Deep Simple Recurrent Unit-Based Transient Modeling Method for High-Speed Circuits

Hanzhi Ma, *Member, IEEE*, Jiarui Qiu, Guangyu Sheng, Wenchao Chen, *Senior Member, IEEE*, and Er-Ping Li, *Fellow, IEEE*

***Abstract*— The need for rapid and precise transient simulation for signal integrity (SI) assessment of high-speed circuits in microwave systems becomes increasingly crucial. Conventional recurrent methods, including recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and gated recurrent units (GRUs) have been employed for transient modeling. However, despite their accuracy, these methods face limitations such as high computational costs and constrained parallel computing capabilities, especially in high-speed circuits characterized by extremely long input bit patterns. To overcome these challenges, this article presents the Deep Simple Recurrent Unit (DSRU), which is a transient modeling method designed to predict the time-domain signal response of high-speed circuits. The DSRU method strategically segments input signals through windowing and processes them through multiple SRU layers, thus incorporating an improved intra-unit gating mechanism to enhance parallelism. The DSRU method is validated in two high-speed circuit examples, which demonstrate its accuracy and efficiency in the modeling and design of high-speed circuits compared to conventional RNN, LSTM, and GRU methods.**

***Index Terms*— Deep simple recurrent unit (DSRU), microwave computer-aided design, signal integrity (SI), transient simulation.**

I. Introduction

As SIGNAL speeds increase and device sizes decrease, high-frequency effects are now recognized as significant factors constraining the functionality of electronic systems. Even the shortest lines can be plagued by issues like ringing, crosstalk, reflection, and ground bounce, which severely impede signal response and compromise signal integrity (SI) [1], [2]. Effectively addressing these SI concerns is crucial for the successful design of high-speed circuits and microwave devices. Designers routinely perform iterative SI evaluations using both microwave circuit simulators and experimental measurements throughout the design process. However, this approach has been proven to be computationally demanding and time-consuming. Hence, the imperative for rapid and precise SI simulation and modeling becomes increasingly crucial for expediting the design cycles.

Transient simulation is a key technique for modeling SI in high-speed circuits, as it captures the temporal dynamic response of circuit elements, including nonlinear signal distortion. Nevertheless, conventional transient simulation methods and tools [3] consistently encounter a substantial computational burden, especially when dealing with high-speed circuits characterized by extremely long input bit patterns. Consequently, new models for transient simulation in high-speed circuits need to be developed to enhance the efficiency of SI analysis.

Recently, artificial intelligence technologies have provided new tools and approaches to address the challenge of handling large-scale data in high-speed circuits and microwave components with growing complexity. Various artificial intelligence-based methods have been widely applied in SI, covering surrogate models [4], [5], [6], [7], [8], design optimization [9], [10], [11], [12], [13], and uncertainty quantifications [5], [14], [15], [16], [17]. Among these methods, artificial neural networks excel at learning intricate patterns from observed behaviors, allowing them to capture inherent correlations in data, and have been significantly applied in high-speed link modeling and SI analysis [18], [19]. Recurrent Neural Network (RNN) is a class of neural networks specially designed for processing sequential data. Unlike feed-forward neural networks, RNN possesses a mechanism that allows them to retain memory or context information, enabling them to handle time-related data more effectively. Thus, RNN [20] and improved RNNs [21], [22] have been used for transient modeling of high-speed circuits. However, due to gradient vanishing and exploding, RNNs perform poorly in processing long sequences. More advanced recurrent models with gating mechanism have been proposed to solve this problem, e.g., long short-term memory (LSTM) [23], [24] and Gated recurrent unit (GRU) based networks [25], [26]. GRU and LSTM primarily differ in their gating mechanisms, state calculation, and parameter counts. GRUs feature two gates that combine memory and hidden states, making them simpler and quicker to train compared to LSTMs, which consist of three gates.

Although the aforementioned recurrent methods can accurately predict output signals for various high-speed circuits, they face high computational costs and lack the ability for parallel computing, particularly when dealing with extremely

Manuscript received 9 March 2024; revised 15 May 2024 and 28 June 2024; accepted 4 July 2024. Date of publication 18 July 2024; date of current version 6 February 2025. This work was supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant ZCLZ24F0102, in part by the National Key Research and Development Program of China under Grant 2023YFB3812500, and in part by the National Natural Science Foundation of China under Grant 62071424 and Grant 62027805. *(Corresponding authors: Hanzhi Ma; Er-Ping Li.)*

The authors are with Zhejiang University–University of Illinois at Urbana–Champaign Institute, Zhejiang University, Haining 314400, China, and also with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: mahanzhi@zju.edu.cn; jrqiu@zju.edu.cn; guangyu.23@intl.zju.edu.cn; wenchaochen@zju.edu.cn; liep@zju.edu.cn).

Digital Object Identifier 10.1109/TMTT.2024.3426595

long input bit patterns. In the RNN method, the computation of each step relies on the output of the previous state, thus introducing sequence dependency and resulting in a low degree of parallelism. This sequence dependency hampers the training process from fully leveraging the parallel computing power of modern multi-core processors, leading to low training efficiency, which hinders fast simulation in SI. Gated RNNs like LSTM and GRU also encounter inefficiencies due to sequential dependencies in state computations at each time step. This impedes effective parallel processing of different time steps within a sequence, resulting in time-consuming training, especially with long sequences. These limitations hinder the optimal use of parallel computing capabilities, impacting efficiency in training processes. Meanwhile, the design of window schemes is also important in RNNs tailored for time series prediction. Window attention [27], featuring a Fourier hybrid layer, has been introduced to enhance global comprehension, effectively capturing long-range dependencies in data. The adaptive window scheme [28] utilizes varying window sizes to normalize data with significant amplitude variations, thereby facilitating model training. However, these windowing methods do not incorporate corresponding structural designs aimed at parallel computing.

In this article, a transient modeling method based on deep simple recurrent unit (DSRU) is proposed for the first time to predict the time-domain signal response of high-speed circuits, which can improve the serial dependency of traditional RNNs and enhance the parallel computing ability of the network. The DSRU network incorporates an intra-unit gating mechanism which enhances parallelism by allowing the computation for the current moment without waiting for all processes from the previous moment to complete. We also propose the data acquisition and preprocessing method of input–output signals in the proposed DSRU model, providing guidance for converting 1-D time-domain sequences into frame-by-frame formats for neural networks and making DSRU more suitable for transient modeling of high-speed circuits. Specifically, we propose window segmentation of input signals in DSRU method, processing them through multiple SRU layers [29], and leveraging the improved gating mechanisms within the unit to enhance parallelism. With its application to two practical high-speed circuit examples, the DSRU method demonstrates advantages in terms of computational complexity and execution speed compared to RNN, LSTM, and GRU methods. The proposed DSRU method provides an effective way for fast and accurate transient simulation of high-speed circuits, which can help accelerate the design cycle and improve design efficiency.

