RESEARCH ARTICLE | FEBRUARY 04 2025

Programming universal unitary transformations on a general-purpose silicon photonic platform

José Roberto Rausell-Campo ; Daniel Pérez-López ; José Capmany Francoy

APL Photonics 10, 026102 (2025)
https://doi.org/10.1063/5.0235712

Programming universal unitary transformations on a general-purpose silicon photonic platform

Cite as: APL Photon. 10, 026102 (2025); doi: 10.1063/5.0235712
Submitted: 29 August 2024 • Accepted: 24 January 2025 •
Published Online: 4 February 2025

José Roberto Rausell-Campo,[1,a)] Daniel Pérez-López,[2] and José Capmany Francoy[1]

AFFILIATIONS

[1] Photonics Research Lab, iTEAM. Universitat Politècnica de Valencia, Valencia 46022, Spain
[2] IPronics Programmable Photonics S.L., Valencia 46010, Spain

[a)] **Author to whom correspondence should be addressed:** joraucam@upv.es

ABSTRACT

General-purpose programmable photonic processors provide a versatile platform for integrating diverse functionalities on a single chip. Leveraging a two-dimensional hexagonal waveguide mesh of Mach–Zehnder interferometers, these systems have demonstrated significant potential in microwave photonic applications. Additionally, they are a promising platform for creating unitary linear transformations, which are key elements in quantum computing and photonic neural networks. However, a general procedure for implementing these transformations on such systems has not been established yet. This work demonstrates the programming of universal unitary transformations on a general-purpose programmable photonic circuit with a hexagonal topology. We detail the steps to split the light on-chip, demonstrate that an equivalent structure to the Mach–Zehnder interferometer with one internal and one external phase shifter can be built in the hexagonal mesh, and program both the triangular and rectangular architectures for matrix multiplication. We recalibrate the system to account for passive phase deviations. Experimental programming of 3 × 3 and 4 × 4 random unitary matrices yields fidelities >98% and bit precisions over five bits. To the best of our knowledge, this is the first time that random unitary matrices are demonstrated on a general-purpose photonic processor and pave the way for the implementation of programmable photonic circuits in optical computing and signal processing systems.

I. INTRODUCTION

In the previous years, general-purpose programmable photonic integrated processors have emerged as a promising platform for the inclusion of a variety of functionalities on a single platform, similar to field-programmable gate arrays in electronics.[1–3] These systems allow for software-controlled manipulation of light paths across a 2D waveguide mesh using their 2 × 2 building blocks, known as programmable unit cells (PUCs). PUCs are typically built from symmetric Mach–Zehnder interferometers with a phase shifter on each arm. The literature reports various waveguide mesh topologies, with triangular, rectangular, and hexagonal being the most common.[2,4,5]

One of the main application areas of programmable photonic circuits is microwave photonic systems,[6] where RF signals are processed in the optical domain, and this benefits from the increased bandwidth and reduced latency when compared with its electronic counterparts.[7] Recently, researchers have presented a general-purpose programmable silicon photonic circuit with a hexagonal topology.[8] They demonstrated the complete system, including the photonic circuit, electronic drivers, and software layer. Their work experimentally showed 12 microwave photonic functionalities, highlighting the versatility of these devices.

Another area of interest is the implementation of unitary linear transformations. Application-specific photonic circuits have demonstrated capabilities for performing unitary matrix multiplications using coherent structures that combine beam splitters and phase shifters within a planar architecture.[9] The basic building blocks, Mach–Zehnder interferometers with internal and external phase shifters, can be connected in various arrangements. Rectangular (Clements)[10] and triangular (Reck)[11] topologies are the two most common approaches. An alternative using a PUC as a building block has also been proposed to reduce area by eliminating the external phase shifter.[12] These coherent circuits inherently apply unitary

transformations, which are crucial for applications such as quantum computing.[13–16] Additionally, unitary systems can be used to unscramble the mixing of optical signals traveling through multimode fibers[17–19] or free-space optical communication channels.[20] Finally, general matrix multiplications can be achieved using SVD decomposition, where two unitary matrices and a diagonal matrix are concatenated.[21] Photonic general matrix multiplications have shown promise for deep learning,[22–24] unconventional computing approaches,[25,26] and RF signal separation.[27–29]

A general procedure for the implementation of linear unitary transformations on general-purpose photonic platforms has not been reported yet. While Perez *et al.* demonstrated programming a rectangular interferometer within a seven-cell hexagonal waveguide mesh, their experimental validation was limited to implementing permutation matrices (unitary transformations with absolute values of elements equal to 0 or 1). This was a consequence of two reasons: First, due to the mesh size, the input vector to be multiplied by the matrix has to be encoded externally, thus compromising coherence and rendering complex matrix-vector multiplications infeasible. Second, it is known that due to fabrication imperfections, the initial state of the phase shifters is not 0 rad and a calibration of that passive phase must be performed.[30] The required steps to perform this calibration were not provided either theoretically or experimentally.

In the following work, we demonstrate how to program arbitrary unitary transformations on a general-purpose programmable photonic chip with a hexagonal topology. We utilize the commercially available Smartlight processor from iPronics Programmable Photonics.[8,31] First, we explain the necessary steps for splitting light to maintain on-chip coherence. Second, we show how to replicate the transfer function of a Mach–Zehnder interferometer with internal and external phase shifters using two programmable unit cells (PUCs). We then demonstrate that both triangular and rectangular architectures can be programmed on this platform and present the recalibration process to measure the passive state of the phase shifters. We experimentally program 4 × 4 and 3 × 3 random unitary matrices. We calculate their fidelity and bit precision and perform random complex matrix-vector multiplications. Finally, we show the programming of a set of quantum logic gates.

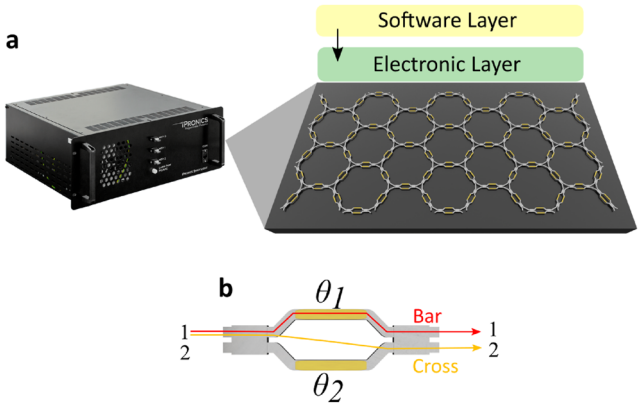

FIG. 1. (a) Smartlight processor from IPronics. It combines a hexagonal waveguide mesh with an electronic and software layer. (b) Programmable unit cell (PUC) of the hexagonal mesh. It consists of two internal phase shifters θ_1 and θ_2.

II. HEXAGONAL PROGRAMMABLE PHOTONIC CIRCUITS

The hexagonal topology has appeared to be the most promising topology for general-purpose programmable processors; see Fig. 1(a). In this section, we provide their fundamental working principle and calibration requirements.

