Extending Energy-Efficient and Scalable DNN Training and Inference with 3D Photonic Accelerator

Juliana Curry, *Graduate Student Member, IEEE,* Yuan Li, *Member, IEEE,* Ahmed Louri, *Fellow, IEEE,* Avinash Karanth, *Senior Member, IEEE,* Razvan Bunescu, *Member, IEEE*

***Abstract*—As deep neural network (DNN) models continue to grow in complexity, analog computing architectures have emerged as a promising solution to meet increasing computational demands. Among these, silicon photonic computing excels at efficiently executing dot product operations while leveraging inherent parallelism. Photonic phase change memory (photonic-PCM) further enhances photonic computing by enabling scalable, non-volatile storage. In this work, we introduce the 3D Large-Scale Photonic Accelerator (LSPA), a novel photonic computing architecture designed for large-scale DNN models. LSPA employs multi-layered 3D stacking of non-volatile photonic-PCM cells, creating a high-density computational fabric that optimizes energy efficiency, flexibility, and scalability. LSPAs custom 3D photonic network enables simultaneous data multicast in two dimensions and accumulation in three dimensions, optimizing communication patterns essential for efficient DNN training. A distinctive feature of LSPA is its ability to execute multiple forward and backward passes in parallel within each mini-batch, reducing latency associated with data movement and photonic-PCM programming. This unique capability combined with high-bandwidth photonic interconnects allows LSPA to sustain efficient training across a wide range of DNN workloads. When evaluated against a range of neural network models including VGG-16, ResNet-50, GoogLeNet, Transformer, GNMT, LLaMA 7B, and LLaMA 30B, LSPA reduces execution time by up to 92% and energy consumption by up to 90%. These results highlight LSPA as a transformative advancement in scalable, high-performance photonic computing for deep learning.**

***Index Terms*—Silicon photonics, PCM, 3D architecture, neural networks.**

I. Introduction

The exponential growth in deep neural network (DNN) model sizes has driven a need for scalable and energy-efficient accelerators [1], [2], [3], [4]. Conventional digital computing architectures struggle to meet these demands due to fundamental limitations such as transistor scaling slowdowns and stringent power constraints. Analog computing has emerged as a promising alternative, reducing power consumption, thus increasing energy efficiency and improving performance of DNN accelerators [5]–[15]. However, existing analog accelerators suffer from key bottlenecks that hinder their scalability. Memristor based accelerators face challenges such as finite switching speeds on the order of $10ns$ that are slower than those of conventional transistors on the order of ps or ns and manufacturing variability that poses significant challenges for reliability and large-scale integration [6]. Silicon photonics utilizing microring resonators (MRRs) and Mach-Zehnder Interferometers (MZIs) achieve high speed operations ([16], [17]) but incur high tuning power requirements and susceptibility to process and temperature fluctuations [18]. Furthermore, both memristor and silicon photonics are restricted to a single operation on a memory or photonic cell at any time, thereby limiting the maximum achievable throughput and scalability.

Photonic phase change memory (photonic-PCM) technology capitalizes on the strengths of both memristor and silicon photonics technologies while effectively circumventing the drawbacks of both technologies specifically for analog computing systems [19], [20]. Photonic-PCM offers high durability, write cycle endurance (10^{12}) [21], non-volatility [22] and thermal stability (minimal 1-5% degradation after sustained exposure) [23], [24], [25] making photonic-PCM ideal for both inference and weights for DNN training and inference. To meet the growing DNN acceleration challenges, three-dimensional (3D) stacking approaches have been proposed that include micro-bumping, hybrid bonding, and monolithic 3-D integrated circuits (ICs) [26], [27], [28], [29]. Some industry examples of 3D packaging technologies include integrated 3D stacked memory by Graphcore IPU-2 [30], the Intel Foveros technology which uses micro-bump technology to stack dies vertically in a face-to-face (F2F) fashion [31] and TSMC's System-on-Integrated-Chips (SoIC) platform enabling the stacking and interconnection of multiple chiplets [32], [33]. While 3D integration leads to increased power density in electronic platforms, photonic 3D stacking 3D stacking achieves high computation density while maintaining energy efficiency.

In this paper we propose the **3D Large-Scale Photonic Accelerator (LSPA)**, a 3D-stacked photonic computing architecture that leverages photonic-PCM for large-scale DNN training. This design introduces several key contributions:

- **3D Vertical Stacking of Passive Photonic Layers**: LSPA stacks multiple non-volatile passive layers of photonic-PCM cells to achieve high computational density while extending energy efficiency, flexibility, and scalability. 3D vertical stacking enables high-density integration in LSPA while achieving low energy and low latency for DNN acceleration. By stacking a configurable number of passive photonic layers, LSPA's architecture is flexible and scalable to meet various application needs. Since the passive photonic layers only consist of MRRs being used as filters that do not need to be tuned periodically, and photonic-PCM cells which are non-volatile, power consumption is minimized.

Juliana Curry and Ahmed Louri are with George Washington University
Yuan Li is with Singapore Institute of Technology
Avinash Karanth is with Ohio University
Razvan Bunescu is with University of North Carolina at Charlotte
Manuscript received March 3, 2025; revised July 21, 2025.

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2025.3591812

- **Efficient Dataflow with 3D Photonic Interconnects:** The unique layout of the LSPA stack architecture combined with the WDM properties of photonic-PCM perform training efficiently without re-programming weights between the forward and backward passes. The stacked photonic-PCM cells are connected through a 3D photonic network to facilitate efficient data multicast in two dimensions and data accumulation in three dimensions, catering to the operation patterns observed in DNN training. The LSPA accelerator can support the parallel execution of forward and backward passes within each mini-batch, effectively hiding the latency of data movements and programming operations needed for the photonic-PCM cells. The main power requirement of the passive photonic layers is programming the photonic-PCM cells with weight kernels which is done using low-power optical pulses and minimized by the custom LSPA architecture layout that can be used in multiple directions to avoid reprogramming photonic-PCM cells between forward and backward passes.
- **Performance Evaluation:** Through extensive modeling and simulation we demonstrate that LSPA outperforms state-of-the-art digital and analog accelerators across various DNN workloads. We consider state-of-the-art analog computing accelerators Pipelayer (resistive memory technology-based architecture), DEAP-CNN (MRR-based architecture), and PTC (photonic-PCM-based architecture) as well as digital accelerator implementations on GPU, FPGA, and CPU platforms and state-of-the-art digital accelerator TPU. Simulation results demonstrate that LSPA reduces execution time by 92% and energy consumption by 90% compared to other accelerators.

II. Background and Motivation

A. Technologies for Analog Computation

Resistive Memory: With DNN models exponentially increasing in size, complexity and energy consumption, emerging technologies such as resistive memory technology [5], [10] and silicon photonics [34], [16] have been exploited for analog dot product operations. Figure 1a shows one such example where a memristor cell is located at each intersection between wires. The memristor cells are programmed to conductance values representing a first vector $\vec{G}$ while voltage levels representing a second vector $\vec{V}$ are applied to the horizontal wires. According to Kirchhoff's Law, a current signal $I = V_1 \times G_1 + V_2 \times G_2 + V_3 \times G_3$ is observed and measured at the bottom of the vertical wire representing the dot product $\vec{V} \cdot \vec{G}$. Memristors are non-volatile but suffer from low operating frequency and scalability due to process variations and fluctuations [8].

Silicon Photonics: Figure 1b illustrates an example of an MRR implementation of an analog dot product operation. Each MRR couples a unique resonant wavelength from the respective horizontal waveguide to the vertical waveguide. We assume three wavelengths with their power levels P_1 P_2, and P_3, respectively, representing a first vector $\vec{P}$, and MRR cells programmed to attenuation factors by θ_1, θ_2, and θ_3 by shifting their resonant wavelengths to represent

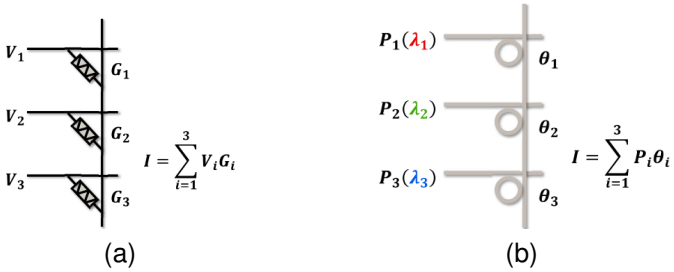


Fig. 1. Analog dot product operations using (a) memristor cells and (b) photonic microring resonator (MRR) cells.

TABLE I
Analog Dot Product Device Comparison

Device	Non-Volatile	Parallel dot-product	Bit Resolution	Tuning Power
Memristor	Yes	No	2-bit [5]	$15\mu W$ [5]
Thermal MRR	No	No	6-bit	1.7 mW [35]
Electronic MRR	No	No	6-bit	1.23 mW [36]
Photonic-PCM	Yes	Yes	8-bit	2 mW [37]

a second vector $\vec{A}$. The light power level measured as a photocurrent signal at the bottom of the vertical waveguide is $I = P_1 \times \theta_1 + P_2 \times \theta_2 + P_3 \times \theta_3$ representing the dot product $\vec{P} \cdot \vec{A}$. Such MRR technology-based dot product architectures require electric or thermal tuning to attenuate signals as well as to compensate for process and temperature variations [18] but can achieve high operating frequency [17] and scalability [16].

The majority of energy consumption and execution time incurred in existing photonic accelerators are rooted in the tuning of MRRs (up to 50% for DEAP-CNN) [16]. Therefore, reducing the energy consumption for MRR tuning has the potential to improve the performance of photonic accelerators as compared in Table I. Electronic tuning at 0.2 pm/V or 24.0 Hz/V will require applying large DC voltages whereas thermal tuning requires thermal heaters for each MRR which can shift an MRR's resonant wavelength within $\phi \pm 0.2$. While effective in shifting the resonant wavelength, crosstalk needs to be avoided from adjacent channels in a multi-channel WDM system resulting in only 6 bits of resolution [38], making thermal tuning challenging for training.

Photonic-PCM: Photonic-PCM reduces energy consumption due to non-volatile tuning, has the added benefit of 8-bit resolution, and is compatible with WDM which provides an extra degree of parallelism. By exploiting the frequency domain in the form of WDM with photonic-PCM, the simultaneous operation of multiple input vectors encoded on different wavelengths are implemented on a photonic-PCM cell without interference. By contrast, a memristor cell can only generate a single current output based on the input voltage and its conductance value at any time. An MRR cell can attenuate the intensity of a single resonant wavelength at any time. One prior analog computing system [19] based on photonic-PCM technology has leveraged this property to improve system throughput for DNN inference.

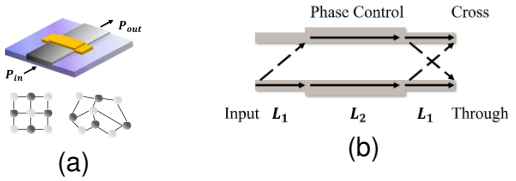


Fig. 2. (a) Architecture of the photonic phase change memory (PCM) cell and (b) Working mechanisms of a directional coupler.

LSPA leverages the non-volatile property of photonic-PCM devices to perform dot product operations similar to that of memristor devices and achieves high operating frequency and scalability similar to using photonic devices. Furthermore, since the programmed attenuation factor of any photonic-PCM cell is not specific to a fixed wavelength, more than one vector dot product can be performed at once on the same photonic-PCM cell. Although parallel dot product operations using WDM have been achieved in prior work [23], we design the layout of the LSPA accelerator architecture to uniquely leverage WDM in order to facilitate the parallel operation of forward and backward passes in DNN training. The layout of LSPA expedites performing dot product operations between weight vectors programmed in photonic-PCM cells and input feature vectors as well as dot product operations between the same weight vectors programmed in photonic-PCM cells and output feature gradient vectors in parallel.

B. Key Photonic Devices for LSPA

Photonic Phase Change Memory: LSPA relies on photonic phase change memory (photonic-PCM) to attenuate photonic signals, thereby performing multiplication [23], [39], [40], [41]. An essential characteristic of photonic-PCM cells is their significant contrast in both electrical (resistivity) and optical (refractive index) properties between the amorphous and crystalline states. Consider an architecture in which a photonic-PCM cell is positioned directly above a waveguide as shown in Figure 2a. The state of this photonic-PCM cell can be either read (through measuring the attenuation of the input light at the output side, $P_{out} - P_{in}$) or programmed (by applying an optical pulse with a certain power level).