The article is organized as follows. Section II outlines the workflow of the DSRU method and illustrates its theoretical advantages over conventional recurrent methods. Section III presents the performance evaluation of the DSRU method through its application to two high-speed circuit examples. Section IV concludes this article.

II. Proposed Transient Modeling Method

A. Workflow of DSRU

A DSRU method is proposed for fast transient modeling of high-speed circuit in this study to address the challenges encountered by traditional RNNs in parallel computation. Fig. 1 describes the network of the proposed DSRU method that comprises four parts with input and output data: an input layer, several single SRU layers, a fully connected layer, and an output layer.

1) Input Layer and Data Sampling: The transient simulation of a high-speed circuit, which includes modeling and analyzing the dynamic behavior of the circuit over time, requires the proposed DSRU method to operate frame-by-frame. This requires the sampling of high-speed data that is in progress to capture the time-dependent response. Suppose the input of a high-speed circuit can be represented as $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$, where n denotes the length of the bit sequence. We can convert the bit sequence x to an input matrix $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times M}$ for the input layer in DSRU using a sliding window with a step size of 1. Here

$$\boldsymbol{X}_1 = \begin{bmatrix} x_1 & x_2 & \cdots & x_M \\ x_2 & x_3 & \cdots & x_{M+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-M+1} & x_{n-M+2} & \cdots & x_n \end{bmatrix} \tag{1}$$

where M denotes the length of sliding window as well as the input dimension of the DSRU network, and N_1 is equal to $n - M + 1$. The window length M should cover the current bit as well as previous bits that impact the output waveform, which can be measured through the single-bit response (SBR), thereby ensuring the causality of the model. Given T is the unit interval and U is the number of unit intervals, the SBR lasts for $U \times T$, and the windowing size M should be larger than U. However, an excessively large size will increase network parameters and reduce operational efficiency. Consequently, the value of M needs to be optimized by balancing running speed, convergence speed, and parameter quantity. M is set to 64 in subsequent experiments, which is larger than the SBR of the channels to enable the model to adapt to different durations of SBR in various high-speed links.

The input matrix can be expressed as $\boldsymbol{X}_S \in \mathbb{R}^{N_S \times M}$ for various sliding step sizes S, where $N_S = N_1/S$. Modifying the step size of sliding window allows for control over the information update speed of the input matrix. High-speed circuit transient modeling always requires managing extended sequences, where N_S is typically significantly larger than M. In DSRU method, we handle the input matrix in batches, adjusting it to $\boldsymbol{X} \in \mathbb{R}^{B \times N \times M}$, where B represents the batch size, and $N = N_S/B$, which is the sequence length of the DSRU model.

2) SRU Layer: The architecture of an individual Simple Recurrent Unit (SRU) is also illustrated in Fig. 1, which incorporates a gating mechanism to regulate the flow of input bit sequence and information. When presented with an input $\boldsymbol{x_t}$, the SRU initiates the computation of a lightly recurrent unit. This computation is driven by a forget gate $\boldsymbol{f}_t$ and the state $\boldsymbol{c}_t$, as follows:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{x_t} + \boldsymbol{v}_f \odot \boldsymbol{c}_{t-1} + \boldsymbol{b}_f\right) \tag{2}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \left(1 - \boldsymbol{f}_t\right) \odot (\boldsymbol{W x_t}). \tag{3}$$

Here, the sigmoid function σ is employed to normalize the input and map it to a range between 0 and 1. In this

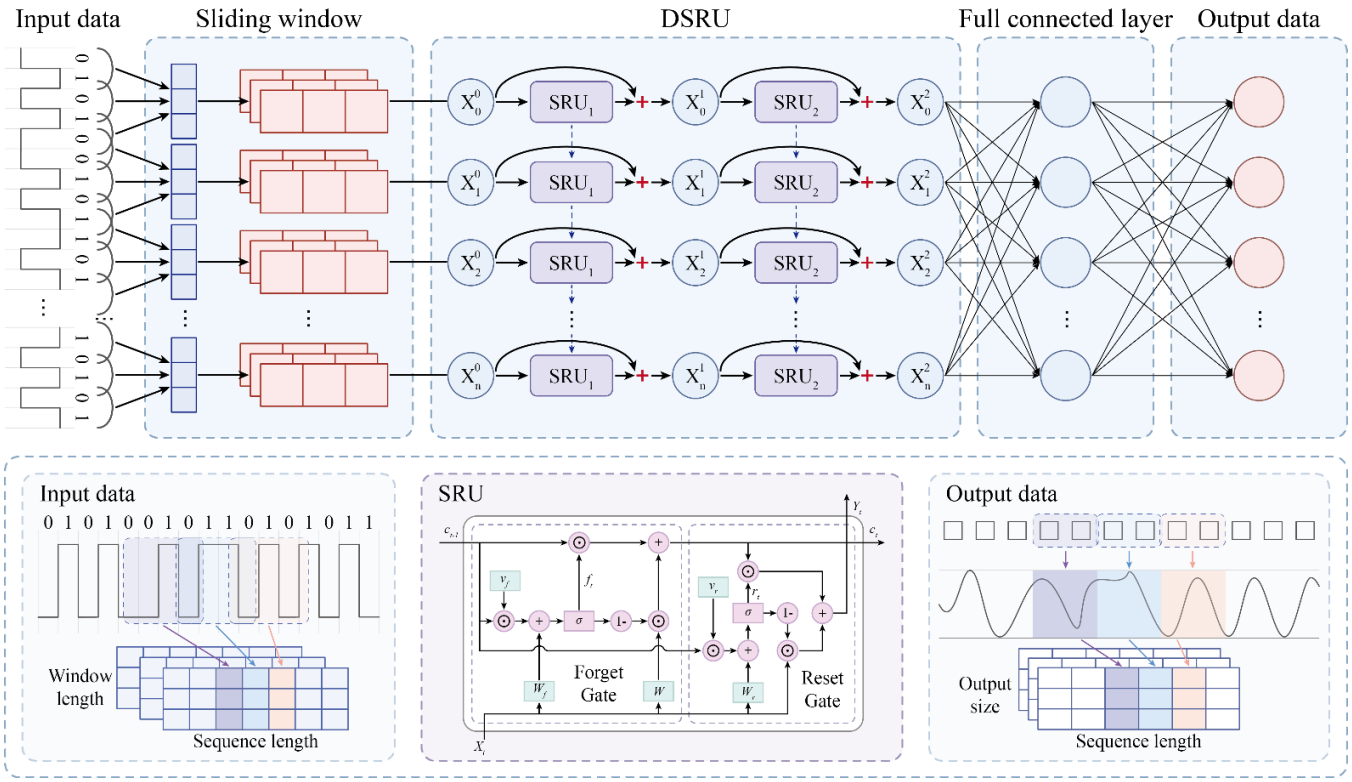


Fig. 1. Workflow of DSRU method with its application to high-speed circuit transient modeling.