A. Programmable unit cell

The basic building block of the hexagonal mesh is the programmable unit cell, which comprises a Mach–Zehnder interferometer with a thermo-optic phase shifter in each of the top and bottom arms. A picture of the PUC is shown in Fig. 1(b). The transfer function of the PUC can be expressed as

$$ie^{i\frac{\theta_1+\theta_2}{2}}\begin{pmatrix}\sin\left(\frac{\theta_1-\theta_2}{2}\right) & \cos\left(\frac{\theta_1-\theta_2}{2}\right)\\ \cos\left(\frac{\theta_1-\theta_2}{2}\right) & -\sin\left(\frac{\theta_1-\theta_2}{2}\right)\end{pmatrix} \tag{1}$$

$$= ie^{i\phi}\begin{pmatrix}\sqrt{1-k} & \sqrt{k}\\ \sqrt{k} & \sqrt{1-k}\end{pmatrix}. \tag{2}$$

The PUC enables the control of the coupling ratio (k) between input 1 and output 2, as well as the phase shift (ϕ) experienced by the optical signal. This control is achieved by adjusting the relative phase applied to the top and bottom arms. In the bar state ($k = 0$), light entering port 1 exits from port 1 and light entering port 2 exits from port 2. Conversely, in the cross state ($k = 1$), light entering port 1 exits from port 2 and light entering port 2 exits from port 1.

B. Calibration

The calibration process aims to find the relationship between the applied current on the thermo-optic actuator and the resulting phase shift, which can be expressed by the following equation:

$$\theta = \theta_0 + \alpha I^2, \tag{3}$$

where α is the proportionality constant between the current and the phase term and θ_0 is the passive phase that appears as a consequence of the fabrication imperfections. This imperfection in the geometry of the waveguides creates a difference in the group index between the top and bottom waveguides of the MZI. In an ideal system, if we set $\theta_1 = \theta_2 = 0$ in (1), the PUC is in the *cross* state. However, variations in the passive phase across different PUCs will cause the initial state to deviate from the ideal and become a random one between *bar* and *cross*. The calibration of each individual PUC in a hexagonal waveguide mesh requires the use of an optimization and auto-routing algorithm.[32,33] Once calibrated, we can define the coupling state of each PUC, enabling functionalities such as beam splitters, optical interconnects, or filters. However, for applications requiring coherence, such as unitary matrix multiplications, calibration of individual phase shifters is not sufficient. We also need to compensate for the difference in phase gained by light traversing different paths with the same number of PUCs. This phase deviation,

also due to fabrication imperfections but fixed for each defined path, can be treated similarly to the passive phase of individual PUCs. In Sec. III, we will demonstrate how to apply this re-calibration for matrix multiplications with rectangular and triangular topologies. The same procedure can be extended to other coherent structures.

C. Smartlight processor

Regarding the experimental demonstration of our work, we use the commercially available Smartlight photonic processor. An image of the full system with a schematic of the integrated circuit is shown in Fig. 1(a). The processor comprises 17 hexagonal cells for a total of 72 programmable unit cells (PUCs). Each PUC has an insertion loss of 0.48 dB and a power consumption of 1.3 mW/π. Light input and output can be done using any of the 28 available optical ports with a fiber-array that introduces a 3-dB insertion loss per facet. Each of the 28 I/O ports features an opto-electronic monitoring unit with on-chip photodetectors, enabling optical power measurement without external units. The system also includes the electronic circuitry and the software layer used to control each PUC. Further details on the device's performance and elements are available in Ref. 8.

III. UNITARY TRANSFORMATIONS ON A PROGRAMMABLE PHOTONIC PROCESSOR

In this section, we detail the configuration steps required within the programmable photonic processor to achieve vector-matrix multiplications.

A. Splitter tree

To ensure coherence throughout the operation, we employed a continuous-wave (CW) laser. The light was then split into a number of paths equal to the number of matrix inputs. For this work, we focused on matrices of sizes 3 × 3 and 4 × 4. Conventionally, splitter trees for dividing light equally are built using cascaded stages of multi-mode interferometers (MMIs) or directional couplers with a fixed 50:50 coupling ratio. However, in a general-purpose programmable photonic processor, different paths may experience varying optical losses due to the differing number of PUCs traversed by light on each path. To address this challenge, we implemented a correction to the coupling ratio programmed within the PUCs. This correction takes into account the measured insertion losses of each PUC. Assuming that the coupling ratio of light to a path with fewer PUCs is denoted by k, to achieve equal light intensity on both output paths, the following relationship applies:

$$(1-k)IL^{N_P} = k, \tag{4}$$

where IL are the linear insertion losses of the PUC and N_P is the difference in number of PUCs between the two paths. Solving k, we obtain

$$k = \frac{1}{1+\frac{1}{IL^{N_P}}}. \tag{5}$$

Two possible implementations for a 1 × 4 splitter tree are depicted using green light in Fig. 2. Gray PUCs represent a random state except those with a green wave on them that refer to the PUCs used for input vector encoding. Data encoding is explained in Sec. III B, but it is necessary to state that only light going out of one of the two output ports of the encoding PUC is used for the matrix multiplication. The light exiting the other port is routed out of the processor before the multiplication stage. To do so, we set some PUCs in the bar state (red) and others in cross (yellow) depending on the programmed splitter tree. White arrows indicate the travel path of the encoded input vector to be multiplied by the matrix.

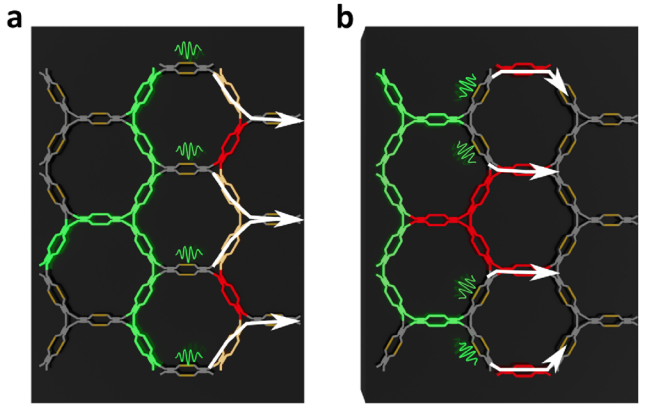

FIG. 2. (a) 1 × 4 symmetric splitter tree and (b) 1 × 4 non-symmetric splitter tree. The PUCs in red are in the bar state, the yellow PUCs are in the cross state, and the green PUCs are in the tunable state with a value dependent on the defined splitter tree. White arrows show the path of the light after the splitting stage.

The symmetric splitter tree in Fig. 2(a) requires three columns of PUCs. First, the input light is divided using a 50:50 coupling. For the second division, we need to be aware that one path will undergo two more PUCs than the other (compare the path difference between the two upper outputs of the splitter tree). Considering the insertion losses of the Smartlight processor and Eq. (5), the coupling ratio in this second stage needs to be adjusted to 56:44.

The non-symmetric splitter tree in Fig. 2(b) is more compact as it only requires two columns. On the other hand, light divided in the first stage undergoes four more PUCs in the lower branch than in the upper. Following the path length compensation, Eq. (5), the coupling ratio must be 61:39. We should note that the PUC in green on the splitter tree may also be in either the bar or cross state. In this configuration, the three middle PUCs in the first column are in the cross state, while the last one is in the bar state. This programming of the PUCs serves two purposes: first, it routes the input light to the two lower modulators, and second, it prevents the formation of resonant structures, allowing the excess light from the input modulators to be directed out of the mesh.

In future large-scale systems, where avoiding resonant structures becomes increasingly complex, the splitting stage can be implemented as a separate, specialized functional block, accessible specifically when the programmable system is employed for matrix multiplication tasks.

