An Electro-Photonic System for Accelerating Deep Neural Networks

CANSU DEMIRKIRAN and FURKAN ERIS, Boston University, USA
GONGYU WANG, JONATHAN ELMHURST, NICK MOORE, NICHOLAS C. HARRIS, and AYON BASUMALLIK, Lightmatter, USA
VIJAY JANAPA REDDI, Harvard University, USA
AJAY JOSHI, Boston University, USA
DARIUS BUNANDAR, Lightmatter, USA

The number of parameters in deep neural networks (DNNs) is scaling at about 5× the rate of Moore's Law. To sustain this growth, photonic computing is a promising avenue, as it enables higher throughput in dominant general matrix-matrix multiplication (GEMM) operations in DNNs than their electrical counterpart. However, purely photonic systems face several challenges including lack of photonic memory and accumulation of noise. In this article, we present an electro-photonic accelerator, ADEPT, which leverages a photonic computing unit for performing GEMM operations, a vectorized digital electronic application-specific integrated circuits for performing non-GEMM operations, and SRAM arrays for storing DNN parameters and activations. In contrast to prior works in photonic DNN accelerators, we adopt a system-level perspective and show that the gains while large are tempered relative to prior expectations. Our goal is to encourage architects to explore photonic technology in a more pragmatic way considering the system as a whole to understand its general applicability in accelerating today's DNNs. Our evaluation shows that ADEPT can provide, on average, 5.73× higher throughput per watt compared to the traditional systolic arrays in a full-system, and at least 6.8× and 2.5× better throughput per watt, compared to state-of-the-art electronic and photonic accelerators, respectively.

CCS Concepts: • **Hardware → Emerging optical and photonic technologies**; **Emerging architectures;**

Additional Key Words and Phrases: Deep learning accelerators, photonic computing

ACM Reference format:
Cansu Demirkiran, Furkan Eris, Gongyu Wang, Jonathan Elmhurst, Nick Moore, Nicholas C. Harris, Ayon Basumallik, Vijay Janapa Reddi, Ajay Joshi, and Darius Bunandar. 2023. An Electro-Photonic System for Accelerating Deep Neural Networks. *ACM J. Emerg. Technol. Comput. Syst.* 19, 4, Article 30 (September 2023), 31 pages.
https://doi.org/10.1145/3606949

Authors' addresses: C. Demirkiran, F. Eris, and A. Joshi, Boston University, 8 St Mary's St, Boston, Massachusetts 02215; emails: cansu@bu.edu, fe@bu.edu, joshi@bu.edu; G. Wang, J. Elmhurst, N. Moore, N. C. Harris, A. Basumallik, and D. Bunandar, Lightmatter, Boston, Massachusetts, USA; emails: gongyu@lightmatter.co, jon@lightmatter.co, nmoore@lightmatter.co, nick@lightmatter.co, ayon@lightmatter.co, darius@lightmatter.co; V. J. Reddi, Harvard University, Boston, Massachusetts, USA; email: vj@eecs.harvard.edu.

1550-4832/2023/09-ART30 $15.00
https://doi.org/10.1145/3606949

1 INTRODUCTION

Deep neural networks (DNNs) have shown to perform impressive humanlike tasks in a range of applications including image and video processing [43], diagnostic medical imaging [81], speech recognition [44], and conversational artificial intelligence [32]. OpenAI's study shows that modern DNN computational requirements have increased 300,000× from AlexNet (2012) to AlphaGo Zero (2018). This general trend is projected to continue as newerand larger DNN models emerge ever so often [7].

Consequently, a variety of solutions have been developed to support the growing compute requirements. These solutions include massively threaded **graphics processing units (GPUs)** [26, 28, 48], field-programmable gate arrays [34, 47, 85], and specialized **application-specific integrated circuits (ASICs)** [21, 23, 35]. While these solutions provide significant architectural and performance benefits for DNN execution, they are based on CMOS transistors—devices that no longer scale in area or energy consumption according to Moore's Law and Dennard Scaling [100].

As an alternative, there is growing interest in using photonic computing architectures for meeting the computational demands of DNNs. The idea of computing with light is not new and has been explored since the 1960s [15, 17, 79]. The advent of integrated photonics, in particular silicon photonics, which has seen widespread integration in commercial CMOS foundries alongside transistors on 300-mm wafers [38] has further propelled research in photonic computing. However, limitations around photonic information storage (no photonic memory) and weak photon-photon nonlinearities (no photonic transistor) make it difficult—if not impossible—to design a general-purpose fully photonic computing architecture. Prior art leverages the highly parallel and efficient linear transformations enabled by photonics to build specialized DNN accelerators with orders of magnitude improvements in speed and energy efficiency when computing **general matrix-matrix multiplication (GEMM)** and convolution operations [12, 71, 87, 88, 91, 92, 106], which accounts for more than 90% of the total number of operations within a DNN network [24].

In this article, we seek to calibrate the expectations of the photonic GEMM technology with respect to building a complete system, including the photonic and non-photonic components needed to make it all work. We set out to answer two key questions. First, *given that photonic accelerators still need electronics (for control, data storage, and nonlinearities), how do we build a complete electro-photonic accelerator architecture that is not bottlenecked by the slower electronics?* To answer this question, we present the microarchitecture of an electro-photonic accelerator called ADEPT, where we match the throughput of the electronic and photonic components. ADEPT comprises high-throughput photo-core(s), various data converters, custom vectorized electronic digital ASIC, and large electronic SRAM arrays. The photo-core is a scalable and highly-efficient photonic tensor core containing **Mach-Zehnder Interferometers (MZIs)** for GEMM operations. In the photo-core, we adopt a **weight stationary (WS)** approach where the weight matrix is programmed into the MZI array. The inputs are routed, one vector at a time, through digital-to-analog converter, **electrical-to-optical (O-E)** converter, the MZI array, O-E converter, and analog-to-digital converter. While the photo-core can handle GEMM operations (over 90% of the overall DNN operations), DNNs rely on a non-trivial amount of non-GEMM operations that are executed in the electrical domain. To match the throughput of the photonic and electronic components, we architect a highly vectorized electronic digital ASIC with multiple digital lanes, where each lane supports basic arithmetic operations that can be used for building more complex non-GEMM operations. To efficiently orchestrate the operations and maximize the performance of ADEPT, we pipeline GEMM and non-GEMM operations and use an efficient buffering scheme to minimize DRAM access overhead. Finally, we evaluate ADEPT in the context of a full system to understand the big picture.

The second question we set out to answer is *(2) how much are the electro-photonic accelerator systems better than purely electronic accelerator systems, when we consider the system as a whole, i.e., accelerator + memory + host processor + communication, running practical real-world applications?* To answer this question, we perform a head-to-head comparison of ADEPT with electronic **systolic arrays (SAs)** in terms of the full system throughput (in **inferences per second (IPS)**), power efficiency (in IPS/W), and power-area efficiency (in IPS/W·mm^2). We use the following three state-of-the-art neural networks from the MLPerf datacenter inference benchmarks [78] that represent a wide range of operations: ResNet-50 [43] for image classification on the ImageNet dataset [82], BERT-large [32] for **natural language processing (NLP)** on the SQuAD v1.1 [74] question-answering dataset, and RNN-T [44] as an LSTM-based speech recognition network on the LibriSpeech [69] speech audio dataset. Our analysis shows that, compared to SAs, ADEPT provides *4.89*×, *3.24*×, and *9.06*× better power efficiency for the full system for ResNet-50, BERT-large, and RNN-T networks, respectively. Compared to the state-of-the-art electronic accelerators, ADEPT performs at least *6.8*× better in terms of IPS/W. In addition, we perform a detailed comparison between ADEPT and current state-of-the-art photonic accelerators. Our analysis shows that compared to state-of-the-art photonic accelerators [61, 71, 91], ADEPT can provide more than *2.5*× better power efficiency for the same batch size and more than *8.3*× better power efficiency when the maximum batch size is used.

In summary, our work is the first to emphasize the importance of considering the entire system to understand the real benefit of the photonic GEMM cores for DNN inference. Our study shows that the impact of the electronic components in an electro-photonic accelerator system is not negligible. However, while an electro-photonic system may be bound by Amdahl's Law, it is still feasible—and beneficial—to build a balanced electronic-photonic system that leverages the highly efficient photonic computing medium. Our work aims to provide practical insights to the community and to encourage architects to explore photonic technology in a more pragmatic way without "missing the forest for the trees." Broadly, we show that while using photonics technology for computing is promising, claims of tera-inferences per second are not realistic when considering the system as a whole.

2 BACKGROUND AND RELATED WORK

This section provides an overview of different ways of accelerating GEMM operations using photonics, a discussion on optical devices and their efficiency and scalability, and an explanation of why performing non-linear operations in photonics is challenging. Last, we detail the evaluation coverage of the state-of-the-art photonic accelerator architectures to demonstrate how our work sheds new insights, specifically from the perspective of a complete system rather than just the accelerator.

2.1 Photonics for GEMM Acceleration

Photonics has been perceived as a promising medium for performing linear operations such as matrix-matrix multiplication—which is a key operation for DNN acceleration. In this section, we aim to provide a short review of the landscape of optical matrix-matrix multiplication. We can divide the existing methods broadly into three categories:

MZI-based methods: An MZI is a configurable photonic device that controls the interference of two light beams by adjusting the relative phase shift between the beams. Previous works based on MZIs [75, 88] are mainly based on the idea that any real-valued matrix can be decomposed into unitary and diagonal matrices by using **singular value decomposition (SVD)**. A universal linear network can then be composed of two universal unitary circuits and an additional column of attenuators as described by Miller [64]. By tuning the amount of phase shift in MZIs, the values can

be reprogrammed and a full MVM operation can be performed by passing modulated input signals through the MZI mesh. This type of GEMM operation has been demonstrated by Shen et al. [88] for vowel recognition application. Lightmatter's Mars chip [75] is another recent example that shows the applicability of the MZI mesh structure. MZI-based structures are widely used in photonic GEMM acceleration as they can effectively represent matrices and perform MVM operations. The disadvantages of such architectures include the large area footprint of MZIs and susceptibility to phase-noise corruption that can be mitigated by error-correction methods [11].

Micro-ring resonator (MRR)/Micro-disk– (MD) based methods: MRR- or MD-based methods typically leverage **wavelength division multiplexing (WDM)** where multiple optical signals, each with a different wavelength, are carried by the same waveguide. These accelerators adopt a **broadcast-and-weight (B&W)** approach for GEMM or MVM operations that was first proposed by Tait et al. [99] in 2014 and demonstrated in 2017 [98]. B&W method uses weighting banks comprised of MRRs or MDs as tunable filters. Each MRR in the bank tunes the signal that shares its resonant wavelength and the sum is collected at the end of the bank by a photodetector—which is effectively a vector dot product. MRR-based approaches are popularly used to accelerate DNN applications [12, 27, 61, 62, 97, 111]. The challenges of such architectures include inter-channel and intra-channel cross-talk and thermal stabilization.

Diffractive optics-based methods: In these methods, **diffractive optical elements (DOEs)** are used to perform MVMs. By encoding the weights of a DNN onto the DOEs, the input can be multiplied by the weight matrix in a single step using the interference patterns produced by the DOE. This method is used by Lin et al. [59] in designing all-optical diffractive deep neural networks as well as by Zhou et al. [114] for designing a diffractive processing unit. Another approach is to use diffractive lenses to implement convolution operations in a DNN. This technique typically uses a combination of lenses (e.g., Fourier lenses) and spatial light modulators to perform convolution [18, 66, 94, 112]. Optalysys combines silicon photonics and Fourier optics to accelerate CNNs, transformers, and fully homomorphic encryption [30, 108].

2.2 Efficiency and Scalability of Photonic Devices

The scalability and efficiency of a photonic GEMM core is dependent on the type of optical devices, the required bit precision, the operation frequency, and the noise sources. Each design presented in the previous works has different tradeoffs and unique limitations. In this section, we provide a brief comparison of MZIs and MRRs, two commonly used photonic devices, to explain our choice of MZIs over MRRs in our design. Our purpose is not to propose a more efficient photonic core than the existing works. Instead, we argue that as long as we are still dependent on electronic devices, which are much slower compared to photonic devices, regardless of the used optical device, the hybrid design will require a system-level evaluation approach to determine its use. In fact, the scalability and efficiency of photonic cores can be further improved as more efficient optical devices are developed. However, as the photonic core gets more efficient, the system-level analysis will be more and more important. Therefore, in this article, we aim to provide insights using an MZI-based design and these insights are applicable to other photonic device-based designs. In this section, our goal is to give a glimpse of the different design options available and their tradeoffs.

MRRs are typically smaller (with a dimension of $\sim$10 μm) than MZIs (with a dimension of $\sim$100 μm) and can provide a better power and area efficiency [6]. However, MZIs have been shown to achieve an extremely high extinction ratio (ER, which is a measure of how precise the light signals can be modulated by the photonic device) of greater than 60 dB [107]. The ER of MRRs is determined by how closely we can achieve critical coupling, which can be limited by the MRR's thermal stability [16]. State-of-the-art demonstrations of a single MRR have their measured ER at $<$25 dB [89]. MRRs with stabilization circuits have been demonstrated in

electro-photonic transceivers for communication [68, 95, 101]. Unfortunately, for DNN inference, we typically need 8-bit precision during computations, which is more than the 1 or 2 bits that are required for NRZ [76] or PAM-4 [96] keyings, respectively, used in communication. As a result, the stabilization circuitry will consume more area and power than what has been demonstrated. Nevertheless, recent works show that it is possible to increase the ER of MRRs by cascading multiple MRRs [27]. Additionally, it has been shown that the thermal cross-talk can be mitigated and tuning efficiency can be improved by several methods including placing air trenches [33], simultaneously controlling the actuators [63], and using photoconductive heaters [50].

In this article, we use the MZI mesh structure similarly to Reference [75]. In silicon photonics, the phase difference in MZIs is achieved by delaying light in one arm using various mechanisms, including the thermo-optic effect (~100s-kHz bandwidth) [42], mechanical effect (≥MHz bandwidth) [72], and electric-field induced electro-optic effect (≥GHz bandwidth) [102]. Recent studies show that NOEM phase shifters can operate at CMOS-compatible voltages (≤1.2 V) with an insertion loss of less than 0.04 dB and a modulation frequency of a few hundred MHz [10]. Compared to the lossy plasma dispersion effect [9] and slow thermo-optic effect [105], using NOEM phase shifters enable both low loss and relatively fast modulation. With a dataflow leveraging the high reuse of these devices, we can build large systems satisfying a low power budget and clock rates on the order of GHz.

2.3 Nonlinear Operations

Any nonlinear operation (e.g., nonlinear activation functions or conditional if-else statements) on the optical electromagnetic waves requires the use of nonlinear optical media [49]. Nonlinear optical activation function has previously been demonstrated using laser-cooled atoms, which absorb light up to some saturation intensity (higher intensity light is absorbed more) [116]. Saturable absorbers, where the amount of light absorbed decreases with increasing light intensity, have also been proposed as optical nonlinear activations [13, 88]. However, the practical implementation of these nonlinear optical activations remains challenging, especially since (1) they have not been miniaturized, and (2) repeated usage of the nonlinear activation function will decay the signal quickly.

