Research Article | Vol. 28, No. 9 / 27 April 2020 / Optics Express 14018
Optics EXPRESS

# Programming multi-level quantum gates in disordered computing reservoirs via machine learning

GIULIA MARCUCCI,[1,2] DAVIDE PIERANGELI,[1,2] PEPIJN W. H. PINKSE,[3] MEHUL MALIK,[4] AND CLAUDIO CONTI[1,2,*]

[1]*Department of Physics, University Sapienza, Piazzale Aldo Moro 2, 00185 Rome, Italy*
[2]*Institute for Complex Systems, National Research Council (ISC-CNR), Via dei Taurini 19, 00185 Rome, Italy*
[3]*Complex Photonic Systems (COPS), MESA+ Institute for Nanotechnology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*
[4]*Institute of Photonics and Quantum Sciences (IPAQS), Heriot-Watt University, Edinburgh, EH144AS, United Kingdom*
[*]*claudio.conti@uniroma1.it*

**Abstract:** Novel machine learning computational tools open new perspectives for quantum information systems. Here we adopt the open-source programming library TensorFlow to design multi-level quantum gates, including a computing reservoir represented by a random unitary matrix. In optics, the reservoir is a disordered medium or a multi-modal fiber. We show that trainable operators at the input and the readout enable one to realize multi-level gates. We study various qudit gates, including the scaling properties of the algorithms with the size of the reservoir. Despite an initial low slop learning stage, TensorFlow turns out to be an extremely versatile resource for designing gates with complex media, including different models that use spatial light modulators with quantized modulation levels.



## 1. Introduction

The development of multi-level quantum information processing systems has steadily grown over the past few years, with experimental realizations of multi-level, or qudit logic gates for several widely used photonic degrees of freedom, such as orbital-angular-momentum and path encoding [1–4]. However, efforts are still needed for increasing the complexity of such systems while still being practical, with the ultimate goal of realizing complex large-scale computing devices that operate in a technologically efficient manner.

A key challenge is the development of design techniques that are scalable and versatile. Recent work outlined the relevance of a large class of devices, commonly denoted as "complex" or "multi-mode" [5,6]. In these systems, many modes or channels are mixed and controlled at input and readout to realize a target input-output operation. This follows the first experimental demonstrations of assisted light transmission through random media [7–10], which demonstrated many applications, including arbitrary linear gates [5], mode conversion, and sorting [11,12].

The use of complex mode-mixing devices is surprisingly connected to leading paradigms in modern machine learning (ML), as the "reservoir computing" (RC) [13], and the "extreme learning machine" (ELM) [13,14]. In standard ML, one trains the parameters (*weights*) of an artificial neural network (ANN) to fit a given function, which links input and output. In RC, due to the increasing computational effort to train a large number of weights, one internal part of the network is left untrained ("the reservoir") and the weights are optimized only at input and readout.

#389432 https://doi.org/10.1364/OE.389432
Journal © 2020 Received 18 Feb 2020; revised 8 Apr 2020; accepted 9 Apr 2020; published 24 Apr 2020

Optics EXPRESS

ML concepts, such as photonic neuromorphic and reservoir computing [15,16], are finding many applications in telecommunications [17,18], multiple scattering [19], image classification [20], metasurfaces [21,22], biophotonics [10], Ising machines [23], integrated and fiber optics [24,25], and topological photonics [26]. Various authors have reported the use of ML for augmenting and assisting quantum experiments [27–31]. The field of machine learning is in turn influenced by quantum physics, for example in the orthogonal units [32,33].

Here we adopt RC-ML to design complex multi-level gates [2,3,34,35], which form a building block for high-dimensional quantum information processing systems. While low-dimensional examples of such gates have been implemented using bulk and integrated optics, efficiently scaling them up to high dimensions remains a challenge.

In quantum key distribution (QKD), one uses at least two orthogonal bases to encode information. High-dimensional QKD offers an increased information capacity as well as an increased robustness to noise over qubit-based protocols [36,37]. Such protocols may be realized by using the photonic spatial degrees of freedom as the encoding (computational) basis, and suitable unitary operators to switch between bases mutually unbiased with respect to the computational basis. However, the security of the QKD protocol may be compromised by the fidelity of such basis transformations, leading to errors in the key rate. An additional consideration is the experimental complexity of such transformations, which can scale rather poorly using established techniques based on bulk optical systems. By using a random medium and I/O readout operators, one can realize such high-dimensional operations in a controllable and scalable manner, relying only on the existing complexity of the disordered medium and a control operation at the input. Here, we explore methodologies to train a disordered medium to function as a multi-level logic gate by using different implementations of ML concepts.

Figure 1 shows the schematic of a device including the complex medium, represented by the unitary operator $\hat{U}$, and two trainable input $\hat{S}^{\text{in}}$ and readout $\hat{S}^{\text{out}}$ operators. $|h^{(1,2)}\rangle$ are hidden states. The use of an optical gate in this manner is related to the use of a disordered medium as a physically unclonable function (PUF) [38–41].

