

# SPIPE: Differentiable SPICE-Level Co-Simulation Program for Integrated Photonics and Electronics

Zhengqi Gao<sup>†</sup>, Jiaqi Gu<sup>‡</sup>, Member, IEEE, Luca Daniel<sup>†</sup>, Fellow, IEEE,  
Ron Rohrer<sup>§</sup>, Life Fellow, IEEE, and Duane S. Boning<sup>†</sup>, Fellow, IEEE

**Abstract**—Heterogeneous photonic-electronic systems, such as co-packaged optics and photonic-electronic artificial intelligence (AI) accelerators, are rapidly gaining traction but also pose significant design challenges due to distinct design methodologies. Digital and analog electronics are typically described using hardware description languages and SPICE, respectively, whereas photonic devices and systems are represented using permittivity tensors on the Yee grid and the Scattering matrix formulation. This disparity necessitates an end-to-end photonic-electronic co-simulation tool to streamline co-design. Most preliminary co-simulation approaches rely on translating photonic compact models into Verilog-A or SPICE models to simulate everything there, which not only introduces the additional complexity of model conversion but also has potential numerical stability problems. Additionally, another critical functionality missing from the current implementation is enabling gradient calculation in these co-simulators, which will be crucial for end-to-end gradient-based electronic-photonic system optimization.

To address these challenges, we introduce SPIPE, a differentiable SPICE-level co-simulation framework for integrated photonic-electronic systems. SPIPE is the first co-simulator to overcome model conversion issues and to provide differentiability. Numerical experiments on several circuits confirm the accuracy of SPIPE when compared to analytical solutions and real-world experimental data. Furthermore, in cases where existing simulators are applicable, SPIPE achieves a runtime reduction of 2~85 $\times$  compared to an industry-standard simulator. SPIPE features an integrated simulation interface with a low usage barrier, opening avenues for more accessible and effective photonic-electronic co-design. SPIPE is open sourced: <https://github.com/zhengqigao/spipe>.

**Index Terms**—heterogeneous photonic-electronic system, photonic-electronic co-simulator

## I. INTRODUCTION

Although the concept of free-space bulk optics has existed for several centuries, the miniaturization of optical components onto an integrated chip is a relatively recent development. There are several technology platforms available for this integration purpose, such as GaAs, InP, and silicon. Among all candidates, silicon stands out due to its compatibility with CMOS technology, which was justified in the early 2000s [1], leading to its widespread adoption under the

<sup>†</sup>Zhengqi Gao, Luca Daniel, and Duane S. Boning are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: {zhengqi, dluca, boning}@mit.edu). Zhengqi Gao is the corresponding author.

<sup>‡</sup>Jiaqi Gu is with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA (e-mail: jiaqi.gu@asu.edu).

<sup>§</sup>Ron Rohrer is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA (e-mail: rr1j@andrew.cmu.edu).

term integrated silicon photonics. Since then, extensive research effort has been put into investigating silicon photonics for various applications. It has enabled the replacement of electronic interconnects with optical interconnects in short links, such as those within data centers and chip-to-chip communication [2], [3], driven by the need for higher data bandwidth and lower latency. Additionally, optical general matrix multiplication units, a.k.a. photonic tensor cores (PTCs), have been successfully demonstrated in experiments [4], [5], [6], serving as building blocks for artificial intelligence (AI) acceleration. A novel reconfigurable optical paradigm has also been introduced, utilizing Mach-Zehnder modulators (MZMs) to enable versatile linear optical signal processing functions at runtime [7], [8]. Beyond these advances, silicon photonics is finding growing applications in other domains, including Lidar [9], [10], quantum computing [11], and biosensing [12].

Most cases mentioned above are heterogeneously co-integrated photonic-electronic systems since electrical signals are always required to drive active photonic modulators (e.g., MZMs and ring modulators). At present, it remains common practice to design electronics and photonics separately, with human expertise serving as the interface (e.g., setting metrics and goals for each sub-component). This separation arises because the design methodologies for electronics and photonics remain largely distinct and specialized within their respective communities, with very few fluent in the language of both disciplines. Specifically, digital and analog electronics are described using hardware description languages [13] and SPICE [14], respectively, whereas photonic devices and systems are represented using permittivity tensors on the Yee grid [15] and the Scattering matrix formulation [16]. Additionally, end-to-end simulation tools capable of holistically addressing both domains are still underdeveloped, with no dominant solution. Most existing co-simulation methods [17], [18], [19], [20], [21], [22], [23], [24], [25], [26] depend on converting photonic compact models into Verilog-A or SPICE for unified simulation. However, this approach not only adds complexity due to model conversion overhead but also introduces potential numerical stability issues. Furthermore, a key missing feature in current implementations is the ability to compute gradients within these co-simulators, a critical capability for enabling gradient-based optimization of photonic-electronic systems.

In this paper, we propose a differentiable SPICE-level co-simulation program for integrated photonics and electronics, abbreviated as SPIPE. We develop a customized differentiable frequency-domain Scattering matrix simulation for the ph-

tonic components and, leveraging the time-domain adjoint method [27], [28], enable differentiable transient simulation for the electronic components as well. SPIPE accepts a text file using an extended SPICE syntax (see Figure 2 and 3) to describe the photonic-electronic circuit and outputs both the analytical optical signal values (i.e., complex numbers representing the magnitude and phase of the electromagnetic (EM) mode coefficients) and the electrical signal values (i.e., real numbers representing voltage or current). SPIPE is the first co-simulator to overcome model conversion issues (e.g., eliminating the need to convert photonic compact models into Verilog-A or SPICE) and provide differentiability. Numerical experiments on several circuits confirm the accuracy of SPIPE when compared to analytical solutions and real-world experimental data. Furthermore, in cases where existing simulators are applicable, SPIPE achieves a runtime reduction of 2~85 $\times$  compared to an industry-standard simulator. In summary, our contribution and the advantages of SPIPE are:

- SPIPE eliminates model conversion issues (i.e., translating photonic models into Verilog-A or SPICE). Instead, it directly utilizes SPICE for electronic simulation and the Scattering matrix method for photonic simulation, with a customized interface to facilitate seamless interaction between the two domains.
- SPIPE is differentiable, enabling the computation of derivatives of an optical signal with respect to voltage or current signals.
- SPIPE features a low usage barrier and an intuitive syntax for defining photonic-electronic systems, making it easy to adopt and use.
- SPIPE will be open-sourced, enabling the community to collaboratively improve and innovate in photonic-electronic system co-design.

The remainder of this paper is structured as follows. Section II provides a review of related works and key preliminaries. Section III details the design of SPIPE. In Section IV, we present numerical experiments demonstrating the accuracy and performance of SPIPE. Finally, we conclude in Section V.

## II. PRELIMINARIES

Since the confirmation of compatibility of silicon photonics with CMOS technology in 2004 [1], research on heterogeneous photonic-electronic circuits has expanded significantly. To support their design and verification, specialized simulators with various purposes have proliferated in recent years, emerging from both academia and industry. For example, Ansys Lumerical offers FDTD for device-level modeling and INTERCONNECT for circuit-level modeling, while Synopsys provides OptSim and OptoCompiler for comprehensive photonic design and simulation. On the academic front, MEEP [29] provides an open-source platform for differentiable EM simulations. TorchONN [30] has been developed as a high-level optical computing modeling framework built on PyTorch. BOSIM [31] focuses primarily on the modeling of optical modulators based on free-carrier plasma dispersion effect, while also supporting passive photonic components such as waveguides and

directional couplers. Other notable tools include GDSfactory, Luceda IPKISS, Tidy3D, OptiSpice [25], Fiona [20], etc.

Amidst these advancements, the development of an end-to-end photonic-electronic co-simulator that can simulate the frequency spectrum of photonic components and the time-domain transient voltage/current in electronic circuits remains relatively unexplored and immature. Two preliminary methodologies have been investigated thus far. The first approach builds on Verilog-A [17], [18], [19], [20]. Specifically, these works declare a new analog variable  $E$ , representing the electric field. Then they associate an array  $E$  of length  $4(N + M)$  to a photonic device with  $N$  input ports and  $M$  output ports. The factor of 4 arises due to two reasons: (i) the bidirectional propagation (forward and backward) of optical signals, and (ii) the lack of native support for complex numbers in Verilog-A, so that the real and imaginary components (or equivalently, the magnitude and phase) must be explicitly expanded into separate variables. Moreover, if  $D$  EM modes are considered, then the factor will change from 4 to  $4D$ .<sup>1</sup> As an example, the authors in [17], [18] develop Verilog-A models for core photonic components and validate their accuracy in systems such as single-sideband modulators, wavelength-division multiplexing (WDM) links, and optical transceivers. However, this methodology has three prominent issues that hinder its widespread adoption. First, photonic compact models are typically provided by foundries in the Scattering matrix formulation, rather than in Verilog-A. Hence, designers must perform an extra model conversion step. Second, Verilog-A is a hardware description language designed for behavioral simulation. Even if a photonic-electronic system is simulated successfully in Verilog-A, translating its electronic part into a gate-level design remains challenging. Third, Verilog-A is inherently non-differentiable, yet derivative computation is essential for gradient-based end-to-end optimization in photonic-electronic systems.<sup>2</sup>

The second approach models photonic devices directly in SPICE [21], [22], [23], [24], [25], [26]. Similar to the conversion into Verilog-A, this method also requires modifying the original schematic of an  $N$ -input-port and  $M$ -output-port photonic device to accommodate  $4(N + M)$  ports in the SPICE schematic. OptiSpice [24] constructs SPICE equivalent models for photonic devices based on their underlying physics. However, this approach requires human expertise to develop a new SPICE equivalent model for each newly encountered photonic device. In contrast, the method proposed in [26] employs a complex vector fitting (CVF) algorithm [32] to convert a given Scattering matrix of a photonic device into a rational pole-residue (or zero-pole) model in the Laplace domain, which is then used to generate an equivalent SPICE circuit model. However, CVF itself may introduce numerical issues when determining zeros and poles [32]. Additionally, modeling photonic devices in SPICE results in the loss of analytical optical signal information (i.e., EM mode coefficients) within a photonic device. For example, a Mach-

<sup>1</sup>Most integrated photonics are designed and operated to host one single EM mode.

