Calibration of Multiparameter Sensors via Machine Learning at the Single-Photon Level

Valeria Cimini,[1,†] Emanuele Polino,[2,†] Mauro Valeri,[2,†] Ilaria Gianani,[1] Nicolò Spagnolo,[2] Giacomo Corrielli,[3,4] Andrea Crespi,[4,3] Roberto Osellame,[3,4] Marco Barbieri,[1,5] and Fabio Sciarrino[2,*]

[1] *Dipartimento di Scienze, Università degli Studi Roma Tre, Via della Vasca Navale 84, Rome 00146, Italy*
[2] *Dipartimento di Fisica, Sapienza Università di Roma, Piazzale Aldo Moro 5, Roma I-00185, Italy*
[3] *Istituto di Fotonica e Nanotecnologie, Consiglio Nazionale delle Ricerche (IFN-CNR), Piazza Leonardo da Vinci, 32, Milano I-20133, Italy*
[4] *Dipartimento di Fisica, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano I-20133, Italy*
[5] *Istituto Nazionale di Ottica—CNR, Largo Enrico Fermi 6, Florence 50125, Italy*

(Received 18 September 2020; revised 29 January 2021; accepted 5 March 2021; published 1 April 2021)

Calibration of sensors is a fundamental step in validating their operation. This can be a demanding task, as it relies on acquiring detailed modeling of the device, which can be aggravated by its possible dependence upon multiple parameters. Machine learning provides a handy solution to this issue, operating a mapping between the parameters and the device response, without needing additional specific information on its functioning. Here, we demonstrate the application of a neural-network-based algorithm for the calibration of integrated photonic devices depending on two parameters. We show that a reliable characterization is achievable by carefully selecting an appropriate network training strategy. These results show the viability of this approach as an effective tool for the multiparameter calibration of sensors characterized by complex transduction functions. Furthermore, the approach is proven to be versatile and promising for mass production, as the same neural network is able to calibrate different devices that have the same structure.

DOI: 10.1103/PhysRevApplied.15.044003

I. INTRODUCTION

Quantum metrology has demonstrated significant advances in the past few years, keeping up with the perspective of a novel generation of quantum sensors with enhanced sensitivity [1–5]. In order to fully exploit these advantages, the device must be known and controlled, so that the measured parameters can be retrieved with good accuracy. This ability then relies on the availability of a trusted calibration of the sensor in hand [6]. Conventional methods for device characterization require, in general, a large set of calibration data and intensive postprocessing. Alternatively, one could rely on refined *a priori* physical models of the sensors, including noise effects, based on measured quantities. Such intensive approaches, however, become impractical for sensors of increasing complexity and unfeasible from the perspective of commercial devices.

Further, in realistic sensors, the user typically has access and control over a set of physical parameters, which in turn modify the internal characteristics of the device. Therefore, the target parameters are not directly evaluated but are inferred from measured quantities. Optical phases, determined by electrical signals via thermo-optic effects, are a case in point [7–10]: here, voltages are the parameters of interest, but the measured optical signal derives from variations of optical phases. Therefore, the sensor inherently works as a transductor, mapping the parameters to be estimated onto measured quantities through a suitable response function, which needs to be characterized as well. In this respect, spurious effects also affect the response function and must be taken into account. This poses major difficulties in the modeling and increases the complexity of an experimental characterization via conventional methods.

A practical approach thus requires a different methodology for sensor calibration. A viable direction is provided by machine-learning techniques in order to circumvent the need to model the sensor response. These are capable of handling large data sets and of solving tasks for which they have not been explicitly programmed; applications range from stock-price predictions [11,12] to the analysis of medical diseases [13]. In the past few years, several

[*] fabio.sciarrino@uniroma1.it
[†] These authors contributed equally to this work.

applications of machine-learning methods in the quantum domain have been reported [14–16], including state and unitary tomography [17–25], the design of quantum experiments [26–32], the validation of quantum technology [33–35], the identification of quantum features [36,37], and the adaptive control of quantum devices [38–54]. Also, photonic platforms can be exploited for the realization of machine-learning protocols [55,56]. Recently, an insight into the application of machine-learning methods for the calibration of a quantum sensor has been reported [57]. In detail, the characterization of an optical phase sensor has been carried out by means of artificial neural networks (NNs) [58]. This has demonstrated its advantages, in that it required no detailed model, it relied on the same states for the calibration as for the estimation, and it demonstrated robustness to finite-size data sets when compared to standard methods. When extending the use of NNs to multiple-parameter scenarios, these features can be preserved and help to solve a crucial issue. Variations of the parameters can affect the sensor with expected forms of behavior, as well as with undesirable crosstalk effects, with the latter generally being hard to model due to their spurious nature. The effective nonanalytical approach of NNs avoids these difficulties.

In this paper, we demonstrate the calibration of multiphase sensors implemented in a femtosecond-laser-written photonic platform. The devices are multiarm interferometers with multiple embedded phases, which are controlled externally by applying a voltage to resistors placed within the sensor. We report on how to build and train a NN to make it capable of working as a reliable and practical calibration tool. We illustrate how the optimization of the NN settings is operated and how ambiguities in the sensor response function are resolved. Our results yield evidence on the viability of a machine-learning approach for the calibration of complex quantum devices.

II. RESULTS

In this section, we detail our use of a feed-forward neural network [59] to approximate the multivariate function that links single-photon detection probabilities, relative to the outputs of an integrated three-mode device, given a certain input, to the voltage settings controlling the interferometer phases.

Regression is one of the most common problems faced by supervised learning algorithms. It consists in finding a map f that links an input vector of features $\vec{x}_i$ to the respective output vector $\vec{y}_i$ consisting of real numbers, for all the examples i in the data set. NNs are very effective at modeling complex nonlinear functions for very-large-dimension data sets described by many features and their performance is particularly good when large training data sets containing many different examples are available.

