Invited Paper

Shedding Light on LLMs: Harnessing Photonic Neural Networks for Accelerating LLMs

Salma Afifi
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, CO, USA
Salma.afifi@colostate.edu

Sudeep Pasricha
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, CO, USA
Sudeep@colostate.edu

Mahdi Nikdast
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, CO, USA
Mahdi.nikdast@colostate.edu

ABSTRACT

Large language models (LLMs) are foundational to the advancement of state-of-the-art natural language processing (NLP) and computer vision applications. However, their intricate architectures and the complexity of their underlying neural networks present significant challenges for efficient acceleration on conventional electronic platforms. Silicon photonics offers a compelling alternative. In this paper, we describe our recent efforts on developing a novel hardware accelerator that leverages silicon photonics to accelerate transformer neural networks integral to LLMs. Our evaluation demonstrates that the proposed accelerator delivers up to 14× higher throughput and 8× greater energy efficiency compared to leading-edge LLM hardware accelerators, including CPUs, GPUs, and TPUs.

CCS CONCEPTS

• Computer systems organization → Optical computing • Computer systems organization → Neural Networks • Hardware → Application-specific VLSI designs

KEYWORDS

Photonic computing, large language models, inference acceleration, optical computing.

ACM Reference format:

Salma Afifi, Sudeep Pasricha, and Mahdi Nikdast. 2024. Shedding Light on LLMs: Harnessing Photonic Neural Networks for Accelerating LLMs. In *Proceedings of ACM/IEEE International conference on computer-aided design (ICCAD'24). Newark, New Jersey, USA, 8 pages.*

ICCAD '24, October 27–31, 2024, New York, NY, USA

ACM 979-8-4007-1077-3/24/10…
https://doi.org/10.1145/3676536.3697137

1 Introduction

In recent years, large language models (LLMs) and transformer-based architectures have emerged as the cornerstone of modern AI applications, demonstrating unprecedented success across diverse fields such as natural language processing (NLP), computer vision, and beyond. These models, epitomized by architectures like GPT-4 and BERT, owe their remarkable performance to complex neural structures and the transformative self-attention mechanism [1]. However, the exponential growth in model size and complexity has exacerbated challenges related to computational efficiency, power consumption, and scalability, particularly in the post-Moore's law era [2].

Conventional hardware accelerators, primarily designed for earlier neural network paradigms like recurrent neural networks (RNNs) and convolutional neural networks (CNNs), struggle to meet the unique demands of LLMs and transformers. The inherent limitations of electronic accelerators—such as lengthy inference times, significant memory overhead, and reduced energy efficiency—are becoming more pronounced as these models scale up. As a result, there is an urgent need for innovative hardware solutions that can effectively address these challenges [1].

Silicon photonics, traditionally utilized for high-throughput communication in Telecom and Datacom applications, has recently garnered attention as a promising technology for accelerating deep learning tasks at the chip level [2]-[10]. Leveraging CMOS-compatible photonic components, such as optical photodetectors and microring resonator (MR) modulator devices, silicon photonics offers a pathway to ultra-fast, energy-efficient computation, particularly for operations like matrix-vector multiplication, which are fundamental to LMMs. The potential for achieving superior

performance scaling with reduced energy costs makes silicon photonics an attractive alternative to purely electronic solutions.

In this paper, we review our recent work on developing a novel approach to accelerating neural networks leveraged in LLMs using silicon photonics. Our hardware architecture is designed to harness the high-speed and low-energy benefits of photonic operations while addressing the unique computational demands of transformers. By integrating silicon photonics into the acceleration pipeline, our approach not only enhances throughput and energy efficiency, but also pushes the boundaries of what is possible with current AI hardware, offering a scalable solution for next-generation LLM and transformer workloads.

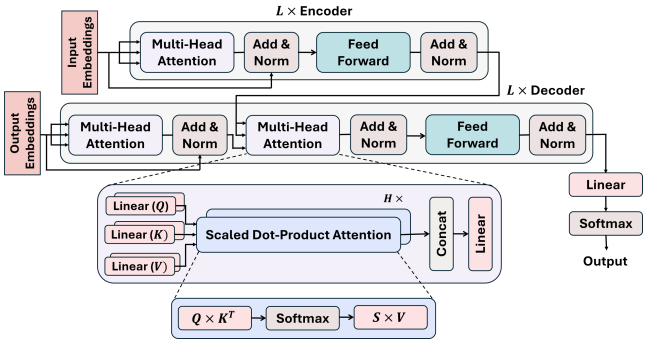


Fig. 1. Transformer neural network architecture overview.

