BITLUME: Precision-Flexible Photonic Computing for Ultra-Fast and Energy-Efficient DNN Acceleration

Chengpeng Xia[1*], Haibo Zhang[1, 2], Hao Zhang[1], Yawen Chen[3], Amanda Susan Barnard[2]
[1]University of Otago [2]Australian National University [3]University of New South Wales
[*]haibo.zhang@otago.ac.nz

***Abstract*—As deep learning expands across emerging domains, computational demands are pushing traditional electronic accelerators to their limits. Silicon photonics has emerged as a promising technology for accelerating deep learning workloads, but precision remains a challenge due to noise and non-idealities. In this paper, we present BITLUME, a novel photonic computing unit that enables multiplications beyond 8-bit precision through a precision-flexible scheme. We further propose an optimized round-truncation algorithm and data mapping strategy for BITLUME to reduce optoelectronic conversions, enhance data reuse, and maintain computational accuracy. A hybrid optoelectronic architecture integrating BITLUME is developed and validated using a prototype built with FPGA, RF, and photonic components, achieving 3.7× lower end-to-end latency than the A100 GPU in dot product. Simulations of training seven DNN models at FP32 show that BITLUME achieves up to 3.35× and 10.78× speedup, and 1.53× and 4.12× energy savings, compared to the state-of-the-art photonic accelerator and A100 GPU, respectively.**

***Index Terms*—Photonic computing, DNN accelerator**

I. Introduction

The scale of deep neural networks (DNNs) has grown exponentially, with models like Switch Transformer and GPT-4 reaching 1.6 trillion and 1.7 trillion parameters [1]. This growth in model complexity has imposed significant computational and energy demands, exemplified by GPT-3's training requiring 190,000 kWh over seven months with 512 A100 GPUs [2], [3]. While GPUs and TPUs have accelerated deep learning, electronic accelerators face fundamental limitations due to the end of Dennard scaling.

Photonic computing, which performs computations in the optical domain, demonstrates promising advantages in computational speed, bandwidth, and energy efficiency. Recent implementations have achieved significant improvements: Lightmatter's accelerator demonstrated 10× speedup with 90% energy reduction versus GPUs [4], while MIT's Lightning platform achieved 6.6× lower inference latency and 15.69× better energy efficiency per MAC [5]. Emerging materials further promise ultra-high-speed optical computing over 100 GHz [6], far beyond current GPU frequencies of ∼1.41 GHz.

However, existing research on photonic computing for DNNs primarily focuses on inference acceleration and typically achieves low precision ranging from 6 to 16 bits [7]–[9]. Achieving high-precision computation in photonic architectures faces challenges such as cascading crosstalk noise, limited photodetector sensitivity, and intrinsic detector noise. Even the state-of-the-art photonic chips fabricated with modern processes and published in Nature only report effective precision of 7.61 bits [10] and 8.3 bits [11], respectively. In accelerator architectures for DNN, precision is a critical factor, especially during training, where higher precision contributes to improved model accuracy. When gradients are quantized to low precision, the training process becomes unstable due to the deviation introduced in the optimization direction. Recent work in [12] reported that once such deviation accumulates beyond a tolerable threshold, the training may crash, leading to severe performance degradation.

To address these challenges, we present BITLUME, a precision-flexible photonic DNN accelerator that can achieve high-precision photonic multiplication operations. The **key idea** of BITLUME is to decompose a high-precision multiplication into multiple low-precision multiplication operations that can be performed in the photonic domain without losing computing accuracy. We further introduce a rounding-truncation scheme to enhance computational efficiency while minimizing precision loss. A hybrid optoelectronic system integrating BITLUME with an FPGA is developed to accelerate DNNs. This paper's main contributions are as follows:

- To address the precision limitations of photonic systems, we propose a precision-flexible lossless scheme and an optimized round-truncation algorithm that enable photonic systems to perform multiplication operations exceeding 32-bit precision. Based on this algorithm, we design BITLUME, a photonic computing unit with a data mapping strategy that maximizes data reuse while achieving precision-flexible multiplication.
- By integrating multiple BITLUME units, we develop a photonic computing core capable of parallel, high-precision multiplication with flexible switching between FP16 and FP32. Our energy analytical model shows that BITLUME is up to 4.36× more energy-efficient than the latest electronic processors at FP32.
- We build a 4.096 GHz hybrid optoelectronic platform combining UltraScale+ RFSoC FPGA, Radio Frequency (RF), and photonic components for high-precision computation. To the best of our knowledge, this represents the first hardware prototype for high-precision and high-frequency photonic computing, achieving 3.7× lower end-to-end latency than the A100 GPU in dot product. Our performance evaluations with seven DNN models show that compared to the Mirage and A100 GPU, in FP32, BITLUME accelerates the training time by 3.35× and 10.78× while consuming 1.53× and 4.12× less energy, respectively.

II. Background and Related Works

A. Photonic Computing and Its Limitations in Precision

Analog multiplication in the photonic domain can be achieved using cascaded Mach-Zehnder Modulators (MZMs). As illustrated in Fig. 1, assuming the initial light intensity is set to 1, applying a voltage of $a = 0.6$ to the first MZM modulates the light to an intensity of 0.6. This output is then fed into the second MZM, where a voltage $b = 0.7$ is applied, further reducing the intensity to $0.6 \times 0.7 = 0.42$. The final output is detected by a photodetector, which converts the optical signal into a voltage proportional to the multiplication result. The result precision of photonic computing depends on the number of distinct intensity levels that can be accurately distinguished in the analog domain [5]. To achieve n-bit precision, there must be at least 2^n distinguishable intensity levels. Due to the limitations in the current manufacturing technologies for photodetectors and lasers, most existing photonic computing systems can only achieve an 8-bit precision (i.e., 256 distinguishable intensity levels) [13]. However, even with 8-bit precision, the multiplication

of 8-bit numbers still has the risk of overflow. To ensure 8-bit output precision, the largest operand for multiplication without overflow is 4-bit since $[1111_{(15)}]^2 = 225 < 256$.

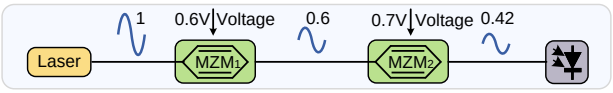

Fig. 1. MZMs-based optical multiplication.

B. Related Works

The majority of existing studies on photonic neural networks exhibit significantly low resolutions. For example, the most advanced photonic chips, fabricated with modern processes and developed by Lightelligence and Lightmatter, report effective precisions of only 7.61 bits [10] and 9.8 bits [11], respectively. Mirage [14] achieves high-precision photonic computation based on Residue Number System (RNS) decomposition. The precision attainable with RNS depends on the chosen moduli set. Under the constraint of a 4-bit photonic input width, the moduli set RNS(3, 5, 7, 11, 13) yields the maximum representable range, M = 15015, corresponding to a precision of only 13.87 bits. However, the complex remainder and restoration processes increase the workload in the electronic domain, creating a bottleneck on both speed and energy efficiency. Authors in [7] proposed 16-bit photonic matrix-vector multipliers by using bit-wise microdisk-based full adders. However, the use of bit-wise full adders limits computational efficiency, and parallelism is constrained by optical path complexity. The principal challenges in achieving high resolution in photonic architectures stem from crosstalk noise, photodetector sensitivity, and photodetector noise (shot noise). Currently, there is no effective method to perform DNN training without compromising accuracy by reducing computational resolution. Therefore, developing a photonics-based DNN training accelerator with superior precision remains a key challenge.

