Efficient stochastic parallel gradient descent training for on-chip optical processor

Yuanjian Wan, Xudong Liu, Guangze Wu, Min Yang, Guofeng Yan, Yu Zhang and Jian Wang

---

Related articles

Low-loss chip-scale programmable silicon photonic processor
Yiwei Xie, Shihan Hong, Hao Yan, Changping Zhang, Long Zhang, Leimeng Zhuang, Daoxin Dai
Opto-Electronic Advances   2023 **6**, 220030   doi: 10.29026/oea.2023.220030

More related article in Opto-Electronic Journals Group website

Efficient stochastic parallel gradient descent training for on-chip optical processor

Yuanjian Wan[1,2†], Xudong Liu[1,2†], Guangze Wu[1,2], Min Yang[1,2], Guofeng Yan[1,2], Yu Zhang[1,2] and Jian Wang[1,2]*

In recent years, space-division multiplexing (SDM) technology, which involves transmitting data information on multiple parallel channels for efficient capacity scaling, has been widely used in fiber and free-space optical communication systems. To enable flexible data management and cope with the mixing between different channels, the integrated reconfigurable optical processor is used for optical switching and mitigating the channel crosstalk. However, efficient online training becomes intricate and challenging, particularly when dealing with a significant number of channels. Here we use the stochastic parallel gradient descent (SPGD) algorithm to configure the integrated optical processor, which has less computation than the traditional gradient descent (GD) algorithm. We design and fabricate a 6×6 on-chip optical processor on silicon platform to implement optical switching and descrambling assisted by the online training with the SPDG algorithm. Moreover, we apply the on-chip processor configured by the SPGD algorithm to optical communications for optical switching and efficiently mitigating the channel crosstalk in SDM systems. In comparison with the traditional GD algorithm, it is found that the SPGD algorithm features better performance especially when the scale of matrix is large, which means it has the potential to optimize large-scale optical matrix computation acceleration chips.

Keywords: optical communications; optical signal processing; channel descrambling; optical neural network chip; silicon photonics

Introduction

The rapid development of the internet and artificial intelligence has put forward ever-increasing requirements for higher and higher data information capacity of fiber and free-space optical communications. Space-division multiplexing (SDM)[1,2] employs multiple orthogonal channels in space to simultaneously carry and transmit data information for capacity scaling, which has been realized in few-mode fibers (FMFs)[3,4], multi-core fiber[5,6], ring-core fiber[7,8], and free-space communication systems[9,10]. However, many SDM schemes suffer from inevitable channel crosstalk because of the mixing between different channels when transmitting data information together, which leads to the degradation in signal quality at the receiver. Therefore, digital signal processing (DSP) is always required to mitigate the channel crosstalk and recover the original signal in the electrical domain[11–13]. Unfortunately, high-speed DSP chips in the electrical domain are highly complex, difficult to design, and high power consumption[14]. In this scenario, a laudable goal

would be to undo the spatial channel mixing in the optical domain[1]. Additionally, optical switching function in SDM optical communication systems is also highly desirable to facilitate flexible data management[15−17].

In recent years, the integrated reconfigurable optical processor can perform linear transformation like filter[18−20], switching[21,22], temporal integration[23], temporal differentiation[24], Hilbert transformation[25] and matrix multiplication[26] and has been widely used in photonic microwave filter[27], channel equalization[28], optical switching[29], multi-task photonic signal processor[20,30−32], quantum computing[33] and optical computing[34−37]. What's more, it has also been reported to undo the crosstalk caused by mode mixing in multi-mode waveguide[38], FMF[39,40] or MZI[29] mesh automatically and get better signal quality after optimization. However, most existing integrated linear optical processor or optical neural networks (ONNs) are implemented by pre-training the weights of the matrix in electronic computer and adjusting the phase shifter voltage according to the training result, which is very inefficient[16,30]. Additionally, due to external disturbance, thermal crosstalk inside the chip and fabrication tolerance of the complementary metal oxide semiconductor (CMOS) process, there is always a certain difference between the actual matrix of the integrated photonic chip and the target matrix, which will increase geometrically as the size of the chip increases. To cope with this problem, researchers proposed on-chip training schemes for ONNs, including gradient descent (GD) algorithm and its deformation[41], such as numerical gradient descent algorithm[29], genetic algorithm (GA)[42,43], particle swarm optimization (PSO) algorithm[44], bacterial foraging optimization algorithm (BFOA)[45] and so on. However, in the optimization process of GD algorithm, it is usually necessary to establish a relatively clear mathematical model, but it is difficult for the training of ONN. Although Swarm Intelligence algorithm like GA, PSO or BFOA does not depend on the establishment of mathematical network models, a sufficiently large population size is required to guarantee the reliability of the optimization results, which leads to increased complexity. Generally, compared to GA, PSO and BFOA algorithms, the SPGD algorithm has less computational load. Furthermore, since the strategy of Swarm Intelligence algorithm is based on guessing and searching, it is difficult to determine their complexity and their results are not stable enough.

In this paper, we use the Stochastic Parallel Gradient Descent algorithm (SPGD) to train the integrated optical processor experimentally, which is universally used in adaptive optics systems[46,47] because of its advantages of simple parameters, low complexity, high convergence efficiency and good stability. Moreover, we compare it with the GD, GA, and PSO algorithms and draw a conclusion that the processor trained by SPGD algorithm needs less computation than other algorithms, especially when the scale of matrix is large. Lastly, using the fabricated on-chip optical processor trained by SPGD, we implement 6 channels self-configurable optical switching and channel descrambling with favorable performance, which may find potential applications in fiber-optic communication systems based on mode-division multiplexing (MDM), a subset of SDM, as illustrated in Fig. 1(a).

Concept and principle

The concept and principle of the integrated optical processor are shown in Fig. 1(b), which can be divided to three parts. Parts (1) and (3) are unitary optical processors, representing unitary matrix U and $V^\dagger$, respectively, and part (2) represents the diagram matrix Σ. As we all know, arbitrary matrix M can be decomposed into two unitary matrixes U, V and a diagram matrix Σ as $M = U\Sigma V^H$ by the singular value decomposition (SVD)[48,49]. When the power split ratio $\rho=0.5$ and the phase shifter phase is θ and ϕ, the transport matrix of Mach–Zehnder interferometers (MZI) can be expressed as:

$$U_{\text{MZI}} = \frac{1}{2}\begin{pmatrix} e^{i\phi}(e^{i\theta}-1) & i(1+e^{i\theta}) \\ ie^{i\phi}(1+e^{i\theta}) & 1-e^{i\theta} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}. \tag{1}$$

It can be seen that $U_{\text{MZI}} \cdot U_{\text{MZI}}^\dagger = E$ so the MZI can represent any rotational operation in a two-dimensional unitary space. In other words, by changing θ and ϕ controlled by voltage we can achieve any second-order unitary matrix $U(2)$. It can be proven that any $N{\times}N$ unitary matrix $U(N)$ can be decomposed into the product of a series of $U(2)$ as:

$$U(N) = R_{12}R_{13}\cdots R_{lm}\cdots R_{n(n-1)}. \tag{2}$$

Therefore, we can use cascaded MZI mesh to construct any $N{\times}N$ unitary matrix and real matrix.