context, 0 represents the complete disposal of information, while 1 denotes its complete retention. $\boldsymbol{W}, \boldsymbol{W}_f$ represent the weight matrix and $\boldsymbol{b}_f$ represents the bias, respectively. $\boldsymbol{v}_f$ is a weight vector used to Hadamard product with $\boldsymbol{c_{t-1}}$. The state $\boldsymbol{c}_t$ integrates information from the previous state $\boldsymbol{c}_{t-1}$ and the current input $\boldsymbol{x_t}$, with the degree of retention of past information determined by the forget gate computed in (2).

The second component of the SRU combines a reset gate $\boldsymbol{r_t}$ and a hidden state $\boldsymbol{h}_t$, expressed as follows:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{v}_r \odot \boldsymbol{c}_{t-1} + \boldsymbol{b}_r) \tag{4}$$

$$\boldsymbol{h}_t = \boldsymbol{r}_t \odot \boldsymbol{c}_t + (1 - \boldsymbol{r}_t) \odot \boldsymbol{x}_t \tag{5}$$

where $\boldsymbol{W}_r, \boldsymbol{v}_r$, and $\boldsymbol{b}_r$ represent the weight matrix, weight vector, and bias, respectively. It is worth mentioning that the design of the forget gate and reset gate in the SRU only uses current and previous inputs, ensuring the causality of the model. To reduce the level of recursion, the two gating units in SRU, the forget gate $\boldsymbol{f}_t$ and the reset gate $\boldsymbol{r}_t$, rely on the intermediate state $\boldsymbol{c}_{t-1}$ from the previous moment rather than the hidden state $\boldsymbol{h}_{t-1}$ of the previous moment. Meanwhile, the application of the Hadamard product reduces the number of training parameters and alleviates the computational burden during both forward and backward propagation in the neural network.

In our applications, variational dropout [30] is applied to SRU layer to prevent overfitting. It uses the dropout mask to randomly discard the weights of the input, output, and recurrent connections. For example, (2) with variational dropout applied can be rewritten as

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{M}_{\boldsymbol{w}} \odot \boldsymbol{W}_f \boldsymbol{x_t} + \boldsymbol{M}_v \odot \boldsymbol{v}_f \odot \boldsymbol{c}_{t-1} + \boldsymbol{b}_f\right) \tag{6}$$

where $\boldsymbol{M_w}$ and $\boldsymbol{M}_v$ are mask matrices, each with the same size as $\boldsymbol{W}_f$ and $\boldsymbol{v}_f$. The position of mask is determined by randomly zeroing the weight matrix in variational inference. By adjusting the proportion of zeros in the matrix, known as the dropout probability, the intensity of regularization can be controlled to prevent overfitting. It is worth noting that, once the random mask is determined, it should repeat at all time steps to avoid the damage to the memory capacity of RNNs in the time dimension.

3) Fully Connected Layer and Output Data: The fully connected layer is employed to transform the representation from the hidden feature space following SRU layers into the output data space. Then, the output layer reshapes the data to obtain temporal results, corresponding to the transient output signal of high-speed circuits.

For each row of the input matrix $\boldsymbol{X}_S$, its updated bit sequence information depends on the last S columns. In other words, each row in $\boldsymbol{X}_S$ contains S current bits and M-S previous bits, while each row in the output matrix is the sampling point corresponding to the current bit. To align the output with the input matrix, assuming that the output sampling points corresponding to a single-bit input are denoted as k, we define the output matrix $\boldsymbol{Y}_S \in \mathbb{R}^{B \times N \times K}$, where $K = S \times k$. For the output signal $\boldsymbol{y}$ at time t, it can be derived from the DSRU method as

$$\boldsymbol{y}_t = \boldsymbol{W}_{\text{out}} \cdot \boldsymbol{h}_t(\boldsymbol{x}_t; \boldsymbol{c}_t) + \boldsymbol{b}_{\text{out}} \tag{7}$$

where $\boldsymbol{W}_{\text{out}}$ represents the output matrix, $\boldsymbol{b}_{\text{out}}$ is the bias, and M denotes the input dimension of the DSRU network.

It is worth to mention that, since various high-speed links often bring different attenuation and non-linearity, to ensure the generality of the proposed model, we normalize the output dataset $\boldsymbol{Y}_S$ before the training process of the DSRU method

$$\boldsymbol{Y} = \frac{\boldsymbol{Y}_S - \boldsymbol{Y}_{\min}}{\boldsymbol{Y}_{\max} - \boldsymbol{Y}_{\min}} \tag{8}$$

where $\boldsymbol{Y} \in \mathbb{R}^{B\times N\times K}$ is the normalized output matrix, $\boldsymbol{Y}_{\max}$ and $\boldsymbol{Y}_{\min}$ represent the maximum and minimum value of the entire dataset $\boldsymbol{Y}_S$, respectively, ensuring consistency between training and inference processes. In practice, in addition to using variational dropout regularization and normalization techniques, we prevent overfitting by monitoring the loss curves of both the training and validation sets to ensure model stability.

B. Comparison With Conventional Recurrent Models

1) Computational Complexity and Parallelization: The DSRU method demonstrates a smaller computational complexity and is more suitable for parallelization in comparison to conventional recurrent methods (RNN, LSTM, and GRU). Specifically, the computational complexity of a single hidden layer of DSRU for a batch size of B is $O(NBd)$, while that of conventional RNN, LSTM, and GRU is $O(NBd^2)$ [31], where N and d denote the sequence length and hidden dimension, respectively.

Meanwhile, the computation in RNN relies on the output from the previous state, creating a sequential dependency where each step at time t depends on the calculation of the previous state h_{t-1}. This prevents parallelization across different time steps, thus hindering efficient processing. Gated recurrent methods, like LSTM and GRU, have similar issues since they depend on previous hidden and cell states, which makes computations sequential and time-consuming. This leads to ineffective use of parallel computing capabilities in modern processors for training, which is particularly detrimental to fast transient simulation in high-speed circuits.