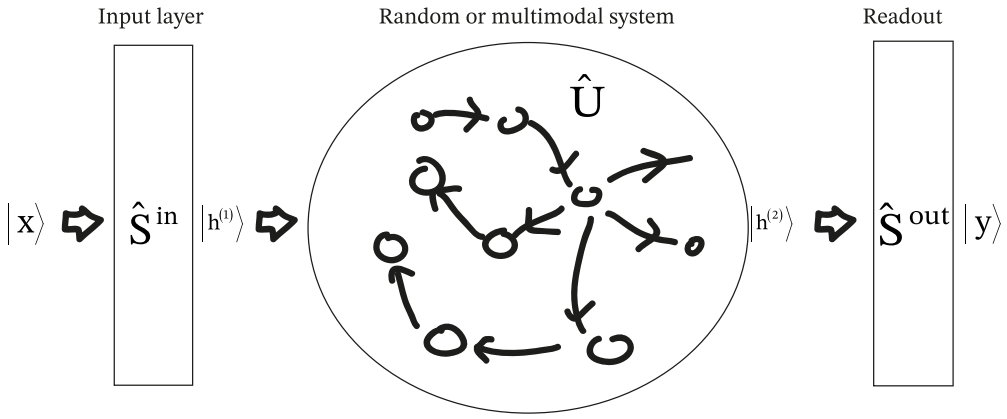


**Fig. 1.** A general optical gate based on a complex random medium; the input state **x** is processed to the input layer with operator $\hat{S}^{\text{in}}$, the system is modeled by the unitary operator $\hat{U}$, and the output is further elaborated by $\hat{S}^{\text{out}}$. By proper design of $\hat{S}^{\text{in,out}}$ the overall transmission realizes a target gate.

In our general framework, we have a random system modeled by a unitary random matrix. We want to use the random medium to perform a computation in a Hilbert space containing many qudits. The random medium is not necessarily a disordered system (for example, a dielectric assembly of scattering particles), but may also be a multimode fiber, or an array of waveguides.

Optics EXPRESS

The input/output relation is represented by a linear unitary matrix operator $U_M$ and only forward modes are considered. The $U_M$ matrix has dimensions $M \times M$, with $M$ the dimension of the *embedding space*.

The "reduced" state vector at input has dimensions $N \times 1$, with $N \leq M$. This models the case in which we use a subset of all the available modes. The input to the reservoir is a "rigged" state vector $\mathbf{x}$ with dimension $M$, where the missing complementing $C$ components are replaced by $C = M - N$ *ancillas*. Our goal is to use the random medium to perform a given operation denoted by a gate unitary matrix

$$T_M = S_M^{\text{out}} \cdot U_M \cdot S_M^{\text{in}}. \tag{1}$$

$S_M^{\text{in}}$ and $S_M^{\text{out}}$ are two "training" operators that are applied at input and output (see Fig. 1) and whose elements can be adjusted. We first consider the presence of the input operator $S_M^{\text{in}} = S_M$, and $S_M^{\text{out}} = \mathbf{1}_M$, which can be implemented by spatial-light modulators (we denote as $\mathbf{1}_M$ the identity matrix with dimension $M$).

We identify two cases: either (i) we know the matrix $U_M$, or (ii) we have to infer $U_M$ from the input/output measurements. We show in the following the way these two problems can be solved by ANNs, where we denote the two families as *non-inferencing* and *inferencing* gates.

## 2. Non-inferencing gates

We consider a target gate with complex-valued input state with dimension $N$, and components $x_1, x_2, \ldots, x_N$. We embed the input vector in a rigged Hilbert space with dimension $M \geq N$, so that the input vector is $\mathbf{x} = \{x_1, x_2, \ldots, x_N, x_{N+1}, \ldots, x_M\}$. We have a linear propagation through a medium with unitary complex transfer matrix $U_M$. The transmission matrix is $T_M = U_M \cdot S_M$, such that the output vector is $\mathbf{y} = T_M \cdot \mathbf{x} = U_M \cdot S_M \cdot \mathbf{x}$. The observed output vector is written as $P \cdot \mathbf{y}$, where $P$ is a $N-$projector operator with dimensions $N \times M$ such that $P = [\mathbf{1}_N|\mathbf{0}]$, with $\mathbf{1}_N$ the identity matrix with size $N \times N$, and $\mathbf{0}$ a null matrix with dimension $N \times C$. The goal is finding the matrix $S_M$ such that

$$P \cdot U_M \cdot S_M = [X_N \,|\, \mathbf{0}] \tag{2}$$

where $X_N$ is the $N \times N$ target gate and $\mathbf{0}$ is the null complement $N \times C$ at dimension $M$. Equation (2) is a matrix equation, which guarantees that the system behaves as a $X_N$ gate on the reduced input.