Amplification-based nonlinear functions made out of semiconductor optical amplifiers in III-V materials, e.g., InP and InGaAs, can combat the loss described above [80]. In principle, an optical DNN accelerator can be built in the III-V platform itself [90], but researchers still prefer to use silicon photonics as it has been monolithically integrated with the CMOS transistors [38] needed for controlling the photonic components. Packaging the III-V module with a silicon photonics module poses a challenge to its feasibility. Even when a practical packaging solution is available, the amount of power needed to maintain the optical signal throughout the entire inference will increase exponentially with the number of neural network layers. We, therefore, conclude that optical nonlinearities are impractical today, and we choose to architect a system that performs these nonlinearities electronically.

2.4 Evaluation of State-of-the-Art Photonic Accelerator Architectures

In this section, we focus on the evaluation methodology of previous photonic accelerator works. Evaluation of several photonic tensor cores in prior works has been isolated from the system surrounding the photonic tensor cores [36, 87, 88, 106, 111]. While the performance numbers are impressive, these accelerators need to be viewed through the lens of a practical system. Some works have combined photonics with electronics [27, 61, 62, 71, 91, 92]. However, these works provided either a conceptual design or a partial system evaluation. Table 1 presents the state-of-the-art works on photonic DNN accelerators that provide a partial system evaluation. This summary sets

Table 1. Comparison against Other Photonic Accelerators

		Non-Photonic Components and Metrics Considered			Benchmarks		
Accelerator	Optical Element	Non-GEMM	On-chip Memory	Off-chip Memory	CNN	NLP	RNN
ADEPT	MZI	✔	✔	✔	✔	✔	✔
Albireo[91]	MRR+MZI	✗	✔	✗	✔	✗	✗
PIXEL[92]	MRR+MZI	✔	✗	✗	✔	✗	✗
PCNNA[62]	MRR	✗	✗	✗	✔	✗	✗
DNNARA[71]	MRR	✔	✔	✗	✔	✗	✗
Holy-Light[61]	MRR	✔	✔	✗	✔	✗	✗

the stage to discuss how one needs to systematically take a full-system view to understand the limits and opportunities for using photonic cores.

2.4.1 Compute vs. Memory. Small on-chip caches (on the order of hundreds of KBs used by previous works [71, 91, 92]) cannot hold large DNN models, input/output data, and intermediate data at the same time and so will need frequent off-chip memory accesses—which will stall the photonic core. Similarly, non-GEMM operations should be performed fast enough not to throttle down the high-throughput photonic core. Therefore, all the electronic components in the accelerator should be architected carefully and analyzed in detail to make fair conclusions about the photonic technology. Unfortunately, the studies of the non-photonic components in the previous works have been limited. In our work, we provide a complete system-level analysis in terms of power and latency including the non-photonic arithmetic units for non-linear operations, data conversion circuits, die-to-die interconnect, and on-chip and off-chip memory.

2.4.2 Benchmarks. Prior photonic accelerators are either specifically designed for CNNs or report results only for CNNs. While these accelerators perform well for convolution operations, most are under-utilized and perform poorly for linear layers. Moreover, several of them use old and small neural networks that do not stress the memory system as much as state-of-the-art neural networks. Additionally, non-CNN networks are typically richer in terms of the variety of operations—which makes the system perspective even more important. Given that non-CNN networks are being more commonly used in recent years, focusing on only CNNs provides a limited perspective on using photonic cores for DNN acceleration. ADEPT is the first photonic accelerator work to report results for non-CNN networks, particularly with BERT-Large and RNN-T, which contain a wider range of operations than CNNs.

Broadly, while previous works are helpful toward understanding the raw capability of photonic compute cores, our key takeaway message here is that it is not just about the raw compute capacity of photonic cores; instead, it is important to look at the system as a whole and understand the general applicability and true benefits of photonic technology in artificial intelligence.

3 FULL-SYSTEM ARCHITECTURE

Our work focuses on understanding the implications of a complete electro-photonic system consisting of a host CPU, DRAM, PCI-e bus, and the electro-photonic accelerator ADEPT (see Figure 1(d)). ADEPT is connected to the host CPU and DRAM through a PCI-e bus. Host CPU handles the compilation and any other operations required by the DNN model that can be performed offline including pre/post-processing (e.g., resizing, decoding, etc.) and precomputation of the phase values for the MZIs. The inference is then performed fully in ADEPT without any interference from the host CPU. In this section, we introduce the ADEPT (micro)architecture, present optimizations that allow it to be efficiently integrated into a full system, and describe the compilation flow so that we can do a full evaluation of an electro-photonic system.

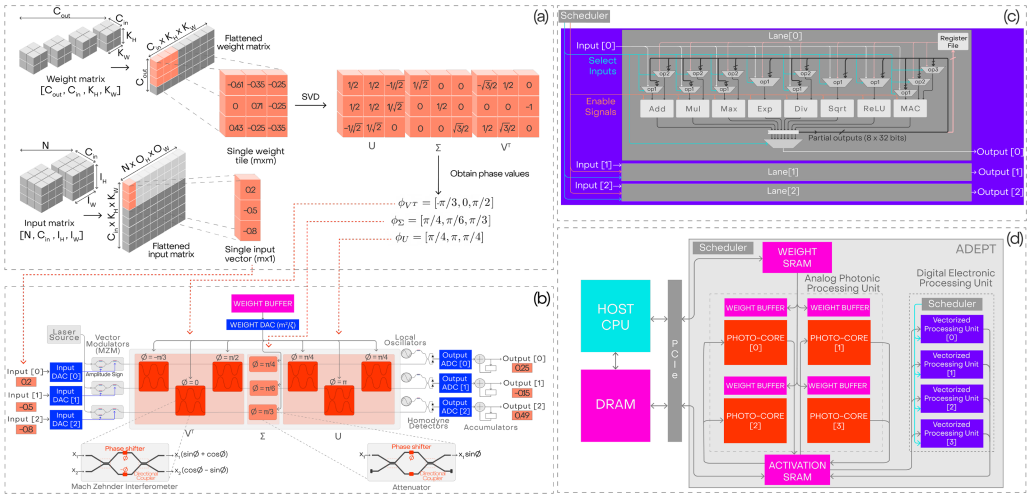

Fig. 1. Diagram showing different components of ADEPT and how operations are performed. (a) Example GEMM operation in the photo-core. (b) Programming input and weight matrices into the photo-core. The $m \times m$ (here $m = 3$ as an example) photo-core consists of $2 \times m(m-1)/2 = 6$ MZIs (for U and V^T) and three attenuators (for Σ). (c) Microarchitecture for a single digital electronic vectorized processing unit. The unit comprises $m = 3$ digital lanes, each consisting of arithmetic units to perform non-GEMM operations. (d) Full system architecture including the host CPU, the DRAM, and ADEPT—interconnected using a PCI-e interface. As an example, we show four photo-cores and four vectorized processing units.

3.1 ADEPT Architecture

ADEPT is an electro-photonic accelerator that contains an analog photonic computing unit for GEMM operations, a custom digital electronic vectorized processing unit for non-GEMM operations, and memory units for storing weight and activation data.

3.1.1 GEMM Operation with Photonics. ADEPT uses MZIs as a building block to perform MVM operations that can eventually be composed into a GEMM operation. The transfer function of an MZI is represented by a 2×2 orthogonal matrix:

$$U(2) = \begin{bmatrix} \sin\phi & \cos\phi \\ \cos\phi & -\sin\phi \end{bmatrix}, \tag{1}$$

where ϕ is the phase difference between the two internal arms of the MZI. Similarly, MZIs can be used as an attenuator for scaling a single value when one arm is blocked.

To perform an MVM using an MZI array, we first need to program the matrix into the MZIs as phase values. Figure 1(a) shows an example of programming a 3×3 matrix M into a 3×3 MZI array in Figure 1(b). The matrix M is first decomposed into the three matrices through the SVD, i.e., $M = U\Sigma V^T$, where U and V^T are $m \times m$ orthogonal matrices and Σ is a diagonal matrix of singular values. $U(m)$ and $V(m)^T$ unitary matrices with $m > 2$ can be composed using either a rectangular [29] or triangular [77] pattern. In this work, we use the rectangular pattern proposed by Clements et al. [29], which outperforms the triangular pattern due to its symmetry and reduced optical depth.

Next, the phases $(\phi_U, \phi_\Sigma, \phi_{V^T})$ needed to program in the matrices U, Σ, and V^T are computed by using the phase decomposition algorithm [29]. The phase decomposition algorithm is an algorithm similar to QR decomposition that breaks a large orthogonal matrix U into a series of 2×2

orthogonal matrices acting on different input rows. Finally, a total of m^2 phase values—equal to the number of elements in M—are programmed into the array to create the matrix M.

An MVM between a matrix M and a vector v_{in} can then be achieved by (1) programming the matrix M in the array of MZIs, (2) encoding the vector v_{in} in the amplitude and phase (0 or π for sign) of the optical signals entering the array, and (3) obtaining the resulting vector $v_{\text{out}} = M \cdot v_{\text{in}}$ at the output of the array. When the vector v_{in} is inserted at GHz rate, a 100×100 array enables us to perform linear operations at 10 **Tera Operations per Second (TOPS)**. A GEMM operation consists of a series of MVM operations. GEMM between two matrices can be achieved by encoding one matrix in the MZI array and by sending the other matrix through the array as optical signals—one vector at a time.

3.1.2 Analog Photonic Computing Unit. The photonic computing unit in ADEPT is an analog unit designed to perform MVM operations that can eventually be composed into a GEMM operation. The unit consists of a set of vector modulators (**Mach-Zehnder Modulators (MZMs)**), an array of MZIs, photo-detectors, **analog-to-digital converters (ADCs)**, and **digital-to-analog converters (DACs)** (see Figure 1(b)). We refer to the unit without ADCs and DACs as the photo-core.

GEMM operations in DNNs (e.g., in the fully connected layer and the two-dimensional (2D) convolution layer) typically involve a multiplication between a weight tensor and an input tensor. The input and weight matrix shapes vary for each layer in a DNN, but the photo-core has a fixed size of $m \times m$. Therefore, matrices bigger than $m \times m$ are divided into $m \times m$ sized submatrix tiles and loaded into the photo-core one by one.

We adopt a WS dataflow in the photo-core, where the weight matrix is programmed into the MZI array and the input vector is encoded in the optical signals. Figure 1(a) shows a simple example of this step. First, the input and weight matrices are flattened if necessary (for 2D convolutions) using "im2col" pre-processing [8], and the weight matrix is broken into submatrix tiles. Each weight tile is then decomposed into two orthogonal matrices (U and V^T) and a diagonal matrix of singular values (Σ) using SVD. Next, each of the three matrices is decomposed into its respective phase values ($\phi_U, \phi_\Sigma, \phi_{V^T}$) using the phase decomposition algorithm [29]. SVD, tiling and phase decomposition are performed *only once upfront* for each weight submatrix tile in the host CPU. Therefore, it does not introduce a latency overhead during the inference. This one-time cost is discussed further in Section 6. Importantly, the total number of phase values is equal to the number of elements in the weight tile. Therefore, the memory footprint required for storing the decomposed parameters is the same as that for storing the original tile. The phase values obtained from above can be directly programmed into the MZI array, as shown in Figure 1(b).

In our WS approach, the weight values of a tile are first transferred from the weight SRAM into the weight buffer. Data from the weight buffer can be programmed into the photo-core at a rate limited by the modulation mechanism of the MZIs—during which the photo-core is inoperable. This overhead is unavoidable but it is fairly small ~10 ns [103]. Once the tile is loaded into the photo-core, the values are maintained in the MZI array while all input vectors that need to be multiplied with this particular tile are fed into the photo-core vector-by-vector. WS dataflow is critical to minimize the number of times we update the values in the MZIs and to amortize the power latency cost of programming the MZI array.

Figure 1(b) also shows an example of how the input and output vectors are programmed and read out, respectively. Each element of the input vector is programmed using an MZM and a phase shifter that encode the amplitude and the sign (0 or π) of the input optical signals, respectively. The output vector ($m \times 1$) of the MVM operation (both amplitude and sign) is detected using m coherent detectors with the help of a local oscillator. The resulting photocurrent is eventually

converted into digital bits, using 8-bit ADCs. The input and the weight DACs are chosen to be 10-bit and 12-bit precise, respectively, which are adequate to guarantee 8-bit precise outputs (see Section 3.1.5). The obtained partial results are accumulated digitally to construct the final output. Due to our WS approach, the partial output vectors generated by a weight tile do not contribute to the same final output result. We write all the partial output vectors generated by the first weight tile of a weight matrix to the activation SRAM. The partial output vectors generated by the next weight tile of the weight matrix are added to the partial output vectors stored in the SRAM and the accumulation result is written back to the SRAM array. This process is repeated for all the weight tiles of the weight matrix.

This approach requires extra SRAM reads and writes for accumulation compared to an **output stationary (OS)** dataflow. However, OS is not efficient in our architecture, because it would require updating the values programmed into the MZIs each cycle. Such a large modulation bandwidth (i.e., in the orders of GHz) cannot be achieved with the NOEM-based MZIs used in our design (with only ∼100 MHz bandwidth). There are MZI designs using p-n or p-i-n junctions that can be modulated at such large bandwidths, but they introduce a higher insertion loss (≥1 dB per MZI) that limits the scalability of the array due to high power consumption.

As an alternative option, one can also consider an IS approach where the input matrix, instead of the weight matrix, is programmed into the MZI array. However, while similar to the WS approach, the IS approach is also not suitable for the architecture of our photo-core. This is because the input matrix of a DNN layer is the output of the previous layer and is computed at runtime. Therefore, SVD and phase decomposition algorithms, both of which have the same computational complexity of a GEMM operation, will also need to be performed on the input matrix (that is programmed into the array) at runtime instead of being performed offline.

3.1.3 Digital Electronic Processing Unit. Although more than 90% of the operations are GEMM operations, a non-trivial amount of non-GEMM operations must also be performed as part of DNN inference. These operations include element-wise non-linear operations (e.g., ReLU, GELU, and sigmoid), reduction operations (e.g., softmax and max-pool), batch and layer normalizations, and element-wise multiplication and addition (e.g, bias). As discussed in Section 2.3, these non-GEMM operations are more effectively performed in the digital domain instead of the analog domain.