Resolution: The photonic-PCM cell represents 0 when programmed to the crystalline state since most of the light is absorbed and represents 1 when programmed to the amorphous state since none of the light is absorbed by the cell. Moreover, photonic-PCM can be programmed to a transient state (between 0 and 1) wherein partial light is absorbed by the cell. The number of distinguishable states between the crystalline and amorphous states determines the resolution of the photonic-PCM cell. For this work we use photonic-PCM cells with 8-bit resolution [42], [43].

Reliability of Photonic-PCM: The reliability and durability of photonic PCM for dot products has been validated in a 600-hour durability test that showcases the resilience of photonic PCM under continuous operation [23]. A 10^{12} write cycling endurance has been reported [21] and 10^{15} endurance is expected [19]. Once written, PCM memory is non-volatile for up to 10 years [22], [20], [44]. Thermal cycling and optical fatigue tests also show that continuous operation under intense optical pulses leads to minimal degradation in performance, around 1-5 % after sustained exposure [45], [46], [24], [25]. Since thermal and optical stability is high, photonic-PCM can resist issues such as overheating, stress accumulation, or light-induced damage, all of which could degrade performance if they were more pronounced [47]. Since the photonic-PCM cell is reconfigurable, non-volatile, and can be written to and read from using optical pulses, photonic-PCM is ideal for implementing weights for both inference and training in LSPA stack. However, it is vital that weights are reused whenever possible to minimize the costly photonic-PCM cell tuning process.

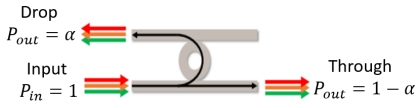


Fig. 3. Transient state operation of MRR in which a fraction α of the power of the light from the input port is coupled to the the drop port, while the remaining $1 - \alpha$ fraction of the power can be observed at the through port.

Directional Coupler: Directional couplers are utilized to couple a specific MRR to two neighboring waveguides in the active photonic layer. Figure 2b illustrates the directional coupler [48] used in the LSPA accelerator architecture. From left to right, it consists of a symmetric coupler, an asymmetric-waveguide-based phase control, and a second symmetric coupler. The split ratio of light power at the cross port over the input port is determined by the length of the symmetric coupler L_1 and the length of the phase control section L_2, among other parameters. In the LSPA accelerator architecture, a directional coupler with a maximum coupling length of $10\mu m$ is sufficient to cover all necessary split ratios as in [23].

Microring Resonators: MRRs can be used for several purposes including modulating, filtering and partially splitting power as shown in Figure 3 [34]. In the transient state shown, a fraction α of the power of the light from the input port on the first horizontal waveguide is coupled to the circular waveguide, and eventually to the drop port on the second horizontal waveguide, while the remaining $1 - \alpha$ fraction of the power can be observed at the through port on the first horizontal waveguide. In on-resonance state, all power from the input will be coupled to the drop port and in off-resonance, all power from the input will continue to the through port. In the passive photonic layers of the LSPA architecture, the MRRs are passive devices that are only used to filter wavelengths and couple optical signals between layers. In the passive photonic layers of the LSPA architecture, the MRRs are passive devices that are only used to filter wavelengths and couple optical signals between layers.

C. DNN Operations

A convolutional layer in DNN models is typically described using eight indices: the height $\langle r \rangle$ and width $\langle s \rangle$ of weight kernels, the height $\langle e \rangle$ and width $\langle f \rangle$ of output feature channels, the height $\langle h \rangle$ and width $\langle w \rangle$ of input feature channels, the number of input feature channels $\langle c \rangle$, and the number of output feature channels $\langle k \rangle$. Since the indices $\langle h \rangle$ and $\langle w \rangle$ can be derived from the other four indices, each convolutional

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2025.3591812

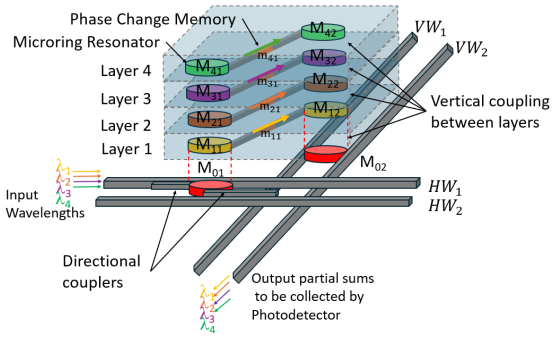


Fig. 4. Forward pass with LSPA devices using vertical coupling of light when supplying light from waveguide HW_1 and measuring the output dot product at the output of waveguide VW_2. The notation M_{ij} is used to indicate the *jth* MRR on the active photonic layer ($i = 0$) or the *ith* passive photonic layer. The notation m_{ij} is used to indicate the *jth* photonic-PCM cell on the *ith* passive photonic layer.

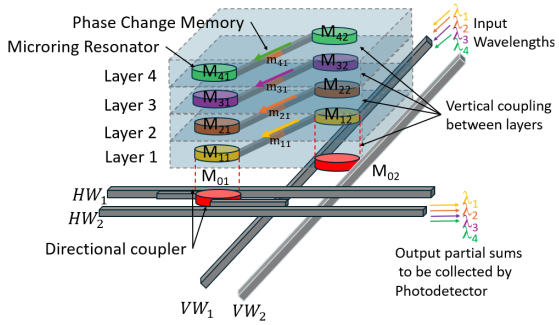


Fig. 5. Backward pass with LSPA devices using vertical coupling of light when supplying light from waveguide VW_1 and measuring the output dot product at the output of waveguide HW_2. Backward pass operation is performed in the opposite direction so that photonic-PCMs do not have to be re-programmed with new weights.

layer can therefore be presented as a six-element vector $\langle R, S, E, F, C, K\rangle$, wherein each element indicates the range of a respective index. The forward pass of a DNN layer entails sliding the weight kernels across the input feature channels and performing the dot product operations between weight vectors Y and input feature vectors X in the respective sliding window, as shown in Equation 1.

$$O_{efk} = \sum_{c}\sum_{s}\sum_{r} Y_{rsck} \times X_{r+e,s+f,c} \tag{1}$$

The output features of layer *i* are first derived and then taken as input features of the subsequent DNN layer, *i+1* after being processed by a proper activation function σ_i, where *i* is the layer index. The backward pass of a DNN layer is performed by sliding the weight kernels across the horizontally and vertically flipped output feature gradient channels and the dot product operations are performed between weight vectors and output feature gradient vectors in the corresponding sliding window, as shown in Equation 2.

$$\frac{\delta L}{\delta X_{hwc}} = \sum_{k}\sum_{s}\sum_{r} Y_{rsck} \times \frac{\delta L}{\delta O_{h-r,w-s,k}} \tag{2}$$

Those flipped output feature channels are zero-padded to ensure the proper dimension sizes of the generated input feature gradient channels. Input feature gradients on the generated input feature gradient channels are positioned in a horizontally and vertically flipped manner, as compared to the respective original input feature channels. The reason for the flipped arrangements of output and input feature gradient channels is the negative signs of both the $\langle r\rangle$ and $\langle s\rangle$ indices in Equation 2. Note the accumulation of the multiplications $Y_{rsck} \times \delta L/\delta O_{h-r,w-s,k}$ is performed along both $\langle r\rangle$ and $\langle s\rangle$ dimensions, as well as the $\langle k\rangle$ dimension. Updating the weight kernels of a DNN layer is achieved by sliding the output feature gradient channels across the input feature channels and performing the dot product operations between output feature gradient vectors and input feature vectors in the corresponding sliding window, as shown in Equation 3. Figure 8 illustrates the dot product operations during the weight updating process of the same convolutional DNN layer. The weight gradients are derived and exploited to update the weight kernels of the current DNN layer on either a per-example or a per-mini-batch basis.

$$\frac{\delta L}{\delta Y_{rsck}} = \sum_{f}\sum_{e} \frac{\delta L}{\delta O_{efk}} \times X_{r+e,s+f,c} \tag{3}$$

III. LSPA Accelerator Architecture

A. LSPA Dot Product

Figure 4 illustrates the forward pass working schemes of the vertical accumulation operation. The wavelengths λ_1, λ_2, λ_3, and λ_4 are coupled to MRR M_{01} in the clockwise direction and then vertically coupled to MRRs above it (M_{11}, M_{21}, M_{31}, M_{41}). At each vertical layer, a specific wavelength is filtered out, attenuated by the respective photonic-PCM cell, and then coupled back to another MRR (M_{12}, M_{22}, M_{32}, M_{42}). The vertically stacked MRRs [49], [50] working at the on-resonance state extract a particular resonant wavelength [51] and couple it back after attenuation. Wavelengths are spaced at least 0.8 nm apart [52], [53], [54] and photonic layers are spaced 5 μm apart to prevent crosstalk and interference between layers [55]. Photonic-PCM cells are pre-programmed with weight kernel values using optical pulses. The influence of dispersion in PCM attenuation can be neglected in the 180-205 THz wavelength range and be corrected by adjusting the input amplitudes of different wavelengths [23].

For instance, wavelength λ_1 is coupled to a local waveguide from M_{11}, attenuated by photonic-PCM cell m_{11}, then coupled to M_{12}, and eventually measured at waveguide VW_2. By contrast, the working scheme shown in Figure 5 describes how the light coming from waveguide VW_1 is vertically coupled, attenuated, and then measured at waveguide HW_2 for the backward pass. Backward pass operation is performed in the opposite direction so that photonic-PCMs do not have to be re-programmed with new weights between the forward and backward passes or subsequent forward passes. By using a different frequency range for waveguides on the y-axis of the active photonic layer and WDM, rose-wise and column-wise accesses double the inputs which can be multiplied with the weights stored on the passive photonic layers simultaneously. If the model is already trained and backward passes are not needed, the second direction is used for a second forward pass. This minimizes reprogramming of photonic-PCM cells which

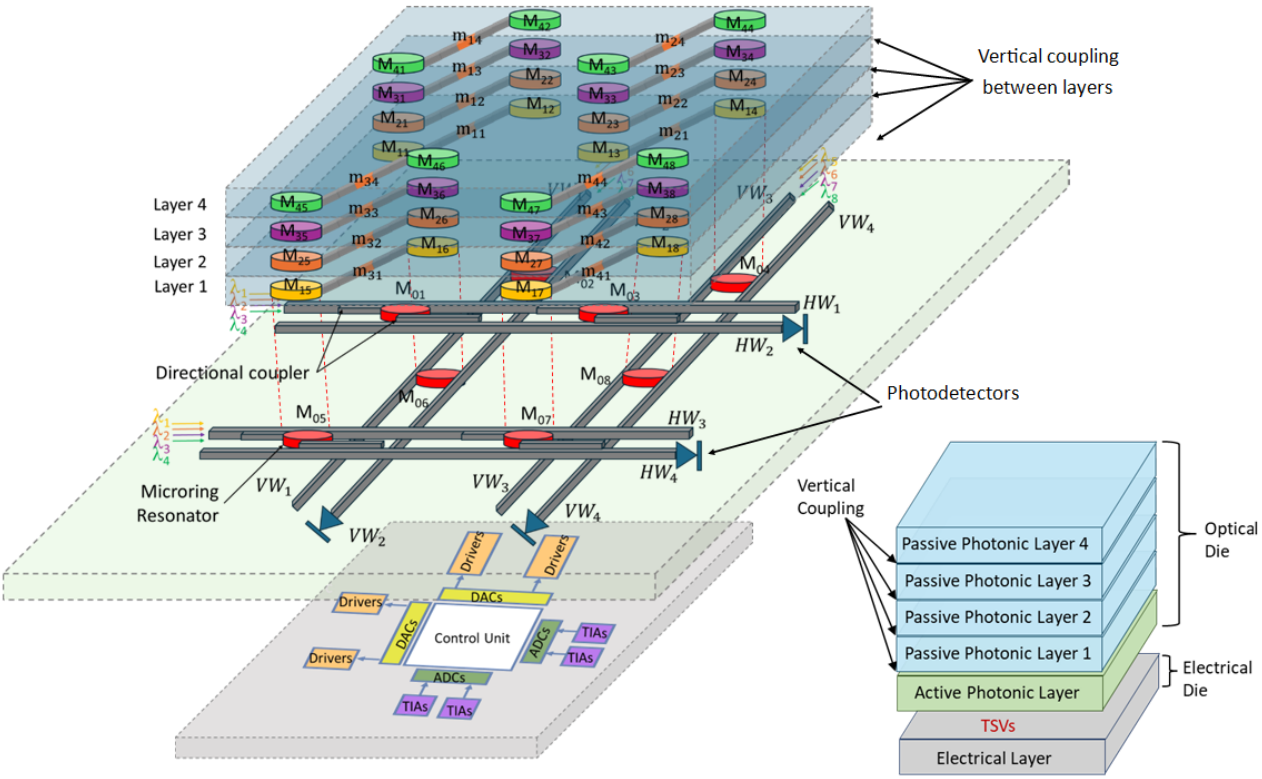


Fig. 6. A sample LSPA stack architecture with an electrical layer, an active photonic layer, and four passive photonic layers. The drivers on the electrical layer are electrically connected to microring resonators (MRRs) working as attenuators on the active photonic layer through vertical through-silicon-vias (TSVs). The transimpedence amplifiers (TIAs) on the electrical layer are electrically connected to photodetectors on the active photonic layer through TSVs. Light is coupled vertically between the active photonic layer and passive photonic layers [49].

is costly in terms of both laser energy and latency. Figure 6 shows the LSPA accelerator stack with four passive photonic layers where each layer integrates four photonic-PCM cells.

