Photonic Matrix Multipliers Light Up Artificial Brains

1 Introduction

Artificial intelligence has taken the world by storm. Despite being in infancy, machine learning techniques have already proven itself in industries everywhere from agriculture to healthcare. However, as we demand more from the technology, we also sacrifice more to it. AI is notoriously computationally expensive: generating a single image using the latest models consumes as much energy as it takes to fully charge a phone [1]. As industry leaders like OpenAI are pouring billions into installing and operating new hardware, environmentally conscious and cost sensitive researchers and developers are searching for alternative methods to run machine learning algorithms.

All current commercial processors are based on the CMOS process. Conceived in the 60's, the technology uses complementary sets of transistors to form logic gates, circuits, and eventually, processors [2]. However, after sixty years of development, the technology is approaching its limits. Continued miniaturization efforts require increasingly complex transistor geometries with plateauing benefits, causing researchers to look away from CMOS for the next generation's computing platform. These post-CMOS technologies include quantum information processing systems, processors based on exotic materials like topological insulators, and optical processors.

Optical processors are particularly suited as a machine learning accelerator [3]. Operating at the speed of light, the technology can meet the high throughput and low latency demands of artificial intelligence at a fraction of the power consumption of CMOS processors. Additionally, unlike most post-CMOS technologies, photonic processors have the potential to leverage existing CMOS manufacturing infrastructure, reducing costs and accelerating development. The matrix multiplication step in machine learning algorithms is particularly suited as a target for acceleration, being relatively stable, computationally expensive, and universally present in all AI systems.

We explore three implementations of photonic matrix multiplication proposed by Tang et al. [4], Ribeiro et al. [5], and Tait et al. [6]. Each of these implementations utilize different properties of matrix multiplication and nanophotonic systems to achieve a unique set of device characteristics. We compare them under three lenses—mathematical underpinnings, device choice, and fabrication—to determine the suitability of each towards accelerating AI workloads.

2 Background

Matrix multiplication is essential to artificial intelligence because artificial intelligence mimics human intelligence [7]. Specifically, artificial intelligence mimics the layout of the cells in our brains—neurons. The structure of a neuron can be broken into three parts: dendrites, the cell body, and the axon. Dendrites are extensions of the cell body that branch off to receive signals from other cells. These signals collect centrally in the cell body until a critical mass is met, triggering an action potential. During an action potential, a signal travels down the axon—another extension of the cell body—until meeting the dendrites of other neurons. At these junctions, called synapses, signals are transmitted between neurons, allowing them to communicate with each other. Finally, the neuron is reset, allowing it to collect signals again.

Although simple individually, the billions of neurons in our brains allow us to do complex cognitive tasks, including reasoning, remembering, and motor control. Specifically, the number of synapses and the strength of each synapse between each pair of neurons are thought encode these abilities. Artificial neural networks (ANN) aim to capture the intelligence of biological neural networks (BNN) while eliminating unnecessary complexity. Artificial neurons are modeled in layers, where each neuron receives the outputs of the previous layer. Specifically, the value of each neuron is a weighted sum of neurons in the previous layer, reflecting synaptic strengths and integration at the cell body, passed through an activation function, which models the 'all-or-nothing' effect of the action potential:

$$y_i = f\left(\sum_j w_{ij} x_j\right)$$

The weighted sum can be rewritten as a matrix multiplication step, where each element denotes the weight between each neuron in the previous layer and each neuron in the current layer. Since all neural networks are based on these operations, matrix multiplication is essential to artificial intelligence.

3 Comparing Device Design (Mathematical Foundation)

Despite its simplicity, matrix multiplication can be viewed from many perspectives and implemented many ways. Each of the three photonic matrix multiplication implementations utilize a different property of matrices to achieve the same desired result.

All implementations place constraints on the matrix. For example, total power out cannot be greater than total power in, so coefficients must be less than one. The first implementation, by Tang et al., also demands that the matrix must be unitary. This property, denoted by $UU^\dagger = I$, can be thought of as the complex extension of orthogonality and results in many properties useful in a photonic matrix multiplication implementation [8]. Notably, a unitary matrix T can be decomposed into the product of a series of phase shifting and diffractive matricies Φ_i and D:

$$T = \Phi_M \cdot D \cdot \Phi_{M-1} \cdots D \cdot \Phi_1$$

$$\Phi = \begin{bmatrix} e^{j\theta_1} & 0 & 0 & 0 \\ 0 & e^{j\theta_2} & 0 & 0 \\ 0 & 0 & e^{j\theta_3} & 0 \\ 0 & 0 & 0 & e^{j\theta_4} \end{bmatrix}$$

The diffractive matrix D can be an arbitrary unitary matrix while each phase shifter Φ_i must be tuned to correctly implement a particular matrix. However, as long as there are as many phase shifting stages as there are columns or rows in T, a specific combination of Φ_i's always exist for any T. Although the proof of this is outside the scope of this paper, the method can be thought of an extension of the Mach-Zehnder interferometer (MZI) for more than two channels.