2 Background

2.1 Large Language Models

LLMs are highly sophisticated language models, distinguished by their immense parameter sizes and exceptional learning capabilities. The foundational architecture for many of these models is the transformer, first introduced in 2017 [11]. As illustrated in Fig. 1, the transformer model consists of two primary components: the encoder and the decoder. The encoder is responsible for converting the input sequence into a continuous, abstract representation, while the decoder incrementally processes this representation to generate outputs, each building on the previous one. Typically, both the encoder and decoder are composed of *N* stacked layers, with each layer containing multi-head attention (MHA) and feed-forward (FF) sub-blocks, complemented by residual connections and layer normalization. The most complex operation within the transformer—the self-attention mechanism—occurs within the MHA block. This mechanism utilizes multiple self-attention heads (H), each generating query (Q), key (K), and value (V) vectors to compute the scaled dot-product attention:

$$Head(X) = attention(Q,K,V) = softmax\left(QK^T/\sqrt{d_K}\right)V, \quad (1)$$

where X represents the input matrix and d_K is the dimension of Q and K. The outputs of the self-attention heads are then concatenated and passed through a linear layer. The FF network follows, consisting of two dense layers separated by a ReLU activation.

Recent LLMs, such as BERT [12], primarily utilize the transformer encoder block, while models like the vision transformer (ViT) are structured with *N* encoder layers followed by a multi-layer perceptron [13]. The complexity and variety of operations within LLMs present significant challenges for their acceleration.

2.2 LLM Hardware Acceleration

To address the memory bottlenecks associated with the large parameter sizes of LLMs, previous efforts have primarily targeted either specific subsets of transformer models or individual layers within the LLMs architecture. For example, [14] introduced an FPGA-based hardware accelerator designed to enhance the performance of the MHA and FF layers by efficiently partitioning their weight matrices, enabling shared hardware resources. Another FPGA-based framework presented in [15] employed a pruning technique and a method for storing sparse matrices to optimize acceleration. Additionally, the TransPIM accelerator [16] utilized an in-memory computing approach, incorporating a novel token-based dataflow to optimize data movements, along with modifications to high-bandwidth memory. The automated VAQF framework in [17] was developed to guide the quantization and FPGA resource mapping specifically for ViTs. Unlike these specialized approaches, our proposed hardware architecture can accelerate a wide range of transformer neural networks, supporting diverse LLMs for both NLP and computer vision applications.

2.3 Silicon Photonics for ANN Acceleration

Recent advancements in optical artificial neural network (ANN) accelerators have demonstrated significant improvements in energy efficiency and performance, making them a compelling alternative to traditional electronic-based hardware accelerators [18]. As a result, the use of silicon photonics for accelerating ANNs has attracted considerable interest from both academia and industry. Previous research has primarily focused on optical acceleration for CNNs, multilayer perceptrons (MLPs), RNNs, and graph neural networks (GNNs). For example, the Crosslight architecture was developed as an optical accelerator for CNNs, featuring optimized vector dot product units for convolution and fully connected layers [19]. Similarly, optical accelerators have been proposed for sparse neural networks [20], RNNs [21], and GNNs [5], showcasing substantial gains in throughput and energy efficiency. However, there has been limited exploration of optical acceleration for more complex models, such as LLMs.

In contrast to these previously discussed neural networks, LLMs present unique challenges due to their complex operations, which involve not only massive matrix multiplications (MatMuls) but also various other operations computed between these MatMuls. Previous efforts in accelerating ANNs using silicon photonics have typically focused on performing MatMuls and vector multiplications optically. However, applying a similar approach to LLMs would necessitate frequent data conversions from the optical domain (for multiplications) to the digital domain (for other operations), resulting in increased latency, higher energy consumption, and the need for substantial on-chip buffering. Our proposed architectural design addresses this challenge by efficiently mapping and pipelining LLM operations across various opto-electric blocks, enabling the execution of most operations—not just multiplications—within the optical domain.

2.4 Optical Devices and Computations

The process of accelerating ANNs using silicon photonics involves imprinting the network's parameters onto optical signals, enabling the execution of multiply-and-accumulate (MAC) operations. This can be achieved using either coherent or non-coherent optical architectures. Coherent architectures utilize a single wavelength, with parameters imprinted onto the optical signal's phase. In contrast, non-coherent architectures leverage multiple wavelengths, with parameters imprinted onto the optical signal's amplitude, allowing for parallel operations across different wavelengths [23].

Our proposed architecture, first introduced in [6], represents the first optical accelerator specifically designed for LLMs, utilizing non-coherent silicon photonics. The core operations in the accelerator are carried out using opto-electronic MR devices, and each device can be tuned to operate at specific a wavelength, known as the resonant wavelengths (λ_{MR}), defined as follows:

$$\lambda_{MR} = \frac{2\pi R}{m} n_{eff}, \tag{2}$$

where R is the MR radius, m is the order of the resonance, and n_{eff} is the effective index of the device. By carefully adjusting the effective index (n_{eff}) of the MR device using a tuning circuit, electronic data can be modulated onto optical signals, resulting in predictable changes in the optical signal's wavelength amplitude as shown in Fig. 2(a).

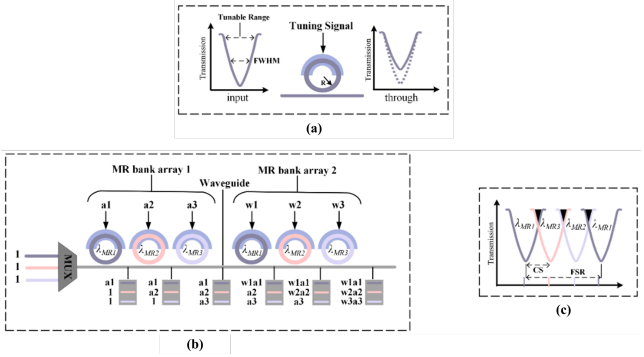


Fig. 2. (a) MR input and through ports' wavelengths after imprinting a parameter onto the signal; (b) MR bank arrays are used to perform multiplication by imprinting input vector (a_1-a_3), followed by weight vector (w_1-w_3); (d) MR bank response and heterodyne crosstalk shown in black, where CS is channel spacing and FSR is free spectral range.