B. LSPA Stack Architecture

Each LSPA accelerator stack includes an electrical layer, an active photonic layer, and several passive photonic layers. The LSPA accelerator architecture includes multiple LSPA stacks whose passive photonic layers are connected to the laser source using space-division multiplexing. Meanwhile, their electrical layers are connected through a 2D electrical torus network to support data exchange between stacks in the digital domain.

1) Electrical Layer: As shown in Figure 6, the electrical layer includes a global buffer for temporary data storage, two sets of digital-to-analog converters (DACs) for converting input feature vectors and output feature gradient vectors from the digital domain to the analog domain in forward and backward passes, respectively, two sets of analog-to-digital converters (ADCs) for converting output feature vectors and input feature gradient vectors from the analog domain to the digital domain in forward and backward passes, respectively. It is the electrical layer's components, ADCs, DACs, and TIAs, that determine the stack footprint, not the photonic layers. Other peripheral functions such as the activation and derivative function of the activation function σ^i as well as the calculation of the weight gradients are performed digitally in this layer. The drivers and transimpedance amplifiers (TIAs) on this layer

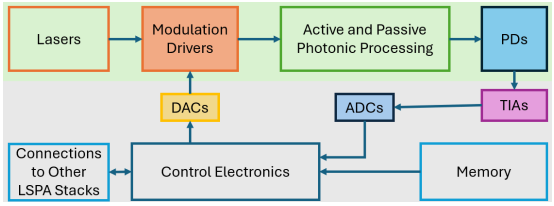


Fig. 7. Overview of interaction between electrical components and photonic processing layers.

are electrically connected to the MRRs and photodetectors on the active photonic layer vertically using through-silicon-vias (TSVs). The electrical layer also serves as the bridge between LSPA stacks since the electrical layers are connected in a 2D torus network (more explanation in Section 4). The torus allows for numerous LSPA chips to be connected for system scaling similar to a systolic array such as TPU-v4 [56]. An overview of how the electrical components interact with the photonic layers is shown in Figure 7

2) Active Photonic Layer: The active photonic layer is positioned between the electrical layer and the first passive photonic layer. Its functions include performing optical signal attenuation and photocurrent detection, as well as evenly distributing optical power along two planar dimensions. Figure 6 shows the architecture of an active photonic layer in green. MRRs attached to waveguides HW_1, HW_3, VW_1, and VW_3 are driven by the respective devices on the electrical layer to generate input feature or output feature gradient vectors

for dot product operations. As shown in Figure 6, the two directional couplers attached to MRR M_{03} are utilized to couple the incoming light from waveguide HW_1 to the MRR M_{03} and the outgoing light from the MRR M_{03} to waveguide HW_2. By tuning the split ratios of the directional couplers at fabrication time, light power coming from waveguides HW_1 and HW_3 can be evenly distributed to MRRs M_{01} and M_{03}, and MRRs M_{05} and M_{07}, respectively. Similarly, the light power coming from waveguides VW_1 and VW_3 can be evenly distributed to MRRs M_{02} and M_{06}, and MRRs M_{04} and M_{08}, respectively. By using MRRs for filtering only, no tuning power is required for MRRs in LSPA. The functions of the active photonic layer include (1) generating input feature or output feature gradient vectors, (2) row-wise and column-wise multicasting the generated vectors and initiating accumulation along $\langle r \rangle$ and $\langle s \rangle$ dimensions, (3) collecting vertically accumulated intermediate results and performing accumulation in a third dimension, and (4) measuring the final results using photodetectors. LSPA's custom architecture accommodates data movement in the $\langle k \rangle$ and $\langle c \rangle$ dimensions, which is typical of 2D transposable memory [57], as well as the $\langle r \rangle$ and $\langle s \rangle$ dimensions.

3) Passive Photonic Layers: Figure 6 shows the architecture of the *ith* passive photonic layer in yellow where a pair of MRRs (e.g., M_{i1} and M_{i2}) are connected through a local waveguide and a photonic-PCM cell is positioned on top of the local waveguide to perform light power attenuation. The passive photonic layers are identical to each other and can be stacked to scale the LSPA architecture to various sizes. The photonic layers are aligned and vertically stacked to ensure seamless coupling of corresponding MRRs. When transitioning LSPA from a theoretical model to a practical deployment, anticipated challenges and considerations when bonding individual layers include layer alignment to ensure efficient optical coupling and minimal losses [58], material compatibility to avoid thermal mismatch [45], and fabrication costs [59]. Fabricating and aligning 10 layers in silicon photonics accurately is already feasible [60], [61], [62].

4) Negative Values: Analog computations in the photonic domain are not limited to positive numbers. The electric field of a photonic waveform is a sinusoidal function that oscillates between positive and negative values. However, photodetectors measure the intensity of photonic signals rather than amplitude and the negative and positive values of the electric field result in the same measured intensity, making direct detection insensitive to the sign of the amplitude. To encode negative values in photonic accelerators, phase shifting is utilized [63], [64], [65]. The total electrical field collected at the photodetectors is defined by Equation 4. A_n is the amplitude of signal n. ϕ is the phase shift of signal n [66]. ω_n is the angular frequency of signal n defined by $\omega = \frac{2\pi c}{\lambda}$ where c is the speed of light in a vacuum, 3.0×10^8 m/s and λ is the wavelength of the signal which matches the resonant wavelength the corresponding MRR in an LSPA passive photonic layer.

$$E(t) = \sum_n A_n e^{i(\omega_n t + \phi_n)} \tag{4}$$

The intensity $I(t)$ detected by the photodetector collecting partial sums on the active photonic layer is described by Equation 5 where ε_0 is the permittivity of free space, c is the speed of light, and $|E(t)|^2$ is the square of the magnitude of the electric field.

$$I(t) = \frac{\varepsilon_0 c |E(t)|^2}{2} \tag{5}$$

The photocurrent collected by the photodetector, I_p is equal to $I(t) * R_d$ where R_d is the responsivity of the photodetector. The photodetector's output analog voltage is equal to $V = I_p * R$ where R is the load resistance. A key takeaway from the definition for the intensity of light is that the intensity of light $I(t)$ is directly proportional to the square of the magnitude of the electric field.

$$I(t) \propto |E(t)|^2 \tag{6}$$

The square of the magnitude of the total electric field $|E(t)|^2$ is expanded out in Equation 7.

$$|E(t)|^2 = \sum_n A_n^2 + 2 \sum_{n<m} A_n A_m cos[(\omega_n - \omega_m)t + (\phi_n - \phi_m)] \tag{7}$$

For two signals with ω_1 and ω_2, the intensity will oscillate between a maximum and a minimum over time, as the signals interfere. The detected intensity will vary as a function of the beat frequency $\Delta\omega = \omega_1 - \omega_2$. Making the phase difference $[(\omega_n - \omega_m)t + (\phi_n - \phi_m)]$ an odd multiple of π, the cosine term will be -1, thus implementing negative numbers.

Negative values are handled in LSPA by always using positive weight values in photonic-PCM cells and encoding negative input values where needed using phase shifters after laser encoding and before the active photonic waveguides. LSPA uses MRRs to modulate the incoming laser source and encode input values onto each wavelength. The necessary phase shift is also introduced by these MRRs.

C. LSPA Stack Operation

1) Forward Pass Execution: Please note the terminology M_{ij} is used to indicate the *jth* MRR on the active photonic layer ($i = 0$) or the *ith* passive photonic layer. On the active photonic layer, four MRRs attached to the horizontal waveguide HW_1 are employed to attenuate the power levels of respective wavelengths $\lambda_1, \lambda_2, \lambda_3$, and λ_4 from the laser source to represent an input feature vector as described in Algorithm 1.

With directional couplers on the active photonic layer, input features are shared along the $\langle k \rangle$ dimension. The directional coupler between the horizontal waveguide HW_1 and MRR M_{01} exhibits a split ratio of 0.5, hence, coupling half of the power of wavelengths λ_{1-4} to the MRR M_{01} while forwarding the remaining half downstream. The directional coupler between the horizontal waveguide HW_1 and MRR M_{03} exhibits a split ratio of 1, hence, coupling all the remaining power of wavelengths λ_{1-4} to the MRR M_{03}. In this way, the same input feature vector is present at two MRR locations, M_{01} and M_{03}. Similarly, another four MRRs attached to the horizontal waveguide HW_3 are employed to attenuate the power levels of respective wavelengths λ_{5-8} from the laser source to represent

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2025.3591812

Algorithm 1 Forward Pass Input Assignment Algorithm

Input Wavelength Assignment ▷ given L, the number of wavelengths, and dimensions H, W, and C
for c = 1 to C **do**
  for h = 1 to H **do**
    for w = 1 to W **do**
      $\lambda_l \leftarrow X_{hwc}$
      $l \leftarrow l + 1$
      if $l > L$ **then**
        $l \leftarrow 1$
      end if
    end for
  end for
end for
Photonic-PCM Cell Weight Assignment ▷ given I, the number of passive photonic layers, J, the number of PCM per layer, and dimensions R, S, C, and K
$i = 1, j = 1$
for r = 1 to R **do**
  for s = 1 to S **do**
    for c = 1 to C **do**
      for k = 1 to K **do**
        $m_{ij} \leftarrow Y_{rsck}$
        $j \leftarrow j + 1$
        if $j > J$ **then**
          $j \leftarrow 1$
        end if
      end for
    end for
    $i \leftarrow i + 1$
  end for
end for

Algorithm 2 Backward Pass Input and Weight Assignment Algorithm

Input Wavelength Assignment ▷ given L, the number of wavelengths, and dimensions H, R, W, S, and K
for k = 1 to K **do**
  for r = 1 to R **do**
    for s = 1 to S **do**
      $\lambda_l \leftarrow \frac{\delta L}{\delta O_{h-r,w-s,k}}$
      $s \leftarrow s + 1$
      $l \leftarrow l + 1$
      if $l > L$ **then**
        $j \leftarrow 1$
      end if
    end for
  end for
end for
Photonic-PCM Cell Weight Assignment ▷ given I, the number of passive photonic layers, J the number of PCM per layer, and dimensions R, S, C, and K
for c = 1 to C **do**
  for k = 1 to K **do**
    for r = 1 to R **do**
      for s = 1 to S **do**
        $m_{ij} \leftarrow Y_{rsck}$
        $j \leftarrow j + 1$
        if $j > J$ **then** $j \leftarrow 1$
        end if
      end for
    end for
    $i \leftarrow i + 1$
  end for
end for

another input feature vector at two MRR locations, M_{05} and M_{07}. Partial accumulations collected at MRRs M_{02} and M_{06} are combined in the shared waveguide VW_2 and measured the by the photodetector at the end of VW_2. Similarly, partial accumulations collected at MRRs M_{04} and M_{06} are combined and measured by waveguide VW_3 and it's photodetector.

The accumulation of partial sums along both $\langle r \rangle$ and $\langle s \rangle$ dimensions are achieved using clusters of vertically aligned photonic-PCM cells in the passive photonic layers. Please note the terminology m_{ij} is used to indicate the *jth* photonic-PCM cell on the *ith* passive photonic layer. Photonic-PCM cells are programmed to attenuation factors representing weights as described in Algorithm 1. In sequential clock cycles, input feature vectors from different sliding windows are supplied from the electrical layer to the active photonic layer to generate the output features.

2) Backward Pass Execution: On the active photonic layer, four MRRs attached to the vertical waveguide VW_1 are employed to attenuate the power levels of respective wavelengths $\lambda_1, \lambda_2, \lambda_3$, and λ_4 from the laser source to represent an output feature gradient vector as described in Algorithm 2.