To maintain the balance between the analog and digital parts of ADEPT, within the digital electronic ASIC, we use the same number of vectorized processing units as the photo-cores. The microarchitecture of a single vectorized processing unit is shown in Figure 1(c). In each vectorized processing unit, we use the same number of lanes as the number of optical lanes (channels) in one photo-core such that output of each optical lane in the photo-core is fed to one lane in the vectorized processing unit via the activation SRAM. Each lane has separate units for multiplication, addition, division, max, square root, and exponential operations (each 32-bit) that enable the system to complete the wide variety of non-GEMM operations. These arithmetic units are implemented as custom digital CMOS circuits. All lanes in the vectorized processing unit can operate in parallel and can be pipelined for non-GEMM operations that require multiple arithmetic operations. Each arithmetic unit uses a multiplexer to choose the input from (1) the activation SRAM, (2) the output of the arithmetic units, or (3) the register files of the vectorized unit, as operands. Here the register files (64 KB each) are used to store the constants (which are loaded up front) for the non-GEMM operations or the outputs of the arithmetic units. Multiplexers are controlled by a scheduler that decides when each arithmetic operation is used. The outputs of digital electronic ASIC are written back to the activation SRAM—to be used in the next layer of the DNN.

To extract the maximum performance from ADEPT, we need to match the throughput of the photo-core and the digital electronic ASIC. It is, however, challenging to design a digital ASIC that

can operate above 2 GHz. Hence, we use n logical units in parallel for each operation within the individual vector lane. Each unit operates at $1/n$ times the clock frequency f_c of the photo-core (each offset by $1/f_c$ to one another) to match the throughput of photo-core.

3.1.4 Data Movement and Storage. ADEPT utilizes two separate SRAM units: one for input/output activations and one for weights. The SRAM units can transfer data between each other through direct memory access and communicate with the host and DRAM through the PCI-e fabric. The two SRAM units are separated, because, generally, a dichotomy exists between the activations and the weights, and data transfer between them is not frequent. The activation SRAM is used to store both input and output activations, because, effectively, the output of one layer is the input of the next layer. At runtime, both the photo-core and the digital electronic ASIC read and write a vector of size m (the size of the photo-core) from and to the activation SRAM. We use separate dedicated read/write ports in the activation SRAM for the photo-core and the digital electronic ASIC.

Transferring a complete weight tile ($m \times m$) from weight SRAM to photo-core in one step requires a large SRAM bandwidth. In contrast, transferring one vector at a time requires a large latency in between tiles. Hence, we use a weight buffer for each photo-core as an intermediate stage. We load the tile for the next set of GEMM operations into the weight buffer, while the photo-core is performing GEMM operations with the current weight values. The data from the weight buffer are then programmed into the photo-core in ∼10 ns [103], minimizing the latency in between consecutive tiles in the photo-core and increases the photo-core's overall utilization and the system throughput.

3.1.5 Numerical Precision and Accuracy. Maintaining numerical precision during DNN computation is one of the main challenges when computing with an analog core. A similar problem exists in photonic computing: The numerical precision of the output vector v_{out} is limited by how well one can encode the input vector v_{in} and the matrix M. The error of the weight matrix M can be calculated by considering that all classical photonic operations (in our case, these are the transformations afforded by U, Σ, and V^T) are unitary/orthogonal transformations. We are then primarily interested in obtaining the magnitude of the total perturbation of the weight matrix M, which can be defined as

$$\Delta M = \sqrt{\langle \|\Delta U\|^2 \rangle + \langle \|\Delta \Sigma\|^2 \rangle + \langle \|\Delta V\|^2 \rangle}. \tag{2}$$

The error of each matrix can be quantified by its normalized Frobenius norm $\|\Delta X\|^2 = \sum_{ij} |\Delta X_{ij}|^2/m$, such that $\Delta X = \sqrt{\|\Delta X\|^2}$, where X is a placeholder for U, Σ, and V^T.

Two realistic device limitations contribute to the perturbation of each transformation: phase encoding error and component error that can cause crosstalk [11, 86, 93, 115]. We consider the error of the phase encoding to be quantified as $\varepsilon_\phi \leq 2^{-b_w}$, where b_w is the bit precision of the weight DAC. The error induced by a single-phase setting error is $\|\Delta X\|^2 \approx \varepsilon_\phi^2/m$ [11]. For the component error, given that the matrix is composed of MZIs, which only use 50:50 DCs, we only need to consider the DC splitting error. The matrix perturbation induced by a single DC error ε_{DC} (typically due to fabrication imperfections) is $\|\Delta X\|^2 \approx 2\varepsilon_{\text{DC}}^2/m$ [11].

Realistically, fabrication imperfections are often correlated: neighboring DCs will be perturbed in a similar manner. However, correlation can add significant complexity, obfuscating mathematical insight. Monte Carlo simulation tools should be used to quantify the errors when correlations exist. For the remainder of the calculation, we assume an uncorrelated error model for both the encoding and the component errors.

The error terms can be added in quadrature under the uncorrelated error model.[1] In each MZI, there are two DCs and a single-phase shifter. In the Σ matrix, there are only m parallel MZIs, thus $\langle \|\Delta\Sigma\|^2 \rangle = \varepsilon_\phi^2 + 4\varepsilon_{\text{DC}}^2$. In the matrices U and V^T, there are $m(m-1)/2$ MZIs. The depth of the photonic circuits in M grow as $O(m)$, and component errors cascade as light propagates down the mesh. A naïve programming of the phases will result ΔM that grows as $O(m^{1/2}\varepsilon)$ [29, 77]. However, a more sophisticated error-corrected programming strategies [11, 40, 41] can achieve a better scaling with respect to the component errors, such that the error grows as $O(m^{1/2}\varepsilon_\phi + m\varepsilon_{\text{DC}}^2)$, which is advantageous when $\varepsilon_{\text{DC}} \leq m^{-1/2}$.

More sophisticated error-corrected programming strategies [11, 40, 41], however, can achieve a better scaling with respect to the component errors, such that

$$\langle \|\Delta U\|^2 \rangle = \langle \|\Delta V^T\|^2 \rangle = \frac{m(m-1)}{2}\left[\frac{\varepsilon_\phi^2}{m} + 2\frac{2\varepsilon_{\text{DC}}^4}{3m}(m+1)\right], \tag{3}$$

where the first term in the square bracket is the contribution due to phase encoding error and the second is due to component error. The corrected programming strategy effectively allows for a squaring of the component errors that is advantageous when $\varepsilon_{\text{DC}} \leq m^{-1/2}$. Finally, the error for the overall matrix M can therefore be defined as in Equation (2).

The precision of the input and output vectors can now be quantified by adding the errors in quadrature: $\Delta v_{\text{out}}^2 = \Delta v_{\text{in}}^2 + \Delta M^2$. The error of the input vector encoding $\Delta v_{\text{in}} \leq 2^{-b_{\text{in}}}$ is quantified by the bit precision of the input DACs b_{in}. The output vector will be captured by ADCs with a bit precision of b_{out}. For the output vector error to be dominated by the ADC precision, we have Δv_{out} must be $\leq 2^{-b_{\text{out}}}$. Assuming a reasonable DC splitting error of < 0.1 % (this quantity should be measured in the fabricated silicon photonic wafer), we can determine that the precision of the output vector can be maintained up to ~8 bits for matrices up to size 256×256 if the input and the weight DAC bit precisions are 10 and 12 bits, respectively. We use these bit precisions in designing our hybrid electro-photonic accelerator.

The discussion above provides important intuition as to how errors compound in an analog photonic computer. The analysis effectively relates the **signal-to-noise ratio (SNR)** of the DACs with the SNR of the output ADC. Note that the input and the weight data to be encoded in the photonic GEMM device can have fewer than b_{in} and b_w bits, respectively—they just have to be encoded using DACs with SNRs commensurate to the prescribed bit precision.

3.2 Optimizations

In this section, we explain the optimizations that help us efficiently orchestrate the operations in ADEPT, reduce the latency overhead caused by the non-GEMM operations and data transfers, and maximize the system performance.

3.2.1 Pipelining Operations. We pipeline GEMM and non-GEMM operations in ADEPT. Specifically, once *an output vector* (after accumulating the partial output results) of a GEMM operation has been generated, that output vector is immediately sent to the digital electronic ASIC for non-linear operations. Therefore, non-GEMM operations begin without the need to wait for the whole GEMM operation to be completed.

In addition, more than one layers including non-GEMM operations can follow one another, or one layer may need to use more than a single logical unit. We further optimize ADEPT by

[1]Realistically, fabrication imperfections are often correlated: neighboring DCs will be perturbed in a similar manner. However, correlation can add significant mathematical complexity. Monte Carlo simulation tools should be used to quantify the errors when correlations exist.

pipelining these non-GEMM operations in the digital electronic ASIC. For example, the softmax layer uses the exponential unit, the max unit, and the multiplication unit. While one element is using the exponential unit, the previous output of the exponential unit uses the max unit. As a result, as long as the data dependency is preserved, different non-GEMM operations or different steps using different arithmetic units in the digital electronic ASIC within a non-GEMM operation can be parallelized and pipelined.

3.2.2 Maximizing the Batch Size and Optimized Buffering. ADEPT's throughput is limited by the rate at which data are input into the photo-core. While the latency and bandwidth of activation and weight SRAM arrays can be designed to match the throughput of the photo-core, the sizes of these arrays are limited. If the activations and weights do not fit within these SRAM arrays, then frequent DRAM accesses would be necessary. These DRAM accesses are slower compared to SRAM accesses and can easily bottleneck the system performance. To avoid being bottlenecked by DRAM latency during runtime, we may want to limit the batch size for a given neural network. However, larger batch sizes provide a better throughput. We, therefore, propose an optimized buffering method that maximizes the batch size stored in the activation SRAM without ever spilling back to the DRAM during runtime. This method takes advantage of the empty space in the SRAM during inference and loads the inputs of the next batch from DRAM efficiently.

We describe this optimized buffering method as a convex optimization problem. Let $\vec{x}_c = [x_c(t_0),\ x_c(t_1), \ldots,\ x_c(t_{max})]$ be a vector representing the activation SRAM array usage while performing inference on a batch of activations over time. Here $t_{i+1} = t_i + \Delta t$ where Δt is some time interval chosen to ensure the optimization problem is tractable for the host CPU. Similarly, $\vec{x}_{pcie} = [x_{pcie}(t_0),\ x_{pcie}(t_1), \ldots,\ x_{pcie}(t_{max})]$ is a vector representing the activation SRAM usage of the data (next input batch) being transferred from DRAM into SRAM over time. For a given $\vec{x}_c$, an optimal $\vec{x}_{pcie}$ data transfer schedule can be obtained by solving the following optimization problem:

$$\begin{aligned} &\text{Maximize:} && \sum_{t=t_0}^{t_{max}} x_{pcie}(t) \\ &\text{Subject to:} && 0 \le x_c(t) + x_{pcie}(t) \le x_{max}; \\ & && x_{pcie} \ge 0;\ x_{pcie}(t_{-1}) = 0;\ x_{pcie}(t_{max}) = x_{input}; \\ & && 0 \le \Delta x_{pcie}(t) \le \text{Max. PCI-e bandwidth} \end{aligned}$$

The constraints in the optimization problem can be understood as follows: the total SRAM usage (1) should be less than the given SRAM size (x_{max}), (2) should not be negative at any time, and (3) should start from zero; (4) the total amount of data transferred will be equal to the input size of the next batch; and (5) the data transfer rate should be slower than the maximum PCI-e bandwidth. The objective function is to maximize the area under the curve of memory usage of the transferred data for the next batch. Maximizing this area guarantees transferring the data *as soon as possible* under the constraint of a maximum PCI-e bandwidth. If the program fails to return a schedule $x_{pcie}(t)$ that meets the specified constraints for a given batch size and maximum PCI-e bandwidth, then a smaller batch size or a larger bandwidth (if it is available on the hardware) should be chosen. We use the above optimization program to find the *largest* batch size that ensures that the memory usage from storing activations of the current batch and the next batch never exceeds the SRAM size. As such, we ensure that all DRAM data transfer for the next batch of inputs can happen simultaneously with the inference of the current batch. The optimized schedule is computed only once by the host CPU before runtime.

3.2.3 Parallelism. ADEPT can be scaled up to include multiple photo-cores. We offer two parallelization strategies for distributing the workload among multiple photo-cores: data parallelism

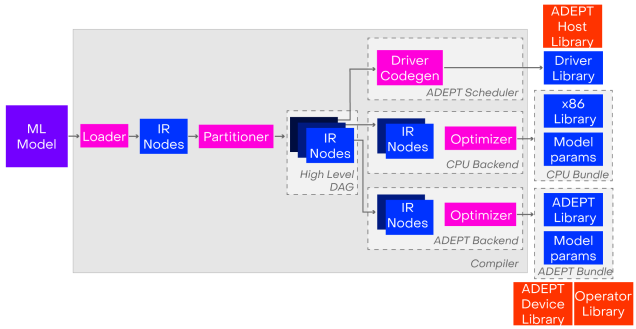

Fig. 2. Execution model. Compilation process of an ML model for ADEPT.

and tile parallelism. Data parallelism aims to accelerate MVMs by copying the same weights to all photo-cores. Each photo-core performs the same operations on different inputs in a batch. Tile parallelism is a finer granularity model parallelism that distributes different tiles of a weight matrix across multiple photo-cores. Unlike data parallelism, all inputs in one batch are sent to all photo-cores.

ADEPT can also use WDM-based parallelism. WDM uses multiple wavelengths for encoding different input vectors at once similar to data parallelism. The scheme requires multiplexing and demultiplexing circuits that can be constructed from microring resonators [16] or cascaded unbalanced MZIs [110]. WDM parallelism is synonymous to data parallelism in terms of throughput, but the same MZI array and weight DACs can be used by all inputs encoded in the wavelengths.

3.3 Execution Model

In this section, we describe the execution model for using ADEPT as part of the full-system. This process is summarized in Figure 2. Here, we take a DNN model and compile it on the host CPU to generate a program in the form of a graph on tensor types. We use ONNX models (exported from the common frameworks, such as Pytorch) and a loader to build a high-level program graph whose nodes are operations on higher-dimensional array datatypes. We create a directed acyclic graph by using a cost-model-based partitioner and annotate nodes based on whether the operations will be executed on a CPU or on the ADEPT device. We use an LLVM-based optimizer on the host CPU for code generation along with the optimizations. We then expand the operations annotated for execution on the ADEPT device into a stream of ADEPT instructions, and perform a scheduling pass to achieve overlap of GEMM operations and non-GEMM operations. We use the annotated program graph to optimize the schedule and pipeline compute on the host CPU and the ADEPT device with communication between the two. The generated code for these three partitions are linked with the corresponding libraries to produce two executable binaries: one for the host and one for ADEPT. It should be noted that the host CPU performs the compilation only once and then offloads the inference to ADEPT.

4 EVALUATION METHODOLOGY

In this section, we describe our evaluation approach when we compare ADEPT against SAs and state-of-the-art accelerators. We provide power, performance, and area analysis for both standalone GEMM cores, as well as for the full system. For our evaluation, we choose three DNNs: ResNet-50[43], BERT-large [32], and RNN-T [44]. These three state-of-the-art networks—all part of the MLPerf inference data-center benchmarks [78] in the *offline* scenario—represent the diversity

in layer types, sizes, and shapes that we observe in DNNs. We combine architecture, circuit, and device level analyses to evaluate the full system.