B. Encoding

Input vector encoding is performed using an array of PUCs; see the PUCs with a green wave on them in Fig. 2. For the case of

the symmetric splitter tree, we used the *bar* port to encode the data, while for the non-symmetric splitter, we used the *cross* port. In the general case, the input vector is an array of complex numbers. We encode its modulus in the amplitude of the optical signal and its angle (phase) using the term ϕ. Assuming that the light intensity at each input port is normalized to 1, the encoded modulus becomes equal to $\sqrt{k}$ or $\sqrt{1-k}$ depending on whether we are using the *cross* or *bar* ports. Focusing on the *cross* encoding and from Eqs. (1) and (2), we find that

$$\cos\left(\frac{\theta_1-\theta_2}{2}\right)=\sqrt{k}, \tag{6}$$

$$\Delta\Theta=\theta_1-\theta_2=2\ \arccos\left(\sqrt{k}\right). \tag{7}$$

Substituting this expression in the phase term of (1), we obtain

$$\phi=\frac{\theta_1+\theta_2}{2}=\theta_2+\ \arccos\left(\sqrt{k}\right), \tag{8}$$

and finally,

$$\theta_1=\phi+\ \arccos\left(\sqrt{k}\right)-\frac{\pi}{2}, \tag{9}$$

$$\theta_2=\phi-\ \arccos\left(\sqrt{k}\right)-\frac{\pi}{2}. \tag{10}$$

According to Eqs. (9) and (10), there are situations where you need phases of different sign. As we are using thermo-optic phase shifters, only positive or negative (depending on the convention) phases can be applied. To overcome these situations, we can add an extra angle of $2 * \pi$ to both θ_1 and θ_2 to have the same sign while maintaining the total output phase. For the *bar* encoding, the $\arccos\left(\sqrt{k}\right)$ term in Eqs. (9) and (10) is substituted by $\arcsin\left(\sqrt{1-k}\right)$.

C. Building block equivalence

To program arbitrary linear interferometers on our programmable photonic chip, we first establish the equivalence between the building blocks used in each system. As mentioned earlier, the programmable unit cell (PUC) serves as the fundamental building block of the general-purpose photonic processor. In contrast, standard decomposition algorithms for linear interferometers typically assume a Mach–Zehnder interferometer (MZI) with a single internal and a single external phase shifter as the building block; see Fig. 3(a). This MZI has the following transfer function:

$$ie^{i\frac{\theta}{2}}\begin{pmatrix} e^{i\phi}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \\ e^{i\phi}\cos\frac{\theta}{2} & -\sin\frac{\theta}{2}\end{pmatrix}. \tag{11}$$

We can show that if two PUCs are concatenated as in Fig. 3(b), the same transfer function is obtained. First, phases of the left PUC must be equal, $\phi_1=\phi_2$, and according to (1), the transfer function is

$$ie^{i\phi}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \tag{12}$$

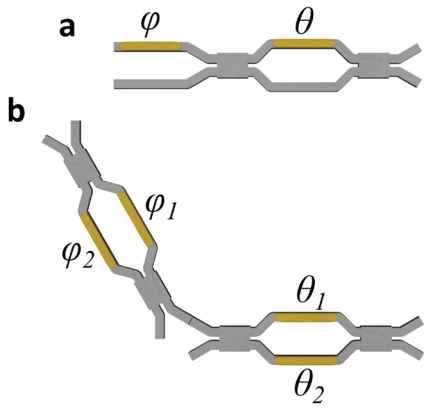

FIG. 3. (a) The standard building block consists of an MZI with one external and one internal phase shifter. (b) Equivalent system that concatenates two PUCs with $\theta_2=0$ and $\phi_1=\phi_2$.

and the PUC is in the cross state, keeping the amplitude constant and adding a phase of $\phi+\frac{\pi}{2}$, which is equivalent to an external phase shifter. Then, the second PUC will be tuned while maintaining $\theta_2=0$. The final linear transformation of the system is consequently the same as in (11).

D. Matrix architecture

We have established that two PUCs can be configured to function equivalently to a Mach–Zehnder interferometer (MZI) with one internal and one external phase shifter, the fundamental building block in many coherent integrated processors. This equivalence allows us to directly translate previously proposed architectures for unitary matrix multiplications on photonic circuits to our general-purpose processor. The use of these architectures in a general-purpose processor introduces extra reconfigurability and loss uniformity to the matrix multipliers, as the connections between tunable elements are performed using MZIs, reducing loss-related errors. These advantages come at the cost of additional insertion losses and electrical elements. The extra complexity of the design and control of the hexagonal processor would be mitigated in a developed photonic ecosystem where their use as a high-level prototyping platform is established. The differences between reconfigurable and feedforward meshes are introduced in Table I. In our case, we show how to implement the rectangular (Clements) and triangular (Reck) architecture. In Fig. 4, we show the distribution of the building blocks (orange boxes) in the Clements [Fig. 4(a)] and in the Reck [Fig. 4(b)] architecture. The blue PUCs represent the equivalent building blocks within the hexagonal mesh. PUCs programmed in cross and bar states ensure that light propagates in a single direction and interacts with other paths only at tunable elements. The Reck architecture necessitates an additional column of horizontal PUCs compared to the Clements topology. As a consequence, in our experiment, the symmetric splitter presented in Sec. III A is not compatible with the Reck architecture using the current mesh size of the Smartlight processor, and we will use the non-symmetric version.

TABLE I. Comprehensive comparison of MZI network architectures.

Parameter	Triangular (Reck)	Rectangular (Clements)	Hexagonal
Loss distribution	Variable across paths	Improved over triangular	Uniform across paths
Reconfigurability	Fixed paths	Fixed paths	Multiple path options. Resilient to failures
MZI Count	$N(N-1)/2$	$N(N-1)/2$	$N(N-1)$[a]
Electrical signals	$N(N-1)$	$N(N-1)$	$2N(N-1)$
Functionalities	Matrix multiplication	Matrix multiplication	Possibility to add extra features

[a]This number corresponds to the number of MZI in the matrix. Extra MZIs kept in a static *cross* or *bar* state are required to route the light along the mesh.

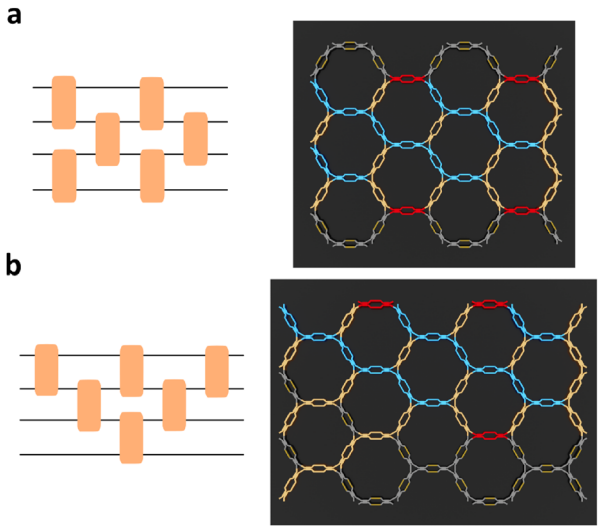

FIG. 4. (a) Schematic of the Clements topology with its translation to the hexagonal mesh and (b) schematic of the Reck topology with its translation to the hexagonal mesh.