The directional couplers between the vertical waveguide VW_1 and MRRs M_{06} and M_{02} split the power of wavelengths $\lambda_1, \lambda_2, \lambda_3$, and λ_4 equally to present the same output feature gradient vector at both MRRs. Similarly, another four MRRs attached to the vertical waveguide VW_3 are employed to attenuate the power levels of respective wavelengths $\lambda_5, \lambda_6, \lambda_7$, and λ_8 from the laser source to represent another output feature gradient vector and couplers split the vector between two MRR locations, M_{08} and M_{04}. This is how the sharing of output feature gradients is facilitated along the $\langle c \rangle$ dimension during backward pass execution.

The accumulation of multiplications along both $\langle r \rangle$ and $\langle s \rangle$ dimensions are achieved by reusing the clusters of vertically aligned photonic-PCM cells in the

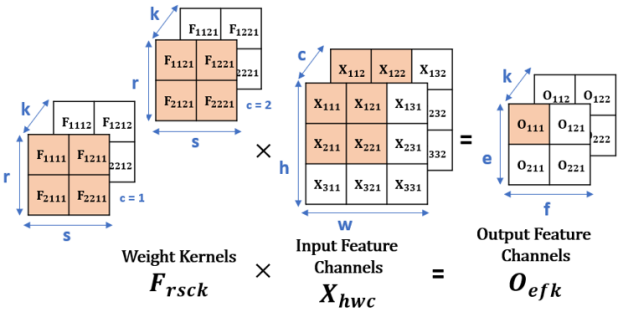


Fig. 8. Dot product operation during the forward pass in convolutional layer $\langle R, S, E, F, C, K \rangle = \langle 2, 2, 2, 2, 2, 2 \rangle$. The computation of output feature O_{111} is highlighted.

backward pass execution as described in Algorithm 2. Please note the weights programmed in photonic-PCM cells remain unchanged during the transition between forward and backward passes. In each clock cycle, output feature gradients from different sliding windows are supplied from the electrical layer to the active photonic layer to generate the remaining input feature gradients.

3) Weight Update execution: The weight gradients are calculated on the electrical layer in the digital domain. The updating of weights programmed in the photonic-PCM cells is achieved by encoding the updated weights on the wavelengths and forwarding them to the respective cells [23], reusing the architecture and mechanism described in the forward pass execution section.

4) Executing Other NN Operations: While LSPA is optimized for CNN models which share weights across different input regions, other NN models can also be executed and expedited with LSPA. Reinterpreting operations into convolutions allows other model types to be mapped to LSPA. For example, fully connected layers can be viewed as convolutions with large kernel sizes and specific stride patterns. Mapping

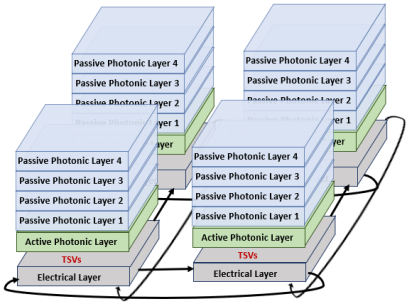


Fig. 9. 4 LSPA stacks connected by torus network on electronic layer

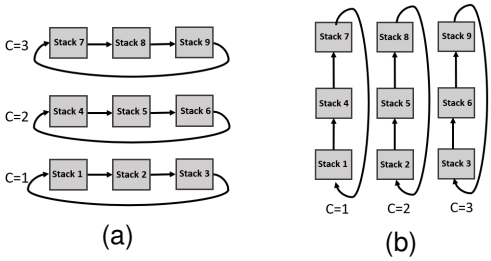


Fig. 10. Top-down View of torus connecting electronic layers of LSPA stacks showing data movement during (a) forward pass and (b) backward pass occurring in parallel

transformer NN models onto LSPA involves decomposing the transformer computational structure into a form suitable for parallel and efficient execution, particularly matrix-heavy operations. Each transformer attention block uses several linear layers for generating query, key, and value matrices from inputs, which are essentially matrix multiplications that map directly onto LSPA's input drivers and weight matrix photonic-PCM cells ($Q = XW^Q$, $K = XW^K$, and $V = XW^V$). Linear transformations and activations are similarly suited to optical computation, where matrix-vector products are performed in parallel across the photonic array, and nonlinear activation functions, such as GeLU or ReLU, can be implemented in the digital domain on the electronic layer. Subsequent operations in the attention mechanism, such as dot-product attention ($\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$), require additional stages: first a matrix multiplication (QK^T), a softmax normalization, and a final matrix multiplication with V. The QK^T operation is particularly well-suited for photonic matrix multiplication, while softmax is approximated using low-power electronic post-processing. Thus, the transformer's heavy reliance on linear algebra operations aligns naturally with LSPAs strength in high-throughput, parallel, low-latency matrix computations. LSPA's memory hierarchy and torus network connecting stacks are designed to keep data close to processing units and minimize hops between stacks to 1 at all times, minimizing latency and bandwidth bottlenecks during processing.

IV. Dataflow and Pipelining

To use LSPA for large-scale NN models, multiple LSPA stacks are combined in a torus network so tiling can be used. We map the NN model parameters to LSPA stacks so that 1 hop maximum is needed between clock cycles. Figure 9 shows an example of 4 LSPA stacks connected in a torus

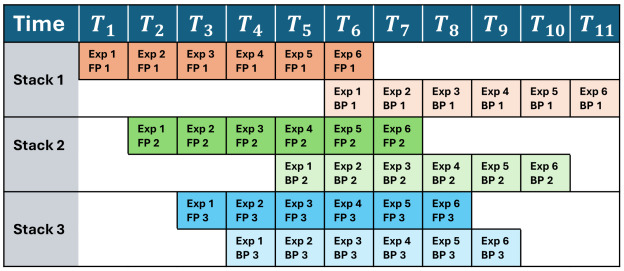


Fig. 11. LSPA pipeline that leverages parallelism between forward pass (FP) and backward pass (BP) of examples in the same mini-batch. Forward pass through layers 1, 2, 3 and then backward pass through the same layers 3, 2, 1 are executed for each example.

network on the electronic layer to share data between stacks. In the example shown in Figure 10a, each input feature X_{hwc} is mapped to LSPA stacks such that each row of LSPA stacks shares the same c index. The input features for $c = 1$ are mapped to the bottom row where each LSPA stack is pre-programmed with weight kernel Y_{rsck} where $c = 1$. Multiplication with the weights stored in the passive photonic layers happens in one cycle. The weights kept in each LSPA stack's passive photonic layers do not change between cycles, only the inputs are forwarded from one LSPA stack to the next stack in the row to be multiplied with the next set of weights. Because of this dataflow pattern, connections between rows of LSPA stacks are not needed.

Assume that a three-layer sample DNN is performed on the LSPA accelerator architecture where each stack accommodates the weight kernels of a respective layer. The training process of an example exhibits a sequential order. Though the separation of the forward and backward modes of LSPA architecture eliminates the need to perform costly weight matrix reordering or duplication, only one of the two modes is utilized at any time, leading to low hardware utilization. We therefore explore the parallelism between examples in the same mini-batch to improve hardware utilization. As the size of neural network and programming operations increase, multiple LSPA stacks are used to implement more static weights and minimize re-programming PCM cells. Weight stationary data flow is also used to mitigate the impact of growing neural networks.

Figure 11 shows the situation of processing a mini-batch of several examples of a three-layer neural network on three stacks. We still assume that each stack can fully accommodate the parameters and computation operations of a corresponding layer. The forward passes of layer 1 for examples 1 through 6 are processed by stack 1. As the time slots progress, stacks 2 and 3 are utilized to process the second and third layers of the forward pass (FP) for all examples. At T_4 the backward mode sections begin processing the backward pass of each layer for each example. T_6 marks the completion of example 1 and the starting point of the full utilization of both mode sections of all three stacks. If only focusing on one example, it will adhere to the sequential order and take six steps to finish. However, by enabling parallelism between examples in the same mini-batch, we can completely fill the pipeline before the end of the current mini-batch. At the end of each mini-batch or processing tile, the weights

are updated optically with picosecond pulses [19] by reusing the wavelength associated with each cell. Vertically aligned cells in different layers are programmed simultaneously using their respective wavelengths. The pipeline utilization depends on two factors: mini-batch size and the number of pipelined layers.

A. Stochastic Gradient Descent Case

Following the assumption that a three-layer neural network can be accommodated by three LSPA stacks, each storing the weight kernels of a respective layer, the adoption of the LSPA pipeline can significantly reduce the execution time. Assume that the number of pipelined layers is l ($l = 3$ in this case) and the number of clock cycles required for photonic-PCM array updating is ω. For a regular pipeline, the overall clock cycles required to process a training example is $2l + 2\omega$. In contrast, for the proposed LSPA pipeline, the overall clock cycles required to process a training example is $2l + \omega$. The reduction is because the LSPA pipeline does not require the additional ω clock cycles to update the weight kernels from the original format to its transposed format between the forward and backward passes. Whether using a regular pipeline or an LSPA pipeline, each stack requires sufficient buffer space to store input features, output features, output feature gradients, input feature gradients, and weight gradients.

B. Mini-Batch Gradient Descent Case

Following the assumption that a three-layer neural network can be accommodated by three LSPA stacks, each storing the weight kernels of a respective layer. The adoption of the LSPA pipeline can reduce the execution time as well as decrease local buffer space. Assume that the number of pipelined layers is l ($l = 3$ in this case), training batch size is b, and the number of clock cycles required for photonic-PCM array updating is ω. For a regular pipeline, the overall clock cycles required to process a training batch is $2l+b+2\omega-2$. In contrast, for the proposed LSPA pipeline, the overall clock cycles required to process a training batch is $2l+b+\omega-1$ The reduction mainly comes from the fact that the LSPA pipeline does not require the additional ω clock cycles to update the weight kernels from the original format to its transposed format between the forward and backward passes.

V. Related Works

Several recent advancements have explored the integration of photonic technologies into DNN accelerators with the goal of improving both performance and energy efficiency. ASCEND [73] is a chiplet-based accelerator that utilizes a custom photonic interconnect network to enable seamless intra- and inter-chiplet broadcast communication. The ASCEND photonic network supports the flexible mapping of diverse convolutional layers, thereby mitigating the scalability limitations of traditional metallic interconnects. However, ASCEND relies entirely on electronic processing elements and is constrained by its optical-electrical (O/E) and electrical-optical (E/O) conversion mechanisms, which are limited to 1-bit per cycle due to the design of its transmitters and receivers. This restriction presents a significant throughput bottleneck. MDA [74] introduces a high-performance and energy-efficient photonic architecture specifically optimized for the concurrent execution of multiple DNNs. MDA employs a dynamically reconfigurable silicon photonic network that can be segmented to interconnect allocated compute resources. This flexibility allows the communication infrastructure to adapt to the dataflow patterns of individual DNNs, thereby improving both computational efficiency and scalability. However, MDA is still limited to 1-bit per cycle E/O and O/E conversions due to its transmitters and receivers. PTC [23] is an analog photnic accelerator that integrated photonic-PCM into photonic waveguides to store weights and perform multiply-accumulate (MAC) operations. By exploiting optical interference and photodetection, PTC enables in-memory computing with high bandwidth and low latency. LSPA extends the use of photonic-PCM by not only utilizing photonic-PCM for non-volatile memory and in-memory MAC operations but also by incorporating 8-bit DACs and ADCs to facilitate high-resolution O/E and E/O conversions. Unlike PTC, LSPA is designed to support both inference and training by enabling parallel forward and backward passes, thus avoiding frequent weight reprogramming. This architectural innovation significantly increases throughput and power efficiency, particularly in compute-intensive training scenarios.