In regards to electronic designs, previous works explored GPUs, many-core architectures, SAs, and so on, for accelerating DNNs [14, 70]. Broadly, SAs achieve a higher efficiency among these different hardware solutions [70] by using dedicated MAC units and reducing data transfers. In fact, they have been used as the main building block in many ASICs for DNN acceleration [22, 25, 52, 53]. In addition, SAs have a similar dataflow to ADEPT—which enables us to make a fair head-to-head comparison for the same array sizes and investigate the benefit of using a photonic core over a purely electronic core. Most of the evaluation section is dedicated to the comparison between SAs and ADEPT. We also provide a comparison of ADEPT against state-of-the-art electronic and photonic accelerators in Section 5.5 for completeness.

4.1 Architecture-level Analysis

We used a mix of SCALE-Sim [83] and RTL simulations for our architecture-level analysis. SCALE-Sim is a simulator built for SA architectures. It takes the SA configuration (i.e., array size and dataflow type) and the neural network configuration (i.e., layer sizes and batch size) as inputs, and calculates the number of cycles needed to execute the neural network. The simulator also generates traces for SRAM and DRAM reads/writes. We modified SCALE-Sim to model the performance of the photo-core in ADEPT. The modifications were added on top of the existing WS dataflow in SCALE-Sim, which is similar with the WS approach of the photo-core. These modifications include adding the latency for programming the weight tile into the MZI array, adding the latency for transferring the weights from the weight SRAM to the weight buffer and overlapping this data transfer latency with continuing GEMM operations.

SCALE-Sim enables us to simulate our dataflow and directly compare the performance of the photo-core with that of SAs. However, it only models GEMM operations. To evaluate non-GEMM operations, we designed the digital electronic ASIC using SystemVerilog RTL. We also incorporated the optimizations described in Section 3.2 in our evaluation. For each DNN, we combined the timing results obtained from SCALE-Sim and RTL simulations to get the overall performance.

4.2 Circuit/Device-level Analysis

For a realistic power, performance, and area comparison, we designed the digital electronic ASIC units and SAs at RTL level and then synthesized them using Cadence Genus [2] with a standard cell library designed in the GF22FDX technology node [3]. The SRAM arrays were generated using an SRAM compiler for GF22FDX.

To minimize impact of slow DRAM transfers on performance, prior works have used large on-chip memory arrays [54, 56]. We follow the same strategy. However, it is challenging to have a single large SRAM array with low access latency. So, instead, we use multiple small sized SRAM sub-arrays to build larger memory arrays. The SRAM sub-arrays were designed to have 64 KB capacity with ∼1 ns access latency. For higher clock frequencies ($f_c > 1$ GHz), we read from multiple arrays, each offset by $1/f_c$ ns with its neighbor. In total, we use 300 MB weight SRAM and 100 MB activation SRAM. We acknowledge that not all our buses connecting the SRAM arrays to the photo-core will have the same latency. For a 700 mm^2 (reported in Section 5.4) chip size, the latency is calculated as ∼1.2 ns (maximum 12 cycles for the 10 GHz system) for the longest distance to travel (from one corner to the diagonally opposite corner) [51]. The throughput of the SRAM accesses is matched with the system clock by operating each SRAM sub-array at 833 MHz, but reading from the different SRAM sub-arrays every 100 ps.

The photo-core is powered by a laser. We calculated the required laser power per channel P analytically by considering (1) the laser wall-plug efficiency, (2) the losses of the various optical

devices, and (3) the SNR needed for an 8-bit output, as follows:

$$P = \frac{(\kappa\ \mathrm{SNR}_{\mathrm{shot}})^2 \cdot (q\ \Delta f/4)}{\eta_{\mathrm{det}} \cdot \eta_{\mathrm{array}} \cdot \eta_{\mathrm{mod}} \cdot \eta_{\mathrm{cpl}} \cdot \eta_{\mathrm{laser}}}, \tag{4}$$

where $\mathrm{SNR}_{\mathrm{shot}}$ is the SNR assuming shot noise only and κ (assumed to be ≈3) accounts for noise contributions (e.g., thermal noise and transistor noise) other than the shot noise. The overall SNR = $\kappa\ \mathrm{SNR}_{\mathrm{shot}} = 2^{b_{\mathrm{out}}}$ with b_{out} being the bit precision of the output ADC. Here q is the elementary charge, and Δf is the bandwidth of the coherent detector (related to the clock frequency). The η's account for the transmissivity from the laser to the detectors. η_{mod} is the transmissivity of the modulator (≈1.2 dB loss [5]), η_{array} is the transmissivity of the MZI array (≈0.04 dB loss per MZI [72] and each signal passes through $2m$+1 MZIs), η_{cpl} is the fiber laser-to-chip coupling efficiency (≈2 dB loss), η_{det} is the efficiency of the photodetectors (≈80% [60]), and η_{laser} is the wall-plug efficiency of the laser (≈20% [67]). All the photonic devices in the photo-core are simulated using Lumerical Maxwell-Equations solver FDTD and circuit-level simulator INTERCONNECT [1]. They have also been fabricated in the GF90WG SiPh process and are characterized at multiple-wafer-scale with the FormFactor CM300 wafer tester.

The necessary bit precisions for the inputs and the weights are 10 and 12 bits, respectively, to guarantee the 8-bit-precise outputs read by the ADCs (see Section 3.1.5). Due to the lack of publicly available DAC prototypes in GF22FDX with our desired precision, for our analysis, we used a 14-bit DAC [46] designed with 28-nm CMOS technology with a 10 GS/s sampling rate and 177-mW power consumption. Note that the power consumption of 10-bit and 12-bit DACs will be less than a 14-bit DAC. Therefore, we scaled the power numbers as follows: A widely accepted **figure of merit (FoM)** for the performance of DACs is $\mathrm{FoM} = 2^B \cdot f_s|_{6(B-1)}\ /P_{\mathrm{DAC}}$. Here B is the bit precision of the DAC, $f_s|_{6(B-1)}$ is the output signal frequency where the spurious free dynamic range has dropped with 6 dB (= 1 bit) in comparison with the expected results (≈ $6B$), and P_{DAC} is the power consumption of the whole DAC [58]. In essence, the power consumption of a DAC—with the same FoM—is proportional to 2^B. Therefore, a 12-bit DAC (for the weights) with the same FoM will consume $2^2 = 4$ times less power than a 14-bit DAC. Similarly, a 10-bit DAC (for the inputs) with the same FoM will consume $2^4 = 16$ times less power than a 14-bit DAC. The 10-bit input and 12-bit weight DACs will then consume 11.06 and 44.25 mW, respectively. Similarly to DACs, we use 10-bit ADCs in 28-nm technology at the output. Within the 10 ns settling time constraint of the MZIs, a single 10 GS/s DAC can be used to program 100 weights into MZIs. Therefore, instead of using m^2 DACs, we use $\lceil m^2/\zeta \rceil$ DACs for weights where ζ is equal to 100. Each ADC has a 5 GS/s sampling rate and consumes 29 mW [39]. The 10-bit inputs require a high ER in the input modulators. This can be achieved by using active optimization approaches [65, 107]. We use two additional MZIs per modulator as variable beam splitters for obtaining a perfect 50:50 splitting in both ends of the modulator. In addition, an equalized phase-dependent loss between the two middle arms can be achieved if the MZM is driven in a differential push-pull manner. The electronic-to-optical (E-O) and O-E conversion power is based on the total energy required to operate the modulator circuitry, which is ~20 fJ/bit, and the detector circuitry, which is ~297 fJ/bit [95]. Each DRAM access is assumed to be 20 pJ/bit [45]. The die-to-die interconnect between the photonic and electronic chiplets consumes 0.3 pJ/bit [31]

5 EVALUATION RESULTS

Our evaluation focuses on answering two questions: (1) How do we build a complete electro-photonic accelerator architecture that is not bottlenecked by the slower electronics? and (2) How much are the electro-photonic accelerator systems better than purely electronic accelerator

systems, when we consider the system as a whole, i.e., accelerator + memory + host processor + communication, running practical real-world applications?

In Section 5.1, to set the stage, we first provide a detailed comparison of stand-alone photo-core against electronic SAs. This comparison helps us determine the ADEPT design that we should use for exploring different architecture optimizations as well as for performing full-system analysis. In Section 5.2, we evaluate the impact of optimizations we introduced in Section 3.2, and in Section 5.3, we analyze the different parallelism methodologies. While these first three sections answer the first question, Section 5.4 answers the second question by comparing the complete ADEPT-based system where all the components and optimizations are taken into account against a similar system where photo-core is replaced with a same-sized SA. Last, in Section 5.5, for completeness, we provide a comparison of ADEPT against state-of-the-art electronic and photonic accelerators.

5.1 Photo-core vs. SAs

The photo-core utilizes light, which oscillates at hundreds of terahertz, and so it has a significant bandwidth advantage over the electronic SAs. The bandwidth in the photo-core is typically limited by the sampling rate of data converters (considered up to 10 GHz in this work), while SAs are constrained due to parasitic resistance, capacitance, and inductance. In fact, in the case of SAs, Cadence Genus with GF22FDX failed to meet the timing requirements for 2 GHz and above. Therefore, we used parallelism instead to effectively operate the SA at higher frequencies. For example, to operate a SA at 10 GHz, we used ten 1-GHz SAs whose clock cycles are offset by 100 ps. The latency of this parallelized SA will still be 1 ns, but its throughput will be synonymous to a single SA operating at 10 GHz. For this analysis, we assume both photo-core and SA are isolated from the system, weights and inputs have been loaded and are available in the SRAM with reads/writes fast enough to keep up with the requirements of both arrays. To provide SAs a strong baseline, we considered OS, WS, and IS for SAs as dataflow can have a significant impact on SA's performance. Figure 3(a) shows the throughput we can achieve when using the three dataflows for SAs for three different benchmarks. We observe that OS performs better than WS and IS for SAs. This is because of the high latency of loading data into the SA between tiles for WS and IS dataflows. Therefore, from here onwards, we use OS for SA in the rest of the comparison.

5.1.1 Throughput. For the throughput comparison, we use a single 128×128 array (the choice of size is justified later in this section) for both the photo-core and the SA. Figure 3(a) shows the comparison between the performance of the photo-core (when using WS dataflow) and that of the SA operating at 1 GHz clock frequency for different batch sizes for three different networks—ResNet-50, BERT-large, and RNN-T (one plot per network). In photo-core, by pipelining the weight transfer from weight SRAM into the weight buffer with GEMM operations, we reduce the latency of loading the weights down to 10 ns, the minimum required by MZIs (see Section 3.1.2 and 3.1.4). In general, photo-core's WS dataflow is more advantageous compared to the OS SA when the weight matrices are large and the input matrices are small (e.g., RNN-T with small batch sizes), because each weight tile needs to be loaded only once.

Throughput vs. Batch Size. From Figure 3(a) we can see that as the batch size increases, throughput (and correspondingly utilization) of the arrays increases and eventually saturates. Among the three DNNs, we observe that the throughput saturates for ResNet-50 and BERT-large more quickly than RNN-T. This is because the small input matrices in RNN-T means that fewer number of vectors are multiplied with the same tile. Thus, the utilization and throughput continue to significantly increase until we have larger batch sizes for RNN-T. In addition, as the batch size increases, latency in between tiles becomes less important, because more time is spent on performing MVM operations in each tile.

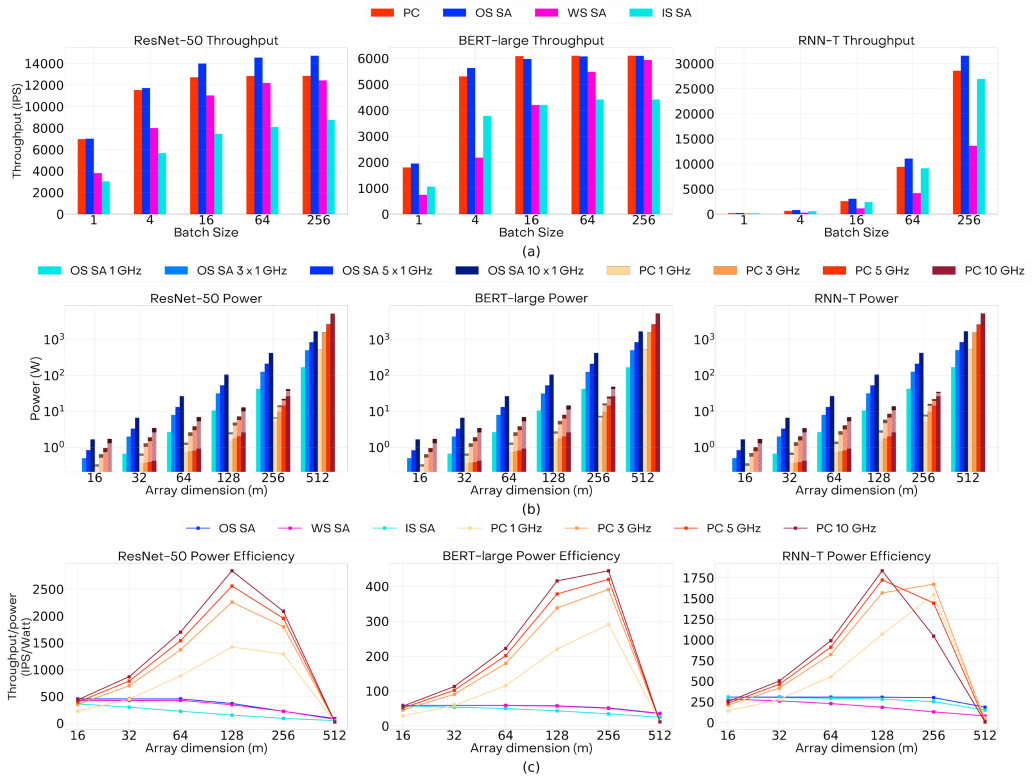

Fig. 3. Photo-core vs SA comparison in terms of throughput (IPS), power (W), and power efficiency (IPS/W). (a) Throughput vs. batch size of 128×128 photo-core (PC) and SAs with OS, WS, and IS dataflows at 1 GHz clock. (b) Power consumption of the OS SA (1, 3×1, 5×1, and 10×1 GHz clock) and the WS photo-core (1, 3, 5, and 10 GHz clock), for different array sizes. For the photo-core we report power in laser (solid color), ADCs/DACs (white diagonal pattern) and E-O/O-E conversion (black diagonal pattern). (c) Power efficiency (in IPS/W) of OS, WS, IS SAs (1, 3×1, 5×1, and 10×1 GHz clock) and WS photo-core (1, 3, 5, and 10 GHz clock) for different array sizes. Here $f_c \times 1$ GHz SA indicates that we use f_c SAs in parallel, each operating at 1 GHz.

Throughput vs. Operating Frequency. One way to increase the throughput of any computing device is to increase the clock frequency. We therefore attempt to increase the clock frequency of the photo-core and the SAs (from 1 to 3, 5, and 10 GHz). The throughput of the SAs increases linearly with the clock frequency. The rate of MVM operations in the photo-core also increases linearly with the clock frequency. However, a fixed 10-ns period is necessary for programming the MZI array and is independent from the clock frequency. Therefore, the increase in the throughput of the photo-core is sub-linear.