Solving the matrix Eq. (2) may be demanding and nontrivial when the number of dimensions grows. In the following, we discuss the use of ML techniques.

The transmission matrix $T_M$ in the rigged space from $\mathbf{x}$ to $\mathbf{y}$ can be written as blocks

$$T_M = \left[\begin{array}{c|c} X_N & \mathbf{0} \\ \hline \mathbf{0} & O_C \end{array}\right] \tag{3}$$

where $O_C$ is a unitary matrix with dimensions $C \times C$ to be determined. If $U_M$ and $S_M$ are unitary, the resulting transmission matrix $T_M$ is also unitary. However, if one uses Eq. (2), the problem may also have a non unitary solution as some channels are dropped at the output. In other words, solving Eq. (3) is not equivalent to solving Eq. (2), and we adopt two different methodologies: one can look for unitary or non unitary solutions by ANN.

By following previous work developed for real-valued matrices [42], we map the complex-valued matrix Eq. (2) into a recurrent neural network (RNN). In the "non-inferencing" case, the matrix $U_M$ is known, and the solution is found by the RNN in Fig. 2. The RNN solves an unconstrained optimization problem, by finding the minimum of the sum of the elements $e_{ij}>0$ of an error matrix $E$. The error depends on a "state matrix" $W_M$, and one trains the elements $w_{ij}$

Optics EXPRESS

of $W_M$ to find the minimum

$$\min_{W_M} E[G(W_M)] = \min_{W_M} \sum_{i,j} e_{ij}[G(W_M)]. \tag{4}$$

In the adopted approach, the sum of the elements $e_{ij}$ is minimal when the *hidden layer* elements $g_{ij}$ of the matrix $G(W)$ are zero. $E$ and $G$ have to be suitably chosen to solve the considered problem. We found two possible $G$ matrices: (i) the "projected"

$$G_P = P \cdot U_M \cdot W_M - X_{N0}, \tag{5}$$

with $X_{N0} = [X_N \,|\, \mathbf{0}]$ as in Eq. (2) and, (ii) the "unitary" [see Eq. (3)]

$$G_U = U_M \cdot W_M - T_M. \tag{6}$$

These two cases are discussed below.