<sup>2</sup>End-to-end optimization has not yet been adopted in practice but is an interesting future research direction.

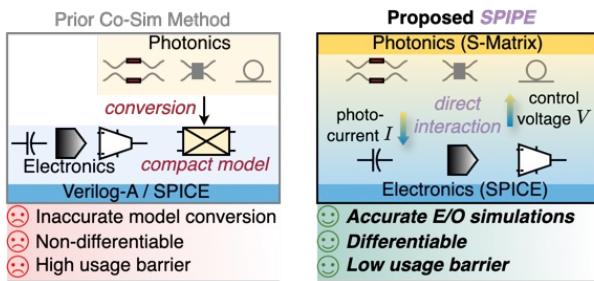


Fig. 1: Left: Previous work converted photonic components into Verilog-A or SPICE models, which suffer from model conversion issues and has high usage barrier. Right: SPIPE preserves the established simulation methodologies for each domain, avoiding model conversion problems, enabling differentiability, and reducing usage barrier.

Zehnder interferometer (MZI) is a composite photonic device typically made up of directional couplers and phase shifters. If a SPICE-equivalent model is used for an MZI, retrieving the analytical optical signal at an intermediate node inside the MZI becomes challenging, if not impossible.

Our key motivation is that converting models into Verilog-A or SPICE is complicated, introduces overhead, and deviates from standard photonic design practices, thereby hindering the broader adoption of end-to-end photonic-electronic co-simulation and co-design. From first principles, an effective co-simulator should preserve the established methodologies of each domain while integrating them seamlessly. This approach ensures that when only electronic or photonic circuits are involved, the co-simulator naturally reverts to conventional tools and workflows familiar to electronic and photonic designers. Our proposed SPIPE adheres to this philosophy, **for the first time** eliminating the need for model conversion (see Figure 1) while also providing differentiability.

### III. PROPOSED METHOD

SPIPE can simulate photonic-electronic systems in both time and frequency domains given a plain text file with SPICE-extended syntax (see Figure 2 and 3). We first describe simulations for purely electronic circuits, followed by pure photonic circuit simulations. Then, we discuss the electrical-optical interfaces (photonic modulators and photodetectors) and conclude with the overall framework of SPIPE.

#### A. Time-Domain Simulation for Electronics

Although earlier analog electronic circuit simulators existed (e.g., IBM TRAC and ECAP), the origin of modern circuit simulation is often attributed to the CANCER project [33], which was later renamed to the more widely known SPICE [14]. Many classical references exist on electronic circuit simulators, so we will not delve into detail and will instead briefly cover the key aspects of circuit theory and simulation for self-containment. The fundamental principles governing an electronic circuit are Kirchhoff's current law

(KCL) and voltage law (KVL). Essentially, SPICE simulators automatically track all constraints imposed by KCLs and KVLs that an analog electronic circuit should satisfy. This procedure is achieved in terms of modified nodal analysis (MNA) [34]. The mathematical abstraction produced by MNA varies depending on the specific type of electronic circuit. For instance, an electronic circuit with only resistors can be represented by the linear system  $\mathbf{G}\mathbf{x}(t) = \mathbf{u}(t)$ . A circuit containing resistors, capacitors, and voltage and current sources is modeled by  $\mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{G}\mathbf{x}(t) = \mathbf{u}(t)$ , where  $\dot{\mathbf{x}}(t)$  represents the derivative of  $\mathbf{x}(t)$  with respect to  $t$ . For a general nonlinear circuit involving MOS transistors, the system can be abstracted as:

$$\dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \theta, t) = 0, \quad (1)$$

where  $\theta$  represents circuit parameters (e.g., resistors, capacitors, and MOS width/length), and the form of  $\mathbf{f}$  is yielded by MNA. Solving this ordinary differential equation (ODE) can be done by numerical integration methods such as Forward Euler or Trapezoidal method given the initial condition  $\mathbf{x}(0)$ .

One key feature of our SPIPE is differentiability. For the electronic part, differentiable computation relies on the adjoint method [28], [27], [35], [36]. Adjoint method efficiently computes the derivatives of voltages, currents, or a metric with respect to a circuit parameter  $\theta$ . Mathematically, if we define a metric  $L = L(\mathbf{x}(T))$ , we observe that  $L$  is also a function of  $\theta$  since  $\mathbf{x}(T)$  implicitly depends on  $\theta$  and  $\mathbf{x}(0)$  through the ODE given in Eq. (1). The adjoint method provides a convenient way to calculate  $\partial L / \partial \theta$  [35]:

$$\frac{\partial L}{\partial \theta} = - \int_T^0 [\mathbf{z}(t)]^T \frac{\partial \mathbf{f}}{\partial \theta} dt, \quad (2)$$

where the term inside the integral is the inner product of two vectors. Here, the vector  $\mathbf{z}(t)$  is governed by another ODE, which should be solved in reverse time [35] after  $\mathbf{x}(T)$  is solved from the original ODE shown in Eq. (1). This reverse-time ODE is usually referred to as the adjoint ODE [35]:

$$\dot{\mathbf{z}}(t) = - \left[ \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]^T \mathbf{z}(t), \quad \text{s.t.} \quad \mathbf{z}(T) = \frac{\partial L}{\partial \mathbf{x}(T)}. \quad (3)$$

In the context of analog circuit simulation, it has been long established that this adjoint ODE also corresponds to an adjoint circuit [28]. In fact, the adjoint circuit is straightforward to build with the lookup table provided in [28].

Fortunately, the SPICE family of simulators is abundant and well-maintained. Rather than reinventing the wheel, we integrate HSPICE and Xyce as subroutines in SPIPE for time-domain electronic circuit simulations. Notably, Xyce already includes an implementation of adjoint sensitivity analysis, whereas HSPICE does not. However, HSPICE provides numerical differentiation for derivative calculations.

#### B. Scattering Matrix Simulation for Photonics

Photonic simulators operate at different levels of abstraction, depending on the scale and complexity of the system. At the device level, full-wave electromagnetic simulations such as the finite-difference time-domain (FDTD) method solve Maxwell's equations on a discretized space-time grid, requir-

ing 10–20 grid points per wavelength and time steps on the order of 10 attoseconds to satisfy numerical stability. As a result, simulating even a single photonic device of a few hundred micrometers in size can involve thousands of grid points. Extending this approach to large-scale photonic circuits with hundreds of components is computationally prohibitive. To address this challenge, system-level simulations typically use the Scattering matrix (S-matrix) approach, which models interconnected components through their input-output relations without requiring full-wave calculations.

For a general passive photonic device with  $N$  left ports and  $M$  right ports, if analytical optical signals  $\mathbf{l}^{\text{in}}(\omega) \in \mathbb{C}^N$  with angular frequency  $\omega$  are injected from the left ports, then the Scattering matrix method models the outputs at the right ports  $\mathbf{r}^{\text{out}}(\omega) \in \mathbb{C}^M$  as:

$$\mathbf{r}^{\text{out}}(\omega) = \mathbf{S}(\omega) \mathbf{l}^{\text{in}}(\omega), \quad (4)$$

where  $\mathbf{S}(\omega) \in \mathbb{C}^{N \times M}$  is the device Scattering matrix (or S matrix). Take the 2-port input and 2-port output active MZM shown in Figure 2 as an example. Its Scattering matrix provides the relation between its inputs  $\{l_1, l_2\}$  and its outputs  $\{r_1, r_2\}$  (where we have omitted the superscript ‘in/out’ for simplicity):

$$\begin{bmatrix} r_1(\omega) \\ r_2(\omega) \end{bmatrix} = \begin{bmatrix} \cos \phi & i \sin \phi \\ i \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} l_1(\omega) \\ l_2(\omega) \end{bmatrix}, \quad (5)$$

where the phase shift  $\phi$  can be adjusted at run-time, either through the thermal-optic effect or the free-carrier plasma dispersion effect [37]. As another example, the Scattering matrix for a straight waveguide with 1-port input and 1-port output is given by:

$$r(\omega) = \exp\left(-j \frac{\omega n_{\text{eff}} L}{c}\right) l(\omega), \quad (6)$$

where  $n_{\text{eff}}$  is the effective EM mode index,  $L$  is the length of the waveguide, and  $c$  is the speed of light in free space.

