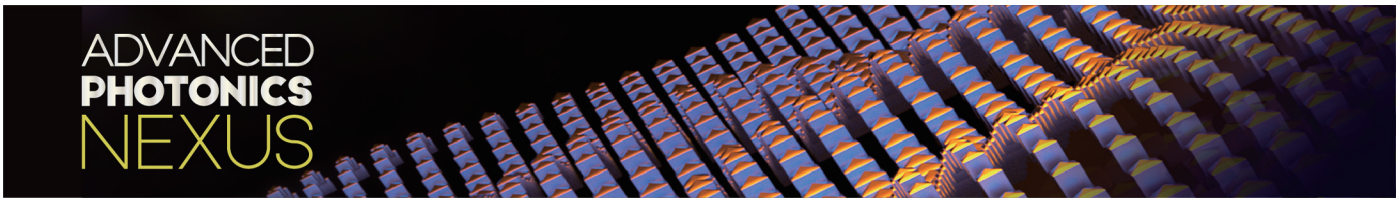
Research Article



Joint device architecture algorithm codesign of the photonic neural processing unit

Li Pei,[a] Zeya Xi,[a] Bing Bai,[a,b,*] Jianshuai Wang,[a] Jingjing Zheng,[a] Jing Li,[a] and Tigang Ning[a]
[a]Beijing Jiaotong University, Institute of Lightwave Technology, Key Lab of All Optical Network & Advanced Telecommunication Network of EMC, Beijing, China
[b]Photoncounts (Beijing) Technology Co. Ltd., Beijing, China

Abstract. The photonic neural processing unit (PNPU) demonstrates ultrahigh inference speed with low energy consumption, and it has become a promising hardware artificial intelligence (AI) accelerator. However, the nonidealities of the photonic device and the peripheral circuit make the practical application much more complex. Rather than optimizing the photonic device, the architecture, and the algorithm individually, a joint device-architecture-algorithm codesign method is proposed to improve the accuracy, efficiency and robustness of the PNPU. First, a full-flow simulator for the PNPU is developed from the back end simulator to the high-level training framework; Second, the full system architecture and the complete photonic chip design enable the simulator to closely model the real system; Third, the nonidealities of the photonic chip are evaluated for the PNPU design. The average test accuracy exceeds 98%, and the computing power exceeds 100TOPS.

Keywords: optics; photonics; Mach–Zehnder interferometer array; photonic neural processing unit design.

Received Jul. 4, 2022; revised manuscript received Feb. 13, 2023; accepted for publication Apr. 6, 2023; published online Jun. 9, 2023.

[DOI: 10.1117/1.APN.2.3.036014]

1 Introduction

In recent years, deep neural networks (DNNs) have involved large amounts of links and large data capacity, enabling the high quality of data processing. DNNs have been applied in image analysis, video tracking, language translation,[1,2] etc. They are mostly deployed in electronic hardware, including the central processing unit, graphical processing unit, and tensor processor unit. However, according to the Moore's law, the electronic devices are insufficient to cope with the explosive information crisis, due to limited complementary metal-oxide semiconductor (CMOS) fabrication techniques.[3] The linearity and nonlinearity of integrated optical devices for analog signal processing can greatly improve the performance and power efficiency of these artificial intelligence (AI) workloads.[4] The calculation core of the photonic neural processing unit (PNPU) is to complete matrix multiplication in a short time. That breaks the bottlenecks caused by system electronics parts. Matrix computation belongs to basic information processing. The PNPU aims to accelerate the calculation in the optical field to meet the growing demand for computing resources and capacity.[5]

A PNPU with consisting of a Mach–Zehnder interferometer (MZI) unit has demonstrated great progress in accelerating the DNN applications, with over 100 GHz photodetection rates and near-zero energy dissipation.[3] Due to the addition of photonic elements, the PNPU has higher data processing speed than an electronic processing unit, and the amount of data that can be processed is greatly improved.[6,7] The photonic characteristics allow PNPU with greater bandwidth and lower power consumption.[8] The PNPU provide some performance advantages for neural network computing, such as shape factor, manufacturability, cost, mechanical stability, and high-speed modulation.[9]

However, the PNPU suffers a challenge in robustness due to the nonideal effects of the MZI unit, which is similar to other neuromorphic systems.[4,6] The nonideal effects include the phase shift produced by the MZI unit's low voltage control resolutions[7,10] and device-level noise on the MZIs caused by the manufacturing imperfection and environment.[11] Meanwhile, the small data set-based fully connected neural networks cannot reveal the accuracy loss due to the nonideal effects. The general

