HyAtten: Hybrid Photonic-digital Architecture for Accelerating Attention Mechanism

Huize Li, Dan Chen✉, Tulika Mitra
School of Computing, National University of Singapore, 119077, Singapore
huizeli@nus.edu.sg, danchen@nus.edu.sg (Corresponding author), tulika@comp.nus.edu.sg

***Abstract*—The wide adoption and substantial computational resource requirements of attention-based Transformers have spurred the demand for efficient hardware accelerators. Unlike digital-based accelerators, there is growing interest in exploring photonics due to its high energy efficiency and ultra-fast processing speeds. However, the significant signal conversion overhead limits the performance of photonic-based accelerators. In this work, we propose HyAtten, a photonic-based attention accelerator with minimize signal conversion overhead. HyAtten incorporates a signal comparator to classify signals into two categories based on whether they can be processed by low-resolution converters. HyAtten integrates low-resolution converters to process all low-resolution signals, thereby boosting the parallelism of photonic computing. For signals requiring high-resolution conversion, HyAtten uses digital circuits instead of signal converters to reduce area and latency overhead. Compared to state-of-the-art photonic-based Transformer accelerator, HyAtten achieves 9.8× performance/area and 2.2× energy-efficiency/area improvement.**

***Index Terms*—photonic computing, attention mechanism, domain specific accelerator**

I. Introduction

Transformer-based neural networks have achieved remarkable success in various domains, such as *natural language processing* (NLP) [1] and *computer vision* (CV) [32]. The core operation of Transformers is the self-attention mechanism, which calculates pairwise correlations between input tokens to enhance inference accuracy. Despite their superior accuracy, the quadratic complexity of self-attention requires substantial computational resources, posing a significant challenge for deploying Transformers, especially in resource-constrained systems. Consequently, there is a pressing need to develop domain-specific hardware accelerators to enable the efficient deployment of Transformers in real-world applications.

Several digital hardware accelerators have been proposed to improve the inference performance of Transformers by reducing redundant memory access and enhancing computational parallelism [1], [9], [11], [32]. While these digital accelerators effectively reduce inference latency, traditional electrical computing platforms face significant limitations as transistor-based chips approach the boundaries of Moore's Law. This results in increased power dissipation, particularly in computation-intensive self-attention processes. As Transformer models continue to grow in size, the high latency and energy consumption faced by digital accelerators will only become more pronounced. In contrast, integrated photonic accelerators present a promising alternative for accelerating deep neural networks, offering ultra-high speeds, extensive parallelism, and low energy consumption.

Various optical systems have been explored to accelerate *convolutional neural networks* (CNNs) [6] and Transformers [34]. However, existing photonic accelerators are highly dependent on high-resolution signal converters to preserve the accuracy of neural network inference, as shown in Figure 2 (a). These high-resolution converters, however, have become a substantial bottleneck, constraining the overall performance of photonic accelerators. For instance, in the state-of-the-art photonic-based Transformer accelerator, Lightening-Transformer [34], signal conversion units, *analog-to-digital converters* (ADCs), *digital-to-analog converters* (DACs), and optical/electrical converters, consume more than 50% of the chip's area. To mitigate the area overhead, Lightening-Transformer shares one ADC among multiple photonic arrays. However, one 32×32 photonic array will generate 1024 signals in one cycle, which can not be efficiently processed by only one ADC. Hence, there is a pressing need to minimize the delay, energy, and area overhead of signal conversion units without sacrificing model accuracy.

Given this context, we introduce HyAtten, a novel photonic-based attention mechanism accelerator designed to minimize signal conversion overhead while maintaining inference accuracy. Our experimental results reveal that over 85% of the analog signals in existing photonic-based Transformer accelerators can be effectively processed using low-resolution converters, such as 4-bit ADCs. Leveraging this insight, HyAtten utilizes low-resolution converters within its photonic circuits to handle these signals, substantially reducing both the latency and area overhead of the photonic components. For the remaining high-resolution signals (exceed the full-scale measurement range of low-resolution converters) that cannot be processed by the low-resolution converters, HyAtten integrates digital circuits to manage these part of attention computations. Importantly, the digital circuits only need to process a small fraction of the data (less than 15%), significantly lowering the overall demand for digital computational resources. The key contributions of HyAtten are as follows:

- We perform extensive experiments on a state-of-the-art photonic Transformer accelerator and find that over 85% of analog signals can be efficiently processed using low-resolution signal converters.
- Based on this insight, we propose HyAtten, an innovative photonic-based attention mechanism accelerator that employs low-resolution converters to handle these signals.
- To further optimize performance, HyAtten incorporates digital circuits to process the remaining high-resolution