A. Experimental platform

The integrated device under study is a three-arm interferometer realized by the femtosecond-laser-writing technique [60,61] and able to perform multiphase estimation protocols [51,62]. The circuit is composed by a sequence of two three-arm beam splitters (tritters) realized through a two-dimensional geometry decomposition and interposed with three internal arms encoding two independent optical phase shifts $\boldsymbol{\Delta\phi} = (\Delta\phi_1, \Delta\phi_2)$ of two of the arms with respect to the third one (reference). These are thermo-optic phase shifts that can be tuned by means of Ohmic resistors. When a set of voltages (V_1, V_2) is applied to the resistors, a different global phase shift is generated along each optical path, thus changing the action of the device. A schematic of the device is shown in Fig. 1. The chip is studied in the single-photon regime. Pairs of photons with wavelength 785nm are generated by a spontaneous parametric down-conversion process in a beta barium borate (BBO) crystal. A single photon of the pair is coupled into a fiber array connected to one of the inputs of the circuit, while the other photon acts as a trigger. Single-photon detectors are placed at the output fibers of the device and coincidence events between the trigger and the photon injected inside the chip are recorded. In this way, the single-photon probabilities $P(i \rightarrow j)$ for each output j $(j = 1, 2, 3)$ are measured by changing the input arm i of the single-photon state and tuning the power dissipated on the internal resistors. In particular, we tune the voltages applied to each resistor independently, while keeping the others off. Thus, these measurements map the output distribution in the single-photon regime onto the control parameters. Notably, in our case, the parameters controlled by the user are the electric voltages and not the internal phases directly. If one is interested in the physical relative phase shifts $\Delta\phi_1$ and $\Delta\phi_2$ resulting from the application of the two voltages V_1 and V_2, a preliminary calibration step is required. The characterization of the voltage-to-phase response is unavoidable for any similar sensor and can result in additional calibration errors of the global response. A possible model [62] is provided by the following approximate response function containing linear and quadratic dependencies from the dissipated power:

$$\Delta\phi_i = \sum_{j=1}^{2} \left(\alpha_{ij} P_{R_j} + \alpha_{ij}^{\mathrm{NL}} P_{R_j}^2 \right), \tag{1}$$

where the dissipated power is $P_{R_j} = V_j^2/R_j$ using an Ohmic approximation for the two resistors, while α_{ij} and $\alpha_{ij}^{\mathrm{NL}}$ are, respectively, the linear and nonlinear response coefficients associated with the phase shift $\Delta\phi_i$ $(i = 1, 2)$ when dissipating power on resistor R_j. Therefore, in the standard characterization, a theoretical model of the circuit is necessary to recover the output probabilities through a fit of the measured probabilities. This, in turn, allows us

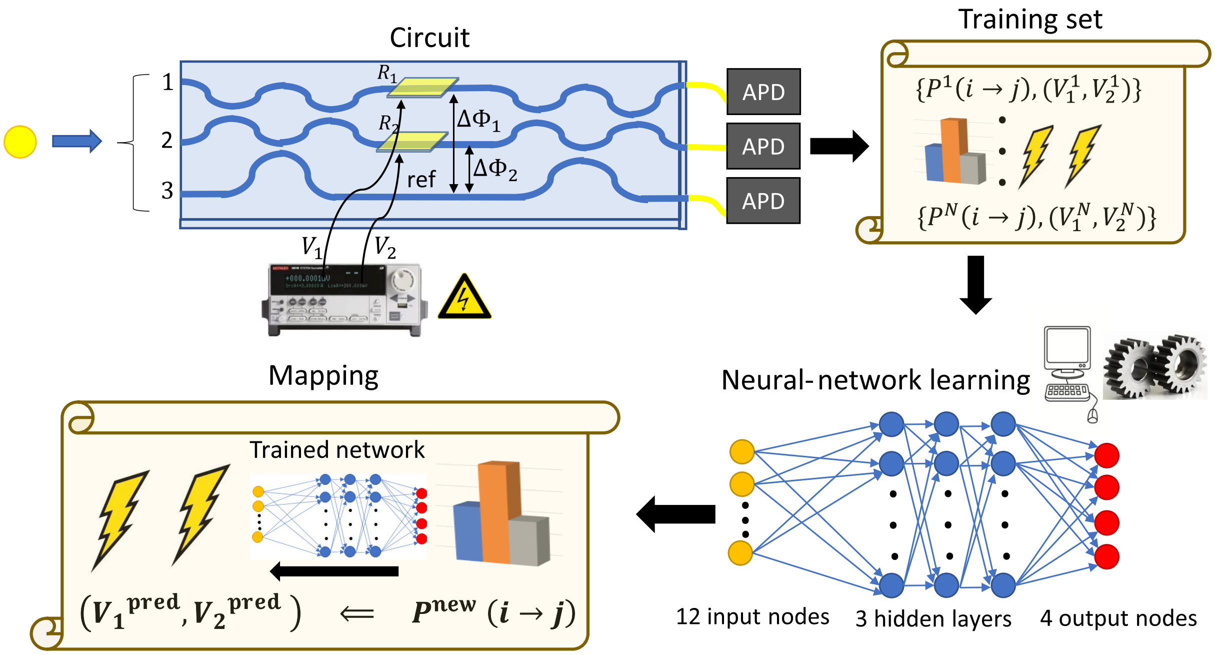

FIG. 1. The conceptual scheme, showing the calibration steps in a clockwise sense. Circuit: single photons are sent along one of the inputs of the three-arm interferometer and are revealed by avalanche photodiodes (APDs). Training set: the output probabilities $P(i \to j)$ (with $i,j = 1,2,3$) are measured as a function of the two applied voltages V_1 and V_2, collecting a total number of N training examples. As explained in the main text, the kick values $(\Delta V_1, \Delta V_2)$ and its probabilities $\tilde{P}(i \to j)$ (not shown in the figure) are also considered for removing ambiguous points. For this reason, the NN has four output nodes. NN learning: a small portion of the data set is used to train the neural network. Mapping: after training, the NN is able to map any new output probability $P^{\text{new}}(i \to j)$ to the corresponding pair of voltages, predicting the values $(V_1^{\text{pred}}, V_2^{\text{pred}})$ and the kick ones.

to extrapolate the dynamic and static parameters of the chip [51,62]. Conversely, the aim of this work is to avoid relying on knowledge about both the theoretical model of the circuit and the response function in Eq. (1). In fact, this would prove inefficient for the characterization of mass-produced devices. The goal, instead, is to generate a mapping between the voltages and the output probabilities using only a limited set of measurements. We exploit the NN approach exactly to realize such a goal (Fig. 1).

