# Causal and Passive Parameterization of S-Parameters Using Neural Networks

Hakki Mert Torun, *Graduate Student Member, IEEE*, Ahmet Cemal Durgun, *Member, IEEE*, Kemal Aygün, *Senior Member, IEEE*, and Madhavan Swaminathan, *Fellow, IEEE*

***Abstract*— Neural networks (NNs) are widely used to create parametric models of S-parameters for various components in electronic systems. The focus of deriving these models has so far been numerical error reduction between the NN-generated S-parameters and the data source. However, this is not sufficient when creating such NNs since it does not guarantee predicted S-parameters to be physically consistent, i.e., passive and causal, which restricts their use cases. This article, therefore, proposes a causality enforcement layer (CEL) and passivity enforcement layer (PEL) that can be used in NNs, which ensures that NN-predicted S-parameters are of a passive and causal system. To achieve this, we utilize Kramers–Kronig relations and singular value properties of S-parameters during the training stage with the purpose of learning a physically consistent representation. This enables end-to-end training where no postprocessing is required to ensure physical consistency. We demonstrate the effectiveness of the presented approach for three different design applications, where the goal is to predict S-parameters from dc to 100 GHz. The results show that when NNs are trained using CEL and PEL, the predicted S-parameters are characterized as 100.0% causal and passive while having the same level of accuracy as NNs that solely focus on error minimization.**

***Index Terms*— Causality, electromagnetic modeling, high-speed channels, microelectronic packaging, neural networks (NNs), passivity.**

## I. Introduction

EVER-increasing demand for higher computational power has created a trend toward increasing the operational frequency of electronics to increase the bandwidth of both chip-to-chip and wireless signaling. Design of such high-frequency systems often requires handling multiple tradeoffs to meet performance metrics and therein arises the necessity to perform a thorough design space exploration (DSE) and optimization to evaluate different circuit architectures, materials, and geometries, where each design choice goes through a rigorous electrical characterization process. This step commonly includes characterizing the frequency response of passive components such as filters, connectors, and interconnects through multiscale, broadband 3-D electromagnetic (EM) simulations and then exporting the resulting S-parameters to a circuit simulator to be evaluated with the active components such as amplifiers, voltage regulators, and I/O drivers.

Although essential for high-frequency design characterization, the involvement of 3-D EM simulations in the DSE loop significantly increases the overall computational complexity, especially when the number of design choices and parameters increases. To this end, machine learning (ML)-based parametric model development is a popular technique to eliminate repetitive broadband full-wave EM simulations that would otherwise be impractical in terms of time and computational resources. Here, a certain amount of training data is collected and a learning-based model is derived to predict the S-parameters for a given set of input parameters. In areas that involve high-frequency electronics such as microwave design and electronic packaging, ML techniques have been used excessively for optimization [1]–[4] and uncertainty quantification (UQ) [5]–[8] of various passive components.

In particular, two techniques stand out for parameterization of S-parameters using neural networks (NNs), namely considering frequency as a regular input parameter [9]–[11] and neural transfer function (neuro-TF)-based models [12]. In the former case, different design settings are first evaluated at a given frequency band, and then, the training data are constructed by replicating the design parameters, where each copy is associated with a particular frequency point. Once trained, the resulting NN can rapidly predict S-parameters at various design settings and can be utilized in an optimization loop to find the parameter combination that provides the desired performance. The frequency as input parameter approach, however, is only applicable to narrowband S-parameters since the growing number of frequency points corresponds to redundant data replication that leads to large memory requirements during the training phase or to smooth frequency responses where the frequency band can be sparsely discretized without losing predictive accuracy [13]. Neuro-TF-based models, on the other hand, first convert the S-parameters into a frequency-domain transfer function format using the vector fitting (VF) algorithm [14] and then train a fully connected NN (FCNN) to predict the poles and residues [15]. This approach eliminates the frequency from the learning problem and enables a memory-efficient training

Manuscript received March 4, 2020; revised June 7, 2020; accepted July 5, 2020. Date of publication August 12, 2020; date of current version October 5, 2020. This work was supported in part by the NSF under Grant No. CNS 16-24731 - Center for Advanced Electronics through Machine Learning (CAEML). *(Corresponding author: Hakki Mert Torun.)*

Hakki Mert Torun and Madhavan Swaminathan are with the 3D Systems Packaging Research Center (PRC), School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: htorun3@gatech.edu; madhavan.swaminathan@ece.gatech.edu).

Ahmet Cemal Durgun and Kemal Aygün are with the Assembly and Test Technology Development (ATTD) Department, Intel Corporation, Chandler, AZ 85226 USA (e-mail: ahmet.c.durgun@intel.com; kemal.aygun@intel.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMTT.2020.3011449

environment. However, the coefficients of the TF can have a very wide and nonlinear spread when the input sample space is large [16], which leads to the need for problem-specific data preprocessing that does not provide ease of automation, such as tracking movement of specific poles and frequency scaling and shifting [17]. Furthermore, the VF algorithm itself is an approximation that can introduce significant errors to the original problem [18].

Another possible direction to address the limitations of previous approaches is to consider each frequency point as a separate output dimension. This results in a very high-dimensional output space if the frequency bandwidth of interest is large and discretized at a small step size. For FCNNs, this significantly increases the number of learnable parameters, which can result in overfitting. A recently developed approach, namely spectral transposed convolutional net (S-TCNN) [19], addresses this high dimensionality by using the 1-D transposed convolutional layers to exploit spatial correlation in the frequency axis. The weight sharing structure of S-TCNN allows to reduce the number of learnable parameters compared with FCNN while preserving the representation capability and, hence, is suitable for predicting broadband frequency responses.

Although previous NN-based techniques have shown to be successful for design optimization and UQ in many different applications, they cannot be directly used for DSE purposes. Previous NN-based modeling strategies solely focus on numerically matching the predicted S-parameters to the data source and overlook the underlying physical phenomena represented by the data. In the case of multiport S-parameters of passive microwave devices and networks, this can result in NN predicted S-parameters to be noncausal and nonpassive, which prevents the predictions to be used in subsequent time- and/or frequency-domain simulations for DSE or multidomain optimization purposes. Hence, the focus of the NN-based parameterization of S-parameters should not solely be error reduction with respect to the data source but also use the domain-knowledge we have to preserve the physical consistency, i.e., causality and passivity, of the predictions to enable an extended scope of use cases. Compared with the knowledge-based methods that make use of a coarse model to improve the overall prediction accuracy [20], [21], the knowledge in this article is used as a constraint to be enforced on the predictions based on the physical phenomena that hold for S-parameters of any passive microwave device and does not rely on a problem-specific coarse model. It should also be noted that enforcing physical consistency at inference time after the model is trained can lead to substantial accuracy loss, along with computational overhead since such enforcement is a sequential procedure where each prediction needs to be handled one-by-one. This corresponds to losing the parallel prediction capability of the NN, in which every NN-based design optimization method relies upon.

In this article, we therefore develop physically consistent NNs to directly learn a physical representation between input parameters and broadband S-parameters while minimizing the numerical error with respect to the training data. We propose two new layers to be used in an NN, namely causality enforcement layer (CEL) and passivity enforcement layer (PEL). In the CEL, we utilize Kramers–Kronig relations and use Hilbert transform to reconstruct the imaginary part of each element in the predicted S-parameter matrix to ensure the time-domain impulse response matrix is causal. In the PEL, we enforce the largest singular value of the predicted S-parameter matrix at each frequency point to be less than 1 by using a minimum-phase passivity enforcement approach. In [22], we showed very preliminary results of the proposed approach. This article significantly extends [22] by providing extensive details of the methodology and a significant change in PEL and overall network architecture, which are then demonstrated on 3 different design applications that emerge in high-speed channel design.

The rest of this article is structured as follows. Section II provides background on S-parameter causality and passivity. Section III presents the proposed NN architecture with CEL and PEL. Section IV shows the application of the proposed model to a differential plated-through-hole (PTH) structure in package core. Section V presents the application to a differential stripline model in package. Section VI presents the application to a ball-grid-array (BGA) model for package-to-board transition, followed by conclusion in Section VII.

## II. Background

Physical consistency of S-parameters is a well-studied subject [23]–[26]. In this section, we provide a brief summary of the fundamental concepts related to causality and passivity and introduce the notations we use in the subsequent sections.

### A. Causality of S-Parameters

S-parameter matrix of a $P$-port and linear time-invariant (LTI) system is said to be causal if every element in the time-domain impulse response matrix cannot produce an output before the input signal, i.e.,

$$h_{ij}(t) = 0, \quad t < 0 \quad \forall i, j \in P. \tag{1}$$

This condition is satisfied when even and odd parts of the transfer functions are related as

$$\begin{aligned} h_{ij}(t) &= h_{ij}^{(e)}(t) + h_{ij}^{(o)}(t) \\ &= h_{ij}^{(e)}(t) + \text{sign}(t) h_{ij}^{(e)}(t) \end{aligned} \tag{2}$$

where $h^{(e)}(t)$ and $h^{(o)}(t)$ are the even and odd parts of $h(t)$, respectively, and $\text{sign}(t)$ is the signum function that equals to 1 when $t > 0$ and $-1$ when $t < 0$. Taking the Fourier transform of (2) gives

$$\begin{aligned} H_{ij}(f) &= \mathcal{F}(h_{ij}(t)) \\ &= H_{ij}^{(e)}(f) + \frac{1}{j\pi f} * H_{ij}^{(e)}(f) \end{aligned} \tag{3}$$

where $*$ is the convolution operation. As $H_{ij}^{(e)}(f)$ is the Fourier transform of an even function, it is real-valued, leading to

$$\begin{aligned} \text{Im}\{H_{ij}(f)\} &= \frac{-1}{\pi f} * \text{Re}\{H_{ij}(f)\} \\ &= \frac{-1}{\pi} \int_{-\infty}^{\infty} \frac{\text{Re}\{H_{ij}(\widetilde{f})\}}{f - \widetilde{f}} \text{d}\widetilde{f}. \end{aligned} \tag{4}$$

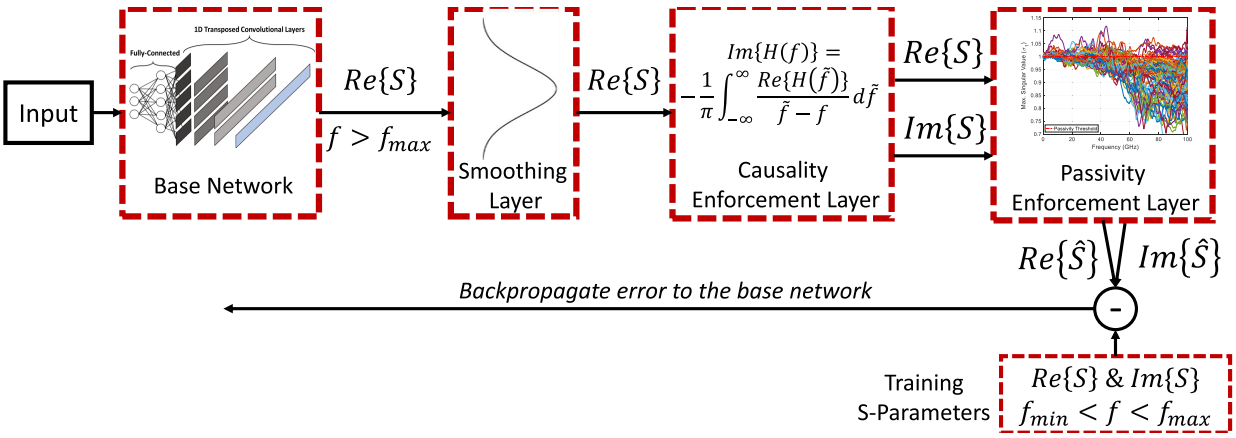


Fig. 1. Proposed physically consistent NN model for passive and causal parameterization of S-parameters.