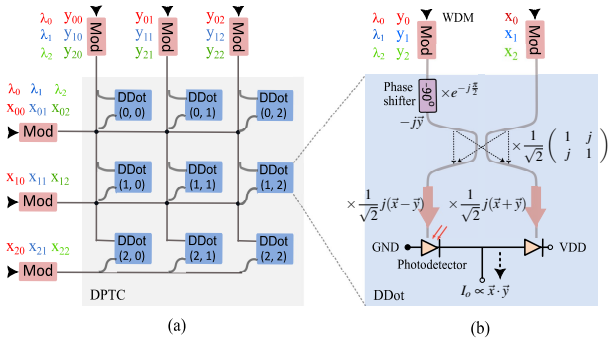


Fig. 1: (a) DPTC array proposed in [34], and (b) DDot unit proposed in [34]

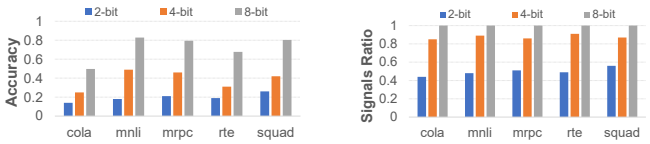


Fig. 2: (a) The model accuracy when employing different ADC resolutions, and (b) the proportion of signals that remain within the resolution limits of various ADCs

signals, minimizing signal conversion overhead.

- Our evaluations demonstrate that HyAtten outperforms existing photonic Transformer accelerators, achieving a 9.8× improvement in performance-per-area and a 2.2× increase in energy-efficiency-per-area.

II. Background and Motivation

A. Transformer and self-attention

Transformer-based neural networks are generally composed of multiple identical blocks, referred to as encoder and decoder blocks. Both types of blocks contain a *multi-head self-attention* (MHA) module, a *feed-forward network* (FFN), shortcut connections, and *layer normalization* (LN). Additionally, the decoder block incorporates cross-attention and masked self-attention modules. For illustration, the structure of a basic encoder block is defined as follows:

$$\begin{cases} \mathbf{X}'_{l+1} = MHA(LN(\mathbf{X}_l)) + \mathbf{X}_l; \\ \mathbf{X}_{l+1} = FFN(LN(\mathbf{X}'_{l+1})) + \mathbf{X}'_{l+1}, \end{cases} \tag{1}$$

where $\mathbf{X}_l$ is the input sequences of l-th layer.

Multi-head Self-Attention (MHA) mechanism consists of H distinct self-attention heads. Within each head, the input vector is linearly projected into three separate vectors: the query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) vectors. The attention function is then computed between these input vectors as follows:

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = softmax(\mathbf{Q}\mathbf{K}^{\mathsf{T}}/\sqrt{d_k})\mathbf{V}, \tag{2}$$

where d_k is $\mathbf{Q}$ and $\mathbf{K}$'s dimension. An intermediate score matrix $\mathbf{S}$ is obtained with $S = Q \times K^{\mathsf{T}}$. As the input sequence grows, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ will grow linearly while the attention computation of matrix $\mathbf{S}$ will grow quadratically. The FFN module usually contains two linear layers with an activation function in between.

B. Optical computing basics

Recently, researchers propose a *dynamically-operated dot-product* (DDot) unit to perform optical dot-products between two vectors $\vec{x}$ and $\vec{y}$ [34]. As Figure 1 (b) shows, the DDot units are designed based on coherent interference. First, the *wavelength-division multiplexing* (WDM) technique encodes each input pairs (x_i, y_i) in the same wavelength λ_i. The WDM light signals are then sent through the two arms of 50 : 50 *directional coupler* (DC) with a $-90°$ *phase shifter* (PS). Consequently, the two output signals become orthogonal in the complex plane. This setup allows each input pair (x_i, y_i) with the same wavelength λ_i to interfere in parallel, while different wavelengths do not interfere. The photo-diode at the end of each output port of DC converts the incident WDM signals into photocurrent. The generated photocurrent is proportional to the accumulated intensities of the WDM signals, representing the square of the optical magnitudes and producing the final output current as $I_o \propto \vec{x} \cdot \vec{y}$.

To enable *general matrix multiplication* (GEMM) operations with optical dot-product engines, a *dynamically-operated photonic tensor core* (DPTC) has been proposed [34]. Researchers designed a compact crossbar array of DDot units to maximize operand sharing within the core, significantly reducing operand modulation costs, as illustrated in Figure 1 (a). This architecture supports efficient sharing of photonic waveguide buses among DDot units, enabling ultra-parallel GEMM operations. A $N_v \times N_h$ DPTC comprises $N_v \times N_h$ DDot units, where N_v and N_h denote the numbers of input waveguides in the vertical and horizontal directions, respectively.

C. Motivations

Problem: Signal conversion costs remain the primary bottleneck for emerging photonic systems. In the state-of-the-art photonic Transformer accelerator, Lightening-Transformer [34], signal conversion units, such as ADCs, DACs, and optical/electrical converters, consume over 50% of the chip area. To reduce this overhead, Lightening-Transformer allocates a single ADC to a 32×32 DPTC array. However, a 32×32 DPTC array generates 1024 analog signals, which cannot be efficiently processed by just one ADC. As a result, photonic devices experience significant delays, idling while awaiting signal conversion.