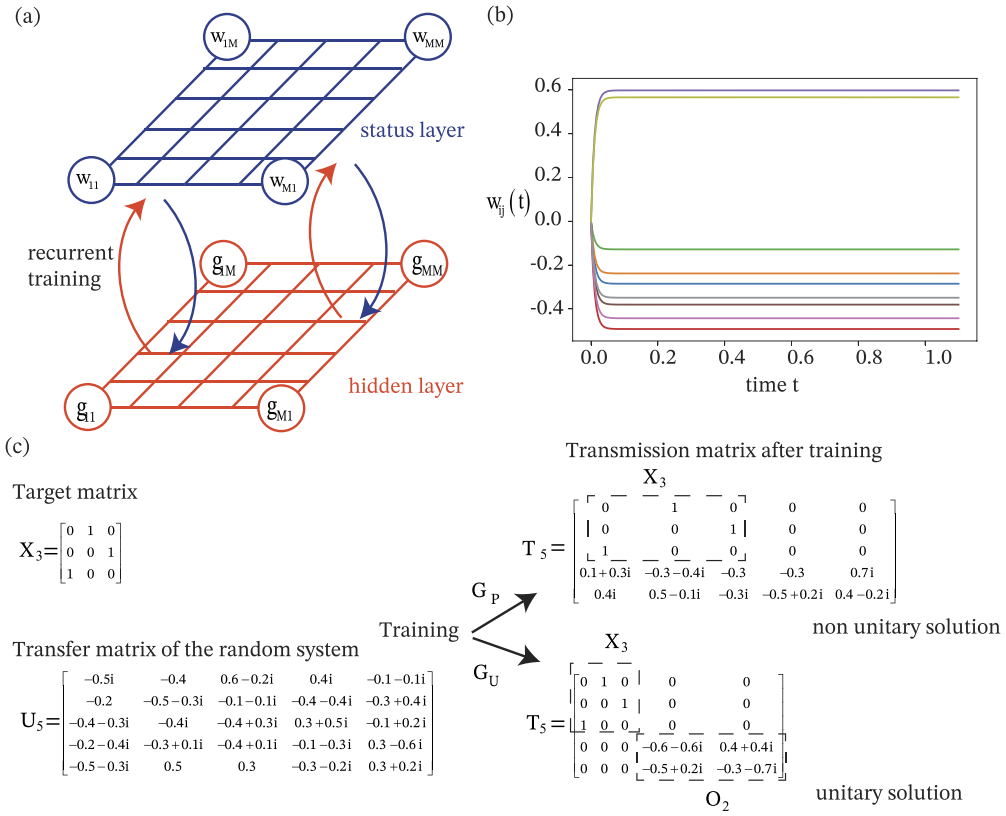


**Fig. 2.** (a) Recurrent neural network for the matrix Eq. (7). The status nodes are denoted by the elements of the matrix $W$, and the hidden state of the system is in the nodes of the matrix $F$; (b) training dynamics for the case $N = M = 3$ with $X_T$ corresponding to a single-qutrit X-gate ($\mu = 100$); the real part of $w_{ij}$ is reported; (c) resulting transfer function for the case $N = 3$ and $M = 5$ in the unitary and non-unitary case. In the latter case, the excess channels are ignored during the training. The resulting transmission channels $T_M$ are displayed, $O_2$ is the unitary complements for $C = M - N = 2$ in the unitary case.

To find the unknown training matrix $S_M$, one starts from an initial guess matrix $W_M(0)$. The guess is then recurrently updated, as in Fig. 2, until a stationary state $W_M(\infty)$ is reached. Once

Optics EXPRESS

this optimization converges, the solution is given by $S_M = W_M(\infty)$. The update equation is determined by a proper choice of the error matrix $E$ as follows.

As the matrices are complex valued, $e_{ij}$ is a function of $g_{ij}$ and $g_{ij}^*$. We set $e_{ij} = e_{ij}(|g_{ij}|^2)$. The corresponding dynamic RNN equation, which for large time gives the solution to the optimization problem, is

$$\frac{dW_M}{dt} = -\mu U_M^\dagger \cdot F[G(W_M)] \tag{7}$$

where $\mu$ is the "learning rate", an hyperparameter, which is set to speed-up the convergence. The elements $f_{ij}$ of the matrix $F$ are $f_{ij} = \frac{de_{ij}}{dg_{ij}^*}$. Letting $e_{ij} = |g_{ij}|^2$, one has $f_{ij} = g_{ij}$.

Equation (7) implies that the RNN is composed of two bidirectionally connected layers of neurons, the output layer with state matrix $W$, and the hidden layer with state matrix $G$. The training corresponds to sequential updates of $F$ and $W$ when solving the ordinary Eq. (7). As shown in [42], this RNN is asymptotically stable and its steady state matrix represents the solution (an example of training dynamics is in Fig. 2(b)).

We code the RNN by TensorFlow and use the ordinary differential equations (ODEs) integrator *odeint*. By construction 7 is a convex optimization problen, any initial condition will converge to the solution, as it happens in our simulations where we use as initial condition a randomly generated complex matrix. In the case $N = M$, as $X_N = X_M$ is a unitary operator, the solution of the recurrent network furnishes a unitary $S_M$ matrix, which solves the problem. For $M>N$ the RNN furnishes a unitary solution $S_M$, and a unitary transfer function $T_M$, only if we embed the target gate $X_N$ in a unitary operator as in 3 with $O_C$ a random unitary matrix.

### 2.1. Single non-inferencing qutrit gate X

For the training of a gate $X_3$ defined by [2,43]

$$X_3 = \sum_{l=0}^{d-1} |l \oplus 1\rangle\langle l| = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{8}$$

The gate $X_3$ is obtained by an embedding dimension $M = 5$ and transfer function $U_5$ as in Fig. 2.

For $G = G_P$, the number of ODEs for the training of the network is minimal ($N = 3$). However, the solution is not unitary, as some channels are dropped out by the $N-$projector. The overall $M \times M$ transmission matrix $T_M$, after the training, is such that $T_M^\dagger \cdot T_M \neq I$ because the solution $S_M$ is not unitary. However, the system always reaches a stationary case.

A unitary solution is found by letting $G = G_U$ and involving the maximum number of ODEs in 7 with a unitary embedding of $X_N$ as in 3, i.e., adopting a further - randomly generated - unitary matrix $O_C$. The key point is that the system *finds a solution for any random unitary rigging of the matrix* $X_N$, that is, for any randomly assigned matrix $O_C$. This implies that we can train all these systems to realize different multi-level gates.

## 3. Inferencing gates

In the case that we do not know the transfer matrix of the system, we can still train the overall transmission matrix by using a neural network and infer $U_M$. Here we use an ANN to determine the training operators without measuring the transfer matrix. Figure 3 shows the scheme of the ANN, where the unitary matrix $U_M$ is represented by its elements $u_{ij}$, and the $w_{ij}$ are the adjustable weights. After training, the resulting $w_{ij}$ are the elements of the solution matrix $S_M$. For the sake of simplicity, we consider $S^{\text{out}} = \mathbf{1}_M$, as above. For a target $X_N$, we build the $T_M$ as in (3) by randomly generating the unitary complement $O_C$. As $T_M$ and $U_M$ are unitary, the

Optics EXPRESS

resulting $S_M$ is also unitary. One can use a non unitary $T_M$ by choosing, for example, $O_C = \mathbf{0}$. Correspondingly - after the training - $S_M$ is not unitary.