The convolution integral in (4) is also known as the Hilbert transform. Rewriting (4) for real and imaginary parts leads to the well-known Kramers–Kronig relations [27], written as

$$\begin{aligned} V(f) &= -\mathscr{H}\{U(f)\} \\ &= \frac{-1}{\pi}\int_{-\infty}^{\infty}\frac{U(\tilde{f})}{f-\tilde{f}}\,\mathrm{d}\tilde{f} \\ U(f) &= \mathscr{H}\{V(f)\} \\ &= \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{V(\tilde{f})}{f-\tilde{f}}\,\mathrm{d}\tilde{f} \end{aligned} \tag{5}$$

where $H(f) = U(f) + jV(f)$ and $\mathscr{H}\{\cdot\}$ is the Hilbert transform operator. Hence, the causality condition in (1) is satisfied if and only if the real and imaginary parts of each element in the S-parameter matrix satisfiy the Kramers–Kronig relations and is related through the Hilbert transform.

### B. Passivity of S-Parameters

Although the causality condition is of utmost importance for using S-parameter representation in any time-domain characterization, a significant portion of microwave analysis is performed solely in the frequency domain. The passivity condition, stating that multiport S-parameters for a passive network cannot generate energy, is of paramount importance for both time- and frequency-domain characterizations since any violation of passivity can lead to unstability in time domain [25] and can be significantly amplified by active components and/or when cascaded to other nonpassive S-parameter blocks.

A $P$-port S-parameter matrix defined within the frequency band $\Omega$ is said to be passive if and only if it is bounded as

$$S^H(f)S(f) \leq I \quad \forall f \in \Omega \tag{6}$$

where $(\cdot)^H$ is the Hermitian transpose operator. This condition can be conveniently checked by obtaining the singular values of $S(f)$ via singular value decomposition (SVD) of the S-parameter matrix at every frequency point as

$$\mathrm{SVD}[S(f)] = U(f)\Sigma(f)V^{-1}(f) \tag{7}$$

where

$$\Sigma(f) = \begin{bmatrix} \sigma_1(f) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_P(f) \end{bmatrix} \tag{8}$$

is the ordered singular value matrix with $\sigma_1(f) > \sigma_2(f) > \cdots > \sigma_P(f)$. The passivity condition in (6) then can be rewritten as

$$\sigma_1(f) \leq 1 \quad \forall f \in \Omega. \tag{9}$$

Throughout this article, we will use (9) to check the passivity of a given S-parameter matrix.

## III. Proposed Causal and Passive NN Architecture for Parameterizing S-Parameters

In this section, we present the proposed NN architecture that guarantees the predicted S-parameters to be physically consistent, i.e., satisfy the conditions in (5) and (9), while maximizing numerical accuracy with respect to the training data that are obtained from full-wave EM simulations.

A high-level block diagram of the proposed model is given in Fig. 1 and consists of four blocks, namely, a base network, a learnable smoothing layer, CEL, and PEL. The base network and the learnable smoothing layer contain all the learnable parameters in the overall model that is trained to minimize the training loss function, while CEL and PEL ensure the physical consistency of the predictions. The output of the last block of the model (PEL) is compared with the training data and the error is backpropagated to update the learnable parameters. CEL and PEL are derived to ensure that training and inference complexity of the NN is not substantially increased by not increasing the number of learnable parameters, vanishing or exploding gradients, or increased computational complexity.

### A. Base Network

The base network contains the majority of the learnable parameters in the model. It uses the frequency as output formulation where each frequency point is considered to be a separate output dimension to be predicted. Hence, the base network is responsible for taking the input parameters of the model and upsampling it to output the multiport frequency response. As we are interested in broadband responses at a very small frequency step size, the output of this block becomes high-dimensional. Utilizing conventional fully connected networks for the base network then results in a significantly increased number of learnable parameters, which can cause overfitting.

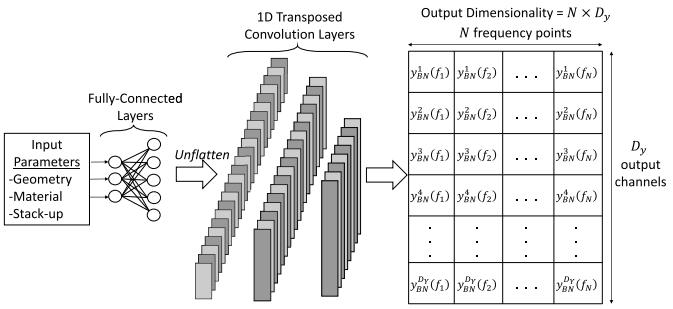


Fig. 2. Overall architecture of the S-TCNN model as the base network.

To address the high dimensionality without giving up the representation capability, the known structure in the data can be exploited. For the case of broadband frequency responses, the structure we have is the spatial correlation of the data points in the frequency axis, i.e., neighboring frequency points being highly correlated with each other.

In this article, we therefore propose to use S-TCNN [19] as the base network of our model. The overall architecture of S-TCNN is given in Fig. 2, where input parameters are first passed through fully connected layers to transform the input space into a latent space that best describes the desired response. Then, this latent representation is upsampled using the 1-D transposed convolutions to form the high-dimensional frequency response. This corresponds to a learnable upsampling operation that preserves the spatial correlation in its output by considering a particular input as the result of a cross correlation operation between the output and the learnable kernel. We refer readers to [19] for a more detailed description of the S-TCNN architecture and comparison to a regular convolutional-type network.

Formally, transposed convolution operation done by a single kernel is equivalent to a zero-padded cross correlation and can be written as matrix multiplication in the form of a Toeplitz matrix [28]. Let $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T$ be the $m$-dimensional input vector and $\mathbf{h} = [w_1, w_2, \ldots, w_k]^T$ be the convolution kernel of size $k$. The output $\mathbf{y}$ can then be written as

$$\mathbf{y} = f(\mathbf{h} *^{\mathsf{T}} \mathbf{x}) = f(H^{\mathsf{T}}\mathbf{x}) \tag{10}$$

with

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad H^{\mathsf{T}} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ w_2 & w_1 & \ddots & \vdots \\ \vdots & w_2 & \ddots & 0 \\ w_k & \vdots & \ddots & w_1 \\ 0 & w_k & \ddots & w_2 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & w_k \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \tag{11}$$

where $*^{\mathsf{T}}$ is the transposed convolution operation, $f(\cdot)$ is the nonlinear activation function, and $n = m + k - 1$ is the dimensionality of the resulting operation. It should be noted that similar to the number of neurons in a fully connected layer, the size and number of kernels are hyperparameters of the S-TCNN model that should be tuned to capture different patterns in the frequency response, such as flat regions, ripples, and resonances. The upsampling ratio of a particular layer can be increased by making use of strided transposed convolutions, where zeros are padded between input points along the convolution axis [28]. This allows achieving higher upsampling ratios with a less number of layers and helps control the depth of the network.

As the output of the base network is to be passed into subsequent layers and not compared with the training data directly, it becomes a latent variable. For reasons that will become clearer in the later sections, we treat this latent variable to be the extrapolated version of the real part of the predicted S-parameter matrix. The overall operation of the base network can then be written as

$$\mathbf{y_{BN}} = f_{\mathrm{BN}}(\mathbf{x}) \tag{12}$$

where $\mathbf{x}$ denotes the input parameters and $\mathbf{y_{BN}}$ is the output with size $N_d \times D_y \times (NM+1)$, with $N_d$ is the number of data in a batch, $D_y$ is the number of output channels, i.e., unique elements in S-parameter matrix, $N$ is the number of frequency points in the training data, and $M$ is the extrapolation factor. It should be noted that in general, we will be working with $P$-port reciprocal systems, and hence, the number of channels will be equal to $D_y = 2(P(P+1)/2)$ unless additional symmetry exists, and the factor of 2 comes from real and imaginary decompositions. The output $\mathbf{y_{BN}}$ is forwarded to the learnable smoothing layer.

### *B. Learnable Smoothing Layer*

Although strided convolution operation allows to increase the upsampling ratio and reduce the number of layers, it can result in uneven overlaps where parts of the output vector are calculated using a larger portion of the input vector than others. This effect is recognized as the checkerboard artifacts [29]. Although the network can ideally learn and adjust its weights to cancel out the checkboard artifacts, similar to [29], we have observed that it is not completely avoided when predicting the frequency responses.

In practice, one can smoothen the frequency response in the inference stage after the network forms its output. However, it is not possible to know which type of filter, and with what settings, should be used for this purpose. In this article, we therefore propose to use an adaptive Gaussian smoothing filter as part of the overall network and assign the standard deviation of it as a learnable parameter of the model for which we learn during the training just as another weight or bias in the base network.

In particular, we use a separate filter for each channel of $\mathbf{y_{BN}}$. This can be written as

$$\mathbf{y_{SL}} = \mathbf{y_{BN}} \circledast \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{-l^2}{2\sigma_i^2}} \tag{13}$$

where

$$\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_{D_y}] \tag{14}$$

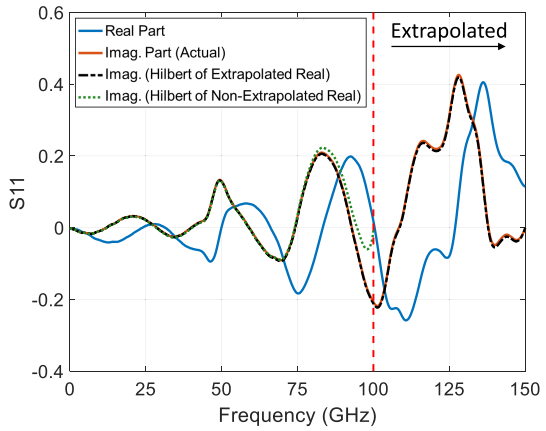


Fig. 3. Illustration of the truncation error and the effect of extrapolation when reconstructing the imaginary part of RL from the real part.

and $l$ denotes the discretization grid to compute the Gaussian kernel and $\sigma_i$ is the learnable standard deviation of the Gaussian kernel per channel. $\circledast$ denotes the channelwise convolution operation, i.e., separate filtering of each frequency response in channels of $\mathbf{y_{BN}}$. In NN terminology, the smoothing layer is equivalent to a channelwise 1-D convolutional layer with no bias and whose weights are fixed to the Gaussian kernel as calculated in (13). It should also be noted here that as the transposed convolution operations used by S-TCNN explicitly exploit the spatial correlation along the frequency axis, it ensures the continuity of the predicted frequency responses. When combined with the learnable smoothing layer, the output of this block becomes a smooth and continuous response.

