Supplementary Information

Self-configuring complex-valued photonic convolution accelerator based on cascaded Mach–Zehnder interferometers: supplemental document

Chaoyi Li[1], Shuai Meng[1], Xudong Zhou[1], Yingjie Hong[1], Wenrui Wu[1], Huifu Xiao[1,*], and Yonghui Tian[1,2,*]

[1]School of Physical Science and Technology, Lanzhou University, Lanzhou, 730000, Gansu, China.
[2]School of Mathematics and Physics, North China Electric Power University, Beijing, 102206, China.

[*]Corresponding authors: xiaohf@lzu.edu.cn; tianyh@lzu.edu.cn

Abstract

This supplementary document provides additional details on chip fabrication, experimental implementation, complex-valued input loading, coherent complex-valued output reconstruction, the closed-loop self-configuration algorithm, and the neural-network training and task-level validation procedure used in this work.

Supplementary information contents

Supplementary Note 1 Chip fabrication details

The photonic chip was fabricated on a silicon-on-insulator (SOI) platform with a 220-nm silicon device layer and a 2-μm oxide cladding layer. The chip was fabricated by the Shanghai Industrial Technology Research Institute (SITRI) and packaged by the SJTU-Pinghu Institute of Intelligent Optoelectronics. During chip layout, thermal isolation trenches were introduced to mitigate thermal crosstalk between adjacent heaters, and fabrication-tolerant bent waveguides and MMI structures were adopted to reduce the sensitivity of the optical paths and coupling regions to process variations. A 1550-nm laser was coupled into and out of the chip through edge couplers, with a coupling loss of approximately 3.5 dB per facet. The MZI network consisted of 16 MZI-based tunable building blocks arranged in an SVD-based Clements-type cascaded topology. Among them, 6 MZI cells implemented the unitary matrix U, 4 MZI-based tunable diagonal weighting elements implemented the diagonal matrix Σ, and the remaining 6 MZI cells implemented the unitary matrix V^{H}. All thermo-optic phase shifters in the MZI network were tuned using TiN heaters. The resistance of each heater was approximately 280 Ω, and the half-wave voltage V_π of the phase shifter was approximately 3.9 V. Each heater was connected to the chip pads through on-chip TiN/Al metal routing, and the electrical signals were finally routed out through an IPEX interposer board.

Supplementary Note 2 Experimental details

The optical source was provided by a multi-channel tunable laser (T1-MS64-12CV), delivering a 1550-nm continuous-wave laser with an output power of 13 dBm. The input signal to be processed was generated by an arbitrary waveform generator (Keysight M8196A) and loaded into an IQ modulator (AP-TFLN-IQ-C-40-0PPS). The sampling rate of the AWG was 92 GSa/s. A polarization controller was used to maximize the coupling efficiency between the optical source and the SCPCA chip. The chip temperature was maintained at 25 °C. All tunable building blocks were reconfigured through thermo-optic tuning driven by a silicon photonic electrical control module (G-128-E-B-V1). After on-chip processing, the reference light and the output signal were combined by a fiber-based 3-dB MMI for coherent detection. The interfered signals were detected by a balanced photodetector (KG-BPD-40-SM-FA) and then captured by a sampling oscilloscope (Keysight 86100D).

Supplementary Note 3 Technical details and theoretical derivation of the input module

To realize complex-valued convolution, the on-chip computing core must accept complex-valued input signals. Let the input complex signal be written as

$$x(t) = x_I(t) + jx_Q(t), \tag{S1}$$

where $x_I(t)$ and $x_Q(t)$ denote the real and imaginary parts of the input signal, respectively. The role of the IQ modulator is to map these two electrical signals onto two orthogonal components of the same optical carrier, thereby generating the corresponding complex optical envelope[1, 2].

Let the input optical carrier be

$$E_{\text{in}}(t) = E_0 e^{j\omega_0 t}, \tag{S2}$$

where E_0 is the carrier amplitude and ω_0 is the optical carrier angular frequency. The IQ modulator typically consists of an in-phase (I) branch and a quadrature (Q) branch, which are driven by $x_I(t)$ and $x_Q(t)$, respectively[1]. The I branch corresponds to the in-phase component, whereas the Q branch corresponds to the quadrature component. A fixed phase offset of 90° is imposed between them at recombination[3]. Therefore, under proper biasing and electrical normalization or predistortion, the IQ modulator can be operated in an approximately linear regime, and the output optical envelope can be expressed as

$$\tilde{E}_{\text{out}}(t) \propto x_I(t) + jx_Q(t), \tag{S3}$$

or equivalently,

$$\tilde{E}_{\text{out}}(t) \propto x(t). \tag{S4}$$

This relation indicates that the IQ modulator directly maps the complex signal $x(t)$ onto the optical field envelope, where the real part is encoded on the in-phase component and the imaginary part is encoded on the quadrature component. Together, they form a complete complex optical signal[1, 2].