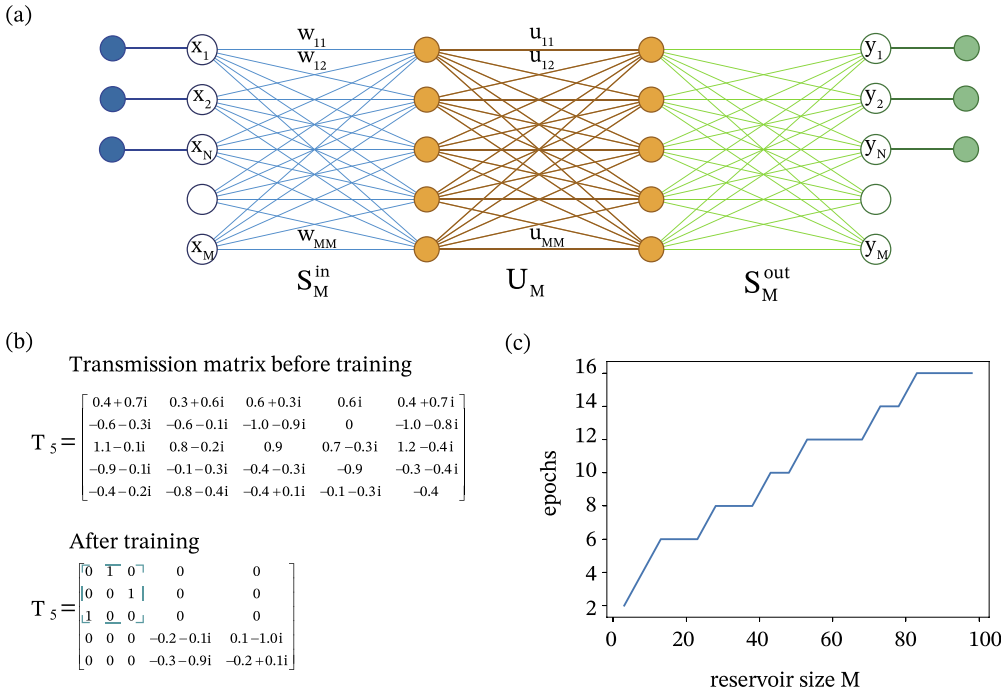


**Fig. 3.** Example of inference training of a random system ($M = 5$) to act as $X_3$ gate. (a) Neural network model (in our example $S_M^{\text{out}}$ is not used); (b) numerical examples for the trasmission matrix $T_M = U_M \cdot S_M^{\text{in}}$ before and after training; (c) scaling properties in terms of training epochs. Parameters: $n_{train} = 100$, $n_{valid} = 50$, $e_{valid} = 10^{-3}$, $n_{epoch} = 6$.

We randomly generate a set of input states $\mathbf{x}_i$, with $i = 1, \ldots, n_{train}$. Each input state is "labelled" with the target output $\mathbf{y}_i = T_M \cdot \mathbf{x}_i$. We remark that $\mathbf{x}_i$ and $\mathbf{y}_i$ are vector with size $M$. A further set of $n_{valid}$ validation rigged vectors is used to validate the training.

For any input $\mathbf{x}_i$ in the training set, we adjust the weights to minimize the error function

$$e_i = \frac{1}{N} \sum_N |\mathbf{y}_i - U_M \cdot W_M \cdot \mathbf{x}_i|^2 \tag{9}$$

with $\mathbf{y}_i = T_M \cdot \mathbf{x}_i$. After this training, we test the accuracy on the validation set. Each cycle of training and validation is denoted as "epoch".

Figure 3 shows the ANN for $N = 3$, and $M = 5$. In our model, we build a matrix $W_M$ of unknown weights. As we deal with complex quantities, $W_M$ is written as $W_M = W'_M + \imath W''_M$ with $W'_M$ and $W''_M$ real-valued matrices, whose elements form the weights of the ANN. Using random matrices as initial states, we end the iteration when the validation cost is below a threshold $\varepsilon_{valid}$.

### 3.1. Single-qutrit inference X-gate

Figure 3 shows the training of a single qutrit gate $X_3$ in 8. Similar results are obtained with other single qudit gates as $X^2$ and $Z$ and for higher dimensions. Training typically needs tens of iterations and scales well with the number of dimensions. Figure 3 shows an example with $N = 3$ and $M = 5$. Figure 3(c) shows that the number of training epochs $n_{epochs}$ scales linearly with the embedding space dimension $M$.