### C. Causality Enforcement Layer

The Kramers–Kroning relations given in (5) suggest that in order to make that the NN generated S-parameters represent a causal system, and it is sufficient to only predict the real part and construct the imaginary part using the Hilbert transform. However, the S-parameter data used for training are bandlimited and tabulated, leading to truncation and discretization errors that need to be accounted for. In other words, if we construct the imaginary part from the real part in the given frequency range, one can never achieve a zero reconstruction error. This is shown in Fig. 3, where the reconstructed imaginary part of the return loss (RL) element begins to deviate from the actual imaginary part as the frequency approaches the maximum available frequency.

In order to minimize the truncation error and maximize numerical accuracy of the NN, we propose to extrapolate the real part of the S-parameters. Let the broadband S-parameters used for training data be written as

$$\begin{aligned} S_{ij}(f) &= U_{ij}(f) + jV_{ij}(f) \\ f &\in \Omega = [f_{\min}, f_{\max}] \end{aligned} \tag{15}$$

where $U(f)$ and $V(f)$ are the real and imaginary parts, respectively. The Hilbert transform integral in (5) can then be split into in-band and out-of-band as

$$V(f) = \frac{-1}{\pi}\left[\int_{f\in|\Omega|} \frac{U(\widetilde{f})}{f-\widetilde{f}}\,\mathrm{d}\widetilde{f} + \int_{f\notin|\Omega|} \frac{U(\widetilde{f})}{f-\widetilde{f}}\,\mathrm{d}\widetilde{f}\right]. \tag{16}$$

The principal of analytic continuation [30] states that it is possible to find $U(f)$ for $f > f_{\max}$ and $f < f_{\min}$ since such information is contained within the observed part of $V(f)$ in the form of out-of-band integration in (16). As NNs are universal approximators, it is possible to find an extrapolated response such that the in-band reconstruction has minimum error. This is shown in Fig. 3, where the extrapolated response found by the NN minimizes the truncation error of the conventional Hilbert transform.

Here, we exploit these properties and use the base network to extrapolate $U(f)$ until $Mf_{\max}$, where $M$ is the extrapolation factor, and backpropagate through in-band reconstruction error to minimize the effect of truncation error. In order to achieve this, we treat the $(NM+1)$ point output of the base network, $\mathbf{y_{BN}}$ in (12), as the extrapolated real part of the S-parameter frequency response, where the addition of 1 represents the dc point. The input of the CEL is then the smoothened version of this quantity, $\mathbf{y_{SL}}$ in (13).

As the Hilbert transform integral, and its derivatives with respect to the weights of the base network, is to be calculated during the training of the overall model, the direct numerical integration approach is not suitable due to the requirement of specialized integration kernels to handle singularities in the denominator. In addition, the direct integration needs to be evaluated at every discrete frequency point of every channel of $\mathbf{y_{SL}}$ for every training data, which can lead to significant computational overhead during both the training and inference of the NN. Another well-known approach to compute the Hilbert transform is through forming the discrete analytical signal of the given sequence, $\mathbf{y_{SL}}$, through fast Fourier transform (FFT) [31]. In this article, we adopt the FFT-based approach since it is highly computationally efficient as it allows for batched computation and is a differentiable operation. Note that FFT-based computation effectively assumes the frequency domain signal to be periodic. However, the symmetricity of the real part of the frequency response indicates the endpoints of the discrete sequence to be equal to each other. This eliminates the sharp discontinuity between the two consecutive periods of the frequency response that would otherwise cause ripple in the Fourier domain and, thus, validates the periodicity assumption.

In particular, we first create the double-sided spectrum of $\mathbf{y_{SL}}$. As $\mathbf{y_{SL}}$ is of size $N_d \times D_y \times (NM+1)$ where the third axis represents the real part of the frequency spectrum, this can be done by copying positive frequency points to the negative parts as

$$\widetilde{\mathbf{y}}_{\mathbf{SL}}^{(i,j,:)} = \left[\mathbf{y}_{\mathbf{SL}}^{(i,j)}[0,\ldots,NM],\ \mathbf{y}_{\mathbf{SL}}^{(i,j)}[NM-1,\ldots,0]\right] \tag{17}$$

where $\widetilde{\mathbf{y}}_{\mathbf{SL}}$ is the double-sided spectrum of size $N_d \times D_y \times 2(NM+1)-1$ and the superscript $(i,j,:)$ denotes that the operation is done in the third axis for every $(i,j)$ pair. Noting that $\widetilde{\mathbf{y}}_{\mathbf{SL}}$ is always an odd-sized sequence along the frequency axis, we take FFT of $\widetilde{\mathbf{y}}_{\mathbf{SL}}$ along the third axis to transform

from frequency domain into a new $\nu$-domain and create the analytical part as

$$\begin{aligned}\widetilde{\mathbf{z}}_{\mathbf{SL}}^{(i,j,:)} &= -\mathscr{H}\{\widetilde{\mathbf{y}}_{\mathbf{SL}}^{(i,j,:)}\} \\ &= -\operatorname{Im}\{\mathcal{F}^{-1}\{\mathbf{Z}^{(i,j)}[\nu]\}\} \end{aligned} \tag{18}$$

where

$$\mathbf{Z}^{(i,j)}[\nu] = \begin{cases} \widetilde{\mathbf{Y}}^{(i,j)}[0], & \nu = 0 \\ 2\widetilde{\mathbf{Y}}^{(i,j)}[\nu], & 1 \le \nu \le (NM+1) \\ 0, & (NM+1) < \nu \le 2(NM+1)-1 \end{cases} \tag{19}$$

and $\widetilde{\mathbf{Y}}^{(i,j)}[\upsilon] = \mathcal{F}\{\widetilde{\mathbf{y}}_{\mathbf{SL}}^{(i,j,:)}\}$ is the discrete Fourier transform of $\widetilde{\mathbf{y}}_{\mathbf{SL}}^{(i,j)}$ in the $\nu$-domain and the superscript $(i, j, :)$ denotes the operation is done along the third axis. The operation in (19) corresponds to creating the discrete analytic signal of $\widetilde{\mathbf{y}}_{\mathbf{SL}}$, which is one-sided in $\nu$-domain as $\mathbf{Z}^{(i,j)}[\nu] = 0$ for $\nu < 0$. After getting rid of redundant negative frequency points, i.e., $n > (NM + 1)$, the resulting $\widetilde{\mathbf{z}}_{\mathbf{SL}}$ is of size $N_d \times D_y \times (NM + 1)$, which represents the imaginary part of S-parameter response at a frequency step size of $f_s$ for every data point.

In order to address the discretization error, $f_s$ can be arbitrarily decreased at this step of the NN by a factor of $K$ by interpolating both $\mathbf{y}_{\mathbf{SL}}$ and $\mathbf{z}_{\mathbf{SL}}$. However, such an interpolation can disrupt the analytic behavior of the output and result in a noncausal response. Here, to preserve the analytic behavior of the output, the interpolation should preserve the orthogonality of $\mathbf{y}_{\mathbf{SL}}$ and $\mathbf{z}_{\mathbf{SL}}$, written as

$$\sum_{n=0}^{NM} \mathbf{y}_{\mathbf{SL}}^{(i,j)}[n]\mathbf{z}_{\mathbf{SL}}^{(i,j)}[n] = 0. \tag{20}$$

Since the spectrum of $\mathbf{Z}^{(i,j)}[\nu]$ is one-sided, one can arbitrarily pad zeros to $\mathbf{Z}^{(i,j)}[\nu]$ for $\nu < 0$, which would preserve the orthogonality condition in (20) while interpolating $\mathbf{y}_{\mathbf{SL}}$ and $\mathbf{z}_{\mathbf{SL}}$ [31]. In order to reduce the frequency step size to $f_s/K$, the procedure in (18) and (19) can be rewritten as

$$\begin{aligned}\widetilde{\mathbf{y}}_{\mathbf{CEL}}^{(i,j,:)} &= K\operatorname{Re}\{\mathcal{F}^{-1}\{\mathbf{Z}^{(i,j)}[\nu]\}\} \\ \widetilde{\mathbf{z}}_{\mathbf{CEL}}^{(i,j,:)} &= -K\operatorname{Im}\{\mathcal{F}^{-1}\{\mathbf{Z}^{(i,j)}[\nu]\}\} \end{aligned} \tag{21}$$

where

$$\mathbf{Z}^{(i,j)}[\nu] = \begin{cases} \widetilde{\mathbf{Y}}^{(i,j)}[0], & \nu = 0 \\ 2\widetilde{\mathbf{Y}}^{(i,j)}[\nu], & 1 \le \nu \le (NM+1) \\ 0, & (NM+1) < \nu \le (2NMK+K). \end{cases} \tag{22}$$

After the analytic signal is formed, we truncate both $\widetilde{\mathbf{y}}_{\mathbf{CEL}}^{(i,j,:)}$ and $\widetilde{\mathbf{z}}_{\mathbf{CEL}}^{(i,j,:)}$ for $n > (NK + 1)$ to get rid of the extrapolated part since the out-of-band predictions are not of interest in this article.

The overall block diagram of CEL is given in Fig. 4, which can be written as a parameterless NN layer that takes a real tensor of size $N_d \times D_y \times (NM + 1)$ as input and constructs a complex output that is the frequency response of a causal system with a frequency step size of $f_s/K$ as

$$\mathbf{S}_{\mathbf{CEL}}{}^{(i,j)}[n] = \widetilde{\mathbf{y}}_{\mathbf{CEL}}^{(i,j)}[n] + j\ \widetilde{\mathbf{z}}_{\mathbf{CEL}}^{(i,j)}[n] \tag{23}$$

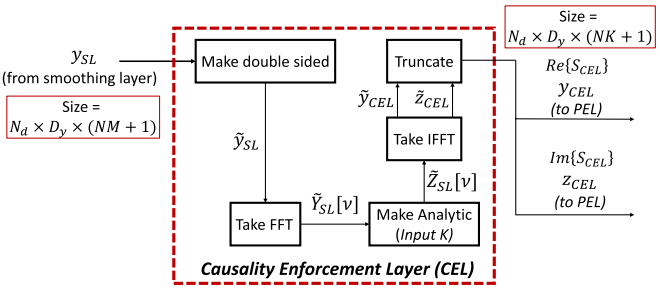


Fig. 4. Block diagram summary of the operations done in CEL.

where $\widetilde{\mathbf{y}}_{\mathbf{CEL}}^{(i,j,:)}$ and $\widetilde{\mathbf{z}}_{\mathbf{CEL}}^{(i,j,:)}$ are calculated as in (21) and (22) and truncated, and $\mathbf{S}_{\mathbf{CEL}}$ is the $N_d \times D_y \times (NK + 1)$ sized complex output tensor of the CEL, which is forwarded to PEL. It should be noted that $K$ and $M$ are hyperparameters of the overall network architecture, just like the number of layers and neurons in a regular NN, and should be chosen based on the application.