If the input signal is further written in polar form as

$$x(t) = A(t)e^{j\varphi(t)}, \tag{S5}$$

then the output optical envelope can be equivalently expressed as

$$\tilde{E}_{\text{out}}(t) \propto A(t)e^{j\varphi(t)}. \tag{S6}$$

Therefore, the IQ modulator does not process the real and imaginary parts as two independent optical signals. Instead, it directly constructs a complex optical waveform with the desired amplitude and phase on a single optical carrier. This input mechanism is naturally compatible with subsequent interference-based complex-valued matrix operations and complex-valued convolution.

Supplementary Note 4 Technical details and theoretical derivation of coherent readout

To recover complex-valued information from the on-chip output field, both the real and imaginary parts of the optical field must be measured. Since direct photodetection can only measure optical intensity and cannot directly recover phase, this work adopts a reference-assisted coherent readout scheme[4, 5]. Specifically, the chip output field E_s and a reference field E_r are jointly fed into a 2×2 3-dB MMI. The differential signal between the two output ports is then measured by a balanced photodetector[6]. By repeating the measurement after introducing an additional 90° phase shift in the reference arm, the real and imaginary parts of the output signal can be extracted separately, thereby enabling full reconstruction of the complex output[7].

Let the output signal field be

$$E_s = E_x + jE_y, \tag{S7}$$

where $E_x = \text{Re}(E_s)$ and $E_y = \text{Im}(E_s)$. The reference field is written as

$$E_r = A_r e^{j\theta_r}, \tag{S8}$$

where A_r is the reference amplitude and θ_r is the reference phase. Under the chosen port and phase convention, an ideal 3-dB MMI can be represented as

$$E_+ = \frac{E_s + E_r}{\sqrt{2}}, \qquad E_- = \frac{E_s - E_r}{\sqrt{2}}. \tag{S9}$$

If the detector responsivity is denoted by R, the differential photocurrent of the balanced detector is

$$I_{\text{diff}} = R\left(|E_+|^2 - |E_-|^2\right) = 2R\,\text{Re}(E_s E_r^*)\,. \tag{S10}$$

Substituting the reference field into the above expression yields

$$I_{\text{diff}}(\theta_r) = 2RA_r\,\text{Re}\Big(E_s e^{-j\theta_r}\Big)\,. \tag{S11}$$

Using $E_s = E_x + jE_y$, one obtains

$$I_{\text{diff}}(\theta_r) = 2RA_r\left(E_x\cos\theta_r + E_y\sin\theta_r\right). \tag{S12}$$

This expression shows that the differential output corresponds to the projection of the complex signal amplitude onto the direction defined by the reference phase[8]. When the reference is in phase with the signal, namely $\theta_r = 0$, the first differential measurement is

$$I_1 = 2RA_r\,\text{Re}(E_s), \tag{S13}$$

from which the real part of the output field is obtained as

$$\mathrm{Re}(E_s) = \frac{I_1}{2RA_r}. \tag{S14}$$

Next, an additional 90° phase shift is introduced into the reference arm, yielding the second differential measurement

$$I_2 = 2RA_r\,\mathrm{Im}(E_s), \tag{S15}$$

from which the imaginary part is recovered as

$$\mathrm{Im}(E_s) = \frac{I_2}{2RA_r}. \tag{S16}$$

Therefore, the reconstructed complex output field can be expressed as

$$E_s = \frac{I_1}{2RA_r} + j\frac{I_2}{2RA_r} = \frac{I_1 + jI_2}{2RA_r}. \tag{S17}$$

In practical measurements, the signs and gains of the two reconstructed quadratures depend on the port definition, differential detection polarity, and reference-phase convention, and can therefore be treated as fixed readout factors in the measured system response.

