ROCKET: An RNS-based Photonic Accelerator for High-Precision and Energy-Efficient DNN Training

Hao Zhang
University of Otago
Dunedin, New Zealand
hao.zhang@postgrad.otago.ac.nz

Haibo Zhang
University of Otago
Dunedin, New Zealand
Haibo.Zhang@anu.edu.au

Chengpeng Xia
University of Otago
Dunedin, New Zealand
chengpeng.xia@postgrad.otago.ac.nz

Zhiyi Huang
University of Otago
Dunedin, New Zealand
zhiyi.huang@otago.ac.nz

Yawen Chen
University Of New South Wales
Sydney, Australia
wendy.chen1@unsw.edu.au

Amanda Barnard
Australian National University
Canberra, Australia
amanda.s.barnard@anu.edu.au

Abstract

In recent years, the rapid development of Deep Neural Networks (DNNs) has posed significant challenges in terms of training duration and costs. High-frequency, low-power photonic computing has emerged as a highly promising solution. However, the substantial cost of data conversion and the limitations introduced by noise in photonic devices continue to hinder the realization of high-precision and energy-efficient DNN training. To address this challenge, we propose a novel photonic accelerator, ROCKET, based on the Residue Number System (RNS). RNS is based on modular arithmetic and enables support for high-precision computation through parallel multi-path low-precision operations. First, we leverage specialized lookup tables to enable high-throughput, low-latency conversions between high-precision and low-precision numerical representations. Next, we design a low-power photonic accelerator architecture utilizing intensity modulators, which minimizes the number of computational components while maximizing data reuse. Subsequently, we propose a hybrid photonic-electronic pipelined dataflow to maximize parallelism within the photonic-electronic computation path. Finally, we develop a high-frequency (4.096 GHz) hybrid photonic-electronic prototype using FPGA, Radio Frequency (RF), and photonic components to validate the feasibility of the ROCKET. Our large-scale simulations on seven mainstream DNN models show that, compared to the A100 GPU, TPU v4, and the state-of-the-art photonic accelerator Mirage, ROCKET achieves speedups of 33×, 243×, and 198×, respectively, while saving energy by factors of 64×, 204×, and 142×.

ICS '25, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-1537-2/25/06
https://doi.org/10.1145/3721145.3734529

CCS Concepts

• **Hardware → Emerging optical and photonic technologies**; **Hardware accelerators**; • **Computer systems organization → Optical computing**; • **Computing methodologies → Artificial intelligence**.

Keywords

DNN training, Photonic computing, High-precision, Energy-efficient, RNS

ACM Reference Format:
Hao Zhang, Haibo Zhang, Chengpeng Xia, Zhiyi Huang, Yawen Chen, and Amanda Barnard. 2025. ROCKET: An RNS-based Photonic Accelerator for High-Precision and Energy-Efficient DNN Training. In *2025 International Conference on Supercomputing (ICS '25), June 08–11, 2025, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3721145.3734529

1 Introduction

The continuous advancement of DNNs is redefining modern life. As the scale of DNNs expands, there is an urgent need for high-performance and energy-efficient systems to accelerate the training process. A notable example is the training of GPT-3, which required 10,000 V100 GPUs running for 14.8 days, consuming 1,287 MWh of electricity [41]. Therefore, there is an urgent need to develop new computing paradigms to effectively address the challenges of high-performance computing demands and large-scale energy consumption.

Photonic computing, as a powerful solution for next-generation high-performance computing, inherently offers advantages

such as low latency, high bandwidth, and low power consumption. The key principle of photonic computing lies in the fact that the clock frequency of photonic devices is significantly faster than that of transistors, with a speed difference of 1-2 orders of magnitude [20]. Additionally, the power consumption of transistor-based circuits is proportional to the cube of the clock frequency f [32], while the power consumption of photonic-electronic accelerators is linearly related to f [67], resulting in less heat generation. Therefore, photonic computing is expected to meet the performance demands of AI hardware in the post-Moore era.

However, due to precision limitations, photonic computing struggles to meet the requirements of high-precision DNN training. Extensive Digital-to-Analog (DA) and Analog-to-Digital (AD) conversions are required before and after photonic computing, which not only create performance bottlenecks but also result in significant power consumption. As bit precision increases, the power consumption of Analog-to-Digital Converters (ADCs) and Digital-to-Analog Converters (DACs) rises correspondingly. Notably, ADCs exhibit an exponential increase in power consumption, with each additional bit of precision approximately quadrupling the energy consumption of the conversion process [36]. Comparing to the high energy consumption of high-precision ADCs (orders of nJ), multiplication operations in the photonic domain only consume tens to hundreds of aJ [52]) (one aJ is 10^{-9} nJ). High-precision ADCs can easily dominate the overall system energy consumption. Furthermore, to ensure computational integrity, the Signal-to-Noise Ratio (SNR) required for photonic devices increases exponentially with bit precision, necessitating corresponding power levels. Therefore, the precision issues in photonic computing have always been a significant challenge.

Due to precision limitations, most photonic accelerators currently focus on DNN inference [30, 49, 50, 54, 63, 69], which is less sensitive to quantization noise. Additionally, some studies have focused on enabling low-precision DNN training [10, 13], but they still face significant challenges in maintaining high model accuracy. Only a few studies have attempted to achieve high-precision DNN training using photonic computing. For example, the state-of-the-art photonic accelerator for DNN training, Mirage [16], still faces significant challenges in terms of high power consumption and large area requirements. This is attributed to the architectural requirements, which mandate the deployment of numerous photonic components such as phase shifters, microring resonators, and a large number of high-bit-width DACs. Generally speaking, achieving high-precision and low-power DNN training remains a critical and significant challenge.

In this paper, we propose ROCKET, an RNS-based photonic accelerator for high-precision and energy-efficient DNN training. In RNS, multiple low-precision computation results combine to reconstruct high-precision outcomes. Each low-precision residue operation performs independently of other residues, demonstrating significant parallelism potential. To obtain low-precision residue operands, a set of pairwise co-prime moduli needs to be selected. Our moduli selection scheme supports any set of co-prime moduli, replacing the fixed three-moduli set. To achieve low-power parallel matrix multiplication under RNS, we design a novel photonic accelerator based on intensity modulators. The architecture of the accelerator is simple, with the basic multiplication component requiring only two modulators. Further, we propose a hybrid photonic-electronic pipeline dataflow to fully exploit the performance potential of this accelerator. For the photonic computing path, we introduce synchronization and data recognition modules to align the high-speed photonic computation frequency with the lower-speed electronic data path frequency. Compared to previously proposed photonic accelerator for DNN inference, Lightning [69], which utilizes Directed Acyclic Graphs (DAGs) for task scheduling, the hybrid pipeline design does not require an extremely time-consuming DAG reconstruction process when changes occur in the training set or batch size, thereby enhancing performance and reducing power consumption.

In particular, this paper makes the following contributions.

- We propose an RNS-based photonic accelerator architecture for high-precision DNN training, which minimizes the number of photonic components while maximizing data reuse. To the best of our knowledge, ROCKET is the first photonic accelerator capable of achieving high-precision and low-power DNN training.
- We propose a hybrid photonic-electronic pipeline dataflow design that addresses the mismatch between the speeds of traditional electronic pipelines and photonic computation frequencies, thereby enabling efficient and accurate DNN training.
- We build a high-frequency 4.096 GHz hybrid photonic-electronic prototype, integrating FPGA, RF components, and photonic elements, to demonstrate the feasibility of the ROCKET accelerator. Experimental results show that ROCKET outperforms the A100 GPU and TPU v4 in dot product operations by 4.7× and 4.0×, respectively. Large-scale simulations across seven mainstream DNN models reveal that ROCKET achieves an average speedup of 33×, 243×, and 198× in training time, while reducing energy consumption by 64×, 204×, and 142× compared to Mirage, A100 GPU, and TPU v4, respectively.

The rest of the paper is organized as follows: Section 2 presents the background and motivation. Section 3 describes

the ROCKET accelerator architecture. Section 4 introduces the dataflow design based on a hybrid photonic-electronic pipeline. Sections 5 and 6 shows the validation prototype setup, performance evaluation, and large-scale simulation results. Section 7 presents related works. Finally, Section 8 concludes the paper.

2 Background and Motivation

2.1 Photonic Multiplication

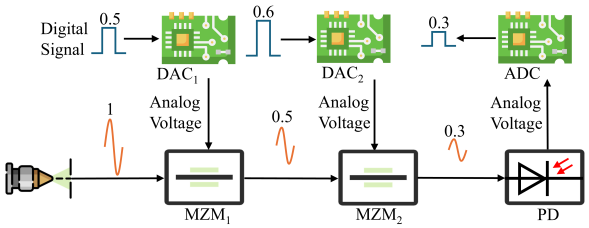

Figure 1: Intensity MZMs-based photonic multiplication.