Optics EXPRESS

## 4. Spatial light modulator implementation

In the general case, one needs a unitary gate to train the complex medium, and a modulator to test different inputs signals. In practical and simplified implementations, the training gate and the input modulator can be made with a single device. It is possible to realize the ML design with a single spatial light modulator (SLM), as sketched in the inset of Fig. 4(a). Reference [1] already gave a recipe for implementing a unitary in a lossy way with a single SLM and a complex medium. However, here we follow the more recent but also lossy technique introduced in Ref. [5]. We consider an input plane wave represented by a constant vector $\mathbf{e}_N = 1, 1, \ldots, 1$ with dimension $N$, where $N$ is the number of pixels in the amplitude and phase SLM.

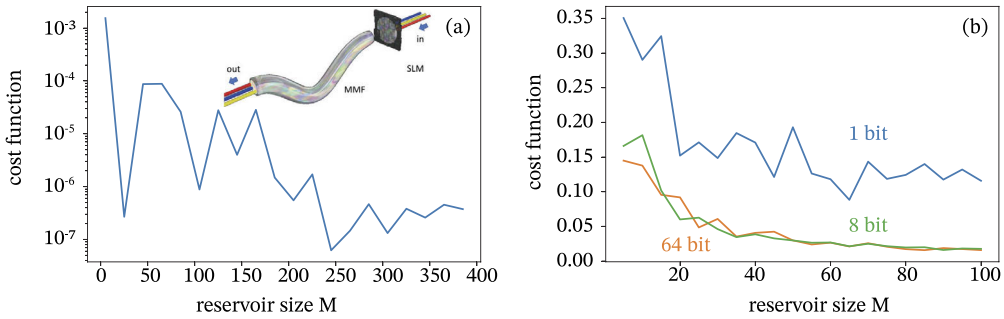


**Fig. 4.** (a) Error after 1000 training epochs versus the size of the reservoir $M$; the inset shows a sketch of the experimental implementation with a single spatial light modulator (SLM). (b) Error versus reservoir size $M$ after 1000 epochs with a single amplitude modulator (with sign) with quantized levels (different bit numbers are indicated).

We want to design a gate with input $\mathbf{x}$ and output $\mathbf{y}$, we generate the input $\mathbf{x}$ by an operator Diag($\mathbf{x}$), which has the first $N$ elements of $\mathbf{x}$ on the diagonal. Assuming that the ML algorithm has produced an operator $S_M$, the actual operator to be implemented on the SLM is $\tilde{S}_M = S_M \cdot \mathrm{Diag}(\mathbf{x})$. Note that $\tilde{S}_M$ encodes the input and hence changes for different inputs [5].

In other words, with a single SLM, after optimization for a given output $\mathbf{y}$, the training realizes $\tilde{S}_M$ for a fixed plane wave input $\mathbf{e}_M = 1, 1, \ldots, 1, 0, \ldots 0$ with $N$ ones and $M - N$ zeros.

### 4.1. Phase-only modulators

A pure phase modulator is implemented by the elements writing the model matrix for $\tilde{S}_M$ as $\cos(\phi_{ij}) + \imath \sin(\phi_{ij})$, with $\phi_{ij}$ the phase of the $i, j$ segment of the SLM. In Fig. 4(a), we show the performance of the training process, focusing on a single qutrit $X-$gate ($N = 3$), and varying the size of the reservoir $M$. If the reservoir is about one order of magnitude larger than the dimension of the gate, the algorithm converges in less than 1000 epochs, and the error decreases with $M$. The observed oscillations are due to the specific realization and are within the discretization error.

### 4.2. Sign modulators and quantized amplitude

A pure amplitude modulator is modeled by a real matrix $\tilde{S}_M$. A combination of an amplitude modulator, such as a digital micromirror device (DMD), along with spatial filtering, enables one to realize positive and negative values for $\tilde{S}_M$ [5]. The elements of the real $\tilde{S}_M$ are trained to provide the target output with the fixed plane wave at the input. Using typical functions in application program interfaces, such as *tensorflow.clip_by_value*, one can clip the values of the amplitude modulation (we use the range $[-1.0, 1.0]$). In contrast with the phase-modulator case, the performance in the amplitude modulation case is reduced. Our numerical experiments show that convergence (corresponding to a cost-function smaller than $10^{-4}$) is not reached. On the contrary, the error reaches a stable minimal value after about 1000 epochs. The minimal

Optics EXPRESS

error decreases with the size of the reservoir (Fig. 4(b)). In Fig. 4(b), we also account for the fact that modulator devices have limited resolution, and accessible modulation levels are quantized with a given number of bits. We can implement the level quantization in TensorFlow by using *tensorflow.quantize_and_dequantize*, after each iteration. In Fig. 4, we show results for phase-only modulation for the 1-bit case, corresponding to modulation levels $-1, 0, 1$, as well as for the 8 and 64-bit cases.