Observation#1: Utilizing low-resolution signal converters can significantly reduce latency and area overhead, but it may also lead to substantial model accuracy loss. According to the latest ADC performance survey [16], published in May 2023, ADC area increases exponentially with resolution. For example, a 5-bit ADC requires twice the area of a 4-bit ADC using the same technology. While a common approach to reduce area overhead is to employ low-resolution ADCs, we evaluated the impact of varying ADC resolutions (2-bit, 4-bit, and 8-bit) on the Lightening-Transformer [34]. Our evaluation

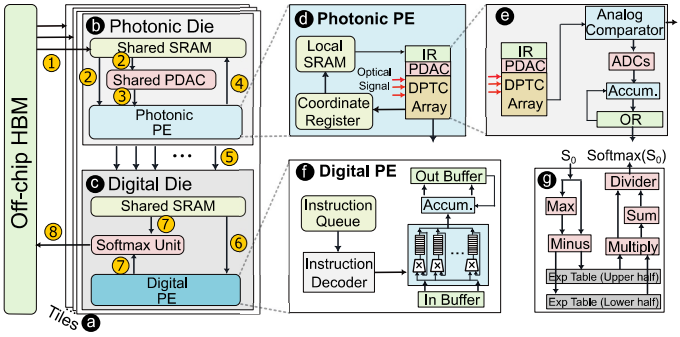


Fig. 3: Architecture and dataflow of HyAtten

utilized five datasets, CoLA, MNLI, MRPC, RTE, and SQuAD, from GLUE [25], running on a BERT-based model. For high-resolution output signals that exceed the ADC's full-scale measurement range, the conversion output was capped at the ADC's maximum value. The accuracy results (ratio between the number of correctly predicted samples by the total number of samples), shown in Figure 2 (a), reveal that using 4-bit ADCs leads to a more than 20% decrease in model accuracy compared to the original 8-bit ADCs. These experiments demonstrate that high-resolution signals are critical to preserving model accuracy and must be appropriately handled.

Observation#2: Only a small fraction of analog signals require high-resolution ADCs. Analogous to the barrel principle, ADC resolution must account for the "short end of the barrel", i.e., the high-resolution signals [34]. If we can effectively address these high-resolution signals, the ADC resolution can be lowered for the remaining signals. Figure 2 (b) illustrates the proportion of signals exceeding the resolution of various ADCs from the above experiments. While 8-bit ADCs process all signals without exceeding their full-scale measurement range, 4-bit ADCs successfully process over 85% of signals, leaving only 15% unprocessed. This indicates that the "barrel short" for 4-bit ADCs accounts for 15% of the total signals.

Our goal: Building on Observation#2, analog signals in photonic-based Transformer accelerators can be categorized into two groups: low-resolution signals ($\leq$4-bit) and high-resolution signals (>4-bit). Low-resolution signals can be efficiently processed using 4-bit ADCs with lower latency and area overhead. As noted in Observation#1, however, high-resolution signals require careful handling to avoid accuracy loss. Rather than introducing high-resolution ADCs, we employ digital circuits to process these signals. Since high-resolution signals constitute less than 15% of the total, the computational overhead imposed on the digital circuits remains minimal.

III. HyAtten

A. Architecture

As depicted in Figure 3 (ⓐ), HyAtten consists of multiple Tiles, each of which includes a photonic die (ⓑ) and a digital die (ⓒ). The photonic die performs highly parallel photonic computing, while the digital die carries out digital computations without the need for signal conversion. Together, the photonic and digital dies collaborate to execute various components of the attention mechanism, such as GEMM and softmax operations. We assume that all data related to the attention mechanism, including matrices Q, K, and V, are stored in off-chip *High-bandwidth Memory* (HBM). The data will be transferred between the HBM and HyAtten Tiles.

Details of Photonic Die. The photonic die includes a shared *static random access memory* (SRAM), a shared *photonic digital-to-analog converter* (PDAC) with modulation units, and a photonic *processing element* (PE) (ⓓ). Both the shared SRAM and PDAC are accessible by all Tiles. Input matrices Q and K from Equation (2) are transferred from the HBM to the shared SRAM. To reduce computational overhead, we apply low-bit quantization (4-bit) to these input matrices, as demonstrated in [34], where low-bit quantization significantly decreases computational demands with minimal accuracy loss. To ensure the input matrices fit within the shared SRAM, HyAtten is designed to support batch-based processing, where matrices are partitioned into smaller batches (either row or column vectors) that are processed sequentially [34]. The shared PDAC receives one batch (e.g., one column vector) of the input matrix Q, converts it to photonic signals, and broadcasts these signals to all Tiles' photonic PEs for processing.

The photonic PE handles the core computations in the attention mechanism, specifically the GEMM operations for $Q \times K^{\mathsf{T}}$ and $S \times V$. As shown in Figure 3 (ⓓ), the photonic PE is equipped with local SRAM, a coordinate register, and a DPTC unit. The local SRAM receives data (e.g., matrix K) from the shared SRAM and transfers it to the DPTC units for processing. Unlike the shared SRAM, data in the local SRAM is routed to a local PDAC for signal conversion. The DPTC unit processes two sets of photonic inputs: one from the shared PDAC and another from the local PDAC, as illustrated in Figure 1 (a). The coordinate register stores the coordinates of some matrix elements, which are used to facilitate data loading from both the shared and local SRAM.