A common technique to perform multiplication in the photonic domain is to cascade two Mach-Zehnder modulators (MZMs) to multiply two input voltages from DACs [19], as shown in Figure 1. DAC_1 applies an input voltage V_1 to MZM_1, generating a light wave with an intensity proportional to V_1. This light wave serves as the carrier signal for MZM_2. Then, DAC_2 applies a second input voltage V_2 to MZM_2, generating a light wave with an intensity proportional to V_2. The result is a double-modulated light wave with an amplitude proportional to $V_1 \times V_2$. The photodetector receives the light intensity from MZM_2 and converts it into an analog signal. Finally, the ADC converts this analog signal into a digital signal. For example, let $V_1 = 0.5$ and $V_2 = 0.6$ represent the input numbers in the electrical domain. By feeding these numbers into the two MZMs shown in Figure 1, the intensity of the output light from the MZM_2 becomes proportional to the multiplication of the two input voltages, $V_1 \times V_2 = 0.3$.

2.2 Bit Precision in Photonic Accelerators

For hybrid photonic-electronic accelerators, the achievable computation precision is primarily determined by two factors: the precision of data converters in the electronic domain, including DACs and ADCs, and SNR during operations in the photonic domain [15].

2.2.1 The Precision of DACs and ADCs. In the photonic domain, the multiplication of an N_{in}-bit input and an N_{w}-bit weight, both encoded by DACs, results in an $N_{\text{out}} = N_{\text{in}} + N_{\text{w}}$-bit output, which is decoded by the ADC. For example, for the multiplication of 8-bit numbers, the output precision needs to be at least 16 bits—this requires using an $b_{\text{ADC}} \geq 16$ ADC to ensure the accuracy of the result. Previous studies have shown that the energy consumption of ADCs is two orders of magnitude higher than that of DACs. More importantly, for each additional bit of ADC precision, the energy consumption increases by approximately a factor of four [16, 36]. For instance, for a 16-bit precision ADC, a single AD conversion requires more than 1 nJ of energy. Given that the energy consumption of multiplication operations in the analog domain is relatively low (tens to hundreds of aJ per operation), the energy consumption of high-precision ADCs can easily dominate the total energy consumption.

2.2.2 The Impact of SNR on Precision. Besides the limitations of data converters, the SNR requirements also restrict the achievable precision. Two primary sources of analog noise—shot noise and thermal noise—play a dominant role in determining the effective SNR. Shot noise arises from the statistical fluctuations in the number of incident photons or generated electrons. It is commonly modeled as a zero-mean Gaussian process, with a variance proportional to the photodetector current and the bandwidth. The corresponding formula is given as follows:

$$I_{\text{shot}}^2 = 2qI_{\text{pc}}\Delta f, \tag{1}$$

where q is the electron charge, I_{pc} is the photodetector current, and Δf is the bandwidth.

Thermal noise, on the other hand, originates from the resistive elements in the trans-impedance amplifier (TIA), and similarly follows a Gaussian distribution with variance determined by temperature, resistance, and bandwidth. The corresponding formula is given as follows:

$$I_{\text{thermal}}^2 = \frac{4kT\Delta f}{R}, \tag{2}$$

where k is the Boltzmann constant, T is the TIA's equivalent noise temperature (in Kelvin), and R is the input resistance.

To achieve a bit precision of b, the system must be capable of resolving 2^b discrete levels, which implies that the required SNR must be at least 2^b. A commonly adopted strategy to meet this requirement is to increase the optical input power, thereby boosting the signal amplitude to reach the necessary SNR. In other words, achieving high precision requires a high SNR, which significantly increases the system's power consumption. Therefore, designing a photonic-electronic accelerator for DNN training that concurrently achieves high precision and energy efficiency is of significant research value and practical importance.

2.3 The Residue Number System

RNS decomposes large-number computations into multiple modular spaces, each corresponding to a set of mutually prime moduli. This enables independent and parallel computation across different moduli, improving computational

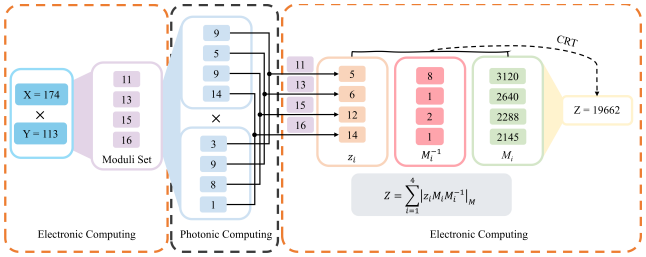

Figure 2: The multiplication process under RNS.

efficiency while supporting high-precision DNN training. In RNS, each number X is represented as a tuple $(x_1, x_2, \ldots, x_n)$, where x_i is obtained by taking the remainder with respect to a set of given co-prime moduli $\mathbb{M} = \{m_1, m_2, \ldots, m_n\}$. Let us take the multiplication of $X = 174$ and $Y = 113$ in the binary number system (BNS) as an example. As shown in Figure 2, in the BNS-to-RNS conversion, given the moduli set $\{11, 13, 15, 16\}$, X and Y are represented as (9, 5, 9, 14) and (3, 9, 8, 1), respectively. The upper limit of the dynamic range of the moduli set is determined by $M = m_1 \times m_2 \times \ldots \times m_n$, meaning that each number N less than M has a unique representation, i.e., the dynamic range is $[0, M)$. The upper limit of the dynamic range for the moduli set (11, 13, 15, 16) is $M = 11 \times 13 \times 15 \times 16 = 34320$.

In RNS, addition and multiplication operations are closed, meaning that the results of these operations can still be represented as remainders within the same modulus set. Consider a modulus set $(m_1, m_2, \ldots, m_n)$ and two input numbers for addition and multiplication, $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$. The specific formula for multiplication is as follows:

$$R = X \times Y = (r_1, r_2, \ldots, r_n), \tag{3}$$

where $r_i = |x_i \times y_i|_{m_i}$. As shown in Figure 2, $R = X \times Y = (|9 \times 3|_{11}, |5 \times 9|_{13}, |9 \times 8|_{15}, |14 \times 1|_{16}) = (5, 6, 12, 14)$.

For RNS-to-BNS conversion, the Chinese Remainder Theorem (CRT) is applied to uniquely reconstruct the integer Z from its remainders and moduli [42]. It is essential to ensure that all moduli are coprime and that the reconstructed integer Z falls within the dynamic range $[0, M)$. The specific formula is as follows:

$$Z = \sum_{i=1}^{n} \left| z_i M_i M_i^{-1} \right|_M \tag{4}$$

where $M_i = \frac{M}{m_i}$, and M_i^{-1} is the multiplicative inverse of M_i modulo m_i, such that $M_i \cdot M_i^{-1} \equiv 1 \pmod{m_i}$. This inverse can be computed using the extended Euclidean algorithm [24]. As shown in Figure 2, in the RNS-to-BNS conversion, the number $Z = 19662$ is uniquely reconstructed using the CRT.

3 ROCKET Accelerator Architecture

3.1 Moduli Selection

Moduli selection plays a critical role in RNS. To prevent overflow and ensure the integrity of multiplication operations under RNS, it is crucial to ensure that the product of the moduli is not less than the maximum possible value of the intermediate results during residue computation. In other words, the choice of bit width for each modulus and the size of the moduli set are crucial. It is noteworthy that in the RNS context, the photonic multiplication of Floating-Point (FP) numbers refers specifically to the multiplication of the mantissa, while other calculations are carried out using electronic components. To meet the low power consumption requirements [36], the ADC precision should not exceed 9 bits, and the DAC precision should not exceed 5 bits. Hence, the maximum value of the selected moduli should not exceed 32. At the same time, to ensure that intermediate results do not overflow, it is necessary to ensure that the dynamic range of the moduli set is at least twice the size of the mantissa of the FP number.