The implementation proposed by Ribeiro et al. also demands that the matrix is unitary but utilizes a different property of unitary matrices. It relies on a pivotal result by Reck et al., which showed that product by a N-by-N unitary matrix can be decomposed into several 2x2 multiplications and a product by a (N-1)-by-(N-1) unitary matrix [9]:

$$U(N) \cdot T_{N,N-1} \cdot T_{N,N-2} \cdot \ldots \cdot T_{N,1} = \left(\begin{array}{cc} \boxed{U(N-1)} & 0 \\ 0 & e^{i\alpha} \end{array} \right)$$

The input unitary matrix, $U(N)$, is multiplied by several transformational matricies $T_{i,j}$, which are identity matrices with a square of elements replaced by a 2-by-2 unitary matrix. Configured with the correct $T_{i,j}$, the resulting unitary matrix has one less dimension, aside from a phase difference. The process can be repeated until we are left with only 2-by-2 matrices, which are much simpler operations easily accomplished in optics using MZIs.

Lastly, the implementation proposed by Tait et al. relies on matrix multiplication in its simplest form: repeated multiplication and addition. Each element in the input vector is multiplied with an element in the matrix and summed together. While this requires an optical element for every element of the matrix, among other disadvantages, its simplicity and lack of preprocessing allows it to be applied in ways that the earlier methods can't.

4 Comparing Device Design (Component Choice)

Each method of matrix multiplication described above can be efficiently implemented using a small set of nanophotonic components. In principle, these designs should be relatively simple, but practical considerations require additional components, such as for off-chip coupling, readout, tuning, or error correction.

In Tang et al.'s implementation, the device structure is straightforward: phase shifting matrices are implemented using phase shifters and diffractive matrices are implemented using directional couplers. The final schematic, shown in Figure 1a, is a series of phase shifters and directional couplers cascaded, matching one-to-one to the mathematical description. Most of the on-chip complexity lies in the phase shifters: since the authors implemented a 10-by-10 matrix, the design required the control of over a hundred phase shifters. The authors also tested transverse-electric and transverse-magnetic polarization of input light, necessitating an additional component.

Ribeiro et al.'s implementation is similarly straightforward: each 2-by-2 matrix multiplication

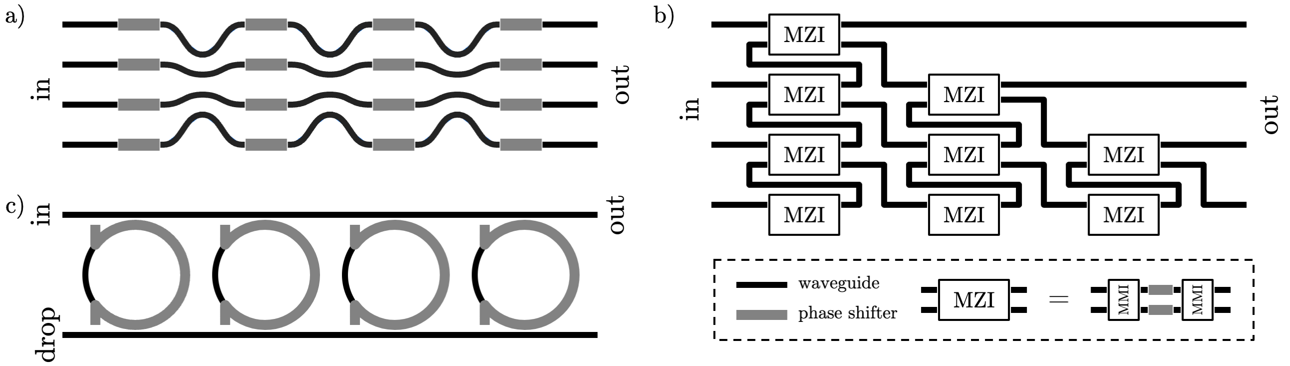

Figure 1. (a) Schematic of device proposed by Tang et al., showing alternating series of phase shifters and directional couplers. (b) Schematic of device proposed by Ribeiro et al., showing cascaded set of Mach-Zehnder interferometers to stepwise decompose the matrix. (c) Schematic of device proposed by Tait et al., showing input/output and drop buses with ring resonators.