It should be emphasized that the above derivation assumes an ideal reference-phase basis and an explicitly known readout gain. In practice, however, the coherent readout chain usually introduces fixed but generally unknown amplitude and phase factors. The amplitude factor may originate from the reference power, detector responsivity, electrical gain, and channel-dependent loss in the readout path, whereas the phase factor may arise from the initial phase difference between the reference and signal arms, additional optical path differences, and other static phase delays in the readout chain[8]. For the m-th output channel, these fixed readout factors can be written as a complex coefficient

$$c_m = \alpha_m e^{-j\phi_m}, \tag{S18}$$

where α_m denotes the fixed amplitude coefficient and ϕ_m denotes the fixed phase offset. The coherently recovered output is then not the ideal $E_{s,m}$, but rather

$$\hat{E}_{s,m} = c_m E_{s,m} = \alpha_m E_{s,m} e^{-j\phi_m}. \tag{S19}$$

Stacking all output channels into vector form gives

$$\hat{\boldsymbol{E}}_s = \boldsymbol{D}_c \boldsymbol{E}_s, \tag{S20}$$

where

$$\boldsymbol{D}_c = \mathrm{diag}(c_1, c_2, \ldots, c_N) = \mathrm{diag}\Big(\alpha_1 e^{-j\phi_1}, \alpha_2 e^{-j\phi_2}, \ldots, \alpha_N e^{-j\phi_N}\Big) \tag{S21}$$

is a diagonal matrix composed of fixed complex readout coefficients.

In actual measurements, the fixed amplitude and phase factors introduced by the coherent readout chain are not separately removed from the system. Instead, they are naturally absorbed into the overall system response observed by the optimization algorithm together with the transmission matrix of the cascaded MZI network itself. If the ideal transmission matrix of the cascaded MZI network under control vector $\boldsymbol{p}$ is denoted by $\boldsymbol{T}_{\mathrm{MZI}}(\boldsymbol{p})$, then the experimentally measured system response should be written as

$$\boldsymbol{Q}(\boldsymbol{p}) = \boldsymbol{D}_c \boldsymbol{T}_{\mathrm{MZI}}(\boldsymbol{p}). \tag{S22}$$

In the self-configuration procedure, the measured response matrix is normalized before constructing the residual, for example,

$$\boldsymbol{Q}_n(\boldsymbol{p}) = \frac{\boldsymbol{Q}(\boldsymbol{p})}{\|\boldsymbol{Q}(\boldsymbol{p})\|_F}, \qquad \boldsymbol{W}_n = \frac{\boldsymbol{W}}{\|\boldsymbol{W}\|_F}. \tag{S23}$$

Therefore, the influence of a readout-induced global amplitude factor is removed by the Frobenius normalization. Channel-dependent fixed complex coefficients, however, are not eliminated by normalization. Instead, their relative amplitude and phase effects remain included in the measured response $\boldsymbol{Q}(\boldsymbol{p})$ and are compensated, within the reachable range of the hardware, by the closed-loop self-configuration process.

Supplementary Note 5 Implementation details of the self-configuration algorithm

The main text has already introduced the normalized residual construction and the bound-constrained Levenberg–Marquardt (LM) formulation for matrix mapping. To avoid repetition, this section focuses on several implementation details that are essential for practical closed-loop programming but are not expanded in the main text, including Jacobian estimation from measured data, feasibility handling for phase variables, damping adaptation, and stopping criteria.

Jacobian estimation from in-loop measurements. In the experiment, the chip response is affected by fabrication imperfections, thermal crosstalk, loss imbalance, and readout drift. As a result, the local sensitivity of the residual vector with respect to the phase-control vector is estimated directly from measured responses rather than from an ideal analytical transfer model. For the current control vector $\boldsymbol{p}_k$, the boundary-safe perturbed point for the i-th control variable is first written as

$$\boldsymbol{p}_k^{(i,+)} = R(\boldsymbol{p}_k + h_i \boldsymbol{e}_i), \tag{S24}$$

where h_i is a small trial perturbation, $\boldsymbol{e}_i$ is the corresponding standard basis vector, and $R(\cdot)$ denotes the reflective boundary operator defined below. The actual perturbation applied to the i-th coordinate is

$$\Delta p_{k,i} = \left[\boldsymbol{p}_k^{(i,+)} - \boldsymbol{p}_k\right]_i. \tag{S25}$$

When $\Delta p_{k,i} \neq 0$, the i-th column of the Jacobian is estimated as

$$\boldsymbol{J}_{:,i}(\boldsymbol{p}_k) \approx \frac{\boldsymbol{F}(\boldsymbol{p}_k^{(i,+)}) - \boldsymbol{F}(\boldsymbol{p}_k)}{\Delta p_{k,i}}. \tag{S26}$$

In this way, the finite-difference denominator corresponds to the actual feasible perturbation after boundary handling rather than the nominal trial step. The Jacobian is therefore obtained from the actual measured input–output behavior of the chip and implicitly includes nonideal effects that are difficult to model explicitly.