Optical switching in optical networks can be described by a matrix. For example, the matrix means the switching state. Besides, crosstalk between channels in and SDM communication system can be represented by
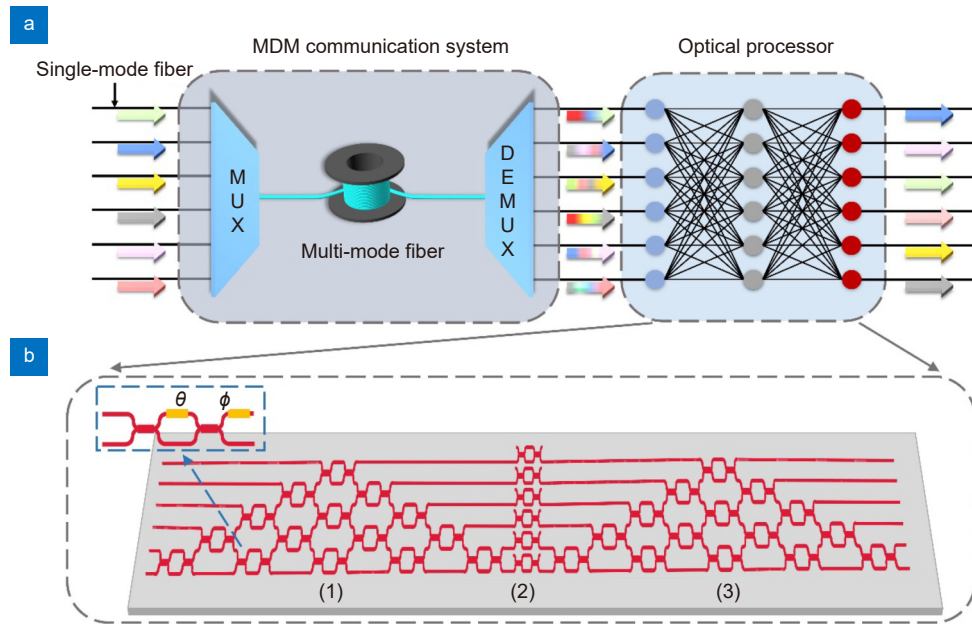
Fig. 1 | (a) Conceptual diagram of the on-chip optical processor for optical switching and channel descrambling in MDM communication systems. **(b)** Schematic configuration of the integrated reconfigurable optical processor. θ and ϕ mean the phase shift of the phase shifters. MDM: mode-division multiplexing; MUX: multiplexer; DEMUX: demultiplexer.

a matrix $\widetilde{U}$, so descrambling is to train the optical processor to enable its transmission matrix U to satisfy $U\widetilde{U} = E$, which means the data information mixed into other channels is returned to the original channel. Thus, we generate a unitary matrix randomly and use

$$J = \sum_{j=1}^{N} \sum_{k=1}^{N} \left| U_{jk}^{\text{sim}} - E_{jk} \right|^2 , \qquad (3)$$

as the Loss function (U^{sim} is $U\widetilde{U}$ and the E is the diagonal matrix) to train the optical processor by Stochastic Parallel Gradient Descent (SPGD) algorithm, which is more suitable for the optimization of complex systems with many control variables and cannot establish accurate mathematical models. The flow chart is shown in Fig. 2. Before optimization begins, we set random perturbation Δu, learning rate γ, the total number of iterations T and randomly generated U, then we reset the voltages applied to all phase shifters u_i and calculate the evaluation function $J[u_i]$. Next we generate random disturbance voltage Δu_i and individually apply $u_i + \Delta u_i$ and $u_i - \Delta u_i$ to calculate $J[u_i + \Delta u_i]$ and $J[u_i - \Delta u_i]$. Then we get $\Delta J = J[u_i + \Delta u_i] - J[u_i - \Delta u_i]$ and update voltages by $u_{i+1} = u_i + \gamma \Delta u \Delta J$. Finally, we judge whether i is greater than T. If i is greater than T, we end the loop, and if i is not greater than T, we continue the loop until the condition is met. Compared to the GD algorithm, it can change all variables together and get the evaluation function $J[u_i]$ rather than changing each variable one by one

to calculate the evaluation function in a iterations, which greatly reduce the amount of computation.

To verify the optimization effect of the algorithm, we use an electronic computer to train the on-chip optical processor for optical switching, optical channel descrambling, and achieving both of them together (i.e. optical channel descrambling and switching), with the obtained results shown in Fig. 3. Figure 3(a–c) are the results of optical switching matrix. The light power distributions matrix is calculated by unitary matrix cascade mentioned above. It is easy to prove that there is a total of 720 6×6 optical switching matrices, all of which are encoded as 1–720 and can be set as targets for optimization. The emulated light power distributions after training are shown in Fig. 3(a) when the switching state is $I_1 - O_2$, $I_2 - O_1$, $I_3 - O_5$, $I_4 - O_6$, $I_5 - O_3$, $I_6 - O_4$, whose normalized light intensity distributions is shown in Fig. 3(b) and the evaluation function changing with iteration rounds of optimization is shown in Fig. 3(c). During the optical channels descrambling process, we randomly generate a set of phases in the part (1) of our chip to emulate the crosstalk matrix $\widetilde{U}$ and optimize the phase shift of the MZI in the part (1) of our chip to minimize evaluation function J. Figure 3(d) shows the normalized light intensity distributions before descrambling and Fig. 3(e) is the normalized light intensity distributions after training. Figure 3(f) is the evaluation function changing with iteration rounds. On this basis, we set optical switching
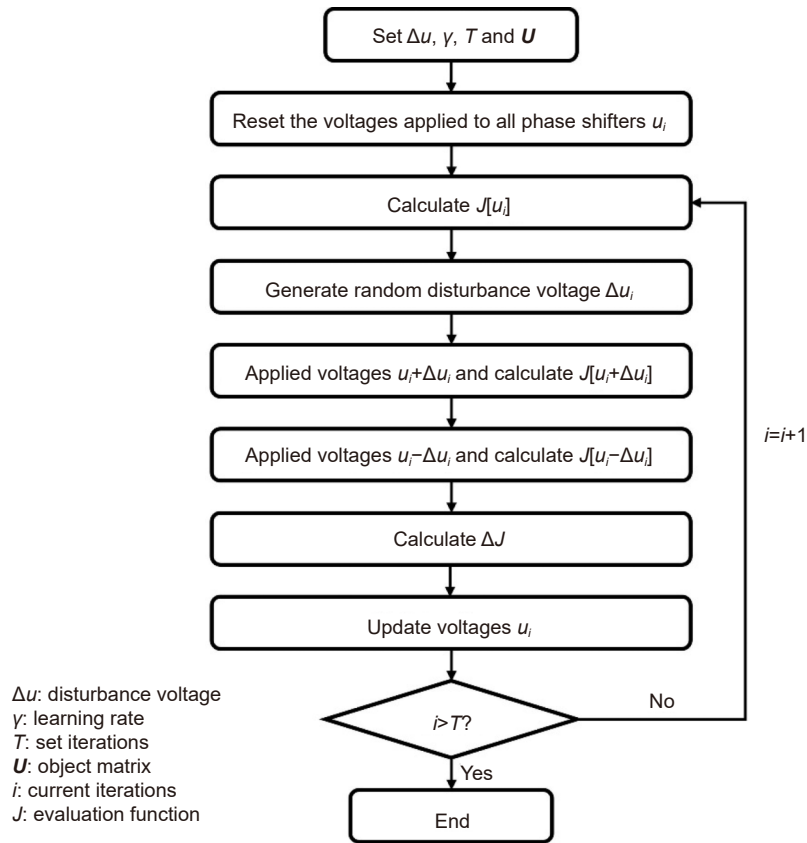
Δu: disturbance voltage
γ: learning rate
T: set iterations
U: object matrix
i: current iterations
J: evaluation function