B. Neural-network architecture and performance

We start by using simulated data to inspect the algorithm requirements both in terms of the network architecture and the amount of training data needed to obtain good performance when evaluating new examples. In our case, the simulated data are obtained based on single-photon input-output probabilities $P(i \to j)$ estimated from experimental measurements through a fitting procedure. In particular, such a probability is obtained from the same reconstruction method employed in Ref. [62]: the entire circuit described in Sec. A is the result of the sequential 3×3 matrix operations $U^{\text{chip}} = U^B U(\boldsymbol{\Delta\phi}) U^A$, where U^A and U^B are the matrices describing the actual tritter transformations, while $U(\boldsymbol{\Delta\phi})$ introduces the phase shifts $\boldsymbol{\Delta\phi}$, tuned by voltages (V_1, V_2) according to Eq. (1). The single-photon output probabilities are obtained as $P(i \to j) = |U_{j,i}^{\text{chip}}|^2$. This preliminary step has the sole purpose of identifying the structure of the NN more efficiently for the calibration of a general three-arm interferometer: while it is instrumental for our verification of the NN method, it is not required for its actual use.

To train the NN, the data are divided into a vector of input features $\vec{x}$ corresponding to the nine input-output probabilities obtained when applying a given pair of controlled voltages, which constitute the elements of the output vector $\vec{y}$:

$$\{\vec{x}, \vec{y}\}_k = \{P(i \to j), (V_1, V_2)\}_k \text{ for } i,j = 1,2,3. \quad (2)$$

The index $k = 1, \ldots, N$ refers to each training example and N represents the total number of training data. The vector of probabilities $\vec{x}$ is constructed using counts extracted from a Poisson distribution the mean values of which correspond to those assessed from the fit. This procedure is needed for the NN to properly account for the presence of such source of uncertainty. A part of the data, viz., 15% of the whole set, is used as a validation set: it is not directly employed for the training but, rather, to obtain an independent estimate on the training error. This is necessary to avoid overfitting.

We test different NN architectures with a different number of hidden layers and neurons per layer, studying the network performance for different activation functions, initialization parameters, and optimization algorithms. Finally, we choose the one that has the smallest root-mean-squared error (RMSE) on the validation set. We note that the probabilities are not independent, due to the constraints imposed by unitarity of the transformation, and the full set of nine probabilities is redundant for the network training. Indeed, only four of them are required, plus their normalization. Given this redundancy, satisfactory results can be achieved training the network with the six probabilities obtained when injecting a photon into the first and second input of the device, respectively. The training consists in tuning the parameters of the model to minimize the RMSE associated with each example of the training set; for this purpose, the gradient of the loss function, corresponding to the summation of the RMSE over all the training examples, with respect to all the parameters of the network, is computed using the back-propagation method [63]. In the next step, the gradient is used to minimize the loss function using the ADAM optimization algorithm [64].

We train the network with the results obtained after the application of 53 different tension values to each of the two resistors in the device. This gives a tension grid with $53 \times 53 = 2809$ different tension pairs associated with the relative input-output probabilities available to train the network. Therefore, each example in the training set, consists of the tension pairs associated with six probability values. The trends of the values predicted by the trained NNs are reported in Fig. 2 as a function of the applied voltages on the two resistors. However, to obtain a good estimation over the full range of accessible tensions, it is necessary to incorporate into each training example the further set of probabilities $\tilde{P}(i \to j)$, which we refer to as kicks, as follows:

$$\left\{\vec{x},\vec{y}\right\}_k = \left\{[P(i \to j), \tilde{P}(i \to j)],\right. \\ \left.(V_1, V_2, V_1 + \Delta V_1, V_2 + \Delta V_2)\right\}_k, \qquad (3)$$

where the values $\tilde{P}(i \to j)$ are added by considering the probabilities obtained by changing V_1 (V_2) of a fixed value ΔV_1 (ΔV_2) and the length of the $\vec{x}$ and $\vec{y}$ vectors is K. This is necessary to remove ambiguities in the evaluation of the overall function, providing additional information to the network. More specifically, this requirement is due to the noninjectivity of the output probabilities, resulting in the presence of multiple parameter points that correspond to the same probability values. Thus, in the absence of additional data sets, it is not possible to distinguish between those points.

The kicks will be *de facto* treated as additional input probabilities by the NN and, as such, once chosen, to allow for the NN to correctly associate their values with the voltages, they need to be kept fixed between the training