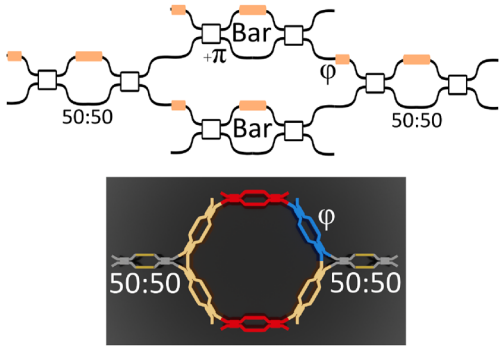

FIG. 5. META-MZI method for the characterization of the passive phase of the phase shifters and its translation to the hexagonal mesh.

E. Phase calibration

As discussed in Sec. II B, the standard calibration method for programmable hexagonal meshes is insufficient for implementing coherent transformations. While it can measure and compensate for the passive phase difference between the top and bottom waveguides of a PUC, it does not account for phase variations arising from different connections between PUCs in a programmed coherent architecture. These additional phase differences can also be treated as a passive phase inherent to the chosen architecture. To address this challenge, we propose a re-calibration procedure that focuses on measuring these inherent passive phase offsets within the architecture. The procedure assumes that the α term in Eq. (3) is still valid and only the θ_0 term should be measured. These offsets will be incorporated as the passive phase of each PUC acting as a phase shifter within the mesh. The re-calibration procedure is adapted from methods used for single-phase actuator calibration in application-specific integrated circuits (ASPICs).[25,34–36] This method relies on the construction of a temporary interferometer using single PUCs. In the ASPIC approach, this structure, also known as a META-MZI, is built using two PUCs configured as 50:50 couplers with all internal PUCs set to the bar state. However, in our general-purpose processor, we have additional PUCs acting as waveguides that are set to the cross state. Once the system is defined, we perform a two-step characterization process for each PUC acting as a phase shifter, Fig. 5.

In the first step, we sweep the current applied to the top arm while keeping the bottom arm current at zero. We measure the output power at the cross port and identify the current corresponding to the maximum power output. This ensures that the PUC is in the cross state. In the second step, we sweep the current of both the top and bottom arms simultaneously. We add the previously measured optimal current as an offset to the top arm current during this sweep. This effectively modifies the phase of the optical signal while maintaining the coupling ratio of the PUC. This phase modification within the created META-MZI results in an interferometric pattern as shown in Figs. 6(a) and 6(b). This measurement can be fitted using Eqs. (1) and (3) to extract the passive phase offset of the phase shifter under test. A step-by-step schematic on how to create the META-MZIs on the programmable mesh for the Clements architecture is presented in Sec. II of the supplementary material. The steps required for the Reck architecture are equivalent.

IV. EXPERIMENTS AND RESULTS

Following re-calibration of the hexagonal mesh, we proceeded to encode unitary matrices on the Smartlight processor. All measurements utilized the same 1550 nm, 10 dBm output power laser source employed for calibration. We programmed the Clements and

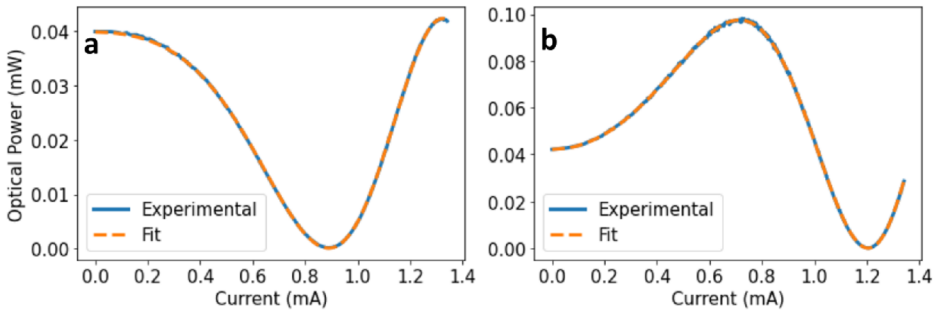

FIG. 6. Optical response of two META-MZI and their fitting curve.

the Reck architecture on the photonic processor using the aforementioned scheme. For each architecture, we tested two matrix sizes: 3 × 3 and 4 × 4. The translation between the unitary matrix weights and the phases that are needed to apply was carried out following the decomposition algorithms provided in Refs. 10 and 11 The decomposition of the 3 × 3 matrices results in the same building block arrangement for both architectures. To differentiate the implementations, we positioned the matrix multipliers in different regions of the mesh. The Clements architecture employed the symmetric splitter tree, while the non-symmetric splitter tree was used with the Reck architecture. During the experiments, for each combination of architecture and matrix size, we generated 1500 random unitary matrices. These matrices were decomposed into the required phase shifts and programmed onto the photonic integrated chip. We measured three figures of merit.

First, we measured the fidelity of the matrices. Fidelity is a common metric in quantum research used to quantify the similarity between two unitary operations. It is calculated using the following equation:

$$\mathscr{F}(U_{\text{exp}}, U) = \frac{\text{Tr}\left(|U^{\dagger}U_{\text{exp}}|^2\right)}{N}, \tag{13}$$

where $U^{\dagger}$ is the adjoint matrix of U, Tr denotes the trace of the matrix, and $|\cdot|^2$ is the element-wise square-modulus of the matrix–matrix multiplication. The highest fidelity (F = 1) indicates