is implemented using a Mach-Zehnder interferometer and phase shifters map to phase shifters. The MZI is implemented using two multi-mode interferometers and a phase shifter on each arm. In the resulting schematic, shown in Figure 1b, the stepwise process of the algorithm is evident: each stage reduces the dimensionality of the operation by one until the final stage is a 1-by-1 matrix product, i.e. a feedthrough. In the actual implementation, the authors additionally tapped the output of each MZI to a grating coupler using a 1% efficient directional coupler to monitor and tune the phase shifters.

Unlike the previous two implementations, Tait et al.'s implementation encodes each value in the input vector as a wavelength rather than as a position among several waveguides. This technique, known as wavelength division multiplexing (WDM), allows the device to use a single waveguide to transmit the entire input vector. To address the elements individually, ring resonators are used to selectively filter in or out light of a particular wavelength. In this implementation, shown in Figure 1c, the bus with the initial WDM-encoded input vector is also interpreted as the output vector, with four ring resonators tuned to different wavelengths to selectively eject light of a particular wavelength into a secondary 'drop' bus. The ring resonators are fitted with phase shifters to tune the exact resonant wavelength: if a resonator is set exactly to the wavelength of its target input vector element, the entirety of the light will be dropped, constituting a product by zero. Meanwhile, if the phase shifters shift the resonant wavelength away from that of the input vector element, most of the optical power will stay in the input/output bus, constituting a product by one. Any coefficient between zero and one can be created by tuning the phase shifter between these extremes. Note, however, that there isn't an explicit 'sum' operator to add up the filtered wavelengths: the total output optical power is already the sum of the optical powers of each constituent wavelength. Additionally, this paper only implements product by a 1-by-N matrix, i.e. a weighted sum. However, the concept can be easily extended to a M-by-N matrix multiplier by simply splitting the initial bus into several by using, for example, a directional coupler.

5 Comparing Fabrication

The three implementations choose nearly identical platforms to fabricate their designs. Each use the silicon on insulator (SOI) material platform, i.e. silicon cores with silicon dioxide cladding. They also choose to target the same 1550nm telecom wavelength, though Tait et al. varied the wavelength between 1546nm and 1552nm to encode the different channels of the input vector. They also use thermo-optic phase shifters, though of different materials: Ribeiro et al. use titanium heaters, Tait et al. use Ti/Pt/Au heaters, and Tang et al. doesn't specify. The dimensions of the waveguides are also similar: all three have 220nm thick waveguides, with Tang et al. choosing a 400nm wide waveguide, Tait et al. choosing a 500nm wide waveguide, and Ribeiro et al. not specifying.

Fabrication of the devices is equally uninteresting: Tang et al. and Ribeiro et al. merely mentioned using external fabrication services to make their devices while Tait et al. additionally specified the electron beam lithography based procedure they followed to manufacture their device.

However, these similarities do allow an apples-to-apples comparison of device characteristics, such as footprint, power consumption, and accuracy, based solely on device principles. Unfortunately, the three papers aren't consistent in specifying these details. For footprint, dimensions are estimated based on figures and photos with scales. Tang et al.'s device is about $0.5\,\mathrm{mm}^2$ for a single stage, and with 14 stages, the footprint is $7\,\mathrm{mm}^2$ total. Meanwhile, Ribeiro et al.'s entire device is $2\,\mathrm{mm}^2$ and Tait et al.'s device is only $0.02\,\mathrm{mm}^2$. Since these papers implement multiplication of different matrix sizes, these metrics must be normalized to be comparable. For a single matrix element, Tang et al.'s design uses $7\,\mathrm{mm}^2 \cdot (10 \cdot 10)^{-1} = 0.070\,\mathrm{mm}^2$, Ribeiro et al.'s design uses $2\,\mathrm{mm}^2 \cdot (4 \cdot 4)^{-1} = 0.125\,\mathrm{mm}^2$, and Tait et al. uses $0.02\,\mathrm{mm}^2 \cdot (1 \cdot 4)^{-1} = 0.005\,\mathrm{mm}^2$. Note, however, that these figures don't consider necessary off chip support.

6 Discussion

Our original objective was to access each of these designs as a potential machine learning accelerator. Now, after thoroughly considering each of them, we may ask: which one's the best?