Figure 3 (ⓔ) illustrates the detailed architecture of a DPTC unit. Unlike prior photonic accelerators that configure one high-resolution ADC for each DPTC array [34], we configure each 64×64 DPTC array with 32 low-resolution ADCs to mitigate signal conversion latency. The DPTC unit operates as follows: two input photonic signals are received from the shared PDAC and local PDAC. These signals are processed through the DPTC array, resulting in photonic currents corresponding to the GEMM operations. Given the use of low-resolution ADCs, some photonic currents may exceed the ADCs' resolution. To handle this, an analog comparator is integrated to detect over-resolution signals and log their coordinates in the coordinate register. Under-resolution signals are processed by the ADCs to obtain their digital results. The memory controller then loads the over-resolution data based on the coordinates in the register and sends these data to the digital die for further processing.

Details of Digital Die. The digital die consists of a shared SRAM, a softmax unit, and a digital PE. The shared SRAM receives two types of data from the photonic die: the GEMM output results of low-resolution signals and the input digital

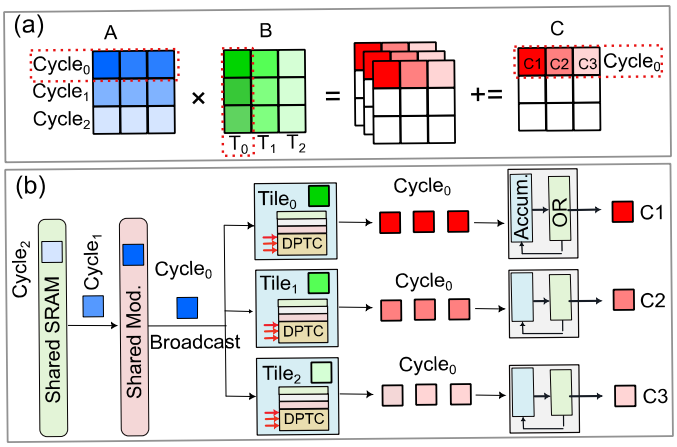


Fig. 4: GEMM operations on multiple photonic Tiles

TABLE I: Hardware Configurations of HyAtten

Component	Area (mm^2)	Power (mW)	Params.	Spec.
Components Shared by all Tiles				
Shared PDAC	0.0016	8	Resolution	4 Bits
			Numbers	1
Shared SRAM	3.68	1.23K	Capacity	2MB
Photonic Die (PD) Properties				
PDAC [21]	0.0748	520	Resolution	4 Bits
			Numbers	64
ADC [14]	0.0057	29.6	Resolution	4 Bits
			Numbers	32
DPTC Array	0.246	624	Size	64 × 64
			Numbers	1
SRAMs [18]	0.06	19	Capacity	32KB
Registers	0.015	5.23	Capacity	8KB
Accumulator	0.0014	0.039	Numbers	32
Ana. Comp.	0.00031	0.019	Numbers	32
PD Total	0.405	1.2K	Numbers	1
Digital Die (DD) Properties				
MAU [3]	0.014	8.2	Numbers	1
Registers	0.002	0.63	Capacity	1KB
Softmax [10]	0.0072	1.134	LUT Size	512B
			Numbers	1
DD Total	0.023	9.96	Numbers	1
HyAtten properties (32 Tiles in total)				
HyAtten	17.38	39.9W	Numbers	1

values of high-resolution signals. It also buffers the computed results before transferring them back to the HBM. The softmax unit handles the softmax operation for the attention score matrix S, which is collaboratively generated by the photonic and digital dies. The digital PE is responsible for completing GEMM operations that cannot be processed by the photonic die. The area and latency overhead of the digital die is minimal, as it processes only a small subset of signals.

Figure 3 (f) illustrates the architecture of the digital PE, which includes an instruction queue, an instruction decoder, an input buffer, a *multiplication-accumulation unit* (MAU), and an output buffer. The instruction queue and decoder work together to manage and schedule the digital PE's execution. Data is received from the shared SRAM by the input buffer, which then forwards it to the MAU. The MAU carries out vector-vector multiplications and accumulations to complete the GEMM operations. The results are subsequently stored in the output buffer for further processing or storage.

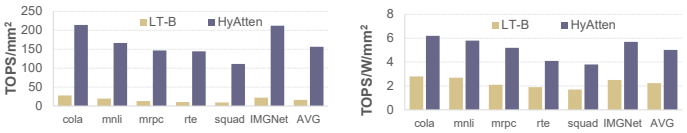
Figure 3 (g) presents the architecture of the softmax unit. Following the approach in [10], we implement the exponent function using a lookup table. To minimize the size of the lookup table, we leverage the property that an exponentiation can be decomposed into the product of two smaller exponentiations. Thus, we employ two smaller lookup tables, an upper half and a lower half, and a multiplier to achieve the desired result. After computing the exponent of the dot-product, the value is accumulated and later used as the denominator in the softmax computation.