The classical 3-moduli set $\{2^k - 1, 2^k, 2^k + 1\}$ and its circuits [23] are clearly unsuitable for high-precision training. When $k = 4$, the supported dynamic range is less than 12 bits. For instance, the multiplication of high-precision BFloat16 numbers requires a dynamic range of at least 16 bits. Because BFloat16 has a 7-bit mantissa and 1-bit hidden bit, multiplication operations will produce a 16-bit intermediate result. To support high-precision DNN training using the BFloat16 format, a 5-moduli set $\{11, 13, 15, 16, 17\}$ is employed for computation in the RNS. A one-to-one fast lookup table structure is used to perform the BNS-to-RNS remainder operation. Since the mantissa of BFloat16 corresponds to only 256 unique integer values in fixed-point representation, the lookup table needs to store only 256 entries. Each entry contains the residues with respect to the five selected moduli, requiring approximately 21 bits (about 3 bytes) of storage. Consequently, the total memory overhead of the lookup table is constrained to within 1 KB. Similarly, the reverse conversion from RNS to BNS is also implemented via lookup tables. These tables are computed once prior to training and reused throughout. Compared to the 1 mW power consumption of traditional specialized circuits [23], the specially designed lookup table structure—implemented at the L1 level of the memory hierarchy—can achieve an access energy of less than 0.5 mW per lookup [46].

3.2 Photonic Multiplication Matrix Unit

DNN training includes two key steps: forward pass and backward pass [68]. Since the core computational processes of both steps involves general matrix-matrix multiplications

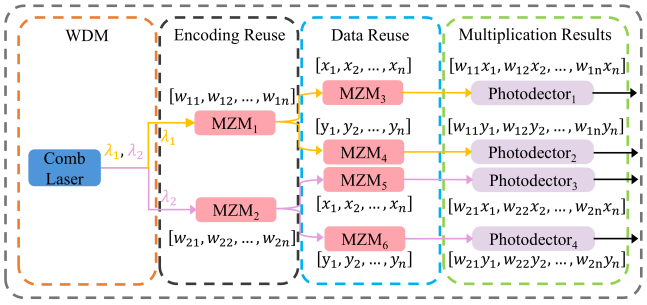

Figure 3: Photonic multiplication matrix units.

(GEMMs), this paper takes forward pass as an example (unless otherwise specified). GEMM operations under RNS are composed of dot products, where each dot product consists of two steps: element-wise multiplication and summation. This section focuses exclusively on the element-wise multiplication performed within the photonic computing core, following the conversion from BNS to RNS. The accumulation process is handled by the adder tree unit, which will be introduced in section 3.3.

In Figure 1, we introduce the basic photonic multiplication unit (PMU). To obtain the element-wise product of two vectors, the elements of the weight vector $\mathbf{w}$ and the input vector $\mathbf{x}$ are modulated in the first and second columns of MZMs, respectively, to achieve photonic multiplication. To achieve parallel computation of multiple sets of element-wise products, the simplest method is to replicate PMU multiple times. The laser source provides light waves of specific wavelengths, which enter different PMUs via a splitter for modulation.

To reduce the usage of computational components and enhance computational performance, we design a novel Photonic Multiplication Matrix Unit (PMMU) and propose a corresponding data mapping scheme, as illustrated in Figure 3. Consider a DNN weight matrix $\mathbf{W}$ containing two row vectors $\mathbf{w_1} = [w_{11}, w_{12}, ..., w_{1n}]$ and $\mathbf{w_2} = [w_{21}, w_{22}, ..., w_{2n}]$, as well as an input matrix $\mathbf{A}$ for a batch of 2 containing two input vectors $\mathbf{x} = [x_1, x_2, ..., x_n]$ and $\mathbf{y} = [y_1, y_2, ..., y_n]$. First, we utilize WDM technology to provide two light waves with different wavelengths, λ_1 and λ_2. These wavelengths are then output to two different channels via a demultiplexer (DEMUX), where MZM_1 and MZM_2 encode the weight parameters of two rows. Unlike the previous simple replication, this design achieves encoding reuse, replacing the original four MZMs with two. The modulated light waves then enter the second column of MZMs via a splitter, with MZM_3 and MZM_4 encoding data identical to that of MZM_5 and MZM_6. This mapping method, by reusing input data already loaded into caches or registers, reduces the need for main memory access, thereby improving system performance and reducing latency and energy consumption. Finally, PD_1 to PD_4 detect the doubly modulated light waves and convert them into voltage signals, which are then sent to ADCs to be converted into digital signals. Considering that the computation paradigm for different remainders is consistent under the RNS, when constructing RNS-based PMMUs, we simply provide the corresponding number of PMMUs based on the size of the moduli set. For example, when the size of the modulus set is 5, the number of required PMMUs is also 5.

3.3 RNS-based Photonic Computing Unit

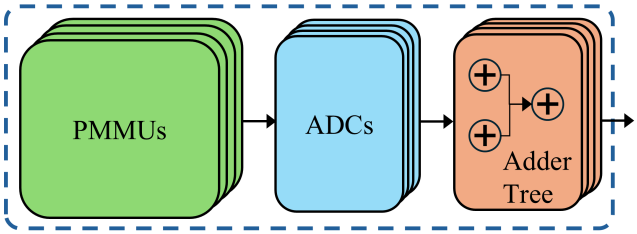

Figure 4: RNS-based photonic computing units.

To compute dot products, a hybrid photonic-electronic computing unit is designed, as illustrated in Figure 4. First, element-wise multiplication under the RNS is performed using PMMUs. The resulting analog voltage signals are then converted into multiplication results in the digital domain through ADC arrays. All multiplication results, along with the corresponding exponent and sign operations, are accumulated in parallel through an adder tree unit. Note that each ADC corresponds to an adder tree unit. The implementation of the adder tree follows the hardware design methodology described in [5]. Although the data samples read from the ADC are 8 bits, we set the bit-width of the adder components to 16 bits to prevent overflow during accumulation. Before entering the adder tree, each sample is padded with an additional 8 zeros. Parallel accumulation is necessary because each ADC readout includes multiple parallel data samples. For instance, if the ADC samples voltage readouts at a rate of 4.096 GS/s, and these digital data are read into the electronic computing path at a frequency of 256 MHz, it means that approximately every 4 ns, the ADC transmits 16 parallel samples to the electronic computing path. After accumulation by the adder tree, the mantissa part of the FP result undergoes a modulus operation to obtain a partial modular dot product result. In addition, the adder tree unit is also employed to aggregate intermediate results until the accumulation of the entire vector is completed.

3.4 ROCKET Accelerator Design

Figure 5 shows the main architectural design of ROCKET. ROCKET achieves tight coupling between the photonic and

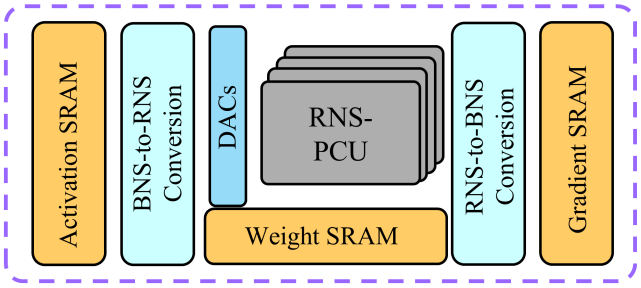

Figure 5: ROCKET Accelerator Architecture Design.

electronic chiplets through 3D integration [29], effectively reducing system latency and optimizing overall energy efficiency. The photonic chiplet includes several key photonic devices, such as MZIs and waveguides, while the electronic chiplet includes SRAMs, DAC/ADC arrays, and adder trees. In this design, the input and output weights, which are partitioned into tiles, first undergo BNS-to-RNS conversion before being sent to the DAC arrays. These operations are all handled within the electronic chiplet. After DA conversion, the data is transmitted to the RNS-based Photonic Computing Unit (RNS-PCU) array for dot product computation. Element-wise multiplication is performed on the photonic chiplet. The results are collected from the photonic chiplet via photodetectors and TIA circuits located on the electronic chiplet. Subsequently, the analog signals are further converted into digital signals through the ADC arrays, and the resulting data are then passed into the adder tree circuits to complete the dot product computation. The output results undergo a final modulus operation before being converted from RNS to BNS. The converted data is then sent to subsequent circuits for further processing, such as performing nonlinear operations.

Data is loaded from off-chip DRAM into on-chip SRAM arrays for reading and writing. In ROCKET, three separate types of SRAM arrays are dedicated to storing activations, weights, and gradients, respectively, along with an additional specialized custom SRAM array designed for constructing fast lookup tables used for BNS-RNS conversions. These arrays, along with other digital circuits, are placed on the electronic chiplet. To match the frequency of the photonic compute path with the electronic compute path, we provide corresponding electronic compute units for the photonic path. For example, when the photonic path operates at a clock frequency of 4 GHz, the electronic compute units operate at 1 GHz. In this scenario, four sets of electronic compute units create four parallel data streams in each digital clock cycle, feeding data into the photonic compute core at a frequency of 4 GHz. Each PMMU is equipped with four dedicated SRAM sub-arrays for each type of SRAM. The same configuration is also applied to other digital circuits. This design ensures that memory access and digital computation are fast enough not to limit the performance of the photonic core.