Fig. 2 | Flow chart of Stochastic Parallel Gradient Descent (SPGD) algorithm.

matrix **T** as the target and optimize the phase shift to make evaluation function J minimum**,** which means optical switching and channel descrambling functions are achieved on the same chip. Figure 3(g) is the normalized light intensity distributions before training with crosstalk when the switching state is $I_1 - O_5$, $I_2 - O_3$, $I_3 - O_2$, $I_4 - O_4$, $I_5 - O_1$, $I_6 - O_6$. Figure 3(h) is the normalized light intensity distributions after training with crosstalk in the same switching state and Fig. 3(i) is the evaluation function changing with iteration rounds.

Experimental results

The diagram of the experimental setup for online training is depicted in Fig. 4(a). Silicon-based photonic chips can be used to function as a 6×6 matrix. Continuous-wave (CW) light at a wavelength of 1550 nm is divided into 6 paths and coupled into the photonic chip in turn through a 1×6 optical micro-electro-mechanical systems (MEMS) switch controlled by a computer. The optical power distribution U_i of the six output ports are detected by optical power meter array and the matrix $U_{\exp} = (U_1, U_2, U_3, U_4, U_5, U_6)$ is the transmission matrix of chip. Polarization controllers (PCs) are used in each path to change the polarization of light to ensure the

maximum coupling efficiency between fibers and optical waveguides. Phase shifters in the chip are driven by a multiple outputs regulated power supply controlled by the same computer. Figure 4(b) shows the microscopy image of optical processor, which is fabricated on the standard 220-nm-silicon-on-insulator (SOI) wafer using 248 nm deep ultraviolet (DUV) lithography and inductively coupled plasma (ICP) etching in United Microelectronics Center (CUMEC). The silicon photonic chip is packaged on a PCB, and light is coupled into the chip through an array of fibers.

In the process of training, we randomly generate an arbitrary exchange matrix **T** as the target of optimization and use the SPGD to optimize the element configuration of the matrix online. The obtained results are shown in Fig. 5. Figure 5(a) is the evaluation function changing with iteration rounds, where the *X*-axis represents the rounds of iteration and the *Y*-axis means the evaluation function defined by:

$$J = \sum_{j=1}^{N} \sum_{k=1}^{N} \left| U_{jk}^{\exp} - T_{jk} \right|^2 . \tag{4}$$

The insets in Fig. 5(a) show the light power distributions when the round of iteration equals 50, 300, and
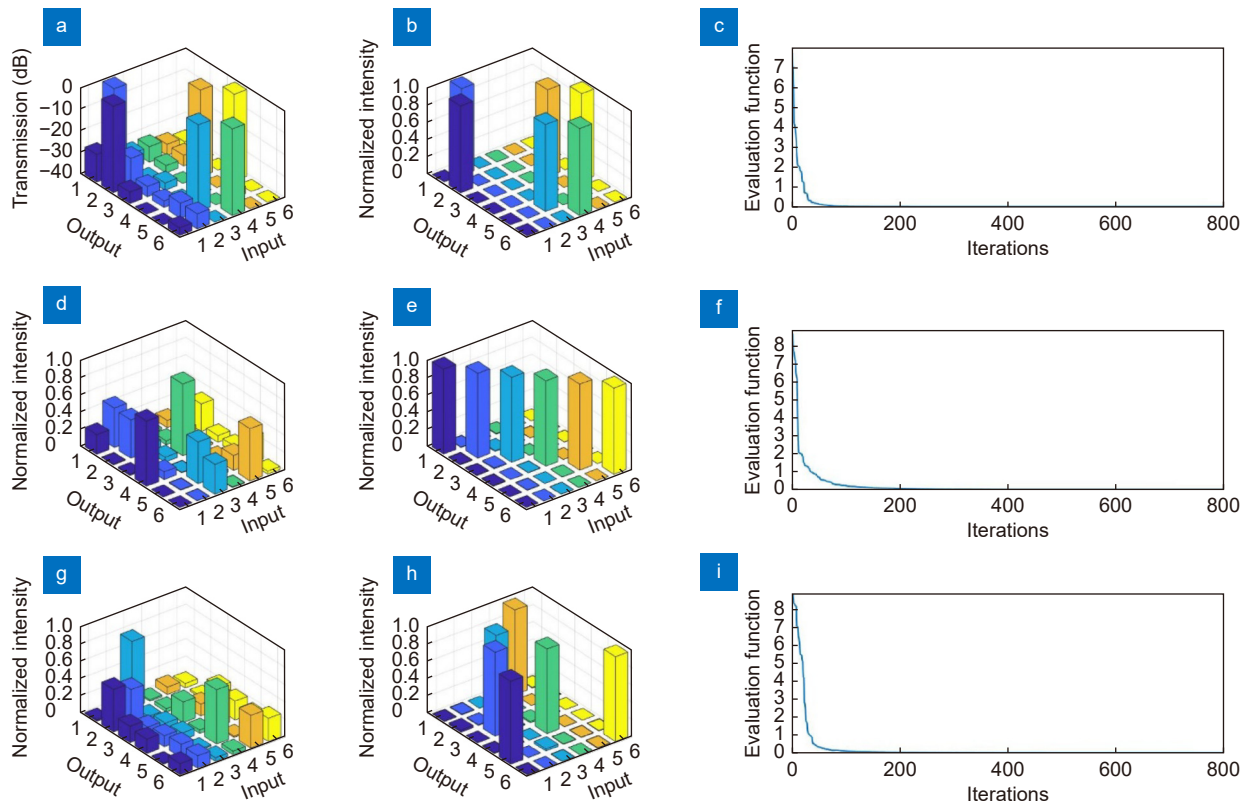
Fig. 3 | Training results in electronic computer for optical switching, optical channel descrambling, and optical channel descrambling and switching. (**a**) Emulated light power distributions and (**b**) normalized light intensity distributions after training when the switching state is $I_1{-}O_2$, $I_2{-}O_1$, $I_3{-}O_5$, $I_4{-}O_6$, $I_5{-}O_3$, $I_6{-}O_4$. (**d**, **e**) Normalized light intensity distributions (d) before and (e) after training when randomly generating a set of phases in the part (1) of our chip to emulate crosstalk. (**g**, **h**) Normalized light intensity distributions (g) before and (h) after training with crosstalk when the switching state is: $I_1{-}O_5$, $I_2{-}O_3$, $I_3{-}O_2$, $I_4{-}O_4$, $I_5{-}O_1$, $I_6{-}O_6$. (**c, f, i**) The evaluation function changing with iteration rounds.
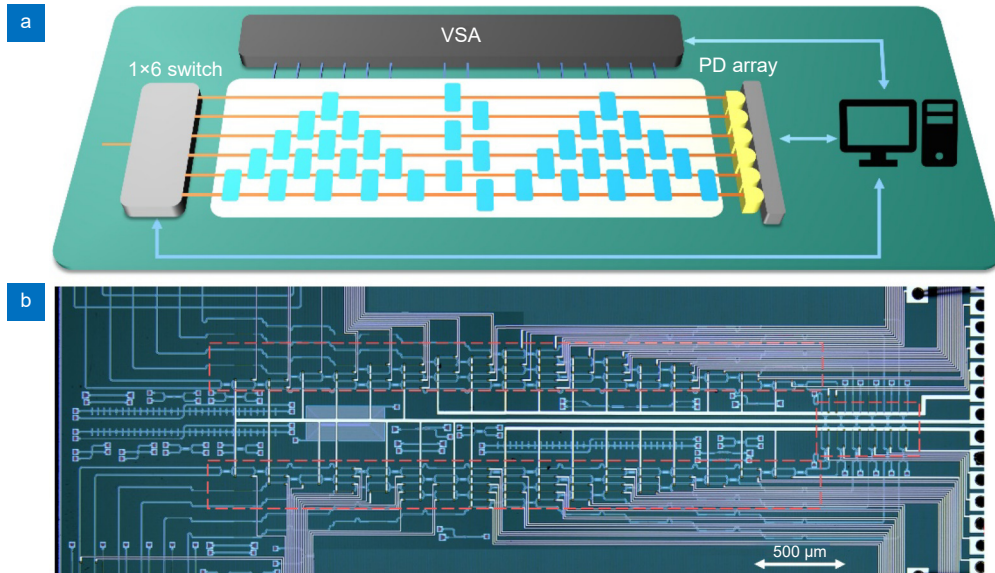