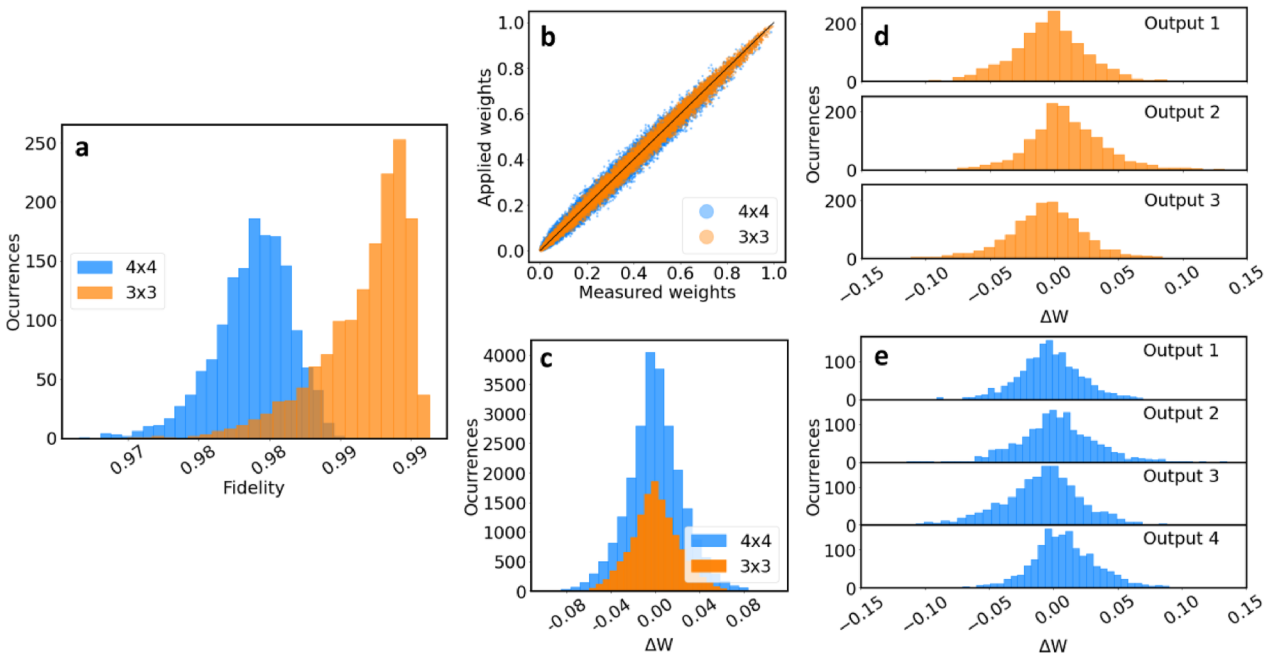

FIG. 7. Results of 1500 random unitary matrices using the Clements architecture with a size of 3 × 3 (orange) and 4 × 4 (blue). (a) Measured fidelity. (b) Comparison between the ideal applied weights and the measured weights. The black line represents the scenario where there are no errors. (c) Difference between the applied and measured weights and (d) and (e) difference between the measured and ideal result of the multiplication of the unitary matrix by a random vector.

TABLE II. Mean and standard deviation of the error at each output of 1500 random vector-matrix multiplications using the Clements architecture.

Size	$O_1(10^{-2})$	$O_2(10^{-2})$	$O_3(10^{-2})$	$O_4(10^{-2})$
3 × 3	0.4 ± 3.2	1.0 ± 3.3	0.6 ± 3.2	⋯
4 × 4	0.3 ± 2.6	0.3 ± 3.1	0.8 ± 3.2	0.9 ± 2.7

perfect similarity between the applied and desired unitary operations. In our chip, we include the splitter tree, input vector, and matrix, allowing for the coherent vector-matrix multiplication. As a result, we encoded in the input vector the complex columns of the adjoint matrix one at a time to perform the matrix–matrix multiplication in (13). The squared-modulus is applied on the photodetection stage. Finally, the trace of the recorded result is calculated to determine the measured fidelity.

Second, we assessed the accuracy of the programmed matrix weights. We encoded in the input vector the *i*th column of the identity matrix, which results in the squared modulus of the *i*th row of the programmed matrix. We repeated this process sequentially for all columns, effectively reconstructing the entire programmed matrix. The measured data are then compared with the ideal matrix, and the r-squared coefficient of the linear regression between the two datasets is calculated to quantify the overall agreement. Additionally, we measure the difference between the measured and ideal weights, calculating the mean and standard deviation of the error. From this measurement, we can approximate the bit precision of our system using the following formula:[37]

$$bit - precision = log2\left(\frac{\max_{weight} - \min_{weight}}{std(err)}\right), \tag{14}$$

where $std(err)$ is the standard deviation of the errors.

Finally, we evaluated the accuracy of on-chip matrix-vector multiplications. We sampled random complex vectors, and they were multiplied by the programmed matrix within the photonic chip. The resulting outputs were measured and compared with the expected outcome of the ideal multiplication. For all measurements, the outputs were normalized by dividing each element by the total measured power.

The fidelity results for the Clements architecture are presented in Fig. 7(a). We achieved an average fidelity of 99.2 ± 0.3 for the 3 × 3 matrix size and a 98.4 ± 0.3 for the 4 × 4 matrix size. In order to verify the stability of those results over time, we measured the fidelity of the same matrix over a 10-hour period obtaining deviations lower than 0.05%. Regarding weight accuracy, Figs. 7(b) and 7(c) show the comparison between the ideal and measured weights. The linear regression returns an r^2 coefficient of 0.992 and 0.984 for the 3 × 3 and 4 × 4, respectively. The standard deviation of the errors between the measured and ideal weights is 0.0215 and 0.0246, corresponding to a bit-precision of 5.5 and 5.35 bits, respectively. The errors for random vector-matrix multiplications (VMMs) are presented in Figs. 7(d) and 7(e) and Table II where we show the error occurred at each output.

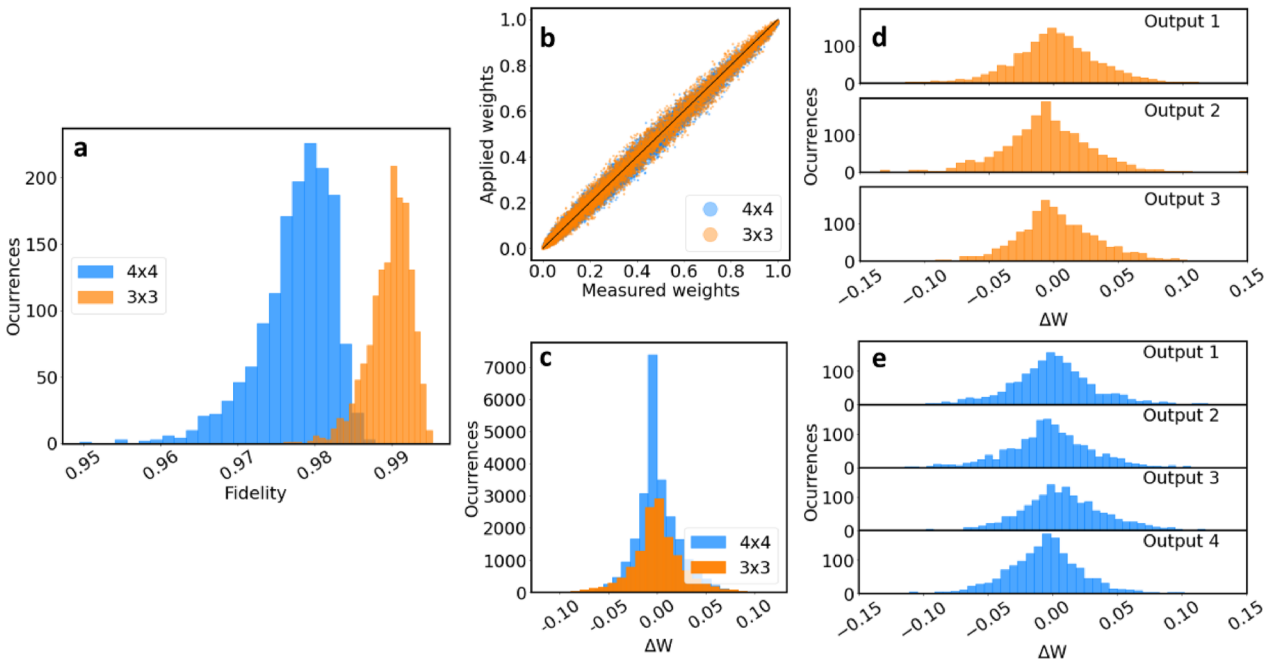

FIG. 8. Results of 1500 random unitary matrices using the Reck architecture with a size of 3 × 3 (orange) and 4 × 4 (blue). (a) Measured fidelity. (b) Comparison between the ideal applied weights and the measured weights. The black line represents the scenario where there are no errors. (c) Difference between the applied and measured weights and (d) and (e) difference between the measured and ideal result of the multiplication of the unitary matrix by a random vector.

TABLE III. Mean and standard deviation of the error at each output of 1500 random vector-matrix multiplications using the Reck architecture.