III. BITLUME: Bit-Precision Photonic Computing Architecture

A. Precision-Flexible Lossless Multiplication via Decomposition

Floating point multiplication is widely used in DNN training to ensure high accuracy. The IEEE floating point format consists of a sign bit s, exponent bits e, and fraction bits m . Multiplying two floating-point numbers A and B involves XORing the sign bits, adding exponents, and multiplying fractions. As the computational complexity of floating-point multiplication is largely dominated by the multiplication of the fraction, we explore photonic computing with flexible precision to accelerate fraction multiplication $m_A \times m_B$.

The **key idea** of our approach is to decompose a high-bit multiplication into multiple 4-bit multiplications that can be executed in the photonic domain without loss of accuracy. To maximize the optical computation efficiency within the 8-bit signal constraint, we set 4 bits as the decomposition unit to avoid multiplication overflow, as discussed in Section II-A. As illustrated in Figure 2(a), given two 11-bit binary mantissas for two FP16 numbers, each is divided into 3 nibbles (4-bit units): $m_A = [a_3, a_2, a_1]$ and $m_B = [b_3, b_2, b_1]$. The multiplication m_A and m_B is then decomposed into the multiplications of a_i and b_j. For each partial product $a_i \times b_j$, it needs to be left-shifted by $4(i + j)$ bits to ensure that the partial product is correctly aligned according to the positions of a_i and b_j in the original operands. The result of $m_A \times m_B$ is obtained by summing all the partial products.

Algorithm 1: Determine Truncated Bit-Widths(M, N)

$W \leftarrow 4 \times \lceil (\max(M+1, N+1) + b/4)/4 \rceil$
// Target total width W.
if $M = N$ **then**
  $w_a = w_b \leftarrow 4 \times \lceil W/8 \rceil$;
end
else if $M > N$; **then**
  $w_b \leftarrow 4 \times \lfloor \frac{N+1}{4} \rfloor$; $w_a \leftarrow W - w_b$;
end
else
  $w_a \leftarrow 4 \times \lfloor \frac{M+1}{4} \rfloor$; $w_b \leftarrow W - w_a$;
end
return (w_a, w_b)

Despite the availability of faster multiplication algorithms such as the Karatsuba algorithm [15] and the Schönhage–Strassen algorithm [16], we adopt the classical decomposition method because it is structurally simpler and better suited for photonic systems: (1) the partial products can be processed in parallel with high data reuse; (2) the electronic logic is lightweight and pipeline-friendly; and (3) the minimal reliance on adders/shifters avoids electronic bottlenecks. In contrast, the Karatsuba algorithm exhibits strong data dependencies due to its divide-and-conquer structure, which limits both parallelism and pipelining opportunities. Its recursive nature also introduces significant hardware overhead and increases the risk of intermediate overflow. The Schönhage–Strassen algorithm, which leverages the Fast Fourier Transform (FFT) for multiplication, is efficient only for large bit widths (typically greater than 100 bits) due to the overhead caused by FFT and inverse FFT operations [17]. We will quantitatively compare the performance of our solution with the Karatsuba and Schönhage–Strassen algorithms in Section III-D.

B. Optimized Decomposition With Round Truncation

In IEEE floating-point multiplication, the product of the mantissas can exceed the bit width of the multiplier, with the final result rounded to fit the target precision. To improve the photonic computing efficiency, we propose an approximate computing scheme based on pre-computation rounding by truncating low-significance bits to reduce computational complexity while preserving numerical accuracy.

Let $M + 1$ and $N + 1$ be the number of bits in two fractions m_A and m_B, respectively, where the extra bit accounts for the hidden bit in IEEE floating-point representation. Note that M and N may not be equal to accommodate varying bit-width requirements (e.g., multiplying an FP16 number with an FP32 number). According to the IEEE 754 standard, the product of two fractions is ultimately rounded to a bit-width of $\max(M+1, N+1)$. Therefore, we first ensure that the output bit-width after truncation exceeds $\max(M+1, N+1)$. To further minimize the rounding error in the final result, we introduce a variable set of guard bits defined as $\text{Guard Bits} = b/4$, where $b \in \{16, 32, 64, 128\}$ denotes the selected floating-point precision (e.g., FP16, FP32, etc.). Let w_a and w_b be the bit-widths after truncation for m_A and m_B, respectively. Since multiplying w_a- and w_b-bit numbers yields at most $w_a + w_b$ bits, we impose the constraint:

$$w_a + w_b \geq \max(M + 1, N + 1) + b/4.$$

Furthermore, to maximize hardware implementation efficiency, both w_a and w_b are set to be multiples of 4 bits:

$$W = 4 \times \lceil (\max(M + 1, N + 1) + b/4)/4 \rceil .$$

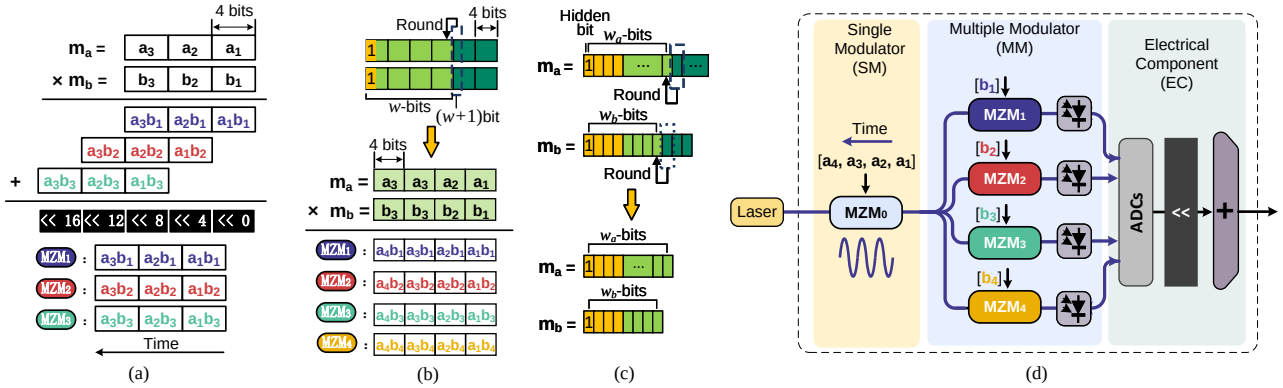

Fig. 2. (a) Mapping of FP16 mantissa multiplication in BITLUME unit, (b) An example of round truncation in fixed bit-width multiplications, (c) Round truncation in variable bit-width multiplications, (d) BITLUME computing unit ($k = 4$).

As shown in lines 2-10 of Algorithm 1, we select a pair (w_a, w_b) such that $w_a + w_b = W$. When $M = N$, i.e., the bit-widths of m_A and m_B are equal, both operands can be truncated to a bit-width of $4 \times \lceil W/(2 \times 4) \rceil$ (rounded to the nearest multiple of 4). As shown in Figure 2 (c), when performing unequal bit-widths $M \neq N$, we first truncate the operand with the smaller bit-width to a multiple of 4, and then adjust the higher-bitwidth operand accordingly. According to the IEEE 754 standard, the fraction bit length of a higher-precision floating-point number is always more than twice that of a lower-precision one. Therefore, even after truncation, the higher-bitwidth operand retains more bits than the lower one.