In DSRU method, since the matrix values $\boldsymbol{W}$, $\boldsymbol{W}_f$, and $\boldsymbol{W}_r$ are fixed in a single iteration, we can calculate them in advance through matrix multiplication to enhance processing speed. For a given moment t, the matrix multiplications in (2)–(5) can be computed in parallel batches, as shown in the following expression:

$$\boldsymbol{U}^T = \begin{pmatrix} \boldsymbol{W} \\ \boldsymbol{W}_f \\ \boldsymbol{W}_r \end{pmatrix} [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L] \tag{9}$$

where $\boldsymbol{U} \in \mathbb{R}^{N\times 3d}$ represents the merged matrix, and $\boldsymbol{W}, \boldsymbol{W}_f, \boldsymbol{W}_r$ are the weight matrix in (2)–(5). By improving the gating mechanism of the recurrent unit and enhancing the degree of parallelism, the computational speed of SRU is significantly improved compared to conventional recurrent methods. In addition, we have adopted the method of compiling pointwise operations into a single fused kernel to parallelize over the dimension of the hidden state in this study. Therefore, the proposed DSRU is designed to allow highly parallelized implementation.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT RECURRENT MODELS

	Parameter number	Forward time (ms)	Backward time (ms)	Total time (ms)
RNN	16640	4.0	9.73	13.73
LSTM	66560	19.42	32.44	51.86
GRU	49920	12.55	29.47	42.02
DSRU	25088	0.11	0.39	0.5

2) Parameter Number and Execution Speed: We assess the proposed DSRU method alongside traditional recurrent methods such as RNN, LSTM, and GRU. Our analysis focuses on the parameter numbers and execution speed of each method when both have two layers with hidden and input sizes set to 64. As outlined in Table I, the DSRU method demonstrated 62.3% and 49.7% reductions in parameter numbers compared to LSTM and GRU-based methods, respectively. This decrease in trainable parameters makes the DSRU network lighter and more conducive to convergence.

Meanwhile, we utilize a method involving the generation of random tensors to evaluate the execution speed of neural networks [29]. A total of 1000 network tests are conducted using random tensors sized $128 \times 50 \times 64$ as the input dataset on a PC equipped with an Intel Core i7-12700 CPU, NVIDIA GeForce RTX 4070 GPU, and 32 GB of RAM. Subsequently, we compute the average execution times of these four transient modeling methods, as presented in Table I. RNN, LSTM, and GRU required 13.73, 51.86, and 42.02 ms, respectively, to complete one forward and backward pass, whereas DSRU accomplished the task in less than 0.5 ms. These findings confirm that the DSRU method achieves more than a tenfold acceleration, making it highly suitable for rapid modeling and time-domain response prediction in high-speed circuits.

III. HIGH-SPEED CIRCUIT RESULTS AND COMPARISON

To verify the proposed transient modeling method, we have chosen two examples of high-speed circuits. The outcomes of these examples are presented and analyzed in this section.

A. Example I: Bent Microstrip Differential Lines

In this example, we focus on a nonlinear high-speed link system framework comprising a transmitter (Tx), a receiver (Rx), and a passive channel using bent microstrip differential lines, as illustrated in Fig. 2. Differential signaling is chosen as an effective solution to address electromagnetic compatibility issues, which offers symmetrical characteristics that provide high noise immunity, low crosstalk, and reduced electromagnetic interference. However, when differential lines with asymmetric layouts, such as those with varying trace lengths between inner and outer lines (e.g., due to bending discontinuities) are utilized, deviations from the ideal 180° constant phase difference in the output may occur. The phase differences induced by asymmetric layouts could potentially convert differential mode signals into common mode signals, resulting in issues like signal attenuation, common mode noise, and other SI concerns [32]. Therefore, in this example,

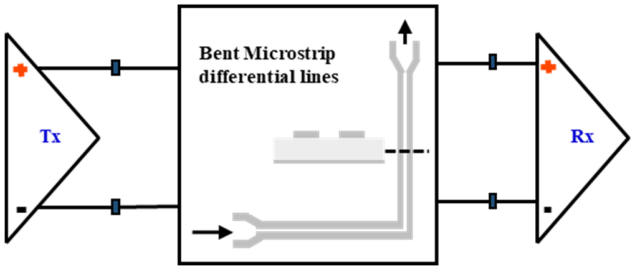
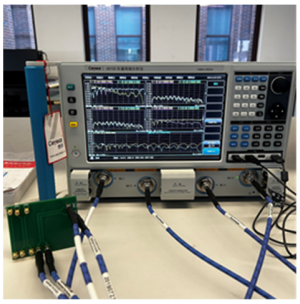
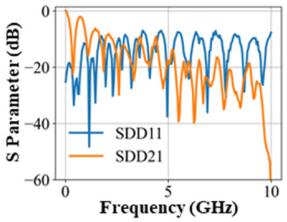


Fig. 2. Bent microstrip differential lines example with nonlinear saturation effect.



(a)

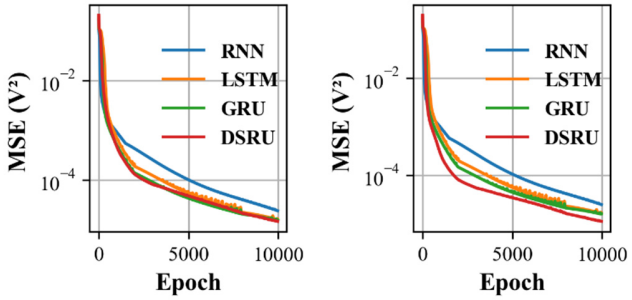


(b)

Fig. 3. (a) S-parameter test of microtrip differential lines. (b) Tested S-parameter results.

TABLE II
COMPARISON OF TRAINING TIME AND ACCURACY OF EXAMPLE I

	Training Time (sec)	Training Data		Validation Data	
		MSE (V^2)	MAE (V)	MSE (V^2)	MAE (V)
RNN	184.97	2.41E-5	3.66E-3	2.45E-5	3.68E-3
LSTM	695.85	1.60E-5	2.98E-3	1.63E-5	2.99E-3
GRU	567.40	1.54E-5	3.00E-3	1.57E-5	3.02E-3
DSRU	20.74	1.45E-5	2.96E-3	1.11E-5	2.57E-3



(a) (b)

Fig. 4. Loss curve for Example I among different methods: (a) Training loss. (b) Validation loss.

we study the SI of bent microstrip differential lines and apply the proposed DSRU method for fast transient modeling. A Vector Network Analyzer (VNA) is employed to measure the S-parameters of the microstrip line, as depicted in Fig. 3. The complete link is simulated using Keysight ADS with experimental S-parameters of the channel. The Tx is driven by a pseudorandom binary sequence (PRBS) with a data rate of 2 Gbps and a voltage ranging from 0 to 1 V. In Rx, nonlinear function $\tanh(\cdot)$ is used to simulate the driver saturation characteristics.