*Address all correspondence to Bing Bai, baibing@bjtu.edu.cn

neural processing unit (NPU) is much more complex than the fixed application NPU accelerator, and it should be configured to apply for different networks and applications. It is highly demanded to develop a methodology to evaluate PNPU with non-ideal effects. The second section of the article mainly introduces the theoretical model, including the calculation theory of MZI and DNN. The third section shows the current chip architecture, including the overall architecture and optical chip architecture. Then based on the chip, the framework of optoelectronic hybrid computing was constructed, including the simulator. Finally, the application testing was demonstrated.

2 Theoretical model

Using programmable MZI array, simple neural network functions can be realized.

2.1 MZI Unit

Commonly, MZI is used in optical circuits, and it usually completes the electrical or thermal modulation of photonic signals. The schematic diagram of a programmable MZI is shown in Fig. 1(a), including two 50:50 evanescent directional couplers and two phase shifters (ϑ and φ). When using it as optical transformation, it can be described by a 2×2 unitary matrix U (2) matrix.[12]

Any unitary transformation can be decomposed into a group of U (2) matrix operations using cascading programmable MZI.[13]

When the multilayer operation is to be carried out, as shown in Fig. 1(b), it realizes matrix multiplication completely optically. In each layer, the input optical signal first undergoes a linear matrix multiplication of a linear unit. The transmitted data are transmitted to the computer layer, and then feed forward and backpropagation algorithms can be trained. The weighted matrix is replaced by $(\vartheta_{ij}, \varphi_{ij})$ of each MZI and optimized by calculating the gradient of the loss function. Each layer is composed of an optical interference unit (OIU) and a nonlinear unit (ONU). As shown in Fig. 2, $f(x)$ in the ONU is a nonlinear function. The main function of OIU is to complete matrix multiplication. With singular value decomposition (SVD) method, the matrix M can be decomposed into

$$M = U\Sigma V^*, \tag{1}$$

where U is the matrix of $m \times m$, $\sum$ is the matrix of $m \times n$, and $V*$ is the conjugate of the matrix of $n \times n$. U, $\sum$, and $V*$ can be realized by the MZI array. In this way, the multiplication of the optical matrix can be accomplished by the MZI array.

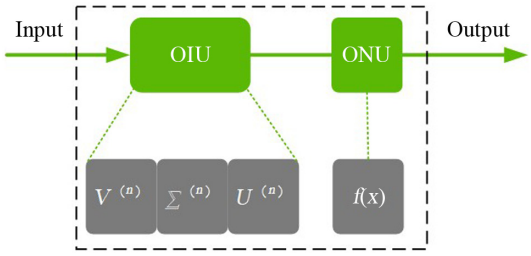


Fig. 2 Single-layer optical interference and nonlinear element on artificial neural network.

As this section has discussed, the optical MZI unit in Fig. 1(b) performs a matrix–vector product. The different sets of vectors can perform a general matrix-matrix product (GEMM) sequentially, which is a key function in the basic linear algebra subprograms (BLAS).[14]

2.2 DNN with MZI Unit

Most of DNN algorithms consist of convolutional (CONV) layers and full connected layers, and they are back-to-back connected and run sequentially. The computer-intensive operations of the CONV layers and fully connected layers are CONV and vector–matrix multiplication.

There are two ways to implement the CONV operation with the MZI unit:

1. Software patching to transform the CONV to the GEMM:

The CONV operation can be transformed to the GEMM operation by performing a "patching" technique, and the GEMM can be naturally operated in the optical MZI unit. The expression of the CONV layer is

$$y_{ij;k} = \sum_{i'j',l} K_{i'j',kl} x_{(s_x i+i')(s_y j+j');l}. \tag{2}$$

Here, the input is x_{ij}. K_{ij} is one pixel of the $W \times H$ feature map with C channels. kl is one element of the $Kx \times Ky \times C'$ convolution kernel with C channels. k is an element's value of the output feature map to next layer. The s_x and s_y are the strides of the convolution.