4 Hybrid Pipeline-based Dataflow

4.1 Overview of Hybrid Pipeline-based Dataflow

ROCKET proposes a hybrid pipeline-based dataflow that considers the differing characteristics of photonic and electronic computing paths, as illustrated in Figure 6. The input and weight matrices are partitioned into smaller blocks, aligned with the sizes of the SRAM and PMMUs. If necessary, these matrices are flattened before partitioning. The partitioned data is divided into three parallel streams: the sign bit is sent to the XOR unit, the exponent parts to the exponent processing unit, and the mantissa parts to the BNS-to-RNS unit for conversion. The converted data must be synchronized in the Operand Synchronization (OS) unit. Once synchronization conditions are met, the data is sent to the DACs for conversion. After the photonic computation is completed, the data are sent to the valid data recognition unit to differentiate between valid data and noise. The valid data are read from the ADCs and then processed through the adder tree to compute partial or complete modular dot products in parallel. It is important to note that the mantissa results at this stage require a modulo operation. Subsequently, these results are transferred to the RNS-to-BNS conversion unit. If there is a need to accumulate partial dot products, this can be iteratively performed using the adder tree. The next unit involves applying a nonlinear function, such as ReLU or softmax. Given the complexity of these nonlinear functions, this stage may require additional clock cycles to compute the final result. Since the nonlinear computations are executed only once per vector dot product, these additional cycles are pipelined across all vector dot products within a DNN layer, adding only a few extra cycles to the last vector dot product. Stages 1-8 are repeated in a pipelined manner until the forward pass is complete. Input and weight gradients are then calculated in a similar manner.

Photonic computing lacks any form of memory or instructions to control the computational dataflow for complex real-world applications. For example, while an arithmetic logic unit (ALU) can simultaneously retrieve two elements from registers, accurately synchronizing input data in a photonic multiplication unit is a significant challenge. This is because the photonic multiplication unit lacks memory and immediately processes signals received from the DACs. Therefore, it is crucial to ensure that all necessary data are simultaneously available in the nearest level of storage; otherwise, any deviation will lead to incorrect computation results. Furthermore, due to the lack of effective data recognition logic in the photonic computing core's output, distinguishing between noise

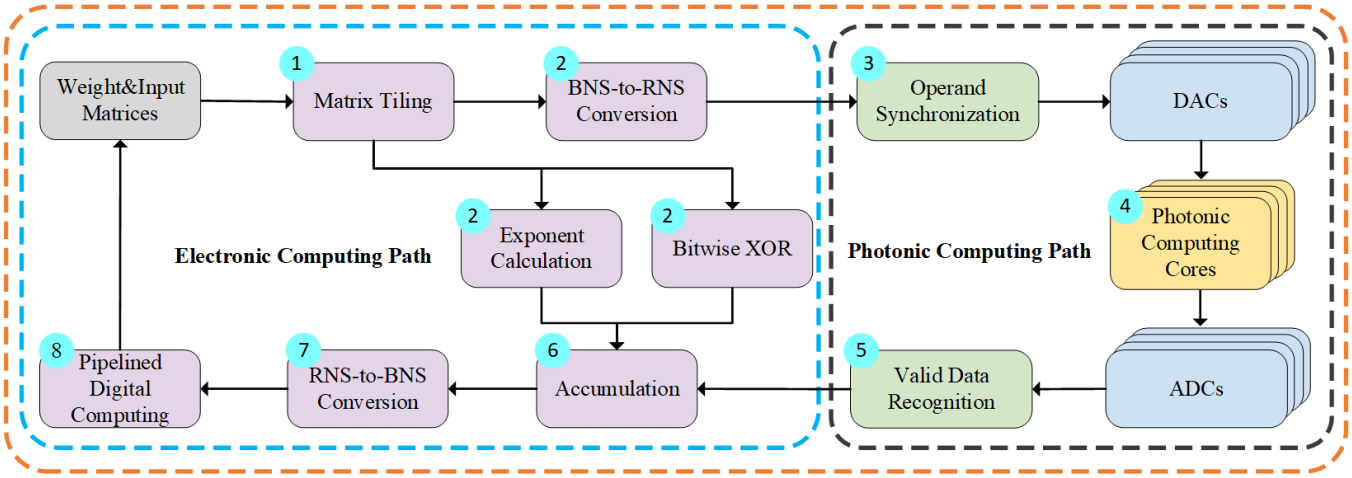

Figure 6: RNS-based hybrid dataflow for the forward pass.

and valid data becomes difficult. To address these challenges, we propose two new stages: operand synchronization and valid data recognition, which will be discussed in detail in Sections 4.2 and 4.3.

4.2 Photonic Operand Synchronization

The photonic operand synchronization unit is responsible for supplying synchronized parallel data streams to the DAC arrays. In the photonic computing core, precise alignment is required during the element-wise multiplication of two high-speed analog voltage streams, $\mathbf{a} = [a_i]$ and $\mathbf{b} = [b_i]$, to prevent computational errors. As illustrated in Fig. 1, if the analog voltage sequences provided by the two DACs are not properly synchronized, the multiplication result will deviate from the expected value (e.g., 0.3), producing an indeterminate erroneous output. Ensuring synchronization between the two analog voltage time series, which must be fed into two intensity modulators at high frequencies, presents significant challenges. For instance, at a frequency of 10 GHz, the time interval between consecutive voltage samples is merely 0.1 ns. When the required data resides across off-chip memory and SRAM, the mismatch in access latencies can lead to desynchronization of the DAC outputs, resulting in computational inaccuracies.

To address these challenges, valid flags (valid[]) are introduced in the OS unit, with each DAC maintaining its own data valid flag. When new data samples become ready for transmission, the corresponding valid flag is automatically set to 1. If no new data samples arrive after the currently valid data is transmitted, the flag is reset to 0. Importantly, during each digital clock cycle, the OS unit automatically monitors the data readiness status of all parallel AXI streams. Once all DACs have their valid flags asserted, the OS unit triggers the synchronized transmission of voltage streams into the photonic computing core.

4.3 Valid Data Recognition

When ROCKET has not yet transmitted the parallel voltage streams to the photonic computation core, the ADC array has already begun digital signal conversion. At this point, each ADC outputs multiple parallel data samples. The photonic computation core itself does not take responsibility for the accuracy of the computation results, and subsequent steps in the data stream continue execution regardless of whether the computation results are correct or not. In such cases, the system is prone to computational errors. Consider the following scenario: the ADC performs analog signal conversion at a frequency of 4.096 GS/s, and given that the clock frequency of the data path is 256 MHz, the ADC transmits 16 parallel samples to the data path approximately every 4 ns. There are three possible situations: the first is that valid data is obtained precisely at the start, meaning that the next 16 samples are all valid; the second is that valid data is obtained starting at a specific sample point, meaning that the 16 samples from that point onward are valid; the third is that all 16 parallel samples are noise, with no valid data present.

To address this challenge, a preamble sequence with a fixed pattern is added to each vector in the digital domain before transmitting the training data to the DACs. The preamble sequence is repeated N times. When the read data matches the detected number of preamble sequences, parallel samples are read to obtain the photonic multiplication results. It is evident that always adding a preamble sequence to each vector increases additional computational tasks. Therefore, we introduce a tail sequence, consisting of a fixed pattern. When the operand synchronization unit starts outputting

parallel voltage streams (i.e., all valid[] states are 1), and after some clock cycles the valid[] states are no longer all 1, indicating that the latest DNN training data is not ready, a separate tail sequence is added in the digital domain. There is no need to add a preamble sequence to each vector before detecting this tail sequence; direct reading of valid data is sufficient. When this tail sequence is detected, the unit re-detects the preamble sequence, waiting to read the next valid data.

5 Experimental Validation

5.1 Prototype Setup

5.1.1 Electronic Components. To validate the feasibility of the ROCKET accelerator, we implemented a system prototype on the Xilinx Zynq UltraScale+ RFSoC ZU28DR FPGA platform [65]. Figure 7 shows a photograph of the ROCKET validation prototype. The RTL implementation was written in Verilog [58] and simulated for verification using a testbench created in the Xilinx Vivado 2024.1 [3] integrated development environment (IDE). The RTL design of the ROCKET data path, along with the Xilinx RF data converter (DAC/ADC) IP and DDR4 DRAM IP, was synthesized and implemented in Vivado IDE to generate the bitstream, which was then executed on the ZCU111 board. The AXI 4 stream protocol [7] was responsible for data transfer between the FPGA programmable logic and the digital-analog conversion modules, while the AXI Lite protocol [2] facilitated the transmission of control signals and parameters between the embedded PetaLinux [4] and FPGA modules. The FPGA was configured to operate at a frequency of 256 MHz, with each DAC and ADC on the Xilinx XM500 RF board [65] configured to a data sampling rate of 4.096 GHz, resulting in 16 data samples per FPGA clock cycle. The DAC samples and ADC samples represent a 4-bit and 8-bit fixed-point number in the analog domain, respectively. It is important to note that when the photonic computing frequency changes, the dataflow of ROCKET does not require any modifications; only the AXI stream width and parallelism need to be adjusted accordingly.