Today's largest models have trillions of parameters, necessitating the multiplication of matrices with millions of elements. Therefore, minimizing the footprint of each element is key. The calculations above show that Tait et al.'s proposal uses the least area per element among the three designs. However, the design requires the input vector to be already encoded using WDM—this may be difficult for larger vectors given necessary wavelength spacing. Additionally, unlike Tang et al. and Ribeiro et al.'s designs, two matrix multiplication operations cannot be cascaded since the output vector is not WDM encoded.

Tang et al. and Ribeiro et al.'s designs are more directly comparable since both require unitary matrices and encode input vector across multiple waveguides rather than with WDM. The metrics above put Tang et al.'s design in the lead, using about half as much area per element as Ribeiro et al.'s design. However, since neither author attempted to optimize their design for area, it is entirely possible for one design to overtake the other in a later iteration.

A more interesting comparison is between these two designs and Tait et al.'s towards applicability in different stages of a model's lifetime. Generally, there are two: training and inference. During training, the model's weights are modified according to some algorithm, such as back propagation, to better model a dataset. Then, during inference, the model is deployed to make decisions on previously unseen data. This is relevant because Tait et al.'s design exposes matrix element values directly as changes in the resonance of a corresponding ring resonator whereas Tang et al. and Ribeiro et al.'s require numerical preprocessing of the matrix to determine the necessary phase shifts to correctly implement the matrix. By exposing matrix values directly, the learning algorithm can also be implemented on the same chip as the matrix multiplier. This was demonstrated using a similar ring resonator/WDM configuration by Feldmann et al. using Hebbian learning [10]. Meanwhile, Tang et al. and Ribeiro et al.'s designs are more suited for the inference stage given the preprocessing required.

7 Future Outlook and Conclusion

We reviewed three implementations of photonic matrix multiplication, comparing them based on their mathematical underpinnings, component choice, and fabrication, with a focus on applicability to machine learning. Overall, each design shows potential as the basis of a future AI accelerator, whether that be in training or inference. However, none of the designs are quite there yet: these implementations don't demonstrate inference of an actual AI model, so potential issues in terms of accuracy, reliability, or robustness may need to be worked out before a practical matrix multiplier can be deployed.

Once these details are solved, however, photonic matrix multipliers may become an unrivaled tool in artificial intelligence, if not a necessity. Harnessing the speed of light, they promise unmatched speed, throughput, and power efficiency—features increasingly valuable as leading AI models become larger and more power hungry. Overall, photonics highlight a potential path towards better, brighter AI systems.

8 References

[1] A. S. Luccioni, Y. Jernite, and E. Strubell, "Power hungry processing: Watts driving the cost of ai deployment?" (2023).

[2] H. Iwai, “Future of nano cmos technology,” Solid-State Electronics **112**, 56–67 (2015).

[3] H. Zhou, J. Dong, J. Cheng, W. Dong, C. Huang, Y. Shen, Q. Zhang, M. Gu, C. Qian, H. Chen, Z. Ruan, and X. Zhang, “Photonic matrix multiplication lights up photonic accelerator and beyond,” Light: Science Applications 2022 11:1 **11**, 1–21 (2022).

[4] R. Tang, R. Tanomura, T. Tanemura, and Y. Nakano, “Ten-port unitary optical processor on a silicon photonic chip,” ACS Photonics **8**, 2074–2080 (2021).

[5] A. Ribeiro, L. Vanacker, W. Bogaerts, and A. Ruocco, “Demonstration of a 4x4-port universal linear circuit,” Optica, Vol. 3, Issue 12, pp. 1348-1357 **3**, 1348–1357 (2016).

[6] A. N. Tait, P. R. Prucnal, T. F. de Lima, B. J. Shastri, and M. A. Nahmias, “Multi-channel control for microring weight banks,” Optics Express, Vol. 24, Issue 8, pp. 8895-8906 **24**, 8895–8906 (2016).

[7] J. Zou, Y. Han, and S. S. So, “Overview of artificial neural networks,” Methods in Molecular Biology **458**, 15–23 (2008).

[8] R. Tanomura, R. Tang, S. Ghosh, T. Tanemura, and Y. Nakano, “Robust integrated optical unitary converter using multiport directional couplers,” Journal of Lightwave Technology **38**, 60–66 (2020).

[9] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, “Experimental realization of any discrete unitary operator,” Physical Review Letters **73**, 58 (1994).

[10] J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. H. Pernice, “All-optical spiking neurosynaptic networks with self-learning capabilities,” Nature 2019 569:7755 **569**, 208–214 (2019).