To enhance throughput and mimic the behavior of neurons in ANNs, our design employs wavelength-division multiplexing (WDM), where multiple optical signals with different wavelengths are multiplexed into a single waveguide. As illustrated in Fig. 2(b) these signals then pass through arrays of MR devices, enabling the simultaneous execution of multiple multiplication operations. As shown, an input activation vector $[a_1, a_2, a_3]$ is multiplied by a weight vector $[w_1, w_2, w_3]$, with the resulting optical signals representing the output vector. However, non-coherent silicon photonics presents challenges such as heterodyne or incoherent crosstalk, which occurs when segments of optical signals from adjacent wavelengths interfere with the MR spectrum of another wavelength (see Fig. 2(c)). Our work addresses these challenges through careful design and optimization, as discussed in the subsequent sections.

3 Silicon Photonic LLM Hardware Accelerator

Accelerating LLMs requires a highly efficient hardware architecture that can adapt to the specific demands of these complex neural networks. In this section, we discuss our silicon-photonic-based accelerator designed specifically for LLMs that was first introduced in [6]. We begin by discussing the tuning circuit design utilized in our accelerator, followed by a detailed exploration of the various optimizations applied to the MR devices. Finally, we present the architecture and design considerations of our optical hardware accelerator tailored for LLMs.

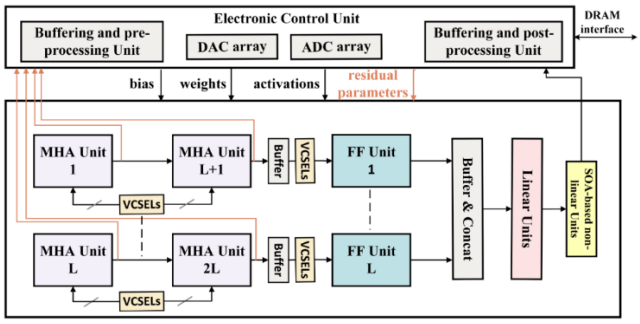


Fig. 3. Overview of the proposed accelerator architecture, from [6].

3.1 MR Tuning Circuit Design

MR devices in non-coherent architectures necessitate an effective tuning circuit, which can be based on electro-optic (EO) or thermo-optic (TO) mechanisms. EO tuning offers rapid operation and lower power consumption but is limited in tuning range. In contrast, TO tuning supports a broader tuning range but comes with increased latency and higher power consumption [21]. To leverage the strengths of both methods while mitigating their weaknesses, our design employs a hybrid tuning approach. EO tuning is used for fast adjustments requiring small shifts in the resonant wavelength ($\Delta\lambda_{MR}$), while TO tuning is reserved for larger shifts. Additionally, we incorporate the thermal eigenmode decomposition (TED) method [20] to minimize the power consumption associated with TO tuning and to reduce thermal crosstalk, ensuring efficient and precise operation of the photonic accelerator.

3.2 MR Bank Design-Space Analysis

Operating in the analog photonic domain introduces various noise sources that can disrupt the correct execution of LLMs. These

noise sources include thermal crosstalk, heterodyne (inter-channel) crosstalk, and homodyne (intra-channel) crosstalk [1]. Our TED-based tuning mechanism, as discussed earlier, mitigates thermal crosstalk between TO tuning circuits. However, in non-coherent architectures, where multiple wavelengths share the same waveguide, heterodyne crosstalk arises as illustrated in Fig. 2(c). This occurs when a portion of an optical signal from one wavelength leaks into the MR spectrum of an adjacent wavelength, leading to signal distortion.

To effectively manage heterodyne crosstalk, it is essential to minimize spectral overlap by optimizing several key factors, including channel spacing (CS), Q-factor tuning, and ensuring that the signal-to-noise ratio (SNR) surpasses the photodetector's sensitivity. The MR design must also provide an adequate tunable range, represented mathematically as 2×FWHM (full width half maximum), to enable error-free parameter imprinting. We optimize MR design for high FWHM and high SNR using the following models:

$$SNR\ (dB) = 10 \times \log_{10}(P_{signal}/P_{noise}), \tag{3}$$

$$P_{signal} = \Phi(\lambda_i, \lambda_j, Q) P_s(\lambda_i, \lambda_j), \tag{4}$$

$$P_{noise} = \sum_{i=1}^{n} \Phi(\lambda_i, \lambda_j, Q) P_S(\lambda_i, \lambda_j)(i \neq j), \tag{5}$$

where Φ is the crosstalk coefficient of the inter-channel crosstalk between neighboring channels λ_i and λ_j, which is given by:

$$\Phi(\lambda_i, \lambda_j, Q) = \left(1 + \left(\frac{2Q(\lambda_i - \lambda_j)}{\lambda_j}\right)^2\right)^{-1}. \tag{6}$$