Due to the bi-directional propagation of optical signals, Eq. (4) is insufficient, and an additional equation is needed to model the signals traveling in the opposite direction. Instead of writing two separate equations, it is standard practice to combine them into one:

$$\begin{bmatrix} \mathbf{r}^{\text{out}}(\omega) \\ \mathbf{l}^{\text{out}}(\omega) \end{bmatrix} = \mathbf{S}(\omega) \begin{bmatrix} \mathbf{l}^{\text{in}}(\omega) \\ \mathbf{r}^{\text{in}}(\omega) \end{bmatrix}, \quad (7)$$

where now  $\mathbf{S}(\omega) \in \mathbb{C}^{(N+M) \times (N+M)}$ . Here  $\mathbf{l}(\omega)^{\text{in}} \in \mathbb{C}^M$  and  $\mathbf{r}(\omega)^{\text{in}} \in \mathbb{C}^N$  are the input optical signals injected from left and right ports, respectively. SPIPE adopts Eq. (7) for bi-directional modeling so that it can work with photonic circuits containing loops, such as recirculating programmable photonic mesh [7].

Now, let us consider a photonic circuit with  $N_d$  components, where each component is described by a Scattering matrix as shown in Eq. (7) and they are connected to others at  $N_p$  ports. We associate each port with two analytical optical signals — one for the ‘forward’ direction and one for the ‘backward’ direction, where ‘forward’ and ‘backward’ are arbitrary but fixed assignments to represent the bidirectional light propagation. Then, overall we have a vector  $\mathbf{x} \in \mathbb{C}^{2N_p}$

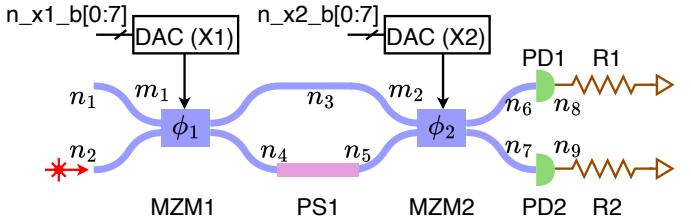


Fig. 2: Circuit schematic of a basic PTC proposed in [38]. The photonic part consists of two MZMs, a passive phase shifter (PS), and two photodetectors (PDs). A laser emits light into the lower port of MZM1. Digital-to-Analog Converters (DACs) provide electrical signals to drive the MZMs, varying their phases  $\{\phi_1, \phi_2\}$ . For illustration purposes, the DACs are depicted with 8-bit resolution. The corresponding text file describing this PTC is shown in Figure 3.

```
* Electronic part, {X1,X2} are DACs
.subckt 8bitDAC b0 b1 b2 b3 b4 b5 b6 b7 out
. . . . . # the 8bitDAC implementation is omitted
.ends 8bitDAC

X1 n_x1_b0 n_x1_b1 n_x1_b2 n_x1_b3 m1 8bitDAC
X2 n_x2_b0 n_x2_b1 n_x2_b2 n_x2_b3 m2 8bitDAC

R1 n8 0 1e3 # R1=1e3 Ω connects n8 to ground
R2 n9 0 1e3

.tran 0 2ns 100 # 100 uniform time samples in [0,2ns]
.print tran v(n8) v(n9)

* Photonic part

Mzm1 n1 n2 n3 n4 m1 <model_name> <param1=value1, ...
Mzm2 n3 n5 n6 n7 m2 <model_name> <param1=value1, ...

Ps1 n4 n5 <param1=value1, ...

Pd1 n6 n8 <model_name> <param1=value1, ...
Pd2 n7 n9 <model_name> <param1=value1, ...

.mode neff=2.35
.freq 193.4thz 193.6thz 3
.source 1.0@n2 # '1.0' optical signal injected to n2
```

Fig. 3: A simplified input netlist representing the circuit shown in Figure 2 for SPIPE. Comments are shown in green starting with \* or #, device definitions (including parameters and connections) are displayed in black, and the control statements are highlighted in orange.

needed to solve at a specific  $\omega$ . Notice that for a general device described by Eq. (7), it actually imposes  $(N + M)$  linear equality constraints on  $\mathbf{x}$ . Consequently, we can formulate a matrix equation  $\mathbf{Ax} = \mathbf{b}$  by rewriting the Scattering matrix relations as linear equality constraints on  $\mathbf{x}$  for each of the  $N_d$  components, and further solve it by inverting matrix  $\mathbf{A}$ , i.e.,  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ . We note that if the photonic circuit response at different frequency points  $\{\omega_1, \omega_2, \dots\}$  are desired, we need to solve  $\mathbf{Ax} = \mathbf{b}$  repeatedly at each frequency point. The above describes a standard Scattering matrix simulation

procedure, which has been widely adopted for system-level integrated photonic simulations in tools such as Lumerical INTERCONNECT and Luceda IPKISS.

In SPIPE, we extend it from two perspectives. First, we make it differentiable. Specifically, considering a parameter  $\theta$  (e.g., the phase shift  $\phi$  in the MZM model shown in Eq. (5) or the length  $L$  in the waveguide model shown in Eq. (6)) of a photonic component, we can calculate the derivative of  $\mathbf{x}$  with respect to it [39]:

$$\frac{d\mathbf{x}}{d\theta} = -\mathbf{A}^{-1} \frac{d\mathbf{A}}{d\theta} \mathbf{x}. \quad (8)$$

It implies a little computational overhead is needed to make the Scattering matrix simulation differentiable, as  $\mathbf{A}^{-1}$  is already needed in solving  $\mathbf{x}$ , and only a limited number of elements in  $d\mathbf{A}/d\theta$  are non-zero. The second extension will be clear in the next subsection.

### C. Electrical and Optical Interface

To enable photonic-electronic co-simulation, it is crucial to accurately model their interfaces, specifically modulators (which convert electrical signals to optical signals) and photodetectors (which convert optical signals to electrical signals).

*a) Modulators:* The conventional Scattering matrix model for a 2-port input and 2-port output active MZM has been shown in Eq. (5). Regardless of the mechanism adjusting  $\phi$  (e.g., thermal-optic effect or the free-carrier plasma dispersion effect [37]), the on-chip implementation typically involves the application of an electrical signal to induce heat dissipation or carrier movement. Noticing such, we can model  $\phi$  using a  $D$ -th order Taylor expansion (around  $v = 0$ ):  $\phi(\omega) = \sum_{d=0}^{D-1} a_d(\omega) \times v^d$ . Note that the coefficient functions  $\{a_d(\omega)\}_{d=1}^D$  should be provided when defining the MZM in SPIPE and that this won't have any modeling inaccuracy as long as  $D$  is sufficiently large. In practice, we anticipate that these coefficients will be supplied by foundries if SPIPE is adopted.

In fact, the model above only states that if a constant voltage  $v$  is applied for a sufficiently long period, then the phase  $\phi$  will ultimately reach a steady value expressed by the Taylor expansion. However, what if the value  $v$  changes before  $\phi$  reaches a steady state? This scenario requires us to adapt the MZM model to explicitly consider the time axis, i.e., dealing with  $v(t)$ . This introduces an open question that has not yet been satisfactorily addressed.

**Assumption 1.** *A photonic modulator responds instantly given an electrical signal to drive it. Namely, when a voltage  $v$  is used to drive the modulator at time  $t$ , the modulator immediately changes to the phase shift  $\phi$  corresponding to  $v$ , i.e.,  $\phi$  doesn't have a transient period.*

We notice that with Assumption 1, whose rationale and validity we will subsequently justify, we can essentially add the time variable into the Scattering matrix in Eq. (5) and update our MZM model:

$$\begin{bmatrix} r_1(\omega, t) \\ r_2(\omega, t) \end{bmatrix} = \begin{bmatrix} \cos \phi(\omega, t) & i \sin \phi(\omega, t) \\ i \sin \phi(\omega, t) & \cos \phi(\omega, t) \end{bmatrix} \begin{bmatrix} l_1(\omega, t) \\ l_2(\omega, t) \end{bmatrix}, \quad (9)$$

$$\phi(\omega, t) = \sum_{d=0}^{D-1} a_d(\omega) \times [v(t)]^d. \quad (10)$$

Eqs. (9)-(10) may seem controversial at first sight, with both time and frequency variables appearing. However, substituting specific time point values (e.g.,  $t_1$  and  $t_2$ , where  $t_1 \neq t_2$ ) into Eqs. (9)-(10), we note that essentially all they do is to decouple the calculation of  $\phi$  at two time points, which is precisely Assumption 1.

Now we explain the validity of Assumption 1. At the contemporary technology node, electronic circuits typically operate at frequencies  $\leq 10$  GHz, and similarly, high-speed DACs can output signals at 10 Giga samples per second. This roughly translates to the  $v(t)$  used to drive the photonic modulator, which is approximately regarded as a constant for every 0.1 ns. If the transient time that  $\phi$  needs to reach its value (e.g., 0.1 ps) is much less than that,  $v(t)$  will appear constant over a sufficiently long period for  $\phi$ , thereby validating Assumption 1. Luckily, this is indeed the case for silicon photonic modulators based on the free-carrier plasma dispersion effect [37], [40]. Moreover, if a more accurate model is desired when a photonic system is integrated with a high-frequency electronic system, e.g., millimeter-wave transceiver, **Assumption 1 can be entirely eliminated**, because Eq. (10) can be extended to consider non-negligible  $\phi$  transient time if we include terms such as  $\{v(t - \Delta t), v(t - 2\Delta t), \dots\}$  in a discretized formulation or  $\{\dot{v}(t), \ddot{v}(t), \dots\}$  in a continuous formulation. For the purposes of demonstrating SPIPE, we stick with Assumption 1 and use Eq. (10).