To enhance SNR of the analog signal, the DAC is configured for differential output. To drive the optical modulator [60] in the prototype, a half-wave radio frequency (RF) voltage ($V_\pi = 5\,\text{V}$) is required. This is achieved by serially connecting two LMH5401EVM differential amplifiers [57] to amplify the signal. The amplified signal is subsequently converted to a single-ended output through the MAX4444EVKIT [6] to facilitate modulation by the optical modulator. Testing with the Moku Pro [25] shows that after amplification and aggregation, the differential RF output signal achieves approximately three times the gain. On the ADC side, a common-mode voltage ($V_{\text{cm}} = 1.25\,\text{V}$) is set to adjust the DC offset of the input signal, ensuring it falls within the ADC's input range. To apply the common-mode voltage V_{cm} and further amplify the analog signal, an LMH5401EVM amplifier is connected to the output of the photodetector.

5.2 Photonic Components

Our prototype utilizes a tunable continuous-wave laser source [1], which supports a high power output of 15 dBm and is set to a wavelength of 1550 nm. For the purpose of performing photonic multiplication, we utilize two 10 GHz Lithium Niobate intensity modulators [60]. To optimize the polarization state of the optical signal transmission, a polarization controller [59] is placed between the two modulators. To lock the optimal operating point of the intensity modulator, we use the MBC-SUPER bias controller [39] to automatically lock and monitor the modulator's output in real-time through the provided Graphical User Interface (GUI). At the output end, a 15 GHz photodetector, RXM15EF [61], is used to convert the optical signal into an analog electrical signal.

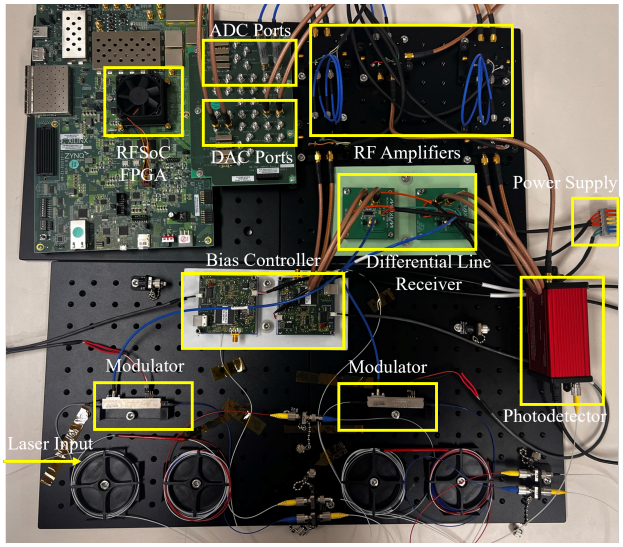

Figure 7: ROCKET validation prototype: integration of photonic and electronic components.

5.2.1 Memory Access. ROCKET is equipped with 4 GB of DDR4 memory, directly connected to the data path to support the training of DNN models. The DNN models requiring training are stored in its DRAM. To manage memory access, a DDR controller has been implemented within ROCKET's data path. The DDR4 memory is capable of achieving a data rate of approximately 170 Gbps [62]. This rate exceeds the aggregate data rate of the two DACs in the prototype responsible for converting DNN inputs and parameters, calculated as $2 \times 4.096\,\text{GS/s} \times 8\,\text{b/S} = 65.536\,\text{Gbps}$. To ensure the system

operates stably while handling high-frequency data streams and to avoid performance issues caused by data burstiness, we have implemented a back-pressure controlled AXI stream with a DRAM buffer. It should be noted that higher photonic computing frequencies or a greater number of DACs would necessitate an increase in DRAM interface bandwidth or the use of High Bandwidth Memory (HBM) [34] with multiple stacks to match bandwidth requirements.

5.3 Data Formats for DNN training

One of the key focuses of this work is to achieve high-precision DNN training. In the experiments, we implemented mixed-precision DNN training using BFloat16 and FP32, with FP32 primarily used for primary weight updates. Unless otherwise specified, the data formats used in the following sections remain consistent with this. The BFloat16 format includes 1 sign bit, 8 exponent bits, and 7 mantissa bits. When multiplying two BFloat16 numbers, the implicit leading 1 of the mantissa cannot be ignored. This implies that multiplying two 8-bit fixed-point numbers can produce up to a 16-bit fixed-point product. To ensure operations remain within the dynamic range, we select a modulus set of $\{11, 13, 15, 16, 17\}$, where M exceeds 2^{19}. Based on the mantissa range of the BFloat16 format and the modulus set, a fast lookup table structure is preprocessed to facilitate BNS-RNS conversions.

5.4 Photonic Multiplication Evaluation

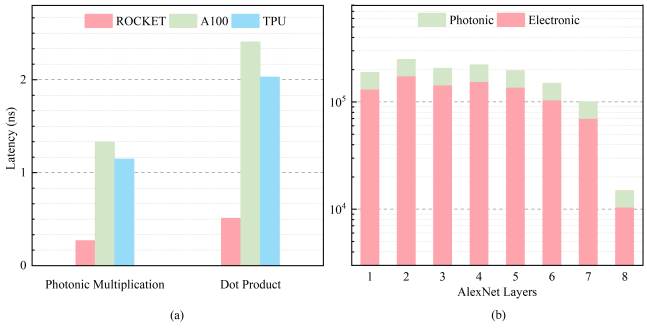

Figure 8: (a) Latency comparison for multiplication and dot product operations. (b) Latency comparison for photonic and electronic computing paths during AlexNet training.

To evaluate the speed of photonic multiplication and dot product operations on our ROCKET prototype, we randomly generated 1,000 pairs of numbers in the BFloat16 format. The computational latency reflects the time from the initiation of a computation request to the moment the result exits the system. Figure 8 (a) compares the computational latency of multiplication and dot product operations using ROCKET versus the A100 GPU and TPU. We observed that ROCKET achieves a latency that is 4.93 times and 4.25 times faster than the A100 and TPU, respectively, in multiplication calculations. Similarly, for dot product operations, ROCKET also demonstrates a latency advantage, being 4.72 times faster than the A100 and 3.98 times faster than the TPU. When estimating latency for a 64×64 GEMM operation, since all computing platforms employ the same tiling strategy and the subsequent computational procedures remain essentially identical to the dot product calculations, we extrapolate that ROCKET will maintain a similar speedup as observed in the dot product operations.

To analyze the impact of ROCKET's electronic computing path on the latency of photonic multiplication, we decomposed the DNN training latency results into two parts: the electronic computing path and the photonic computing path. As shown in Figure 8 (b), our statistics for the AlexNet model during the training of different layers indicate that the photonic computing path contributes, on average, approximately 30.42% of the total latency, representing a 39.16% reduction compared to the electronic computing path. Although the photonic part still contributes to some extent to the latency in multiplication tasks, its impact is significantly lower than that of the electronic part. Meanwhile, the overall performance improvement of the system is constrained by bottlenecks in the electronic part, including but not limited to memory, pipeline modules, and other electronic computation modules.

6 Large-scale Simulations

6.1 Simulation Setup

6.1.1 Performance Models. We incorporate device-level parameters from electronic [12, 34, 47] and photonic computing devices [31, 64, 66], as well as empirical measurements of hybrid dataflows on FPGA platforms, and develop a discrete-time event-driven simulator based on the PyTorch framework [40] to evaluate system-level performance during DNN training and inference. The hardware parameters of the baseline systems—including the A100 GPU, TPU v4, Mirage, and Lightning—are obtained from publicly available sources, as detailed in [16, 27, 38, 69]. In the electronic computation path, we used a First-In-First-Out (FIFO) queue-based pipeline for dataflow scheduling.

6.1.2 Area and Power Models. To evaluate the area and power consumption of ROCKET, we synthesized the RTL of ROCKET's electronic computation path using Cadence Genus synthesis software with a commercial 65 nm process library [55] to obtain the netlist data. To calculate the power consumption, we annotated the toggle rates of the digital gates using waveforms generated by the Vivado testbench.

Table 1: Electronic and Photonic Device Parameters.