Fig. 4 | (**a**) Schematic of experimental configuration. (**b**) Microscopy image of optical processor. VSA: voltage source array; PD: photodetector array.

600, respectively. The light power and normalized intensity distributions after training are shown in Fig. 5(b) and 5(c), respectively, and the crosstalk of all channels is less than −12.5 dB. In order to reveal the reconfigurabil-ity of our chip, we choose another switching state as the target, and the light power and normalized intensity distributions are shown in Fig. 5(d) and 5(e), respectively.

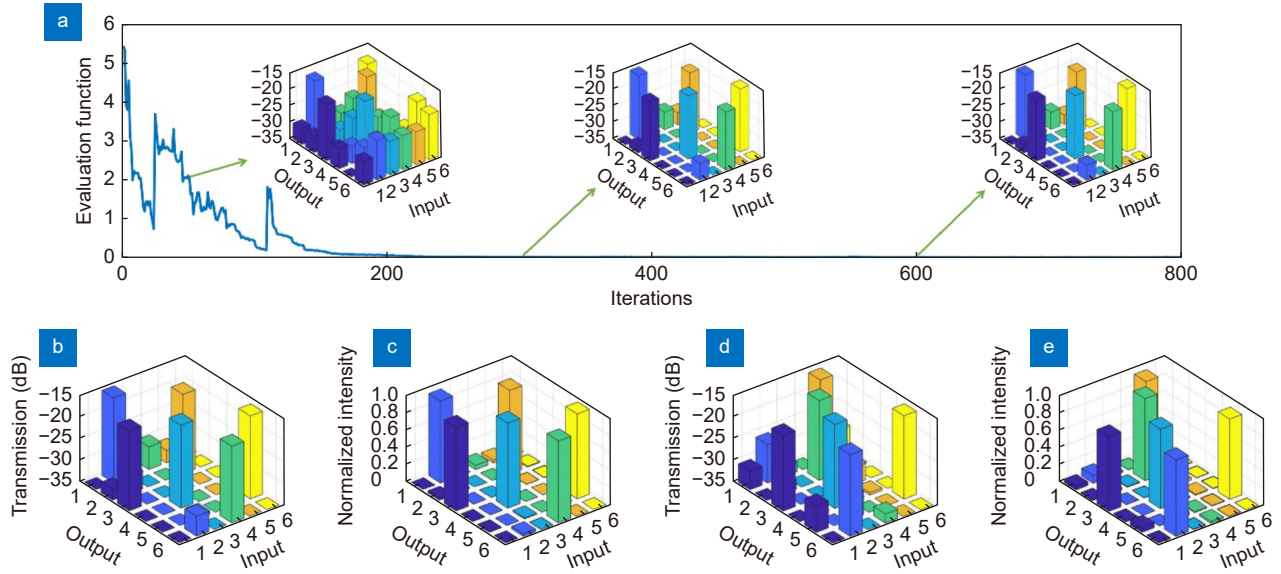In addition to the optical switching, we also perform

Fig. 5 | Online training results for optical switching at a wavelength of 1550 nm. (a) The evaluation function changing with iteration rounds when the switching state is I_1–O_3, I_2–O_1, I_3–O_4, I_4–O_6, I_5–O_2, I_6–O_5. The insets figures show the light power distributions when the round of iteration equals 50, 300, and 600, respectively. **(b)** The measured light power distributions after training. **(c)** The normalized light intensity distributions of measured results. **(d, e)** The measured light power distributions and normalized light intensity distributions when the switching state is I_1–O_3, I_2–O_6, I_3–O_4, I_4–O_2, I_5–O_1, I_6–O_5.

online automatic configuration of arbitrary unitary matrices for mitigating for crosstalk between channels in MDM communication systems. Before the experiment starts, we randomly generate a set of voltages applied to the phase shifter in part (1) of the chip, which is equivalent to constructing a unitary matrix and emulate the mixing between six modes in FMF. During the experiment we use the SPGD algorithm to configure the phase shifter voltage to make the evaluation function defined by

$$J = \sum_{j=1}^{N} \sum_{k=1}^{N} \left| U_{jk}^{\mathrm{exp}} - T_{jk} \right|^2 . \tag{5}$$

The evaluation function changing with iteration rounds is shown in Fig. 6(a) and insets show the light power distributions when the round of iteration equals 1, 300, and 600, respectively. The light power distributions of 6 outputs before descrambling is shown in Fig. 6(b) and the distributions after descrambling is shown in Fig. 6(c). It is obvious that the crosstalk decreases as the iterations increases and the crosstalk is less than 11 dB after descrambling. Figure 6(d) and 6(e) shows descrambling results of training when generating another crosstalk matrix.