Reflective boundary handling. In the implementation, each phase-control variable is constrained by its experimentally accessible tuning range rather than by an exactly prescribed interval of $[0, 2\pi]$. The only requirement is that the available tuning span of each control channel covers at least one full 2π phase cycle. Therefore, the feasible domain is written as

$$\Omega = \left\{\boldsymbol{p} \in \mathbb{R}^P \,\middle|\, p_i^{\min} \leq p_i \leq p_i^{\max},\ \Delta_i = p_i^{\max} - p_i^{\min} \geq \Delta_{2\pi,i},\ i = 1, \ldots, P\right\}, \tag{S27}$$

where $\Delta_{2\pi,i}$ denotes the control-variable span required to induce one complete 2π phase shift in the i-th tuning channel. This definition is applicable whether the control variable is represented as an equivalent phase or as a driving voltage.

Instead of direct clipping, a reflective boundary rule is used so that the update remains continuous when a variable crosses its feasible interval. For the i-th scalar variable x, define

$$\bar{x}_i = \mathrm{mod}\big(x - p_i^{\min},\, 2\Delta_i\big)\,, \qquad \bar{x}_i \in [0, 2\Delta_i), \tag{S28}$$

and then map it back to $[p_i^{\min}, p_i^{\max}]$ by

$$r_i(x) = \begin{cases} p_i^{\min} + \bar{x}_i, & 0 \le \bar{x}_i \le \Delta_i, \\ p_i^{\max} - (\bar{x}_i - \Delta_i)\,, & \Delta_i < \bar{x}_i < 2\Delta_i. \end{cases} \tag{S29}$$

The vector form is

$$R(\boldsymbol{p}) = [r_1(p_1),\, r_2(p_2),\, \ldots,\, r_P(p_P)]^T\,. \tag{S30}$$

This treatment avoids artificial accumulation at the boundaries and keeps both the trial update and the finite-difference perturbation within the experimentally valid control range. Meanwhile, it does not require the boundary values themselves to coincide with an exact 2π phase interval; it only requires the accessible range to span at least one complete phase period.

Closed-loop update and damping adaptation. After obtaining the Jacobian, the LM step $\boldsymbol{s}_k$ is computed from

$$\big(\boldsymbol{J}_k^T \boldsymbol{J}_k + \mu_k \boldsymbol{I}\big)\, \boldsymbol{s}_k = -\boldsymbol{J}_k^T \boldsymbol{F}_k, \tag{S31}$$

where $\boldsymbol{F}_k = \boldsymbol{F}(\boldsymbol{p}_k)$, $\boldsymbol{J}_k = \boldsymbol{J}(\boldsymbol{p}_k)$, and $\mu_k > 0$ is the damping factor. The unconstrained candidate vector is first written as

$$\tilde{\boldsymbol{p}}_{k+1} = \boldsymbol{p}_k + \boldsymbol{s}_k, \tag{S32}$$

and the boundary-feasible trial vector is then obtained through the reflective operator,

$$\boldsymbol{p}_{k+1}^{\mathrm{trial}} = R(\tilde{\boldsymbol{p}}_{k+1}). \tag{S33}$$

The actual feasible step after boundary handling is therefore

$$\boldsymbol{d}_k = \boldsymbol{p}_{k+1}^{\mathrm{trial}} - \boldsymbol{p}_k. \tag{S34}$$

The trial control vector is written to the chip, and a new response measurement is taken to evaluate the actual reduction in the objective function. The objective function is defined as

$$f(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{F}(\boldsymbol{p})\|_2^2. \tag{S35}$$

To quantify the reliability of the local linear model, the gain ratio is defined as

$$\rho_k = \frac{f(\boldsymbol{p}_k) - f(\boldsymbol{p}_{k+1}^{\mathrm{trial}})}{\frac{1}{2}\|\boldsymbol{F}_k\|_2^2 - \frac{1}{2}\|\boldsymbol{F}_k + \boldsymbol{J}_k \boldsymbol{d}_k\|_2^2}. \tag{S36}$$