The DSRU method is employed to model the transient performance of this example and predict the output signals after Rx. In this implementation, we set $S = 2$, $M = 64$, $k = 64$, $B = 128$ for the networks as illustrated in Section II-A. The selection of the batch size is made while considering both processor memory constraints and computational speed. A dataset consisting of 300 000 bits is utilized, with 12 800 bits allocated for training, 12 800 bits for testing, and the remaining bits for validation. The high-speed link features fixed transmitters, receivers, and channels, thereby establishing a deterministic relationship between input and output signals. The reason for the large proportion of the validation set is that high-speed circuit transient modeling methods need to be capable of learning input/output characteristics from a small amount of data and utilizing this acquired knowledge to predict the time-domain response of long sequences. In the training procedure, the mean square error (MSE) function is employed as the loss function, the Adam method is used as the optimization function, and we set the dropout probability to 0.3. The utilization of a two-layer model in the trials achieves a harmonious balance between training efficiency and prediction accuracy, making it suitable for modeling nonlinear high-speed circuits.

Table II presents the computation time and accuracy of the DSRU method for the bent microstrip differential line example, while Fig. 4 illustrates the training and validation loss curve for Example I. It also compares these results with those from the RNN, LSTM, and GRU methods, which are considered as reference methods and use the same network configurations as the DSRU method. Metrics such as MSE and mean absolute error (MAE) are employed to evaluate the disparities between actual data and the results obtained during model training or validation. Among these four methods, the training and validation accuracy of the proposed DSRU method surpasses those of the RNN and LSTM methods. In addition, the validation MSE and MAE results of the DSRU method are smaller than those of the GRU method. These results indicate that DSRU demonstrates superior computational accuracy compared to the other three methods and does not produce garbage outputs. It is worth to mention that the DSRU method only needs 20.74 s to complete the training process, which is significantly shorter than those of the other methods. The shortest training time among compared methods, which is achieved by RNN, is 184.97 s—around nine times longer than the DSRU method.

We utilize these four well-trained deep learning models to predict output signals with different input bits for this example. As shown in Fig. 5, the red line, which represents the prediction results from the DSRU method, exhibits the closest fit to the transient simulation results from Keysight ADS (blue line). In other words, the DSRU model can predict the output signal more accurately than the RNN, LSTM, and GRU models. Considering that the eye diagram is a crucial

TABLE III
EYE DIAGRAM PREDICTION COMPARISON OF EXAMPLE I

	Eye Height (V)	Error (V)	Eye Width (ps)	Error (ps)
ADS	0.545	-	365.0	-
RNN	0.543	0.002	382.5	17.5
LSTM	0.544	0.001	375.0	10.0
GRU	0.544	0.001	377.5	12.5
DSRU	0.546	0.001	370.0	5.0

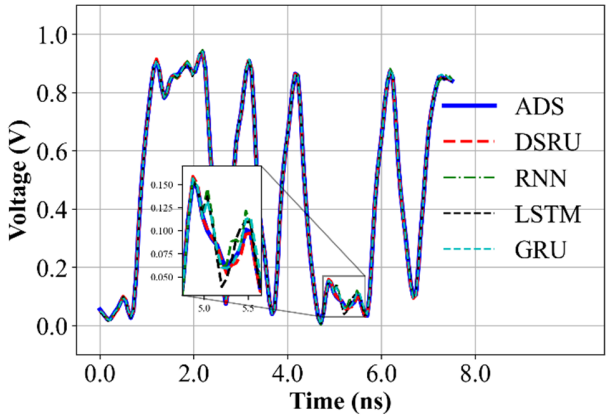


Fig. 5. Prediction results comparison among ADS, RNN, LSTM, GRU, and DSRU of Example I.

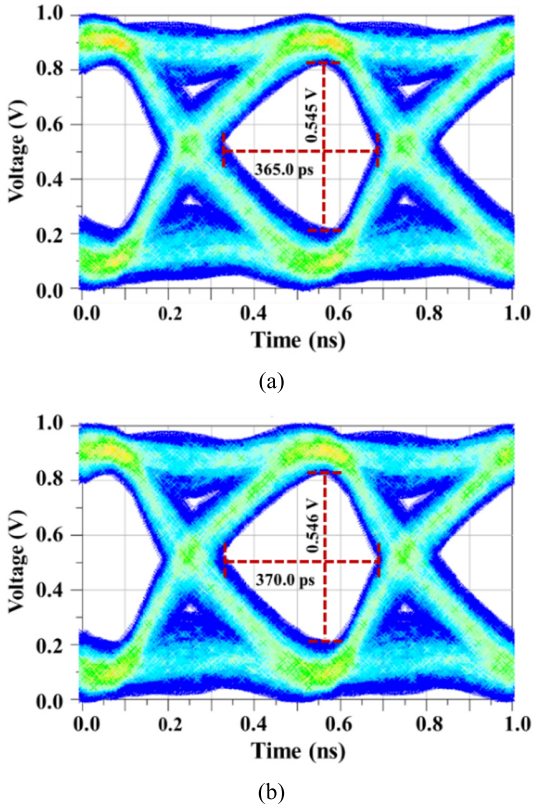


Fig. 6. Eye diagram results of output signals in Example I: (a) ADS. (b) DSRU.

metric for assessing SI of high-speed circuits, we also compare the eye diagram results obtained from transient simulations in ADS with those obtained from the abovementioned four deep learning methods. Fig. 6 describes the details of the transient eye diagrams obtained from ADS and the DSRU model, which verify that the output signal from DSRU fits well with ADS results. Meanwhile, Table III provides the numerical results of eye characteristics from ADS, RNN, LSTM, GRU, and DSRU. The predictive accuracy of eye height and eye width from the DSRU is superior to those from the RNN, LSTM, and GRU.

B. Example II: PCIe

In this study, a more complex and practical example is employed to validate the DSRU method. Peripheral Component Interconnect Express (PCIe) serves as a high-speed serial computer expansion bus standard that establishes a high-bandwidth and low-latency data transmission channel for connecting essential hardware components on the motherboard, including graphics cards, storage devices, network cards, and more. PCIe 4.0, which is the focus of our investigation, as illustrated in Fig. 7, represents the fourth generation of the PCIe standard. It features a data transfer rate of 16 Gbps per lane, doubling that of PCIe 3.0. This doubling of bandwidth enables the transmission of more data over the same number of lanes. In this example, the Tx model is configured with a voltage of 1.05 V, a rise time of 12 ps, and an output resistance of 50 Ω, aligning with the specifications outlined in the PCIe 4.0 standard.