TABLE II
LSPA Device Parameters

Device	Parameter	Value
DAC [67]	Resolution	8-Bit
	Power	$50mW$
	Speed	$14GS/s$
	Area	$11,000\mu m^2$
ADC [68]	Resolution	8-Bit
	Power	$15mW$
	Speed	$10GS/s$
	Area	$2,850\mu m^2$
TIA [69]	Power	$3mW$
	Area	$11,000\mu m^2$
Microring Resonator	Tuning Power	$0.2mW$ [70]
	Insertion Loss	$0.01dB$ [71]
	Area	$93\mu m^2$
Photodetector	Power	$1.1mW$ [72]
	Area	$40\mu m^2$ [72]
	Sensitivity	$-25dBm$ [23]
Directional Coupler	Insertion Loss	$0.1dB$ [23]
	Coupling Length	$10\mu m$ [23]
Waveguide Crossing	Insertion Loss	$0.03dB$ [23]
Microcomb	Area	$1.4mm^2$ [70]
Laser Source	Wall-Plug Efficiency	0.2 [70]
Photonic-PCM	Array Size	32×32
	Cell Size	$30\mu m \times 30\mu m$ [23]
	Cell Tuning Pulse	$200ns$ [23]
	Resolution	8-Bit [42], [43]

VI. Evaluation Methodology

A. Evaluation Setup

We extend the open-source ASTRA-Sim simulator [75] to model the proposed LSPA and other baseline architectures in both training and inference tasks. The energy and latency of training are simulated using ASTRA-Sim to model both the communication protocols and hardware behavior of the

evaluated architectures, enabling accurate estimation of total energy consumption and execution time across diverse neural network models, based on parameter counts and model sizes. The computation load for all analog computing baselines (Pipelayer, DEAP, PTC) and LSPA is derived based on the total number of operations, operating frequency, and active device count. The communication volume is dictated by the DNN model under evaluation and is uniformly applied across all baselines. To ensure a fair comparison, all architectures are constrained to a 400 mm^2 die fabricated in 5 nm technology, equipped with 16-bit precision arithmetic, 32 MiB of on-chip SRAM, 32 GiB of High-Bandwidth Memory (HBM), and a total off-chip memory bandwidth of 1200 GB/s.

We use the same restrictions when setting up GPU, FPGA, and CPU accelerators for a fair comparison. The GPU baseline is modeled with approximately 60 Streaming Multiprocessors (SMs), each equipped with 128 KB of L1 cache. The chip includes a 32 MiB shared L2 cache and leverages 32 GiB of HBM2e organized into 4 stacks, each with 8 channels at 300 GB/s bandwidth, yielding a total of 1200 GB/s. To remain within the 400 mm budget, the architecture allocates roughly 60% of area to compute units, 20% to SRAM/cache, and 20% to interconnect and peripheral logic. The design supports tensor core acceleration for FP16 operations, achieving a peak throughput of approximately 50 TFLOPs.

The FPGA baseline consists of a dense array of configurable logic blocks (CLBs) optimized for 16-bit arithmetic, along with approximately 15,000 DSP slices for high-throughput operations. The architecture includes 32 MiB of distributed block RAM, configurable as scratchpad or cache, and interfaces with 32 GiB of HBM2e via a 2.5D silicon interposer. The HBM is partitioned into 4 stacks, each with 300 GB/s bandwidth. The die area is apportioned as 60% for compute fabric, 25% for memory/cache and controllers, and 15% for NoC and peripheral logic. The estimated peak FP16 throughput is 25 TFLOPs, reflecting realistic DSP-backed acceleration.

The CPU baseline is composed of 64 out-of-order cores, each equipped with 64 KB of L1 cache and 512 KB of private L2 cache. A shared 32 MiB L3 cache spans the entire chip in a multi-banked configuration to support high bandwidth and low contention. SIMD acceleration is supported via 512-bit wide vector units (AVX-512 class), enabling efficient 16-bit operations. The memory system includes 32 GiB of HBM2e organized into 4 stacks, each delivering approximately 300 GB/s, for a total bandwidth of 1200 GB/s. The estimated peak FP16 throughput is 12 TFLOPs, constrained by the available core count, vector width, and achievable clock frequency within the area and power budget. While this configuration is optimistic relative to typical general-purpose server CPUs, it is intended to represent a DNN-optimized manycore processor fabricated in 5 nm technology and confined to a $400mm^2$ die. The model assumes high SIMD utilization and efficient memory access, more characteristic of custom accelerator-class CPUs than conventional data center architectures.

The key parameters of LSPA architecture are listed in Table II. We assume 8-bit ADC and DAC modules as we assume the photonic-PCM cell resolution is 8-bit. We choose ADC and DAC modules with very high (over $10GS/s$) speed as their sampling rate determines the operating speed of the overall LSPA architecture. As a result, the power consumption and area of such high-speed ADC and DAC modules are relatively high. We determine the number of LSPA stacks per chip to be 16 with their electrical layers connected by a 2D torus network and HBM channels evenly distributed to each stack. Current 2.5D CoWoS (Chip on Wafer on Substrate) technology supports configurations of up to 8 chiplets integrated with high bandwidth memory (HBM) [76], [77]. While 16 LSPA stacks are not yet feasible, research is moving towards 3D IC integration with larger numbers of chiplets. We determine the size of the photonic-PCM cell array per passive photonic layer to be 32×32 and the number of passive layers to be 9 since 3×3 weight kernels are common for NN models, including those evaluated [2], [3], [1], [4], [78] [79], and up to 10 bonded layers are feasible for current technology [60], [61], [62]. An 8×8 array has been prototyped and published in Nature [23] while a 64×64 array is projected [23]. A larger array size will lead to significantly higher laser power consumption. But this threshold is higher than in conventional 2D design [39] due to the additional vertical dimension. It is possible to connect numerous LSPA chips for system scaling similar to a systolic array such as TPU-v4 [56]. For details, please refer to the power model section below.

B. Power Model

The power consumption of the off-chip laser source (P_{1aser}) is obtained from Equation 8, when assuming the photodetector sensitivity $P_{rs} = -25dBm$ and the system margin $C_{system} = 4dB$. Note that the overall insertion loss C_{loss} is obtained by accumulating the insertion loss of each component along a silicon photonic communication channel. The power consumption per wavelength is 2.5 mW. The energy values of the laser and other peripheral circuitry (DACs, ADCs, TIAs, and photodetectors) are 0.06 nJ and 1.77 nJ, respectively.

$$P_{laser} = P_{rs} + C_{loss} + M_{system} \tag{8}$$

The proposed LSPA architecture is compared against other state-of-the-art digital and analog computing systems for DNN training tasks using a 16-bit fixed-point data format when using two photonic-PCM cells to represent each weight, LSPA-16, and 8-bit weights with one photonic-PCM cell each, LSPA-8. The same 8-bit ADCs are used for LSPA-16 but with lowered throughput since two samples are needed to capture each 16-bit value. The result is combined in the electrical layer using bit-shifting to reconstruct a 16-bit value. Similarly, LSPA-32 can be executed with four pbotonic-PCM cells to represent each 32-bit weight and four ADC samples to read the result which is reconstructed with bit-shifting in the control unit on the electrical layer. Typical deep learning models cannot fully fit in only photonic PCM cells. These cells are considered a medium for computation. Each LSPA chip is equipped with 32MiB SRAM and connected to 32GiB HBM for data storage as [56]. The adopted weight-stationary dataflow minimizes data movement to 1 hop maximum but does not completely eliminate data movement. In the presence

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2025.3591812

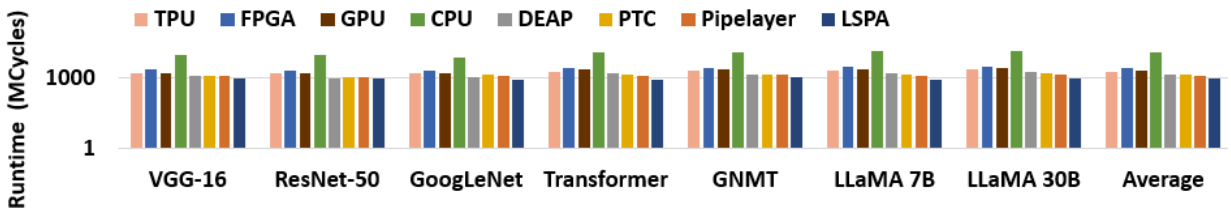


Fig. 12. Execution time comparison for training.

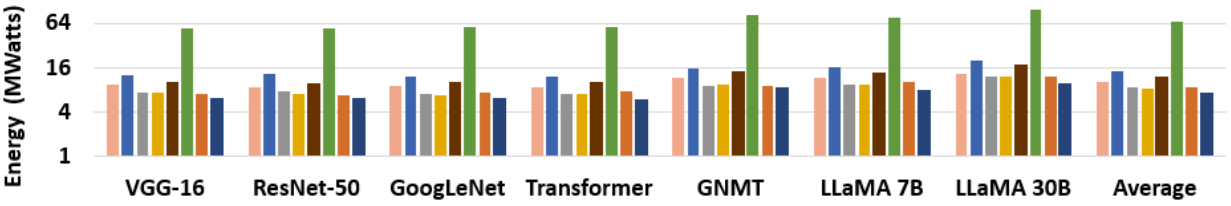


Fig. 13. Energy comparison for training.

of a mismatch between hardware resource demand and provision, tiling operations with multiple LSPA stacks are required as described in Section IV.

Baseline TPU [80] is a digital systolic-array-based computing system with 8-bit data width. We extend it to support 16-bit training at the cost of lowering its throughput. Baseline Pipelayer [10] is an analog memristor-based electrical computing system with 16-bit data width for training. Baseline DEAP [16] is an analog MRR-based photonic computing system with a 6-bit data width. We raise its data width to 16-bit for training tasks by performing a multiple-accumulate operation on multiple MRRs. Baseline PTC is an analog photonic-PCM-based photonic computing system with a 5-bit data width [23]. We take an approach similar to [10] to raise its data width.

C. Evaluation Benchmarks

We choose six DNN models for evaluation: VGG-16 [3], ResNet-50 [1], GoogLeNet [4], using the ImageNet dataset, as we ll as Google Neural Machine Translation (GNMT) [78] using the WMT16 EN-DE dataset, Transformer [79] using the WMT17 EN-DE dataset and LLaMA 7B and LLaMA 30B [81] using the SuperGLUE dataset [82]. VGG-16 has 16 weight layers, 13 convolutional and 3 fully connected, which are organized into 5 convolutional blocks each followed by a max-pooling layer for a total of 21 layers and 138 million parameters [3]. Resnet-50 which has 50 layers in total including convolutional layers, batch normalization, ReLU activations, 16 residual blocks, and 25.6 million parameters [1]. GoogLeNet has 22 layers consisting of multiple inception modules that each have multiple convolutional layers and a pooling layer and a total of 6.8 million parameters. GNMT has a total of 16 layers, 8 in the encoder and 8 in the decoder, and 1 billion parameters [78]. The Transformer model used is the original Transformer model with 6 encoder layers and 6 decoder layers and 213 million parameters [79]. LLaMA 7B has 32 transformer layers and 7 billion parameters [81], and LLaMA 30B has 60 transformer layers with 30 billion parameters [81]. Typical deep learning models cannot fully fit in only photonic PCM cells. These cells are considered a medium for computation. Each LSPA chip is equipped with 32MiB SRAM and connected to 32GiB HBM for data storage as [56]. The adopted weight-stationary dataflow minimizes data movement to 1 hop maximum but does not completely eliminate data movement. In the presence of a mismatch between hardware resource demand and provision, tiling operations with multiple LSPA stacks are required as described in Section IV.

VII. Experiment Results

A. DNN Training

Execution Time: Figure 12 shows the execution time comparison of LSPA against other baseline architectures when assuming a mini-batch size of 128. LSPA reduces the average execution time for DNN training by 52% compared to TPU, 27% compared to Pipelayer, 34% compared to DEAP, 34% compared to PTC, 65% comopared to the FPGA, 56% compared to the GPU, and 92% compared to the CPU. In general, LSPA reduces training execution time by 51% on average.

Energy Consumption: Figure 13 shows the energy consumption comparison of LSPA against other baseline architectures when assuming a mini-batch size of 128. LSPA reduces average training energy consumption by 31% compared to the TPU, 18% compared to Pipelayer, 17% compared to DEAP, 16% compared to PTC, 51% compared to the FPGA, 41% compared to the GPU, and 90% compared to the CPU. In general, LSPA reduces training energy consumption by 38% on average. The reduced execution time and energy of LSPA is attributed to the high speed and high computation density of the unique 3D photonic architecture. Other analog computing including Pipelayer (resistive memory technology-based), DEAP-CNN (MRR-based), and PTC (photonic-PCM based) require more off-chip DRAM accesses and tuning operations for their respective analog devices as they accommodate both the original and the transposed weight matrices needed for training. LSPA maintains a significant number of weights encoded in the photonic-PCM cells and reuses weights in both forward and backward passes during training, greatly reducing

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2025.3591812

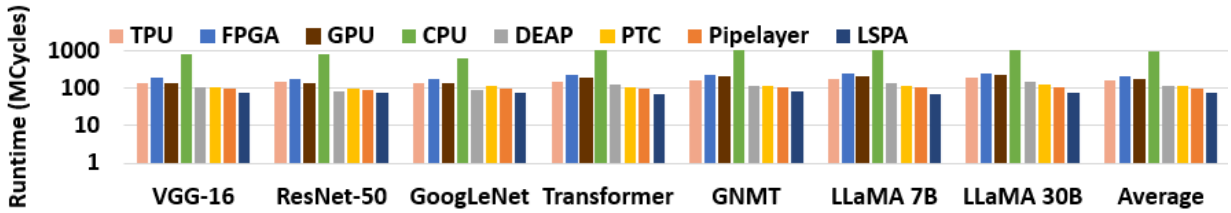


Fig. 14. Execution time comparison for inference.