As shown in Fig. 3, the 'patching' technique works transform the CONV operation to the GEMM operation. After the kernel strides over the input image with the fixed step, we can get a

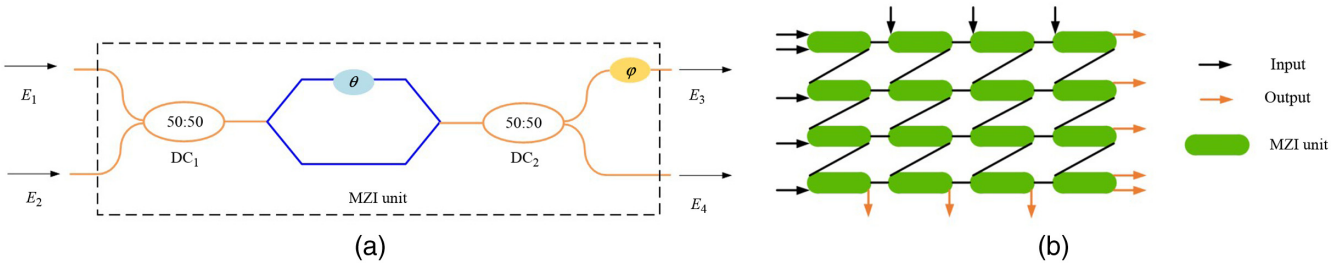


Fig. 1 (a) Schematic diagram of programmable MZI. (b) Silicon photonic neural network based on MZI array with eight input ports and eight output ports.

bunch of the overlapped matrices, called "patches." The patches are then rearranged to express a matrix X with the size of $(Kx \times Ky \times C) \times (W' \times H')$. The 4D kernel matrix is rearranged to form a 2D matrix K with the size of $(Kx \times Ky \times C) \times C'$. It can be computed by performing the matrix product of the $Y = K * X$. The CONV operation is transformed to the GEMM operation after the patching technique.

2. Hardware configurable delay chain (CDC):

The other method for transforming the CONV to the GEMM is adding a CDC and an adder in the output of the MZI units. An example of 2D 3×3 CONV with stride of 1 is shown in Fig. 4. The vector–matrix multiply expression can be expanded to be $X * K_{3\times3} = (x_1 * k_1, x_1 * k_2, x_1 * k_3)$. Here the X is the input vector; and k_1, k_2, and k_3 are the column vectors of the $K_{3\times3}$ matrix. Each row of the input feature map can be represented as $X = (x_1, x_2, x_3, x_4, \ldots)$. And each row of the 3×3 CONV output is

$$\begin{cases} O_1 = x_1 * k_1 + x_2 * k_2 + x_3 * k_{11} \\ O_2 = x_2 * k_1 + x_3 * k_2 + x_4 * k_{11} \\ O_3 = x_3 * k_1 + x_4 * k_2 + x_5 * k_{11}. \end{cases} \tag{3}$$

A CDC and an adder are employed to complete the above transformation in Fig. 4. Every cycle's output of adder corresponds to Eq. (3). The CDC can be configured to have different latency to support different stride sizes of the CONV. Different "add" channels of the feature map can be operated in the different MZI units in parallel. The second method is used in this study. The details will be shown in the next section.

2.3 Challenge

Due to the underlying nonideal characteristics of photonic MZI devices and arrays,[15–18] such as device manufacturing variations, limited phase encoding precision, and thermal cross talk. Codesign between the MZI-based NPU and algorithms is needed to compensate or reduce these effects. A simulator for device-circuit-algorithm codesign is required to facilitate the exploration of design of photonic MZI based NPU. In addition, when the NPU scale increases, several challenges limit the performance of the design.

1. Peripheral circuit: The overhead estimation of peripheral circuits, such as digital-to-analog converters (DACs), analog-to-digital converters (ADCs), lasers, and driver circuits, is crucial in designing an MZI-based NPU. It is better than a CMOS NPU in terms of the area efficiency and latency;
2. Bandwidth: The photonic MZI arrays work at high frequency, so it consumes a large number of operands every cycle. But the on-buffer function can be done inside the MZI arrays. It makes the peripheral's input bandwidth as critical factor for the utilization of MZI array;
3. Nonidealities of devices: The nonideal effects of an MZI device and array include limited resolution in voltage control