We emphasize that the updated modulator models presented in Eq. (9) incorporate the time axis, representing the second extension we made to the standard Scattering matrix simulation. For other passive photonic components (e.g., waveguides), we continue to use Eqs. (4) and (7). Unlike modulators whose Scattering matrices need to be re-evaluated at every time step, their Scattering matrices do not depend on time. Despite this extension, the solution technique (i.e., formulating as  $\mathbf{Ax} = \mathbf{b}$ ) discussed in Section III-B still applies.

*b) Photodetectors:* The modulators discussed above convert electrical signals into optical signals, while photodetectors (PDs) perform the reverse conversion, transforming optical signals into electrical signals. Let us denote the analytical optical signal which will be injected into a PD (e.g., at node n6 or n7 in Figure 2) as  $p(\omega, t)$ , which has been calculated by Scattering matrix simulation. Again, we re-emphasize that because we have incorporated the time axis in the Scattering matrix simulation, so  $p(\omega, t)$  depends on both  $\omega$  and  $t$ . For a pure photonic circuit only containing passive components,  $p(\omega, t)$  will reduce to  $p(\omega)$ .

Recall that  $p(\omega, t)$  represents the magnitude of the EM mode at frequency channel with angular frequency equal to  $\omega$  and at time  $t$ . It is physically grounded that the optical power associated with such an analytical optical signal is proportional to  $|p(\omega, t)|^2$ , i.e.,  $P \propto |p(\omega, t)|^2$ . The output photocurrent of PDs is well established and is given by:

$$I_{\text{pd}} = \sum_{n=1}^{N_c} R_n \times P_n \propto \sum_{n=1}^{N_c} R_n \times |p(\omega_n, t)|^2, \quad (11)$$

where  $N_c \in \mathbb{Z}^+$  is the number of frequency channels,  $R_n$  is the responsivity associated with the  $n$ -th channel (unit: A/W), and  $P_n$  is the optical power of the  $n$ -th channel received by the PD (unit: W). Thus, with a slight abuse of the notation, we can define:

$$I_{\text{pd}}(t) = \sum_{n=1}^{N_c} R(\omega_n) \times |p(\omega_n, t)|^2. \quad (12)$$

Note that  $|p(\omega_n, t)|^2$  is unitless, and  $R(\omega_n)$  has unit Watts. It can be understood that the proportionality constant in the expression  $P_n \propto |p(\omega_n, t)|^2$  is absorbed into  $R_n$ , yielding  $R(\omega_n)$ .

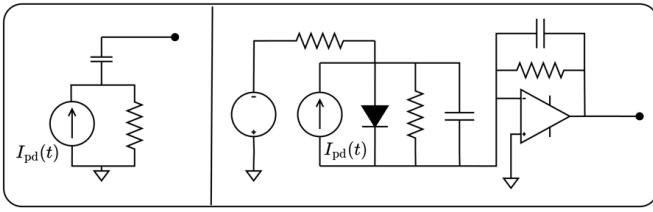


Fig. 4: Two PD-equivalent source circuit models are shown. The left shows a simple model given in [41], while the right depicts a more detailed model.

The generated photocurrent acts as an independent current source that drives the electronic circuit connected to the PD. Ideally, the electronic equivalent circuit model of a PD would include only this photocurrent source. However, in practice, additional elements such as source resistance and capacitance need to be considered, as illustrated on the left side of Figure 4. Furthermore, a trans-impedance amplifier (TIA) circuit may be required to amplify the electrical signal, as depicted on the right side of Figure 4. SPIPE has a few built-in electronic equivalent source circuit models for PDs and also allows users to define new ones. In the following subsection, we will integrate these elements and present the overall simulation methodology of SPIPE.

#### D. SPIPE Flow

Algorithm 1 summarizes the main steps of SPIPE. In a photonic-electronic co-integrated system, it is natural for the modulator driving voltage to influence the photocurrent. For example, in the PTC schematic shown in Figure 2, the voltage at m1 affects the photocurrent output of PD1. However, the reverse impact described in Definition 1 is relatively uncommon. Before delving deeper, it is essential to formally define this reverse impact as **feedback** in our context.

**Definition 1.** *Feedback occurs if any photocurrent  $I_{\text{pd},i}(t)$  impacts any modulator driving voltage  $v_j(t)$ .*

Based on this definition, an immediate corollary is that for a circuit without feedback, Step 6 in Algorithm 1 will always yield the same  $\mathbf{V}_{\text{new}}$ , regardless of  $\mathbf{I}$ . More importantly, this implies that Algorithm 1 requires only two iterations to simulate a photonic-electronic circuit without feedback.

To better understand Algorithm 1, we provide intermediate schematics corresponding to the circuit shown in Figure 2

#### Algorithm 1 SPIPE Algorithm

- Require:** An electronic-photonic circuit described with SPIPE syntax
- Ensure:** Optical signals, node voltages, and branch currents  
Electronic circuit  $\mathcal{E}$ , photonic circuit  $\mathcal{P}$
- 1: Identify where modulators and photodetectors present, and denote their numbers as  $N_{\text{md}}$  and  $N_{\text{pd}}$ , respectively.
  - 2: Cut the connections between  $\mathcal{E}$  and  $\mathcal{P}$  at all modulator locations. Append modulator-equivalent load circuit models to  $\mathcal{E}$ .
  - 3: Cut the connections between  $\mathcal{E}$  and  $\mathcal{P}$  at all PD locations. Append PD-equivalent source circuit models to  $\mathcal{E}$ .
  - 4: Initialize the driving voltages for the modulators  $\mathbf{V} = \{v_i(t)\}_{i=1}^{N_{\text{md}}}$  and the photocurrents  $\mathbf{I} = \{I_{\text{pd},i}(t)\}_{i=1}^{N_{\text{pd}}}$ .
  - 5: **while** convergence not reached **do**
  - 6:     Simulate  $\mathcal{E}$  with photocurrents  $\mathbf{I}$  using SPICE, yielding updated voltages  $\mathbf{V}_{\text{new}}$ .
  - 7:     Simulate  $\mathcal{P}$  with the updated voltages  $\mathbf{V}_{\text{new}}$  using Scattering matrix simulation, yielding updated photocurrents  $\mathbf{I}_{\text{new}}$ .
  - 8:     Update  $\mathbf{V} \leftarrow \mathbf{V}_{\text{new}}$  and  $\mathbf{I} \leftarrow \mathbf{I}_{\text{new}}$ .

during the execution of SPIPE, following the steps outlined in Algorithm 1. These intermediate steps are illustrated in Figure 5. The modulator-equivalent load circuits are highlighted within the dashed orange blocks, while the PD-equivalent source circuits are shown within the dashed green blocks. Here, the equivalent circuits have been simplified for illustration purposes, and meanwhile, we note that SPIPE also allows users to define their own equivalent circuits. It is evident from Figure 2 that the circuit does not exhibit the feedback defined previously, which is also reflected in Figure 5, where the voltages  $\{v_1, v_2\}$  do not depend on  $\{I_{\text{pd},1}, I_{\text{pd},2}\}$ .

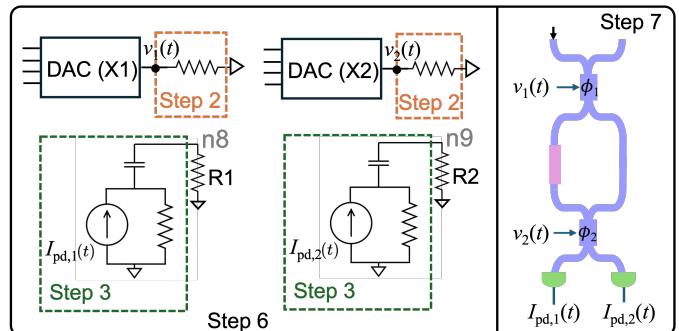


Fig. 5: Intermediate steps when simulating the circuit shown in Figure 2 using SPIPE. This circuit doesn't have feedback, i.e., the electronic simulation output  $\{v_1, v_2\}$  does not depend on  $\{I_{\text{pd},1}, I_{\text{pd},2}\}$ . SPIPE allows users to define customized equivalent circuits in the dashed orange and green blocks.

Finally, it is worth noting that most photonic-electronic circuits existing today do not exhibit feedback, making Algorithm 1 sufficient. However, we proactively look into the future and have extended SPIPE for circuits with feedback. Let us denote the SPICE simulation in Step 6 as  $g(\cdot)$  and the

Scattering matrix simulation in Step 7 as  $f(\cdot)$ . Essentially, we are looking for the solution pair  $(\mathbf{I}, \mathbf{V})$  such that  $g(\mathbf{I}) = \mathbf{V}$  and  $f(\mathbf{V}) = \mathbf{I}$ , which can be formulated as solving:

$$\min_{\mathbf{V}, \mathbf{I}} L = \|g(\mathbf{I}) - \mathbf{V}\|^2 + \|f(\mathbf{V}) - \mathbf{I}\|^2. \quad (13)$$

Since this is a physical system, we know the global minimum  $L = 0$  is attainable. Algorithm 1 can be regarded as one way to solve Eq. (13) which works well when  $g(\cdot)$  is a constant (i.e.,  $g(\mathbf{I}_1) = g(\mathbf{I}_2)$  for any  $\mathbf{I}_1$  and  $\mathbf{I}_2$ , representing a circuit without feedback). When feedback exists, the iterative approach in Algorithm 1 might be inefficient. To address such cases, SPIPE can enable Xyce backend, which supports adjoint sensitivity calculating the derivative of  $g(\cdot)$ . Moreover, SPIPE implements a differentiable Scattering matrix so that the derivatives of  $f(\cdot)$  are also available. Thus, a gradient-descent optimization routine can be used to solve Eq. (13).