To evaluate the effectiveness of the round truncation scheme, we generate 1024 groups of normally distributed random floating-point numbers and perform multiplication operations. The relative error under different bit-widths is computed using the following formula:

$$\text{Relative Error} = \frac{\sqrt{\sum (R - \hat{R})^2}}{\sqrt{\sum R^2}},$$

where R denotes the original multiplication result, and $\hat{R}$ represents the result after applying the round truncation algorithm. Figure 3 illustrates the relative errors between the rounded truncation scheme and the ground truth across different floating-point precisions. The highest relative error occurs under FP16, reaching approximately 0.09%. As the bit-width increases, the error progressively decreases, reaching as low as 1.26×10^{-17} under FP128. This demonstrates that our method effectively preserves high computational accuracy while enhancing the computational efficiency of the photonic accelerator.

C. Precision-Flexible Photonic Computing Unit

Based on decomposition-based precision-flexible multiplication and round truncation, we design a photonic computing unit named BITLUME to support precision-flexible multiplication. As illustrated in Figure 2 (d), BITLUME contains three modules. The SM module contains a single modulator MZM_0 used to encode data a_i. The MM module contains k modulators MZM_1-MZM_k to encode data b_j,

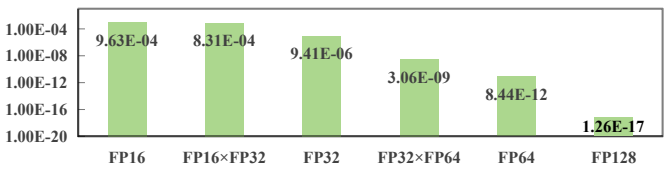

Fig. 3. Relative error across different floating-point precisions.

where $k = 4$ for the example given in Figure 2. MZM_0 is connected to MZM_1-MZM_k with a $1 \times k$ fiber optic splitter to split the output from MZM_0 evenly to MZM_1-MZM_k. Each modulator in the MM module is further connected to a photodetector for optical-to-electrical conversion. The EC module includes ADCs, shifters, and an adder tree to sum partial products into the final result.

We further design a mapping strategy to perform the multiplications of nibbles on BITLUME. The pseudocode of the mapping strategy is given in Algorithm 2, where m and n are the bit-widths of numbers A and B, respectively. Figure 2 illustrates the mapping of the mantissa multiplications for FP16 and FP32. The **key design principle** of our mapping strategy is to maximize data reuse, allowing data to be reused in multiple nibble multiplications after digital-to-analog conversion. This is essential because optoelectronic conversion is a major source of energy consumption in photonic systems. Moreover, in voltage-based MZM modulation, different voltages need to be applied when different data passes through the modulator. However, when the same data enters the MZM in different time steps, no further voltage adjustment is required. Furthermore, when the multipliers are identical, data only needs to be requested from memory once, and DAC conversion is also needed only once, allowing continuous modulation in the MZM. Following this principle, we map each b_i to a MZM in the MM module (lines 3-10 in Algorithm 2) and map all the a_is to MZM_0 in the SM module across time steps (lines 11-13 in Algorithm 2).

Figure 2 (a) shows the mapping of decomposition-based multiplication when $k = 3$. Both A and B are decomposed into 3 nibbles where $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$. According to Algorithm 2, b_1, b_2 and b_3 are encoded through MZM_1, MZM_2 and MZM_3, respectively, and (a_1, a_2, a_3) is encoded via MZM_0 in three time steps. Figures 2 (b) and (c) show the mapping of round truncation-based multiplication when $k = 4$. All nibble multiplications labeled with the same color in Figure 2 (b) are performed in the same MZM in the MM module. For example, a_1b_1, a_2b_1, a_3b_1 and a_4b_1 are performed in MZM_1. If the number of nibbles in B is larger than the number of MZMs in the MM module (i.e., $m > k$), the nibbles in B need to be mapped over $t = \lceil m/k \rceil$ cycles to complete the computation (line 1 in Algorithm 2).

D. Comparison and Complexity Analysis

According to the mapping strategy in Algorithm 2, we analyze the time and energy costs of executing decomposition on BITLUME. A time step corresponds to converting a 4-bit digital value into an analog signal and modulating it via an MZM. As mappings to SM and MM can be performed concurrently, the total time steps required

Algorithm 2: Data Mapping for BITLUME(m, n)

$j \leftarrow 0$; $t \leftarrow \lceil m/k \rceil$; // Number of cycles
for $l \leftarrow 1$ *to* t **do**
  // Mapping strategy for B.
  for $z \leftarrow 1$ *to* k **do**
    if $j \leq m-1$ **then**
      $f_b : b_j \mapsto \text{MZM}_z$; $j \leftarrow j+1$;
    end
    else
      break;
    end
  end
  // Mapping strategy for A.
  for $i \leftarrow 0$ *to* $n-1$ **do**
    $f_a : a_i \mapsto \text{MZM}_0$;
  end
end

are $C = n \times \lceil \frac{m}{k} \rceil$. Energy consumption depends on the number of data reads and conversions, given by $T_m = n \times \lceil \frac{m}{k} \rceil + m$. The cost also depends on how A and B are mapped. If A is mapped to SM, the time cost is $n\lceil \frac{m}{k} \rceil$; otherwise, it becomes $m\lceil \frac{n}{k} \rceil$. Optimal mapping minimizes either time or energy: $\texttt{Min}\left(n\lceil \frac{m}{k} \rceil,\ m\lceil \frac{n}{k} \rceil\right)$ or $\texttt{Min}\left(n\lceil \frac{m}{k} \rceil + m,\ m\lceil \frac{n}{k} \rceil + n\right)$. For instance, in FP32 multiplication with a 24-bit mantissa, maximum reuse is achieved at $k = 6$, requiring only 12 reads and conversions.

We compare BITLUME (k=4), its rounding-truncation variant BITLUME_RT, and Karatsuba and FFT-based methods, all implemented with MZM-based photonic multipliers, in terms of photonic multiplication and data movement counts. Data movement refers to DAC reads and conversions from cache, which are closely tied to energy efficiency. As shown in Table I, for precisions below FP32, BITLUME's multiplication count is comparable to the other algorithms. Karatsuba is limited at low bit-widths due to overflow risks when reduced to 4-bit multiplications, requiring decomposition into sub-4-bit units. FFT, operating in the complex domain, incurs 3–4 real multiplications per complex operation, leading to significant overhead at low precision. While BITLUME has higher multiplication counts than Karatsuba and FFT beyond FP64, its optimized mapping and truncation schemes reduce data movement for FP128 and below. In contrast, both Karatsuba's recursion and FFT's complex multiplications suffer from limited data reuse, limiting mapping efficiency.

E. Photonic Dot Product on BITLUME.

Matrix multiplication constitutes a major portion of the computational workload in DNN training, especially in convolution operations, multiply-and-accumulate (MAC) operations, gradient computation, and weight updates. Matrix multiplication can be viewed as a series of dot product computations. As shown in Figure 4, MAC and convolution are used as examples to illustrate how BITLUME efficiently performs computation through data reuse.

TABLE I
COMPARISON OF MULTIPLICATION AND MAPPING COUNTS

Count	Multiplication				Data movement			
	FP16	FP32	FP64	FP128	FP16	FP32	FP64	FP128
BITLUME	9	36	196	841	6	18	70	261
BITLUME_RT	4	16	64	225	4	8	36	114
Karatsuba	9	36	117	363	18	72	234	726
FFT	20	64	128	256	40	128	256	512

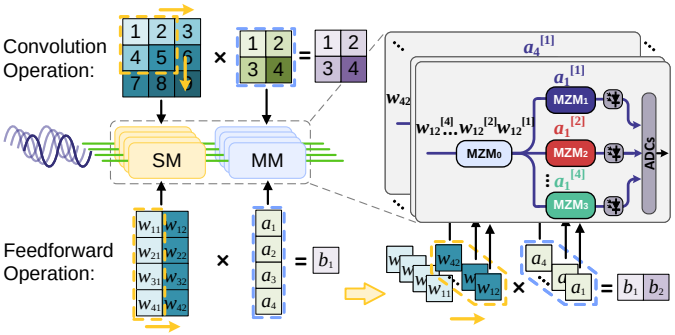

Fig. 4. Photonic dot product with data reuse.