Here, $(\lambda_i - \lambda_j)$ represents the channel spacing CS, which is an optimizable parameter within the free spectral range (FSR) under consideration. The signal power P_S reaching the MR sensitive to λ_j, is defined as:

$$P_S = \psi(\lambda_i, \lambda_j) P_{in}(i), \tag{7}$$

where P_{in} is the input power to the waveguide, and ψ represents the signal power loss before the MR with resonance wavelength λ_j . The crosstalk-induced power suppression in the waveguide can be modeled as a through loss, calculated as γ times the signal power before passing through the MR, where γ and ψ are given by:

$$\gamma(\lambda_i, \lambda_j, Q) = \left(1 + \left(\frac{2Q(\lambda_i - \lambda_j)}{\lambda_j}\right)^{-2}\right)^{-1}, \tag{8}$$

$$\psi(\lambda_i, \lambda_j) = \prod_{k=1}^{(k-1)<j} \gamma(\lambda_i, \lambda_k, Q). \tag{9}$$

The FWHM is calculated using the following model:

$$FWHM = \frac{\lambda_{res}}{Q - factor}, \tag{10}$$

where λ_{res} is the resonant wavelength of the MR being considered. To ensure the lowest optical power level (P_{lpar}) remains higher than P_{noise} relative to P_{signal}, the following relationship must hold:

$$10\log_{10}\left(\frac{P_{signal}}{P_{lpar}}\right) < 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right), \tag{11}$$

where P_{lpar} is defined in terms of P_{signal} as:

$$P_{lpar} = \frac{P_{signal} \times R_{tune}}{N_{levels}}. \tag{12}$$

Substituting P_{lpar} in (11) yields:

$$10\log_{10}\left(\frac{N_{levels}}{R_{tune}}\right) < SNR, \tag{13}$$

where N_{levels} represents the number of amplitude levels required to represent across the available R_{tune}. For an n-bit parameter representation, N_{levels} will be 2^n. In cases where positive and negative values are separately represented, N_{levels} will be 2^{n-1}. The relationship between R_{tune} and SNR is given by:

$$R_{tune} > N_{levels} \times 10^{-\frac{SNR}{10}}. \tag{14}$$

By utilizing these models, we can determine the optimal design space for our MR banks, ensuring high SNR and tunable range. Based on our design-space exploration analysis, explained thoroughly in [6], the optimal values for R_{tune}, *Q, SNR,* and *CS* are 0.45, 6500, 24.3, and 1, respectively.

3.3 Multi-Head Attention (MHA) Unit Design

The primary challenge in accelerating transformer inference lies in the time-intensive and successive MatMuls. However, these operations can be decomposed into vector dot-product operations, as demonstrated in prior work on optical CNN acceleration [24]. Focusing on the self-attention computations within each head, the MatMul operation ($Q.K^T$) cannot proceed until the K^T matrix is generated and stored. This sequence creates significant power and latency overhead, as it necessitates generating the *K* matrix ($K = XW_K$) optically, converting it to the digital domain, buffering the values, generating K^T, and then converting it back to the optical domain to perform the next MatMul ($Q.K^T$). By using MatMul decomposition, this operation can be restructured into two cascaded MatMul steps:

$$Q.K^T = Q.(X.W_K)^T = (Q.W_K^T).X^T. \tag{15}$$

This approach, illustrated by the top four MR bank arrays in Fig. 4(a), eliminates the need for intermediate buffering during the computation of $Q.K^T$. The first two MR bank arrays generate Q, and with W_K^T and X^T pre-stored and used to tune the MRs in the

subsequent two MR bank arrays, the output of equation (15) can be obtained directly in the optical domain without any intermediate buffering or costly opto-electric conversions. To further reduce latency and power overhead, we propose incorporating the scaling factor in equation (1) directly into the weight matrix (W_K^T) stored in the electronic control unit (ECU). This modification allows the MR tuning values to be $W_K^T{}_i/\sqrt{d_k}$, eliminating the need for an additional MR bank array for the scaling operation.

In most optical ANN accelerators, such as [19], MatMul operations are executed sequentially, using separate MAC units with MR bank arrays to perform the multiplications. These partial sums are then accumulated and added. However, since multiple consecutive MatMul operations are involved in the attention computation, our approach avoids intermediate value accumulation. Instead, the individual multiplication results from the first MR bank array are passed directly to the following MR bank arrays, with the final summation of all multiplications and partial sums occurring just before the softmax block, as depicted in Fig. 4(a). This method minimizes the latency and power costs associated with early summations, intermediate buffering, and opto-electric conversions. Additionally, as discussed in Section 3.2, we have minimized crosstalk noise, which is typically a concern with such MR arrangements.

Following the computation of ($Q.K^T$) by the upper MR bank arrays in Fig. 4(a), partial sums are accumulated using balanced photodetectors (BPDs). BPDs accommodate both positive and negative parameter values by placing separate positive and negative arms for the same waveguide. The BPD subtracts the sum from the negative arm from the positive arm, with the results then converted to the digital domain for softmax computation.