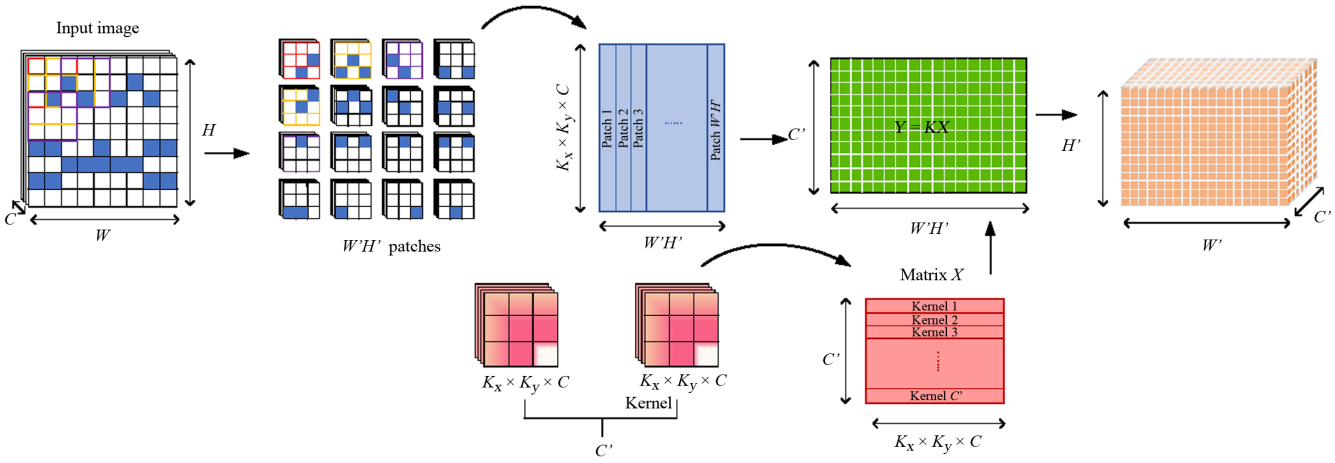


Fig. 3 Patching transform of the CONV to GEMM.

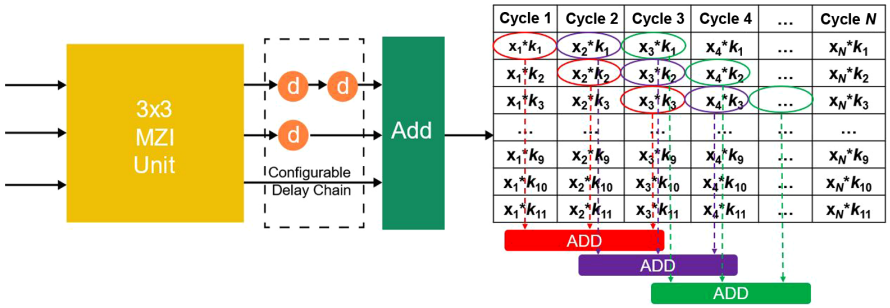


Fig. 4 CDC maps the CONV into the MZI unit.

and gamma noise in the phase shifters. There are two major aspects of the nonideal effects of the MZI device: phase shift and device-level noise on the MZIs. The phase shifter is produced by the MZI device's electronic control resolution. This means the optical device's control voltage cannot be within an arbitrary precision value. Hence, there will be some weight encoding errors when mapping the weight with a higher precision to the real photonic chip. The interval between two phase levels is quadratically enlarged as the voltage increases, leading to a larger phase encoding error.

3 Joint Device-architecture-algorithm Codesign

3.1 Architecture Design

To develop a simulator for the PNPU, a baseline architecture is required. As shown in Fig. 5, the proposed high-level architecture of a general-purpose scalable photonic MZI-based NPU system includes four parts: a photonic MZI chip, field programmable gate arrays (FPGAs) (a digital application-specific integrated circuit chip), peripheral drivers, and transimpedance amplifier (TIA) modules. The FPGA fetches input data from the host module and performs some nonlinearity operation (such as activation function and pooling) and kernel/feature data management in the double data rate and local buffer. The drivers and TIA modules can receive input images from off-chip and send the results to off-chip. The operations are carried out on a photonic chip in the form of light.

The input/output bandwidth of the photonic chip plays a significant role in the performance of the PNPU. We propose two different methods to optimize the input/output bandwidth of the photonic chip. In Sec. 2, we have introduced the hardware CDC, which will save the photonic chip's output bandwidth by reducing the output ports number.