In the feedforward MAC operations, each element a_i of the input vector a from the previous layer is mapped to a distinct MM module of a BITLUME unit, while the weight vector is mapped to the SM module. Since the input vector a remains constant within the layer, only the weight vector needs to be shifted, as indicated by the yellow arrows, to perform multiple MAC operations. This mapping strategy significantly enhances data reuse and reduces data movement.

Similarly, for convolution operations, each weight in the convolution kernel is mapped to an MM module. As the kernel weights remain fixed during convolution, they only need to be mapped once and can be reused as the kernel slides over the input feature map.

F. Architecture of BITLUME Core

Figure 5 (a) illustrates the multi-core architecture of the BITLUME photonic core, where data communication occurs via the on-chip interconnect bus. We design a BITLUME core that integrates multiple BITLUME units to accommodate both precision and flexibility in computation, as shown in Figure 5 (b). Each core comprises 18 computing units, including SM, MM, and Variable MZM (VM) modules. Each SM module contains one MZM, and each MM module contains 2 MZMs as illustrated in Figures 2 (d) and 5 (c). The VM is introduced to support the dynamic configuration of computing precision. The VM consists of two parallel MZMs, each with two inputs and outputs. It is connected to a tunable coupler at the end to select the output port. Hence, the two MZMs in the VM module can be individually used as two SMs modules or collectively used as one MM module. Below, we explain the operation of the BITLUME core under different precision configurations.

For FP32, the waveguides marked in red colour are used to interconnect the SM and two MM modules. The VM module is

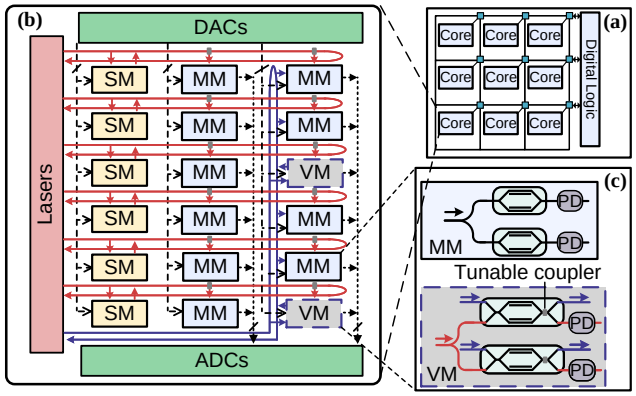

Fig. 5. (a) BITLUME multi-core interconnection. (b) Photonic core architecture. (c) Multiple and variable modulators.

configured to operate as an MM module, where the data is output from the bottom of the VM and controlled through the tunable coupler. Hence, each SM is connected to 2 MM modules (i.e., 4 MZMs in the MM modules), which is an optimal configuration for round-truncated FP32 according to the analysis given in Section III-C. For FP16 computation, the red waveguides are used to connect the SM modules to the MM modules in the middle column, allocating one MM to each SM. The VM module is configured to work as two SM modules and connects to the two MM modules in the right column using the blue waveguide.

This design avoids idleness caused by different matching modes of SM and MM, enabling the BITLUME core to flexibly switch between round-truncated FP32 and FP16 computations and achieve maximum device utilization in both modes. Our BITLUME core can be scaled to achieve increased parallelism in multiplications by proportionally integrating additional SM, MM, and VM modules.

IV. Photonic-Electric Hybrid Architecture

A. Overview of Photonic-Electronic Hybrid Architecture

In this section, we present our hybrid photonic-electronic architecture based on the BITLUME core for deep learning acceleration. As depicted in Figure 6, our architecture uses two independent SRAM units: Activation SRAM and Weight SRAM. The SRAM units can transfer data between each other through direct memory access (DMA) and communicate with the host and DRAM via the PCIe structure. A control unit manages the selection of DNN model parameters and the flow of activation data.

The data in the SRAM is parallelized and sent to the Synchronous Data Streamer (SDS) into three parts: the sign bit of the floating-point numbers is routed to the XOR unit, the exponent is directed to the exponent processing unit, and the mantissa is routed to the photonic computing core. In our architecture, SDS synchronizes the data streams in the electrical signal to match the operating frequency of the photonic dot product processor. The operating frequencies of DACs and ADCs determine the speed of the photonic processor.

After the BITLUME core completes mantissa multiplication, the result passes to the electronic addition tree, FP dot product accumulator, and nonlinear pipeline processing. Parallel add trees accumulate low-bit numbers. Parallel sign and exponent units handle sign bit XOR, exponent addition, and comparison, which combine with mantissa addition results in the FP accumulator to produce dot product outputs. Finally, results undergo nonlinear pipeline processing and are stored back to SRAM and DRAM.

B. Synchronous Data Stream

Photonic cores can operate at ultra-high clock frequencies, depending on the frequencies of DACs/ADCs and modulators. However, electronic processors typically operate within the range of 100 MHz to 4 GHz [18]. To bridge the gap, we parallelize the use of n digital data paths in each independent photonic processing unit to match the throughput of BITLUME and digital circuits. Each data path operates at a frequency of $\frac{1}{n}$ of the photonic unit clock frequency f.

To multiply two numbers using two cascaded MZMs, our designed data stream synchronization module consists of two parts. One part is dedicated to the input of the BITLUME core. We integrate a counter into the SDS and assign a ***valid*** flag to each DAC. The data streaming module triggers the voltage output only when all ***valid*** flags are asserted (i.e., all DACs are ready), ensuring synchronized operation. The other part handles the output from the BITLUME core. As shown in Figure 6, to prevent shift and addition operations from becoming bottlenecks in computation speed, we use n groups of parallel shifters and adders across the n data paths. Parallel shifters allow multiple shift operations to be completed within a single cycle [19]. The shifted values are then aggregated within an addition tree.

C. Pipeline Parallel Digital Computing

To prevent non-photonic operations from becoming bottlenecks, addition, shifting, and non-linear functions are executed using pipelined parallel computing. The pipelined floating-point parallel dot product in our architecture consists of three components: 1) a shift-adder tree for operand alignment and accumulation, and 2) a floating-point dot product accumulator. As shown in the right part of Figure 6, our design utilizes parallel floating-point dot product accumulators to handle vector products, where the mantissa multiplication results from multiple BITLUME cores are combined with sign bits and exponent bits in the accumulator to obtain the vector dot product result. In the aggregation process, the fraction must undergo alignment, summation, normalization, and final rounding. Before the final rounding, the most significant bit of the mantissa may not be within the expected range. At this point, Leading Zero Detection (LZD) determines the number of leading zeros in the mantissa. After the computation is completed, the result is passed to the Digital Processor for non-linear computations, such as ReLU and softmax.