Another challenge in the MHA is the softmax operation, which restricts parallelism as it requires the completion of all previous MatMul results. To address this, we propose two optimization strategies. First, we avoid computationally expensive division and numerical overflow by using the log-sum-exp trick, as employed in previous works like [7]:

$$Softmax(\chi_i) = \frac{\exp(\chi_i - \chi_{max})}{\sum_{j=1}^{d_k} \exp(\chi_j - \chi_{\max})}, \quad (16)$$

$$= \exp\left(\chi_i - \chi_{max} - \ln\left(\sum_{j=1}^{d_k} \exp(\chi_j - \chi_{max})\right)\right).$$

The softmax operation is thus divided into four tasks: finding χ_{max}, subtraction, natural logarithm (*ln*), and exponential *(exp)*. Finding χ_{max} and subtraction can be handled by simple digital circuits. As shown in Fig. 4(a), the ADC output is buffered while also being fed to a comparator circuit, allowing χ_{max} to be computed in parallel with the MatMuls. The *ln* and *exp* operations can be computed using look-up tables (LUTs) [23], which also provide the final softmax output as an analog value from the LUT's memristor cell, which is directly used to tune the MR bank array. Additionally, our scaled dot-product attention design enables high parallelism by synchronizing the bottom vertical cavity surface emission laser (VCSEL) array (Fig. 4(a)) to activate only when the softmax operation is complete.

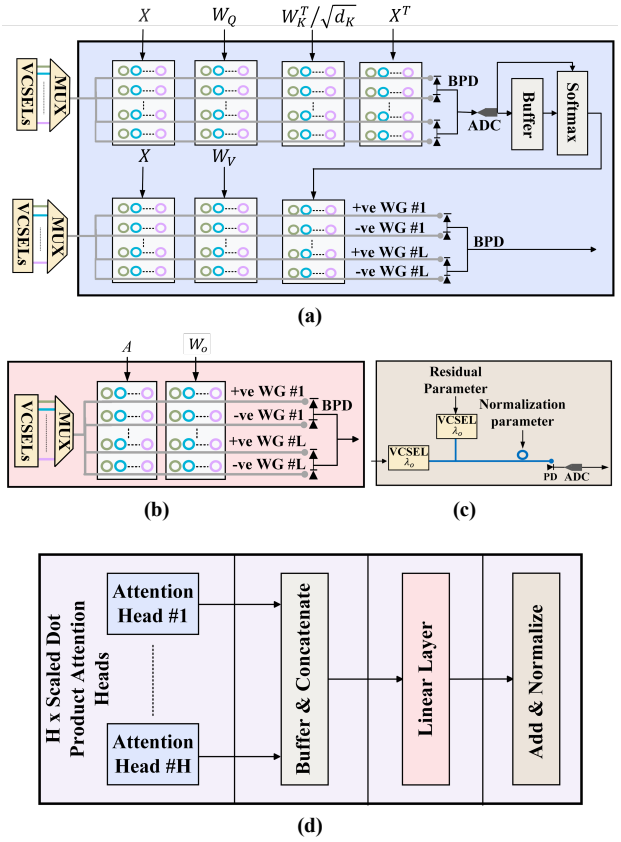


Fig. 4. (a) Attention head unit comprised of seven MR bank arrays for MatMul operations, each with dimension $K \times N$; (b) Linear layer comprised of an MR bank array with dimension $K \times N$; (c) Add and Normalization layers using coherent photonic summation and an MR for imprinting the normalization parameter; (d) MHA unit composed of H attention heads, buffer and concatenate block, linear layer, and an add and normalize block. From [6].

The linear layer in MHA is also implemented optically using two MR bank arrays (Fig. 4(b)). To add the MHA input to its current output, thereby implementing the residual connection, we employ coherent photonic summation. As shown in Fig. 4(c), the output signal from the linear layer directly drives a VCSEL with wavelength λ_o. Another VCSEL with the same wavelength is driven by the residual connection value(i), and when the two waveguides meet, they interfere, resulting in the summation of the two values. Coherent summation is achieved using a laser phase-locking mechanism [24], ensuring that the VCSEL output signals are phase-aligned for constructive interference. Finally, layer normalization (LN) is performed optically using a single MR tuned by the LN parameter, as depicted in the complete MHA architecture in Fig. 4(d).

3.4 Feed Forward (FF) Unit Design

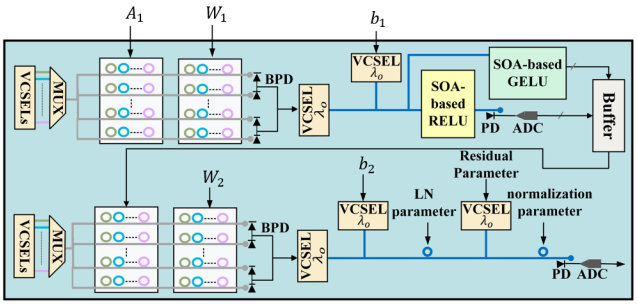
The FF unit, depicted in Fig. 5(a), comprises two fully connected (FC) layers, separated by a non-linear activation function. Each FC layer is accelerated using two MR bank arrays with dimensions $K \times N$: one array imprints the input activations, while the other performs the matrix multiplication between the inputs and the weight matrices. Bias values are added via coherent photonic summation, as discussed earlier. For the non-linear activation, an optical ReLU unit is implemented using semiconductor-optical-amplifiers (SOAs). When the gain in an SOA is adjusted near 1, it exhibits nearly linear behavior, mimicking the ReLU function. Previous work [25] has shown how SOAs can also be utilized to implement other non-linear functions, such as sigmoid and tanh. This expands our architecture's scope to include optical implementation of the GELU function, used in ViT, instead of ReLU. The GELU function can be approximated as follows [22]:

$$GELU(x) = x\Phi(x) = 0.5x(1 + \tanh\left[\sqrt{2/\Pi}\,(x + 0.044715x^3)\right]) = x\sigma(1.702x). \quad (17)$$

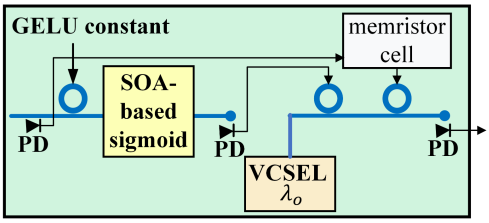
As illustrated in Fig. 5(b), the first multiplication between 1.702 and x is performed using a single MR, while the Sigmoid function is computed using the SOA-based implementation described above. The final multiplication of the input with the sigmoid output is carried out using two MRs. A low-power local storage mechanism is employed to store the analog input signal from the photodetector (PD) in a memristor cell, which is then used to directly tune the second MR. The output from the non-linear unit is buffered and used to tune the MRs in the first bank array of the second FC layer (Fig. 5(b)), where it is multiplied by the weight matrix (W_2). After the second FC layer, a normalization layer is implemented using an MR. The residual connection is then added through coherent photonic summation, followed by a final normalization layer, also implemented with an MR.

3.5 LLM Hardware Accelerator Architecture

The overall architecture (Fig. 3) is tailored to accelerate a range of transformer models used across different LLMs. It consists of two sets of MHA units and one set of FF units, each with a dimension of L. This configuration allows for efficient reuse of units across both the encoder and decoder blocks. In the encoder block, the first VCSEL array drives the input exclusively to the second set of MHA units. The MHA unit itself is split into two segments: one before and one after the softmax operation. Since the softmax operation (as defined in equation (1)) cannot be executed until the first segment is completed, these segments cannot be parallelized. However, the MatMul operations in the second segment can be run in parallel with those in the FF unit. In the decoder block, the first VCSEL array drives the input to the initial set of MHA units, with its output serving as the input for the second MHA unit, which in turn drives the FF unit. This method of VCSEL-reuse reduces laser power consumption and minimizes inter-channel crosstalk by sharing single VCSEL arrays across rows in each MR bank array to imprint input activations.



(a)



(b)

Fig. 5. (a) FF block composed of four-MR bank arrays with dimensions $K \times N$, SOA-based ReLU and GELU units, and bias and residual connection additions, done with coherent photonic summation; (b) GELU unit composed of three MRs, a semiconductor-optical-amplifiers (SOA), and a VCSEL. From [6].

4 Experiments and Results

We conducted extensive simulation-based analyses to evaluate the efficiency of the proposed hardware architecture. Our study focused on accelerating four transformer neural network models: Transformer-base [11], BERT-base [12], Albert-base [26], and ViT-base [13]. The model parameters, including the dimensionality of the input/output (d_{model}) and FF layers (d_{ff}), are summarized in Table 1. We developed a Python-based simulator to estimate the area, performance, and energy consumption for running each model. The area, performance, and energy metrics for all electronic buffers in TRON were estimated using CACTI [27] at 28nm. Additionally, the electronic circuit for the softmax operation was synthesized using Xilinx Vivado at 28nm, and the power/delay estimates from this synthesis were incorporated into our analysis. Model training and accuracy assessments were performed using TensorFlow 2.9.

The accuracies achieved and the datasets associated with each model are summarized in Table 2. The Transformer, BERT, and Albert models were applied to NLP tasks, including language translation and sentiment analysis. The ViT model was evaluated on an image classification task, with pre-training on ImageNet followed by fine-tuning on CIFAR-10. Our analysis revealed that 8-bit model quantization yields accuracy levels comparable to those of full (32-bit) precision models (see Table 2); hence, we focused on 8-bit precision for transformer models.

The specific architectural details of each hardware accelerator such as the numbers of the computational blocks, were determined through detailed design space analysis as explained in [6]. More

details of optoelectronic parameters considered in our analysis are also described in the same work.

Table 1: Transformer models and parameter counts, from [6].

Model	Params	Layers	Heads	d_{model}	d_{ff}
Transformer-base	52M	2	8	512	2048
BERT-base	108M	12	12	768	3072
Albert-base	12M	12	12	768	3072
ViT-base	86M	12	12	768	3072

Table 2: Transformer model performances, from [6].