Size	$O_1(10^{-2})$	$O_2(10^{-2})$	$O_3(10^{-2})$	$O_4(10^{-2})$
3×3	0.2 ± 3.6	0.4 ± 3.6	0.3 ± 3.3	...
4×4	0.04 ± 3.5	0.2 ± 3.4	0.8 ± 3.4	0.6 ± 3.1

For the Reck architecture, the measured fidelities are shown in Fig. 8(a). We obtained an average fidelity with the 3×3 and 4×4 matrices of 99.0 ± 0.3 and 97.8 ± 0.5. The comparison between the applied and measured square-modulus of the complex weights is presented in Figs. 8(b) and 8(c). The r^2 coefficient of the linear regression is 0.993 for both matrix sizes. The standard deviation of the error between the measured and ideal scenario is 0.026 for the 3×3 matrix and 0.0216 for the 4×4 matrix, which translates in a bit-precision of 5.27 and 5.53, respectively. Errors of the random VMMs are presented in Figs. 8(d) and 8(e). The mean and standard deviation are presented in Table III.

A. Quantum gates

Linear photonic integrated circuits have also shown promising results by providing computational advantages in quantum systems.[38] As a final step, we demonstrate the potential of general-purpose programmable photonic circuits in quantum computing applications by programming a set of quantum logic gates and comparing the experimental results with the ideal matrix. We implement the CNOT gate as shown in Eq. (15), the Pauli Y gate extended to a 4×4 matrix as presented in Eq. (16), and the Hadamard gate as shown in 17. The experimental results are illustrated in Fig. 9(a) for the CNOT gate, Fig. 9(b) for the Pauli Y gate, and Fig. 9(c) for the Hadamard gate. We achieved root mean squared errors of 0.020, 0.021, and 0.035, respectively. Regarding fidelity, we achieved highly competitive results of 0.99, 0.99, and 0.97, respectively,

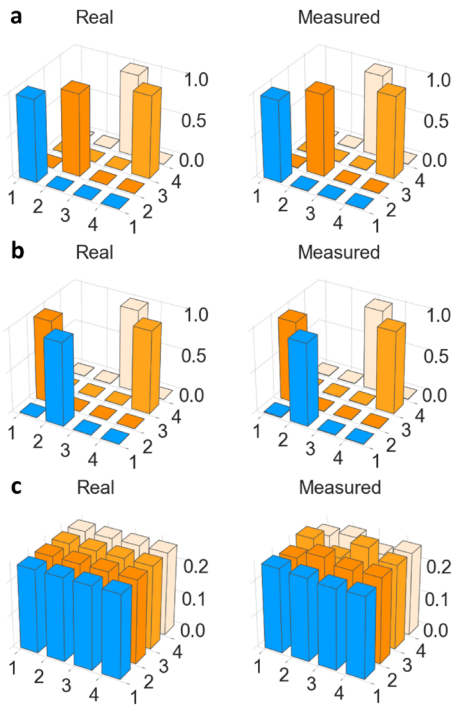

FIG. 9. (a) CNOT gate, (b) Pauli Y gate, and (c) Hadamard gate.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{15}$$

$$\begin{pmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{pmatrix}, \tag{16}$$

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}. \tag{17}$$

V. DISCUSSION AND CONCLUSIONS

In this study, we have experimentally demonstrated the implementation of 3×3 and 4×4 random unitary matrices on a general-purpose programmable photonic processor. This marks, to the best of our knowledge, the first instance where random unitary matrices have been programmed on these types of platforms using both the rectangular and triangular arrangements. The success of this implementation is attributed to the adaptation of the META-MZI algorithm for the calibration of the architecture-specific passive phases and to the construction of a building block mathematically equivalent to the Mach–Zehnder with one internal and one external phase shifter. Our finding indicates that both the rectangular and triangular architectures give similar results in terms of performance. The small difference in fidelity and bit-precision is within the margins of error. The obtained fidelities are compatible with quantum experiments.[36,43] Moreover, five-bit precisions have been demonstrated to be sufficient for quantized deep neural networks,[44–46] highlighting the capability of the processor to support advanced computational tasks. To highlight these statements, we demonstrated the capabilities of the general-purpose processors on quantum tasks by programming the CNOT, Pauli Y, and Hadamard gates, achieving RMSEs of less than 0.035 and fidelities greater than 97%.

While our results are promising, it is important to acknowledge the scalability challenges associated with the system. Compared to application-specific systems, our general-purpose processor exhibits additional losses. In the splitter tree, we use PUCs instead of 3-dB MMI, which present lower insertion losses in the commercially available fabrication process. Furthermore, we have also demonstrated that as the size of the matrix scales, we need to add extra PUCs to the splitter tree. The matrix part also introduces extra losses. We require twice the number of PUCs compared to an ASPIC. Our

TABLE IV. Comparison of modulation technologies.

Effect	Speed	Losses	Footprint	Consumption	References
Thermo-optic	10 s MHz	0.2 dB	<100 μm	20 mW P_π	39
Plasma	40 GHz	4 dB	4 mm	6.3V V_π	40
EO-Pockles BTO	2 GHz	1.3 VdB	1 mm	<100 nW P_π	41
EO TFLN	20 GHz	0.05 dB	5 mm	7.33V V_π	42

system presents 0.48 dB insertion loss for PUC. Reducing this number becomes crucial for large-scale implementations. PUCs based on thermo-optic phase shifters with <0.1 dB of losses have been already demonstrated,[47] opening the path for high-dimensional photonic linear transformers. It is also possible to increase the size of the matrix multiplication by dividing the multiplication process into lower size multiplications using a compiler.[48] A detailed study on how the system scales compared to application-specific feedforward meshes is provided in the supplementary material.

Regarding the power consumption of the system, each PUC in the processor consumes 1.3 mW/π. The splitter tree requires 14 PUCs, which is an extra average consumption of 18.2 mW if compared with an ASPIC. For the matrix stage, we require not only the tunable elements but also PUCs to be in the *bar* and *cross* state for light routing. Moreover, each building block requires four phase shifters instead of the two used in current ASPICs. The total average consumption of the matrix is 54.6 and 61.1 mW for a 4 × 4 matrix size using the Clements and Reck architecture, respectively. These values can be reduced by using non-volatile phase change materials[49] for the static PUCs and by using alternative building blocks for more compact and less power-consuming architectures.[12]

An important metric is the number of multiply-and-accumulate (MAC) operations per second that can be achieved. In our case, we used thermo-optic phase shifters with a switching speed limited to several microseconds for both vector and matrix encoding. As our system can perform complex operations, the number of MAC/s is $MAC/s = 4N^2DR$, where DR is the data rate. Then, our 4 × 4 matrices can provide up to 0.64 GMAC/s. The inclusion of high-speed modulators for the input vector encoding can increase this figure to the TMAC/s range. For example, using a DR of 20 GS/s will translate into 1.28 TMAC/s. No additional changes would be necessary as high-speed photodetectors working up to 40 GHz are already integrated in the processor. The choice of modulation technology should depend on the application requirements, as there are trade-offs between speed, footprint, losses, and power consumption. A comparative analysis of some of these technologies is presented in Table IV.

The use of a general-purpose processor presents significant advantages, particularly in terms of reducing costs associated with design and fabrication. The flexibility of these processors also allows for the integration of extra functionalities, such as photonic filters or delay lines, which has been already combined with linear transformations in ASPICs for different coherent applications.[28,50] Future work could explore how the integration of these systems could impact the performance and precision of the unitary transformation.