5.1.2 Power Consumption. Figure 3(b) compares the average power consumed by the WS photo-core (laser, ADC/DAC, and E-O/O-E conversion) and the OS SA of different sizes. For this analysis, we use a batch size of 256 to ensure that the throughput is nearly saturated for all networks. Overall, the photo-core's power consumption is smaller than the SA counterpart up to an array size of 256×256.

For the SAs, the power consumption increases linearly with the number of **processing elements (PEs)** (quadratically with the array size). For the photo-core, the laser power increases

exponentially with the depth m of the array. This is because the linear increase in the number of optical devices on an optical channel causes an exponential impact on optical power loss (in watts) in the propagating signal. As a result, it can be seen in Figure 3(b) that laser power increases drastically as the tile size gets larger and dominates the power consumption for large array sizes. For an $m \times m$ photo-core, we need m DACs and m E-O conversion circuits for the input vector, and m ADCs and m O-E conversion circuits for the output vector. These input/output DACs/ADCs perform a conversion each cycle. Additionally, we need DACs for programming the $m \times m$ weight matrix. These DACs for programming the weights into the MZIs are not used each cycle. The weights are programmed into the MZI once for each tile, and the DACs are not used until all MVMs for the corresponding tile are finished. The average power consumption of DACs/ADCs increases as the array size increases, because the latency drops. Effectively, the same number of conversions are performed within a shorter duration of time.

5.1.3 Power Efficiency. Figure 3(c) shows the power efficiency (IPS/W) of electronic SAs and photo-cores for different array sizes and frequencies. We observe that, for the photo-core, 128×128 is the most power-efficient array size for all three networks and all four clock frequencies. This can be explained by the fact that beyond a certain size, the laser power starts dominating the power consumption of the photo-core. Additionally, beyond a certain array size, the utilization decreases and so the throughput saturates. Therefore, due to the exponentially increasing laser power and saturating throughput, we observe a drop in the power efficiency beyond an array size of 128×128. For SAs, the power increases quadratically with the array dimension m. However, because the throughput increases less than quadratically with m, the power efficiency decreases as the array size increases.

Across different frequencies and array dimensions, we observe that photo-core can provide up to *9.87*×, *9.32*×, and *7.69*× better power efficiency than OS SA for ResNet-50, BERT-large, and RNN-T, respectively, when only GEMM operations are considered. *Overall, we observe that for the same clock frequency, while the throughput is comparable, photo-core provides a better power efficiency than the best performing SA.* As we show that 128×128 is the most power-efficient array size for the photo-core, we will use this array size for the further evaluations.

5.2 Optimizations

As discussed in Section 3.2, non-GEMM operations and data transfers introduce latency and energy overhead and are important in system evaluation. In this section, we quantify these overheads and show the impact of the optimizations we apply on performance of ADEPT for different types of DNNs.

5.2.1 Pipelining. Figure 4 shows the impact of pipelining operations on the inference time of ADEPT when running ResNet-50, BERT-large, and RNN-T. For ResNet-50, the max-pool, average-pool, ReLU activations and softmax layers; for BERT-large, the layer norm, GELU, and softmax operations; and for RNN-T, the element-wise addition and multiplication, sigmoid, and tanh operations (within an LSTM layer) are computed in the digital electronic ASIC. The non-GEMM operations comprise a small percentage of the networks' operations, but they can lead to a large overhead if not pipelined carefully. When pipelined, the non-GEMM operations and the GEMM operations can be performed in parallel.

In Figure 4, we can see that ResNet-50 has the least amount of overhead due to non-GEMM operations. With batch normalizations folded, ReLU becomes the most frequent non-GEMM operation, which can be effectively overlapped with the GEMM operations. In BERT-large and RNN-T, the division and exponential operations in GELU, softmax, sigmoid, and tanh increase the

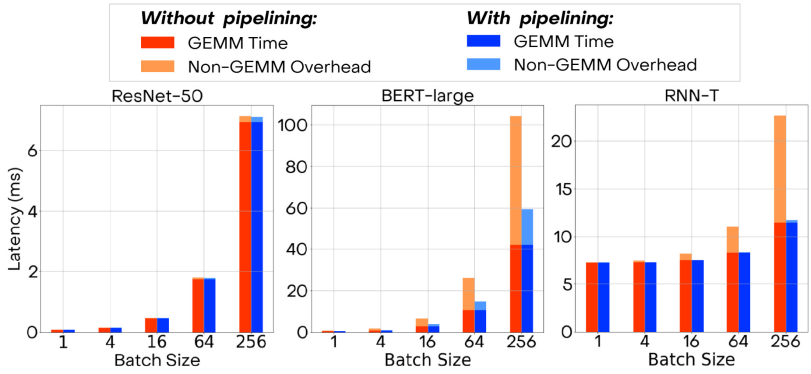

Fig. 4. Pipelining evaluation. Latency of ADEPT with a 128×128 photo-core operating at 10-GHz clock with and without pipelining the GEMM and non-GEMM operations. Here the latency is for one batch of inputs for three networks. The results are calculated for varying batch sizes.

number of cycles spent in the digital electronic ASIC. As batch size increases, GEMM operations are performed more efficiently, because more input vectors are multiplied with the same tile—weights are re-used more frequently. However, the cycles spent on non-GEMM operations increase linearly with batch size. Effectively, we observe a larger increase in the time spent on the non-GEMM operations than the increase in time spent on the GEMM operations with increasing batch size. As a result, a smaller portion of the non-GEMM operations can be overlapped with the GEMM operations. We observe a reduction in latency of up to 5.73% in ResNet-50, 43.03% in BERT-large, and 48.22% in RNN-T when we pipeline the non-GEMM and GEMM operations.

5.2.2 Optimized Buffering. Until now, we used a large batch size of 256 to evaluate the saturated throughput of both ADEPT and SAs. However, given that the SRAM arrays have limited sizes, an inference with batch size of 256 may not fit within the activation SRAM.

For this analysis, we choose a 100 MB activation SRAM and a 300 MB weight SRAM to ensure that the weights of all the three networks can comfortably fit within ADEPT. Figure 5 shows the usage of the activation SRAM array for the current batch and the next batch when using our optimized DRAM access mechanism (see Section 3.2.2). We limit the batch size to the maximum value where inference on the entire batch can be completed without any DRAM transfers (58, 88, and 50 for ResNet-50, BERT-large, and RNN-T, respectively). The activation SRAM stores the inputs and outputs of all GEMM and non-GEMM operations over time. If the GEMM and non-GEMM operations are running at the same time (pipelined), then the memory usage includes both of the operations' activation data. Figure 5 shows that the networks do not use the whole SRAM array throughout the inference. This creates an opportunity to transfer the inputs for the next batch.

We compare the performance of our optimized buffering technique against double buffering [84]: a common method for minimizing the impact of data transfer latency. In double buffering, one half of the memory is used for the current inference while the other half is used for transferring the inputs for the next inference. As a result, the maximum batch sizes of this scheme, for the three networks, are half of those of the optimized buffering scheme. For ResNet-50 and BERT-large, optimized buffering technique increases the throughput only by 1.3% and 0.4% compared to double buffering. This is because these two networks have already high utilization in the photo-core and their throughputs are saturated for the considered batch sizes. Remarkably, however, optimized data transfer increases the throughput of RNN-T by 89.7% over double buffering.

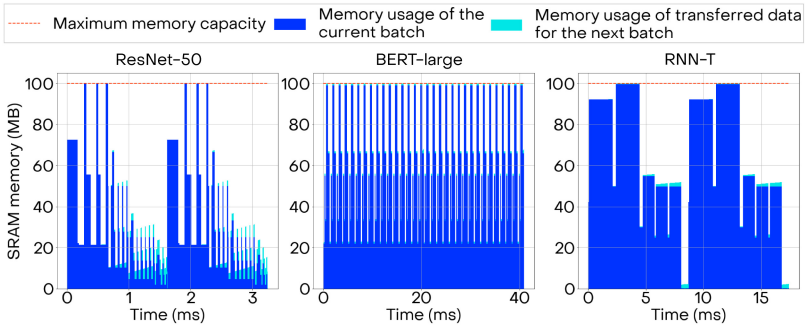

Fig. 5. Activation SRAM usage for computing on the current batch of inputs along with data transfer for the next batch of inputs within ADEPT. Both input and output activations for the current batch must be stored in the activation SRAM (dark blue) while the input data are transferred for the next batch (light blue). A 128 × 128 photo-core at 10-GHz clock is used with batch sizes of 58, 88, and 50 for ResNet-50, BERT-large, and RNN-T, respectively, to fully use the 100-MB activation SRAM capacity.

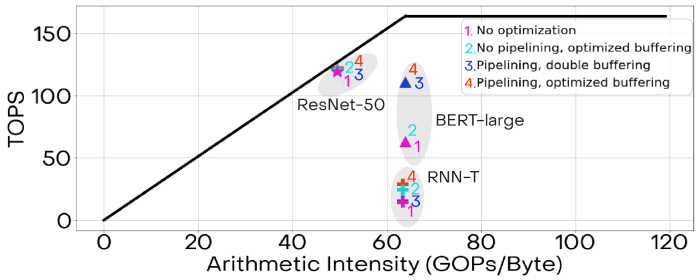

Fig. 6. Impact of optimizations. Roofline plot showing the effect of optimizations on ADEPT with a single 128 × 128 photo-core. The arithmetic intensity is calculated using MAC operations over activation SRAM reads/writes.

Impact of Optimizations. Figure 6 summarizes the impact of the two optimizations—pipelining and optimized DRAM buffering, on ADEPT at a system level. The roofline is the peak throughput of the photo-core, and the memory ceiling is derived from the bandwidth of the activation SRAM. The baseline (no optimization) refers to the case without any pipelining and with double buffering.

Comparing the three networks, ResNet-50 has a smaller arithmetic intensity (AI) and is memory-bound. We see that the performance of ResNet-50 without the optimizations is close to the roofline; thus, further optimizations only marginally improve the performance. BERT-large significantly benefits from pipelining with a 1.76× better throughput because of the frequent non-GEMM operations. In contrast, using the optimized DRAM buffering, which enables us to use larger batch sizes compared to double buffering, does not help because of the already saturated utilization of the photo-core for small batch sizes. RNN-T has a lower utilization compared to the other two networks. The utilization is mainly limited by the recurrent nature of the network, which requires frequent change of weight tiles and the frequent non-GEMM operations in the LSTM layers. Therefore, increasing batch size by using the optimized DRAM buffering increases the performance significantly—by 1.92× and pipelining improves the throughput for RNN-T by 1.83×.

The analysis presented in this section highlights the importance of taking non-GEMM operations and memory limitations into account and using different types of DNNs for evaluation. *The non-GEMM operations and memory limitations limit the throughput of photo-core, but it is possible*

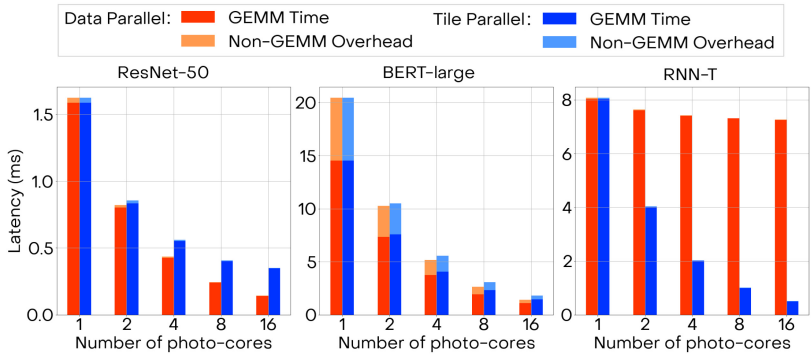

Fig. 7. Parallelism evaluation. Latency of ADEPT (128 × 128 photo-core at 10 GHz clock) when executing the three neural networks with different photo-core counts using data parallelism and tile parallelism.

to go around these limitations and improve the performance by using the right optimizations such as pipelining and efficiently buffering the data.

5.3 Parallelism

We consider three types of parallelism: data parallelism, tile parallelism, and WDM parallelism (see Section 3.2.3). Figure 7 shows how the latency scales with increasing number of photo-core counts for both data and tile parallelism. We use the batch sizes previously considered (see Section 5.2.2), i.e., 58, 88, and 50 for ResNet-50, BERT-large, and RNN-T, respectively. We keep these values constant as we increase the number of photo-cores.

Data parallelism provides an almost linear decrease in inference latency with increasing photo-core count when the number of input vectors within a batch is large enough to be shared among the photo-cores. The latency is dominated by MVM operations for large inputs sizes, and so as the number of photo-cores increases, the throughput proportionally increases. We observe this in ResNet-50 and BERT-large where the input matrices are large enough to be spread among the photo-cores and we can maintain high utilization. In contrast, when the number of input vectors per core decreases, the reduction in latency saturates due to the decrease in the utilization of the photo-cores. We observe this in RNN-T. Data parallelism provides 11.30×, 14.47×, and 1.11× lower latency for ResNet-50, BERT-large, and RNN-T when we increase the photo-core count from 1 to 16.

The advantage of tile parallelism is limited by the number of tiles in a GEMM layer. The networks with larger weight matrices (i.e., BERT-large and RNN-T) better exploit this parallelism. Tile parallelism provides 11.24×, 16.0×, and 4.62× lower latency for BERT-large, RNN-T, and ResNet-50, respectively, when the photo-core count increases from 1 to 16.

Multiple photo-cores means a linear increase in the area and the power consumption for the analog photonic computing unit. WDM provides an opportunity to reduce this area increase. WDM allows the input vectors to be mapped across the different wavelengths that are routed to same photo-core. Therefore, WDM offers the same throughput as data parallelism without using multiple copies of the MZI array and weight DACs. When we compare data parallelism with n photo-cores against a single photo-core leveraging n wavelengths in WDM, the photo-core with WDM uses $(n-1)\ m^2$ fewer MZIs and $(n-1)\ m^2/\zeta$ fewer weight DACs.