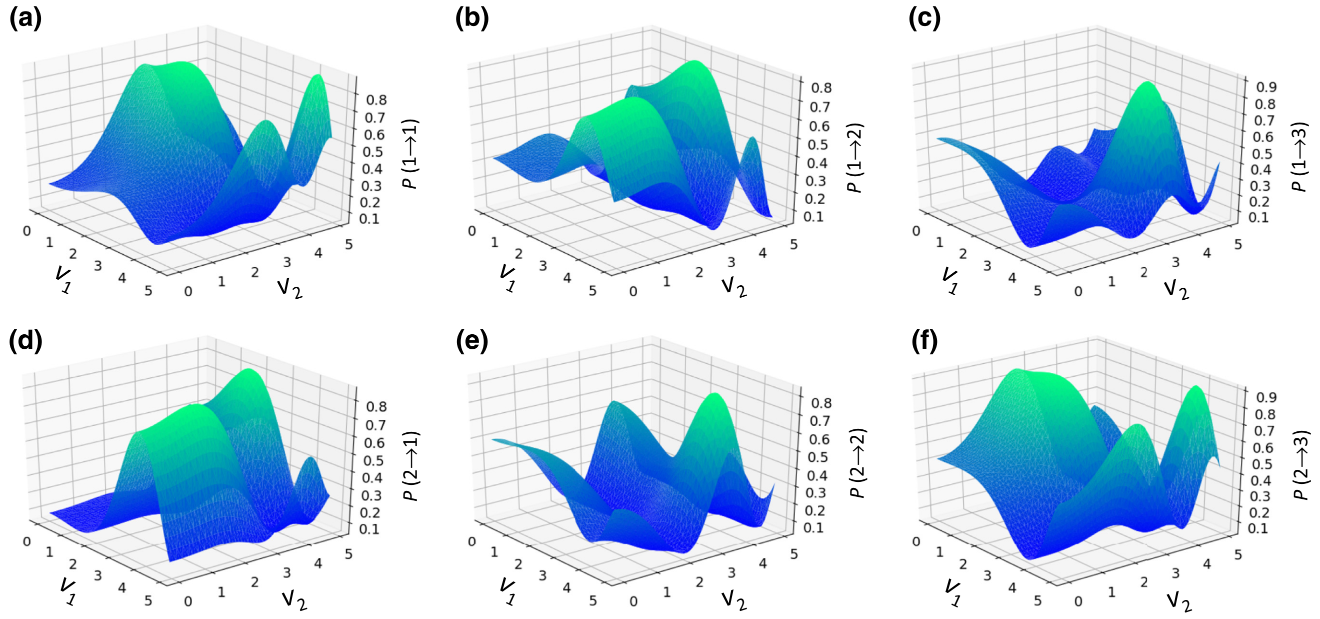

FIG. 2. Examples of input-output probabilities $P(i \to j)$ for each input i and output j $(i,j = 1,2,3)$ of the chip, resulting from predictions of a NN trained with simulated data, as a function of the applied voltages V_1,V_2 over two Ohmic resistors. (a)–(c) The output probabilities $P(1 \to j)$ obtained by injecting single photons along input 1. (d)–(f) The output probabilities $P(2 \to j)$ obtained by injecting single photons along input 2.

and the test. The optimal values for ΔV_i need to be tailored to the functions at hand, as they will depend on the steepness and periodicity of the probabilities. In our case, we implement kicks associated with $\Delta V_1 = 0.57\ V$ and $\Delta V_2 = 2.27\ V$.

If we compare the network results on the validation set, the RMSE is improved by 85% due to the additional information provided by the tension kick, when evaluated over the full range of tension values. The advantage obtained is independent of the specific value of ΔV_1 and ΔV_2, as long as they are large enough to yield information about different regions of the inspected functions. Enlargement of the data in each training example requires us to increase the number of nodes in the input and output layers of the network. The architecture that allows us to achieve the best performance, among the ones considered, is a network with 12 input nodes and four output nodes, separated by three hidden layers with 200 nodes each (Fig. 1). All the nodes, except for the output ones, which are activated by a linear function, are activated by a rectified linear unit (reLu) function initializing their weights, with random values extracted from a normal distribution centered on zero and with variance $\sigma^2 = 2/n$, where n is the number of neurons in the previous layer. In particular, as is often the case in regression problems, the linear activation function provides the best performance compared to those obtained with other activation functions, i.e., the reLu, the sigmoid, and the hyperbolic tangent.

The training can be stopped when the loss function on the validation set stops decreasing. In our case, this happens always before 250 epochs have been completed. During each epoch, all the training data are passed to the NN, which adjusts its internal weights. Moreover, to make the algorithm more efficient, the whole training set can be divided into small random batches, which are iteratively analyzed during each training epoch. To analyze the variability among different training runs, we study the results obtained, starting from the same data sets, after performing 50 independent training runs of the network. The mean value of the normalized root-mean-squared error (NRMSE) is given by

$$\varepsilon = \frac{1}{\sqrt{K}} \frac{||\vec{y} - \vec{\tilde{y}}||}{y_{\max} - y_{\min}}, \tag{4}$$

where $|| \cdot ||$ indicates the Euclidean norm. It is calculated on the validation set in such a configuration, yielding a value of $\varepsilon = 0.015 \pm 0.001$. After the network is trained, its performance is evaluated on an independent test set of 100 different examples selected randomly among the possible tension pairs of the 53 × 53 grid, adding Poissonian noise on the detection events that determine the probability values. Notably, since both the training and the test data include random Poissonian noise, the resulting NN is robust when evaluating new noisy examples. To quantify how close the network estimation of the tension values is to the true values, we evaluate the cosine similarity between the vector of network outputs $\vec{y}$, corresponding to the four tensions for all the test examples, and the expected results $\vec{\tilde{y}}$, as follows:

$$c = \frac{\vec{y} \cdot \vec{\tilde{y}}}{||\vec{y}|| \cdot ||\vec{\tilde{y}}||}, \tag{5}$$

which is equal to 1 when the prediction $\vec{\tilde{y}}$ is equal to the true value $\vec{y}$. Moreover, to estimate how much the cosine similarity depends on the random sample selected, we compute its value on 500 repetitions, each containing 100 examples extracted randomly among the available ones, obtaining a value of $c = 0.999 \pm 0.001$.

Once the network architecture is established, we investigate how the NRMSE on the validation set and the cosine similarity between the network estimation and the true tension values change, reducing the number of tension pairs used for the training. For all the different configurations, the size of the training set, and consequently of the validation set, changes depending on the dimension of the tension grid used. Conversely, the number of examples making up the test set remains fixed at 100. For all the new training configurations, the test data are still extracted randomly from the largest grid. This choice is made to assess the extent to which reducing the data for the training affects the final network estimation of new examples. Figure 3 reports the NRMSE on the validation set, obtained from multiple training runs of the network and the cosine similarity among the network estimation and the expected values in the test set. As expected, the NRMSE achieved by the network decreases as the number of training examples increases, allowing a better reconstruction of the function mapping the input vector onto the output one. A better reconstruction of this function grants higher network performance on the independent test set, as shown by the growth of the cosine similarity between the reconstructed tension vectors by the NN and the real one. The error in the cosine similarity values gives an indication of the variability linked to the analysis of different examples that randomly fall in different regions of the probability functions. In parallel, the error in the NRMSE values depends on the results of different training runs of the algorithm that, starting with random initial parameters, can end up in slightly different conditions.