A broadcast and CDC method are introduced to save the input bandwidth of the photonic chip, as shown in Fig. 6. When the kernel filter shifter in the input feature map with a stride of 1, the adjacent two patches have six pixels overlap, and only three pixels are newly input. The delay chain can keep the last two cycles' history values and output nine pixels every cycle. The input CDC can save the bandwidth about 66.7% in this case.

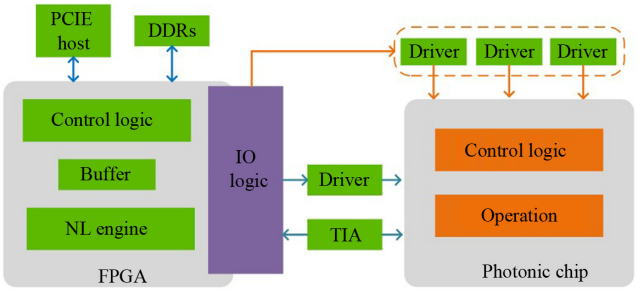


Fig. 5 High-level architecture of a general-purpose scalable photonic MZI-based NPU system.

3.1.1 Photonic chip

The photonic chip receives the feature map data and kernel data sent from the digital chip, performs the CONV/GEMM operation, and sends the data back to the digital chip with the following steps. The photonic chip consists of the laser source, programmable logic controller, spot size convector, modulator, beam splitter, MZI arrays, delay chain, adder, and photodetector.

There are two different architectures of the photonic chip: the CDC-output photonic chip and the CDC-input photonic chip. Figure 7(a) demonstrates the CDC-output photonic chip, and it can save the chip's output bandwidth, whereas Fig. 7(b) shows the CDC-input photonic chip, which is used to save the input bandwidth of the chip. Both transmit data by broadcasting, but the former operates at the output end and the latter works at the input end.

3.1.2 Full-flow simulator based on design architecture

An end-to-end full-flow simulator is developed to explore the design space of the PNPU. The full-flow simulator has the following submodules (Fig. 8): a training framework tool, a photon-based NPU compiler, and a back-end evaluation tool. The full-flow simulator can support the evolution of the training and interface accuracy and support the evaluation of the power, performance, and area (PPA) of the PNPU.

The training framework is different from the traditional digital NPU training framework; there are three major differences. At first, the control weight of the MZI array should be calculated from a complex SVD operation after getting the traditional weight matrix. Next, our interface system is a hybrid photonic–electronic system that can implement different amounts of precision of the interface. The training framework should consider the mix precision. Then the nonideal effects of the MZI array must be taken into consideration, which can

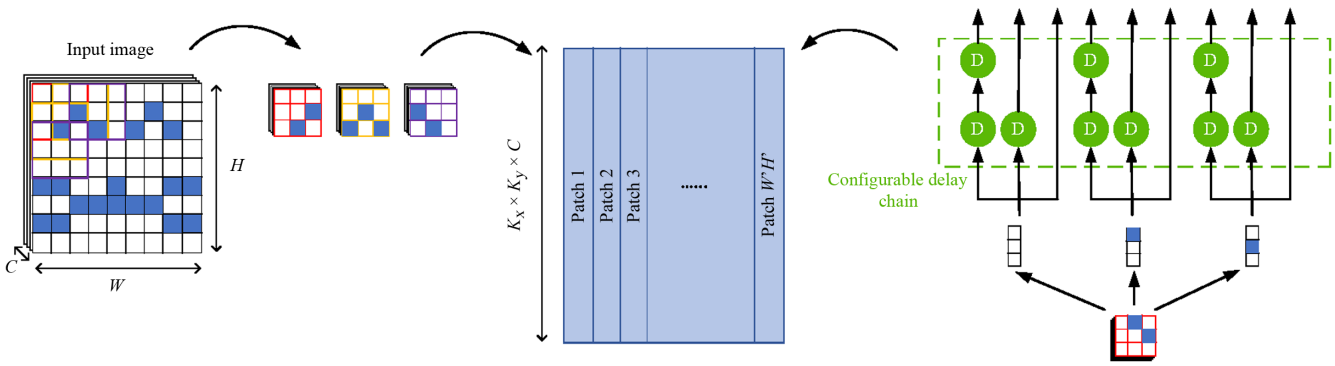


Fig. 6 CDC input for saving bandwidth.