### *D. Passivity Enforcement Layer*

As explained in detail in Section II, the NN predicted S-parameters are passive if and only if the maximum singular value condition in (9) is satisfied. In order to check this condition, the flattened complex S-parameter representation, $\mathbf{S}_{\mathbf{CEL}}$, of size $N_d \times D_y \times (NK + 1)$ should be reshaped into the batched matrix form of $N_d \times P \times P \times (NK + 1)$ for a $P$-port network, where the first and last dimensions are the numbers of data and frequency points, respectively.

Since the reshaped $\mathbf{S}_{\mathbf{CEL}}$ matrices are complex-valued and NN training and inference with complex values are not possible, we use isomorphism [32] to transform $\mathbf{S}_{\mathbf{CEL}}$ as

$$\mathbf{S}_{\mathbf{P}} = \begin{bmatrix} \operatorname{Re}\{\mathbf{S}_{\mathbf{CEL}}\} & \operatorname{Im}\{\mathbf{S}_{\mathbf{CEL}}\} \\ -\operatorname{Im}\{\mathbf{S}_{\mathbf{CEL}}\} & \operatorname{Re}\{\mathbf{S}_{\mathbf{CEL}}\} \end{bmatrix} \tag{24}$$

where $\mathbf{S}_{\mathbf{P}}$ is of size $N_d \times 2P \times 2P \times (NK + 1)$. This representation allows using regular NN computations and other linear algebraic operations for complex-valued matrices. In order to check and enforce passivity to $\mathbf{S}_{\mathbf{P}}$, one needs to determine its maximum singular value, $\sigma_1$, at every data and frequency point. However, most efficient SVD algorithms contain sequential bidiagonalization operations and cannot be efficiently parallelized. This would add a significant computational overhead to the training of the NN since, in order to characterize passivity of every $2P \times 2P$ matrix in $\mathbf{S}_{\mathbf{P}}$, one needs to perform $N_d(NK + 1)$ sequential SVD and gradient operations at every iteration of the training process.

To limit the computational overhead, we propose to use an upper bound to $\sigma_1$ that can be calculated using only matrix–matrix multiplications and Hadamard products, hence, can be massively parallelized [33]. Let

$$\begin{aligned} C(f) &= \operatorname{tr}(S^H(f)S(f)) \\ D(f) &= \operatorname{tr}((S^H(f)S(f))^2) \end{aligned} \tag{25}$$

where $\operatorname{tr}(\cdot)$ is the trace operator. An upper bound to maximum singular value of a $P \times P$ S-parameter matrix can then be

calculated as

$$\sigma_1(f) \leq \hat{\sigma}_1(f)$$

where

$$\hat{\sigma}_1(f) = \sqrt{\frac{C(f)}{P} + \left(\frac{P-1}{P}\left(D(f) - \frac{C(f)^2}{P}\right)\right)^{0.5}}. \quad (26)$$

To minimize the number of matrix–matrix multiplications that has greater than quadratic time complexity, $C(f)$ and $D(f)$ can be calculated efficiently using Hadamard products as

$$C(f) = \sum_{i=1}^{P} |S_{ii}(f)|^2 \quad (27)$$

$$D(f) = \sum_{i,j=1}^{P} [(S^H(f)S(f)) \circ (S(f)S^H(f))]_{ij} \quad (28)$$

where $\circ$ is the Hadamard product. The proof of the upper bound follows from the bounds for eigenvalues derived using a Cauchy–Schwarz-type inequality [34] and the fact that $\sigma_1(f)$ is the square root of the largest eigenvalue of $S^H(f)S(f)$. Experiments and further details regarding the tightness of this bound for various types of matrices can be found in [33]. For typical S-parameter matrices of interest in this article, the average relative deviation between the maximum singular value and its upper bound is found to be approximately 3%. We further emphasize here that (27) and (28), thereby (26), are simple multiplication and accumulation operations, hence, can be easily parallelized both during the training and inference.

Once $\hat{\sigma}_1(f)$ is obtained, there are several ways to enforce passivity to every matrix in $\mathbf{S_P}$. One way is to find $\hat{\sigma}_{\max} = \max(\hat{\sigma}_1(f))$ for every data point and divide the whole frequency spectrum of each element in $\mathbf{S_{CEL}}$ by $\hat{\sigma}_{\max}$ if it is greater than 1 [22]. Note that this enforcement would increase the numerical error between the predicted and actual S-parameter responses. As this enforcement is to be done during the training of the network, weights and biases of the base network are automatically adjusted to minimize these deviations. However, we have observed that, in practice, this results in overreducing $\sigma_1(f)$ near dc point, where losses are minimal.

Another approach is to perform frequency-dependent enforcement, i.e., divide $\mathbf{S_P}(f)$ by $\sigma_1(f)$ if passivity is violated. This is shown to be the optimal enforcement in the frequency domain in the sense that $\mathbf{S_P}(f)$ is minimally perturbed at every frequency point [35]. However, frequency-dependent enforcement results in disrupting Kramers–Kronig relations, and hence, the resulting S-parameters become noncausal.

In this article, we propose a new causality-preserving passivity enforcement technique that can be utilized in an NN environment. We view the frequency-dependent passivity enforcement operation as filtering in the frequency domain, which is written as

$$\mathbf{S_{PEL}}(f) = \mathbf{S_{CEL}}(f) \odot \mathbf{\Sigma}(f) \quad (29)$$

where $\mathbf{\Sigma}(f)$ is the complex-valued passivity enforcement filter of size $N_d \times (NM+1)$ and $\odot$ operator is used to indicate that the filtering is done along the frequency axis of each channel of $\mathbf{S_{CEL}}$ at every data point. In order to inflict minimal changes to $\mathbf{S_{CEL}}(f)$, the desired magnitude spectrum of $\mathbf{\Sigma}(f)$ at each data point can be written as

$$|\Sigma(f)| = \begin{cases} \dfrac{1}{\hat{\sigma}_1(f)}, & \text{for } \hat{\sigma}_1(f) > 1 \\ 1, & \text{for } \hat{\sigma}_1(f) \leq 1. \end{cases} \quad (30)$$

It has been shown that if $\Sigma(f)$ is a minimum-phase filter and $\mathbf{S_{CEL}}$ is the frequency response of a causal function, $\mathbf{S_{PEL}}$ also represents the frequency response of a causal system [23].

A minimum-phase frequency response can be formed solely from its magnitude spectrum [36]. Here, we propose to exploit these to construct a minimum-phase passivity enforcement filter from the desired magnitude spectrum in (30) as

$$\Sigma(f) = |\Sigma(f)|e^{j\theta(f)}$$
$$\theta(f) = \mathscr{H}\{\log|\Sigma(f)|\} \quad (31)$$

where $\log(\cdot)$ is the natural logarithm operator and the Hilbert transform is taken using the FFT-based approach outlined in (17)–(19). An example of a complex-valued passivity enforcement filter constructed using the proposed approach is also given in Fig. 5.

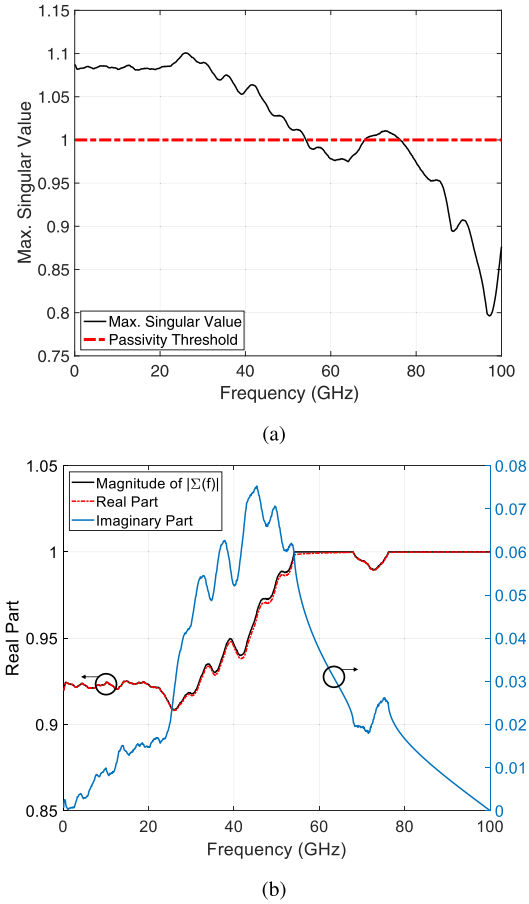


Fig. 5. Illustration of constructing the proposed minimum-phase passivity enforcement filter. (a) Singular value to be filtered. (b) Magnitude and real and imaginary parts of the filter.

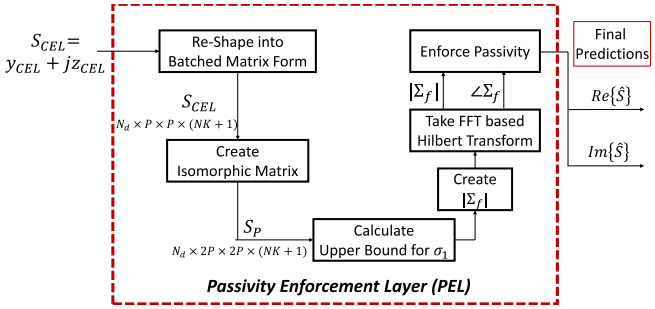


Fig. 6. Block diagram summary of the operations done in PEL.

Overall operations that PEL performs are summarized in Fig. 6. Similar to CEL, PEL can be written as a parameterless NN layer that takes $N_d \times D_y \times (NK+1)$ complex tensor as input and constructs a complex tensor that is the final predicted response as

$$\begin{aligned}\hat{\mathbf{S}}[n] &= f(\mathbf{S_{CEL}}[n]) \\ &= \mathbf{S_{CEL}}[n] \odot \mathbf{\Sigma}[n], \quad n = 0, \ldots, NK \end{aligned} \tag{32}$$

where $\hat{S}[n]$ is the $N_d \times D_y \times (NK+1)$ sized complex output tensor that represents the predicted broadband S-parameters of a causal and passive system.

### *E. Training Methodology*

The training of the proposed network architecture can be performed using the conventional backpropagation method since all the operations are differentiable. Although CEL and PEL are both comprised of computationally efficient and parallelizable operations, batched matrix calculations in (26)–(28) can create some computational overhead during the training.