We conclude with several important remarks regarding SPIPE. First, SPIPE uses transistor-level models for electronic components and Scattering matrices for photonic components. These modeling abstractions align with industry standards, as they are the same representations used by foundries in process design kits (PDKs). As such, once PDKs are provided, SPIPE can be directly applied to simulate electronic-photonic systems without requiring additional model calibration. In our paper, the device parameters are set to their typical values following previous literature. Second, the availability of gradient information in SPIPE paves the way for end-to-end design optimization of electronic-photonic systems—an exciting direction we leave for future exploration. Third, as illustrated in Figure 3, basic familiarity with SPICE syntax and relevant simulation parameters (e.g., effective mode index, frequency range of interest) is sufficient for users to begin utilizing SPIPE.

#### IV. NUMERICAL RESULTS

We implement SPIPE in Python, integrating subroutine calls to HSPICE and Xyce for electronic circuit simulation. A few equivalent circuit models for modulators and PDs are implemented with the ‘.subckt’ syntax in SPIPE, which can be parsed by HSPICE or Xyce. For photonic circuit simulation, we develop a differentiable Scattering matrix simulation in PyTorch and implement the Scattering matrices of key photonic components used in our experiments. Specifically, given an electronic-photonic circuit in SPIPE, we first parse it into two sub-netlist files: one for the electronic part and one for the photonic part. This parsing process involves string manipulation and is efficient, as SPIPE adopts a SPICE-like syntax. At a higher level, a Python system program orchestrates the co-simulation. Specifically, it invokes HSPICE or Xyce on the electronic netlist using the provided photocurrent values, and retrieves the resulting modulator driving voltages. These voltages are then passed to our customized differentiable Scattering matrix simulator to compute the updated photocurrents. The data exchange between the electronic (HSPICE/Xyce) and photonic (Scattering matrix) simulators occurs in Python using PyTorch tensors. All numerical simulations are conducted on a Linux server equipped with an Intel Xeon E5 CPU.

#### A. Programmable Photonic Mesh

The programmable photonic mesh [7] is a recently developed reconfigurable computing paradigm capable of realizing various optical functions at run-time, which is also referred to by some as a photonic FPGA. A programmable photonic mesh consists of basic units connected in a specific topology, most commonly square, triangular, or hexagonal [7]. In this section, we begin with its passive version to verify SPIPE. Specifically, each basic unit comprises two directional couplers (DCs) and two phase shifters, which are fixed after fabrication in the passive version, as illustrated in Figure 6.

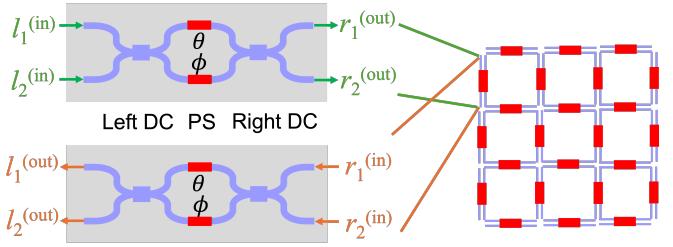


Fig. 6: Left: A passive basic unit is made up of a left and a right directional coupler (DC) and two phase shifters. Right: An example of 3-by-3 square mesh, made up of connecting basic units (highlighted by red rectangles) in square cells.

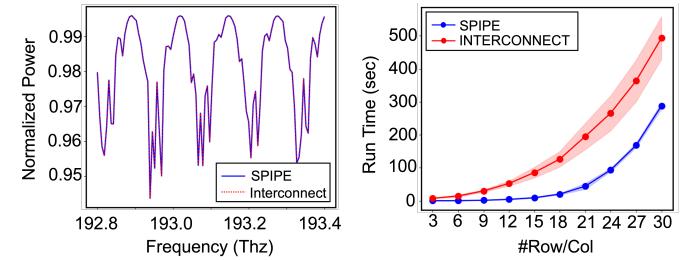


Fig. 7: Left: Frequency response obtained by SPIPE and Lumerical INTERCONNECT on the same 30-by-30 square mesh under identical conditions (e.g., same coupling ratios, measured port, and EM mode effective index). Right: Runtime comparison between Lumerical INTERCONNECT and SPIPE versus the number of rows (equal to columns) in a square mesh, averaged over 10 simulations. Shaded regions indicate the range of  $\pm 3$  standard deviations.

We create a custom script that generates square meshes of arbitrary size and randomly initializes the coupling ratios of the DCs. We then use SPIPE and a commercial-standard photonic circuit simulator, Lumerical INTERCONNECT, to simulate these meshes. It is worth noting that in current silicon photonic hardware implementations, mesh sizes are typically fewer than 10 [7]. The largest mesh size used in our experiments, a 30-by-30 configuration, is therefore already forward-looking from a scalability perspective and serves as a meaningful benchmark for evaluating SPIPE. Moreover, SPIPE currently employs only the direct method (i.e., matrix inversion) to solve the system-level Scattering matrix. For very large mesh sizes, iterative methods are

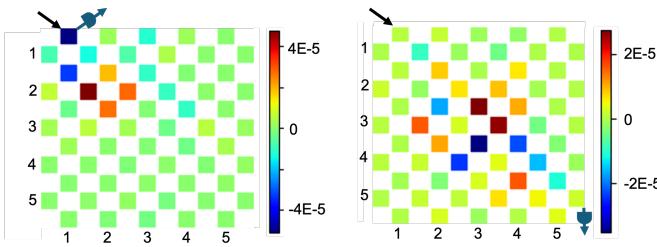


Fig. 8: The gradient of the photocurrent with respect to the driving voltage of each modulator in a basic unit is computed by SPIPE in a 5-by-5 mesh and visualized as a heatmap. Each colored square represents a basic unit, oriented either vertically or horizontally. The optical input port is indicated by a black arrow, while the measured photocurrent is marked by a blue arrow. The left and right panels are two different cases.

TABLE I: SPIPE simulation MSE difference and runtime against Lumerical INTERCONNECT and gradient error.

#Row/Col	#Device	Sim. Err	Runtime	Grad. Err
3	144	2.7E-10	85.8×	2.22E-7
6	504	5.6E-10	34.5×	2.20E-7
9	1080	3.2E-9	20.8×	7.38E-8
12	1872	1.7E-9	12.8×	7.13E-8
15	2880	3.4E-9	9.3×	1.30E-7
18	4104	6.2E-9	5.7×	1.03E-7
21	5544	1.2E-9	4.4×	8.94E-8
24	7200	2.4E-9	2.7×	9.05E-8
27	9072	3.9E-9	2.2×	1.84E-7
30	11160	6.9E-9	2.0×	1.79E-7

\* Here  $\#Device = 12 \cdot (\#Row + 1) \cdot (\#Row)$ . Gradient error is calculated by comparing to that obtained via numerical perturbation.

expected to offer greater efficiency, and we consider this an important direction for future work. Figure 7 demonstrates that with the current implementations, SPIPE produces results consistent with Lumerical INTERCONNECT while achieving a significantly shorter run-time. The third column in Table I further reports the quantitative simulation error when comparing SPIPE with Lumerical INTERCONNECT.

Next, we restore the passive DCs to active modulators, reverting to the original programmable photonic mesh proposed in [7]. The driving voltages on the active modulators are randomly initialized. We further add a PD to the mesh and then use SPIPE to compute the derivative of its photocurrent with respect to the driving voltages applied to all modulators (i.e., the derivative of  $f(\cdot)$  in Eq. (13)). Results are shown in the last column of Table I. In a 3-by-3 mesh with 144 active modulators, the derivative is a 144-dimensional vector, differing from numerical perturbation by a mean absolute error of 2.2E-7. Figure 8 further shows the gradient map calculated by SPIPE using a 5-by-5 mesh as an example. Take the left part of Figure 8 as an example, the optical input and output ports are positioned at the top left of the mesh. The gradient heatmap aligns with expectations — basic units farther from

the input and output ports exert less influence on the output photocurrent, resulting in lower gradient values. Moreover, the heatmap shows gradient absolute values on the order of 1E-5, while the error of gradient calculation is on the order of 1E-7 as reported in Table I, which is 1% of the gradient magnitude, justifying the accuracy of SPIPE's gradient calculation. Moreover, we evaluate the gradient computation time and find that it typically accounts for 70%–80% of the total simulation time. For example, given  $\#row = 3$  and 6, the proportion of time spent on gradient computation is 73.4% and 81.9%, respectively.