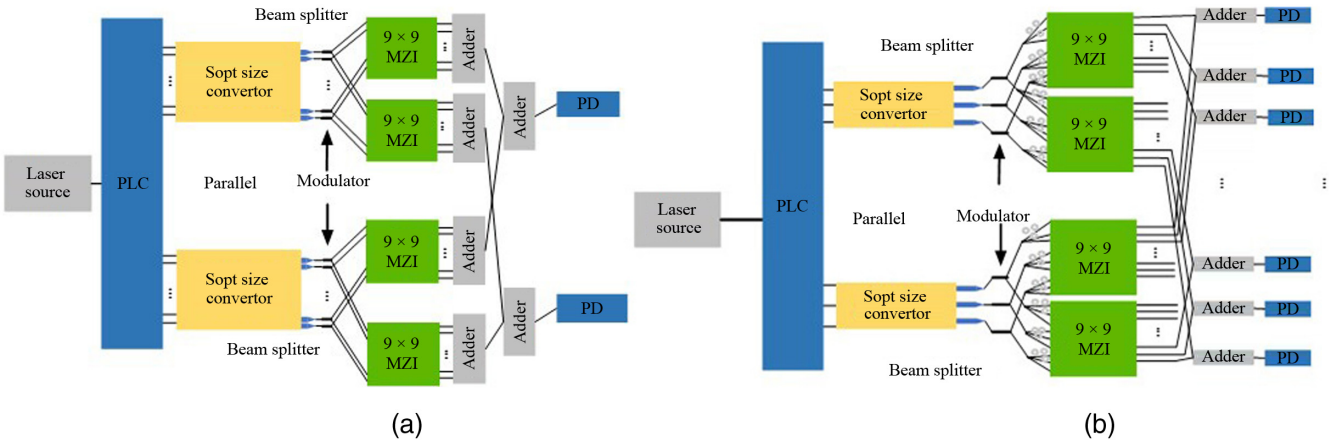


Fig. 7 (a) CDC-output photonic chip architecture and (b) CDC-input photonic chip architecture.

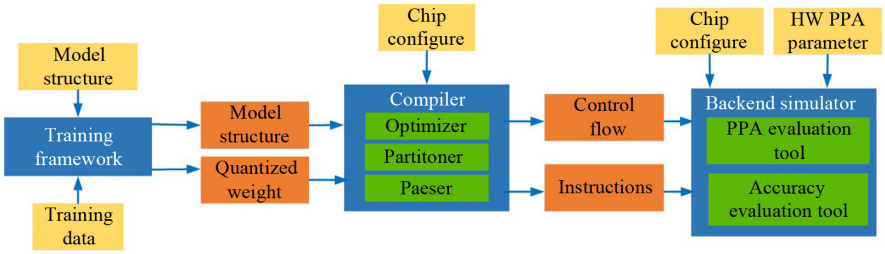


Fig. 8 End-to-end full flow simulator architecture.

augment the robustness of the interface results. We adapt a noise-aware training and quantization scheme[19] to enhance the robustness of the PNPU.

The compiler's target is to optimize the computation and unleash the computation power of PNPU by mapping the model into a highly optimized commands/instructions sequence. It consists of three parts of submodules. They are the optimizer, the parser, and the partitioner. The optimizer prunes the network structure and fine-tunes the weight value so it can reduce the model complexity and improve the inference efficiency. The parser takes the pruned network structure and the hardware configuration as input. It can transform the network structure into computation graph intermediate representation, control flow, and data flow information. The partitioner chooses the best parallelism model and maps the different layers of operation into electronic or photonic logic to perform parallelism. The compiler should also know the hardware configuration to make the best decision. The output data are the control flow information and internally defined commands, and they will be sent to the back-end simulator.

The back-end simulator is used to evaluate two aspects of the design. The first one is the PPA of the PNPU. The second one is the inference accuracy. The PPA evaluation tool models the full design of the PNPU system, which includes buffers, DAC, ADC, modulator, TIA, digital logic, and a laser. The accuracy evaluation tool is used to simulate the forward inference accuracy. It does not directly model the nonidealities of devices. Instead, it extracts the MZI array operation model and adds some random shifter/noise to the output of the MZI operation result.