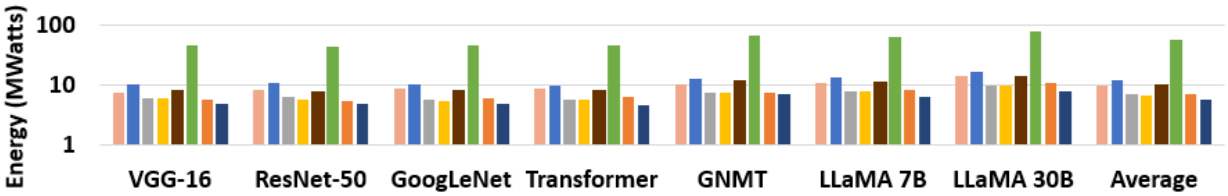


Fig. 15. Energy comparison for inference.

TABLE III
COMPUTATION DENSITY AND ENERGY EFFICIENCY

Architecture	TOPS/mm^2	TOPS/Watt
TPU [56]	0.69	1.62
FPGA	0.06	0.10
Pipelayer [10]	3.56	2.12
DEAP [16]	10.29	2.45
PTC [23]	20.51	5.13
GPU	0.125	0.17
CPU	0.030	0.06
LSPA-16	47.61	11.58
LSPA-8	184.37	41.45

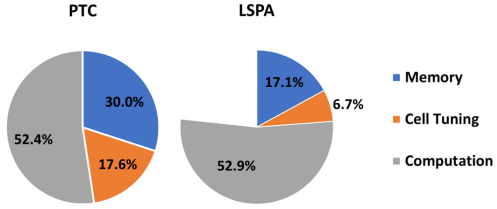


Fig. 16. Execution time breakdown comparison between PTC and LSPA when performing stochastic gradient training of VGG-16 model.

DRAM accesses and tuning time which reduces execution time more significantly than energy consumption. The CPU baseline exhibits significantly higher execution time and energy consumption compared to specialized accelerators due to its general-purpose architecture. Unlike GPUs or photonic accelerators, the CPU lacks highly parallel compute units and tensor cores optimized for matrix operations, resulting in lower throughput and longer runtimes for DNN workloads.

B. DNN Inference

Execution Time: The execution time comparison of LSPA against other baseline architectures for inference is shown in Figure 14. LSPA reduces the average execution time of DNN inference by 53% compared to the TPU, 26% compared to Pipelayer, 34% compared to DEAP, 33% compared to PTC, 65% compared to the FPGA, 56% compared to the GPU, and 92% compared to the CPU. In general, LSPA reduces inference execution time by 51% on average.

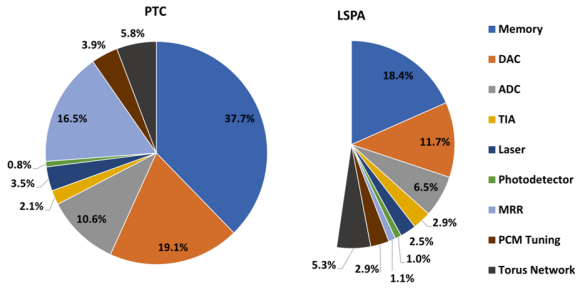


Fig. 17. Energy consumption breakdown comparison between PTC and LSPA when performing stochastic gradient training of VGG-16 model.

Energy Consumption: The energy consumption comparison of LSPA against other baseline architectures for inference is shown in Figure 15. On average, LSPA reduces inference energy consumption by 41% compared to TPU, 19% compared to Pipelayer, 17% compared to DEAP, 16% compared to PTC, 35% compared to the FPGA, 17% compared to the GPU, and 51% compared to the CPU. In general, LSPA reduces inference energy consumption by 28% on average. Similar to training, high speed computation of photonic devices and high computation density because of LSPA's parallelizable 3D architecture are mainly responsible for the reduction in execution time and energy. Performance is further optimized by a weight stationary dataflow that minimizes high latency and high energy photonic-PCM tuning.

C. Breakdown Analysis

Execution Time Breakdown: Figure 16 shows the execution time breakdowns of the PTC baseline and LSPA when performing training on the VGG-16 model with the stochastic gradient descent approach. Breakdown analysis was done with PTC since it is the only other architecture evaluated based on the photonic-PCM technology. 30% and 18% of the overall execution time of the PTC baseline are utilized for off-chip DRAM accesses and tuning of photonic-PCM cells. PTC requires more off-chip DRAM accesses and photonic-PCM cell tuning operations as it accommodates both the original and the transposed weight matrices. LSPA takes a similar

TABLE IV
TRAINING ACCURACY ALONG WITH THERMAL DRIFT EFFECTS AT 0% (I), 0.5% (II), AND 1% (III) ARE ADDED.

	FP32	BFloat 16	LSPA-16			LSPA-8		
			I	II	III	I	II	III
VGG-16	71.5	71.0	70.5	70.5	70.1	69.9	69.9	69.8
ResNet-50	76.2	75.5	75.1	75.1	75.0	74.7	74.7	74.5
GoogLeNet	69.6	69.2	68.3	68.3	68.2	68.1	68.1	67.5
GNMT	26.8	26.8	25.9	25.9	25.7	24.5	24.5	24.1
Transformer	28.2	28.2	27.7	27.7	27.5	26.3	26.3	25.6
LLaMA 7B	76.9	76.3	74.3	73.6	74.3	73.9	73.2	73.0
LLaMA 30B	70.3	69.9	67.6	66.9	67.5	66.7	67.4	66.2

amount of execution time for actual computations compared to PTC. However, the time required for off-chip DRAM accesses (17%) and photonic-PCM cell tuning operations (7%) are significantly lower. This is because LSPA has a notable fraction of weights encoded in the photonic-PCM cells and because in LSPA photonic-PCM weights are reused in both forward and backward passes during training.

Energy Consumption Breakdown: Figure 17 shows the energy consumption breakdowns of the PTC baseline and LSPA architecture when performing training on the VGG-16 model with the stochastic gradient descent approach. The Off-chip DRAM, input vector generation (including DACs and MRRs), and output vector generation (including photodetectors, TIAs, and ADCs) take 38%, 35%, and 13% of the overall energy consumption in the PTC baseline, respectively. The Off-chip DRAM, input vector generation (including DACs and MRRs), and output vector generation (including photodetectors, TIAs, and ADCs) take 18%, 23%, and 10% of the overall energy consumption in the PTC baseline, respectively. LSPA requires less energy for off-chip DRAM and input vector generation because the input features (forward pass) and loss gradients (backward pass) are reused for dot product operations with weights encoded in different columns of photonic-PCM cells in the dot product array of LSPA.

Electric vs. Photonic The area and power overhead of the electrical layers include memory, DACs, ADCs, TIAs, and the torus network. As shown in Figure 17, the electrical layer takes up the majority of energy consumption at 87.6%. The components on active and passive photonic layers including lasers, photodetectors, MRR modulation and PCM tuning take up only 12.4% of energy consumption. This disparity highlights the advantage of photonics in reducing power consumption, thereby enhancing the energy efficiency and overall performance of DNN accelerators like LSPA.

D. Computation Density and Energy Efficiency

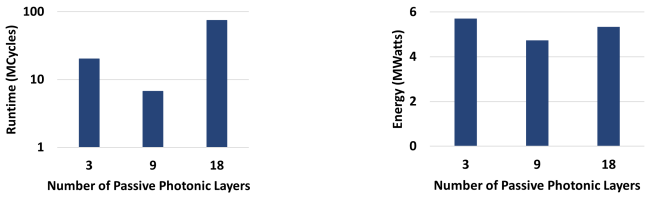
We compare the computation density (measured by TOPS/mm^2) and energy efficiency (measured by TOPS/Watt) of LSPA and previous electrical and photonic architectures in Table III. Please note that we assume $5nm$ technology for all electrical architectures for a fair comparison. We make the following observations: (1) conventional electrical architectures like TPU and Pipelayer achieve relatively low computation density due to the constraint on operating frequency; (2) MRR-based photonic architecture DEAP achieves higher computation density, albeit large optical device footprint due to much higher operating frequency, but suffers from low energy efficiency due to excess tuning power for temperature and process variation compensation; (3) photonic-PCM-based architecture PTC and LSPA achieve higher computation density and energy efficiency, wherein LSPA outperforms PTC due to the parallelism in vertical stacked passive layers. LSPA-8 achieves a much higher computation density and energy efficiency by trading off with accuracy. Theoretically, LSPA-8 would have $4\times$ the computation density and energy efficiency than LSPA-16, but it is only $3.87\times$ and $3.58\times$ respectively because area limitations of $400mm^2$ restrict the electrical components (DACs, ADCs, and TIAs) needed for 16-bit reconstruction.

E. Training Accuracy

LSPA-16 configuration is employed for performance and energy efficiency evaluations against its digital counterpart, TPU [56], which utilizes a customized 16-bit floating-point data format called BFloat16. While photonic-PCM offers significant advantages in terms of parallelism and energy efficiency, its reliability under varying thermal conditions remains a critical concern. This is due to the intrinsic behavior of GST-based photonic-PCM which store weights through phase transitions between crystalline and amorphous states. These state transitions are inherently sensitive to temperature fluctuations, leading to what is commonly referred to as thermal drift. In photonic-PCM systems, thermal drift manifests as changes in the refractive index with temperature, around 0.001 to 0.01 per degree Celsius [83]. Even minor deviations in the refractive index can affect the optical interference patterns used to perform MAC operations, potentially degrading computational accuracy. As deep learning accelerators must operate in diverse and sometimes thermally unstable environments, evaluating the impact of temperature-induced noise is essential for assessing the robustness and practical viability of LSPA in real-world applications.

To capture this effect, we introduce thermal drift as a source of random noise in our simulations evaluating model training accuracy under three scenarios: 0%, 0.05% and 1% thermal noise. Table IV summarizes the top-1 accuracy for VGG-16, ResNet-50, and GoogLeNet trained on the ImageNet dataset under various numerical formats, including 32-bit floating-point FP32, 16-bit customized floating-point BFloat16, 16-bit fixed-point, and 8-bit fixed-point (resolution of a single photonic PCM cell). In addition we report the Bilingual Evaluation Understudy (BLEU) scores for Transformer on the WMT17 EN-DE dataset and GNMT on the WMT16 EN-DE dataset, as well as the accuracy of LLaMA 7B and LLaMA 30B on

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2025.3591812



(a) (b)

Fig. 18. Runtime (a) and Energy (b) performance of LSPA-16 on Resnet-50 inference with a varying number of passive photonic layers.

the SuperGLUE benchmark. The results indicate that the 16-bit fixed-point format incurs only minor accuracy degradation compared to FP32 and BFloat16 with a maximum loss of 2.4% and 2.8% for LLaMA 30B respectively, even in the presence of 1% thermal noise. However, when using the 8-bit fixed-point format - where each weight is represented by a single photonic-PCM cell - accuracy losses increase to 3.7% and 4.1% for LLaMA 30B respectively. While this format significantly reduces hardware cost, it highlights the trade-off between resolution and rebustness in thermally sensitive photonic computing platforms.

F. Study on Number of Passive Photonic Layers

By stacking a configurable number of passive photonic layers, LSPA's architecture is flexible and scalable to meet various application needs. To study the impact of varying the number of passive photonic layers performance, we analyze energy and latency of performing inference of Resenet-50 on LSPA-16 with 3, 9, and 18 passive photonic layers. Keeping in mind that fabricating and aligning 10 layers in silicon photonics accurately is already feasible [60], [61], [62]. Continued innovation in materials, design methodologies, and integration techniques is expected to expand the practical limits of passive photonic layer integration in the near future [84], [85]. With 3 passive photonic layers, the number of LSPA stacks increases by $1.88\times$ compared to the number of LSPA stacks required when using 9 passive photonic layers, for the same $400\ \mathrm{mm}^2$ die area limit. Conversely, when using 18 passive photonic layers, the number of LSPA stacks decreases by $1.69\times$ compared to the 9-layer configuration. Increasing the number of layers per LSPA stack enables greater parallelism and higher compute density but also exacerbates challenges such as thermal hotspots, thermal crosstalk, and elevated data movement requirements. As shown in Figure 18, when using 18 passive photonic layers, the higher number of stacks results in increased memory fetches during photonic-PCM weight reprogramming, leading to higher latency. On the other hand, with only 3 passive layers, both energy consumption and runtime are higher than in the 9-layer configuration, due to limited parallelism and the need for more frequent photonic-PCM reprogramming. Overall, increasing the number of passive photonic layers improves computational density but introduces diminishing returns beyond a certain point due to thermal and memory bottlenecks. The 9-layer configuration provides an optimal trade-off between parallelism, energy efficiency, and runtime performance.