B. Dataflow

Details of Transmission. Figure 3 shows the data transmission process within HyAtten, using the $Q \times K^{\mathsf{T}}$ operation as an example, with a similar procedure followed for $S \times V$. In ①, the input matrices Q and K are loaded from the off-chip HBM into the shared SRAM on HyAtten's photonic die. In ②, matrix K is transferred to the local SRAM of the photonic PE, while matrix Q is sent to the shared PDAC for conversion into photonic signals. In ③, the photonic signals corresponding to matrix Q are broadcast to all Tiles' photonic PEs. Simultaneously, matrix K is converted into photonic signals, and both matrices undergo GEMM operations to produce the result matrix S. In ④, the result matrix S, along with the digital values of the high-resolution signals, are stored in the shared SRAM. The digital values of the high-resolution signals are transferred to the digital die in ⑤. In ⑥, the digital PE then processes these high-resolution signals, performing the remaining GEMM operations. In ⑦, the matrix S is sent to the softmax unit. The results will be stored back to the HBM in ⑧.

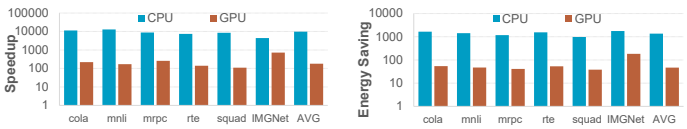
Matrices partition and data mapping. To address the size limitations of the DPTC array, the input matrices must be divided into smaller sub-matrices that align with the dimensions of the array. We adopt a Tile-based matrix partitioning and data mapping strategy, as illustrated in Figure 4 (a). The input matrices, A and B, are partitioned into shards, with each shard matching the size of the DPTC array. Matrix A is stored in the shared SRAM and accessed sequentially in a column-wise manner, while matrix B is stored in the local SRAM of the photonic PE, with different shards distributed across different Tiles. For instance, as shown in Figure 4 (a), the first sub-matrix of matrix B is stored in Tile_0. Given the limited capacity of the local SRAM, we process subsets of input matrices at a time. For example, if the local SRAM can accommodate two sub-matrices, GEMM operations will be executed in batches of two sub-matrices at a time.

Details of Calculation. Figure 4 (b) illustrates the Tile-based GEMM operations. We assume that matrix A has been partitioned and stored in the shared SRAM, while matrix B has also been partitioned and stored in the local SRAM across different Tiles. During cycle_0, the first sub-matrix of matrix A is sent to the shared PDAC, where it is converted into photonic signals and broadcast to all Tiles' photonic PEs. Simultaneously, the corresponding sub-matrix of matrix B is converted to photonic



(a) (b)

Fig. 5: (a) Performance per unit area, and (b) energy efficiency per unit area



(a) (b)

Fig. 6: (a) HyAtten speedups compared to CPU and GPU, and (b) HyAtten energy saving compared to CPU and GPU

signals by the local PDAC. The resulting photonic currents produced by the DPTC arrays are processed by the ADCs to generate the outputs of the sub-matrix multiplication. In the following two cycles, cycle_1 and cycle_2, the second and third sub-matrices of matrix A are sequentially sent to the shared PDAC, where they are converted into photonic signals and broadcast to all Tiles. These signals multiply with the same sub-matrix of matrix B as in cycle_0. The results from cycle_0 are accumulated to generate the first row vector of the output matrix, and this process continues for subsequent sub-matrix multiplications, following the same procedure.

IV. Experimental Evaluation

A. Experimental Setup

System Setup. We modified an existing Python-based simulator [34] to evaluate the latency, power, area, and energy efficiency of HyAtten during Transformer inference. The area, leakage power, and access energy of the memory system are modeled using PCACTI [18] in 14 nm. We model HBM with a bandwidth 1TB/s to supply data to the photonic system. The area and energy consumption of the digital die, including the softmax unit, MAU, and accumulator, are derived from SPICE circuit simulations [3]. Similar to [7], [8], we scale the power of the ADC [14] and PDAC [21] based on the bit-width and frequency requirements of the photonic PE. To further optimize, we replace one 8-bit ADC with 16 4-bit ADCs, as high-resolution ADCs can be constructed from multiple low-resolution units [16]. Unlike conventional accelerators that share a single ADC across multiple arrays, HyAtten equips each array with 32 ADCs to minimize signal conversion latency. Table I provides a detailed breakdown of the device parameters used.

Models, datasets, training, and inference settings. We evaluate the efficiency and accuracy of HyAtten using two widely recognized Transformer models: DeiT-T for vision tasks [24] and BERT-base for NLP tasks [4]. The models are tested on: ImageNet [17] for vision and GLUE [25] for NLP. Both weight and activation quantization are applied using low-bit precision [5]. Additionally, noise-aware training is employed, with both encoding and systematic noise injected during training to reflect real-world conditions [34].