In order to further minimize this overhead, we propose using a two-step training methodology. As the S-parameters used for training data are passive at all frequencies, minimization of the training error corresponds to making the singular values of the predicted S-parameters close to their actual values during the training process. Exploiting this, for the first $L_f$ iterations, i.e., gradient updates, of the total $L$ iterations allocated for the training process, we bypass PEL and directly compare $\mathbf{S_{CEL}}$ with the training data to calculate the error. We then activate PEL to guarantee passivity, causing a slight jump in the training error due to nonpassivities, which is then minimized for the remaining $(L - L_f)$ iterations. For the application examples given in Sections IV–VI, we have found that this results in almost identical training and test error, but with a lower computational time used for the training.

As for the training error metric, instead of the conventional mean-squared error (MSE) loss, we propose using the modified loss function that is more suitable for predicting frequency responses [19], which can be written as

$$L = \sqrt{\frac{1}{N_d D_y} \sum_{n=1}^{N_d} \sum_{d=1}^{D_y} L_f^{(n,d)}} \tag{33}$$

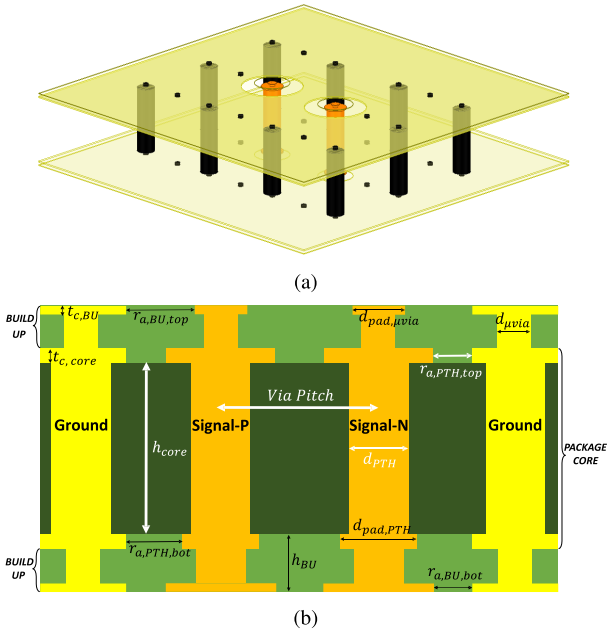


Fig. 7. Geometry of the differential PTH structure in package core. (a) Isometric view. (b) Cross section [22].

where

$$L_f^{(n,d)} = \sqrt{\frac{1}{N} \sum_{m=0}^{N} (S^{(n,d)}[m] - \hat{S}^{(n,d)}[mK])^2}$$

and $S^{(n,d)}[m]$ is the frequency response for the $n$th data point of the $d$th channel of the training S-parameters at the $m$th frequency point, and the factor of $K$ in the predicted response, $\hat{S}^{(n,d)}[mK]$, comes from (21) and (22) where frequency step size of the predictions is reduced by a factor of $K$. The loss function in (33) is based on the scaled $\ell^2-$norm of the error between the predicted and the actual frequency responses, averaged over $N_d \times D_y$ different frequency responses in the training set. Compared with the conventional MSE loss, this corresponds to response-to-response error calculation rather than a comparison at individual frequency points. We refer readers to [19] for a more detailed explanation of the loss function and its comparison to MSE loss.

## IV. Application I: Differential PTH Pair in Package Core

The first application chosen to demonstrate the proposed method is modeling a differential PTH pair in package core along with the microvias that connect immediate build-up (BU) layers to the package core. Such structures are common in off-chip high-speed channels and often times can be the bottleneck that limits the total channel bandwidth. Hence, it is important to optimize its design to minimize distortions in the signal. The objective here then is to learn a causal and passive mapping from the geometrical parameters of the PTH structure to four-port single-ended broadband S-parameters such that it can be later used in time-domain simulations to capture its effect on high-speed signaling. The input

TABLE I
CONTROL PARAMETERS OF THE PTH STRUCTURE

| Parameter | | Unit | Min | Max |
|---|---|---|---|---|
| $\mu$-via Diameter | $d_{\mu\text{-via}}$ | $\mu$m | 30 | 70 |
| $\mu$-via Pad Diameter | $d_{\text{pad, }\mu\text{-via}}$ | $\mu$m | 31 | 140 |
| BU Layer Thickness | $h_{BU}$ | $\mu$m | 20 | 35 |
| $\mu$-via Top Antipad Radius | $r_{a,\text{ BU, TOP}}$ | $\mu$m | 100 | 500 |
| $\mu$-via Bot. Antipad Radius | $r_{a,\text{ BU, BOT}}$ | $\mu$m | 100 | 500 |
| PTH Pitch | $v_p$ | $\mu$m | 300 | 1200 |
| Core Thickness | $h_{Core}$ | $\mu$m | 100 | 1200 |
| BU Copper Thickness | $t_{c,\text{ BU}}$ | $\mu$m | 10 | 20 |
| Core Copper Thickness | $t_{c,\text{ CORE}}$ | $\mu$m | 11 | 40 |
| PTH Diameter | $d_{PTH}$ | $\mu$m | 100 | 250 |
| PTH Pad Diameter | $d_{\text{pad, PTH}}$ | $\mu$m | 110 | 500 |
| PTH Top Antipad Radius | $r_{a,\text{ PTH, TOP}}$ | $\mu$m | 50 | 500 |
| PTH Bot. Antipad Radius | $r_{a,\text{ PTH, BOT}}$ | $\mu$m | 50 | 500 |

parameters comprise a 13-D space and their corresponding bounds are given in Fig. 7 and Table I. Note that the large sample space is chosen to contain various technology nodes to avoid creating different models for different manufacturing technologies.

### A. Simulation and Model Setup

In order to create the predictive model, 680 samples based on Latin hypercube sampling (LHS) are determined. These are then fed into a full-wave EM solver to generate their corresponding S-parameters between 0.1 and 100 GHz at 100-MHz frequency steps, where the structure is excited using coaxial waveports at the antipads of microvias. After the data are collected, 550 out of 680 samples are used for the training of the model. As the PTH structure is a partially symmetric and reciprocal system, the target data to be trained, i.e., output channels of the model, is determined to be the real and imaginary parts of the frequency responses of $S_{11}$, $S_{12}$, $S_{13}$, $S_{14}$, $S_{33}$, and $S_{34}$, corresponding to a total of 12 000 output dimensions. We use $K = 2$ in (21) and $M = 1.5$ in (19) to reduce the frequency step size of the predicted response to 50 MHz. The model then predicts the S-parameters from dc up to 100 GHz at 50-MHz steps as the extrapolated part is truncated.

### B. Results

We compare the proposed methodology (S-TCNN + CEL + PEL) with S-TCNN that also uses the learnable smoothing layer and a deep fully connected NN (DNN) that considers frequency as an additional input parameter to the model. The metric to assess the numerical accuracy for both models is chosen as the normalized mean squared error (NMSE) over each frequency response in the test set, which is given as

$$\text{NMSE} = \frac{1}{N_d D_y} \times \sum_{d=1}^{D_y} \sum_{n=1}^{N_d} \times \left( \frac{\sum_{m=1}^{N} (S_{n,d}[m] - \hat{S}_{n,d}[mK])^2}{\sum_{m=1}^{N} \left( S_{n,d}[m] - \frac{1}{N} \sum_{m=1}^{N} S_{n,d}[m] \right)^2} \right) \quad (34)$$

TABLE II
COMPARISON OF MODELS ON TEST DATA FOR PTH MODEL

| | DNN | S-TCNN | S-TCNN + CEL + PEL |
|---|---|---|---|
| **NMSE** | 9.42% ± 0.33% | 5.08% ± 0.31% | 5.34% ± 0.09% |
| **Av. Causality Metric** | 7.49% | 11.17% | 100.0% |
| **Range of $\sigma_1$** | [0.615, 1.285] | [0.653, 1.131] | [0.607, 0.999] |
| **Inference Time** (for 1K broadband S-Params.) | 0.26 s | 0.13 s | 0.91 s |

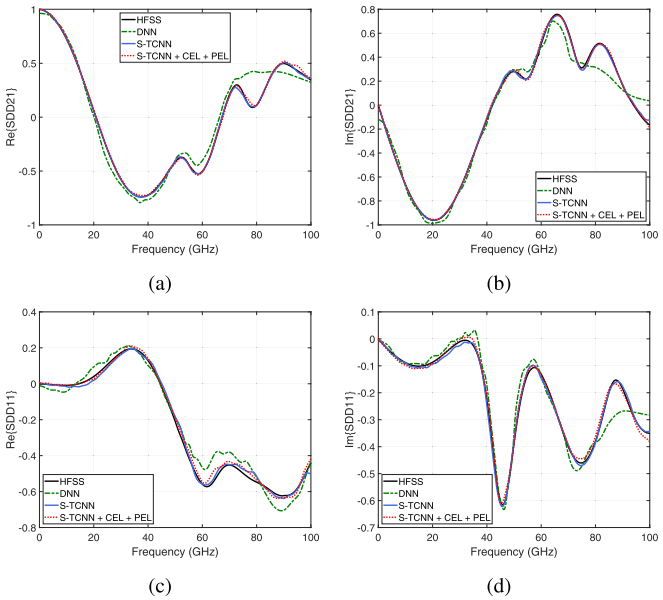


Fig. 8. Comparison of predicted S-parameters with 3-D EM simulations for different test cases of PTH model. (a) and (b) Real and imaginary parts of differential IL. (c) and (d) Real and imaginary parts of differential RL.

where $N_d = 130$ is the number of validation data and $D_y = 12$ represents the real and imaginary parts of the learned S-parameters.

Table II summarizes the NMSE values for each model along with the causality metrics and passivity violations of the predicted S-parameters that are calculated using a commercial tool. In order to assess the predictive accuracy, we trained each model ten times using randomly initialized weights and report the mean and standard deviation of the NMSE values in Table II. The DNN model performed significantly worse compared with the other models and had an average NMSE of 9.42%, whereas S-TCNN and S-TCNN + CEL + PEL models performed similarly with an average NMSE of 5.08% and 5.34%, respectively. Predicted real and imaginary parts of differential insertion loss (IL) and RL terms for different test cases are also shown in Fig. 8.