VIII. Conclusions

We propose LSPA, a novel 3D accelerator architecture for DNN training, leveraging photonic-PCM technology. LSPA is designed to optimize DNN inference and training workloads by enabling data multicast across two dimensions and data accumulation across three dimensions, aligning with the inherent computational patterns of DNN workloads. By harnessing additional computational parallelism in the frequency domain and implementing optimized pipeline scheduling, LSPA facilitates the concurrent execution of forward and backward passes within each batch. This significantly reduces expensive data movement overhead and minimizes the need for frequent re-programming of photonic-PCM cells. Simulation results demonstrate that LSPA outperforms state-of-the-art accelerator architectures, achieving up to 92% speedup in terms of execution time and up to 90% reduction in energy consumption.

IX. Acknowledgements

This research was partially supported by NSF grants CCF 2311543, 23224644, CCF 213946, CCF-1936794, CCF-2324645, and CCF-2311544. We sincerely thank the anonymous reviewer for their excellent and constructive feedback.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[5] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai, "Metal–oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.

[6] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature nanotechnology*, vol. 8, no. 1, pp. 13–24, 2013.

[7] F. Alibart, E. Zamanidoost, and D. B. Strukov, "Pattern classification by memristive crossbar circuits using ex situ and in situ training," *Nature communications*, vol. 4, no. 1, p. 2072, 2013.

[8] X. Liu, M. Mao, B. Liu, H. Li, Y. Chen, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu *et al.*, "Reno: A high-efficient reconfigurable neuromorphic computing accelerator design," in *Proceedings of the 52nd Annual Design Automation Conference*, 2015, pp. 1–6.

[9] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.

[10] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *2017 IEEE international symposium on high performance computer architecture (HPCA)*. IEEE, 2017, pp. 541–552.

[11] O. Krestinskaya, K. N. Salama, and A. P. James, "Learning in memristive neural network architectures using analog backpropagation circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 719–732, 2018.

[12] Y. Luo and S. Yu, "Accelerating deep neural network in-situ training with non-volatile and volatile memory based hybrid precision synapses," *IEEE Transactions on Computers*, vol. 69, no. 8, pp. 1113–1127, 2020.

[13] K. Prabhu, A. Gural, Z. F. Khan, R. M. Radway, M. Giordano, K. Koul, R. Doshi, J. W. Kustin, T. Liu, G. B. Lopes *et al.*, "Chimera: A 0.92-tops, 2.2-tops/w edge ai accelerator with 2-mbyte on-chip foundry resistive ram for efficient training and inference," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1013–1026, 2022.

[14] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.

[15] C. Silvano, D. Ielmini, F. Ferrandi, L. Fiorin, S. Curzel, L. Benini, F. Conti, A. Garofalo, C. Zambelli, E. Calore *et al.*, "A survey on deep learning hardware accelerators for heterogeneous hpc platforms," *ACM Computing Surveys*, 2023.

[16] V. Bangari, B. A. Marquez, H. Miller, A. N. Tait, M. A. Nahmias, T. F. De Lima, H.-T. Peng, P. R. Prucnal, and B. J. Shastri, "Digital electronics and analog photonics for convolutional neural networks (deap-cnns)," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–13, 2019.

[17] G. T. Reed, G. Mashanovich, F. Y. Gardes, and D. Thomson, "Silicon optical modulators," *Nature photonics*, vol. 4, no. 8, pp. 518–526, 2010.

[18] D. A. Miller, "Device requirements for optical interconnects to silicon chips," *Proceedings of the IEEE*, vol. 97, no. 7, pp. 1166–1185, 2009.

[19] J. Feldmann, M. Stegmaier, N. Gruhler, C. Ríos, H. Bhaskaran, C. D. Wright, and W. Pernice, "Calculating with light using a chip-scale all-optical abacus," *Nature communications*, vol. 8, no. 1, p. 1256, 2017.

[20] C. Ríos, M. Stegmaier, P. Hosseini, D. Wang, H. Bhaskaran, and C. D. Wright, "Integrated all-photonic non-volatile multi-level memory," *Nature Photonics*, vol. 9, no. 11, pp. 725–732, 2015.

[21] S. Kim, G. W. Burr, W. Kim, and S.-W. Nam, "Phase-change memory cycling endurance," *MRS Bulletin*, vol. 44, no. 9, pp. 710–714, 2019.

[22] M. Wuttig and N. Yamada, "Phase-change materials for rewriteable data storage," *Nature Materials*, vol. 6, no. 11, pp. 824–832, 2007.

[23] 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.

[24] K. Faraj *et al.*, "Phase change material thermal energy storage systems for cooling applications in buildings: a review," *Renewable and Sustainable Energy Reviews*, vol. 119, p. 109579, 2020.

[25] J. Qiu, X. Fan, Y. Shi, S. Zhang, X. Jin, W. Wang, and B. Tang, "Thermally conductive composite phase change materials with excellent thermal management capability for electronic devices," *Journal of Materials Chemistry A*, vol. 7, p. 2137121377, 2019.

[26] L. Yang, R. M. Radway, Y.-H. Chen, T. F. Wu, H. Liu, E. Ansari, V. Chandra, S. Mitra, and E. Beigné, "Three-dimensional stacked neural network accelerator architectures for ar/vr applications," *IEEE Micro*, vol. 42, no. 6, pp. 116–124, 2022.

[27] J. Kim, L. Zhu, H. M. Torun, M. Swaminathan, and S. K. Lim, "Micro-bumping, hybrid bonding, or monolithic? a ppa study for heterogeneous 3d ic options," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1189–1194.

[28] X. Hu, Y. Xu, Y. Hu, and Y. Xie, "Swimminglane: A composite approach to mitigate voltage droop effects in 3d power delivery network," in *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014, pp. 550–555.

[29] S. Ravichandran, V. Smet, M. Swaminathan, and R. Tummala, "Demonstration of glass-based 3d package architectures with embedded dies for high performance computing," in *2022 IEEE 72nd Electronic Components and Technology Conference (ECTC)*, 2022, pp. 1114–1120.

[30] Graphcore, "The graphcore ipu: A new processor for machine intelligence," 2020. [Online]. Available: https://www.graphcore.ai/white-papers

[31] D. Ingerly, S. Amin, L. Aryasomayajula, A. Balankutty, D. Borst, A. Chandra, K. Cheemalapati, C. Cook, R. Criss, K. Enamul *et al.*, "Foveros: 3d integration and the use of face-to-face chip stacking for logic devices," in *2019 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2019, pp. 19–6.

[32] D. Yu, "Tsmc packaging technologies for chiplets and 3d," 2021, presented by Dr. Douglas Yu, R&D Vice President and TSMC Distinguished Fellow.

[33] TSMC, "Tsmc 3dfabric technologies," https://3dfabric.tsmc.com/english/dedicatedFoundry/technology/3DFabric.htm, 2024, accessed: 2024-11-22.

[34] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, "Silicon microring resonators," *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, 2012.

[35] M. J. Filipovich, Z. Guo, M. Al-Qadasi, B. A. Marquez, H. D. Morison, V. J. Sorger, P. R. Prucnal, S. Shekhar, and B. J. Shastri, "Silicon photonic architecture for training deep neural networks with direct feedback alignment," *Optica*, vol. 9, no. 12, pp. 1323–1332, 2022.

[36] H. Jung, K. Y. Fong, C. Xiong, and H. X. Tang, "Electrical tuning and switching of an optical frequency comb generated in aluminum nitride microring resonators," *Optics letters*, vol. 39, no. 1, pp. 84–87, 2014.

[37] H. Zhang, L. Zhou, J. Xu, L. Lu, J. Chen, and B. Rahman, "Silicon microring resonators tuned with gst phase change material," in *2018 Asia Communications and Photonics Conference (ACP)*. IEEE, 2018, pp. 1–3.

[38] M. J. Filipovich, Z. Guo, B. A. Marquez, H. D. Morison, and B. J. Shastri, "Training deep neural networks in situ with neuromorphic photonics," in *2020 IEEE Photonics Conference (IPC)*. IEEE, 2020, pp. 1–2.

[39] J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. H. Pernice, "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature*, vol. 569, no. 7755, pp. 208–214, 2019.

[40] C. Wu, H. Yu, S. Lee, R. Peng, I. Takeuchi, and M. Li, "Programmable phase-change metasurfaces on waveguides for multimode photonic convolutional neural network," *Nature communications*, vol. 12, no. 1, p. 96, 2021.

[41] H. Zhu, J. Gu, C. Feng, M. Liu, Z. Jiang, R. T. Chen, and D. Z. Pan, "Elight: Toward efficient and aging-resilient photonic in-memory neurocomputing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 3, pp. 820–833, 2022.

[42] D. Dang, B. Lin, and D. Sahoo, "Litecon: An all-photonic neuromorphic accelerator for energy-efficient deep learning," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 19, no. 3, pp. 1–22, 2022.

[43] I. Giannopoulos, A. Sebastian, M. Le Gallo, V. P. Jonnalagadda, M. Sousa, M. Boon, and E. Eleftheriou, "8-bit precision in-memory multiplication with projected phase-change memory," in *2018 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2018, pp. 27–7.

[44] J. Zhou, T. Xu, S. Wang, T. C. Chong, and S. R. P. Silva, "Low-power, nanosecond switching in plasmonic phase-change materials," *Advanced Optical Materials*, vol. 7, no. 6, p. 1801175, 2019.

[45] W. Zhou, B. Dong, N. Farmakidis, and et al., "In-memory photonic dot-product engine with electrically programmable weight banks," *Nature Communications*, vol. 14, p. 2887, 2023.

[46] C. Ros, P. Hosseini, R. A. Taylor, and H. Bhaskaran, "Advanced photonic phase change materials and memories," *Nature Photonics*, vol. 8, no. 12, pp. 104–113, 2014.

[47] X. Wang, H. Qi, X. Hu, Z. Yu, S. Ding, Z. Du, and Q. Gong, "Advances in photonic devices based on optical phase-change materials," *Molecules*, vol. 26, no. 9, p. 2813, 2021.

[48] Z. Lu, H. Yun, Y. Wang, Z. Chen, F. Zhang, N. A. Jaeger, and L. Chrostowski, "Broadband silicon photonic directional coupler using asymmetric-waveguide based phase control," *Optics express*, vol. 23, no. 3, pp. 3795–3808, 2015.

[49] J. T. Bessette and D. Ahn, "Vertically stacked microring waveguides for coupling between multiple photonic planes," *Optics Express*, vol. 21, no. 11, pp. 13 580–13 591, 2013.

[50] M. Sumetsky, "Vertically-stacked multi-ring resonator," *Optics Express*, vol. 13, no. 17, pp. 6354–6375, 2005.

[51] B. E. Little, S. T. Chu, H. A. Haus, J. Foresi, and J.-P. Laine, "Microring resonator channel dropping filters," *Journal of lightwave technology*, vol. 15, no. 6, pp. 998–1005, 1997.

[52] P. Dong, R. Lively, N. N. Feng, W. M. Green, J. Michel, and L. C. Kimerling, "Low loss silicon waveguides for application of optical interconnects," *Optics Express*, vol. 16, no. 8, pp. 5947–5954, 2008.

[53] F. Xia, L. Sekaric, and Y. Vlasov, "Ultracompact optical buffers on a silicon chip," *Nature Photonics*, vol. 1, no. 1, pp. 65–71, 2007.

[54] Q. Xu, J. Shakya, and M. Lipson, "Micrometer-scale silicon electro-optic modulator," *Nature*, vol. 435, no. 7040, pp. 325–327, 2005.

[55] C. Xiang, W. Jin, O. Terra, and et al., "3d integration enables ultralow-noise isolator-free lasers in silicon photonics," *Nature*, vol. 620, pp. 78–85, 2023.