Moreover, we also implement both optical switching and channel descrambling (two important modules in the MDM system) on the same chip together. Similar to

descrambling, we also randomly generate a unitary matrix $\widetilde{U}$ to emulate the crosstalk and we use the part (2) of this chip to configure the matrix U to enable $\widetilde{U}U = E$. The obtained results are shown in Fig. 7. Figure 7(a) is the training process and the insets show the light power distributions when the round of iteration equals 1, 100, and 400, respectively. The results of the light power distributions after training are shown in Fig. 7(b) and 7(c) when the switching state is $I_1 - O_4$, $I_2 - O_1$, $I_3 - O_5$, $I_4 - O_6$, $I_5 - O_2$, $I_6 - O_2$. Figure 7(d) and 7(e) are the results of training when we set another switching state of $I_1 - O_5$, $I_2 - O_3$, $I_3 - O_1$, $I_4 - O_6$, $I_5 - O_2$, $I_6 - O_3$.

Finally, in order to compare the signal quality before and after descrambling and verify the application of our photonic chip in the fiber-optic communication system with high-speed advanced modulation signals, we set up the experimental configuration shown in the Fig. 8(a). The 16-ary quadrature amplitude modulation (16-QAM) signal at 20 Gbaud rate is modulated onto an optical carrier by a lithium niobate in-phase and quadrature (IQ) modulator driven by the arbitrary waveform generator (AWG). The light is then split into six branches through a 1×6 power coupler, passing through six fiber delay lines for decorrelation between each channel. The output of the chip is connected to a coherent receiver, where the received 16-QAM signals and local oscillator (LO) are mixed for coherent detection. The detected signals
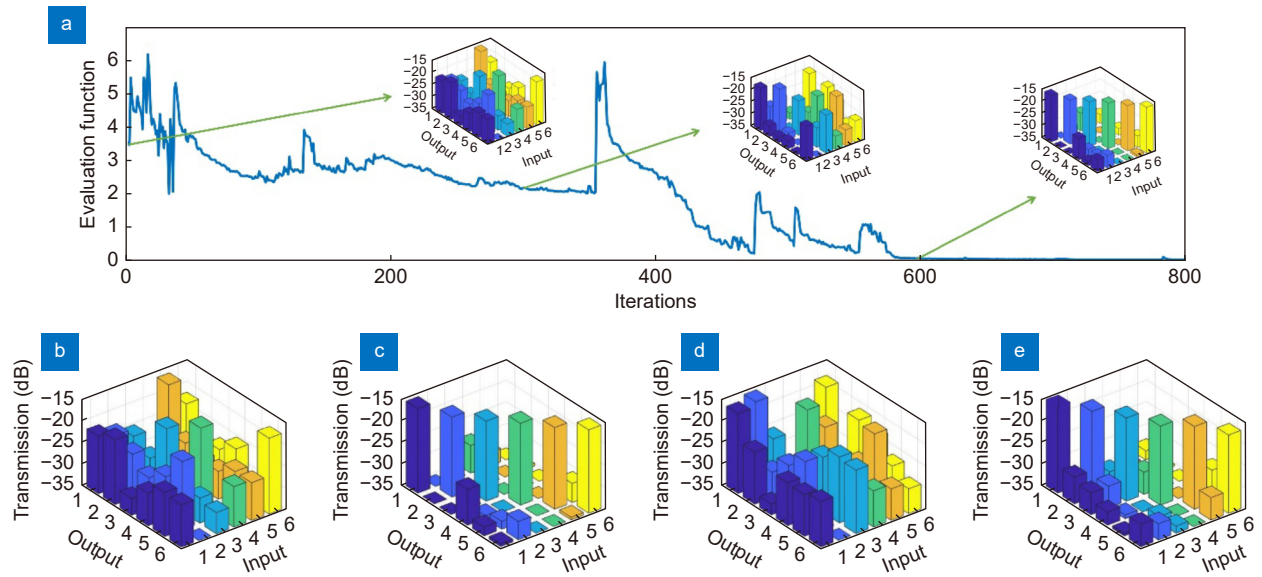
Fig. 6 | Online training results for optical channel descrambling at a wavelength of 1550 nm. (**a**) The evaluation function changing with iteration rounds. The insets show the light power distributions when the round of iteration equals 1, 300, and 600, respectively. (**b**) The light power distributions before training. (**c**) The light power distributions after training. (**d**, **e**) The results of training when generating another matrix $\tilde{U}$.
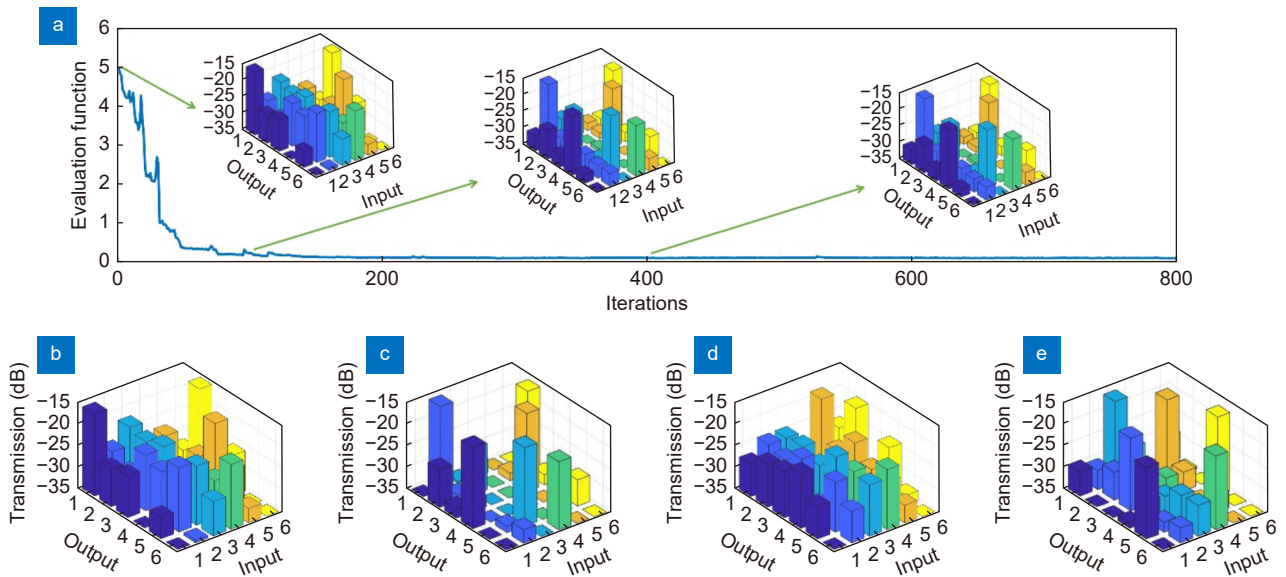