In terms of physical consistency, S-parameters predicted by S-TCNN and DNN had average causality quality metrics of 11.17% and 7.49%, respectively, and none of the S-parameters predicted by either model were causal, whereas all the S-parameters predicted by the proposed model were found to be causal with a quality metric of 100.0%. We further characterize the maximum singular values of S-parameters

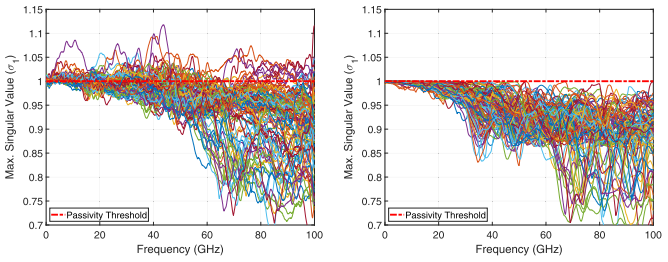


Fig. 9. Passivity characterization of the predicted S-parameters for every case in the test set. (a) S-TCNN. (b) S-TCNN + CEL + PEL.

predicted by each model for every case in the test set. We observe that S-TCNN + CEL + PEL guarantees passivity of the predicted S-parameters as $\sigma_1$ of all predictions are less than 0.99, while the predictions done by using S-TCNN and DNN do not necessarily result in a passive response as most of the predicted responses have $\sigma_{\max} > 1$. The maximum singular values of the predicted S-matrices for S-TCNN and S-TCNN + CEL + PEL model are further given in Fig. 9 to show the effect of PEL on the passivity.

In terms of inference run times, it took 0.13 s for the S-TCNN model to generate 1000 frequency responses compared with 0.91 s for S-TCNN + CEL + PEL and 0.26 s for DNN, showing that the operations done in CEL and PEL have minimal computational overhead to the overall model. Note that for a fair comparison, we have tuned the hyperparameters of all the models to achieve the highest prediction accuracy. The best-performing DNN model had 132 632 learnable parameters and consisted of three hidden layers with 250 neurons in each layer (14-250-250-250-12). The S-TCNN model had 43 048 learnable parameters and consisted of two fully connected layers with 30 neurons (13-30-30), followed by five 1-D transposed convolutional layers, each having 30 channels. The kernel size and stride for these layers were 32, 4, 4, 4, and 2 and 1, 2, 2, 4, and 2, respectively. The base network of the S-TCNN + CEL + PEL model had the same S-TCNN architecture except for a stride of 3 that was used for the last layer. We have implemented each model using PyTorch [37], and for all the models, we used exponential linear units (ELUs) [38] as the activation function. The training was performed using the Adam optimizer [39] with an initial learning rate (LR) of 0.01 while reducing LR by $\times 0.5$ at every 500 iterations for a total of 3000 gradient updates. We have used the same model settings for the applications in the following sections.

To show the significance of physical consistency of the predicted S-parameters, we use the predictions of the two best-performing models (S-TCNN and S-TCNN + CEL + PEL) in a subsequent time-domain characterization and compare with the data obtained from 3D EM simulations. We perform a differential time-domain reflectometry (TDR) and transmission (TDT) using a 15-ps 0%–100% rise time cosine-edge step and pulse as differential excitation, respectively. These are common time-domain characterization techniques for high-speed channel analysis and their quality is of utmost importance to make design choices. The results of these characterizations are given in Fig. 10. The TDR simulation in Fig. 10(a) clearly shows that S-parameters predicted by the S-TCNN are noncausal and significantly distort the characterization since a substantial impedance change can be observed before the structure is excited at $t = 0$, which is not present for S-parameters predicted by the proposed model. A similar behavior is also observed for the TDT simulation in Fig. 10(b), where a substantial input power is leaked before the excitation, which can cause nonrealistic intersymbol interference (ISI) when a long bit pattern is simulated. The results clearly show that although predicted S-parameters have the same level of numerical accuracy with respect to the 3-D EM simulation, pure NN predicted S-parameters are not physically consistent and cannot be used in the subsequent simulations.

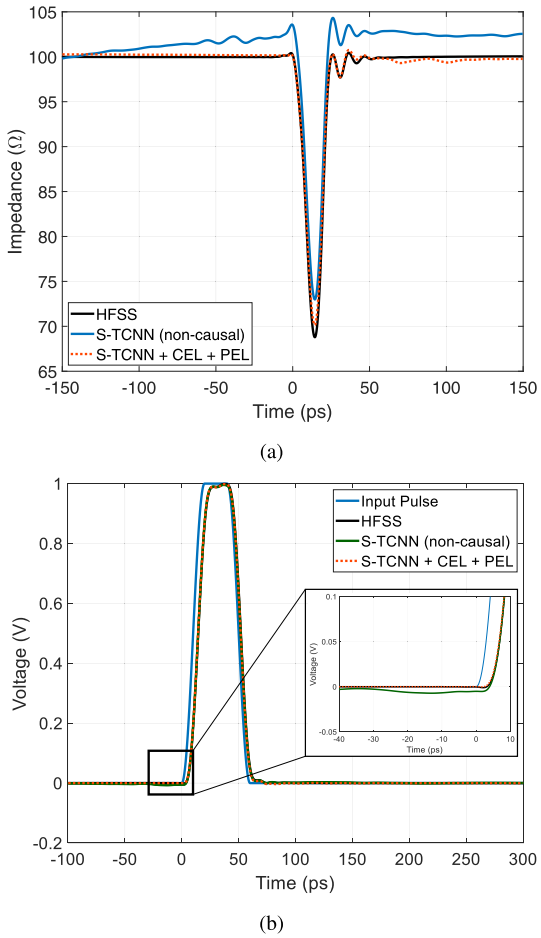


Fig. 10. Time-domain characterization of predicted S-parameters for excitations with 15-ps 0%–100% cosine-edge rise time. (a) TDR. (b) TDT.

## V. Application II: Differential Stripline Pair in Package

The second application we choose in this article is parameterizing the frequency response of a differential stripline structure in package. Such transmission lines are commonly a

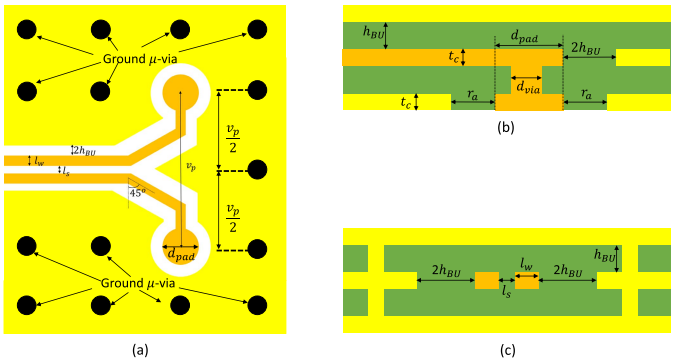


Fig. 11. Geometry of the differential stripline pair in package. (a) Top view. (b) Front view. (c) Side view.

TABLE III
CONTROL PARAMETERS OF THE STRIPLINE STRUCTURE

| Parameter | | Unit | Min | Max |
|---|---|---|---|---|
| Line Width | $l_w$ | $\mu$m | 15 | 75 |
| Pair Spacing | $l_s$ | $\mu$m | 30 | 60 |
| $\mu$-via Diameter | $d_{via}$ | $\mu$m | 30 | 70 |
| $\mu$-via Pad Diameter | $d_{pad}$ | $\mu$m | 31 | 140 |
| $\mu$-via Antipad Radius | $r_a$ | $\mu$m | 50 | 500 |
| Via Pitch | $v_p$ | $\mu$m | 300 | 1200 |
| Copper Thickness | $t_c$ | $\mu$m | 10 | 20 |
| BU Layer Thickness | $h_{BU}$ | $\mu$m | 20 | 35 |

TABLE IV
COMPARISON OF MODELS ON TEST DATA FOR STRIPLINE MODEL

| | DNN | S-TCNN | S-TCNN + CEL + PEL |
|---|---|---|---|
| **NMSE** | 4.73% ± 0.22% | 3.08% ± 0.29% | 2.47% ± 0.43% |
| **Av. Causality Metric** | 8.24% | 13.86% | 100.0% |
| **Range of $\sigma_1$** | [0.818, 1.008] | [0.788, 1.004] | [0.812, 1.000] |
| **Inference Time** (for 1K broadband S-Params.) | 0.11 s | 0.09 s | 0.61 s |

part of the chip-to-board escape route in high-speed channels, and their impedance can greatly affect the achievable communication bandwidth. Since their return path and transition to microvias are nonuniform, they need to be characterized using computationally expensive 3-D EM simulations. In this section, we therefore derive a parametric model to predict four-port single-ended S-parameters such that it can be later used for detailed parametric analysis. The geometry of the differential stripline pair is given in Fig. 11 and is parameterized using eight parameters as in Table III.

### A. Simulation Setup

Similar to the PTH application in Section IV, we determine 940 samples based on LHS and simulate broadband S-parameters for each sample between 0.1 and 100 GHz at 100-MHz steps, where the ports are defined at front of the stripline and antipads of the microvias; 750 of 940 samples are used for training, and the rest is used for validation. The output parameters, output dimensionality, and settings of the model are taken as the same as described in Section IV-A.

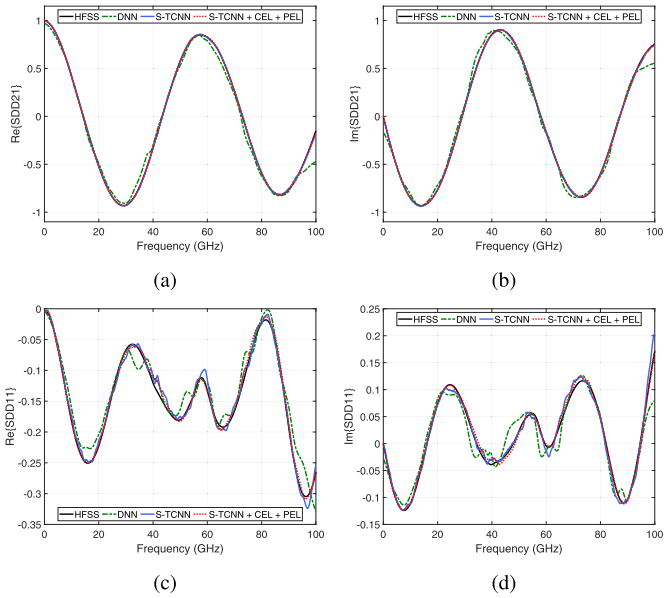


Fig. 12. Comparison of predicted S-parameters for different test cases of stripline model. (a) and (b) Real and imaginary parts of differential IL. (c) and (d) Real and imaginary parts of differential RL.

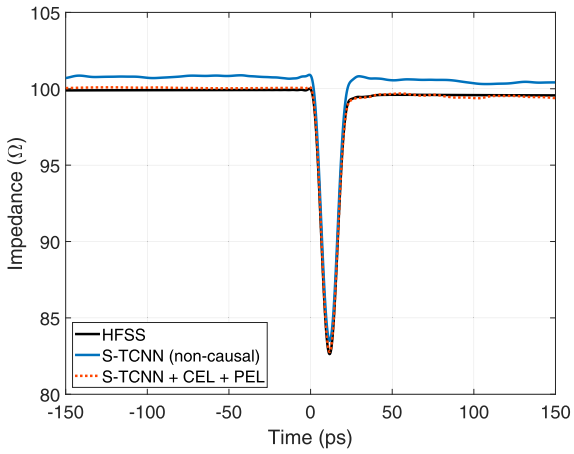


Fig. 13. Comparison of differential TDR of predicted S-parameters for the stripline model for a cosine-edge step with 15-ps 0%–100% rise time.