This WDM approach is, however, limited by the amount of optical power that can be injected into a single waveguide. High optical power causes strong nonlinear absorption in the waveguide and the peak optical power is limited to ~30 mW for ensuring signal integrity [19]. For a 128 × 128

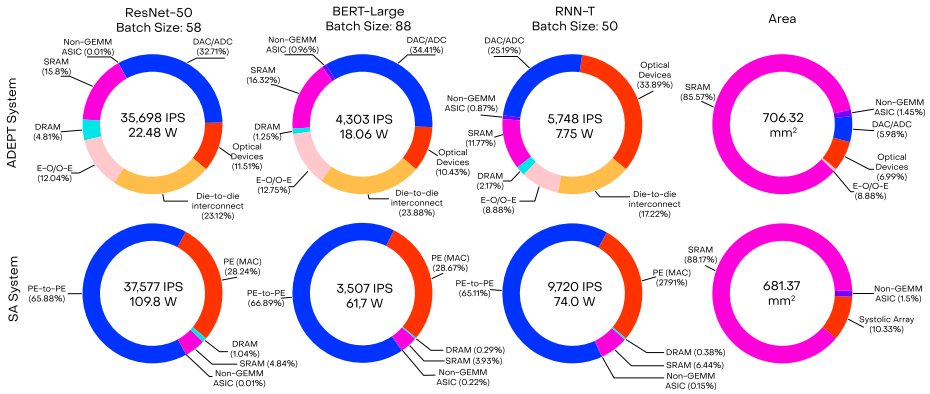

Fig. 8. Average total (static and dynamic) power distribution and area distribution of ADEPT (128 × 128, 10 GHz photo-core) and the SA system (128 × 128, 10 × 1 GHz array, OS dataflow).

array operating at 10 GHz with a single wavelength, the peak laser power per waveguide is 20.7 mW, which is under the 30-mW limit. If we want to leverage WDM, then we need to scale down the photo-core size so that the total injected optical power stays under the nonlinearity limit. For example, for multiplexing 2 wavelengths in an optical waveguide, we need to reduce the photo-core size to 64 × 64. This configuration achieves 1.4× better IPS/W/mm^2 compared to two 64 × 64 photo-cores with a single wavelength and 1.23× better IPS/W/mm^2 compared to a single 128 × 128 photo-core with a single wavelength, on average among the three evaluated networks, ResNet-50, BERT-large, and RNN-T, when all cores are operating at a 10-GHz clock frequency.

5.4 System-level Comparison

In this section, to answer the main question of how much the real benefit in a complete system is, we include all the components of the system and the optimizations discussed in Section 3.2, and provide a full system-level comparison between 128 × 128 WS ADEPT and a 128 × 128 OS SA (see Figure 8).

From Figure 8, we can see that the optical devices in the photo-core (i.e., laser, MZIs, modulators) used for the GEMM computation take up only between 10 and 35% of the overall power consumption in the ADEPT system depending on the DNN model. The other components of the system (i.e., ADCs/DACs, O-E/E-O conversions, die-to-die communication, and SRAM) consume significant power—which proves the necessity of the system-level evaluation. For the SA, data transfer between the register files of the PEs dominate the power consumption of the SA system. We observe that SRAM dominates the area distribution for both electronic SAs and ADEPT for the chosen configuration.

In ADEPT, the photo-core and the digital electronic ASIC are in different chiplets to take advantage of the technology nodes that provide the best performance for each individual electronic and photonic ICs. The two chiplets are 3D integrated through an interposer. In the latter, the SA and the rest of the electronic components share the same chiplet. The optimizations used for ADEPT are also applied to the SA system.

Our analysis shows that a system with ADEPT consumes 4.88× (109.8 W vs. 22.48 W), 3.42× (61.7 W vs. 18.06 W) and 9.55× (74.0 W vs. 7.75 W) less power for ResNet-50, BERT-large, and RNN-T, respectively. This translates to *4.89*×, *3.24*×, and *9.06*× better power efficiency (IPS/W). Also, ADEPT provides *4.5*×, *2.97*×, and *8.34*× better power-area efficiency (IPS/W·mm^2) compared to a SA. *This shows us that although including the system components in evaluation decreases the*

Table 2. Comparison against State-of-the-Art Electronic and Photonic Accelerators

	ADEPT (This work)		Eyeriss [22]	Eyeriss v2 [25]	UNPU [57]	TPU v3 [52]
Tech Node	90-nm photonics + 22-nm CMOS		65 nm	65 nm	65 nm	16 nm
Clock rate	10 GHz		200 MHz	200 MHz	200 MHz	940 MHz
Benchmark	AlexNet	ResNet-50	AlexNet	AlexNet	AlexNet	ResNet-50
Batch size	192	58	4	1	15	N/A
IPS	217, 201	35,698	35	102	346	32,716
IPS/W	7,476.78	1,587.99	124.80	174.80	1,097.50	18.18
IPS/W/mm^2	10.59	2.25	10.18	N/A	68.59	0.01

Table 3. Comparison against State-of-the-art Photonic Accelerators

	ADEPT (This work)				Albireo-C [91]	DNNARA [71]	HolyLight-A [61]
Clock rate	10 GHz				5 GHz	1.2 GHz	1.28 GHz
Benchmark	AlexNet		ResNet-50		AlexNet	ResNet-50	AlexNet
Batch size	1	192	1	58	1	1	N/A
IPS	6,478	217, 201	12,641	35,698	7,692	9,345	50,000
IPS/W	872.17	7,476.78	1,021.17	1,587.99	344.17	100	900
IPS/W/mm^2	1.23	10.59	1.59	2.25	2.75	0.45	40.07

performance of the stand-alone photo-core, we can still benefit from using photo-cores instead of SAs in a system.

5.5 Comparison against DNN Accelerators

For completeness, in this section, we compare the full ADEPT system against state-of-the-art electronic [22, 25, 52, 57] and photonic [61, 71, 91] accelerators.

5.5.1 Electronic Accelerators. Besides the traditional SAs, more flexible electronic accelerator architectures have been proposed and shown to perform more efficiently. Table 2 compares ADEPT against state-of-the-art electronic accelerators. Much of the prior work focuses on AlexNet, and so we added ADEPT's results for AlexNet. Broadly, for AlexNet and ResNet-50 inference, while is not the most area efficient, ADEPT provides at least 6.8× higher IPS/W than other electronic accelerators.

5.5.2 Photonic Accelerators. In Section 2, using Table 1 we discussed that previous works on photonic accelerators have not provided a full system evaluation. For completeness, in Table 3 we provide a quantitative comparison against the state-of-the-art photonic accelerators. The numbers reported in Table 3 are highly dependent on the various design choices, i.e., careful consideration of optical device choices, ADC/DAC choices, the on-chip memory sizes, non-linear units, communication links, and so on, and the comprehensiveness of the evaluation. Previous works use very small on-chip memory arrays (in KBs). These small on-chip memory arrays have a small area and power consumption but require frequent DRAM transfers. Not all previous works have considered this DRAM transfer overhead. When designing ADEPT, we considered the sizes of the weights and the activations of the neural networks. The high throughput goal of the system necessitates an adequately large SRAM array that enables the DNN inference to run without being bottlenecked by the off-chip data transfers. We can see that the full system of ADEPT is not the most area efficient (due to large SRAM arrays), but it can provide 2.5× better IPS/W than Albireo-C and 10.2× better IPS/W than DNNARA for the same batch size of 1. Although the batch size is not reported in HolyLight, ADEPT's and HolyLight's power efficiencies are comparable when ADEPT uses a batch size of 1. However, ADEPT's activation SRAM array is adequate to store even larger batch sizes, which increases the utilization of the photo-core—providing a better overall system performance. When

the maximum batch size is used for ADEPT, it can provide more than 8.3× better power efficiency compared to the other three photonic accelerators.

It should be noted that the goal of this article is not to claim a more performant photonic core. In contrast, we aim to highlight the importance of a system-level analysis when evaluating photonic accelerators and encourage the community to adopt a pragmatic approach. The reasons that we achieve better results compared to other photonic accelerators despite their lack of system-level analysis can be listed as (1) we use low loss MEMS-based MZIs [72] (0.04 dB) enabling a power efficient large (128×128) MZI mesh in ADEPT, (2) we use large SRAM arrays enabling large batch sizes and better utilization of the photo-core, and (3) the choice of data converters (ADCs/DACs) is different in different designs.

6 DISCUSSION

We sought to develop a balanced architecture that benefits from accelerating GEMM using photonics (1) without being bottlenecked by digital electronic operations or storage overhead and (2) more than compensates for the overheads of electrical-optical and analog-digital conversions. To this end, it is essential to carefully formulate performance metrics to clearly see the system-level benefit of using photonics. In particular, we use IPS as the throughput performance metric instead of TOPS. The TOPS metric fails to consider processing unit utilization that is not likely to be unity.

In our proposed architecture, we perform electrical-optical and analog-digital conversions for input and output vectors each cycle. Although the overhead of performing conversions can improve with the process technology developments, it will remain a fundamental limitation for the speed and efficiency of the system. Hence, it may be worth performing more operations in the optical domain. However, this increases losses in optical devices that lowers SNR and lowers bit precision (see Equation (4)). Similarly, the limited bandwidth of MZI leads to a 10-ns weight programming latency that limits system performance. Using a WS approach, we minimize the impact of 10-ns MZI latency and power consumption of weight DACs.

In the photo-core, the dynamic range of values is limited due to the output ADCs. During MVM operations, the product of 8-bit inputs and weights results in more than 16 bits of information. ADCs reduce the precision of the MVM outputs back to 8-bit. Although we discussed how to preserve 8-bit accuracy at the output vectors in Section 3.1.5, the information loss in the partial outputs causes accuracy degradation in DNNs. However, extra training efforts can ameliorate the accuracy and keep it in the desired range. We confirmed that 8-bit precision is sufficient to maintain the accuracy of the benchmarks we used (e.g., ResNet-50, BERT-large, and RNN-T) within 1% of the FP32 accuracy after performing several epochs of quantization-aware retraining [55, 109]. QAT is performed in FP32. We model the inference accuracy by tiling GEMM operations and quantizing each input and output vector and weight tile for each MVM operation. While 8-bit precision is adequate for inference, training in the photo-core requires higher precision, which would lead to a higher power (roughly scales with 2^B where B is the number of bits). More intelligent training schemes may be needed to overcome this problem [37, 113].

Our study shows that SRAM dominates the area of both ADEPT and electronic SAs. It is beneficial to have large local SRAMs to accommodate large batches of inputs. However, SRAM size is limited in a chip-scale system. Therefore, scaling out to multiple chips is required to increase the SRAM cache size. The problem of designing a scaled-out system with multiple chips with multiple photo-cores, mapping a DNN model onto the many accelerators, and orchestrating the communication between them is part of our future work.

Our analyses show that different types of DNNs exhibit different photo-core utilization due to the differing shapes of weight and input matrices. Vatsavai et al. [104] show that irregular matrix

shapes, specifically in layers such as depthwise and pointwise convolution, might result in low utilization in photonic tensor cores and propose a reconfigurable architecture. For our rectangular MZI mesh architecture, such reconfigurability is not possible as multiplication and addition operations are not performed in separate blocks. However, in our architecture, it is possible to turn off unused optical channels (similarly to that in optical networks [20]) to save power when utilization is low. We leave the exploration of this runtime management as part of future work.

To run DNN inference using the MZI-based photo-core, first, we need to perform SVD and phase decomposition on the original weight values of the DNN. For a given DNN, the weight values are fixed for inference. Therefore, the cost of performing SVD and phase decomposition is a one-time-only cost and will be amortized over all inferences. For completeness, we determined this one-time cost as follows. We performed SVD and unitary matrix decomposition on a 128×128 matrix with randomly chosen values between $[-1,1]$. We used the SVD function from the NumPy Linear Algebra submodule to decompose the matrix. Then we implemented the phase decomposition method for a square mesh as described in Clements et al. [29]. For 10 iterations of the experiment that we ran on a 2 sixteen-core 2.8-GHz Intel Gold 6242 [4], the average energy consumption is 1728.9 joules (DRAM energy included, calculated using PyRAPL [73]) and the average time is 11.5 seconds. For ResNet50, the total number of 128×128 tiles is 1756. This would consume approximately 2724.7 kJ of energy and take approximately 5 hours. This process can be further accelerated using GPU implementations.

7 CONCLUSION

In this article, we proposed and evaluated an end-to-end hybrid system for accelerating DNN inference containing a new electro-photonic accelerator called ADEPT. We showed that accelerating DNN inference with photonics requires tight interplay between the photonic compute units for GEMM operations and the electronic logic units for non-GEMM operations. The result is a balanced electro-photonic system architecture that has a throughput that is similar to the throughput of a system utilizing the widely accepted SA architecture while consuming significantly less power. With the introduced optimization methods for pipelining operations and data transfers, we showed that we can leverage the high throughput of the photonics GEMM accelerator without being bottlenecked by electronic units. Overall, we are optimistic that photonic computing is nigh, and we are looking forward to the application of the technology in real-life. Given its advantage over purely electronic systems in terms of IPS/W or IPS/W·mm^2, we are confident that the technology will find its rightful place within the Cambrian explosion of artificial intelligence accelerators.

ACKNOWLEDGMENTS

We thank Saumil Bandyopadhyay (MIT), Ryan Hamerly (MIT), Alexander Sludds (MIT), Leila Delshadtehrani (BU), and Zahra Azad (BU) for the valuable discussions and their insightful suggestions.

REFERENCES

[1] (nd). Ansys. Retrieved from https://www.ansys.com/products/photonics

[2] (nd). Genus Synthesis Solution. Retrieved from https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html

[3] (nd). GF22nm FD-SOI Technology. Retrieved from https://globalfoundries.com/sites/default/files/product-briefs/pb-22fdx-26-web.pdf

[4] (nd). Intel Xeon Gold 6242 Processor (22m Cache, 2.80 GHz) Product Specifications. Retrieved from https://ark.intel.com/content/www/us/en/ark/products/192440/intel-xeon-gold-6242-processor-22m-cache-2-80-ghz.html

[5] Suguru Akiyama, Takeshi Baba, Masahiko Imai, Takeshi Akagawa, Masashi Takahashi, Naoki Hirayama, Hiroyuki Takahashi, Yoshiji Noguchi, Hideaki Okayama, Tsuyoshi Horikawa, and Tatsuya Usuki. 2012. 12.5-Gb/s operation

with 0.29-V·cm VπL using silicon Mach-Zehnder modulator based-on forward-biased pin diode. *Opt. Expr.* 20, 3 (Jan. 2012), 2911–2923. https://doi.org/10.1364/OE.20.002911

[6] M. A. Al-Qadasi, L. Chrostowski, B. J. Shastri, and S. Shekhar. 2022. Scaling up silicon photonic-based accelerators: Challenges and opportunities. *APL Photon.* 7, 2 (Feb. 2022), 020902. https://doi.org/10.1063/5.0070992

[7] Dario Amodei. 2020. AI and Compute. Retrieved from https://openai.com/blog/ai-and-compute/

[8] Andrew Anderson, Aravind Vasudevan, Cormac Keane, and David Gregg. 2017. Low-memory GEMM-based convolution algorithms for deep neural networks. *CoRR* abs/1709.03395 (2017). http://arxiv.org/abs/1709.03395

[9] Tom Baehr-Jones, Ran Ding, Yang Liu, Ali Ayazi, Thierry Pinguet, Nicholas C. Harris, Matt Streshinsky, Poshen Lee, Yi Zhang, Andy Eu-Jin Lim, Tsung-Yang Liow, Selin Hwee-Gee Teo, Guo-Qiang Lo, and Michael Hochberg. 2012. Ultralow drive voltage silicon traveling-wave modulator. *Opt. Expr.* 20, 11 (May 2012), 12014–12020. https://doi.org/10.1364/OE.20.012014

[10] Reza Baghdadi, Michael Gould, Shashank Gupta, Mykhailo Tymchenko, Darius Bunandar, Carl Ramey, and Nicholas C. Harris. 2021. Dual slot-mode noem phase shifter. *Opt. Expr.* 29, 12 (Jun. 2021), 19113–19119. https://doi.org/10.1364/OE.423949