	Devices	Unit area (mm^2)	Unit Power (W)
Electronic	Memory controller	0.129	0.0186
	Hybrid module	15.75	4.96
	Conversion module	0.02	0.0006
	HBM [34]	81.1	7.41
	DAC [47]	0.043	0.0091
	ADC [12]	0.013	0.0094
	Nonlinear module	0.035	0.038
Photonic	Modulator [64]	0.165	
	Photodetector [31]	3.2E-5	3.8E-3
	Laser [66]	0.01	

Table 2: Validation accuracy compared to various data formats.

Models	ROCKET	Mirage	FP32	Bfloat16	INT8
AlexNet	56.75	55.64	56.77	56.69	51.26
ResNet18	75.10	75.17	75.12	71.01	66.25
VGG16	69.89	69.02	69.90	69.82	64.45
VGG19	71.52	70.54	71.55	71.50	66.49
GPT-2	41.26	39.41	41.29	41.15	36.21
DLRM	78.55	76.41	78.59	78.50	72.98
BERT	69.95	67.98	69.98	69.90	64.30

Using resource consumption data from the electronic computing path, we derived the area and power consumption for the electronic chip. Furthermore, by employing scaling equations provided in [53], these parameters were adjusted to 7 nm technology. Additionally, photonics device parameters obtained from prior work [69] allowed us to derive the total area and power consumption of ROCKET. Based on the above model, we can obtain the key photonic and electronic parameters used in large-scale simulations, as shown in Table 1. Here, the hybrid module refers to the photonic operand synchronization and valid data recognition unit, the conversion module refers to the dedicated SRAM for BNS-RNS conversion, and the nonlinear module refers to the nonlinear logic computation unit.

6.1.3 DNN Models. Seven mainstream DNN models are evaluated. Among them, the vision models AlexNet [28], ResNet18 [22], VGG16 [14], and VGG19 [51] were trained on the ImageNet dataset [17] with a batch size set to 256. The natural language processing model GPT-2 [44] was trained using the WebText training set [44]. The recommendation system model DLRM [37] was trained on the Criteo dataset [11]. Additionally, the question-answering system model BERT [18] was trained on the SQuAD v1.1 dataset [45].

6.1.4 Accelerators for Comparison. We compared ROCKET with the electronic accelerators A100 GPU [38], TPU v4 [27], and the state-of-the-art photonic DNN training accelerator, Mirage [16]. Among them, the A100 GPU serves as the representative of general-purpose accelerators, the TPU v4 as the representative of contemporaneous application-specific integrated circuit (ASIC) accelerators, and Mirage as the representative of the latest photonic accelerators for DNN training. ROCKET operates at a frequency of 100 GHz and can perform 2,250 photonic multiplications per clock cycle. Additionally, we compared ROCKET with the previously proposed state-of-the-art photonic DNN inference accelerator, Lightning [69].

6.2 Simulations Results

6.2.1 Validation Accuracy. We evaluated the validation accuracy of ROCKET across different DNN models, as shown in Table 2. To ensure a fair comparison, the same training parameters were used in all experiments. It can be seen that ROCKET consistently delivers validation accuracy comparable to FP32 training. Furthermore, ROCKET demonstrates higher validation accuracy compared to Mirage.

6.3 Power and Area

Table 3 compares the energy consumption per Multiply-Accumulate (MAC) operation of the RNS-PCU in the ROCKET accelerator with different accelerator platforms. To intuitively compare the advantages of photonic computing, the energy consumption per MAC operation reported here refers specifically to the RNS-PCU unit. It can be seen that ROCKET features the highest clock frequency at 100 GHz and the lowest energy consumption per MAC operation. While it is evident that the Mirage chip surpasses GPUs and TPUs in terms of computational speed, it falls short in energy management compared to these processors.

Figure 9 shows the peak power and area breakdown for ROCKET. It can be seen that SRAM accesses contribute the majority of power consumption (66.2%) in ROCKET. This is because DNN training using BFloat16 + FP32 requires frequent SRAM operations. With the development of more low-precision DNN training methods, there is potential to reduce the overall data storage requirements and the energy consumption per SRAM access. Notably, in our design, data converters only consume 6.2% of the total power—contrary to the typical situation in analog accelerators where data converter power dominates. This is mainly due to the reduced bit precision of DACs/ADCs, leading to an exponential decrease in their power consumption. The reduction in bit precision also decreases the required SNR during analog operations, while laser power shows a certain decrease, although optical losses prevent an exponential reduction in laser power.

Table 3: Energy consumption comparison of per MAC operation on different accelerators

Accelerators	ROCKET	Mirage	A100	TPU
pJ/MAC	7.9 (fJ)	0.21	0.07	0.07
$f(Hz)$	100G	10G	1.41G	1.05G

Additionally, the use of multiple moduli increases the component count, leading to higher power consumption in other components (SRAM arrays, TIAs, accumulators, etc.), which significantly reduces the relative contribution of data converter power.

Figure 9 (b) shows that most of the area is occupied by photonic devices and SRAM. All components together occupy 794.3 mm^2, with the photonic chiplet occupying 392.16 mm^2 and the electronic chiplet occupying 402.14 mm^2. Since the photonic and electronic chiplets are stacked via 3D integration, the total area can be considered as the larger of the two chiplets (402.14 mm^2). The ROCKET chip exhibits a larger footprint compared to the Mirage chip, being 1.65 times larger. In contrast, it is smaller when compared to the A100 chip (826 mm^2) and the TPU v4 chip (600 mm^2), with the ROCKET chip being approximately 2 times smaller than the A100 and 1.49 times smaller than the TPU v4.

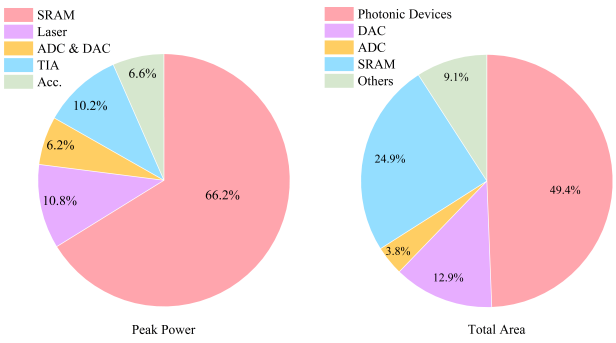

Figure 9: The peak power consumption and area breakdown for ROCKET photonic accelerator.

6.4 Performance of DNN Training

Figure 10 illustrates the average speedup and energy savings achieved by ROCKET in training various DNN models compared to Mirage, A100 GPU, and TPU v4. Notably, the Mirage accelerator requires the use of BFP format for DNN training, consistent with parameters described in [16]. For the DNN training, we considered the computational times in both the electronic computing path and the photonic computing path. Similarly, the total system energy consumption includes energy expenditures from both these paths. The results indicate that ROCKET achieves a speedup in DNN training times by factors of 33×, 243×, and 198×, and energy savings by factors of 64×, 204×, and 142× compared to Mirage, A100 GPU, and TPU v4, respectively.

Figure 10: Speedup and energy saving compared to state-of-the-art accelerators.

6.5 ROCKET as an DNN Inference Accelerator

This paper focuses on DNN training; however, since inference is a subset of training, ROCKET can also be used to accelerate DNN inference. We compared ROCKET with the state-of-the-art photonic DNN inference accelerator Lightning, as shown in Table 4. For fairness, we used 8-bit quantized integers uniformly during inference. The results indicate that ROCKET improves the average inference time by 5.28× and achieves a 1.83× improvement in average energy consumption compared to Lightning. This is because Lightning requires additional graph construction time. Furthermore, Lightning needs to identify preambles after each photonic computation.

6.6 Noise and Process Variation Management in ROCKET

In photonic computing cores, computational errors may arise due to shot noise, thermal noise, optical path losses, and process variations. As discussed in Section 2.2.2, increasing the optical input power is a common approach to enhance the SNR in order to mitigate noise interference and achieve the desired computational precision. However, during prolonged system operation, higher input power may cause device heating, which can induce phase drift, resonance wavelength shifts, and other effects that degrade the computational accuracy and system stability. To achieve efficient thermal

Table 4: Speedup and energy savings compared to state-of-the-art photonic accelerator for DNN Inference.

DNN	AlexNet	ResNet18	VGG16	VGG19	GPT-2	DLRM	BERT
Speedup	5.41×	5.30×	5.38×	5.23×	5.26×	5.17×	5.25×
Energy Savings	1.92×	1.84×	1.87×	1.80×	1.78×	1.84×	1.82×

management, we adopt the closed-loop thermal feedback control system proposed in [56] to dynamically stabilize system performance. In addition, process variations may cause V_π drift and bias point drift in Mach–Zehnder MZIs, leading to computational errors. Several methods have been proposed to minimize or calibrate these errors, including design optimization approaches [35], self-calibration techniques [9, 21], and post-fabrication trimming methods [26]. Given that the underlying sources of these errors are largely architecture-independent, these approaches are equally applicable to ROCKET and other photonic hardware systems.