### B. Photonic-Electronic Co-Integrated System

In this subsection, we consider photonic-electronic co-integrated systems. We mainly compare SPIPE to analytical solutions, reported experimental data in real-world measurements, and Verilog-A implementation.

```

`include "disciplines.vams"
`include "my_discipline.vams" // define optical discipline

module wg(in_real, in_imag, out_real, out_imag);
    input in_real, in_imag;
    output out_real, out_imag;
    optical in_real, in_imag;
    optical out_real, out_imag;

    parameter real pi = 3.14159265359; // pi
    parameter real length = 100e-6; // waveguide length
    parameter real wl = 1550e-9; // wavelength
    parameter real neff = 3.4; // effective mode index
    parameter real c = 3e8; // light speed
    parameter real theta= -2 * pi * neff * length / c / wl;
    parameter real alpha = 1.0; // transmission loss
    analog begin
        E(out_real) <+ alpha * (E(in_real) * cos(theta)
                                - E(in_imag) * sin(theta));
        E(out_imag) <+ alpha * (E(in_real) * sin(theta)
                                + E(in_imag) * cos(theta));
    end
endmodule

```

Fig. 9: A simplified Verilog-A implementation of a waveguide. Since Verilog-A does not support complex numbers, the real and imaginary parts must be handled separately.

a) **PTC Proposed in NetCast and LT**: We now consider photonic-electronic co-integrated circuits, beginning with the circuit shown in Figure 2 and 3. An 8-bit DAC is designed using the open-source Skywater 130nm PDK, operating at approximately 0.1 Gsps (equivalent to 10 ns per sample) in SPICE. As described in NetCast [38], the difference between the outputs at nodes n8 and n9 in Figure 2 is proportional to the product of the DAC outputs X1 and X2, which can be calculated analytically.

We perform both Verilog-A and SPIPE simulations on this circuit. Inspired by [17], we implement key photonic components, including waveguides, modulators, photodetectors, and electronic DACs, in Verilog-A. A simplified waveguide Verilog-A module is shown in Figure 9. The Verilog-A modules are simulated using Cadence Spectre. The Verilog-A simulation completes within 10 seconds, whereas our SPIPE takes approximately 79.4 seconds (see Table II for a detailed runtime

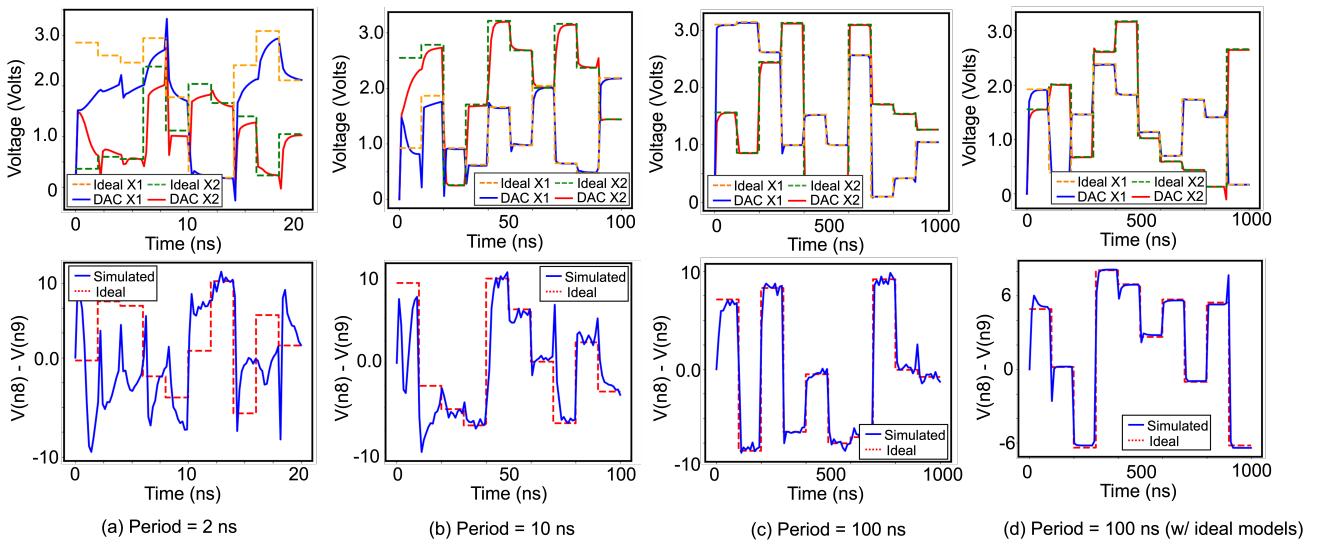


Fig. 10: Top row: Comparison of simulated results from SPIPE with ideal values at the DAC output ports. Bottom row: The voltage difference at output nodes n8 and n9 (solid blue line) is proportional to the product of DAC outputs X1 and X2 (red dashed line). The discrepancy between the simulated and ideal results arises from non-idealities (e.g., electronic circuit transient response, PD noise) that are accounted for in SPIPE but not in the ideal model. Comparing (a)–(c), we observe that the transient behavior of the DACs affects the final output. Since we use an open-source 130nm PDK, the RC time constant of the designed DAC is in the range of several nanoseconds.

breakdown). It is important to emphasize that this difference is expected, as Verilog-A is a behavioral modeling approach that operates at a higher level of abstraction compared to SPICE, making it significantly faster. For instance, modeling an 8-bit DAC in SPICE requires numerous MOS transistors, whereas in Verilog-A, it can be reduced to a single line: `v(out) <+ (vmax - vmin) * discrete + vmin`, in the simplest case. However, our goal is to develop a SPICE-level photonic-electronic co-simulator, where longer simulation time is an inherent inevitable trade-off for higher accuracy.

The simulation results of SPIPE are presented in Figure 10, comparing them with analytical results. In Figures 10 (a) to (c), we progressively increase the DAC input signal period from 2 ns to 100 ns. It is evident that as the signal period increases, the outputs at both the DACs and nodes n8 and n9 align more closely with their ideal values, confirming that the transient behavior of the DACs significantly impacts the outputs. Furthermore, when comparing Figures 10 (c) and (d), we observe that the discrepancy between the SPIPE simulation and the ideal values does not arise from systematic errors in SPIPE, but rather from the consideration of non-ideal models in SPIPE. The simulation results of Verilog-A are omitted, as they are not our primary focus. However, we observe that in the ideal case, the Verilog-A simulation output aligns with the analytical response, similar to Figure 9 (d).

More importantly, SPIPE offers a ‘cycle-accurate’ power simulation for PTC (and any photonic-electronic circuits), which represents a crucial step in verifying the promised PTC potential — low power consumption. Note that PTC itself doesn’t have a clock cycle, while the term ‘cycle’ refers to that of DACs, which deliver operands to the PTC in sync with clock cycles. The power consumption over time is plotted

in Figure 11. The photonic power consumption (red line) primarily comes from the optical laser source and a tiny portion of optical loss. The red, green, and orange lines show minimal variation over time, which is barely visible compared to the blue curve. Summing all four lines, we estimate that the PTC consumes between 70 mW~140 mW, aligning perfectly with the hardware results reported in [43].

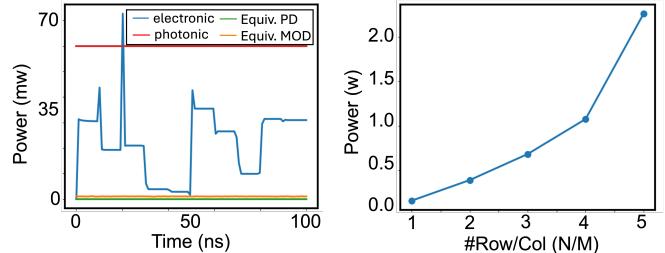


Fig. 11: Left: Power is plotted along time in NetCast PTC. ‘Equiv.’ is short for ‘Equivalent’. Right: In LT [42], power averaged in time is plotted along the number of rows/columns.

We further explore the extended PTC design proposed in LT [42]. The  $N$ -by- $M$  PTC in [42], with  $N_c$  wavelength channels, includes  $NM$  DDots (compound passive photonic units) and  $(N + M)N_c$  DACs. In our experiment, we set  $N = M$  and choose  $N_c = 4$ , reusing DACs from the 130nm process. The simulation results at DDots’ outputs are similar to Figure 10, and are omitted. More importantly, we evaluate PTC scalability by plotting power consumption versus the number of rows/columns in Figure 11 and analyzing the runtime breakdown of SPIPE in Table II. The power estimates align with [42], and the runtime results indicate that HSPICE simulation for electronics is the main bottleneck.

TABLE II: Number of modulators in different PTC experiments and the run-time (seconds) breakdown. <sup>†</sup>Because Verilog-A does behavioral modeling, it is expected to run much faster than SPIPE. However, our goal is to develop a SPICE-level photonic-electronic co-simulator, where longer simulation time is an inherent inevitable trade-off for higher accuracy.

# MODs	SPIPE			†Convert to Verilog-A [17]
	HSPICE	S-Matrix	Overall	
Figure 2 [38]	2	75.1	0.02	79.4
1 × 1 PTC [42]	8	267.1	0.2	267.5
3 × 3 PTC [42]	24	797.9	0.5	798.5
5 × 5 PTC [42]	40	876.1	0.9	877.7

*b) Optical Phased Array (OPA):* As shown in Fig. 12, a 1D OPA [44] consists of  $N$  optical emitting elements arranged in a linear configuration, with each element spaced by a fixed distance  $d$ . The relative phase difference between the  $n$ -th element and the first element, denoted as  $\phi_n$ , determines the interference pattern of the emitted light, which in turn dictates the direction of the output beam. The array factor (AF), which describes the far-field radiation pattern, is given by:  $AF(\theta) = |\sum_{n=0}^{N-1} \exp(j(nkd \cos \theta + \phi_n))|$ , where  $k = \frac{2\pi}{\lambda}$  represents the wave number. The phase difference  $\phi_n$  may result from unintended factors, such as waveguide imbalances, or from deliberate phase tuning via active modulators driven by electronic circuits.