## 5. Conclusion

We have investigated the use of machine learning paradigms for designing linear multi-level quantum gates by using a complex transmitting multi-modal system. The developed algorithms are versatile and scalable when the unitary operator for the random system is either known or unknown. As the underlying problem is linear, the computational complexity is polynomial. For example, in the inferencing case (in which the transmission matrix of the complex medium is not known), the number of training epochs scales linearly with the system size. In an experimental realization at a large scale, the training time may be larger than the time during which the physical system is stable. In optical experiments with fibers or diffusing media, the transmission matrix changes with a time scale of the order of a few hours. Correspondingly, one can resort to continuous training, which can be optimized by the use of reinforcement learning, as reported in [44].

In conclusion, we gave evidence that reservoir computing enhanced by machine learning enables to design generalized single-qudit gates. The overall methodology is easily implemented by the TensorFlow application program interface and can be directly adapted to experimentally retrieved data. The method can be generalized to more complex information protocols, and embedded in real-world multi-modal systems.

## Funding

Horizon 2020 Framework Programme QuantERA grant QUOMPLEX, by National Research Council (CNR), Grant agreement ID 731473, the Dutch Research Council (NWO) Grant no. 680.91.037, and FWF Project.

## Disclosures

The authors declare no conflicts of interest.

## References

1. S. R. Huisman, T. J. Huisman, T. A. W. Wolterink, A. P. Mosk, and P. W. H. Pinkse, "Programmable multiport optical circuits in opaque scattering materials," Opt. Express **23**(3), 3102–3116 (2015).
2. A. Babazadeh, M. Erhard, F. Wang, M. Malik, R. Nouroozi, M. Krenn, and A. Zeilinger, "High-Dimensional Single-Photon Quantum Gates: Concepts and Experiments," Phys. Rev. Lett. **119**(18), 180510 (2017).
3. M. Malik, M. Erhard, M. Huber, M. Krenn, R. Fickler, and A. Zeilinger, "Multi-photon entanglement in high dimensions," Nat. Photonics **10**(4), 248–252 (2016).
4. 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, "Reconfigurable quantum photonic processor based on silicon nitride waveguides," Opt. Express **27**(19), 26842–26857 (2019).
5. M. W. Matthès, P. del Hougne, J. de Rosny, G. Lerosey, and S. M. Popoff, "Turning Optical Complex Media into Universal Reconfigurable Linear Operators by Wavefront Shaping," (2018). ArXiv:1810.05688.
6. P. Zhao, S. Li, X. Feng, S. M. Barnett, W. Zhang, K. Cui, F. Liu, and Y. Huang, "Universal linear optical operations on discrete phase-coherent spatial modes," (2018). ArXiv:1801.05092.
7. I. M. Vellekoop and A. P. Mosk, "Focusing coherent light through opaque strongly scattering media," Opt. Lett. **32**(16), 2309–2311 (2007).
8. I. M. Vellekoop, "Feedback-based wavefront shaping," Opt. Express **23**(9), 12189–12206 (2015).
9. R. Horstmeyer, H. Ruan, and C. Yang, "Guidestar-assisted wavefront-shaping methods for focusing light into biological tissue," Nat. Photonics **9**(9), 563–571 (2015).