3.2 Optimization of Algorithm

The photonic chip's architecture and MZI array size have a big influence on the performance of the PNPU system. The multiply accumulate (MAC) utilization is determined by the MZI array size and the network structure, and the required input/output bandwidth depends on the photonic architecture and MZI array size. Therefore, we use the full-flow simulator to get the optimized MZI array size for the PNPU. The input data set for the first network (Net-1) is "MNIST." The size of the MZI array is 28×28. The structure of the DNN is the CONV layer of 16×16, the maximum pool layer is 32×32, and the full connection layer output is 128×10. The second network (Net-2) selects a 32×32 MZI array to cooperate with the larger DNN structure. The third network (Net-3) uses the 224×224 MZI array and the structure of visual geometry group (VGG)16.

As the size of the MZI array increases, the utilization ratio of the MZI unit decreases, as shown in Fig. 9(a). The MZI utilization is also related to the DNN architecture, and the Net-3 decreases less than the Net-1 and Net-2. However, when the MZI array size increases, the required bandwidth increases, but the perf/power decreases when the size is larger than 24 [Fig. 9(b)]. The power of the laser and peripheral circuit increases rapidly when the size of the MZI array increases. Thus, to make a trade-off among the utilization ratio, bandwidth, and perf/power, the MZI array size should be smaller than 32×32.

As discussed in Sec. 2.3, the nonidealities of the MZI unit and the DAC accuracy significantly affect the computing accuracy of the NPU. We use the full-flow simulator to evaluate the effects of the nonidealities and voltage control precision.

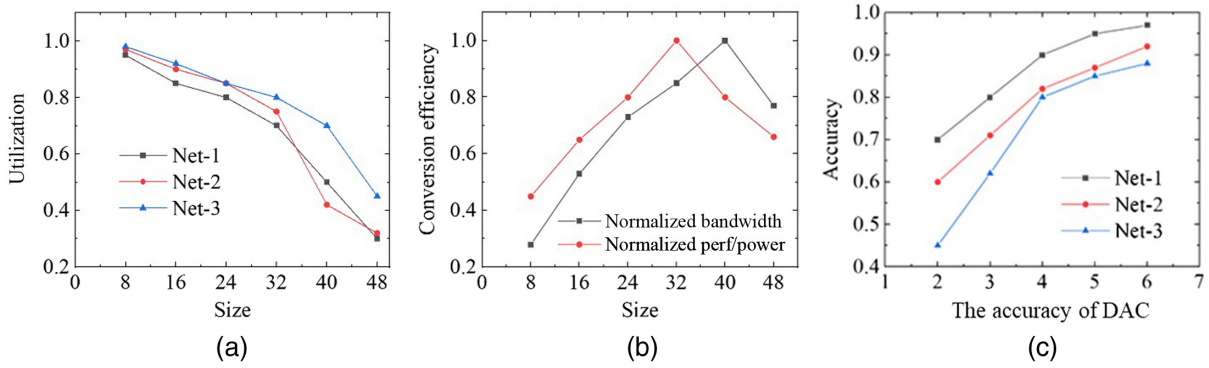


Fig. 9 (a) Utilization of different MZI array sizes. (b) Normalized bandwidth and normalized perf/power of the NPU for different MZI sizes. (c) The accuracies of different DAC control bits.

Table 1 Summary of data set and DNN architecture.

Parameter	Recommend
MZI size	32 × 32
Input DAC precision	4 bits
Output ADC precision	4 bits
CDC I/O pattern	input
Total MZI number	8
MZI broadcast number	4
MZI array parallel number	2

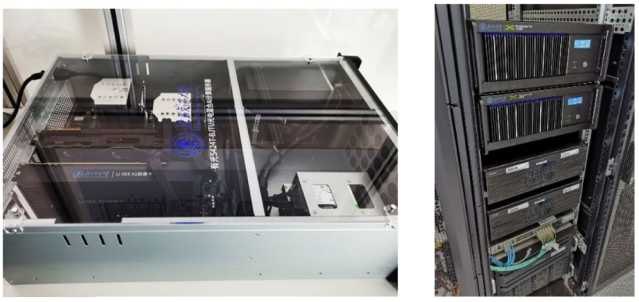
The inference accuracy results of different voltage controls from 3 to 6 bits DAC resolution are shown in Fig. 9(c). The fewer bits of voltage control DAC, the lower accuracy will be gotten. Consequently, the final network is shown in Table 1.

3.3 Construction and Application of Server

Based on the architecture design and algorithm optimization of the above photonic chip, we have completed the chip fabrication. The photonic chip cut from a wafer is shown in Fig. 10.