SUPPLEMENTARY MATERIAL

The supplementary material includes an in-depth study of the scalability of feedforward meshes on a general-purpose photonic processor, as well as step-by-step details of the phase calibration process.

ACKNOWLEDGMENTS

This work was supported by the H2020-ICT2019-2 Neoteric 871330 project, the European Research Council (ERC) Advanced Grant program under grant Agreement No. 101097092 (ANBIT), the ERC Starting Grant program under grant Agreement No. 101076175 (LS-Photonics Project), the EUR2022-134023 grant funded by CIN/AEI/10.13039/501100011033 and the European Union (NextGenerationEU/PRTR), and the COMCUANTICA/005 and COMCUANTICA/006 grants, funded by the European Union through NextGenerationEU (PRTR-C17) with the support of the Spanish Ministry of Science and Innovation and the Generalitat Valenciana.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

José Roberto Rausell-Campo: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Writing – original draft (lead); Writing – review & editing (equal). **Daniel Pérez-López**: Methodology (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal). **José Capmany Francoy**: Conceptualization (equal); Funding acquisition (lead); Investigation (equal); Supervision (lead); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]W. Bogaerts, D. Pérez, J. Capmany, D. A. B. Miller, J. Poon, D. Englund, F. Morichetti, and A. Melloni, "Programmable photonic circuits," Nature **586**, 207–216 (2020).

[2]D. Pérez, I. Gasulla, and J. Capmany, "Field-programmable photonic arrays," Opt. Express **26**, 27265–27278 (2018).
[3]D. Pérez, I. Gasulla, L. Crudgington, D. J. Thomson, A. Z. Khokhar, K. Li, W. Cao, G. Z. Mashanovich, and J. Capmany, "Multipurpose silicon photonics signal processor core," Nat. Commun. **8**, 636 (2017).
[4]D. Pérez-López, A. M. Gutierrez, E. Sánchez, P. DasMahapatra, and J. Capmany, "Integrated photonic tunable basic units using dual-drive directional couplers," Opt. Express **27**, 38071–38086 (2019).
[5]L. Zhuang, C. G. H. Roeloffzen, M. Hoekman, K.-J. Boller, and A. J. Lowery, "Programmable photonic signal processor chip for radiofrequency applications," Optica **2**, 854–859 (2015).
[6]D. Marpaung, J. Yao, and J. Capmany, "Integrated microwave photonics," Nat. Photonics **13**, 80–90 (2019).
[7]D. A. B. Miller, "Attojoule optoelectronics for low-energy information processing and communications," J. Lightwave Technol. **35**, 346–396 (2017).
[8]D. Pérez-López, A. Gutierrez, D. Sánchez, A. López-Hernández, M. Gutierrez, E. Sánchez-Gomáriz, J. Fernández, A. Cruz, A. Quirós, Z. Xie, J. Benitez, N. Bekesi, A. Santomé, D. Pérez-Galacho, P. DasMahapatra, A. Macho, and J. Capmany, "General-purpose programmable photonic processor for advanced radiofrequency applications," Nat. Commun. **15**, 1563 (2024).
[9]N. C. Harris, J. Carolan, D. Bunandar, M. Prabhu, M. Hochberg, T. Baehr-Jones, M. L. Fanto, A. M. Smith, C. C. Tison, P. M. Alsing, and D. Englund, "Linear programmable nanophotonic processors," Optica **5**, 1623–1631 (2018).
[10]W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walsmley, "Optimal design for universal multiport interferometers," Optica **3**, 1460–1465 (2016).
[11]M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," Phys. Rev. Lett. **73**, 58–61 (1994).
[12]B. A. Bell and I. A. Walmsley, "Further compactifying linear optical unitaries," APL Photonics **6**, 070804 (2021).
[13]N. C. Harris, G. R. Steinbrecher, M. Prabhu, Y. Lahini, J. Mower, D. Bunandar, C. Chen, F. N. C. Wong, T. Baehr-Jones, M. Hochberg, S. Lloyd, and D. Englund, "Quantum transport simulations in a programmable nanophotonic processor," Nat. Photonics **11**, 447–452 (2017).
[14]J. Carolan, C. Harrold, C. Sparrow, E. Martín-López, N. J. Russell, J. W. Silverstone, P. J. Shadbolt, N. Matsuda, M. Oguma, M. Itoh, G. D. Marshall, M. G. Thompson, J. C. F. Matthews, T. Hashimoto, J. L. O'Brien, and A. Laing, "Universal linear optics," Science **349**, 711–716 (2015).
[15]J. Wang, F. Sciarrino, A. Laing, and M. G. Thompson, "Integrated photonic quantum technologies," Nat. Photonics **14**, 273–284 (2020).
[16]C. Taballione, M. C. Anguita, M. de Goede, P. Venderbosch, B. Kassenberg, H. Snijders, N. Kannan, W. L. Vleeshouwers, D. Smith, J. P. Epping, R. van der Meer, P. W. H. Pinkse, H. van den Vlekkert, and J. J. Renema, "20-Mode universal quantum photonic processor," Quantum **7**, 1071 (2023).
[17]A. Annoni, E. Guglielmi, M. Carminati, G. Ferrari, M. Sampietro, D. A. B. Miller, A. Melloni, and F. Morichetti, "Unscrambling light—Automatically undoing strong mixing between modes," Light: Sci. Appl. **6**, e17110 (2017).
[18]K. Choutagunta, I. Roberts, D. A. B. Miller, and J. M. Kahn, "Adapting Mach–Zehnder mesh equalizers in direct-detection mode-division-multiplexed links," J. Lightwave Technol. **38**, 723–735 (2020).
[19]H. Zhou, Y. Zhao, X. Wang, D. Gao, J. Dong, and X. Zhang, "Self-Configuring and reconfigurable silicon photonic signal processor," ACS Photonics **7**, 792–799 (2020).
[20]S. SeyedinNavadeh, M. Milanizadeh, F. Zanetto, G. Ferrari, M. Sampietro, M. Sorel, D. A. B. Miller, A. Melloni, and F. Morichetti, "Determining the optimal communication channels of arbitrary optical systems using integrated photonic processors," Nat. Photonics **18**, 149–155 (2024).
[21]D. A. B. Miller, "Self-configuring universal linear optical component [Invited]," Photonics Res. **1**, 1–15 (2013).
[22]Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," Nat. Photonics **11**, 441–446 (2017).
[23]S. Pai, Z. Sun, T. W. Hughes, T. Park, B. Bartlett, I. A. D. Williamson, M. Minkov, M. Milanizadeh, N. Abebe, F. Morichetti, A. Melloni, S. Fan, O. Solgaard, and D. A. B. Miller, "Experimentally realized in situ backpropagation for deep learning in photonic neural networks," Science **380**, 398–404 (2023).
[24]H. Zhang, M. Gu, X. D. Jiang, J. Thompson, H. Cai, S. Paesani, R. Santagati, A. Laing, Y. Zhang, M. H. Yung, Y. Z. Shi, F. K. Muhammad, G. Q. Lo, X. S. Luo, B. Dong, D. L. Kwong, L. C. Kwek, and A. Q. Liu, "An optical neural chip for implementing complex-valued neural network," Nat. Commun. **12**, 457 (2021).
[25]M. Prabhu, C. Roques-Carmes, Y. Shen, N. Harris, L. Jing, J. Carolan, R. Hamerly, T. Baehr-Jones, M. Hochberg, V. Čeperić, J. D. Joannopoulos, D. R. Englund, and M. Soljačić, "Accelerating recurrent Ising machines in photonic integrated circuits," Optica **7**, 551–558 (2020).
[26]C. Roques-Carmes, Y. Shen, C. Zanoci, M. Prabhu, F. Atieh, L. Jing, T. Dubček, C. Mao, M. R. Johnson, V. Čeperić, J. D. Joannopoulos, D. Englund, and M. Soljačić, "Heuristic recurrent algorithms for photonic Ising machines," Nat. Commun. **11**, 249 (2020).
[27]W. Zhang, A. Tait, C. Huang, T. Ferreira de Lima, S. Bilodeau, E. C. Blow, A. Jha, B. J. Shastri, and P. Prucnal, "Broadband physical layer cognitive radio with an integrated photonic processor for blind source separation," Nat. Commun. **14**, 1107 (2023).
[28]P. M.-C. Romero, J. R. Rausell-Campo, D. Pérez-Galacho, X. Li, T. Qing, T. Wang, and D. Pérez-López, "Integrated microwave photonics coherent processor for massive-MIMO systems in wireless communications," IEEE J. Sel. Top. Quantum Electron. **29**, 6101112 (2023).
[29]W. Zhang, J. C. Lederman, T. Ferreira de Lima, J. Zhang, S. Bilodeau, L. Hudson, A. Tait, B. J. Shastri, and P. R. Prucnal, "A system-on-chip microwave photonic processor solves dynamic RF interference in real time with picosecond latency," Light: Sci. Appl. **13**, 14 (2024).
[30]S. Bandyopadhyay, A. Sludds, S. Krastanov, R. Hamerly, N. Harris, D. Bunandar, *et al.*, "Single chip photonic deep neural network with accelerated training," Nat. Photon. **18** (2024) 1335–1343 https://doi.org/10.1038/s41566-024-01567-z
[31]See https://ipronics.com/ for more information about the Smartlight photonic processor.
[32]A. López-Hernández, M. Gutiérrez-Zubillaga, and D. Pérez-López, "Automatic self-calibration of programmable photonic processors," in *2022 IEEE Photonics Conference (IPC)* (IEEE, 2022), pp. 1–2.
[33]A. López, D. Pérez, P. DasMahapatra, and J. Capmany, "Auto-routing algorithm for field-programmable photonic gate arrays," Opt. Express **28**, 737–752 (2020).
[34]S. Lin, Y. Zhang, Z. Wu, S. Zeng, Q. Gao, J. Li, X. Yu, and S. Yu, "Power-efficient programmable integrated multiport photonic interferometer in CMOS-compatible silicon nitride," Photonics Res. **12**, A11–A20 (2024).
[35]C. Alexiev, J. C. C. Mak, W. D. Sacher, and J. K. S. Poon, "Calibrating rectangular interferometer meshes with external photodetectors," OSA Continuum **4**, 2892–2904 (2021).
[36]C. Pentangelo, N. Di Giano, S. Piacentini, R. Arpe, F. Ceccarelli, A. Crespi, and R. Osellame, "High-fidelity and polarization-insensitive universal photonic processors fabricated by femtosecond laser writing," Nanophotonics **13**, 2259–2270 (2024).
[37]W. Zhang, C. Huang, H.-T. Peng, S. Bilodeau, A. Jha, E. Blow, T. F. de Lima, B. J. Shastri, and P. Prucnal, "Silicon microring synapses enable photonic deep learning beyond 9-bit precision," Optica **9**, 579–584 (2022).
[38]L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, A. E. Lita, T. Gerrits, S. W. Nam, V. D. Vaidya, M. Menotti, I. Dhand, Z. Vernon, N. Quesada, and J. Lavoie, "Quantum computational advantage with a programmable photonic processor," Nature **606**, 75–81 (2022).
[39]N. C. Harris, Y. Ma, J. Mower, T. Baehr-Jones, D. Englund, M. Hochberg, and C. Galland, "Efficient, compact and low loss thermo-optic phase shifter in silicon," Opt. Express **22**, 10487–10493 (2014).
[40]S. Y. Siew, B. Li, F. Gao, H. Y. Zheng, W. Zhang, P. Guo, S. W. Xie, A. Song, B. Dong, L. W. Luo, C. Li, X. Luo, and G. Q. Lo, "Review of silicon photonics technology and platform development," J. Lightwave Technol. **39**, 4374–4389 (2021).