[11] Saumil Bandyopadhyay, Ryan Hamerly, and Dirk Englund. 2021. Hardware error correction for programmable photonics. *Optica* 8, 10 (Oct. 2021), 1247–1255. https://doi.org/10.1364/OPTICA.424052

[12] Viraj Bangari, Bicky A. Marquez, Heidi Miller, Alexander N. Tait, Mitchell A. Nahmias, Thomas Ferreira de Lima, Hsuan-Tung Peng, Paul R. Prucnal, and Bhavin J. Shastri. 2020. Digital Electronics and Analog Photonics for Convolutional Neural Networks (DEAP-CNNs). *IEEE J. Select. Top. Quant. Electr.* 26, 1 (2020), 1–13. https://doi.org/10.1109/JSTQE.2019.2945540

[13] Qiaoliang Bao, Han Zhang, Zhenhua Ni, Yu Wang, Lakshminarayana Polavarapu, Zexiang Shen, Qing-Hua Xu, Dingyuan Tang, and Kian Ping Loh. 2011. Monolayer graphene as a saturable absorber in a mode-locked laser. *Nano Res.* 4, 3 (2011), 297–307. https://doi.org/10.1007/s12274-010-0082-9

[14] Sathwika Bavikadi, Abhijitt Dhavlle, Amlan Ganguly, Anand Haridass, Hagar Hendy, Cory Merkel, Vijay Janapa Reddi, Purab Ranjan Sutradhar, Arun Joseph, and Sai Manoj Pudukotai Dinakarrao. 2022. A survey on machine learning accelerators and evolutionary hardware platforms. *IEEE Des. Test* 39, 3 (2022), 91–116. https://doi.org/10.1109/MDAT.2022.3161126

[15] T. E. Bell. 1986. Optical computing: A field in flux. *IEEE Spectrum* 23, 8 (1986), 34–38. https://doi.org/10.1109/MSPEC.1986.6371053

[16] 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. 2012. Silicon microring resonators. *Laser Photon. Rev.* 6, 1 (2012), 47–73. https://doi.org/10.1002/lpor.201100017

[17] H. John Caulfield. 1987. Parallel N4 weighted optical interconnections. *Appl. Opt.* 26, 19 (Oct. 1987), 4039–4040. https://doi.org/10.1364/AO.26.004039

[18] Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein. 2018. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Sci. Rep.* 8, 1 (2018), 12324.

[19] Chao Chen and Ajay Joshi. 2013. Runtime management of laser power in silicon-photonic multibus NOC architecture. *IEEE J. Select. Top. Quant. Electr.* 19, 2 (2013), 3700713–3700713. https://doi.org/10.1109/JSTQE.2012.2228170

[20] Chao Chen and Ajay Joshi. 2013. Runtime management of laser power in silicon-photonic multibus NOC architecture. *IEEE J. Select. Top. Quant. Electr.* 19, 2 (2013), 3700713–3700713. https://doi.org/10.1109/JSTQE.2012.2228170

[21] Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. 2014. DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. *SIGARCH Comput. Archit. News* 42, 1 (Feb. 2014), 269–284. https://doi.org/10.1145/2654822.2541967

[22] Y. Chen, T. Krishna, J. S. Emer, and V. Sze. 2017. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE J. Solid-State Circ.* 52, 1 (2017), 127–138. https://doi.org/10.1109/JSSC.2016.2616357

[23] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, and O. Temam. 2014. DaDianNao: A machine-learning supercomputer. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*. 609–622. https://doi.org/10.1109/MICRO.2014.58

[24] Yiran Chen, Yuan Xie, Linghao Song, Fan Chen, and Tianqi Tang. 2020. A survey of accelerator architectures for deep neural networks. *Engineering* 6, 3 (2020), 264–274. https://doi.org/10.1016/j.eng.2020.01.007

[25] Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. 2019. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE J. Emerg. Select. Top. Circ. Syst.* 9, 2 (2019), 292–308. https://doi.org/10.1109/JETCAS.2019.2910232

[26] Zhilu Chen, Jing Wang, Haibo He, and Xinming Huang. 2014. A fast deep learning system using GPU. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'14)*. 1552–1555. https://doi.org/10.1109/ISCAS.2014.6865444

[27] Qixiang Cheng, Jihye Kwon, Madeleine Glick, Meisam Bahadori, Luca P. Carloni, and Keren Bergman. 2020. Silicon photonics codesign for deep learning. *Proc. IEEE* 108, 8 (2020), 1261–1282. https://doi.org/10.1109/JPROC.2020.2968184

[28] Jack Choquette, Wishwesh Gandhi, Olivier Giroux, Nick Stam, and Ronny Krashinsky. 2021. NVIDIA A100 tensor core GPU: Performance and innovation. *IEEE Micro* 41, 2 (2021), 29–35. https://doi.org/10.1109/MM.2021.3061394

[29] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. 2016. Optimal design for universal multiport interferometers. *Optica* 3, 12 (Dec. 2016), 1460–1465. https://doi.org/10.1364/OPTICA.3.001460

[30] Edward Cottle, Florent Michel, Joseph Wilson, Nick New, and Iman Kundu. 2020. Optical convolutional neural networks–combining silicon photonics and fourier optics for computer vision. arXiv:2103.09044. Retrieved from https://arxiv.org/abs/2103.09044

[31] Behzad Dehlaghi and Anthony Chan Carusone. 2016. A 0.3 pJ/bit 20 Gb/s/Wire parallel interface for die-to-die communication. *IEEE J. Solid-State Circ.* 51, 11 (2016), 2690–2701. https://doi.org/10.1109/JSSC.2016.2596773

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805

[33] Po Dong, Wei Qian, Hong Liang, Roshanak Shafiiha, Ning-Ning Feng, Dazeng Feng, Xuezhe Zheng, Ashok V. Krishnamoorthy, and Mehdi Asghari. 2010. Low power and compact reconfigurable multiplexing devices based on silicon microring resonators. *Opt. Expr.* 18, 10 (May 2010), 9852–9858. https://doi.org/10.1364/OE.18.009852

[34] Clément Farabet, Yann LeCun, Koray Kavukcuoglu, Eugenio Culurciello, Berin Martini, Polina Akselrod, and Selcuk Talay. 2011. Large-scale fpga-based convolutional networks. In *Scaling Up Machine Learning: Parallel and Distributed Approaches*, 399–419.

[35] Clément Farabet, Berin Martini, Benoit Corda, Polina Akselrod, Eugenio Culurciello, and Yann LeCun. 2011. Neuflow: A runtime reconfigurable dataflow processor for vision. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR'11) Workshops*. 109–116. https://doi.org/10.1109/CVPRW.2011.5981829

[36] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja, J. Liu, C. D. Wright, A. Sebastian, T. J. Kippenberg, W. H. P. Pernice, and H. Bhaskaran. 2021. Parallel convolutional processing using an integrated photonic tensor core. *Nature* 589, 7840 (2021), 52–58. https://doi.org/10.1038/s41586-020-03070-1

[37] Sean Fox, Julian Faraone, David Boland, Kees Vissers, and Philip H. W. Leong. 2019. Training deep neural networks in low-precision with high accuracy using FPGAs. In *Proceedings of the International Conference on Field-Programmable Technology (ICFPT'19)*. 1–9. https://doi.org/10.1109/ICFPT47387.2019.00009

[38] K. Giewont, K. Nummy, F. A. Anderson, J. Ayala, T. Barwicz, Y. Bian, K. K. Dezfulian, D. M. Gill, T. Houghton, S. Hu, B. Peng, M. Rakowski, S. Rauch, J. C. Rosenberg, A. Sahin, I. Stobert, and A. Stricker. 2019. 300-mm monolithic silicon photonics foundry technology. *IEEE J. Select. Top. Quant. Electr.* 25, 5 (2019), 1–11. https://doi.org/10.1109/JSTQE.2019.2908790

[39] Mingqiang Guo, Jiaji Mao, Sai-Weng Sin, Hegong Wei, and Rui P. Martins. 2020. A 5 GS/s 29 mW interleaved sar adc with 48.5 dB SNDR using digital-mixing background timing-skew calibration for direct sampling applications. *IEEE Access* 8 (2020), 138944–138954. https://doi.org/10.1109/ACCESS.2020.3012699

[40] Ryan Hamerly, Saumil Bandyopadhyay, and Dirk Englund. 2022. Accurate self-configuration of rectangular multiport interferometers. *Phys. Rev. Appl.* 18, 2 (2022), 024019. https://link.aps.org/doi/10.1103/PhysRevApplied.18.024019

[41] Ryan Hamerly, Saumil Bandyopadhyay, and Dirk Englund. 2022. Stability of self-configuring large multiport interferometers. *Phys. Rev. Appl.* 18, 2 (2022), 024018. https://link.aps.org/doi/10.1103/PhysRevApplied.18.024018

[42] Nicholas C. Harris, Yangjin Ma, Jacob Mower, Tom Baehr-Jones, Dirk Englund, Michael Hochberg, and Christophe Galland. 2014. Efficient, compact and low loss thermo-optic phase shifter in silicon. *Opt. Expr.* 22, 9 (May 2014), 10487–10493. https://doi.org/10.1364/OE.22.010487

[43] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[44] Yanzhang He, Tara N. Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo yiin Chang, Kanishka Rao, and Alexander Gruenstein. 2018. Streaming end-to-end speech recognition for mobile devices. *CoRR* abs/1811.06621 (2018). http://arxiv.org/abs/1811.06621

[45] M. Horowitz. 2014. Computing's energy problem (and what we can do about it). In *Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC'14)*. 10–14.

[46] Hung-Yi Huang, Xin-Yu Chen, and Tai-Haur Kuo. 2021. A 10-GS/s NRZ/Mixing DAC With switching-glitch compensation achieving SFDR gt; 64/50 dBc over the first/second nyquist zone. *IEEE J. Solid-State Circ.* 56, 10 (2021), 3145–3156. https://doi.org/10.1109/JSSC.2021.3079111

[47] Huimin Li, Xitian Fan, Li Jiao, Wei Cao, Xuegong Zhou, and Lingli Wang. 2016. A high performance fpga-based accelerator for large-scale convolutional neural networks. In *Proceedings of the 26th International Conference on Field Programmable Logic and Applications (FPL'16)*. 1–9. https://doi.org/10.1109/FPL.2016.7577308

[48] Loc N. Huynh, Youngki Lee, and Rajesh Krishna Balan. 2017. DeepMon: Mobile GPU-based deep learning framework for continuous vision applications. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'17)*. Association for Computing Machinery, New York, NY, 82–95. https://doi.org/10.1145/3081333.3081360

[49] John David Jackson. 1975. *Classical Electrodynamics*. Wiley, New York, NY.

[50] Hasitha Jayatilleka, Hossam Shoman, Lukas Chrostowski, and Sudip Shekhar. 2019. Photoconductive heaters enable control of large-scale silicon photonic ring resonator circuits. *Optica* 6, 1 (Jan. 2019), 84–91. https://doi.org/10.1364/OPTICA.6.000084

[51] Ajay Joshi, Christopher Batten, Yong-Jin Kwon, Scott Beamer, Imran Shamim, Krste Asanovic, and Vladimir Stojanovic. 2009. Silicon-photonic clos networks for global on-chip communication. In *Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chip*. 124–133. https://doi.org/10.1109/NOCS.2009.5071460

[52] Norman P. Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson. 2020. A domain-specific supercomputer for training deep neural networks. *Commun. ACM* 63, 7 (Jun. 2020), 67–78. https://doi.org/10.1145/3360307

[53] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA'17)*. Association for Computing Machinery, New York, NY, 1–12. https://doi.org/10.1145/3079856.3080246

[54] Sangpyo Kim, Jongmin Kim, Michael Jaemin Kim, Wonkyung Jung, Minsoo Rhu, John Kim, and Jung Ho Ahn. 2021. BTS: An accelerator for bootstrappable fully homomorphic encryption. *CoRR* abs/2112.15479 (2021). https://arxiv.org/abs/2112.15479

[55] Raghuraman Krishnamoorthi. 2018. Quantizing deep convolutional networks for efficient inference: A whitepaper. *CoRR* abs/1806.08342 (2018). http://arxiv.org/abs/1806.08342

[56] Adam Lavely. 2022. *Powering Extreme-Scale HPC with Cerebras WaferScale Accelerators*. Technical Report. Cerebras Systems.

[57] Jinmook Lee, Changhyeon Kim, Sanghoon Kang, Dongjoo Shin, Sangyeob Kim, and Hoi-Jun Yoo. 2019. UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision. *IEEE J. Solid-State Circ.* 54, 1 (2019), 173–185. https://doi.org/10.1109/JSSC.2018.2865489

[58] Xing Li and Lei Zhou. 2020. A survey of high-speed high-resolution current steering DACs. *J. Semiconduct.* 41, 20060024 (Oct. 2020), 111404. https://doi.org/10.1088/1674-4926/41/11/111404

[59] Xing Lin, Yair Rivenson, Nezih T. Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan. 2018. All-optical machine learning using diffractive deep neural networks. *Science* 361, 6406 (2018), 1004–1008. https://doi.org/10.1126/science.aat8084

[60] Stefan Lischke, Dieter Knoll, Christian Mai, Lars Zimmermann, Anna Peczek, Marcel Kroh, Andreas Trusch, Edgar Krune, Karsten Voigt, and A. Mai. 2015. High bandwidth, high responsivity waveguide-coupled germanium p-i-n photodiode. *Opt. Expr.* 23, 21 (Oct. 2015), 27213–27220. https://doi.org/10.1364/OE.23.027213

[61] Weichen Liu, Wenyang Liu, Yichen Ye, Qian Lou, Yiyuan Xie, and Lei Jiang. 2019. HolyLight: A nanophotonic accelerator for deep learning in data centers. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'19)*. 1483–1488. https://doi.org/10.23919/DATE.2019.8715195

[62] Armin Mehrabian, Yousra Al-Kabani, Volker J. Sorger, and Tarek El-Ghazawi. 2018. PCNNA: A photonic convolutional neural network accelerator. In *Proceedings of the 31st IEEE International System-on-Chip Conference (SOCC'18)*. https://doi.org/10.1109/socc.2018.8618542

[63] Maziyar Milanizadeh, Douglas Aguiar, Andrea Melloni, and Francesco Morichetti. 2019. Canceling thermal cross-talk effects in photonic integrated circuits. *J. Lightw. Technol.* 37, 4 (2019), 1325–1332. https://doi.org/10.1109/JLT.2019.2892512

[64] David A. B. Miller. 2013. Self-configuring universal linear optical component. *Photon. Res.* 1, 1 (Jun. 2013), 1–15. https://doi.org/10.1364/PRJ.1.000001

[65] David A. B. Miller. 2015. Perfect optics with imperfect components. *Optica* 2, 8 (Aug. 2015), 747–750. https://doi.org/10.1364/OPTICA.2.000747

[66] Mario Miscuglio, Zibo Hu, Shurui Li, Jonathan K. George, Roberto Capanna, Hamed Dalir, Philippe M. Bardet, Puneet Gupta, and Volker J. Sorger. 2020. Massively parallel amplitude-only fourier neural network. *Optica* 7, 12 (2020), 1812–1819. https://opg.optica.org/optica/abstract.cfm?URI=optica-7-12-1812

[67] Gerard Mourou, Bill Brocklesby, Toshiki Tajima, and Jens Limpert. 2013. The future is fibre accelerators. *Nat. Photon.* 7, 4 (2013), 258–261. https://doi.org/10.1038/nphoton.2013.75

[68] Kishore Padmaraju and Keren Bergman. 2014. Resolving the thermal challenges for silicon microring resonator devices. *Nanophotonics* 3, 4-5 (2014), 269–281.