V. Energy Consumption and Speed Analysis

A. Analysis on Energy Consumption of the BITLUME unit

For the BITLUME hybrid optoelectronic architecture, the primary energy consumption is considered from two factors: 1) system drive and, 2) data movement, which can be modelled as follows:

$$E = [\frac{k}{\rho^2} \cdot \frac{h\nu}{\eta} \cdot \max\left(2^{2N_b+1}, \frac{C_d V_r}{e}\right)] + [R(E_{\text{mod}} + E_{\text{DAC}}) + Wr(E_{\text{pd}} + E_{\text{ADC}})]. \quad (1)$$

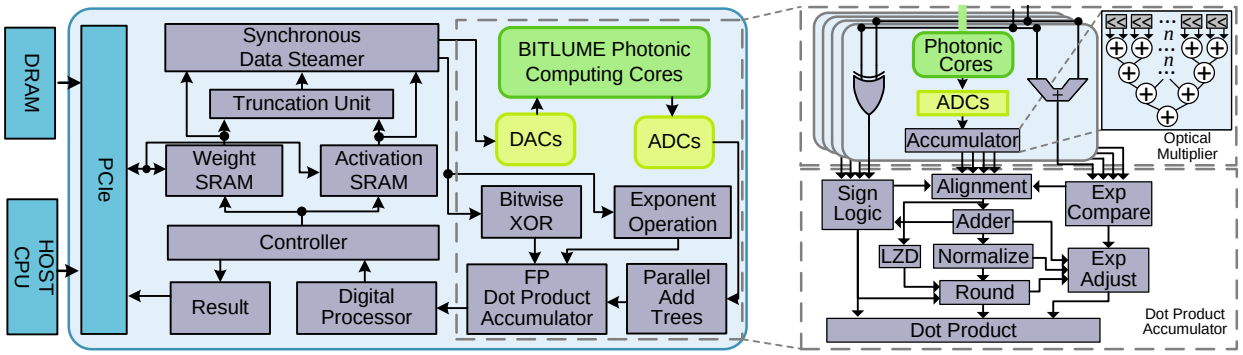

Fig. 6. Overview of photonic-electronic hybrid architecture and pipeline parallel digital computing.

The first item of energy consumption is the driving energy. Here, k represents the number of photodetectors that need to be charged. Since the light is evenly divided into k parts within the system, the optical power must be multiplied by k to compensate for the division losses ($k = 3$ for 16-bit and $k = 6$ for 32-bit). The precision loss is denoted as ρ, while η is the detection efficiency ($\eta = 0.2$). The photon energy is given by $h\nu$, where h is Planck's constant and ν is the frequency of the photon. Importantly, the minimum required energy is primarily determined by the larger of the laser driving energy or the noise energy consumption. In the case of laser drive, the photodetector capacitance $C_d = 1$ fF and the voltage $V_r = 0.5$ V are key factors [20]. The noise energy consumption, at high modulation frequencies, is typically dominated by relative intensity noise (RIN) and depends on the required signal-to-noise ratio (SNR), which must exceed 2^{N_b}, where N_b ($N_b = 8$) is the bit precision.

The second item is the data movement energy consumption. To perform multiplication in the BITLUM unit, we must first retrieve data from memory, modulate k signals at the input, and detect k signals at the output to return them to memory. We must account for the intrinsic costs of the driving and receiving circuits, including modulators E_{mod}, detectors E_{pd}, and memory I/O E_{DAC} and E_{ADC}. These energy costs are obtained from photonic links in [21], and related to the numbers of data reading R and writing Wr.

Based on the above energy model and the given parameters [20], [21], [22], [23] with all optical device parameters set at room temperature, we can obtain the energy consumption of different precisions. Fig. 7 (a) compares the energy consumption with various fixed-point precision per BITLUME unit. 8-bit fixed-point photonic computing consumes 15.04 fJ/MAC, while FP32 requires 78.05 fJ/MAC with multiplication data reuse. Optimizing dot-product data reuse (Section III-E) further lowers multiplication energy. In contrast, 7 nm GPUs and TPUs consume approximately 0.07 pJ/MAC and 0.34 pJ/MAC at 8-bit and 32-bit computations, respectively [18]. Dedicated FPGAs consume around 15 pJ/MAC for 8-bit computations [24]. This means that BITLUME's energy efficiency in the photonic domain is up to 4.36× greater than that of the latest electronic computing systems operating at 32-bit precision.

B. Analysis on Speed of the BITLUME Unit

The high-speed implementation of photonic multiplication operations is primarily constrained by the encoding and decoding of input and output signals in optoelectronic devices. The computation speed is mainly determined by the modulator operating frequency and the number of operations. The processing throughput of a standard BITLUME unit ($k = 3$) can be modelled as follows:

$$TP = \frac{f}{C} \times D_b, \tag{2}$$

where f is the operating frequency of BITLUME, C is the total time step, and D_b is the amount of data processed for that multiplication, measured in bytes. According to Algorithm 2, $C = n \times \lceil \frac{m}{k} \rceil$ and $D_b = \frac{N_p}{4}$, where N_p represents the bit precision of the computation.

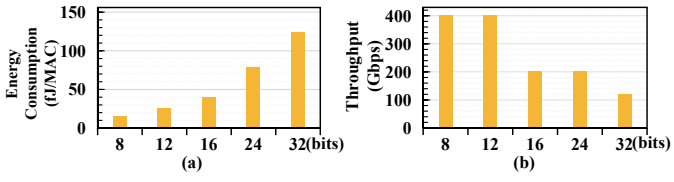

Fig. 7. Energy and throughput for a single unit with fixed-point precision.

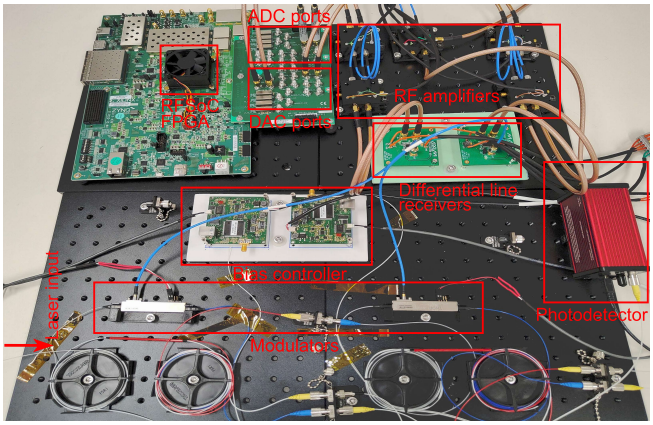

Fig. 8. BITLUME experimental setup with photonic and electronic devices.

The commercial lithium niobate ($LiNbO_3$) or indium phosphide (InP) optical modulators are widely used in fiber optics and data centers with operating frequencies up to 50 GHz. Additionally, emerging materials such as plasmonic and thin-film lithium niobate can operate at frequencies ranging from 100 to 500 GHz [6]. Assuming a 100 GHz modulator [22], a single standard BITLUME unit achieves a throughput of 200 Gbps at 24-bit fixed-point, as illustrated in Fig. 7 (b), with performance varying across different fixed-point precisions.

VI. BITLUME Prototype

A. Prototype Setup

Digital system setup: To validate the feasibility of our digital system, including memory, DAC/ADC, parallel shifter, adder tree, digital processor, controller, and so on, we develop the system using the Xilinx Zynq UltraScale + RFSoC ZU28DR FPGA platform [25]. Figure 8 illustrates the experimental device setup of BITLUME. The RTL implementation and cycle-accurate testing are conducted using Verilog language on Xilinx Vivado 2024.1 [26]. We integrate the BITLUME data path RTL design with the Xilinx RF Data Converter (DAC/ADC) IP and DDR4 DRAM IP to generate the bitstream in Vivado, which is then programmed into the ZCU111 board. The AXI4 stream protocol is employed for data transfer between the FPGA programmable logic and the DACs and ADCs, while the AXI Lite protocol is used for control signal and parameter transfer between the embedded PetaLinux [27] and the FPGA module.