B. Compare to State-of-the-art Photonic Accelerator

Baseline: We select the base version of the Lightening-Transformer [34], which features a 4-bit DPTC core, as the baseline system, denoted as LT-B. LT-B is a photonic-based Transformer accelerator that relies on high-resolution signal converters (e.g., 4-bit DAC and 8-bit ADC) to process all analog signals. To mitigate the area overhead associated with signal converters, LT-B configures each DPTC array with a single ADC. For a fair comparison, we use identical photonic device parameters for both LT-B and HyAtten. To eliminate the influence of chip area in our evaluation, we report performance and energy consumption normalized to per unit area.

TABLE II: Accuracy Comparison

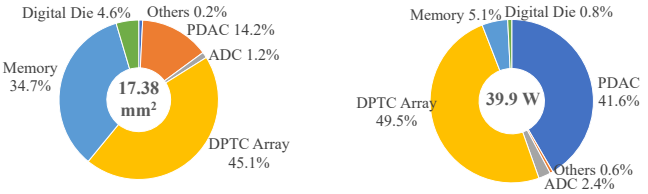
Model	BERT		DeiT
Datasets	GLUE (MRPC)	SQuAD	ImageNet-1K
Original	86.11	81.44	71.43
LT-B	85.77	81.16	71.19
HyAtten	85.89	81.22	71.27

Performance per unit area: Figure 5 (a) dispalys the speedups achieved by HyAtten relative to the photonic baseline. Across all six datasets, HyAtten delivers a 9.8× speedups per unit area. This improvement can be attributed to two key factors. First, HyAtten replaces each high-resolution ADC with multiple low-resolution ADCs, significantly reducing signal conversion latency without increasing chip area. Second, HyAtten employs a digital die to handle the 15% of signals that require high resolution ADCs, thereby avoiding excessive signal conversion overhead while incurring only a small area penalty. In contrast, the baseline LT-B system relies on high-resolution ADCs for all signals, which inefficiently allocates resources to process the 85% of signals that are low-resolution.

Energy efficiency per unit area: Figure 5 (b) illustrates the energy savings achieved by HyAtten compared to the photonic baseline. When processing all datasets using the DeiT-T and BERT models, HyAtten demonstrates a 2.2× energy reduction (normalized to the same chip area). These energy savings primarily stem from HyAtten's ability to eliminate the significant energy overhead associated with high-resolution signal converters. While the baseline LT-B system uses high-resolution ADCs to convert all signals, leading to substantial energy consumption, HyAtten replaces these high-resolution ADCs with low-resolution ADCs and digital dies, which perform GEMM operations with considerably lower energy demands.

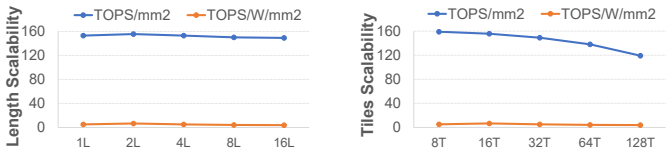
C. Compare to State-of-the-art Digital Accelerators

In Figure 6, we compare HyAtten against, a single Nvidia A100 GPU and an Intel Core i7-9750H CPU, to highlight its significant performance and energy efficiency improvements. Figure 6 (a) demonstrates that HyAtten delivers the highest performance, surpassing both CPU and GPU platforms. It



(a) (b)

Fig. 7: (a) Area breakdown, and (b) Power breakdown



(a) (b)

Fig. 8: (a) Sequence length scalability, and (b) Tiles scalability

achieves over 100× speedup per unit area relative to the A100 GPU, largely due to the high processing speed enabled by photonic computing. Figure 6 (b) illustrates that HyAtten also exhibits superior energy efficiency, achieving over 50× greater efficiency per unit area compared to the GPU. This improvement is primarily attributed to the PDAC multiplexing within the DPTC array.

D. Accuracy Comparison

Table II reports the accuracy comparison between GPU, LT-B, and HyAtten, all running the same model, datasets, and bit-widths. HyAtten maintains an accuracy loss of less than 0.3% compared to Transformers with the same bit-widths running on the GPU. Additionally, HyAtten demonstrates a 0.2% accuracy improvement over LT-B. This accuracy improvement is primarily due to the integration of the digital die. As discussed in Section II-C, high-resolution signals play a crucial role in maintaining model accuracy. The noise introduced by high-resolution ADCs in LT-B negatively affects these signals, leading to accuracy degradation. In contrast, HyAtten processes high-resolution signals using the noise-free digital die, thereby preserving model accuracy.

E. System Efficiency Analysis

Area Breakdown. Figure 7 (a) presents the area breakdown of HyAtten, which occupies a total area of 17.38mm^2. The DPTC array (including the *Mach-Zehnder modulator* (MZM), phase shifter, and photonic detector) accounts for the largest share at 45.1%, followed by the memory system at 34.7%, and the PDAC at 14.2%. The remaining components, such as the ADC and digital die, contribute less than 10% of the total area. Notably, HyAtten is designed with multiple ADCs for each DPTC array without increasing the overall area overhead. Additionally, the inclusion of a digital die to process around 15% of the data introduces only a minimal increase in area.