Model	Dataset(s)	Accuracy (32-bit)	Accuracy (8-bit)
Transformer-base	Ted_hrlr_translate	66.73%	70.4%
BERT-base	Sentiment-Analysis-of-IMDB-Movie-Reviews	85.8%	85.8%
Albert-base	Sentiment-Analysis-of-IMDB-Movie-Reviews	88.3%	88.7%
ViT-base	ImageNet/Cifar-10	97.7%	98.0%

4.1 Architecture Component-Wise Analysis

To provide insights into the key components within our hardware accelerator, TRON, we first provide a detailed breakdown of power consumption and latency in Fig. 7. The analysis reveals that MatMul operations within the attention heads account for more than half of the architecture's power overhead. This significant power consumption is attributed to the large matrix dimensions involved in the MHA blocks of each attention head, which necessitate numerous digital-to-analog converters (DACs), known for their high-power demands. Additionally, the sequential dependencies within the attention head significantly contribute to the overall latency overhead. However, in the Albert model, which shares all attention and feedforward (FF) parameters across layers [4], the number of active DACs is minimized, resulting in reduced overall power consumption.

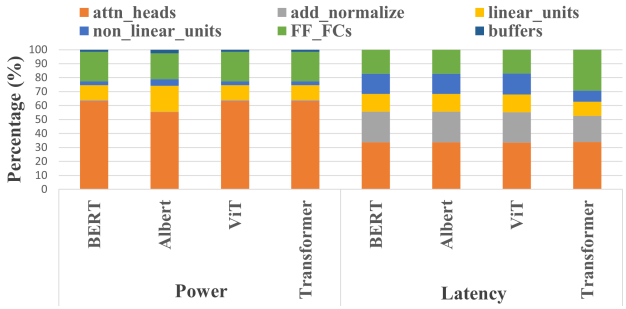


Fig. 6. Power, latency breakdown across TRON components from [6].

4.2 Comparison with State-of-the-Art LLM Accelerators

We evaluated our accelerator's performance by comparing its execution against several processors and state-of-the-art transformer accelerators, including the Tesla V100-SXM2 GPU, TPU v2, Intel Xeon CPU, TransPIM [16], and two FPGA-based transformer accelerators, FPGA_Acc1 [14] and FPGA_Acc2 [15]. VAQF [17], which targets vision transformers, and FPGA_Acc2, designed for traditional encoder-decoder architectures and transformer-based language models, are limited to the models they specifically target. We utilized the reported power, latency, and energy values from these accelerators, along with results from executing models on GPU/CPU/TPU platforms, to estimate the giga-operations per second (GOPS) and energy per bit (EPB) for each model.

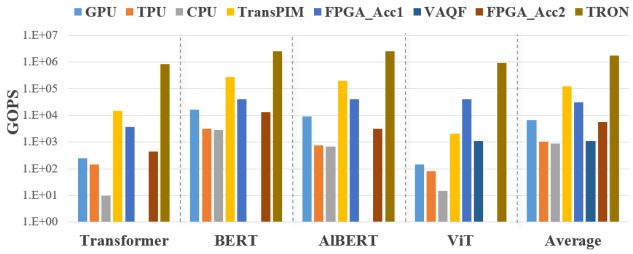


Fig. 7. Throughput comparison across transformer accelerators, from [6].

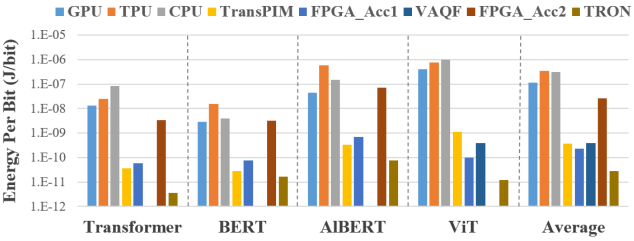


Fig. 8. EPB comparison across transformer accelerators, from [6].

As shown in Fig. 8, TRON delivers an average of 262×, 1631×, 1930×, 14×, and 55× higher GOPS than the GPU, TPU, CPU, TransPIM, and FPGA_Acc1 platforms, respectively. When comparing TRON to transformer-specific accelerators, it achieves an average of 352× higher GOPS than FPGA_Acc2 for transformer, BERT, and Albert models, and 846× higher GOPS than VAQF for ViT. The significant throughput advantage of TRON can be attributed to its high-speed execution in the optical domain and minimal reliance on digital/electronic computations.

Fig. 9 illustrates the EPB comparison. On average, TRON attains 4231×, 12397×, 10971×, 14×, and 8× lower EPB than the GPU, TPU, CPU, TransPIM, and FPGA_Acc1, respectively. When compared to model-specific accelerators, TRON achieves an average of 802× lower EPB than FPGA_Acc2 for transformer, BERT, and Albert models, and 32× lower EPB than VAQF for ViT. These EPB improvements stem from TRON's low-latency operations and relatively lower power consumption compared to the other computational platforms.

5 Conclusion

This paper has presented our recent work on the design of an innovative hardware accelerator called TRON that leverages silicon photonics to boost the performance of transformer neural

networks critical to LLMs. Our architecture demonstrates at least 14× higher throughput and 8× greater energy efficiency compared to eight different processing platforms and state-of-the-art LLM accelerators. These results highlight the significant potential of silicon photonics in enabling energy-efficient, high-throughput inference acceleration for LLMs. Although this work focuses on hardware architecture, integrating software optimization techniques to minimize the memory footprint of LLMs could yield even greater improvements in both throughput and energy efficiency.