### B. Results

The results are summarized in Table IV. The DNN model had an average NMSE of 4.73% compared with 3.08% and 2.47% for S-TCNN and S-TCNN + CEL + PEL, respectively. Fig. 12 further compares the predicted differential IL and RLs to 3-D EM simulations. It can be seen that the convolutional-type models captured both smooth and resonant parts of the frequency response, whereas the DNN model had lower accuracy around the resonances. Predicted S-parameters using the S-TCNN and DNN model had an average causality quality metric of 13.86% and 8.24%, respectively, compared to 100.0% with the proposed model. For the best-performing two models, the effect of noncausality is further demonstrated in the differential TDR plot for a test case in Fig. 13.

As the transmission line structure has a linearly increasing loss trend, the maximum singular value of its S-parameters

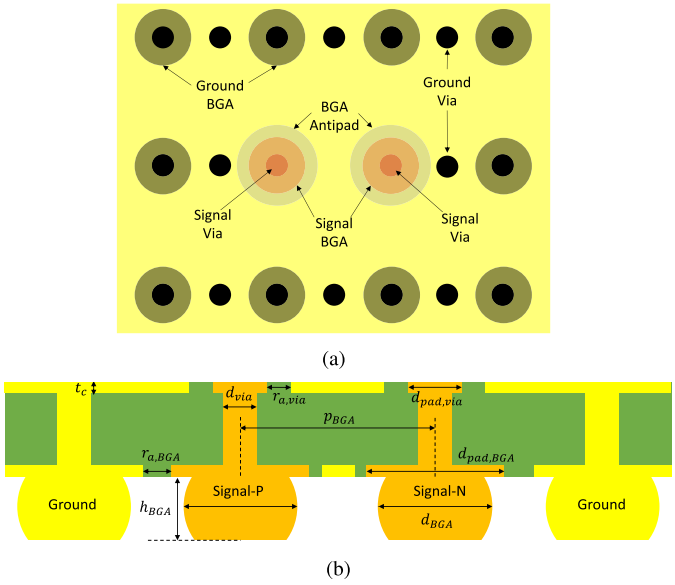


Fig. 14. Geometry of the differential BGA structure in package-to-board transition. (a) Top view. (b) Cross section.

linearly decreases as well. The high predictive quality of the models corresponds to capturing this linear trend very accurately. This results in almost passive S-parameter predictions for S-TCNN and DNN models as the range of $\sigma_1$ is bounded by 1.004 and 1.008, respectively. However, the passivity is not guaranteed for further predictions that are not in training and test sets, whereas the proposed model guarantees this as shown by the range of $\sigma_1$ being bounded by 1.000 for the test set. The inference time for each model to generate 1000 broadband S-parameter shows the minimal computational overhead property of the proposed layers.

## VI. Application III: Differential BGA Pair and Cascade Analysis

The third application to demonstrate the effectiveness of the proposed model is a differential microvia to BGA transition for package-to-board connections. These structures are the last step of the chip-to-board interconnects. Due to the relatively larger antipad diameters, such transitions can be highly capacitive and disrupt the signaling quality when operating at higher frequencies. As such, they need to be carefully designed to ensure that the desired bandwidth can be achieved. We therefore derive the parameterized model of the BGA structure given in Fig. 14, where its geometry is parameterized using nine variables within the bounds given in Table V. The output parameters are the same as described in the PTH section, which represents 12 000 dimensions.

Once the BGA model is verified, we then use it in cascade with the models for stripline and PTH to analyze the full vertical package-to-board transition in time domain with the goal of demonstrating the significance of physical consistency of each block used in the cascade analysis.

TABLE V
Control Parameters of the BGA Structure

| Parameter | | Unit | Min | Max |
|---|---|---|---|---|
| Via Diameter | $d_{via}$ | $\mu$m | 30 | 70 |
| Via Pad Diameter | $d_{via,pad}$ | $\mu$m | 31 | 140 |
| Via Antipad Radius | $r_{a,\ via}$ | $\mu$m | 50 | 500 |
| BU Layer Thickness | $h_{BU}$ | $\mu$m | 20 | 35 |
| Copper Thickness | $t_c$ | $\mu$m | 10 | 20 |
| BGA Pitch | $p_{BGA}$ | $\mu$m | 300 | 1200 |
| BGA Diameter | $d_{BGA}$ | $\mu$m | 180 | 960 |
| BGA Height | $h_{BGA}$ | $\mu$m | 70 | 770 |
| BGA Antipad Radius | $r_{a,BGA}$ | $\mu$m | 100 | 1050 |

TABLE VI
Comparison of Models on Test Data for BGA Model

| | DNN | S-TCNN | S-TCNN + CEL + PEL |
|---|---|---|---|
| **NMSE** | 4.74% ± 0.25% | 2.17% ± 0.35% | 2.46% ± 0.22 % |
| **Av. Causality Metric** | 9.83% | 13.71% | 100.0% |
| **Range of $\sigma_1$** | [0.963, 1.277] | [0.962, 1.109] | [0.8225, 1.000] |
| **Inference Time** (for 1K broadband S-Params.) | 0.18 s | 0.01 s | 0.57 s |

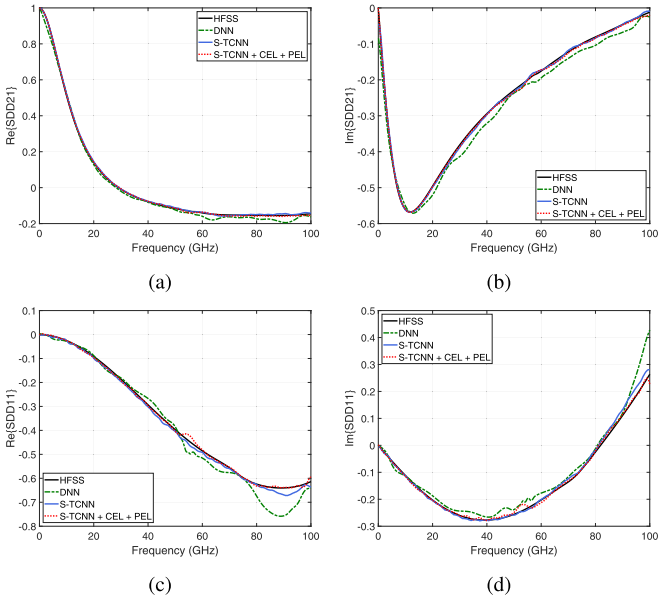


Fig. 15. Comparison of predicted S-parameters with 3-D EM simulations for different test cases of BGA model. (a) and (b) Real and imaginary parts of differential IL. (c, d) Real and imaginary parts of differential RL.

### A. Simulation Setup

For the BGA model, a total of 830 samples are collected and characterized between 0.1 and 100 GHz at 100-MHz steps using 3-D EM solver; 700 of such simulations are used to train the predictive model and remaining 130 samples are used for validation. The ports are defined at antipads of microvias and bottom of the BGA, i.e., at the circular cross section, where a perfect electric conductor (PEC) is used as the reference for the BGA ports.

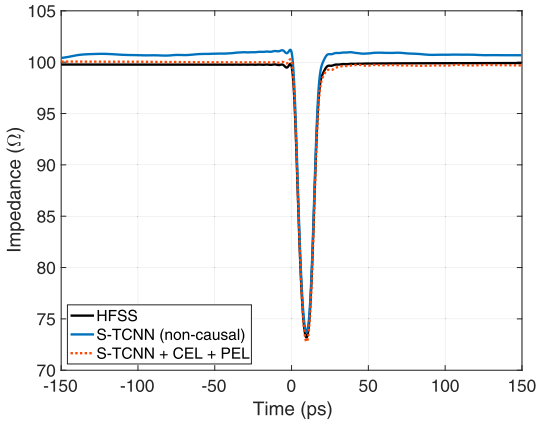


Fig. 16. Comparison of differential TDR of predicted S-parameters for the BGA model for a cosine-edge step with 15-ps 0%–100% rise time.

### B. Results

The comparison results are given in Table VI. Similar to the previous application examples, the DNN model had the worst predictive accuracy with an average NMSE of 4.74%, where convolutional-type models performed significantly better with an average NMSE of 2.17% and 2.46% for S-TCNN and S-TCNN + CEL + PEL, respectively. A further comparison of actual and predicted S-parameters is also given in Fig. 15. The computational overhead to inference time required to predict 1000 frequency responses is also observed to be minimal for the BGA model.

S-TCNN and DNN predicted S-parameters had an average causality quality metric of 13.71% and 9.83%, and the maximum $\sigma_1$ was calculated to be 1.109 and 1.277, respectively, which shows significant causality and passivity violations. All the predicted S-parameters for the proposed model were characterized as completely causal and passive. The effect of physical consistency is further shown in Fig. 16 via a differential TDR simulation.

### C. Cascade Analysis

After the models for stripline, PTH and BGA are derived and their accuracies are verified, and they can be cascaded together to obtain S-parameter predictions for the full vertical package-to-board transition as in Fig. 17. This allows for analyzing the TDR of the complete vertical transition that is required to perform a DSE and/or UQ to determine the most feasible design parameters to maintain a certain impedance along the transition. This TDR analysis given in Fig. 18 shows that as the proposed technique produces causal S-parameter blocks, the resulting TDR from the cascade analysis is causal as well and has very good agreement with the full-wave simulation. The TDR obtained through S-parameters predicted by the S-TCNN model without CEL and PEL, on the other hand, shows noticeable causality violations.

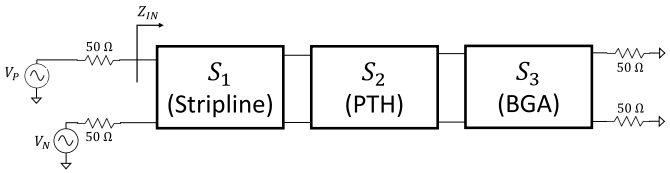


Fig. 17. Schematic for analyzing the impedance of package-to-board transition by cascading predicted S-parameters.

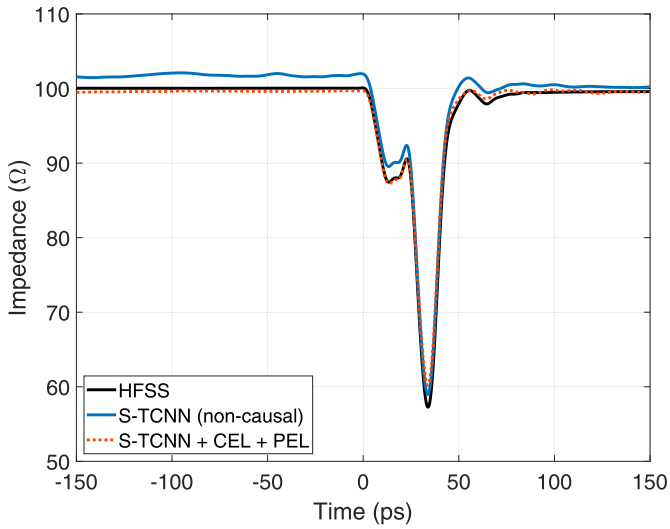


Fig. 18. TDR comparison for cascaded analysis for a cosine-edge step input with 15-ps 0%–100% rise time.