7 Related Work

Most existing photonic DNN accelerators primarily focus on enhancing the efficiency of the DNN inference, as the DNN training requires high-precision computation, particularly for gradient calculations that demand a relatively high dynamic range. These accelerators are based on MRR [33, 43], Mach-Zehnder interferometers (MZI) [8, 48], and hybrid implementations of MRR and MZI [49, 50]. Mirage adopts a hybrid MRR and MZI design, achieving high-performance DNN training under RNS [16]. For any modulus m, each multiplication unit in Mirage contains $\lceil \log_2 m \rceil$ phase shifters and $2\lceil \log_2 m \rceil$ MRRs. During the MAC process, to support modular reduction by a modulus m, the 2π phase range must be mapped to $2\pi/m$ at each phase shifter. To ensure computational integrity, power consumption must be increased to meet the required SNR. Under the same per-MAC energy budget, Mirage consumes 17.2 times more power compared to the systolic array when performing 4-bit mantissa calculations in the BFP format [68]. On the contrary, ROCKET requires only two intensity modulators per photonic multiplication operation. Additionally, through the design of the photonic accelerator, we minimize the use of computing components (such as DACs and MZMs) and maximize data reuse. This approach enables us to support high-precision numerical computations, such as BFloat16, at low power levels. The experimental results indicate that, compared to Mirage, ROCKET achieves a speedup of 33× in DNN training time and an energy savings of 64×.

In addition, most existing research has focused on photonic hardware performance and architectural design, with little attention paid to the study of dataflow. Previous research has shown that when dataflow stalls occur, latency can increase by five orders of magnitude or more [69]. For example, some works [19, 52] employ a stop-and-go method where the photonic core remains idle during the execution of digital path computations until the next layer of computation is initiated, significantly limiting the advantages of photonic computing. Lightning proposes a count-action-based Directed Acyclic Graph (DAG) scheduling method for DNN inference [69]. This scheduling method requires an additional DAG construction process, which needs to be reconstructed whenever the inference task size changes, greatly increasing execution time. For DNN training, the DAG must be rebuilt whenever the training set or batch size changes. Our work designs a dataflow based on a hybrid photonic-electronic pipeline, eliminating the need for time-consuming composition processes. Experimental results indicate that ROCKET achieves a 5.2× improvement in average inference time and a 1.8× increase in average energy consumption.

8 Conclusions

To achieve photonic computing tailored for high-precision DNN training, we propose ROCKET, an RNS-based photonic accelerator. To enable efficient photonic computation under RNS, we propose a novel photonic accelerator architecture and design a dataflow based on a hybrid pipeline. Finally, we build a high frequency 4.096 GHz hybrid photonic-electronic prototype using FPGA, RF devices, and photonic components, demonstrating the feasibility and potential of advanced photonic computing. Large-scale simulations show that compared to Mirage, A100 GPU, and TPU v4, ROCKET achieves speedups of 33×, 243×, and 198×, and energy savings of 64×, 204×, and 142×, respectively, across seven mainstream DNN models.

Acknowledgments

We are sincerely grateful to all the anonymous reviewers for their valuable feedback and constructive comments. Additionally, we extend our sincere appreciation to the Pawsey Supercomputing Centre for its support of the photonic accelerator project.

References

[1] 2024. IQTLS Tunable Laser Source. https://photonicsolutioncenter.com/products/iqtls-tunable-laser-source/.

[2] Advanced eXtensible Interface. 2022. Advanced eXtensible Interface. https://en.wikipedia.org/wiki/Advanced_eXtensible_Interface Accessed: 2024-08-15.

[3] Inc. Advanced Micro Devices. 2024. AMD Vivado Design Suite. https://www.xilinx.com/products/design-tools/vivado.html.

[4] Inc. Advanced Micro Devices. 2024. Petalinux Tool. https://www.xilinx.com/products/design-tools/embedded-software/petalinux-sdk.html Accessed: 2024-08-15.

[5] Kosmas Alexandridis and Giorgos Dimitrakopoulos. 2024. Online Alignment and Addition in Multiterm Floating-Point Adders. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2024).

[6] Inc. Analog Devices. 2024. MAX4444-MAX4445: Ultra-High-Speed, Low-Distortion, Differential-to-Single-Ended Line Receivers with Enable Data Sheet. https://www.analog.com/media/en/technical-documentation/datasheets/MAX4444-MAX4445.pdf.

[7] ARM. 2022. AMBA® AXI-Stream Protocol Specification. https://developer.arm.com/documentation/ihi0051/a/Interface-Signals/Transfer-signaling/Handshake-process Accessed: 2024-08-15.

[8] Hengameh Bagherian, Scott Skirlo, Yichen Shen, Huaiyu Meng, Vladimir Ceperic, and Marin Soljacic. 2018. On-chip optical convolutional neural networks. *arXiv preprint arXiv:1808.03303* (2018).

[9] Saumil Bandyopadhyay, Ryan Hamerly, and Dirk Englund. 2021. Hardware error correction for programmable photonics. *Optica* 8, 10 (2021), 1247–1255.

[10] Viraj Bangari, Bicky A Marquez, Heidi Miller, Alexander N Tait, Mitchell A Nahmias, Thomas Ferreira De Lima, Hsuan-Tung Peng, Paul R Prucnal, and Bhavin J Shastri. 2019. Digital electronics and analog photonics for convolutional neural networks (DEAP-CNNs). *IEEE Journal of Selected Topics in Quantum Electronics* 26, 1 (2019), 1–13.

[11] BARS. 2022. Criteo dataset bars x4 split. https://github.com/openbenchmark/BARS/tree/main/datasets/Criteo#criteo_x4. Accessed: 2024-08-16.

[12] Gregory Cooke, Naftali Weiss, Peter Schvan, Pascal Chevalier, Andreia Cathelin, and Sorin P Voinigescu. 2022. Track and hold amplifier investigation for 100-GHz bandwidth, 200-GS/s ADC front ends. *IEEE Solid-State Circuits Letters* 5 (2022), 54–57.

[13] Dharanidhar Dang, Bill Lin, and Debashis Sahoo. 2022. LiteCON: An all-photonic neuromorphic accelerator for energy-efficient deep learning. *ACM Transactions on Architecture and Code Optimization (TACO)* 19, 3 (2022), 1–22.

[14] Abhipraya Kumar Dash. n.d.. VGG-16 Architecture. https://iq.opengenus.org/vgg16/. Accessed: 2024-08-16.

[15] Cansu Demirkiran, Furkan Eris, Gongyu Wang, Jonathan Elmhurst, Nick Moore, Nicholas C Harris, Ayon Basumallik, Vijay Janapa Reddi, Ajay Joshi, and Darius Bunandar. 2023. An electro-photonic system for accelerating deep neural networks. *ACM Journal on Emerging Technologies in Computing Systems* 19, 4 (2023), 1–31.

[16] Cansu Demirkiran, Guowei Yang, Darius Bunandar, and Ajay Joshi. 2024. Mirage: An RNS-Based Photonic Accelerator for DNN Training. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 73–87.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[18] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[19] Johannes Feldmann, Nathan Youngblood, Maxim Karpov, Helge Gehring, Xuan Li, Maik Stappers, Manuel Le Gallo, Xin Fu, Anton Lukashchuk, Arslan S Raja, et al. 2021. Parallel convolutional processing using an integrated photonic tensor core. *Nature* 589, 7840 (2021), 52–58.

[20] Dusan Gostimirovic and Winnie N Ye. 2017. Ultracompact CMOS-compatible optical logic using carrier depletion in microdisk resonators. *Scientific reports* 7, 1 (2017), 12603.

[21] Ryan Hamerly, Saumil Bandyopadhyay, and Dirk Englund. 2022. Stability of self-configuring large multiport interferometers. *Physical Review Applied* 18, 2 (2022), 024018.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[23] Ahmad Hiasat. 2019. A residue-to-binary converter with an adjustable structure for an extended RNS three-moduli set. *Journal of Circuits, Systems and Computers* 28, 08 (2019), 1950126.

[24] Anton Iliev and Nikolay Kyurkchiev. 2018. The faster extended Euclidean algorithm. In *Collection of scientific works from conference*. 21–26.

[25] Liquid Instruments. 2024. MokuPro. https://www.liquidinstruments.com/products/hardware-platforms/mokupro/ Accessed: 2024-08-15.