Optics EXPRESS


10. D. Pierangeli, V. Palmieri, G. Marcucci, C. Moriconi, G. Perini, M. De Spirito, M. Papi, and C. Conti, "Deep optical neural network by living tumour brain cells," (2018). ArXiv:1812.09311.
11. D. Fu, Y. Zhou, R. Qi, S. Oliver, Y. Wang, S. M. H. Rafsanjani, J. Zhao, M. Mirhosseini, Z. Shi, P. Zhang, and R. W. Boyd, "Realization of a scalable laguerre-gaussian mode sorter based on a robust radial mode sorter," Opt. Express **26**(25), 33057–33065 (2018).
12. N. K. Fontaine, R. Ryf, H. Chen, D. T. Neilson, K. Kim, and J. Carpenter, "Laguerre-Gaussian mode sorter," (2018). ArXiv:1803.04126.
13. D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," Neural Netw. **20**(3), 391–403 (2007).
14. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing **70**(1-3), 489–501 (2006).
15. F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," Opt. Express **20**(20), 22783–22795 (2012).
16. G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," Nanophotonics **6**(3), 561–576 (2017).
17. N. Borhani, E. Kakkava, C. Moser, and D. Psaltis, "Learning to see through multimode fibers," Optica **5**(8), 960–966 (2018).
18. S. Lohani, E. M. Knutson, M. O'Donnell, S. D. Huver, and R. T. Glasser, "On the use of deep neural networks in optical communications," Appl. Opt. **57**(15), 4180–4190 (2018).
19. Y. Sun, Z. Xia, and U. S. Kamilov, "Efficient and accurate inversion of multiple scattering with deep learning," Opt. Express **26**(11), 14678–14688 (2018).
20. X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," Science **361**(6406), 1004–1008 (2018).
21. G. Favraud, J. S. T. Gongora, and A. Fratalocchi, "Evolutionary photonics: Evolutionary photonics for renewable energy, nanomedicine, and advanced material engineering," Laser Photonics Rev. **12**(11), 1870047 (2018).
22. N. Mohammadi Estakhri, B. Edwards, and N. Engheta, "Inverse-designed metastructures that solve equations," Science **363**(6433), 1333–1338 (2019).
23. D. Pierangeli, G. Marcucci, and C. Conti, "Large-scale photonic ising machine by spatial light modulation," Phys. Rev. Lett. **122**(21), 213902 (2019).
24. K. Wu, J. García de Abajo, C. Soci, P. Ping Shum, and N. I. Zheludev, "An optical fiber network oracle for np-complete problems," Light: Sci. Appl. **3**(2), e147 (2014).
25. D. Englund, H. Larochelle, M. Soljačić, M. Hochberg, M. Prabhu, N. C. Harris, S. Skirlo, S. Zhao, T. Baehr-Jones, X. Sun, and Y. Shen, "Deep learning with coherent nanophotonic circuits," Nat. Photonics **11**(7), 441–446 (2017).
26. L. Pilozzi, F. A. Farrelly, G. Marcucci, and C. Conti, "Machine learning inverse problem for topological photonics," Commun. Phys. **1**(1), 57 (2018).
27. M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, "Automated search for new quantum experiments," Phys. Rev. Lett. **116**(9), 090405 (2016).
28. T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, "Reinforcement learning with neural networks for quantum feedback," Phys. Rev. X **8**, 031084 (2018).
29. 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**(4), 044033 (2018).
30. Z. A. Kudyshev, S. Bogdanov, T. Isacsson, A. V. Kildishev, A. Boltasseva, and V. M. Shalaev, "Rapid classification of quantum sources enabled by machine learning," (2019). ArXiv:1908.08577.
31. S. Leedumrongwatthanakun, L. Innocenti, H. Defienne, T. Juffmann, A. Ferraro, M. Paternostro, and S. Gigan, "Programming linear quantum networks with a multimode fiber," Nat. Photonics (2019). ArXiv:1902.10678.
32. L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljačić, and Y. Bengio, "Gated Orthogonal Recurrent Units: On Learning to Forget," (2017). ArXiv: 1706.02761.
33. R. Dangovski, L. Jing, and M. Soljacic, "Rotational Unit of Memory," (2017). ArXiv:1710.09537.
34. A. Muthukrishnan and C. R. Stroud, "Multivalued logic gates for quantum computation," Phys. Rev. A **62**(5), 052309 (2000).
35. Y.-M. Di and H.-R. Wei, "Elementary gates for ternary quantum logic circuit," (2011). ArXiv:1105.5485.
36. M. Mirhosseini, O. S. Magaña-Loaiza, M. N. O'Sullivan, B. Rodenburg, M. Malik, M. P. J. Lavery, M. J. Padgett, D. J. Gauthier, and R. W. Boyd, "High-dimensional quantum cryptography with twisted light," New J. Phys. **17**(3), 033033 (2015).
37. S. Ecker, F. Bouchard, L. Bulla, F. Brandt, O. Kohout, F. Steinlechner, R. Fickler, M. Malik, Y. Guryanova, R. Ursin, and M. Huber, "Entanglement distribution beyond qubits or: How I stopped worrying and learned to love the noise," Phys. Rev. X **9**, 041042 (2019).
38. S. A. Goorden, M. Horstmann, A. P. Mosk, B. Škorić, and P. W. H. Pinkse, "Quantum-secure authentication of a physical unclonable key," Optica **1**(6), 421–424 (2014).
39. T. B. H. Tentrup, W. M. Luiten, R. van der Meer, P. Hooijschuur, and P. W. H. Pinkse, "Spatially encoded light for Large-alphabet Quantum Key Distribution," New J. Phys. **21**(12), 123044 (2019).
40. M. Leonetti, S. Karbasi, A. Mafi, E. DelRe, and C. Conti, "Secure information transport by transverse localization of light," Sci. Rep. **6**(1), 29918 (2016).

Optics EXPRESS

41. A. Di Falco, V. Mazzone, A. Cruz, and A. Fratalocchi, "Perfect secrecy cryptography via mixing of chaotic waves in irreversible time-varying silicon chips," Nat. Commun. **10**(1), 5827 (2019).
42. J. Wang, "Recurrent neural networks for solving linear matrix equations," Comput. & Math. with Appl. **26**(9), 23–34 (1993).
43. X. Gao, M. Krenn, J. Kysela, and A. Zeilinger, "Arbitrary *d*-dimensional pauli *x* gates of a flying qudit," Phys. Rev. A **99**(2), 023825 (2019).
44. J. Jašek, K. Jiráková, K. Bartkiewicz, A. Černoch, T. Fürst, and K. Lemr, "Experimental hybrid quantum-classical reinforcement learning by boson sampling: how to train a quantum cloner," Opt. Express **27**(22), 32454–32464 (2019).