The server [Fig. 11(a)] is built based on the architecture in Fig. 5, and it is tested in a data center [Fig. 11(b)]. In the test tasks of video recognition and image segmentation, the average test accuracy exceeds 98%, and the computing power exceeds 100TOPS.

Fig. 10 The photonic chip.



(a) (b)

Fig. 11 (a) Setup server. (b) Server application test.

4 Conclusion

In this article, we present a multidirectional collaborative design scheme from device to circuit to algorithm. We have developed an end-to-end photonic neural network simulator that can monitor from multiple directions and visually display the results of data flow. On this basis, we have completed the design and fabrication of a photonic chip and developed a set of servers that can be used for AI tasks. This provides a new idea for follow-up development and utilization of photonic chips. On-chip integrated photonic circuits are an ideal platform for AI. However, to transform the experimental demonstration into a real processor, it is necessary to overcome some key emerging technologies, such as computational bandwidth, intelligent control strategy, and all-optical neural network. In short, the PNPU has great potential in emerging AI applications, but how to comprehensively improve the optical computing system is still challenge.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61827817).

References

1. Y. Li et al., "EDD: efficient differentiable DNN architecture and implementation co-search for embedded AI solutions," in *57th ACM/IEEE Des. Autom. Conf. (DAC)*, pp. 1–6 (2020).

2. T. Young et al., "Recent trends in deep learning based natural language processing," *IEEE Computat. Intell. Mag.* **13**, 55–75 (2018).
3. L. Vivien et al., "Zero-bias 40 Gbit/s germanium waveguide photodetector on silicon," *Opt. Express* **20**, 1096–1101 (2012).
4. Z. He et al., "Noise injection adaption: end-to-end ReRAM crossbar non-ideal effect adaption for neural network mapping," in *Proc. 56th ACM/IEEE Des. Autom. Conf. (DAC)*, 2–6 June 2019, pp. 1–6 (2019).
5. H. Zhou et al., "Photonic matrix multiplication lights up photonic accelerator and beyond," *Light Sci. Appl.* **11**, 30 (2022).
6. A. S. Rekhi et al., "Analog/mixed-signal hardware error modeling for deep learning inference," in Proc. 56th Annu. Des. Autom. Conf. (2019).
7. L. Song et al., "PipeLayer: a pipelined ReRAM-based accelerator for deep learning," in *Proc. IEEE Int. Symp. High Performance Comput. Architecture (HPCA)*, 4–8 February 2017, pp. 541–552 (2017).
8. X. Luo et al., "Metasurface-enabled on-chip multiplexed diffractive neural networks in the visible," *Light Sci. Appl.* **11**(1), 158 (2022).
9. F. Ashtiani, A. J. Geers, and F. Aflatouni, "An on-chip photonic deep neural network for image classification," *Nature* **606**, 501–506 (2022).
10. M. Hu et al., "Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Proc. 53nd ACM/EDAC/IEEE Des. Autom. Conf. (DAC)*, 5–9 June 2016, pp. 1–6 (2016).
11. N. C. Harris et al., "Efficient, compact and low loss thermo-optic phase shifter in silicon," *Opt. Express* **22**, 10487–10493 (2014).
12. P. Sunil et al., "Matrix optimization on universal unitary photonic devices," *Phys. Rev. Appl.* **11**, 064044 (2019).
13. Y. Shen et al., "Deep learning with coherent nanophotonic circuits," *Nat. Photonics* **11**, 441–446 (2017).
14. Z. Zhao et al., "Design technology for scalable and robust photonic integrated circuits: invited paper," in *Proc. IEEE/ACM Int. Conf. Comput.-Aid. Des. (ICCAD)*, 4–7 November 2019, pp. 1–7 (2019).
15. D. Dang et al., "BPLight-CNN: a photonics-based backpropagation accelerator for deep learning," *ACM Journal on Emerging Technologies in Computing Systems* **17**(4), 1–26 (2021).
16. L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the Web]," *IEEE Signal Process. Mag.* **29**(6), 141–142 (2012).
17. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Handbook of Systemic Autoimmune Diseases*, Vol. **2** (2009).
18. J. Deng et al., "ImageNet: a large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. & Pattern Recognit.* (2009).
19. J. Gu et al., "ROQ: a noise-aware quantization scheme towards robust optical neural networks with low-bit controls," in *Proc. Des., Autom. & Test in Eur. Conf. & Exhibit*, Grenoble, France, pp. 1586–1589 (2020).

Biographies of the authors are not available.