[26] Hasitha Jayatilleka, Harel Frish, Ranjeet Kumar, John Heck, Chaoxuan Ma, Meer N Sakib, Duanni Huang, and Haisheng Rong. 2021. Post-fabrication trimming of silicon photonic ring resonators at wafer-scale. *Journal of Lightwave Technology* 39, 15 (2021), 5083–5088.

[27] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, et al. 2023. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*. 1–14.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[29] Tao Li, Jie Hou, Jinli Yan, Rulin Liu, Hui Yang, and Zhigang Sun. 2020. Chiplet heterogeneous integration technology—Status and challenges. *Electronics* 9, 4 (2020), 670.

[30] Weichen Liu, Wenyang Liu, Yichen Ye, Qian Lou, Yiyuan Xie, and Lei Jiang. 2019. Holylight: A nanophotonic accelerator for deep learning in data centers. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1483–1488.

[31] Dennis Maes, Luis Reis, Stijn Poelman, Ewoud Vissers, Vanessa Avramovic, Mohammed Zaknoune, Gunther Roelkens, Sam Lemey, Emilien Peytavit, and Bart Kuyken. 2022. High-speed photodiodes on silicon nitride with a bandwidth beyond 100 Ghz. In *CLEO: Science and Innovations*. Optica Publishing Group, SM3K–3.

[32] Sanu K Mathew, Mark A Anders, Brad Bloechel, Trang Nguyen, Ram K Krishnamurthy, and Shekhar Borkar. 2005. A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS. *IEEE Journal of Solid-State Circuits* 40, 1 (2005), 44–51.

[33] Armin Mehrabian, Yousra Al-Kabani, Volker J Sorger, and Tarek El-Ghazawi. 2018. PCNNA: A photonic convolutional neural network accelerator. In *2018 31st IEEE International System-on-Chip Conference (SOCC)*. IEEE, 169–173.

[34] Micron Technology, Inc. 2023. HBM3e: High Bandwidth Memory. https://www.micron.com/products/memory/hbm/hbm3e. Accessed: 2023-08-13.

[35] Asif Mirza, Amin Shafiee, Sanmitra Banerjee, Krishnendu Chakrabarty, Sudeep Pasricha, and Mahdi Nikdast. 2022. Characterization and optimization of coherent MZI-based nanophotonic neural networks under fabrication non-uniformity. *IEEE Transactions on Nanotechnology* 21 (2022), 763–771.

[36] Boris Murmann. 2020. Mixed-signal computing for deep neural network inference. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29, 1 (2020), 3–13.

[37] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. 2019. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091* (2019).

[38] Inc. NVIDIA. 2021. NVIDIA A100 Tensor Core GPU. https://www.nvidia.com/en-au/data-center/a100/

[39] Inc. Oz Optics. 2024. Super Modulator Bias Controller. https://www.ozoptics.com/ALLNEW_PDF/DTS0165.pdf.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[41] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350* (2021).

[42] Dingyi Pei, Arto Salomaa, and Cunsheng Ding. 1996. *Chinese remainder theorem: applications in computing, coding, cryptography*. World Scientific.

[43] Jiaxin Peng, Yousra Alkabani, Shuai Sun, Volker J Sorger, and Tarek El-Ghazawi. 2020. Dnnara: A deep neural network accelerator using residue arithmetic and integrated photonics. In *Proceedings of the 49th International Conference on Parallel Processing*. 1–11.

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[45] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[46] Akshay Krishna Ramanathan, Gurpreet S Kalsi, Srivatsa Srinivasa, Tarun Makesh Chandran, Kamlesh R Pillai, Om J Omer, Vijaykrishnan Narayanan, and Sreenivas Subramoney. 2020. Look-up table based energy efficient processing in cache support for neural network acceleration. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 88–101.

[47] Hannes Ramon, Michiel Verplaetse, Michael Vanhoecke, Haolin Li, Johan Bauwelinck, Peter Ossieur, Xin Yin, and Guy Torfs. 2021. A 100-GS/s four-to-one analog time interleaver in 55-nm SiGe BiCMOS. *IEEE Journal of Solid-State Circuits* 56, 8 (2021), 2539–2549.

[48] Yichen Shen, Nicholas C Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, et al. 2017. Deep learning with coherent nanophotonic circuits. *Nature photonics* 11, 7 (2017), 441–446.

[49] Kyle Shiflett, Avinash Karanth, Razvan Bunescu, and Ahmed Louri. 2021. Albireo: Energy-efficient acceleration of convolutional neural networks via silicon photonics. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 860–873.

[50] Kyle Shiflett, Dylan Wright, Avinash Karanth, and Ahmed Louri. 2020. Pixel: Photonic neural network accelerator. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 474–487.

[51] Karen Simonyan. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[52] Alexander Sludds, Saumil Bandyopadhyay, Zaijun Chen, Zhizhen Zhong, Jared Cochrane, Liane Bernstein, Darius Bunandar, P Ben Dixon, Scott A Hamilton, Matthew Streshinsky, et al. 2022. Delocalized photonic deep learning on the internet's edge. *Science* 378, 6617 (2022), 270–276.

[53] Aaron Stillmaker and Bevan Baas. 2017. Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm. *Integration* 58 (2017), 74–81.

[54] Febin Sunny, Asif Mirza, Mahdi Nikdast, and Sudeep Pasricha. 2021. CrossLight: A cross-layer optimized silicon photonic neural network accelerator. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1069–1074.

[55] Cadence Design Systems. [n. d.]. Genus Synthesis Solution. https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html. Accessed: 2024-08-16.

[56] Min Tan, Kaixuan Ye, Da Ming, Yuhang Wang, and Junbo Feng. 2021. Towards electronic-photonic-converged thermo-optic feedback tuning. *Journal of Semiconductors* 42, 2 (2021), 023104.

[57] Inc. Texas Instruments. 2024. LMH5401 Evaluation Module. https://www.ti.com/tool/LMH5401EVM.

[58] Donald Thomas and Philip Moorby. 2008. *The Verilog® hardware description language*. Springer Science & Business Media.

[59] Inc. Thorlabs. 2024. FPC562 - Fiber Polarization Controller. https://www.thorlabs.com/thorproduct.cfm?partnumber=FPC562.

[60] Inc. Thorlabs. 2024. LNA2322 - 10 GHz Intensity Modulator with Internal Photodetector. https://www.thorlabs.com/thorproduct.cfm?partnumber=LNA2322.

[61] Inc. Thorlabs. 2024. RXM15EF - Multimode Ultrafast Receiver. https://www.thorlabs.com/thorproduct.cfm?partnumber=RXM15EF.

[62] Inc. Transcend Information. 2024. FAQ - Transcend Support. https://www.transcend-info.com/Support/FAQ-292.

[63] Sairam Sri Vatsavai and Ishan G Thakkar. 2022. Photonic reconfigurable accelerators for efficient inference of cnns with mixed-sized tensors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 11 (2022), 4337–4348.

[64] Stefan Wolf, Heiner Zwickel, Wladislaw Hartmann, Matthias Lauermann, Yasar Kutuvantavida, Clemens Kieninger, Lars Altenhain, Rolf Schmid, Jingdong Luo, Alex K-Y Jen, et al. 2018. Silicon-organic hybrid (SOH) Mach-Zehnder modulators for 100 Gbit/s on-off keying. *Scientific reports* 8, 1 (2018), 1–13.

[65] Xilinx. 2023. ZCU111 Evaluation Kit. https://www.xilinx.com/products/boards-and-kits/zcu111.html Accessed: 2024-08-15.

[66] Xiaoxiao Xue, Pei-Hsun Wang, Yi Xuan, Minghao Qi, and Andrew M Weiner. 2017. Microresonator Kerr frequency combs with high conversion efficiency. *Laser & Photonics Reviews* 11, 1 (2017), 1600276.

[67] Zhoufeng Ying, Chenghao Feng, Zheng Zhao, Shounak Dhar, Hamed Dalir, Jiaqi Gu, Yue Cheng, Richard Soref, David Z Pan, and Ray T Chen. 2020. Electronic-photonic arithmetic logic unit for high-speed computing. *Nature communications* 11, 1 (2020), 2154.

[68] Sai Qian Zhang, Bradley McDanel, and HT Kung. 2022. Fast: Dnn training under variable precision block floating point with stochastic rounding. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 846–860.

[69] Zhizhen Zhong, Mingran Yang, Jay Lang, Christian Williams, Liam Kronman, Alexander Sludds, Homa Esfahanizadeh, Dirk Englund, and Manya Ghobadi. 2023. Lightning: A reconfigurable photonic-electronic smartnic for fast and energy-efficient inference. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 452–472.