Fig. 7 | Online training results for optical channel descrambling and switching at a wavelength of 1550 nm. (**a**) The evaluation function changing with iteration rounds when the switching state is I_1-O_4, I_2-O_1, I_3-O_5, I_4-O_6, I_5-O_3, I_6-O_2. The insets show the light power distributions when the round of iteration equals 1, 100, and 400, respectively. (**b**) The light power distributions before training. (**c**) The light power distributions after training. (**d**, **e**) The results of training when generating another matrix $\tilde{U}$ and the switching state is I_1-O_5, I_2-O_3, I_3-O_1, I_4-O_6, I_5-O_2, I_6-O_4.

are collected by an optical real-time oscilloscope (OSC) and sent to computer to get the constellations and the bit-error rate (BER) by off-line DSP. The erbium-doped fiber amplifier (EDFA) is used to amplify the signal light. Meanwhile, a variable optical attenuator (VOA) is used to change the optical signal-to-noise ratio (OSNR) of the output signal for measuring the BER transmission performance. The obtained results are shown in Fig. 8(b–f). The BER performance under four conditions is shown in

Fig. 8(b). The BER performance after descrambling is nearly one-tenth of it before descrambling at the same OSNR, so it is clear that the quality of the signal is improved significantly after descrambling in our chip. The constellation chart of back-to-back is shown in Fig. 8(c) while the constellation chart of signal passing through the chip without crosstalk is shown in Fig. 8(d). Figure 8(e) is the constellation chart of signal with random perturbations applied by the part (1) of the chip and Fig. 8(f)
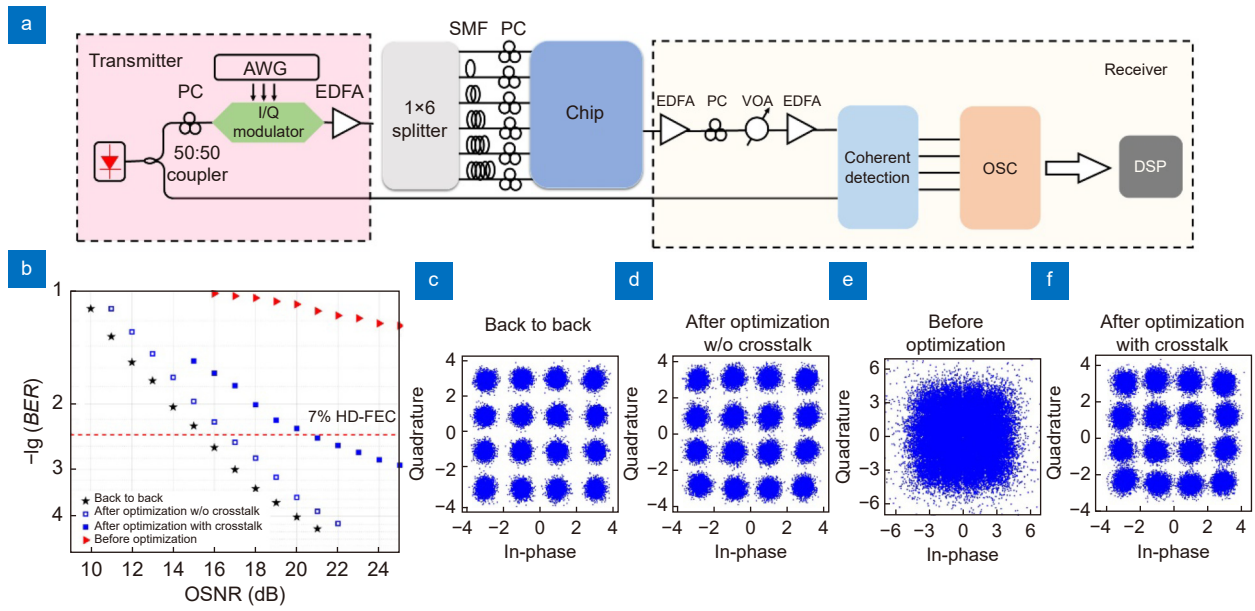
Fig. 8 | Experimental setup and measured results for optical channel descrambling. (**a**) Experimental setup for the 6×6 optical descrambling systems. (**b**) The measured BER performance for back-to-back, optimization without crosstalk, before optimization with crosstalk, and after optimization with crosstalk systems. (**c**) The measured constellation chart at the back-to-back. (**d**) The measured constellation chart without crosstalk. (**e**) The measured constellation chart before optimization with crosstalk. (**f**) The measured constellation chart after optimization with crosstalk. PC: polarization controller; AWG: arbitrary waveform generator; EDFA: erbium-doped fiber amplifier; VOA: variable optical attenuator; OSC: oscilloscope; DSP: digital signal processing.

is the constellation chart after descrambling. By comparing Fig. 8(e) and 8(f), it is obvious that the signal quality is greatly improved and the designed and fabricated on-chip optical processor assisted by SPGD training has a significant effect on signal descrambling.

Discussion

At present, the scale of neural networks based on MZI mesh has reached 64×64[50]. The online learning of such large-scale matrices is very time-consuming. To get the transform matrix, we need to toggle the optical input port in turn through a 1×6 MEMS optical switch and collect optical light intensity distribution through the photodetector array. Besides, the photocurrents should be converted into digital signals, and the updated voltages of phase shifters also need to be converted to analog signals. However, commonly available ADC and DAC chips on the market typically have a bandwidth not exceeding 1 GHz and the MEMS optical switch speeds typically range from kilohertz to megahertz, with some advanced designs achieving switching times in the nanosecond range, which is the main time consumption of matrix online training. Traditional GD algorithm firstly generates a set of variables $u_1, u_2, ..., u_n$ (n is the number of variables) randomly and calculates the evaluation

function $J[u_1, u_2, ..., u_n]$. Then it applies perturbation Δu to all variables in turn and get the transport matrix of the photonic chip to calculate the evaluation function $J_k[u_1, u_2, ..., u_k + \Delta u, ..., u_n](k = 1, 2, ..., n)$. The partial derivative for u_k is calculated by $J'_k = \frac{J_k - J}{\Delta u}$ and the gradient of the evaluate function J can be expressed as: $\nabla J = (J'_1, J'_2, ..., J'_n)$. Finally, we update the variables in new round and so on until the evaluate function is less than the threshold. Therefore, in order to train a $N×N$ unitary matrix with $N(N-1)$ phase shifts, we need update the voltage $N(N-1)$ times, measure the transmission matrix $N(N-1)$ times, and calculate the evaluation function $N(N-1)$ times in one iteration round by the GD algorithm. If the number of iterations is T, we need to update the variables and get the transform matrix $N(N-1)×T$ times. Additionally, Swarm Intelligence algorithms like GA and PSO need to generate a group, in which each individual represents a set of variables to be optimized. During training, all individuals in the group should be calculated, and the best individual should be selected. Therefore, we need to calculate the transmission matrix and evaluation function M (M represents the population size, typically ranging from 10 to 100) times in one iteration round. Thus, we need update the variables and get the transform matrix $M×T$ times. Whereas,

Table 1 | Performance of different algorithms.