C. Results on experimental data

Here, the network architecture resulting from the previous analysis is implemented and tested by training the NN directly on actual experimental data. This is possible due to the network efficiency in learning the function that maps the input-output probabilities to the tensions applied. In this way, a good estimate of the tension values

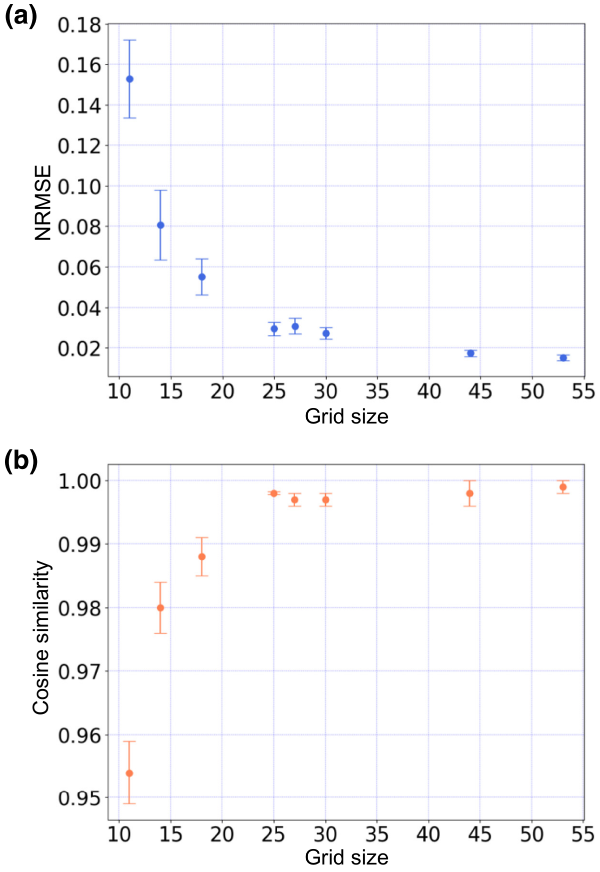

FIG. 3. NN training with different values of the data grid size. The training performance is shown in terms of (a) the NRMSE, computed over the validation set, and (b) the cosine similarity on the test set.

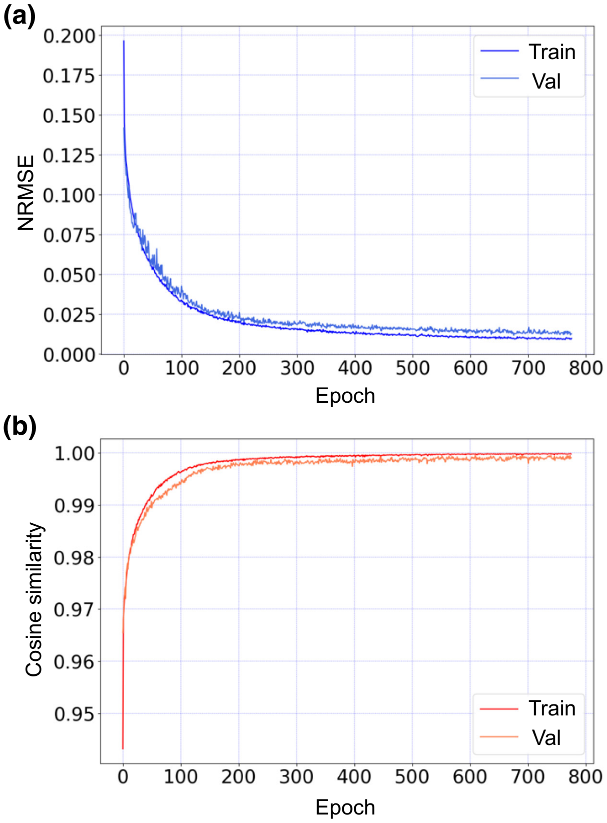

FIG. 4. A study of the NN performance by using the experimental data set for both the training and the validation stages: (a) the NRMSE and (b) the cosine similarity are shown as a function of the computed epochs.

is obtained. If the explicit model given in Eq. (1) is available, this same procedure is also effective to calibrate the response of the device explicitly in terms of the phases; an exact description of the further propagation and measurement steps is not needed. We want to test the robustness of the developed NN with respect to its hyperparameters, such as the number of nodes and layers. For this purpose, we calibrate another integrated circuit with different output probabilities but the same structure as the first one. More specifically, we use the data obtained from a 50 × 50 tension grid applied on a second integrated device with the same layout, to train the same network as the one described above. The training with such data takes longer to reach the region where the loss function on the validation set stops decreasing. This is shown in Fig. 4, where the NRMSE and the cosine similarity are reported as a function of the training epochs. The network tension estimations when 100 new data are acquired are reported in Fig. 6, showing the ability of the network to make accurate predictions of the applied voltages. To visualize clearly the difference among the predictions obtained by the NN and the expected tension values, we report in the inset of Fig. 5 the same points of Fig. 6 together with the true tension values. To study the variability among multiple test batches, we compute the standard deviation over the NRMSE obtained

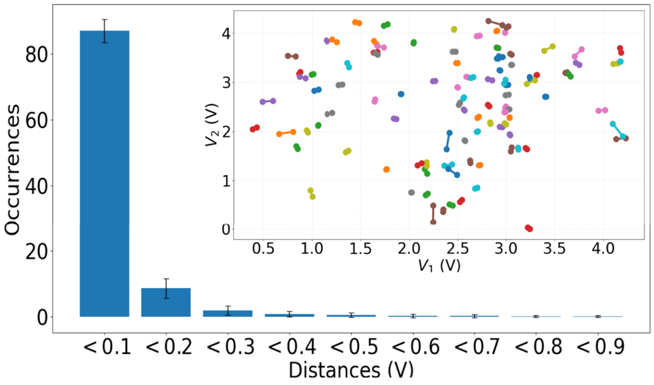

FIG. 5. The histogram of the two-point distances obtained over the test set of 100 points. The error bar represents the standard deviation computed over 500 different random batches of test sets. In the inset, we report one of the 500 random test sets consisting of 100 points represented in the same color and linked by a line to the 100 tension pairs predicted by the NN.