REFERENCES

[1] Salma Afifi, Febin Sunny, Mahdi Nikdast and Sudeep Pasricha. 2024. Accelerating Neural Networks for Large Language Models and Graph Processing with Silicon Photonics. *IEEE/ACM DATE*.

[2] Febin Sunny, Mahdi Nikdast and Sudeep Pasricha. 2023. Cross-layer design for ai acceleration with non-coherent optical computing. *ACM GLSVLSI*.

[3] Febin Sunny, Ebad Taheri, Mahdi Nikdast and Sudeep Pasricha. 2024. Silicon Photonic Network-on-Interposer Design for Energy Efficient Convolutional Neural Network Acceleration on 2.5D Chiplet Platforms. *IEEE/ACM DATE*.

[4] Kyle Shiflett, et al. 2021. Albireo: Energy-efficient acceleration of convolutional neural networks via silicon photonics. *ISCA*.

[5] Salma Afifi, Febin Sunny, Amin Shafiee, Mahdi Nikdast and Sudeep Pasricha. 2023. GHOST: A Graph Neural Network Accelerator using Silicon Photonics. *IEEE/ACM CASES (ESWEEK)*.

[6] Salma Afifi, Febin Sunny, Mahdi Nikdast and Sudeep Pasricha. 2023. TRON: Transformer Neural Network Acceleration with Non-Coherent Silicon Photonics. *ACM GLSVLSI*.

[7] Hanqing Zhu, et al. 2024. Lightening-Transformer: A Dynamically-operated Optically-interconnected Photonic Transformer Accelerator. *HPCA*.

[8] Ebad Taheri, Amin Mahdian, Sudeep Pasricha and Mahdi Nikdast. 2024. SwInt: A Non-Blocking Switch-Based Silicon Photonic Interposer Network for 2.5D Machine Learning Accelerators. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*.

[9] Amin Shafiee, Sanmitra Banerjee, Krishnendu Chakrabarty, Sudeep Pasricha and Mahdi Nikdast. 2024. Analysis of Optical Loss and Crosstalk Noise in MZI-based Coherent Photonic Neural Networks. *IEEE/OPTICA Journal of Lightwave Technology*.

[10] Febin Sunny, Amin Mirza, Ishan Thakkar, Mahdi Nikdast and Sudeep Pasricha. 2021. ARXON: A Framework for Approximate Communication over Photonic Networks-on-Chip. *IEEE TVLSI*.

[11] Ashish Vaswani, et al. 2017. Attention is All you Need. *NIPS*.

[12] Jacob Devlin, et al. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*.

[13] Alexey Dosovitskiy, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

[11] Siyuan Lu, et al. 2020. Hardware accelerator for multi-head attention and position-wise feed-forward in the transformer. *IEEE SOCC*.

[12] Panjie Qi, et al. 2021. Accelerating framework of transformer by hardware design[' and model compression co-optimization. *IEEE ICCAD*.

[16] Minxuan Zhou, et al. 2022. TransPIM: A Memory-based Acceleration via Software-Hardware Co-Design for Transformer. *IEEE HPCA*.

[17] Mengshu Sun, et al. 2022. VAQF: Fully automatic software-hardware co-design framework for low-bit vision transformer. *arXiv e-prints*.

[18] Febin Sunny, Ebad Taheri, Mahdi Nikdast and Sudeep Pasricha. 2021. A Survey on Silicon Photonics for Deep Learning. *ACM JETC*.

[19] Febin Sunny, Asif Mirza, Mahdi Nikdast and Sudeep Pasricha. 2021. CrossLight: A Cross-Layer Optimized Silicon Photonic Neural Network Accelerator," *IEEE/ACM DAC*, 2021.

[20] Febin Sunny, Mahdi Nikdast and Sudeep Pasricha. 2022. SONIC: A Sparse Neural Network Inference Accelerator with Silicon Photonics for Energy-Efficient Deep Learning. *IEEE/ACM ASPDAC*.

[21] Febin Sunny, Mahdi Nikdast and Sudeep Pasricha. 2022. RecLight: A recurrent neural network accelerator with integrated silicon photonics. *ISVLSI*.

[22] Den Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv e-prints*.

[23] Febin Sunny, Mahdi Nikdast and Sudeep Pasricha. 2023. Cross-Layer Design for AI Acceleration with Non-Coherent Optical Computing. *ACM GLSVLSI*.

[24] Febin Sunny, Asif Mirza, Mahdi Nikdast and Sudeep Pasricha. 2021. ROBIN: A Robust Optical Binary Neural Network Accelerator. *ACM TECS*.

[25] Kristof Vandoorne, et al. 2011. Parallel Reservoir Computing Using Optical Amplifiers. *IEEE TNN*.

[26] Zhenzhong Lan, et al. 2019. Albert: A lite bert for self-supervised learning of language representations. *ICLR*.

[27] HP Labs: CACTI. [Online]: https://www.hpl.hp.com/research/cacti/.