A large ρ_k indicates that the measured reduction agrees well with the model-predicted reduction. In this case, the trial update is accepted, $\boldsymbol{p}_{k+1} = \boldsymbol{p}_{k+1}^{\mathrm{trial}}$, and the damping factor is decreased. By contrast, a small or negative ρ_k implies that the local model is less reliable. In this case, the trial update is rejected, $\boldsymbol{p}_{k+1} = \boldsymbol{p}_k$, and the damping factor is increased so that the subsequent update becomes more conservative. This adaptive mechanism improves stability when the chip operates in a highly nonlinear or drifting regime.

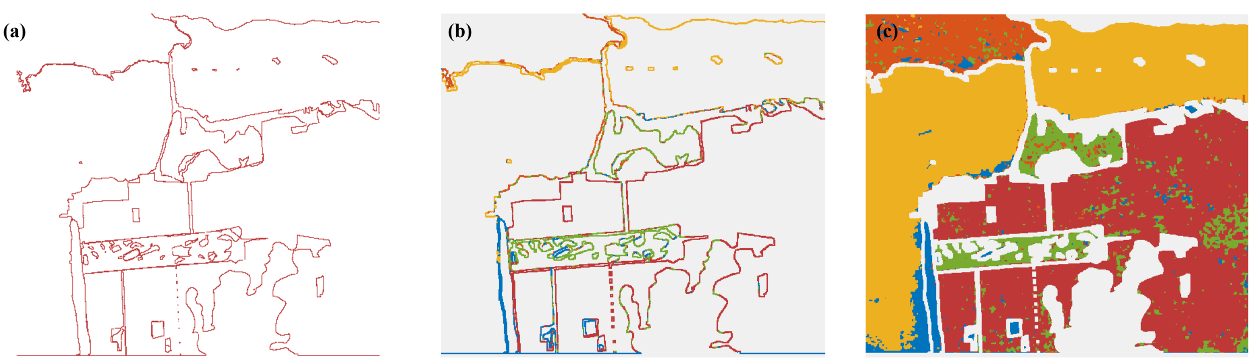
Stopping criteria and final solution. The closed-loop iteration is terminated when one of the following conditions is satisfied: the residual norm $\|\boldsymbol{F}_k\|_2$ falls below a preset tolerance, the gradient-related quantity $\|\boldsymbol{J}_k^T \boldsymbol{F}_k\|_\infty$ falls below a preset threshold, the actual feasible step norm $\|\boldsymbol{d}_k\|_2$ becomes sufficiently small, or the maximum iteration number is reached. The final control vector is denoted by $\boldsymbol{p}^*$, and the corresponding normalized measured response $\boldsymbol{Q}_n(\boldsymbol{p}^*)$ is taken as the experimentally realized matrix.

Supplementary Note 6 Neural-network training and task-level validation

For the PolSAR San Francisco classification task, we employed a compact complex-valued convolutional neural network that directly processed the six independent elements of the Hermitian coherency matrix, namely T_{11}, T_{12}, T_{13}, T_{22}, T_{23}, and T_{33}, represented in a complex-valued tensor format. Each input sample was constructed as a complex-valued local patch of size $12 \times 12 \times 6$. In our implementation, the network consisted of two cascaded complex-valued convolutional layers. Each convolution layer was followed by complex batch normalization and split ReLU activation, and an average-pooling layer of size 2×2 was inserted between the two convolutional layers[9]. The kernel size of both convolutional layers was 3×3, and the numbers of output channels were 32 and 64, respectively. After the second convolutional layer, global average pooling was applied to obtain a 64-dimensional complex feature vector. The real and imaginary parts were then concatenated into a 128-dimensional real-valued vector, which was finally fed into a linear classifier to generate the class prediction.

During training, the Adam optimizer was adopted with a batch size of 128, a learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay factor of 10^{-4}[10]. The validation accuracy was evaluated after each epoch. During testing, the classification accuracy and confusion matrix were calculated over the labeled region of interest (ROI).

Under this network configuration, the purely electronic implementation achieved an overall classification accuracy of 86%. More specifically, the classification accuracy at class boundaries was 54.53%, whereas the accuracy inside the interior regions of different classes reached 96.08%. The corresponding classification result is shown in Supplementary Fig. 1.