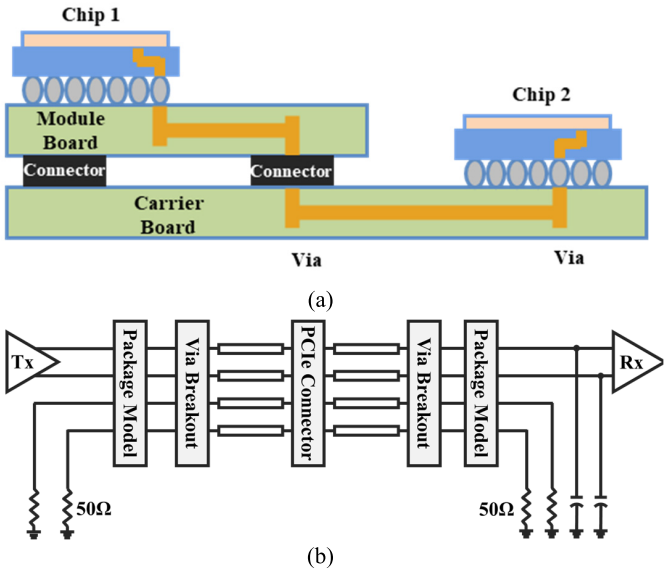


Fig. 7. (a) Structure of the end-to-end channel with a connector of PCIe 4.0 topology. (b) Equivalent circuit simulated in ADS.

For transient modeling of this PCIe example, we employ a DSRU model comprising two hidden layers. Similarly, the architecture of the DSRU model includes 64 input neurons, 64 hidden neurons for each layer, and 128 output neurons. A dataset consisting of 300 000 bits is employed, with 12 800 bits allocated for training, an additional 12 800 bits for testing, and the remaining bits reserved for validation

TABLE IV
COMPARISON OF DSRU, RNN, LSTM, GRU, AND ADS FOR EXAMPLE II

	Training Time (sec)	Training Data		Validation Data		Prediction Accuracy of Eye Diagram			
		MSE (V^2)	MAE (V)	MSE (V^2)	MAE (V)	Eye Height (V)	Error (V)	Eye Width (ps)	Error (ps)
ADS	-	-	-	-	-	0.385	-	48.75	-
RNN	195.79	8.82E-6	2.18E-3	9.77E-6	2.32E-3	0.379	0.006	49.06	0.31
LSTM	799.10	10.14E-6	2.46E-3	10.23E-6	2.47E-3	0.377	0.008	49.38	0.63
GRU	608.29	6.28E-6	1.88E-3	6.38E-6	1.88E-3	0.377	0.008	49.06	0.31
DSRU	22.21	5.81E-6	1.88E-3	3.90E-6	1.58E-3	0.386	0.001	49.06	0.31

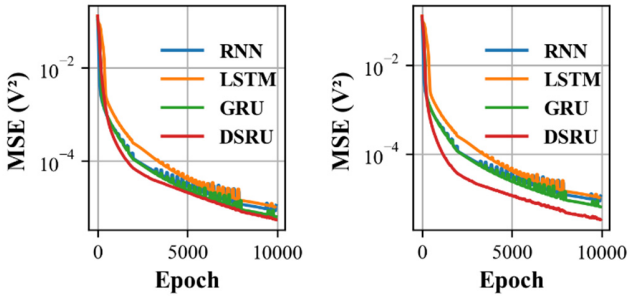


Fig. 8. Loss curve for Example II among different methods: (a) Training loss. (b) Validation loss.

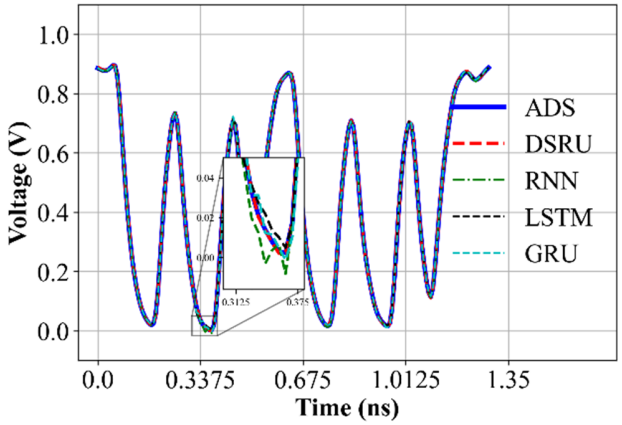


Fig. 9. Prediction results comparison among ADS, RNN, LSTM, GRU, and DSRU of Example II.

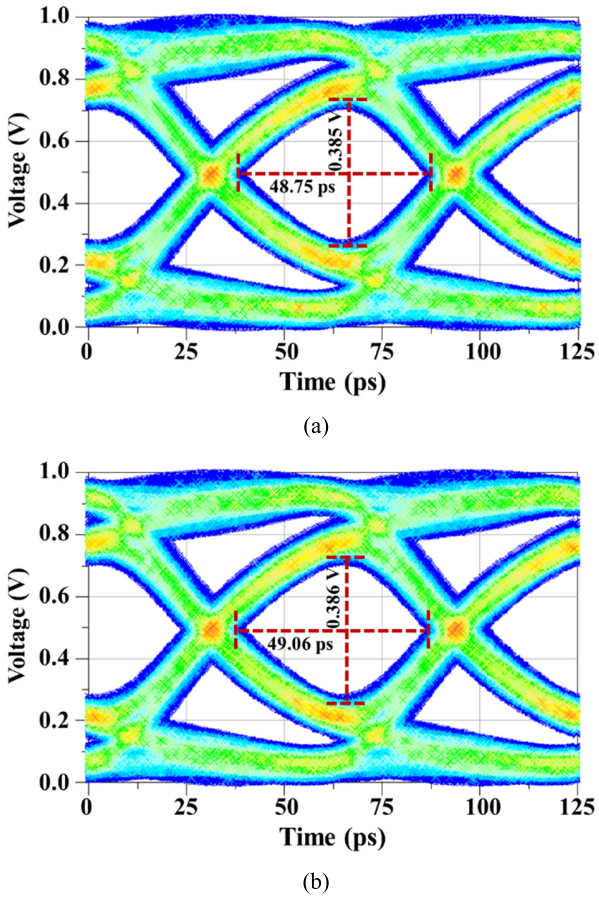


Fig. 10. Eye diagram results of output signals in Example II: (a) ADS. (b) DSRU.

purposes. Simultaneously, we employ three conventional recurrent methods—RNN, LSTM, and GRU—each configured with the same layer and parameter settings. This approach enables us to thoroughly validate the efficiency and accuracy of the DSRU method against established benchmarks.