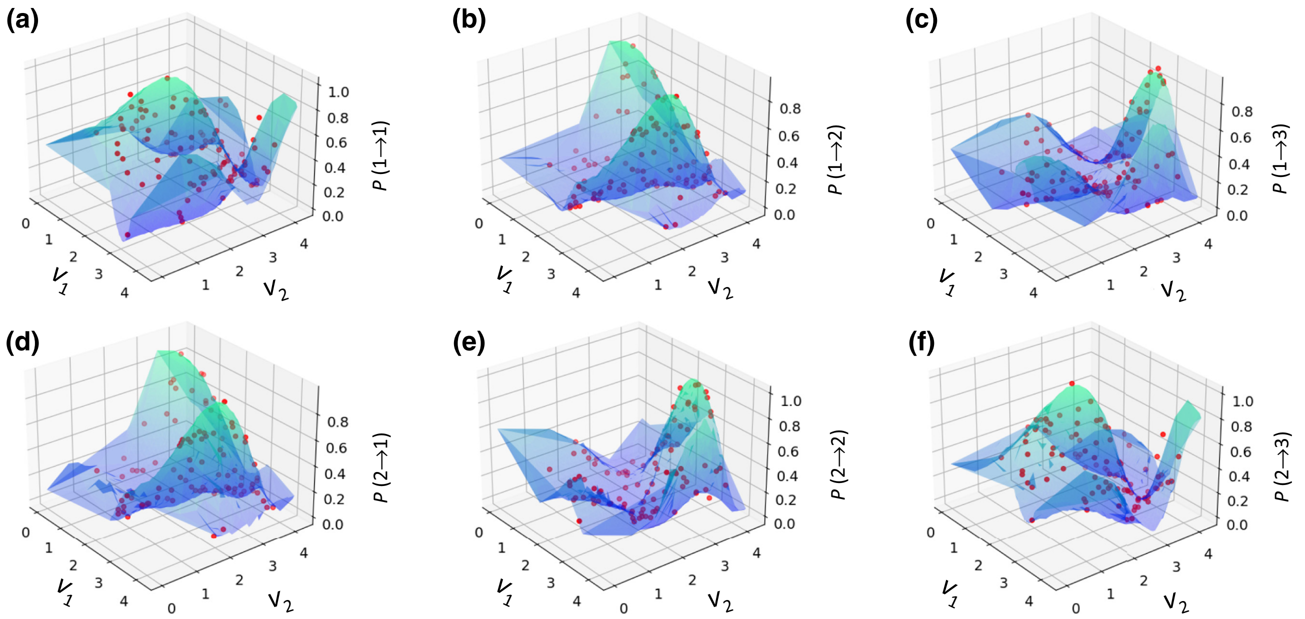

FIG. 6. A comparison between the voltages predicted from the NN (red dots) and the grid of experimental values (blue surface) for different single-photon input-output probabilities. The red predicted points are the same ones as reported in the inset of Fig. 5

on 500 different random test sets, obtaining a value of $\varepsilon = 0.022 \pm 0.009$ which, as expected, is compatible with the results obtained on the validation set and reported in Fig. 4(a). To quantify the difference among the predictions and the true values, we also compute the two-point distances of all the 100 points for all the 500 test repetitions. The histogram reported in Fig. 5 shows that more than 80% of the predicted points are at a distance from the true tension values of less than $0.1V$, demonstrating the accuracy of the NN predictions. In this way, we show that the parameters chosen for the NN can also be applied for calibrating different circuits that share the same structure. This feature is key from the perspective of mass-produced devices with the same layout.

This black-box approach is by no means limited to calibrations for metrology and can find more extensive applications. For instance, the same NN could be used to extract the voltage settings necessary to obtain target splitting ratios among the three output probabilities. Also in this case, in which our device is operated as an intensity partitioner, the NN could provide a practical solution to assess its actual capabilities.

III. CONCLUSIONS

We report on the application of a NN-based algorithm to perform the calibration of integrated devices depending on two parameters. In this investigation, we rely on knowledge of a model to identify the most appropriate regime for collecting the training set. However, this is by no means a necessary step in that the use of the NN itself incorporates the same information that would be present in the model. Remarkably, the NN is able to account for spurious effects such as, in our case, crosstalk between thermal actuators, which are otherwise intricate to describe.

Concerning the scalability of the NN approach, two aspects need to be considered. First, this method is not expected to mitigate the growth of the required number of collected data points for an increasing system size. The gradients of the response function also significantly affect the collection size. Second, the scaling of the network itself will depend on the complexity of the studied system. However, these aspects are well balanced by the fact that NNs are intrinsically well suited to handling large data sets. Overall, this approach can still be beneficial in tackling more complex systems, depending on a large number of parameters, that are otherwise intractable through standard fitting procedures.

Moreover, even if the NN approach does not require a full description of the device, it can be foreseen that some basic modeling of it could nevertheless be beneficial. The successful characterization of two devices based on a single approximate model shows that the NN performance does not depend heavily on the level of detail of the model. In the same vein, some anticipation of the device output might reveal whether ambiguities may be present in the chosen range of parameters. We show that this is easily accounted for by introducing additional data as input to the NN.

This study brings forward machine-learning applications in two respects: it goes beyond optimization of the employed resources when these are severely constrained and it shows that this characterization method extends beyond the single-parameter regime. The obtained results yield evidence that the NN can provide an effective, robust, and reliable tool for the calibration of complex sensors that depend on multiple parameters, with the advantage of requiring no detailed model of their internal operation.

ACKNOWLEDGMENTS

This work is supported by the European Research Council (ERC) Advanced Grant CAPABLE ("Composite integrated photonic platform by femtosecond laser micromachining," Grant Agreement No. 742745), by the Amaldi Research Center, funded by the Ministero dell'Istruzione dell'Università e della Ricerca (Ministry of Education, University and Research) program "Dipartimento di Eccellenza" (CUP:B81I18001170001), and by the European Union's Horizon 2020 research and innovation program under the PHOQUSING ("PHOtonic Quantum SamplING machine") project Grant Agreement No. 899544 and the STORMYTUNE ("Spectral-Temporal Metrology with Tailored Quantum Measurements") project Grant Agreement No. 899587. I.G. is supported by the Ministero dell'Istruzione, dell'Università e della Ricerca Grant of Excellence Departments (ARTICOLO 1, COMMI 314-337 LEGGE 232/2016). N.S. acknowledges funding from Sapienza Università via Bando Ricerca 2018: Progetti di Ricerca Piccoli, project "Multiphase estimation in multiarm interferometers."

[1] M. G. Paris, Quantum estimation for quantum technology, Int. J. Quantum Inf. **7**, 125 (2009).

