.

His current research interests include the design of antennas and other passive RF components for sub-terahertz (THz) applications.

Madhavan Swaminathan (Fellow, IEEE) received the B.E. degree from the Regional Engineering College at Tiruchirapalli (now NITT), Tiruchirapalli, India, in 1985, and the M.S. and Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY, USA, in 1989 and 1991, respectively.

He formerly held the John Pippin Chair in Microsystems Packaging and Electromagnetics at the School of Electrical and Computer Engineering (ECE) and was the former Director of the 3D Systems Packaging Research Center (PRC), Georgia Institute of Technology. He is currently the Head of the Department of Electrical Engineering, Pennsylvania State University, State College, PA, USA. He is also the Site Director of the NSF Center for Advanced Electronics through Machine Learning and he leads the Heterogeneous Integration Area at the SRC JUMP ASCENT Center. Prior to joining the Georgia Institute of Technology, he was with IBM, USA, working on packaging for supercomputers. He is the author of more than 530 refereed technical publications and holds 31 patents. He is the primary author and the coeclitor of three books and five book chapters, and the founder and co-founder of two startup companies.

Dr. Swarninathan is the Founder of the IEEE Conference on Electrical Design of Advanced Packaging and Systems (EDAPS), a premier conference sponsored by the IEEE Electronics Packaging Society (EPS). He has served as a Distinguished Lecturer for the IEEE Electromagnetic Compatibility (EMC) Society.