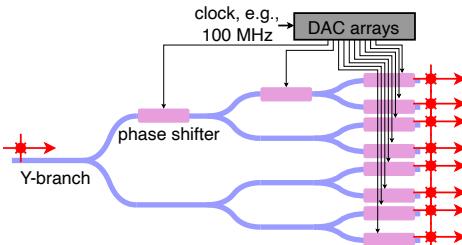


Fig. 12: A schematic of a 1D OPA made up of a 3-column Y-splitter tree. The active modulators are highlighted in pink.

We simulate a 1D OPA composed of a 3-column photonic Y-branch splitting tree (a total of 7 Y-branches), with active modulators on the upper arms of the Y-branches, driven by DACs. The input signals at all 7 DACs are identical, and the ideal DAC output, along with the real simulated output, is shown on the left of Figure 13. The AF plot is shown on the right. It displays beam steering at different angles about every 10 ns, consistent with the DAC clock cycle. This example demonstrates the effectiveness of SPIPE in end-to-end modeling of the temporal behaviors of OPA under the influence of electronic driver circuits.

## V. CONCLUSIONS

In this paper, we present SPIPE, a differentiable SPICE-level co-simulation program for integrated photonics and electronics. SPIPE offers unique advantages: (i) a low usage barrier for defining photonic-electronic circuits, enabled by an extended SPICE syntax, (ii) full differentiability effectively handling photonic-electronic circuits with feedback, and (iii) no model conversion problem. SPIPE delivers comparable

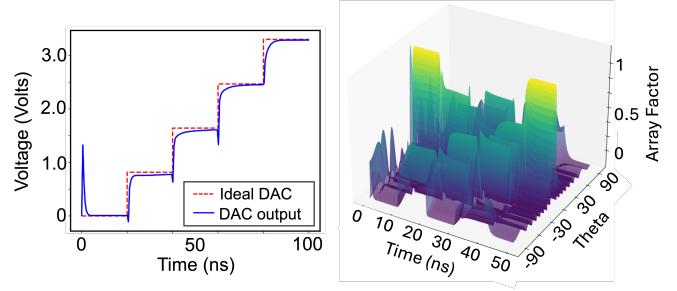


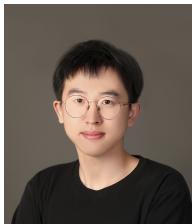
Fig. 13: Left: The ideal DAC output and the real simulated output. Right: AF is plotted along time and angle  $\theta$ .

simulation accuracy to analytical solutions, real-world experimental data, and existing simulators (where existing ones are applicable) while significantly improving efficiency by  $2\sim 85\times$ . SPIPE enables direct interactions between photonic and electronic components while preserving established simulation principles of each domain — SPICE for electronics and Scattering matrix for photonics. SPIPE lowers the barrier to photonic-electronic co-simulation and provides a solid foundation for co-design.

## REFERENCES

- [1] R. Soref, "The past, present, and future of silicon photonics," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 12, no. 6, pp. 1678–1687, 2006.
- [2] M. J. R. Heck, H.-W. Chen, A. W. Fang, B. R. Koch, D. Liang, H. Park, M. N. Sysak, and J. E. Bowers, "Hybrid silicon photonics for optical interconnects," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 17, no. 2, pp. 333–346, 2011.
- [3] N. Jouppi, G. Kurian, S. Li, P. Ma, R. Nagarajan, L. Nai, N. Patil, S. Subramanian, A. Swing, B. Towles *et al.*, "Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–14.
- [4] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Laroche, D. Englund *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature photonics*, vol. 11, no. 7, pp. 441–446, 2017.
- [5] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, "Photonics for artificial intelligence and neuromorphic computing," *Nature Photonics*, vol. 15, no. 2, pp. 102–114, 2021.
- [6] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja *et al.*, "Parallel convolutional processing using an integrated photonic tensor core," *Nature*, vol. 589, no. 7840, pp. 52–58, 2021.
- [7] W. Bogaerts, D. Pérez, J. Capmany, D. A. Miller, J. Poon, D. Englund, F. Morichetti, and A. Melloni, "Programmable photonic circuits," *Nature*, vol. 586, no. 7828, pp. 207–216, 2020.

- [8] Z. Gao, X. Chen, Z. Zhang, U. Chakraborty, W. Bogaerts, and D. S. Boning, "Automatic synthesis of light-processing functions for programmable photonics: theory and realization," *Photonics Research*, vol. 11, no. 4, pp. 643–658, 2023.
- [9] X. Zhang, K. Kwon, J. Henriksson, J. Luo, and M. C. Wu, "A large-scale microelectromechanical-systems-based silicon photonics lidar," *Nature*, vol. 603, no. 7900, pp. 253–258, 2022.
- [10] C. V. Poulton, A. Yaacobi, D. B. Cole, M. J. Byrd, M. Raval, D. Vermeulen, and M. R. Watts, "Coherent solid-state lidar with silicon photonic optical phased arrays," *Optics letters*, vol. 42, no. 20, pp. 4091–4094, 2017.
- [11] C. Taballione, T. A. Wolterink, J. Lugani, A. Eckstein, B. A. Bell, R. Grootjans, I. Visscher, J. J. Renema, D. Geskus, C. G. Roeloffzen *et al.*, "8×8 programmable quantum photonic processor based on silicon nitride waveguides," in *Frontiers in Optics*. Optical Society of America, 2018, pp. JTu3A-58.
- [12] E. Luan, H. Shoman, D. M. Ratner, K. C. Cheung, and L. Chrostowski, "Silicon photonic biosensors using label-free detection," *Sensors*, vol. 18, no. 10, p. 3519, 2018.
- [13] S. Palnitkar, *Verilog HDL: a guide to digital design and synthesis*. Prentice Hall Professional, 2003, vol. 1.
- [14] L. W. Nagel, "Spice2: A computer program to simulate semiconductor circuits," *College of Engineering, University of California, Berkeley*, 1975.
- [15] J. A. Kong, "Theory of electromagnetic waves," *New York*, 1975.
- [16] Lueda Photonics, "Scatter matrix models in caphe," 2025, accessed: 2025-02-28. [Online]. Available: [https://academy.lucedaphotonics.com/ikpiss/guides/caphe\\_models\\_frequency/scatter\\_matrix](https://academy.lucedaphotonics.com/ikpiss/guides/caphe_models_frequency/scatter_matrix)
- [17] C. Sorace-Agaskar, J. Leu, M. R. Watts, and V. Stojanovic, "Electro-optical co-simulation for integrated cmos photonic circuits with verilog-a," *Optics express*, vol. 23, no. 21, pp. 27180–27203, 2015.
- [18] H. Fang, Y. Liu, L. Wu, X. Xu, L. Du, C. Sun, and J. Zhao, "Accurate time-domain and frequency-domain co-simulation approach for oeics design with verilog-a," *Optics Express*, vol. 32, no. 2, pp. 1764–1775, 2024.
- [19] K. Kawahara and T. Baba, "Electro-optic co-simulation in high-speed silicon photonics transceiver design using standard electronic circuit simulator," 2024. [Online]. Available: <https://arxiv.org/abs/2410.02282>
- [20] Y. Liu, B. Hu, Z. Liu, P. Chen, L. Du, J. Liu, X. Li, W. Zhang, and J. Xu, "Fiona: Photonic-electronic cosimulation framework and transferable prototyping for photonic accelerator," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 1–9.
- [21] D. Ming, Y. Wang, Z. Wang, K. X. Wang, C. Qiu, and M. Tan, "Ephic models: General spice photonic models for closed-loop electronic-photonic co-simulation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 4, pp. 1819–1831, 2024.
- [22] D. A. Q. Dominguez, "An Electronic-Photonic Co-Design Approach with piel," 2024, accessed: 2024-11-01. [Online]. Available: [https://piel.readthedocs.io/en/latest/resources/fsic\\_2024\\_presentation/index.html](https://piel.readthedocs.io/en/latest/resources/fsic_2024_presentation/index.html)
- [23] T. Smy and P. Gunupudi, "Robust simulation of opto-electronic systems by alternating complex envelope representations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 1139–1143, 2012.
- [24] T. Smy, P. Gunupudi, S. McGarry, and W. N. Ye, "Circuit-level transient simulation of configurable ring resonators using physical models," *Journal of the Optical Society of America B*, vol. 28, no. 6, pp. 1534–1543, 2011.
- [25] P. Gunupudi, T. Smy, J. Klein, and Z. J. Jakubczyk, "Self-consistent simulation of opto-electronic circuits using a modified nodal analysis formulation," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 4, pp. 979–993, 2010.
- [26] Y. Ye, T. Ullrick, W. Bogaerts, T. Dhaene, and D. Spina, "Spice-compatible equivalent circuit models for accurate time-domain simulations of passive photonic integrated circuits," *Journal of Lightwave Technology*, vol. 40, no. 24, pp. 7856–7868, 2022.
- [27] J. Li, D. Ahsanullah, Z. Gao, and R. Rohrer, "Circuit theory of time domain adjoint sensitivity," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 7, pp. 2303–2316, 2023.
- [28] S. Director and R. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Transactions on Circuit Theory*, vol. 16, no. 3, pp. 318–323, 1969.
- [29] A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. D. Joannopoulos, and S. G. Johnson, "Meep: A flexible free-software package for electromagnetic simulations by the fDTD method," *Computer Physics Communications*, vol. 181, no. 3, pp. 687–702, 2010.
- [30] J. Gu, H. Zhu, C. Feng, Z. Jiang, R. T. Chen, and D. Z. Pan, "L2light: Enabling on-chip learning for optical neural networks via efficient in-situ subspace optimization," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [31] X. Chen, Z. Wang, Y.-S. Chang, J. Xu, J. Feng, P. Yang, Z. Wang, and L. H. Duong, "Modeling and analysis of optical modulators based on free-carrier plasma dispersion effect," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 5, pp. 977–990, 2019.
- [32] S. Grivet-Talocia and B. Gustavsen, *Passive macromodeling: Theory and applications*. John Wiley & Sons, 2015.
- [33] L. Nagel and R. Rohrer, "Computer analysis of nonlinear circuits, excluding radiation (cancer)," *IEEE Journal of Solid-State Circuits*, vol. 6, no. 4, pp. 166–182, 1971.
- [34] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.
- [35] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in neural information processing systems*, vol. 31, 2018.
- [36] L. S. Pontryagin, *Mathematical theory of optimal processes*. Routledge, 2018.
- [37] N. C. Harris, Y. Ma, J. Mower, T. Baehr-Jones, D. Englund, M. Hochberg, and C. Galland, "Efficient, compact and low loss thermo-optic phase shifter in silicon," *Optics express*, vol. 22, no. 9, pp. 10487–10493, 2014.
- [38] R. Hamerly, A. Sludds, S. Bandyopadhyay, Z. Chen, Z. Zhong, L. Bernstein, and D. Englund, "Netcast: low-power edge computing with wdm-defined optical neural networks," *arXiv preprint arXiv:2207.01777*, 2022.
- [39] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, no. 15, p. 510, 2008.
- [40] J. Witzens, "High-speed silicon photonics modulators," *Proceedings of the IEEE*, vol. 106, no. 12, pp. 2158–2182, 2018.
- [41] L. Chrostowski and M. Hochberg, *Silicon photonics design: from devices to systems*. Cambridge University Press, 2015.
- [42] H. Zhu, J. Gu, H. Wang, Z. Jiang, Z. Zhang, R. Tang, C. Feng, S. Han, R. T. Chen, and D. Z. Pan, "Lightening-transformer: A dynamically-operated optically-interconnected photonic transformer accelerator," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024, pp. 686–703.
- [43] Z. Zhong, M. Yang, J. Lang, C. Williams, L. Kronman, A. Sludds, H. Esfahanizadeh, D. Englund, and M. Ghobadi, "Lightning: A reconfigurable photonic-electronic smartnic for fast and energy-efficient inference," in *Proceedings of the ACM SIGCOMM 2023 Conference*, 2023, pp. 452–472.
- [44] P. F. McManamon, T. A. Dorschner, D. L. Corkum, L. J. Friedman, D. S. Hobbs, M. Holz, S. Liberman, H. Q. Nguyen, D. P. Resler, R. C. Sharp *et al.*, "Optical phased array technology," *Proceedings of the IEEE*, vol. 84, no. 2, pp. 268–298, 1996.