The FPGA is configured to operate at a frequency of 256 MHz, with 16 samples per FPGA clock cycle, to generate a 4.096 GHz analog signal in conjunction with the Xilinx XM500 RF board [25]. The DAC and ADC are set to a data sampling rate of 4.096 GS/s, allowing the BITLUME optical platform to perform computations at a frequency of 4.096 GHz. Higher computation rates can be achieved by configuring DAC and ADC with higher sampling rates.

RF amplification setup: To minimize common-mode noise and crosstalk in the analog signal, we employ differential signal outputs at the DAC. To drive the optical modulator in the prototype, which requires an RF half-wave voltage of $V_{\text{pi}} = 4.5$ V, the RF signal must meet this requirement, whereas the XM500 board provides a DAC output voltage of approximately 1 V. Consequently, we use two LMH5401EVM differential amplifiers [28] in series to amplify the signal. The amplified signals are then aggregated by a differential line receiver for transmission to the modulator. Testing shows that the RF signal could provide approximately 3.0 V at the output stage after accounting for signal loss. At the ADC receiving end, a $V_{\text{cm}} = 1.25$ V

common-mode voltage is required for DC-coupled signals. Therefore, we also connect an LMH5401 amplifier at the photodetector output to apply the V_{cm} and amplify the analog signal.

Photonic components setup: Our optical component setup uses a tunable continuous-wave light source with a wavelength set to 1550 nm and a power of 6.6 dBm. We employed two 10 GHz Lithium Niobate Mach-Zehnder modulators [29] to achieve variable-bit optical multiplication. The transfer function of the MZM exhibits a sinusoidal shape, with significant nonlinearity occurring at the peaks and null points of the sine curve. During continuous operation of the MZM, the operating point tends to drift due to noise, leading to increased nonlinearity. To address this issue, a feedback control system, as shown in Figure 8 consisting of two MBC-SUPER bias controllers [30] is employed. This system locks the operating point near the quadrature point by monitoring the modulator's output signal, ensuring that the operating region remains approximately linear. At the output end, a 15 GHz photodetector RXM15EF [31] is used to detect light intensity and convert it into an analog signal.

B. Prototype Experiment Results

End-to-end latency: To evaluate the speed of photonic multiplications and dot product operations on our BITLUME experimental platform, we randomly generated 1000 sets of numbers and vectors of size 8. These operations are executed with varying precisions on the 4.096 GHz platform. The computation latency reflects the time from the moment a computation request arrives until the result exits the system. Figures 9 (a) and (b) compare the computation latency of multiplication and dot product operations using BITLUME versus the A100 GPU. We observed that for FP16 and FP64 dot product, BITLUME outperforms A100 by 3.9× and 3.6× in latency, respectively. On average, compared to the A100, the BITLUME experimental platform decreases the latency for multiplication by 3.8×, and for dot product operation by 3.7×. In Figure 9 (c), we decompose the computation latency in the dot products into two components: photonic core and electronic core. The optical core accounts for an average of about 41.5% of the total latency, which is 17% less than the digital core.

Computational accuracy: We evaluate computational accuracy by analyzing the error between photonic results and actual values. We quantized the analog output signal to the range of 0 to 256 to ensure 8-bit precision. The quantization step size was set to approximately 1, with each measurement exhibiting an error range of [-0.5, 0.5]. Fig. 10 (a) shows the error distribution, with the x-axis as the error and the y-axis as its probability density. The errors fit a Gaussian distribution with a mean of 0.0021 and a standard deviation of 0.15, which is significantly smaller than the quantization step size of 1. This aligns with prior findings that photonic noise under multiple sources approximates a Gaussian distribution [32] and shows that the effective precision of the system can reach 8-bit.

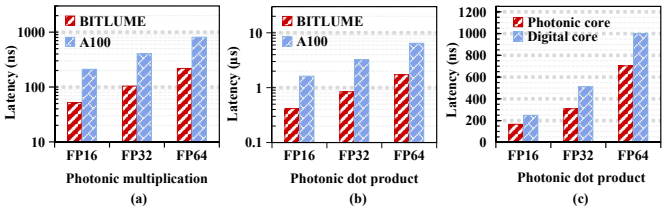

Fig. 9. End-to-end computation latency for (a) multiplication and (b) dot product operation; (c) Latency breakdown for dot product.

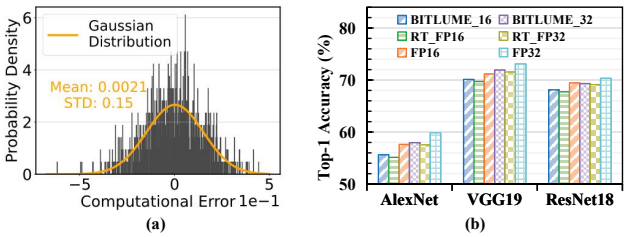

Fig. 10. (a) photonic computational error; (b) Top-1 accuracy for image classification training.

Training accuracy: To validate the training accuracy of BITLUME, we conducted image classification training on the AlexNet [33], VGG19 [34], and ResNet18 [35]models using the ImageNet dataset [36] with a batch size of 256. The error model is incorporated into the training process. As shown in Figure 10 (b), BITLUME_16 and BITLUME_32 indicate BITLUME executing training in FP16 and FP32 precision, respectively. Compared to digital accelerators, BITLUME's Top-1 training accuracy shows a difference of 1.99% for AlexNet, 1.04% for VGG19, and 1.33% for ResNet18 under FP16 precision. Under FP32 precision, the differences are 1.9% for AlexNet, 1.15% for VGG19, and 1.01% for ResNet18. Moreover, after applying round truncation, BITLUME exhibits only a slight reduction in average accuracy, 0.41% for FP16 and 0.32% for FP32.

VII. Performance Evaluation

A. Experimental Methodology

To evaluate the area and performance of an integrable-level chip, we extend the variable optical computing architecture design depicted in Figure 5 to incorporate 256 BITLUME cores. We utilize 100 GS/s manufactured DACs [37] and ADCs [23], as well as 100 GHz modulators [22] and photodetectors [38].

Area and power models: To evaluate the area and power of BITLUME, the digital logic and SRAM arrays are implemented in RTL and synthesized using Cadence Genus with a 65 nm library [39]. We employ two SRAM arrays, each with 8 MB size, consisting of 32 kB memory banks with an access latency of $\leq$ 1 ns. We obtain the area and power of electrical devices from the synthesis results. All electrical device parameters are scaled to 7 nm technology using the scaling equations from [40]. Finally, we can obtain the full chip area and power by combining the optical and electrical device parameters.