## VII. Conclusion

In this article, we have shown that NN models that are used to predict S-parameters do not guarantee the physical consistency of predictions. Focusing purely on reducing numerical accuracy when creating the NN is therefore not suitable for predicting frequency responses. To this end, we have proposed CEL and PEL, which ensures that the predictions represent the frequency response of a causal and passive system.

In CEL, we have shown that truncation and discretization errors limit the direct use of Kramers–Kronig relations, and their effect can be minimized by NN-based extrapolation and causality-preserving interpolation through exploiting properties of analytic signals. In PEL, we have proposed a new minimum-phase passivity enforcement filter to perform frequency-dependent passivity correction while not disrupting causality properties. Furthermore, we have used an easily parallelized and computationally efficient upper bound to maximum singular value for the complex S-parameter matrix to avoid costly SVD operations to minimize computational overhead.

We have demonstrated the effectiveness of our proposed model for three different applications, where the goal was to parameterize a four-port single-ended S-parameter matrix from dc to 100 GHz, which corresponds to 12 000 output dimensions. In all of the applications, the predictive accuracy of S-TCNN models with and without CEL and PEL was highly

accurate when compared with the 3-D EM simulations and were very similar to each other. However, in the absence of PEL and CEL, the predicted S-parameters showed significant causality and passivity violations with causality quality metrics in the range of 11.17%–13.86% and maximum singular values in the range of 1.004–1.131 for the three applications. On the other hand, the proposed model has resulted in 100.0% causal and passive S-parameters for all the test cases of all three application examples. The effect of physical consistency for each application is further demonstrated by using predicted S-parameters in the subsequent TDT and TDR simulations.

## REFERENCES


[1] N. Knudde *et al.*, "Data-efficient Bayesian optimization with constraints for power amplifier design," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Aug. 2018, pp. 1–3.

[2] 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.

[3] H. M. Torun, M. Swaminathan, A. K. Davis, and M. L. F. Bellaredj, "A global Bayesian optimization algorithm and its application to integrated system design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 4, pp. 792–802, Apr. 2018.

[4] S. Koziel, A. A. A. T. Siguresson, and F. V. Vidarsson, "Accurate design-oriented modeling of compact microwave couplers in constrained domains," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Aug. 2018, pp. 1–3.

[5] J. E. Rayas-Sanchez 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.

[6] R. Trinchero, M. Larbi, H. M. Torun, F. G. Canavero, and M. Swaminathan, "Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters," *IEEE Access*, vol. 7, pp. 4056–4066, 2019.

[7] H. M. Torun, J. A. Hejase, J. Tang, W. D. Beckert, and M. Swaminathan, "Bayesian active learning for uncertainty quantification of high speed channel signaling," in *Proc. IEEE 27th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2018, pp. 311–313.

[8] Y. Ye, D. Spina, G. Antonini, and T. Dhaene, "Parameterized macromodeling of stochastic linear systems for frequency- and time-domain variability analysis," in *Proc. IEEE 22nd Workshop Signal Power Integrity (SPI)*, May 2018, pp. 1–4.

[9] Q.-J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 4, pp. 1339–1350, Apr. 2003.

[10] H. Kabir, L. Zhang, M. Yu, P. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Mag.*, vol. 11, no. 3, pp. 105–118, May 2010.

[11] J. Jin, F. Feng, J. Zhang, S. Yan, W. Na, and Q. Zhang, "A novel deep neural network topology for parametric modeling of passive microwave components," *IEEE Access*, vol. 8, pp. 82273–82285, 2020.

[12] Y. Cao, G. Wang, and Q.-J. Zhang, "A new training approach for parametric modeling of microwave passive components using combined neural networks and transfer functions," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 11, pp. 2727–2742, Nov. 2009.

[13] H. M. Torun, M. Larbi, and M. Swaminathan, "A Bayesian framework for optimizing interconnects in high-speed channels," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Aug. 2018, pp. 1–4.

[14] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microw. Wireless Compon. Lett.*, vol. 18, no. 6, pp. 383–385, Jun. 2008.

[15] F. Feng, V.-M.-R. Gongal-Reddy, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of microwave components using adjoint neural networks and pole-residue transfer functions with EM sensitivity analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 1955–1975, Jun. 2017.

[16] Z. Zhao, F. Feng, W. Zhang, J. Zhang, J. Jin, and Q.-J. Zhang, "Parametric modeling of EM behavior of microwave components using combined neural networks and hybrid-based transfer functions," *IEEE Access*, vol. 8, pp. 93922–93938, 2020.

[17] F. Feng, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of EM behavior of microwave components using combined neural networks and pole-residue-based transfer functions," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60–77, Jan. 2016.

[18] B. Gustavsen, "Improving the pole relocating properties of vector fitting," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1587–1592, Jul. 2006.

[19] H. M. Torun *et al.*, "A spectral convolutional net for co-optimization of integrated voltage regulators and embedded inductors," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2019, pp. 1–8.

[20] F. Wang and Q.-J. Zhang, "Knowledge-based neural models for microwave design," *IEEE Trans. Microw. Theory Techn.*, vol. 45, no. 12, pp. 2333–2343, Dec. 1997.

[21] W. Na, F. Feng, C. Zhang, and Q.-J. Zhang, "A unified automated parametric modeling algorithm using knowledge-based neural network and $l_1$ optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 3, pp. 729–745, Mar. 2017.

[22] H. M. Torun, A. C. Durgun, K. Aygun, and M. Swaminathan, "Enforcing causality and passivity of neural network models of broadband S-parameters," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2019, pp. 1–3.

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

[24] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, "Stability, causality, and passivity in electrical interconnect models," *IEEE Trans. Adv. Packag.*, vol. 30, no. 4, pp. 795–808, Nov. 2007.

[25] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1755–1769, Sep. 2004.

[26] L. L. Barannyk, H. A. Aboutaleb, A. Elshabini, and F. D. Barlow, "Spectrally accurate causality enforcement using SVD-based Fourier continuations for high-speed digital interconnects," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 5, no. 7, pp. 991–1005, Jul. 2015.

[27] J. S. Toll, "Causality and the dispersion relation: Logical foundations," *Phys. Rev.*, vol. 104, no. 6, pp. 1760–1770, Dec. 1956.

[28] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," Jan. 2018, *arXiv:1603.07285*. [Online]. Available: http://arxiv.org/abs/1603.07285

[29] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, Oct. 2016.

[30] A. Dienstfrey and L. Greengard, "Analytic continuation, singular-value expansions, and Kramers-Kronig analysis," *Inverse Problems*, vol. 17, no. 5, pp. 1307–1320, Oct. 2001.

[31] L. Marple, "Computing the discrete-time 'analytic' signal via FFT," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2600–2603, Sep. 1999.

[32] D. W. Robinson, "A note on a simple matrix isomorphism," *Math. Mag.*, vol. 32, no. 4, pp. 213–215, 1959.

[33] J. K. Merikoski, H. Sarria, and P. Tarazaga, "Bounds for singular values using traces," *Linear Algebra Appl.*, vol. 210, pp. 227–254, Oct. 1994.

[34] H. Wolkowicz and G. P. H. Styan, "Bounds for eigenvalues using traces," *Linear Algebra Appl.*, vol. 29, pp. 471–506, Feb. 1980.

[35] 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.

[36] F. M. Tesche, "On the use of the Hilbert transform for processing measured CW data," *IEEE Trans. Electromagn. Compat.*, vol. 34, no. 3, pp. 259–266, Aug. 1992.

[37] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Eds. Lane Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035.

[38] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*. [Online]. Available: https://arxiv.org/abs/1511.07289

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

**Hakki Mert Torun** (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2016, and the M.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2019, where he is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering.

He has coauthored more than 30 refereed technical publications. His current research interests include developing machine learning algorithms for system-level design optimization and modeling with the applications in signal and power integrity in high-speed channels, microwave electronics, and VLSI systems.

Mr. Torun was a recipient of the 2019 Georgia Tech ECE Graduate Student Excellence Award and the Best Student Paper Award of the IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS) in 2018.

**Ahmet Cemal Durgun** (Member, IEEE) received the B.S.E.E. degree, the B.S. degree in mathematics, and the M.S.E.E. degree from Middle East Technical University, Ankara, Turkey, in 2005, 2006, and 2008, respectively, and the Ph.D. degree in electrical engineering from Arizona State University, Tempe, AZ, USA, in 2013.

He is currently with the Assembly and Test Technology Development Department, Intel Corporation, Chandler, AZ, USA, as an Analog Engineer in microelectronic packaging. His research interests include signal integrity of high-speed I/O interfaces, fine pitch interconnects for 2.5-D and 3-D integration, machine learning applications of microelectronic packaging, and applied electromagnetics.

**Kemal Aygün** (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2002.

In 2003, he joined Intel Corporation, Chandler, AZ, USA, where he is currently a Senior Principal Engineer and manages the Electrical Core Competency Group, High-Speed I/O Team. He has coauthored five book chapters and more than 70 journal and conference publications. He holds 51 U.S. patents. His current research interests include novel technologies along with electrical modeling and characterization techniques for microelectronic packaging.

Dr. Aygün was a recipient of the SRC-GRC Mahboob Khan Outstanding Mentor Award in 2008 and 2015. He is also serving as a Distinguished Lecturer for the IEEE Electronics Packaging Society.

**Madhavan Swaminathan** (Fellow, IEEE) received the B.E. degree from Regional Engineering College, Tiruchirapalli (now NITT) 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 is currently the John Pippin Chair in Microsystems Packaging and Electromagnetics at the School of Electrical and Computer Engineering (ECE), a Professor of ECE with a joint appointment at the School of Materials Science and Engineering (MSE), and the Director of the 3D Systems Packaging Research Center (PRC), Georgia Tech (GT). He also serves as the Site Director for the NSF Center for Advanced Electronics through Machine Learning (CAEML) and the Theme Leader for Heterogeneous Integration, SRC JUMP ASCENT Center. He formerly held the position of Founding Director at the Center for Co-Design of Chip, Package, System (C3PS), the Joseph M. Pettit Professor in Electronics in ECE, and the Deputy Director of the Packaging Research Center (NSF ERC), GT. Prior to joining GT, he was with IBM, working on packaging for supercomputers.

Dr. Swaminathan is the author of over 500 refereed technical publications, holds 32 patents, primary author and co-editor of three books, founder and co-founder of two start-up companies, and the founder of the IEEE Conference Electrical Design of Advanced Packaging and Systems (EDAPS), a premier conference sponsored by the EPS Society. He has served as the Distinguished Lecturer for the IEEE EMC Society.