**Zhengqi Gao** is currently pursuing his Ph.D. in the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. He received his M.S. and B.E. degrees from the School of Microelectronics at Fudan University in 2018 and 2021, respectively. His research focuses on design automation for electronic and photonic integrated circuits, novel computing paradigms, and applied machine learning. His work has been presented at leading venues in electronic design automation, photonics, and machine learning, such as ICCAD, DAC, IEEE JLT, Optica Photonics Research, NeurIPS, ICLR, and ECCV. He interned at Baidu, Shanghai Qizhi AI Institute, Nvidia Research, and Apple. He has received several honors, including being named an ML and Systems Rising Star in 2024, securing second place at the CVPR'23 TTM Challenge, receiving Editor's Highlights from both Optica Express in 2024 and Photonics Research in 2023, and having a spotlight paper (top 5%) at ICLR 2023. He has also been awarded the Biren Scholarship and the Integrated Circuits Scholarship, both in 2020, among other recognitions.



**Jiaqi Gu** (Member, IEEE) received the B.E. degree in Microelectronic Science and Engineering from Fudan University, Shanghai, China in 2018. He is currently a post-graduate student studying for his Ph.D. degree in the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA. He is an Assistant Professor in the School of Electrical, Computer and Energy Engineering at Arizona State University (ASU), Tempe, AZ, USA. His current research interests include machine learning, efficient algorithms and architecture design for high-performance AI, next-generation AI computing with emerging technology, and electronic-photonic design automation. He has published over 90 journal articles and refereed conference papers in the related field. He served as technical reviewers for over 18+ international journals/conferences, such as IEEE TCAD, TODAES, DAC, ICCAD, ISVLSI, GLVLSI, IEEE TNNLS, NeurIPS, CVPR, ICML, ICCV, ECCV, AAAI, IROS, Nature Communications, Science Advances, IEEE JSTQE, APL, IEEE PTL, etc. He also served as the Technical Program Committee member of DAC, ICCAD, ICCD, AAAI, CLEO, etc. He has received the Best Paper Award at ASP-DAC 2020, the Best Paper Finalist at DAC 2020, the Best Poster Award at NSF Workshop on Machine Learning Hardware (2020), the ACM/SIGDA Student Research Competition First Place (2020), and the ACM Student Research Competition Grand Finals First Place (2021), the Best Paper Award at IEEE TCAD 2021, won the Robert S. Hilbert Memorial Optical Design Competition 2022, won Margarida Jacome Dissertation Prize at UT Austin ECE (2023), Graduate School Outstanding Dissertation Award at UT Austin (2024), the Best Paper Award Candidate at DATE 2025, and the NVIDIA Academic Grant Program Award (2025).



**Ron Rohrer** (Life Fellow, IEEE) received the S.B. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1960, and the M.S. and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA, in 1961 and 1963, respectively. He is currently Professor Emeritus of Electrical and Computer Engineering at Carnegie Mellon University.

He has been awarded six patents and has authored five textbooks and over 100 technical papers. Dr. Rohrer's many awards include Alexander Von Humboldt Senior Fellow at RWTH Aachen from 1972 to 1973; the ASEE 1979 Terman Award; the IEEE CAS 1990 Van Valkenburg Award; the IEEE 1993 Education Medal "for innovation in bringing electrical engineering practice into the classroom and merging academic research with industrial need;" the 1996 NEC Computer and Communication Prize, a worldwide honor for pioneering contributions in electronics; the 2002 Kaufman Award, presented by the EDA Consortium; the IEEE CAS 2009 Belevitch Award; and the IEEE 2012 Kirchhoff Award. He was the Founding Editor of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems from 1981 to 1984 and served as the President of the IEEE Circuits and Systems Society in 1987. In 1989, he was inducted into the National Academy of Engineering "for contributions to circuit simulation that have enabled deep submicron IC design." He was the Elected Fellow of the IEEE in 1980 "for theoretical contributions and practical software for computer-aided circuit design."



**Luca Daniel** (Fellow, IEEE) is currently a Professor of electrical engineering and computer science with the Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include development of numerical techniques related to uncertainty quantification, inverse problems, robust optimization, parameterized model order reduction, integral equation solvers, evaluating and improving robustness of deep neural networks and of magnetic resonance imaging scanners, silicon photonics integrated systems, and electrical power distribution networks. Prof. Daniel was the recipient of the Best-Paper awards from several journals of the Institutes of Electrical and Electronics Engineers (IEEE), including Transactions on Power Electronics, Transactions on Computer Aided Design, and Transactions on Components and Manufacturing, 14 Best-Paper awards at international conferences, IBM Corporation Faculty Award, IEEE Early Career Award in Electronic Design Automation, Spira Award for Excellence in Teaching from the MIT School of Engineering, the Best Ph.D. Thesis awards from both the Department of Electrical Engineering and Computer Sciences and the Department of Applied Mathematics at the University of California at Berkeley, and the Outstanding Ph.D. Dissertation Award in electronic design automation from the Association for Computing Machinery (ACM).



**Duane S. Boning** (Fellow, IEEE) is the Clarence J. LeBel Professor in Electrical Engineering and Computer Science in the EECS Department at MIT. He is affiliated with the MIT Microsystems Technology Laboratories, and serves as MTL Associate Director for Computation and CAD. From 2004 to 2011, he served as Associate Head of the EECS Department at MIT. Dr. Boning received his S.B. degrees in electrical engineering and in computer science in 1984, and his S.M. and Ph.D. degrees in electrical engineering in 1986 and 1991, respectively, all from the Massachusetts Institute of Technology. He was an NSF Fellow from 1984 to 1989, and an Intel Graduate Fellow in 1990. From 1991 to 1993 he was a Member Technical Staff at the Texas Instruments Semiconductor Process and Design Center in Dallas, Texas, where he worked on semiconductor process representation, process/device simulation tool integration, and statistical modeling and optimization. Dr. Boning is a Fellow of the IEEE, and has served as Editor in Chief for the IEEE Transactions on Semiconductor Manufacturing, and as chairman of the CFI/Technology CAD Framework Semiconductor Process Representation Working Group. He is a member of the IEEE, Electrochemical Society, Eta Kappa Nu, Tau Beta Pi, Materials Research Society, Sigma Xi, and the Association of Computing Machinery.