Algorithm	Numbers of update	Matrix sizes			
		N=6	*N*=10	*N*=16	*N*=32
GD	$N(N-1)\times T$	690	3870	13200	93248
GA	$M\times T$	1048	9046.67	39732	171200
PSO	$M\times T$	1024	5912	31056	116145
SPGD	$3\times T$	297.9	1092.6	4752.6	18053.1

the SPGD algorithm used in our experiment apply perturbation Δu to all variables together. Consequently, it only needs to update the voltage of phase shifters three times. As a result, we only need to update the variables and obtain the transformation matrix $3\times T$ times during the optimization. This significantly reduces the time and power consumption for optimization. In order to further verify this conclusion, we study the performance of GD, GA, PSO and SPGD algorithms to train the same 6×6, 10×10, 16×16, and 32×32 matrices numerically. Specifically, we generate a random unitary matrix as crosstalk and use the above four algorithms to optimize the on-chip processor to make the $\left| \tilde{U}U - E \right|^2$ minimum. A threshold is set for making the program automatically jump out of the loop when the evaluate function is below it and the number of iterations is recorded. Due to the random of the SPGD algorithm, we take the average of the calculated results to ensure the reliability of the results. It is worth mentioning that both GA and PSO algorithm need a large enough population size to ensure the performance of optimization but a too large group will lead to the waste of computation. Therefore, we train the photonic chip with difference population size and took the best. The results are shown in the Table1.

From Table 1, It can be seen that the SPGD algorithm requires much less computation compares to the GD, GA and PSO algorithms. Especially, when the matrix sizes are 6×6, 10×10 and 16×16, there is not much difference of the total amount of computation between these four algorithms. However, When the matrix size increases to 32×32, the number of updates for SPGD algorithm is close to only one fifth of the traditional GD algorithm. This can be explained by the fact that the number of iterations for the GD algorithm is $N(N-1)\times T$, indicating a square relationship with *N*. Consequently, it is foreseeable that the difference will become more apparent as the size of the matrix expands further. Therefore, the algorithm has significant implications for larger scale optical matrix computation acceleration chips.

Conclusion

We propose and demonstrate an integrated reconfigurable optical processor trained by SPGD algorithm and use it for unique optical switching and channel descrambling. On this basis, we apply it to optical communication systems with 20-Gbaud 16-QAM signals and measure the constellations and BER performance, through which we can draw the conclusion that the 6×6 integrated optical processor has the application prospect in robust optical switching and channel descrambling of MDM-based fiber-optic communication systems. In addition, it can also be applied to various application scenarios in SDM-based fiber and free-space optical communication systems. Lastly, we compare it with traditional GD algorithm and show its distinct advantages, especially when the matrix size is large. The suggestion that the SPGD algorithm has the potential to optimize large-scale optical matrix computation acceleration chips points to a significant breakthrough in computational optics. This innovation holds the promise of accelerating complex optical matrix computations, which are fundamental to various optical processing applications, including signal processing and optical computing. It hints at a future where optical processors can handle even more intricate tasks efficiently.

References

1. Richardson DJ, Fini JM, Nelson LE. Space-division multiplexing in optical fibres. *Nat Photonics* **7**, 354–362 (2013).
2. Berdagué S, Facq P. Mode division multiplexing in optical fibers. *Appl Opt* **21**, 1950–1955 (1982).
3. Ryf R, Fontaine NK, Wittek S et al. High-spectral-efficiency mode-multiplexed transmission over graded-index multimode fiber. In *2018 European Conference on Optical Communication (ECOC)* 1–3 (IEEE, 2018); http://doi.org/10.1109/ECOC.2018.8535536.
4. van Uden RGH, Correa RA, Lopez EA et al. Ultra-high-density spatial division multiplexing with a few-mode multicore fibre. *Nat Photonics* **8**, 865–870 (2014).
5. Ding QC, Liu B, Ren JX et al. High-security SCMA-OFDM multi-core fiber transmission system based on a regular hexagon chaotic codebook. *Opt Express* **30**, 36010–36024 (2022).
6. Puttnam BJ, Eriksson TA, Mendinueta JMD et al. Modulation

formats for multi-core fiber transmission. *Opt Express* **22**, 32457–32469 (2014).

7. Zhu L, Zhu GX, Wang AD et al. 18 km low-crosstalk OAM + WDM transmission with 224 individual channels enabled by a ring-core fiber with large high-order mode group separation. *Opt Lett* **43**, 1890–1893 (2018).

8. Zhu GX, Hu ZY, Wu X et al. Scalable mode division multiplexed transmission over a 10-km ring-core fiber using high-order orbital angular momentum modes. *Opt Express* **26**, 594–604 (2018).

9. Wang J, Yang JY, Fazal IM et al. Terabit free-space data transmission employing orbital angular momentum multiplexing. *Nat Photonics* **6**, 488–496 (2012).

10. Zhao YF, Liu J, Du J et al. Experimental demonstration of 260-meter security free-space optical data transmission using 16-QAM carrying orbital angular momentum (OAM) beams multiplexing. In *Optical Fiber Communication Conference* Th1H. 3 (OSA, 2016); http://doi.org/10.1364/OFC.2016.Th1H.3.

11. Koebele C, Salsi M, Milord L et al. 40km transmission of five mode division multiplexed data streams at 100 Gb/s with low MIMO-DSP complexity. In *37th European Conference and Exposition on Optical Communications* Th. 13. C. 3 (OSA, 2011); http://doi.org/10.1364/ECOC.2011.Th.13.C.3.

12. Randel S, Corteselli S, Badini D et al. First real-time coherent MIMO-DSP for six coupled mode transmission. In *2015 IEEE Photonics Conference (IPC)* 1–2 (IEEE, 2015); http://doi.org/10.1109/IPCon.2015.7323761.

13. Randel S, Sierra A, Mumtaz S et al. Adaptive MIMO signal processing for mode-division multiplexing. In *Optical Fiber Communication Conference* OW3D. 5 (OSA, 2012); http://doi.org/10.1364/OFC.2012.OW3D.5.