[2] V. Giovannetti, S. Lloyd, and L. Maccone, Advances in quantum metrology, Nat. Photonics **5**, 222 (2011).

[3] L. Pezzé and A. Smerzi, in *Atom Interferometry, Proceedings of the International School of Physics Enrico Fermi*, edited by G. M. Tino and M. A. Kasevich (IOS Press, Amsterdam, 2014), p. 691.

[4] S. Pirandola, B. R. Bardhan, T. Gehring, C. Weedbrook, and S. Lloyd, Advances in photonic quantum sensing, Nat. Photonics **12**, 724 (2018).

[5] E. Polino, M. Valeri, N. Spagnolo, and F. Sciarrino, Photonic quantum metrology, AVS Quantum Sci. **2**, 024703 (2020).

[6] I. Gianani, M. G. Genoni, and M. Barbieri, Assessing data postprocessing for quantum estimation, IEEE J. Sel. Top. Quantum Electron. **5**, 6500207 (2020).

[7] F. Flamini, L. Magrini, A. S. Rab, N. Spagnolo, V. D'Ambrosio, P. Mataloni, F. Sciarrino, T. Zandrini, A. Crespi, R. Ramponi, and R. Osellame, Thermally reconfigurable quantum photonic circuits at telecom wavelength by femtosecond laser micromachining, Light: Sci. Appl. **4**, e354 (2015).

[8] J. Carolan, C. Harrold, C. Sparrow, E. Martin-Lopez, 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 (2015).

[9] 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 (2017).

[10] C. Taballione, T. A. W. Wolterink, J. Lugani, A. Eckstein, B. A. Bell, R. Grootjans, I. Visscher, D. Geskus, C. G. H. Roeloffzen, J. J. Renema, I. A. Walmsley, P. W. H. Pinkse, and K.-J. Boller, 8 × 8 reconfigurable quantum photonic processor based on silicon nitride waveguides, Opt. Express **27**, 26842 (2019).

[11] J. L. Ticknor, A Bayesian regularized artificial neural network for stock market forecasting, Expert Syst. Appl. **40**, 5501 (2013).

[12] D. Enke, M. Grauer, and N. Mehdiyev, Stock market prediction with multiple regression, fuzzy type-2 clustering and neural networks, Procedia Comput. Sci. **6**, 201 (2011), complex adaptive sysytems.

[13] N. Ganesan, K. Venkatesh, M. Rama, and A. M. Palani, Application of neural networks in diagnosing cancer disease using demographic data, Int. J. Comput. Appl. **1**, 76 (2010).

[14] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).

[15] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Phys. Rep. **810**, 1 (2019).

[16] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[17] N. Spagnolo, E. Maiorino, C. Vitelli, M. Bentivegna, A. Crespi, R. Ramponi, P. Mataloni, R. Osellame, and F. Sciarrino, Learning an unknown transformation via a genetic approach, Sci. Rep. **7**, 14316 (2017).

[18] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Reconstructing quantum states with generative models, Nat. Machine Intelligence **1**, 155 (2019).

[19] A. M. Palmieri, E. Kovlakov, F. Bianchi, D. Yudin, S. Straupe, J. D. Biamonte, and S. Kulik, Experimental neural network enhanced quantum tomography, Npj Quantum Inf. **6**, 20 (2020).

[20] A. Rocchetto, S. Aaronson, S. Severini, G. Carvacho, D. Poderini, I. Agresti, M. Bentivegna, and F. Sciarrino, Experimental learning of quantum states, Sci. Adv. **5**, eaau1946 (2019).

[21] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, Machine learning method for state preparation and gate synthesis on photonic quantum computers, Quantum Sci. Technol. **4**, 024004 (2019).

[22] T. Giordani, A. Suprano, E. Polino, F. Acanfora, L. Innocenti, A. Ferraro, M. Paternostro, N. Spagnolo, and F. Sciarrino, Machine Learning-Based Classification of Vector Vortex Beams, Phys. Rev. Lett. **124**, 160401 (2020).

[23] M. Neugebauer, L. Fischer, A. Jäger, S. Czischek, S. Jochim, M. Weidemüller, and M. Gärttner, Neural-network

quantum state tomography in a two-qubit experiment, Phys. Rev. A **102**, 042604 (2020).

[24] G. Torlai, B. Timar, E. P. L. van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin, R. G. Melko, and M. Endres, Integrating Neural Networks with a Quantum Simulator for State Reconstruction, Phys. Rev. Lett. **123**, 230504 (2019).

[25] E. S. Tiunov, V. V. Tiunova (Vyborova), A. E. Ulanov, A. I. Lvovsky, and A. K. Fedorov, Experimental quantum homodyne tomography via machine learning, Optica **7**, 448 (2020).

[26] R. Nichols, L. Mineh, J. Rubio, J. C. Matthews, and P. A. Knott, Designing quantum experiments with a genetic algorithm, Quantum Sci. Technol. **4**, 045012 (2019).

[27] A. A. Melnikov, H. P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Active learning machine learns to create new quantum experiments, Proc. National Acad. Sci. **115**, 1221 (2018).

[28] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, Automated Search for new Quantum Experiments, Phys. Rev. Lett. **116**, 090405 (2016).

[29] L. O'Driscoll, R. Nichols, and P. Knott, A hybrid machine learning algorithm for designing quantum experiments, Quantum Machine Intelligence **1**, 5 (2019).

[30] K. K. Sabapathy, H. Qi, J. Izaac, and C. Weedbrook, Production of photonic universal quantum gates enhanced by machine learning, Phys. Rev. A **100**, 012326 (2019).

[31] M. Krenn, M. Erhard, and A. Zeilinger, Computer-inspired quantum experiments, Nat. Rev. Phys. **2**, 649 (2020).

[32] X. Gao, M. Erhard, A. Zeilinger, and M. Krenn, Computer-Inspired Concept for High-Dimensional Multipartite Quantum Gates, Phys. Rev. Lett. **125**, 050501 (2020).

[33] I. Agresti, N. Viggianiello, F. Flamini, N. Spagnolo, A. Crespi, R. Osellame, N. Wiebe, and F. Sciarrino, Pattern Recognition Techniques for Boson Sampling Validation, Phys. Rev. X **9**, 011013 (2019).