[41] F. Eltes, C. Mai, D. Caimi, M. Kroh, Y. Popoff, G. Winzer, D. Petousi, S. Lischke, J. E. Ortmann, L. Czornomaz, L. Zimmermann, J. Fompeyrine, and S. Abel, "A $BaTiO_3$-based electro-optic Pockels modulator monolithically integrated on an advanced silicon photonics platform," J. Lightwave Technol. **37**, 1456–1462 (2019).
[42] Y. Zheng, R. Wu, Y. Ren, R. Bao, J. Liu, Y. Ma, M. Wang, and Y. Cheng, "Photonic neural network fabricated on thin film lithium niobate for high-fidelity and power-efficient matrix computation," Laser Photonics Rev. **18**, 2400565 (2024).
[43] N. Maring, A. Fyrillas, M. Pont, E. Ivanov, P. Stepanov, N. Margaria, W. Hease, A. Pishchagin, A. Lemaître, I. Sagnes, T. H. Au, S. Boissier, E. Bertasi, A. Baert, M. Valdivia, M. Billard, O. Acar, A. Brieussel, R. Mezher, S. C. Wein, A. Salavrakos, P. Sinnott, D. A. Fioretto, P.-E. Emeriau, N. Belabas, S. Mansfield, P. Senellart, J. Senellart, and N. Somaschi, "A versatile single-photon-based quantum computing platform," Nat. Photonics **18**, 603 (2024).
[44] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei, "The Era of 1-bit LLMs: All large language models are in 1.58 bits," arXiv:2402.17764 [cs.CL] (2024).
[45] H. Li, S. De, Z. Xu, C. Studer, H. Samet, and T. Goldstein, "Training quantized nets: A deeper understanding," in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17* (Curran Associates, Inc., Red Hook, NY, 2017), pp. 5813–5823.
[46] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Salt Lake City, UT, USA* (IEEE, 2018), pp. 2704–2713.
[47] N. C. Harris, D. Bunandar, A. Joshi, A. Basumallik, and R. Turner, "Passage: A wafer-scale programmable photonic communication substrate," in *2022 IEEE Hot Chips 34 Symposium (HCS)* (IEEE, 2022), pp. 1–26.
[48] Z. Guo, A. N. Tait, B. A. Marquez, M. Filipovich, H. Morison, P. R. Prucnal, L. Chrostowski, S. Shekhar, and B. J. Shastri, "Multi-level encoding and decoding in a scalable photonic tensor processor with a photonic general matrix multiply (GeMM) compiler," IEEE J. Sel. Top. Quantum Electron. **28**, 8300714 (2022).
[49] M. Wuttig, H. Bhaskaran, and T. Taubner, "Phase-change materials for non-volatile photonic applications," Nat. Photonics **11**, 465–476 (2017).
[50] M. Nakajima, K. Tanaka, and T. Hashimoto, "Scalable reservoir computing on coherent linear photonic processor," Commun. Phys. **4**, 20 (2021).