Supplementary Fig. 1: Region-wise evaluation of classification performance. (a) Definition of the boundary and interior regions in the test area. (b) Classification accuracy for samples located in the boundary region. (c) Classification accuracy for samples located in the interior region.

In the optical experiment, the linear convolution operations in the first convolutional layer were implemented using the photonic complex-valued convolution accelerator, while the remaining network modules were still executed digitally. The final end-to-end classification accuracy was 84.98%. This result indicates that, in a real PolSAR classification task, replacing the first-layer convolution computation with the photonic accelerator caused only a very small performance degradation, thereby validating the practical task-level applicability of the proposed accelerator.

To quantify the necessity of convolution acceleration, we counted the arithmetic complexity of a single forward inference on one $12 \times 12 \times 6$ complex-valued input patch under a unified operation-counting rule. Under this convention, the first convolutional layer required 172,800 complex multiply-accumulate (MAC) operations, corresponding to 1,382,400 real-valued arithmetic operations in an electronic system. The second convolutional layer required 165,888 complex MAC operations, corresponding to 1,327,104 real-valued arithmetic operations. Therefore, the two convolutional layers alone required 2,709,504 real-valued arithmetic operations.

By comparison, the first complex batch-normalization layer required approximately 25,600 real-valued operations, the first split ReLU required approximately 6,400 operations, the average-pooling layer required approximately 6,400 operations, the second complex batch-normalization layer required approximately 4,608 operations, the second split ReLU required approximately 1,152 operations, the global average-pooling layer required approximately 1,152 operations, and the final linear classifier required approximately 1,280 operations. In other words, all non-convolution modules together accounted for only about 46,592 real-valued arithmetic operations. Under this unified counting rule, one complete forward inference of the entire network required approximately 2,756,096 real-valued arithmetic operations in total, among which the convolution layers accounted for 98.31%.

This result indicates that, in this complex-valued CNN, nearly the entire inference burden is concentrated in the convolution layers. Convolution is therefore the most critical and most worthwhile target for acceleration.

References

[1] Dong, P. *et al.* In-phase/quadrature modulation using directly reflectivity-modulated laser. *Optica* **7**, 929–933 (2020). URL https://opg.optica.org/optica/abstract.cfm?URI=optica-7-8-929.

[2] Drayss, D. *et al.* Optical arbitrary waveform generation (OAWG) using actively phase-stabilized spectral stitching. *Light: Science & Applications* **14**, 353 (2025).

[3] Xu, M. *et al.* High-performance coherent optical modulators based on thin-film lithium niobate platform. *Nature Communications* **11**, 3911 (2020).

[4] Ip, E., Lau, A. P. T., Barros, D. J. F. & Kahn, J. M. Coherent detection in optical fiber systems. *Optics Express* **16**, 753–791 (2008). URL https://opg.optica.org/oe/abstract.cfm?URI=oe-16-2-753.

[5] Ly-Gagnon, D.-S., Tsukamoto, S., Katoh, K. & Kikuchi, K. Coherent detection of optical quadrature phase-shift keying signals with carrier phase estimation. *Journal of Lightwave Technology* **24**, 12–21 (2006). URL https://opg.optica.org/jlt/abstract.cfm?URI=jlt-24-1-12.

[6] Painchaud, Y., Poulin, M., Morin, M. & Têtu, M. Performance of balanced detection in a coherent receiver. *Optics Express* **17**, 3659–3672 (2009). URL https://opg.optica.org/oe/abstract.cfm?URI=oe-17-5-3659.

[7] Wang, Y. *et al.* Ultrahigh-speed graphene-based optical coherent receiver. *Nature Communications* **12**, 5076 (2021).

[8] Skahan, P. J., Gundavarapu, S., Nguyen, K. N., Baney, D. M. & Blumenthal, D. J. Monolithically integrated dual-channel coherent receiver with widely tunable local oscillator for 100 Gbps dual-polarization quadrature phase shift keying applications. *Optics Letters* **40**, 4313–4316 (2015). URL https://opg.optica.org/ol/abstract.cfm?URI=ol-40-18-4313.

[9] Lee, C. Y., Hasegawa, H. & Gao, S. Complex-valued neural networks: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica* **9**, 1406–1426 (2022).

[10] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization (2017). URL https://arxiv.org/abs/1412.6980. arXiv:1412.6980.