Performance models: We benchmark using PyTorch, decomposing benchmarks layer-wise for evaluating the performance of BITLUME. We evaluate the performance of BITLUME with three multiplication strategies (decomposition, rounding truncation, and Karatsuba) against that of A100 GPU [18] and Mirage [14]. Mirage is a state-of-the-art high-bit photonic DNN accelerator using a Residue Number System-based decomposing method. We adjust the electrical and optical device parameters of Mirage to align with those of BITLUME to ensure a fair comparison across architectures. The Karatsuba [15] is implemented within BITLUME's computational units for evaluation. The full system latency and energy consumption are evaluated by integrating the layer-wise data and device-level parameters in the FPGA and photonic platform. Specifically, we decompose the benchmarks' task types, computational load, and data volume, evaluating the workload for each layer under the same dataset while considering memory access, photonic/electronic links, computing delay, and energy consumption. For the GPU, PyTorch

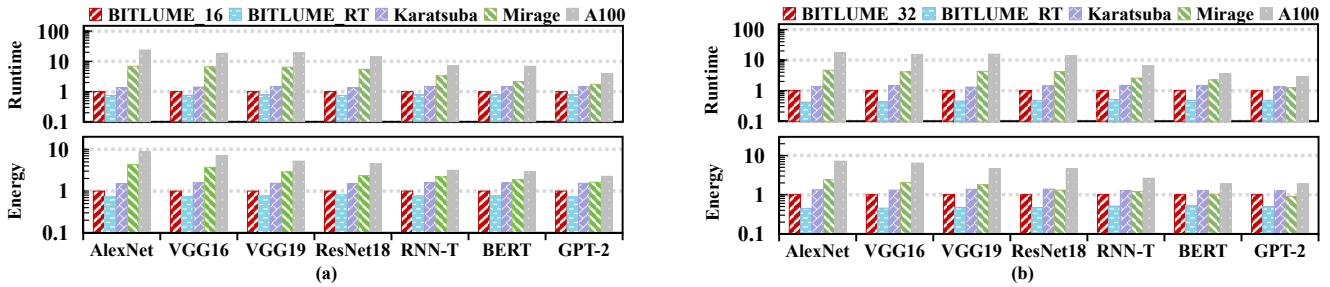

Fig. 11. Performance comparison between BITLUME, Round Truncation-based BITLUME, Karatsuba-based photonic computing, Mirage, and A100: (a) normalized runtime and energy for FP16; and (b) FP32 on DNNs training.

Profiler is used to monitor latency and memory usage, deriving energy consumption from power measurements obtained through Nvidia-smi.

Benchmarks and datasets: Seven benchmarks, AlexNet [33], VGG16, VGG19 [34] and ResNet18 [35] are trained with the ImageNet dataset [36] with an input batch size of 256. A Recurrent Neural Network Transducer (RNN-T) [41] is trained on the LibriSpeech dataset [42]. A transformer-based model, GPT-2 [43], is trained on the WebText set [43], and the BERT model [44] is trained on the SQuAD v1.1 [45] dataset. The model training is conducted on the A100 running CentOS 7.4 with CUDA 11.8 and PyTorch support.

B. Experimental Results

Area and power: Table II presents the detailed breakdown of area and power for the BITLUME chip. In the digital part, the digital logic units occupy 10.24 mm^2, the memory controller takes up 7.4 mm^2, and the data synchronization module occupies 76.16 mm^2. Their respective power consumptions are 10.72 W for the memory controller and 24.96 W for the synchronization module. The components required for optoelectronic conversion have a significant impact on the overall system in terms of both area and power. We adopt the key parameters from state-of-the-art DACs and ADCs: the DAC consumes 70.2 W and occupies 482.2 mm^2, while the ADC consumes 56 W and occupies 243.6 mm^2. In terms of photonic components, BITLUME integrates advanced manufacturable photonic devices, among which the modulators occupy the largest area, reaching 646.8 mm^2. Furthermore, based on the analysis of photon computing power in Section V, we can calculate the overall power of the photon system, which is 0.73 W. Combining the area of digital and photonic components, the digital components of the BITLUME chip require an area of 1555.01 mm^2 and power of 184.46 W. Notably, due to the large unit area of photonic devices, the modulator occupies 41.61% of the total area, making the overall system area slightly larger than that of electronic chips, with the A100 core die occupying a area of 826 mm^2. This is expected to improve with advancements in photonic device miniaturization. The BITLUME chip's total power is 2.16× lower than the A100's (400W).

Training speedup and energy saving in FP16 precision: Figure 11 (a) shows performance improvements in average runtime and energy consumption of BITLUME during training compared to the A100 GPU, Mirage at FP16 precision. We normalize BITLUME's performance in FP16 mode to 1. We consider time and energy for data writing, reading, and computing in the electronic domain, as well as computation in photonic cores. For runtime, we use our platform's measured computation times for a single DNN layer across different layer types, calculating total time by multiplying computation time by the number of layers in various DNNs. Energy consumption is obtained by multiplying the power of electronic cores by runtime.

As shown in Figure 11 (a), BITLUME demonstrates superior time and energy efficiency compared to Karatsuba-based photonic computing and Mirage. The recursive Karatsuba algorithm relies on the intermediate results from preceding stages. Its complex preprocessing (including addition, data partitioning, and memory access) introduces considerable overhead to the electronic system. In Mirage, complex modulo and reconstruction operations also significantly increase the workload in the electronic domain and reduce the overall system efficiency. On average, in FP16 precision, BITLUME achieves a speedup of 1.43×, 4.65× and 13.43×, and an energy saving of 1.54×, 2.7× and 4.81×, compared to the Karatsuba, Mirage, and A100, respectively. Additionally, compared to BITLUME_16, BITLUME_RT achieves improvements in speed and energy of 26.7% and 23.1% in FP16 precision.

TABLE II
ELECTRIC AND PHOTONIC DEVICE PARAMETERS.

	Devices	Area (mm^2)	Portion	Power (W)	Portion
Electric	Controller	7.4	0.48%	10.72	5.81%
	Sync module	76.16	4.89%	24.96	13.53%
	HBM [46]	81.1	5.21%	7.41	4.02%
	SRAM	7.4	0.48%	3.88	2.10%
	DAC [37]	482.2	31.01%	70.2	38.07%
	ADC [23]	243.6	15.65%	56	30.37%
	Digital logic	10.24	0.66%	10.56	5.73%
Photonic	MZM [22]	646.8	41.61%	0.73	0.40%
	Photodetector	0.1	0.01%		
	Laser [47]	0.01	0.001%		
	Total	1555.01	100%	184.46	100%

Training speedup and energy saving in FP32 precision: We trained seven models using BITLUME, GPU, and Mirage at FP32. We normalized the performance of BITLUME_32 to 1. As shown in Figure 11 (b), it can be observed that the performance improvement of BITLUME diminishes compared to other methods as the bit-width increases. With higher bit widths, BITLUME requires more frequent data reads and writes to memory, which reduces the overall system efficiency. On average, BITLUME_32 improves training time by 1.41×, 3.35× and 10.78×, and energy consumption by 1.36×, 1.53× and 4.12× compared to Karatsuba, Mirage and A100 in FP32, respectively. Additionally, compared to BITLUME_32, BITLUME_RT achieves improvements in speed and energy of 54.2% and 50.7%.

VIII. CONCLUSIONS

To address computational precision limitations in photonic systems, we propose the BITLUME unit with a variable precision algorithm, enabling computations beyond FP32. By optimizing data mapping and core architecture, we improve energy efficiency through data reuse and minimize optoelectronic conversions. We analyze BITLUME's energy consumption and speed, identifying key performance factors. A high-precision 4.096 GHz hybrid optoelectronic prototype is built to explore BITLUME's potential. Simulation results demonstrate the accuracy and effectiveness of BITLUME.

REFERENCES

[1] A. Koubaa, "Gpt-4 vs. gpt-3.5: A concise showdown," 2023.

[2] NVIDIA, "Scaling language model training to a trillion parameters," *Online. URL https://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/*, 2007.

[3] Katyanna, Quach, "Ai me to the moon... carbon footprint for 'training gpt-3' same as driving to our natural satellite and back." https://www.theregister.com/2020/11/04/gpt3_carbon_footprint_estimate/, 2020.

[4] J. Koetsier, "Photonic supercomputer for ai: 10x faster, 90% less energy, plus runway for 100x speed boost," *Forbes*, 2021.