After 10 000 epochs of training, all four deep learning methods exhibit favorable training and validation loss, as illustrated in Table IV. Fig. 8 also shows the training and validation loss curve for Example II. The DSRU model outperforms the others by achieving superior MSE and MAE values for both training and validation datasets. Notably, the time consumption of the DSRU model is significantly lower than those of the other three models, as illustrated in Table IV. The DSRU model completes the 10 000 training epochs in just 22.21 s, while the RNN, LSTM, and GRU models require 195.79, 799.10, and 608.29 s, respectively. Considering that the same network configuration is used in this study to evaluate the running speed and accuracy of different methods, it is noteworthy that other comparative methods require additional network layers or larger datasets to achieve comparable accuracy. This subsequently leads to increased computation time.

The trained DSRU model is utilized to predict the output signal and corresponding eye diagram for this PCIe example, along with the other three reference models. In Fig. 9, the predicted output signals generated from the deep learning methods are compared with the transient simulation results obtained from ADS. While all four methods align well with the overall trend of the signal, a closer examination of the details reveals that the predictions from the DSRU model match the

simulation results obtained from ADS more closely. In addition, the eye diagram is employed to verify the prediction results. Fig. 10 illustrates the eye diagram results of output signals obtained from ADS and the DSRU methods, showing very close agreement between them. Table IV further compares the eye height and eye width of the output eye diagram among the four deep learning methods. Compared with RNN, LSTM, and GRU, the DSRU method demonstrates the smallest error in eye height and eye width when compared with the ADS simulation results. Overall, the application of this PCIe example underscores that the proposed DSRU method can rapidly generate an accurate transient model for high-speed circuits.

IV. Conclusion

This article proposes a transient modeling method DSRU for high-speed circuits. The DSRU method segments input signals through windowing, subsequently passes them through multiple SRU layers, and then uses a fully connected layer to yield time-domain output. To address the challenges posed by the high computational costs and limited parallel computing capabilities inherent in traditional recurrent methods, the distinctive network architecture of the proposed approach, marked by intra-unit gating, heightened parallelism, and simplified intermediate variables, notably reduces the required training time. The efficacy of the DSRU method is verified through its application to two high-speed circuit examples, highlighting its accuracy and efficiency in comparison to traditional RNN, LSTM, and GRU methods. Overall, the DSRU method stands out as a promising approach for efficient transient modeling of high-speed circuits. Our ongoing efforts are directed toward enhancing the generality of this method, with the goal of enabling a single training process to adapt to various high-speed circuits with different configurations.

References

[1] J. Ren, D. Oh, S. Chang, and F. Lambrecht, "Statistical link analysis of high-speed memory I/O interfaces during simultaneous switching events," in *Proc. IEEE-EPEP Electr. Perform. Electron. Packag.*, Oct. 2008, pp. 25–28.

[2] J. Feng, B. Dhavale, J. Chandrasekhar, Y. Tretiakov, and D. Oh, "System level signal and power integrity analysis for 3200 Mbps DDR4 interface," in *Proc. IEEE 63rd Electron. Compon. Technol. Conf. (ECTC)*, Sep. 2013, pp. 1081–1086.

[3] A. Lamecki and M. Mrozowski, "Equivalent SPICE circuits with guaranteed passivity from nonpassive models," *IEEE Trans. Microw. Theory Techn.*, vol. 55, no. 3, pp. 526–532, Mar. 2007.

[4] O. Akinwande, S. Erdogan, R. Kumar, and M. Swaminathan, "Surrogate modeling with complex-valued neural nets for signal integrity applications," *IEEE Trans. Microw. Theory Techn.*, vol. 72, no. 1, pp. 478–489, Jan. 2024.

[5] F. Treviso, R. Trinchero, P. Keski-Opas, I. Kelander, and F. G. Canavero, "Sensitivity analysis of passive intermodulation due to electrical contacts," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 3, pp. 760–769, Jun. 2022.

[6] H. Ma, E.-P. Li, A. C. Cangellaris, and X. Chen, "Support vector regression-based active subspace (SVR-AS) modeling of high-speed links for fast and accurate sensitivity analysis," *IEEE Access*, vol. 8, pp. 74339–74348, 2020.

[7] T. Nguyen et al., "Comparative study of surrogate modeling methods for signal integrity and microwave circuit applications," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 11, no. 9, pp. 1369–1379, Sep. 2021.

[8] 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.

[9] 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.

[10] K. Son et al., "Reinforcement-learning-based signal integrity optimization and analysis of a scalable 3-D X-point array structure," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 12, no. 1, pp. 100–110, Jan. 2022.

[11] H. Ma, D. Li, E.-P. Li, A. C. Cangellaris, and X. Chen, "A fast optimization method for high-speed link inverse design with SVR-AS algorithm," *IEEE Trans. Signal Power Integrity*, vol. 1, pp. 22–31, 2022.

[12] H. H. Zhang, Z. S. Xue, X. Y. Liu, P. Li, L. Jiang, and G. M. Shi, "Optimization of high-speed channel for signal integrity with deep genetic algorithm," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 4, pp. 1270–1274, Aug. 2022.

[13] J. Zhang, Y.-D. Wang, Y. Wu, K. Kang, and W.-Y. Yin, "Inverse design of on-chip interconnect via transfer learning-based deep neural networks," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 13, no. 6, pp. 878–887, Sep. 2023.

[14] M. Swaminathan, O. W. Bhatti, Y. Guo, E. Huang, and O. Akinwande, "Bayesian learning for uncertainty quantification, optimization, and inverse design," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4620–4634, Nov. 2022.

[15] Y. Guo et al., "Extrapolation with range determination of 2-D spectral transposed convolutional neural network for advanced packaging problems," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 13, no. 10, pp. 1533–1544, Oct. 2023.

[16] P. Manfredi and R. Trinchero, "A probabilistic machine learning approach for the uncertainty quantification of electronic circuits based on Gaussian process regression," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 8, pp. 2638–2651, Aug. 2022.

[17] M. Telescu, R. Trinchero, N. Soleimani, N. Tanguy, and I. S. Stievano, "Stochastic time-domain mapping for comprehensive uncertainty assessment in eye diagrams," *IEEE Trans. Electromagn. Compat.*, vol. 65, no. 6, pp. 1930–1938, Dec. 2023.

[18] Y. Zhao, T. Nguyen, H. Ma, E.-P. Li, A. C. Cangellaris, and J. E. Schutt-Ainé, "Modular neural network-based models of high-speed link transceivers," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 13, no. 10, pp. 1603–1612, Oct. 2023.