14. Diamantopoulos NP, Shariati B, Tomkos I. On the power consumption of MIMO processing and its impact on the performance of SDM networks. In *2017 Optical Fiber Communications Conference and Exhibition (OFC)* 1–3 (IEEE, 2017).

15. Yang L, Zhou T, Jia H et al. General architectures for on-chip optical space and mode switching. *Optica* **5**, 180–187 (2018).

16. Clements WR, Humphreys PC, Metcalf BJ et al. Optimal design for universal multiport interferometers. *Optica* **3**, 1460–1465 (2016).

17. Cao XP, Zheng S, Long Y et al. Mesh-structure-enabled programmable multitask photonic signal processor on a silicon chip. *ACS Photonics* **7**, 2658–2675 (2020).

18. Little BE, Chu ST, Haus HA et al. Microring resonator channel dropping filters. *J Lightwave Technol* **15**, 998–1005 (1997).

19. Pérez D, Gasulla I, Capmany J. Programmable multifunctional integrated nanophotonics. *Nanophotonics* **7**, 1351–1371 (2018).

20. Zhuang LM, Roeloffzen CGH, Hoekman M et al. Programmable photonic signal processor chip for radiofrequency applications. *Optica* **2**, 854–859 (2015).

21. Ribeiro A, Ruocco A, Vanacker L et al. Demonstration of a 4 × 4-port universal linear circuit. *Optica* **3**, 1348–1357 (2016).

22. Lu LJ, Zhao SY, Zhou LJ et al. 16 × 16 non-blocking silicon optical switch based on electro-optic Mach-Zehnder interferometers. *Opt Express* **24**, 9295–9307 (2016).

23. Ferrera M, Park Y, Razzari L et al. On-chip CMOS-compatible all-optical integrator. *Nat Commun* **1**, 29 (2010).

24. Dong JJ, Zheng AL, Gao DS et al. High-order photonic differentiator employing on-chip cascaded microring resonators. *Opt Lett* **38**, 628–630 (2013).

25. Liu WL, Li M, Guzzon RS et al. A fully reconfigurable photonic

26. Zhou HL, Dong JJ, Cheng JW et al. Photonic matrix multiplication lights up photonic accelerator and beyond. *Light Sci Appl* **11**, 30 (2022).

27. Fandiño JS, Muñoz P, Doménech D et al. A monolithic integrated photonic microwave filter. *Nat Photonics* **11**, 124–129 (2017).

28. Kouloumentas C, Tsokos C, Groumas P et al. Multi-rate and multi-channel optical equalizer based on photonic integration. *IEEE Photonics Technol Lett* **32**, 1465–1468 (2020).

29. Zhou HL, Zhao YH, Wang X et al. Self-configuring and reconfigurable silicon photonic signal processor. *ACS Photonics* **7**, 792–799 (2020).

30. Pérez D, Gasulla I, Crudgington L et al. Multipurpose silicon photonics signal processor core. *Nat Commun* **8**, 636 (2017).

31. Lu LJ, Zhou LJ, Chen JP. Programmable SCOW mesh silicon photonic processor for linear unitary operator. *Micromachines (Basel)* **10**, 646 (2019).

32. Zhou HL, Zhao YH, Xu GX et al. Chip-scale optical matrix computation for pagerank algorithm. *IEEE J Sel Top Quantum Electron* **26**, 8300910 (2020).

33. Zheng Y, Zhai CH, Liu DJ et al. Multichip multidimensional quantum networks with entanglement retrievability. *Science* **381**, 221–226 (2023).

34. Tian Y, Zhao Y, Liu SP et al. Scalable and compact photonic neural chip with low learning-capability-loss. *Nanophotonics* **11**, 329–344 (2022).

35. Zhang H, Gu M, Jiang XD et al. An optical neural chip for implementing complex-valued neural network. *Nat Commun* **12**, 457 (2021).

36. Feng CH, Gu JQ, Zhu HQ et al. A compact butterfly-style silicon photonic–electronic neural chip for hardware-efficient deep learning. *ACS Photonics* **9**, 3906–3916 (2022).

37. Shen YC, Harris NC, Skirlo S et al. Deep learning with coherent nanophotonic circuits. *Nat Photonics* **11**, 441–446 (2017).

38. Annoni A, Guglielmi E, Carminati M et al. Unscrambling light —automatically undoing strong mixing between modes. *Light Sci Appl* **6**, e17110 (2017).

39. Tanomura R, Tang R, Soma G et al. All-optical dual-polarization MIMO processor based on integrated optical unitary converter. In *2022 European Conference on Optical Communication (ECOC)* 1–4 (IEEE, 2022).

40. Tanomura R, Tang R, Ghosh S et al. Robust integrated optical unitary converter using multiport directional couplers. *J Lightwave Technol* **38**, 60–66 (2020).

41. Hughes TW, Minkov M, Shi Y et al. Training of photonic neural networks through *in situ* backpropagation and gradient measurement. *Optica* **5**, 864–871 (2018).

42. Zhang H, Thompson J, Gu ML et al. Efficient on-chip training of optical neural networks using genetic algorithm. *ACS Photonics* **8**, 1662–1672 (2021).

43. Shao R, Zhang G, Gong X. Generalized robust training scheme using genetic algorithm for optical neural networks with imprecise components. *Photonics Res* **10**, 1868–1876 (2022).

44. Zhang T, Wang J, Dan YH et al. Efficient training and design of photonic neural network through neuroevolution. *Opt Express* **27**, 37150–37163 (2019).

45. Cong GW, Yamamoto N, Inoue T et al. On-chip bacterial foraging training in silicon photonic circuits for projection-enabled nonlinear classification. *Nat Commun* **13**, 3261 (2022).

46. Geng C, Luo W, Tan Y et al. Experimental demonstration of using divergence cost-function in SPGD algorithm for coherent beam combining with tip/tilt control. *Opt Express* **21**, 25045–25055 (2013).

47. Vorontsov MA, Sivokon VP. Stochastic parallel-gradient-descent technique for high-resolution wave-front phase-distortion correction. *J Opt Soc Am A* **15**, 2745–2758 (1998).

48. Reck M, Zeilinger A, Bernstein HJ et al. Experimental realization of any discrete unitary operator. *Phys Rev Lett* **73**, 58–61 (1994).

49. Harris NC, Carolan J, Bunandar D et al. Linear programmable nanophotonic processors. *Optica* **5**, 1623–1631 (2018).

50. Harris NC, Braid R, Bunandar D et al. Accelerating artificial intelligence with silicon photonics. In *Optical Fiber Communication Conference (OFC) 2020* W3A. 3 (Optica Publishing Group, 2020);http://doi.org/10.1364/OFC.2020.W3A.3.

Acknowledgements

Competing interests

The authors declare no competing financial interests.

Scan for Article PDF