[34] F. Flamini, N. Spagnolo, and F. Sciarrino, Visual assessment of multi-photon interference, Quantum Sci. Technol. **4**, 024008 (2019).

[35] P. A. Knott, A search algorithm for quantum state engineering and metrology, New J. Phys. **18**, 073033 (2016).

[36] V. Cimini, M. Barbieri, N. Treps, M. Walschaers, and V. Parigi, Neural Networks for Detecting Multimode Wigner Negativity, Phys. Rev. Lett. **125**, 160504 (2020).

[37] V. Gebhart and M. Bohmann, Neural-network approach for identifying nonclassicality from click-counting data, Phys. Rev. Res. **2**, 023150 (2020).

[38] A. Hentschel and B. C. Sanders, Machine Learning for Precise Quantum Measurement, Phys. Rev. Lett. **104**, 063603 (2010).

[39] A. Hentschel and B. C. Sanders, Efficient Algorithm for Optimizing Adaptive Quantum Metrology Processes, Phys. Rev. Lett. **107**, 233601 (2011).

[40] N. B. Lovett, C. Crosnier, M. Perarnau-Llobet, and B. C. Sanders, Differential Evolution for Many-Particle Adaptive Quantum Metrology, Phys. Rev. Lett. **110**, 220501 (2013).

[41] C. Bonato, M. S. Blok, H. T. Dinani, D. W. Berry, M. L. Markham, D. J. Twitchen, and R. Hanson, Optimized quantum sensing with a single electron spin using real-time adaptive measurements, Nat. Nanotechnol. **11**, 247 (2016).

[42] P. Palittapongarnpim, P. Wittek, E. Zahedinejad, S. Vedaie, and B. C. Sanders, Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics, Neurocomputing **268**, 116 (2017).

[43] J. Liu and H. Yuan, Control-enhanced multiparameter quantum estimation, Phys. Rev. A **96**, 042114 (2017).

[44] S. Paesani, A. A. Gentile, R. Santagati, J. Wang, N. Wiebe, D. P. Tew, J. L. O'Brien, and M. G. Thompson, Experimental Bayesian Quantum Phase Estimation on a Silicon Photonic Chip, Phys. Rev. Lett. **118**, 100503 (2017).

[45] A. Lumino, E. Polino, A. S. Rab, G. Milani, N. Spagnolo, N. Wiebe, and F. Sciarrino, Experimental Phase Estimation Enhanced by Machine Learning, Phys. Rev. Appl. **10**, 044033 (2018).

[46] P. Palittapongarnpim and B. C. Sanders, Robustness of quantum-enhanced adaptive phase estimation, Phys. Rev. A **100**, 012106 (2019).

[47] H. T. Dinani, D. W. Berry, R. Gonzalez, J. R. Maze, and C. Bonato, Bayesian estimation for quantum sensing in the absence of single-shot detection, Phys. Rev. B **99**, 125413 (2019).

[48] G. Liu, M. Chen, Y.-X. Liu, D. Layden, and P. Cappellaro, Repetitive readout enhanced by machine learning, Machine Learning: Sci. Technol. **1**, 015003 (2020).

[49] Y. Peng and H. Fan, Feedback ansatz for adaptive-feedback quantum metrology training with machine learning, Phys. Rev. A **101**, 022107 (2020).

[50] K. Rambhatla, S. E. D'Aurelio, M. Valeri, E. Polino, N. Spagnolo, and F. Sciarrino, Adaptive phase estimation through a genetic algorithm, Phys. Rev. Res. **2**, 033078 (2020).

[51] M. Valeri, E. Polino, D. Poderini, I. Gianani, G. Corrielli, A. Crespi, R. Osellame, N. Spagnolo, and F. Sciarrino, Experimental adaptive Bayesian estimation of multiple phases with limited data, Npj Quantum Inf. **6**, 92 (2020).

[52] K. Craigie, E. Gauger, Y. Altmann, and C. Bonato, Resource-efficient adaptive Bayesian tracking of magnetic fields with a quantum sensor, arXiv preprint arXiv:2008.08891 (2020).

[53] S. P. Nolan, A. Smerzi, and L. Pezzè, A machine learning approach to Bayesian parameter estimation, arXiv preprint arXiv:2006.02369 (2020).

[54] L. J. Fiderer, J. Schuff, and D. Braun, Neural-network heuristics for adaptive Bayesian quantum estimation, arXiv preprint arXiv:2003.02183 (2020).

[55] L. Bernstein, A. Sludds, R. Hamerly, V. Sze, J. Emer, and D. Englund, Freely scalable and reconfigurable optical hardware for deep learning, arXiv preprint arXiv:2006.13926 (2020).

[56] F. Flamini, A. Hamann, S. Jerbi, L. M. Trenkwalder, H. P. Nautrup, and H. J. Briegel, Photonic architecture for reinforcement learning, New J. Phys. **22**, 045002 (2020).

[57] V. Cimini, I. Gianani, N. Spagnolo, F. Leccese, F. Sciarrino, and M. Barbieri, Calibration of Quantum Sensors by Neural Networks, Phys. Rev. Lett. **123**, 230502 (2019).

[58] S. Haykin, *Neural Networks and Learning Machines* (Pearson Prentice Hall, Upper Saddle River, NJ, 2009).

[59] C. C. Aggarwal, *Neural Networks and Deep Learning* (Springer, Berlin, 2018), p. 497.

[60] R. R. Gattass and E. Mazur, Femtosecond laser micromachining in transparent materials, Nat. Photonics **2**, 219 (2008).

[61] G. Della Valle R. Osellame, and P. Laporta, Micromachining of photonic devices by femtosecond laser pulses, J. Opt. A: Pure Appl. Opt. **11**, 013001 (2008).

[62] E. Polino, M. Riva, M. Valeri, R. Silvestri, G. Corrielli, A. Crespi, N. Spagnolo, R. Osellame, and F. Sciarrino, Experimental multiphase estimation on a chip, Optica **6**, 288 (2019).

[63] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (The MIT Press, Cambridge, Massachusetts, 2016).

[64] D. P. Kingma and J. Ba, ADAM: A method for stochastic optimization, arXiv:1412.6980 (2014).