[69] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'15)*. 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964

[70] Biagio Peccerillo, Mirco Mannino, Andrea Mondelli, and Sandro Bartolini. 2022. A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives. *J. Syst. Arch.* 129 (2022), 102561. https://doi.org/10.1016/j.sysarc.2022.102561

[71] Jiaxin Peng, Yousra Alkabani, Shuai Sun, Volker J. Sorger, and Tarek El-Ghazawi. 2020. DNNARA: A deep neural network accelerator using residue arithmetic and integrated photonics. In *Proceedings of the 49th International Conference on Parallel Processing (ICPP'20)*. Association for Computing Machinery, New York, NY. https://doi.org/10.1145/3404397.3404467

[72] M. Poot and H. X. Tang. 2014. Broadband nanoelectromechanical phase shifting of light on a chip. *Appl. Phys. Lett.* 104, 6 (2014), 061101. https://doi.org/10.1063/1.4864257

[73] Powerapi-Ng. (nd). Powerapi-ng/pyrapl: A library to measure the python energy consumption of python code. Retrieved from https://github.com/powerapi-ng/pyRAPL

[74] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2383–2392. https://doi.org/10.18653/v1/D16-1264

[75] Carl Ramey. 2020. Silicon photonics for artificial intelligence acceleration : Hotchips 32. In *IEEE Hot Chips Symposium (HCS'20)*. IEEE, 1–26. https://doi.org/10.1109/HCS49909.2020.9220525

[76] Hannes Ramon, Michael Vanhoecke, Jochem Verbist, Wouter Soenen, Peter De Heyn, Yoojin Ban, Marianna Pantouvaki, Joris Van Campenhout, Peter Ossieur, Xin Yin, et al. 2018. Low-power 56Gb/s NRZ microring modulator driver in 28nm FDSOI CMOS. *IEEE Photon. Technol. Lett.* 30, 5 (2018), 467–470.

[77] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. 1994. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* 73 (Jul. 1994), 58–61. Issue 1. https://doi.org/10.1103/PhysRevLett.73.58

[78] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, R. Chukka, C. Coleman, S. Davis, P. Deng, G. Diamos, J. Duke, D. Fick, J. S. Gardner, I. Hubara, S. Idgunji, T. B. Jablin, J. Jiao, T. S. John, P. Kanwar, D. Lee, J. Liao, A. Lokhmotov, F. Massa, P. Meng, P. Micikevicius, C. Osborne, G. Pekhimenko, A. T. R. Rajan, D. Sequeira, A. Sirasao, F. Sun, H. Tang, M. Thomson, F. Wei, E. Wu, L. Xu, K. Yamada, B. Yu, G. Yuan, A. Zhong, P. Zhang, and Y. Zhou. 2020. MLPerf inference benchmark. In *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA'20)*. 446–459. https://doi.org/10.1109/ISCA45697.2020.00045

[79] Oskar A. Reimann and Walter F. Kosonocky. 1965. Progress in optical computer research. *IEEE Spectrum* 2, 3 (1965), 181–195. https://doi.org/10.1109/MSPEC.1965.5531775

[80] G. Roelkens, J. Van Campenhout, J. Brouckaert, D. Van Thourhout, R. Baets, P. Rojo Romeo, P. Regreny, A. Kazmierczak, C. Seassal, X. Letartre, G. Hollinger, J. M. Fedeli, L. Di Cioccio, and C. Lagahe-Blanchard. 2007. III-V/Si photonics by die-to-wafer bonding. *Mater. Today* 10, 7 (2007), 36–43. https://doi.org/10.1016/S1369-7021(07)70178-5

[81] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-NET: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'15)*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (Eds.). Springer International Publishing, Cham, 234–241.

[82] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[83] Ananda Samajdar, Yuhao Zhu, Paul N. Whatmough, Matthew Mattina, and Tushar Krishna. 2018. SCALE-sim: Systolic CNN accelerator. *CoRR* abs/1811.02883 (2018). http://arxiv.org/abs/1811.02883

[84] Jose Carlos Sancho and Darren J. Kerbyson. 2008. Analysis of double buffering on two different multicore architectures: Quad-core Opteron and the Cell-BE. In *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing*. 1–12. https://doi.org/10.1109/IPDPS.2008.4536316

[85] M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar, I. Durdanovic, E. Cosatto, and H. P. Graf. 2009. A massively parallel coprocessor for convolutional neural networks. In *Proceedings of the 20th IEEE International Conference on Application-specific Systems, Architectures and Processors*. 53–60. https://doi.org/10.1109/ASAP.2009.25

[86] Amin Shafiee, Sanmitra Banerjee, Krishnendu Chakrabarty, Sudeep Pasricha, and Mahdi Nikdast. 2022. LoCI: An analysis of the impact of optical loss and crosstalk noise in integrated silicon-photonic neural networks. In *Proceedings of the Great Lakes Symposium on VLSI*. 351–355.

[87] Bhavin J. Shastri, Alexander N. Tait, T. Ferreira de Lima, Wolfram H. P. Pernice, Harish Bhaskaran, C. D. Wright, and Paul R. Prucnal. 2021. Photonics for artificial intelligence and neuromorphic computing. *Nat. Photon.* 15, 2 (2021), 102–114. https://doi.org/10.1038/s41566-020-00754-y

[88] Yichen Shen, Nicholas C. Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, and Marin Soljačić. 2017. Deep learning with coherent nanophotonic circuits. *Nat. Photon.* 11, 7 (2017), 441–446. https://doi.org/10.1038/nphoton.2017.93

[89] Shen, Yun, Wang, Xiaodong, Zhang, Wei, Qiu, Ciyuan, and Cheng, Xiulan. 2017. Fabrication of depletion type microring modulator with high extinction ratio and high coupling quality factor. *MATEC Web Conf.* 139 (2017), 00066. https://doi.org/10.1051/matecconf/201713900066

[90] B. Shi, N. Calabretta, and R. Stabile. 2020. Deep neural network through an inp SOA-based photonic integrated cross-connect. *IEEE J. Select. Top. Quant. Electr.* 26, 1 (2020), 1–11. https://doi.org/10.1109/JSTQE.2019.2945548

[91] Kyle Shiflett, Avinash Karanth, Razvan Bunescu, and Ahmed Louri. 2021. Albireo: Energy-efficient acceleration of convolutional neural networks via silicon photonics. In *Proceedings of the ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA'21)*. 860–873. https://doi.org/10.1109/ISCA52012.2021.00072

[92] K. Shiflett, D. Wright, A. Karanth, and A. Louri. 2020. PIXEL: Photonic neural network accelerator. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA'20)*. 474–487. https://doi.org/10.1109/HPCA47549.2020.00046

[93] Farhad Shokraneh, Simon Geoffroy-Gagnon, and Odile Liboiron-Ladouceur. 2020. The diamond mesh, a phase-error-and loss-tolerant field-programmable MZI-based optical processor for optical neural networks. *Opt. Expr.* 28, 16 (2020), 23495–23508.

[94] Xiubao Sui, Qiuhao Wu, Jia Liu, Qian Chen, and Guohua Gu. 2020. A review of optical neural networks. *IEEE Access* 8 (2020), 70773–70783. https://doi.org/10.1109/ACCESS.2020.2987333

[95] Chen Sun, M. Wade, Yunsup Lee, J. Orcutt, L. Alloatti, M. Georgas, Andrew Waterman, J. Shainline, Rimas Avizienis, Sen Lin, B. Moss, R. Kumar, F. Pavanello, A. Atabaki, Henry Cook, Albert J. Ou, J. Leu, Yu hsin Chen, K. Asanović, Rajeev J. Ram, M. Popovic, and V. Stojanović. 2015. Single-chip microprocessor that communicates directly using light. *Nature* 528 (2015), 534–538.

[96] Jie Sun, Ranjeet Kumar, Meer Sakib, Jeffrey B Driscoll, Hasitha Jayatilleka, and Haisheng Rong. 2018. A 128 Gb/s PAM4 silicon microring modulator with integrated thermo-optic resonance tuning. *J. Lightw. Technol.* 37, 1 (2018), 110–115.

[97] Febin Sunny, Asif Mirza, Mahdi Nikdast, and Sudeep Pasricha. 2021. CrossLight: A Cross-layer optimized silicon photonic neural network accelerator. *CoRR* abs/2102.06960 (2021). https://arxiv.org/abs/2102.06960

[98] Alexander N. Tait, Thomas Ferreira De Lima, Ellen Zhou, Allie X. Wu, Mitchell A. Nahmias, Bhavin J. Shastri, and Paul R. Prucnal. 2017. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* 7, 1 (2017), 1–10.

[99] Alexander N. Tait, Mitchell A. Nahmias, Bhavin J. Shastri, and Paul R. Prucnal. 2014. Broadcast and weight: An integrated network for scalable photonic spike processing. *J. Lightw. Technol.* 32, 21 (2014), 3427–3439.

[100] Thomas N. Theis and H.-S. Philip Wong. 2017. The end of moore's law: A new beginning for information technology. *Comput. Sci. Eng.* 19, 2 (2017), 41–50. https://doi.org/10.1109/MCSE.2017.29

[101] Yvain Thonnart, Mounir Zid, José Luis Gonzalez-Jimenez, Guillaume Waltener, Robert Polster, Olivier Dubray, Florent Lepin, Stéphane Bernabé, Sylvie Menezo, Gabriel Parès, Olivier Castany, Laura Boutafa, Philippe Grosse, Benoît Charbonnier, and Charles Baudot. 2018. A 10Gb/s Si-photonic transceiver with 150 μW 120 μs-lock-time digitally supervised analog microring wavelength stabilization for 1Tb/s/mm2 die-to-die optical networks. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'18)*. 350–352. https://doi.org/10.1109/ISSCC.2018.8310328

[102] E. Timurdogan, C. V. Poulton, M. J. Byrd, and M. R. Watts. 2017. Electric field-induced second-order nonlinear optical effects in silicon waveguides. *Nat. Photon.* 11, 3 (2017), 200–206. https://doi.org/10.1038/nphoton.2017.14

[103] Xin Tu, Chaolong Song, Tianye Huang, Zhenmin Chen, and Hongyan Fu. 2019. State of the art and perspectives on silicon photonic switches. *Micromachines* 10, 1 (2019). https://doi.org/10.3390/mi10010051

[104] Sairam Sri Vatsavai and Ishan G. Thakkar. 2022. Photonic reconfigurable accelerators for efficient inference of cnns with mixed-sized tensors. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 41, 11 (2022), 4337–4348.

[105] Michael R. Watts, Jie Sun, Christopher DeRose, Douglas C. Trotter, Ralph W. Young, and Gregory N. Nielson. 2013. Adiabatic thermo-optic mach-zehnder switch. *Opt. Lett.* 38, 5 (Mar. 2013), 733–735. https://doi.org/10.1364/OL.38.000733

[106] Gordon Wetzstein, Aydogan Ozcan, Sylvain Gigan, Shanhui Fan, Dirk Englund, Marin Soljačić, Cornelia Denz, David A. B. Miller, and Demetri Psaltis. 2020. Inference in artificial intelligence with deep optics and photonics. *Nature* 588, 7836 (2020), 39–47. https://doi.org/10.1038/s41586-020-2973-6

[107] C. M. Wilkes, X. Qiang, J. Wang, R. Santagati, S. Paesani, X. Zhou, D. A. B. Miller, G. D. Marshall, M. G. Thompson, and J. L. O'Brien. 2016. 60dB high-extinction auto-configured mach–zehnder interferometer. *Opt. Lett.* 41, 22 (Nov. 2016), 5318–5321. https://doi.org/10.1364/OL.41.005318

[108] J. Wilson. (nd). The multiply and fourier transform unit: A micro-scale optical processor. https://optalysys.com/wp-content/uploads/2022/04/Multiply_and_Fourier_Transform_white_paper_12_12_20.pdf

[109] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. 2020. Integer quantization for deep learning inference: Principles and empirical evaluation. arXiv:2004.09602. Retrieved from https://arxiv.org/abs/2004.09602

[110] Hongnan Xu and Yaocheng Shi. 2018. Flat-top cWDM (De)Multiplexer based on MZI with bent directional couplers. *IEEE Photon. Technol. Lett.* 30, 2 (2018), 169–172. https://doi.org/10.1109/LPT.2017.2779489

[111] Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Andreas Boes, Thach G. Nguyen, Sai T. Chu, Brent E. Little, Damien G. Hicks, Roberto Morandotti, Arnan Mitchell, and David J. Moss. 2021. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* 589, 7840 (2021), 44–51. https://doi.org/10.1038/s41586-020-03063-0

[112] Tao Yan, Jiamin Wu, Tiankuang Zhou, Hao Xie, Feng Xu, Jingtao Fan, Lu Fang, Xing Lin, and Qionghai Dai. 2019. Fourier-space diffractive deep neural network. *Phys. Rev. Lett.* 123, 2 (2019), 023901.

[113] Guandao Yang, Tianyi Zhang, Polina Kirichenko, Junwen Bai, Andrew Gordon Wilson, and Christopher De Sa. 2019. SWALP: Stochastic weight averaging in low-precision training. *CoRR* abs/1904.11943 (2019). http://arxiv.org/abs/1904.11943

[114] Tiankuang Zhou, Xing Lin, Jiamin Wu, Yitong Chen, Hao Xie, Yipeng Li, Jingtao Fan, Huaqiang Wu, Lu Fang, and Qionghai Dai. 2021. Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nat. Photon.* 15, 5 (2021), 367–373.

[115] Ying Zhu, Grace Li Zhang, Bing Li, Xunzhao Yin, Cheng Zhuo, Huaxi Gu, Tsung-Yi Ho, and Ulf Schlichtmann. 2020. Countering variations and thermal effects for accurate optical neural networks. In *Proceedings of the 39th International Conference on Computer-Aided Design*. 1–7.

[116] Ying Zuo, Bohan Li, Yujun Zhao, Yue Jiang, You-Chiuan Chen, Peng Chen, Gyu-Boong Jo, Junwei Liu, and Shengwang Du. 2019. All-optical neural network with nonlinear activation functions. *Optica* 6, 9 (Sep. 2019), 1132–1137. https://doi.org/10.1364/OPTICA.6.001132

Received 5 December 2022; revised 25 March 2023; accepted 26 May 2023