[19] S. Choi et al., "Deep reinforcement learning-based optimal and fast hybrid equalizer design method for high-bandwidth memory (HBM) module," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 13, no. 11, pp. 1804–1816, Nov. 2023.

[20] 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.

[21] A. Faraji, M. Noohi, S. A. Sadrossadat, A. Mirvakili, W. Na, and F. Feng, "Batch-normalized deep recurrent neural network for high-speed nonlinear circuit macromodeling," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4857–4868, Nov. 2022.

[22] A. Faraji, S. A. Sadrossadat, A. Moftakharzadeh, M. Nabavi, and Y. Savaria, "Deep independent recurrent neural network technique for modeling transient behavior of nonlinear circuits," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 13, no. 5, pp. 688–699, May 2023.

[23] Y. Luo et al., "Fast response prediction method based on bidirectional long short-term memory for high-speed links," *IEEE Trans. Microw. Theory Techn.*, vol. 71, no. 6, pp. 2347–2359, Sep. 2023.

[24] Z. Wang, Z. Xu, J. He, H. Delingette, and J. Fan, "Long short-term memory neural equalizer," *IEEE Trans. Signal Power Integrity*, vol. 2, pp. 13–22, 2023.

[25] J. Qiu, H. Ma, and S. Tan, "Deep gated recurrent unit network for high-speed links modeling," in *Proc. IEEE 7th Int. Symp. Electromagn. Compat. (ISEMC)*, Oct. 2023, pp. 1–3.

[26] A. Faraji, S. A. Sadrossadat, W. Na, F. Feng, and Q. J. Zhang, "A new macromodeling method based on deep gated recurrent unit regularized with Gaussian dropout for nonlinear circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 7, pp. 2904–2915, Jul. 2023.

[27] N. T. Tran and J. Xin, "Fourier-mixed window attention: Accelerating informer for long sequence time-series forecasting," 2023, *arXiv:2307.00493*.

[28] O. W. Bhatti, H. M. Torun, and M. Swaminathan, "HilbertNet: A probabilistic machine learning framework for frequency response extrapolation of electromagnetic structures," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 2, pp. 405–417, Apr. 2022.

[29] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple recurrent units for highly parallelizable recurrence," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1–12.

[30] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. Adv. in Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1–14.

[31] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple recurrent units for highly parallelizable recurrence," 2017, *arXiv:1709.02755*.

[32] Y. Ye, D. Spina, P. Manfredi, D. V. Ginste, and T. Dhaene, "A comprehensive and modular stochastic modeling framework for the variability-aware assessment of signal integrity in high-speed links," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 2, pp. 459–467, Apr. 2018.

Hanzhi Ma (Member, IEEE) received the B.S. degree and Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2017 and 2022, respectively.

She is currently an Assistant Professor with Zhejiang University, and an Adjunct Assistant Professor with University of Illinois Urbana-Champaign Champaign, IL, USA. She has authored or co-authored more than 60 technical papers and served as a Guest Editor for IEEE Transactions on Components, Packaging and Manufacturing Technology. Her research interests include machine learning techniques for EMI/SI/PI analysis and signal integrity analysis for neuromorphic chip.

Dr. Ma received the President's Memorial Award from IEEE Electromagnetic Compatibility Society in 2020 and 2021, and the Best student paper from Asia-Pacific International Symposium on Electromagnetic Compatibility in 2022. She has also served as a reviewer for five technical journals and as a TPC Member for more than 10 IEEE conferences.

Jiarui Qiu received the B.S. degree in communication engineering from the School of Mechanical, Electrical and Information Engineering, Shandong University, Jinan, China, in 2022. He is currently pursuing the Ph.D. degree in electronics science and technology with the College of Information Science and Electronic Engineering, Zhejiang University–University of Illinois at Urbana–Champaign Institute, Zhejiang University, Haining, China.

His current research interests include machine learning techniques for signal integrity and SI/PI analysis for neuromorphic chip.

Guangyu Sheng received the B.S. degree in electrical engineering and automation from the School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, China, in 2023. He is currently pursuing the M.S. degree in electronic information with Zhejiang University.

His current research interests include electromagnetic compatibility and signal integrity analysis of high-speed and high-frequency integrated circuits.

Wenchao Chen (Senior Member, IEEE) received the B.E. degree in information engineering from Xi'an Jiaotong University, Xi'an, China, in 2006, the M.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2014.

He is an Associate Professor with the ZJU-UIUC Institute, Zhejiang University, Haining, China. He received the Natural Science Foundation of China (NSFC) for Excellent Young Scholars in 2021, and the Zhejiang Provincial Natural Science Foundation for Outstanding Young Scholars in 2020. His research interests include modeling of advanced electronic devices and integrated circuits.

Er-Ping Li (Fellow, IEEE) is currently a Qiushi Distinguished Professor with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, and was the Founding Dean of the Joint Institute of Zhejiang University—University of Illinois at Urbana–Champaign, Zhejiang University, Haining, China. Prior to joining Zhejiang University, he worked with the Singapore Research Institute and University as a Principal Scientist, Professor, and the Senior Director. He has authored or co-authored more than 400 papers published in the referred international journals, and also authored two books published by John-Wiley-IEEE Press and Cambridge University Press. He holds over 50 patents. His research interests include electrical modeling and design of micro/nanoscale integrated circuits, 3-D electronic package integration, and nano-plasmonic technology.

Dr. Li is a Fellow of the Singapore Academy of Engineering and USA Electromagnetics Academy. He was a recipient of IEEE EMC Technical Achievement Award in 2006, Singapore IES Prestigious Engineering Achievement Award and Changjiang Chair Professorship Award in 2007, 2015 IEEE Richard Stoddard Award on EMC, 2021 IEEE EMC Laurence G. Cumming Award, and Zhejiang Natural Science 1st Class Award. He served as an Associate Editor for IEEE Microwave and Wireless Components Letters from 2006 to 2008 and for IEEE Transactions on Electromagnetic Compatibility from 2006 to 2021. He is currently an Associate Editor for IEEE Transactions on Signal and Power Integrity and the Executive Editor-in-Chief of the Electromagnetics Science. He has been the General Chair and the Technical Chair for many international conferences. He was the President of 2006 International Zurich Symposium on EMC, the Founding General Chair of Asia-Pacific EMC Symposium, General Chair of 2008, 2012, 2016, 2018, 2022 APEMC, and 2010 IEEE Symposium on Electrical Design for Advanced Packaging Systems. He has been invited to give 120 invited talks and plenary speeches at various international conferences and forums.