[5] Z. Zhong, M. Yang, J. Lang, C. Williams, L. Kronman, A. Sludds, H. Esfahanizadeh, D. Englund, and M. Ghobadi, "Lightning: A reconfigurable photonic-electronic smartnic for fast and energy-efficient inference," in *Proceedings of the ACM SIGCOMM Conference*, 2023, pp. 452–472.

[6] Y. Qi and Y. Li, "Integrated lithium niobate photonics," *Nanophotonics*, vol. 9, no. 6, pp. 1287–1320, 2020.

[7] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang, "Holylight: A nanophotonic accelerator for deep learning in data centers," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1483–1488.

[8] K. Shiflett, A. Karanth, R. Bunescu, and A. Louri, "Albireo: Energy-efficient acceleration of convolutional neural networks via silicon photonics," in *ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, 2021, pp. 860–873.

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

[10] S. Hua, E. Divita, S. Yu, B. Peng, C. Roques-Carmes, Z. Su, Z. Chen, Y. Bai, J. Zou, Y. Zhu *et al.*, "An integrated large-scale photonic accelerator with ultralow latency," *Nature*, vol. 640, no. 8058, pp. 361–367, 2025.

[11] S. R. Ahmed, R. Baghdadi, M. Bernadskiy, N. Bowman, R. Braid, J. Carr, C. Chen, P. Ciccarella, M. Cole, J. Cooke *et al.*, "Universal photonic artificial intelligence acceleration," *Nature*, vol. 640, no. 8058, pp. 368–374, 2025.

[12] F. Zhu, R. Gong, F. Yu, X. Liu, Y. Wang, Z. Li, X. Yang, and J. Yan, "Towards unified int8 training for convolutional neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1969–1979.

[13] F. P. Sunny, E. Taheri, M. Nikdast, and S. Pasricha, "A survey on silicon photonics for deep learning," *ACM Journal of Emerging Technologies in Computing System*, vol. 17, no. 4, pp. 1–57, 2021.

[14] C. Demirkiran, G. Yang, D. Bunandar, and A. Joshi, "Mirage: An rns-based photonic accelerator for dnn training," in *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2024, pp. 73–87.

[15] T. E. Pogue and N. Nicolici, "Karatsuba matrix multiplication and its efficient custom hardware implementations," *IEEE Transactions on Computers*, 2025.

[16] J. Nie, Q. Zhu, M. Li, and X. Sun, "Quantum circuit design for integer multiplication based on schönhage–strassen algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4791–4802, 2023.

[17] A. Emerencia, "Multiplying huge integers using fourier transforms," *Online slides. URL http://www.cs.rug.nl/ãndo/pdfs/Ando_Emerencia_ multiplying_huge_integers_using_fourier_transforms_presentation.pdf*, 2007.

[18] Nvidia, Inc, "Nvidia a100 gpu," https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf., 2021.

[19] Y. Hilewitz and R. B. Lee, "A new basis for shifters in general-purpose processors for existing and advanced bit manipulations," *IEEE Transactions on computers*, vol. 58, no. 8, pp. 1035–1048, 2008.

[20] K. Nozaki, S. Matsuo, T. Fujii, K. Takeda, M. Ono, A. Shakoor, E. Kuramochi, and M. Notomi, "Photonic-crystal nano-photodetector with ultrasmall capacitance for on-chip light-to-voltage conversion without an amplifier," *Optica*, vol. 3, no. 5, pp. 483–492, 2016.

[21] M. A. Nahmias, T. F. De Lima, A. N. Tait, H.-T. Peng, B. J. Shastri, and P. R. Prucnal, "Photonic multiply-accumulate operations for neural networks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–18, 2019.

[22] S. Wolf, H. Zwickel, W. Hartmann, M. Lauermann, Y. Kutuvantavida, C. Kieninger, L. Altenhain, R. Schmid, J. Luo, A. K.-Y. Jen *et al.*, "Silicon-organic hybrid (soh) mach-zehnder modulators for 100 gbit/s on-off keying," *Scientific reports*, vol. 8, no. 1, pp. 1–13, 2018.

[23] R. Nguyen, A. Mellati, A. Fernandez, A. Iyer, A. Fan, B. Reyes, C. Abidin, C. Nani, D. Albano, F. Ahmad *et al.*, "18.4 a 200gs/s 8b 20fj/cs receiver with¿ 60ghz afe bandwidth for 800gb/s optical coherent communications in 5nm finfet," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67. IEEE, 2024, pp. 344–346.

[24] T. Nguyen, S. Williams, M. Siracusa, C. MacLean, D. Doerfler, and N. J. Wright, "The performance and energy efficiency potential of fpgas in scientific computing," in *2020 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*. IEEE, 2020, pp. 8–19.

[25] Advanced Micro Devices, Inc, "Zynq ultrascale+ rfsoc zcu111," https://www.xilinx.com/products/boards-and-kits/zcu111.html, 2024.

[26] A. M. Devices, "Amd vivado design suite," https://www.xilinx.com/products/design-tools/vivado.html, 2024.

[27] Advanced Micro Devices, "Petalinux tool," https://www.xilinx.com/products/design-tools/embedded-software/petalinux-sdk.html, 2024.

[28] Texas Instruments, Inc, "Lmh5401 evaluation module," https://www.ti.com/tool/LMH5401EVM, 2024.

[29] Thorlabs, Inc, "Lna2322 - 10 ghz intensity modulator," https://www.thorlabs.com/thorproduct.cfm?partnumber=LNA2322, 2024.

[30] Ozoptics, Inc, "Super modulator bias controller," https://www.ozoptics.com/ALLNEW_PDF/DTS0165.pdf, 2024.

[31] Thorlabs. Inc, "Rxm15ef - multimode ultrafast receiver," https://www.thorlabs.com/thorproduct.cfm?partnumber=RXM15EF, 2024.

[32] A. Sludds, R. Hamerly, S. Bandyopadhyay, Z. Zhong, Z. Chen, L. Bernstein, M. Ghobadi, and D. Englund, "Demonstration of wdm-enabled ultralow-energy photonic edge computing," in *2022 Optical Fiber Communications Conference and Exhibition (OFC)*, 2022, pp. 1–3.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 84–90, 2012.

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

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

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[37] F. Ahmad, A. Mellati, A. Fernandez, A. Iyer, A. Fan, B. Reyes, C. Abidin, C. Nani, D. Albano, F. Solis *et al.*, "18.3 an 8b 160gs/s 57ghz bandwidth time-interleaved dac and driver-based transmitter with adaptive calibration for 800gb/s coherent optical applications in 5nm," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67. IEEE, 2024, pp. 342–344.

[38] D. Maes, L. Reis, S. Poelman, E. Vissers, V. Avramovic, M. Zaknoune, G. Roelkens, S. Lemey, and E. Peytavit, "High-speed photodiodes on silicon nitride with a bandwidth beyond 100 ghz," in *CLEO: Science and Innovations*. Optica Publishing Group, 2022, pp. 1–2.

[39] Cadence, "Genus synthesis solution," https://www.cadence.com/en_US/home.html, 2023.

[40] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of cmos device performance from 180 nm to 7 nm," *Integration*, vol. 58, pp. 74–81, 2017.

[41] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[42] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[43] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[45] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[46] Micron Technology, "Hbm3e," https://www.micron.com/products/memory/hbm/hbm3e, 2023.

[47] X. Xue, P.-H. Wang, Y. Xuan, M. Qi, and A. M. Weiner, "Microresonator kerr frequency combs with high conversion efficiency," *Laser & Photonics Reviews*, vol. 11, no. 1, p. 1600276, 2017.