Power Breakdown. Figure 7 (b) presents the power distribution for HyAtten, which consumes a total of 39.9W. The DPTC array and PDAC dominate the power usage, contributing about 49.5% and 41.6%, respectively. The remaining components, such as the memory system, ADC, and digital die, collectively account for less than 10% of the total power consumption. Importantly, despite configuring additional ADCs for each DPTC array, the power overhead remains minimal. Additionally, the digital die, which processes around 15% of the data, introduces only a negligible increase in overall power consumption.

Scalability Discuss. Figure 8 (a) presents the performance of HyAtten across various sequence lengths. As sequence length increases, HyAtten will process more submatrices, but the overhead associated with handling individual matrices remains relatively stable. Figure 8 (b) illustrates the impact of configuring different numbers of Tiles in HyAtten. While each Tile maintains a high internal throughput, increasing the number of Tiles leads to a rise in data transfer between HyAtten and the HBM, resulting in a reduction in overall throughput.

V. Related Work

Current Accelerators for Transformer. *Field Programmable Gate Arrays* (FPGA) and *Application-specific integrated circuit* (ASIC) architectures are commonly used to accelerate sparse matrix-vector multiplication, such as graph processing [2], [26]–[30]. Recently, researchers aim at accelerating attention mechanism with FPGA and ASIC architectures [1], [15], [32]. Many memory-centric architectures are also proposed to accelerate attention mechanism, such as *processing in memory* (PIM) [9]–[13], [31], [33]. While these digital accelerators effectively reduce inference latency, traditional electrical computing platforms face significant limitations as transistor-based chips approach the boundaries of Moore's Law.

Emerging Photonic Architectures. Integrated photonic accelerators present a promising alternative for accelerating deep neural networks, offering ultra-high speeds, extensive parallelism, and low energy consumption. Various optical systems have been explored to accelerate *convolutional neural networks* (CNNs) [6], [19], [20], [22], [23] and Transformers [34].

VI. Conclusion

This paper presents HyAtten, a novel attention mechanism accelerator that leverages hybrid photonic and digital computing. HyAtten incorporates a photonic die with low-resolution ADCs to efficiently process low-resolution signals, while a digital die handles high-resolution signals without the overhead of signal conversion. Experimental results demonstrate that HyAtten achieves superior performance and energy efficiency with minimal accuracy loss.

VII. Acknowledgments

This research is partially supported by the National Research Foundation, Singapore under its Competitive Research Program Award NRF-CRP23-2019-0003.

REFERENCES

[1] Z. Bai, P. Dangi, H. Li, and T. Mitra, "SWAT: Scalable and Efficient Window Attention-based Transformers Acceleration on FPGAs," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, ser. DAC '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3649329.3658488

[2] D. Chen, H. He, H. Jin, L. Zheng, Y. Huang, X. Shen, and X. Liao, "Metanmp: Leveraging cartesian-like product to accelerate hgnns with near-memory processing," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3579371.3589091

[3] F. Corti, A. Reatti, E. Cardeli, A. Faba, and H. Rimal, "Improved Spice Simulation of Dynamic Core Losses for Ferrites With Nonuniform Field and Its Experimental Validation," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, pp. 12 069–12 078, 2021.

[4] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," *arXiv preprint arXiv:1902.08153*, 2019.

[6] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja *et al.*, "Parallel convolutional processing using an integrated photonic tensor core," *Nature*, vol. 589, no. 7840, pp. 52–58, 2021.

[7] Y. Huang, L. Zheng, H. Liu, Z. Zhou, D. Chen, P. Yao, Q. Wang, X. Liao, and H. Jin, "MeG^2: In-Memory Acceleration for Genome Graphs Analysis," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*, 2023, pp. 1–6.

[8] Y. Huang, L. Zheng, P. Yao, Q. Wang, X. Liao, H. Jin, and J. Xue, "Accelerating Graph Convolutional Networks Using Crossbar-based Processing-In-Memory Architectures," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2022, pp. 1029–1042.

[9] H. Li, D. Chen, and T. Mitra, "SADIMM: Accelerating Sparse Attention using DIMM-based Near-memory Processing," *IEEE Transactions on Computers*, no. 01, pp. 1–12, 2024.

[10] H. Li, H. Jin, L. Zheng, X. Liao, Y. Huang, C. Liu, J. Xu, Z. Duan, D. Chen, and C. Gui, "CPSAA: Accelerating Sparse Attention Using Crossbar-Based Processing-In-Memory Architecture," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 6, pp. 1741–1754, 2024.

[11] H. Li, Z. Li, Z. Bai, and T. Mitra, "ASADI: Accelerating Sparse Attention Using Diagonal-based In-Situ Computing," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024, pp. 774–787.

[12] C. Liu, H. Liu, H. Jin, X. Liao, Y. Zhang, Z. Duan, J. Xu, and H. Li, "ReGNN: a ReRAM-based heterogeneous architecture for general graph neural networks," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, ser. DAC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 469–474. [Online]. Available: https://doi.org/10.1145/3489517.3530479

[13] C. Liu, K. Wu, H. Liu, H. Jin, X. Liao, Z. Duan, J. Xu, H. Li, Y. Zhang, and J. Yang, "A ReRAM-Based Processing-In-Memory Architecture for Hyperdimensional Computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2024.