[56] N. Jouppi, G. Kurian, S. Li, P. Ma, R. Nagarajan, L. Nai, N. Patil, S. Subramanian, A. Swing, B. Towles, C. Young, X. Zhou, Z. Zhou, and D. Patterson, "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings," in *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, June 2023, pp. 1–14.

[57] G. M. Marega, H. G. Ji, Z. Wang, M. Tripathi, A. Radenovic, and A. Kis, "Large-scale integrated vector-matrix multiplication processor based on monolayer mos," *Nature Electronics*, vol. 6, pp. 991–998, 2023.

[58] M. Notaros, T. Dyer, A. Garcia Coleto, A. Hattori, K. Fealey, S. Kruger, and J. Notaros, "Mechanically-flexible wafer-scale integrated-photonics fabrication platform," *Scientific Reports*, vol. 14, no. 1, p. 10623, 2024.

[59] K. T. Sullivan and et al., "Wafer-scale fabrication of 3d integrated photonic devices," *IEEE Transactions on Electron Devices*, vol. 68, no. 7, pp. 3283–3288, 2021.

[60] P. Raghavan, M. Schuster, Y. Wu, Z. Chen *et al.*, "Advances in google neural machine translation," in *Proceedings of the Association for Computational Linguistics (ACL)*. ACL, 2021.

[61] Y. Liu, X. Sun, Z. Zhang *et al.*, "Silicon photonics for advanced communications and computing: Perspectives and challenges," *Optica*, vol. 8, no. 11, pp. 1365–1384, 2021.

[62] S. Tanzilli *et al.*, "On the genesis of photonic integrated circuits," *Nature Photonics*, vol. 12, no. 4, pp. 239–246, 2018.

[63] M. Al-Qadasi, L. Chrostowski, B. Shastri, and S. Shekhar, "Scaling up silicon photonic-based accelerators: Challenges and opportunities," *APL Photonics*, vol. 7, no. 2, 2022.

[64] H. Sun, Q. Qiao, Q. Guan, and G. Zhou, "Silicon photonic phase shifters and their applications: A review," *Micromachines*, vol. 13, no. 9, p. 1509, 2022.

[65] B. Wu, S. Liu, J. Cheng, W. Dong, H. Zhou, J. Dong, M. Li, and X. Zhang, "Real-valued optical matrix computing with simplified mzi mesh," *Intelligent Computing*, vol. 2, p. 0047, 2023.

[66] R. Paschotta, "Photodetection: Optical and electrical powers," 2009, accessed: 2025-02-26. [Online]. Available: https://www.rp-photonics.com/spotlight_2009_11_13.html

[67] P. Caragiulo, O. E. Mattia, A. Arbabian, and B. Murmann, "A compact 14 gs/s 8-bit switched-capacitor dac in 16 nm finfet cmos," in *2020 IEEE Symposium on VLSI Circuits*. IEEE, 2020, pp. 1–2.

[68] J. Liu, M. Hassanpourghadi, and M. S.-W. Chen, "A 10gs/s 8b 25fj/cs 2850um 2 two-step time-domain adc using delay-tracking pipelined-sar tdc with 500fs time step in 14nm cmos technology," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 160–162.

[69] M. Rakowski, Y. Ban, P. De Heyn, N. Pantano, B. Snyder, S. Balakrishnan, S. Van Huylenbroeck, L. Bogaerts, C. Demeurisse, F. Inoue *et al.*, "Hybrid 14nm finfet-silicon photonics technology for low-power tb/s/mm 2 optical i/o," in *2018 IEEE Symposium on VLSI Technology*. IEEE, 2018, pp. 221–222.

[70] H. Zhu, J. Gu, H. Wang, Z. Jiang, Z. Zhang, R. Tang, C. Feng, S. Han, R. T. Chen, and D. Z. Pan, "Dota: A dynamically-operated photonic tensor core for energy-efficient transformer accelerator," *arXiv preprint arXiv:2305.19533*, 2023.

[71] Y. Li, A. Louri, and A. Karanth, "Spacx: Silicon photonics-based scalable chiplet accelerator for dnn inference," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2022, pp. 831–845.

[72] Z. Huang, C. Li, D. Liang, K. Yu, C. Santori, M. Fiorentino, W. Sorin, S. Palermo, and R. G. Beausoleil, "25 gbps low-voltage waveguide si–ge avalanche photodiode," *Optica*, vol. 3, no. 8, pp. 793–798, 2016.

[73] Y. Li, K. Wang, H. Zheng, A. Louri, and A. Karanth, "Ascend: A scalable and energy-efficient deep neural network accelerator with photonic interconnects," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 7, pp. 2730–2741, 2022.

[74] Y. Li, A. Louri, and A. Karanth, "A high-performance and energy-efficient photonic architecture for multi-dnn acceleration," *IEEE Transactions on Parallel and Distributed Systems*, vol. 35, no. 1, pp. 46–58, 2023.

[75] W. Won, T. Heo, S. Rashidi, S. Sridharan, S. Srinivasan, and T. Krishna, "Astra-sim2. 0: Modeling hierarchical networks and disaggregated systems for large-model training at scale," in *2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2023, pp. 283–294.

[76] I. HPC. (2024, June) Intel demonstrates integrated optical i/o chiplet. Accessed: 2024-11-21. [Online]. Available: https://insidehpc.com/2024/06/intel-demonstrates-integrated-optical-i-o-chiplet/

[77] O. C. News. (2024, April) Ofc 2024: Intel develops 4tbps sipho oci chiplet. Accessed: 2024-11-21. [Online]. Available: https://opticalconnectionsnews.com/2024/04/ofc-2024-intel-develops-4tbps-sipho-oci-chiplet/

[78] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[80] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, 2017, pp. 1–12.

[81] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[82] A. Wang, A. Praja, J. Schwartz, S. Bowman, L. Dong, W. Hsu, L. Zettlemoyer, and M. Gardner, "Superglue: A stickier benchmark for general-purpose language understanding systems," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019. [Online]. Available: https://arxiv.org/abs/1905.10547

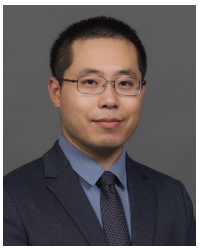
[83] X. Wang, H. Qi, X. Hu, Z. Yu, S. Ding, Z. Du, and Q. Gong, "Advances in photonic devices based on optical phase-change materials," *Molecules*, vol. 26, no. 9, p. 2813, 2021. [Online]. Available: https://doi.org/10.3390/molecules26092813

[84] B. Shen, R. Polson, and R. Menon, "Increasing the density of passive photonic-integrated circuits via nanophotonic cloaking," *Nature communications*, vol. 7, no. 1, p. 13126, 2016.

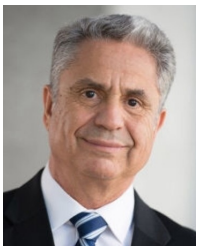
[85] P. Kaur, A. Boes, G. Ren, T. G. Nguyen, G. Roelkens, and A. Mitchell, "Hybrid and heterogeneous photonic integration," *APL photonics*, vol. 6, no. 6, 2021.

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2025.3591812

Juliana Curry received the B.S. degree in Electrical Engineering from The George Washington University in 2019. She is currently pursuing the Ph.D. degree in Computer Engineering at the George Washington University with a focus on photonic neural network accelerators. Her research interests include silicon photonics and analog computing, machine learning applications and acceleration, and sustainable and robust system design.



Yuan Li received the B.S. degree in Physics from the University of Science and Technology of China in China in 2010, the M.S. degree in Microelectronics from the University of Newcastle upon Tyne in the U.K. in 2011, as well as the Ph.D. degree in Computer Engineering from the George Washington University in the U.S. in 2022. His research interests include AI hardware acceleration, chiplet-based heterogeneous integration, and emerging technologies for computing and communication. More information can be found at https://gwuyuan.github.io/mysite/



Ahmed Louri is the David and Marilyn Karlgaard Endowed Chair Professor of Electrical and Computer Engineering at George Washington University, where he also directs the High-Performance Computing Architectures and Technologies Laboratory (HPCAT). Dr. Louri received a Ph.D. degree in Computer Engineering from the University of Southern California in 1988. From 1988 to 2015, he was a Professor of Electrical and Computer Engineering at the University of Arizona. From 2010 to 2013, Dr. Louri served as a program director in the US National Science Foundation's (NSF) Directorate for Computer and Information Science and Engineering (CISE). He directed the core computer architecture program and was on the management team of several cross-cutting programs. While at NSF, Dr. Louri initiated multidisciplinary research programs in several key areas of computer architecture, high-performance computing, sustainability, emerging technologies, resiliency, and security. Dr. Louri conducts research in the broad area of computer architecture and parallel computing, with emphasis on interconnection networks, scalable parallel computing systems, versatile and flexible computing systems, and power-efficient, reliable, and secure Network-on-Chips (NoCs) for multicore architectures. Recently, he has been concentrating on energy-efficient, reliable, and high-performance many-core architectures; accelerator-rich reconfigurable heterogeneous architectures; secure network-on-chips for multicores and SoCs; approximate computing and communications; machine learning techniques for efficient computing, memory, and interconnect systems; machine learning acceleration; heterogeneous manycore architectures & chiplet-based designs; emerging interconnect technologies (photonic, wireless, RF, hybrid) for multicore architectures and chip multiprocessors (CMPs); future parallel computing models and architectures (including convolutional neural networks, deep neural networks, and approximate computing); cloud-computing and data centers. He has published more than 200 refereed journal articles and peer-reviewed conference papers and is the inventor of several US and international patents. His early work explored optics unique properties to advance computing and communications. He was instrumental in bringing optical interconnects into mainstream research in computing and played a critical role in bridging the gap between the computer architecture and optics research communities. Dr. Louri is a Life Fellow of IEEE, and a recipient of numerous awards, including the 2020 IEEE McCluskey Technical Achievement Award and the 2024 IEEE Computer Society Golden Core Award. He currently serves as Editor-in-Chief of IEEE Transactions on Sustainable Computing.

Avinash Karanth received his Ph.D. and M.S. from The University of Arizona in August 2006 and May 2003 respectively. Presently, he is the Chair of the School of Electrical Engineering and Computer Science (EECS) at Ohio University. He is also Joseph K. Jachinowski Professor in the School of EECS where he leads the Technologies for Emerging Computer Architecture Laboratory (TEAL) at Ohio University. His research interests include Computer Architecture, Machine Learning, Hardware Accelerators, Network-on-Chips (NoCs), Emerging Technologies (nanophotonics, wireless), Hardware Security, and Exascale Networks. He has received the prestigious NSF CAREER Award in 2011, Presidential Research Scholar Award in 2017, Best Paper Award at the ICCD 2013 conference and his papers have been nominated for Best Paper at IEEE Design and Test in Europe (DATE) in 2019, IEEE Symposium on Network-on-Chips (NoCs) in May 2010 and IEEE Asia & South Pacific Design Automation Conference (ASP-DAC) in January 2009. His research has been sponsored by National Science Foundation (NSF), Air Force Research Lab (AFRL), Ohio Department of Higher Education (ODHE) and Advanced Micro Devices (AMD) grants. Further, he has published 100+ articles in peer-reviewed IEEE and ACM journals and conferences. He is an Associate Editor for IEEE Transactions on Computers and IEEE Transactions on Cloud Computing and he has been a co-Guest Editor for IEEE Transactions on Emerging Topics for Computing ('15-'16) and Journal of Parallel and Distributed (JPDC) ('10-'11). He was the co-Chair of the architecture track at IPDPS-2020 and vice-Chair of the EDA track at DAC-2021 and DAC-2022 conferences. Dr. Karanth has served on the Program Committee of HPCA (2019), DAC (2018-19), NoCs (2016-19), MPSoC (2014-2019) ACM Nanocom (2016), Hot Interconnects ('10,16,'17,'19) and external Program Committee for MICRO'12. He has served on multiple NSF panels and several departmental committees. He is the Senior Member of IEEE.

Razvan Bunescu Razvan Bunescu received the PhD degree in computer science from the University of Texas at Austin, in 2007, with a dissertation on machine learning methods for information extraction. He is an associate professor in computer science at University of North Carolina at Charlotte. His research interests lie in the general area of machine learning, with a focus on applications in computational linguistics, biomedical informatics, computer architecture, software engineering, and computational creativity. He has received the SIGSOFT Distinguished Paper Award at the 38th IEEE/ACM International Conference on Software Engineering (ICSE) in 2016. He is a member of IEEE and ACM.