[14] J. Liu, M. Hassanpourghadi, and M. S.-W. Chen, "A 10GS/s 8b 25fJ/c-s 2850um2 Two-Step Time-Domain ADC Using Delay-Tracking Pipelined-SAR TDC with 500fs Time Step in 14nm CMOS Technology," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65, 2022, pp. 160–162.

[15] L. Lu, Y. Jin, H. Bi, Z. Luo, P. Li, T. Wang, and Y. Liang, "Sanger: A Co-Design Framework for Enabling Sparse Attention using Reconfigurable Architecture," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 977–991. [Online]. Available: https://doi.org/10.1145/3466752.3480125

[16] B. Murmann, "ADC Performance Survey 1997-2024," [Online]. Available: https://github.com/bmurmann/ADC-survey.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[18] A. Shafaei, Y. Wang, X. Lin, and M. Pedram, "FinCACTI: Architectural Analysis and Modeling of Caches with Deeply-Scaled FinFET Devices," in *2014 IEEE Computer Society Annual Symposium on VLSI*, 2014, pp. 290–295.

[19] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, "Photonics for artificial intelligence and neuromorphic computing," *Nature Photonics*, vol. 15, no. 2, pp. 102–114, 2021.

[20] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature photonics*, vol. 11, no. 7, pp. 441–446, 2017.

[21] T. Sridarshini, S. I. Gandhi, and V. J. U. Firthouse, "Compact 4-bit all optical digital to analog converter based on photonic crystal ring resonators," *Laser Physics*, vol. 30, no. 11, p. 116206, 2020.

[22] F. Sunny, A. Mirza, M. Nikdast, and S. Pasricha, "CrossLight: A cross-layer optimized silicon photonic neural network accelerator," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1069–1074.

[23] A. N. Tait, T. F. De Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Neuromorphic photonic networks using silicon photonic weight banks," *Scientific reports*, vol. 7, no. 1, p. 7430, 2017.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 347–10 357. [Online]. Available: https://proceedings.mlr.press/v139/touvron21a.html

[25] A. Wang, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[26] Q. Wang, L. Zheng, Z. An, H. Huang, H. Zhu, Y. Huang, P. Yao, X. Liao, and H. Jin, "High-Performance and Resource-Efficient Dynamic Memory Management in High-Level Synthesis," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, ser. DAC '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3649329.3655945

[27] Q. Wang, L. Zheng, A. Hu, Y. Huang, P. Yao, C. Gui, X. Liao, H. Jin, and J. Xue, "A Data-Centric Accelerator for High-Performance Hypergraph Processing," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2022, pp. 1326–1341.

[28] Q. Wang, L. Zheng, Y. Huang, P. Yao, C. Gui, X. Liao, H. Jin, W. Jiang, and F. Mao, "GraSU: A Fast Graph Update Library for FPGA-based Dynamic Graph Processing," in *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 149–159. [Online]. Available: https://doi.org/10.1145/3431920.3439288

[29] Q. Wang, L. Zheng, J. Yuan, Y. Huang, P. Yao, C. Gui, A. Hu, X. Liao, and H. Jin, "Hardware-Accelerated Hypergraph Processing with Chain-Driven Scheduling," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2022, pp. 184–198.

[30] Q. Wang, L. Zheng, J. Zhao, X. Liao, H. Jin, and J. Xue, "A Conflict-free Scheduler for High-performance Graph Processing on Multi-pipeline FPGAs," *ACM Trans. Archit. Code Optim.*, vol. 17, no. 2, May 2020. [Online]. Available: https://doi.org/10.1145/3390523

[31] J. Xu, H. Liu, Z. Duan, X. Liao, H. Jin, X. Yang, H. Li, C. Liu, F. Mao, and Y. Zhang, "ReHarvest: An ADC Resource-Harvesting Crossbar Architecture for ReRAM-Based DNN Accelerators," *ACM Trans. Archit. Code Optim.*, vol. 21, no. 3, Sep. 2024. [Online]. Available: https://doi.org/10.1145/3659208

[32] H. You, Z. Sun, H. Shi, Z. Yu, Y. Zhao, Y. Zhang, C. Li, B. Li, and Y. Lin, "ViTCoD: Vision Transformer Acceleration via Dedicated Algorithm and Accelerator Co-Design," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2023, pp. 273–286.

[33] M. Zhou, W. Xu, J. Kang, and T. Rosing, "TransPIM: A Memory-based Acceleration via Software-Hardware Co-Design for Transformer," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2022, pp. 1071–1085.

[34] H. Zhu, J. Gu, H. Wang, Z. Jiang, Z. Zhang, R. Tang, C. Feng, S. Han, R. T. Chen, and D. Z. Pan, "Lightening-Transformer: A Dynamically-Operated Optically-Interconnected Photonic Transformer Accelerator," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024, pp. 686–703.