

Coupled Nonlinear Delay Systems as Deep Convolutional Neural Networks

Bogdan Penkovsky,^{*} Xavier Porte, Maxime Jacquot, Laurent Larger, and Daniel Brunner
*FEMTO-ST/Optics Dept., UMR CNRS 6174, Université Bourgogne Franche-Comté,
 15B avenue des Montboucons, 25030 Besançon Cedex, France*



(Received 4 March 2019; revised manuscript received 20 June 2019; published 1 August 2019)

Neural networks are transforming the field of computer algorithms, yet their emulation on current computing substrates is highly inefficient. Reservoir computing was successfully implemented on a large variety of substrates and gave new insight in overcoming this implementation bottleneck. Despite its success, the approach lags behind the state of the art in deep learning. We therefore extend time-delay reservoirs to deep networks and demonstrate that these conceptually correspond to deep convolutional neural networks. Convolution is intrinsically realized on a substrate level by generic drive-response properties of dynamical systems. The resulting novelty is avoiding vector matrix products between layers, which cause low efficiency in today's substrates. Compared to singleton time-delay reservoirs, our deep network achieves accuracy improvements by at least an order of magnitude in Mackey-Glass and Lorenz time series prediction.

DOI: [10.1103/PhysRevLett.123.054101](https://doi.org/10.1103/PhysRevLett.123.054101)

Neural networks have emerged as the current disruptive computational concept. When cascading multiple network layers, these systems set the benchmark in multiple challenging tasks [1]. In such *deep* neural networks, layers are dedicated to highlight specific aspects of the input-information, and previous layers commonly serve as input of consecutive layers. Such a hierarchical arrangement is crucial for boosting the computational performance. In deep convolutional neural networks (CNN), layers convolute their input with spatial filters. By increasing filter width and step size, deeper layers focus on more general features, while local features are highlighted in earlier layers [2].

In the wake of deep neural networks' success, it was realized that their emulation on Turing or von Neumann machines is highly inefficient. This stimulated strong interest in the realization of neural networks in physical substrates whose architecture submit to the networks' topology. Particularly photonic systems, which offer key advantages for parallelization, are considered a promising future alternative. However, directly mapping the complex topology of a deep neural network onto a hardware substrates presents a significant challenge. Of essential importance are therefore concepts which strike a balance between architectural complexity and hardware implementation simplicity.

Among the various neural network architectures, reservoir computers [3] have emerged as especially interesting theoretical model systems [4–6] and promising candidates for hardware implementations. A reservoir computer is a complex recurrent neural network and conceptually corresponds to a high-dimensional nonlinear dynamical system. Training is restricted to the connections between the reservoir and its

output, and hence the nonlinear dynamical system's topology remains constant. This strongly assists implementations in physical substrates, resulting in a large number of realizations in nonlinear photonic [7] and other physical systems [8]. Yet, precisely this simplicity raised fundamental concerns regarding deep reservoirs. Recently it was found that, comparable to deep convolutional networks, a continuous change of *spatial* frequency in the response of consecutive layers appears beneficial [9,10]. The workhorse of the field have been nonlinear delay systems implementing time delay reservoirs (TDRs) [7,11,12]. These offer a compromise between good computing performance and exceptional ease of hardware implementation and serve as model systems for more complex hardware substrates [13–15].

We report on a deep reservoir scheme comprising hierarchically coupled nonlinear delay oscillators exhibiting dynamics on multiple timescales. Crucially, coupling between different layers is constant and training remains limited to the readout weights, in contrast to a proposed deep hardware TDR [16]. This is an essential simplification as it adheres to the conceptual simplicity motive, which strongly fosters hardware implementation. We find that cascading significantly and qualitatively improves computational performance when compared to a single layer reservoir of identical size. Crucially, our architectural simplicity curbs the challenges particular to physically implementing complex and large networks.

In Fig. 1, we schematically illustrate our deep TDR concept. Dynamics are governed by the following set of equations:

$$\tau_i \dot{x}_i(t) = -x_i(t) - \delta_i y_i(t) + \beta_i \sin^2[d_i(t) + b_i], \quad (1)$$

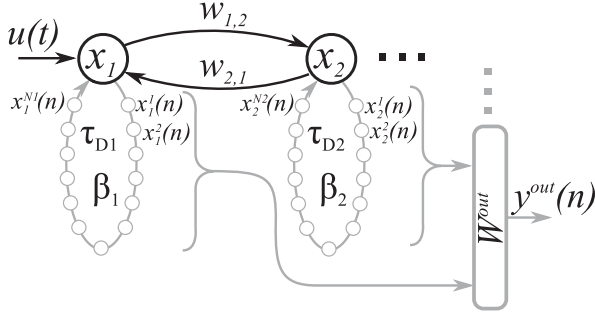


FIG. 1. Schematic of cascaded nonlinear oscillators acting as deep network, here consisting of two layers. Two coupled nonlinear delay systems $x_1(t)$ with states $x_2(t)$ implement individual time delay reservoir layers. Information is injected into the first system, and nonlinear nodes are coupled instantaneously according to weights $w_{1,2}$ and $w_{2,1}$. The readout layer has access to all layers.

$$\dot{y}_i(t) = x_i(t), \quad (2)$$

$$d_i(t) = x_i(t - \tau_{Di}) + \sum_{p=\pm 1} w_{i+p,i} x_{i+p}(t) + \rho_i u(t), \quad (3)$$

$$\rho_{i \neq 1} = 0, \quad \delta_i = 0. \quad (4)$$

The state of the delay-coupled node in layer $i \in \{1, \dots, I\}$ is given by $x_i(t)$, and we use the \sin^2 nonlinearity often employed in photonic TDRs [17,18]. Due to inertia, dynamics generally experience low-pass (LP) filtering according to a fast time constant τ_i , which can be extended to bandpass (BP) filtering when a slow time constant δ_i is added [19]. Each layer's nonlinearity is weighed by bifurcation parameter β_i , and the nonlinearity's argument contains constant bias b_i and a time-dependent drive $d_i(t)$, see Eq. (3). Drive $d_i(t)$ features self-feedback delayed by τ_{Di} and potentially bidirectional coupling to adjacent layers according to coefficients $w_{i \pm 1, i}$. Only the first layer is coupled to $u(t)$, see Eq. (4). External drive $u(t)$ encodes the information to be processed $s(t)$ according to the temporal masking procedure which implements a linear matrix multiplication [12,20]. We have employed a desynchronized information injection procedure in which each value of $s(t)$ is kept for an input-masking length of $0.8\tau_{Di}$.

According to Eq. (3), layer i is coupled to layer $i+1$ ($i-1$) according to the fixed connection weight $w_{i+1,i}$ ($w_{i-1,i}$), and coupling to $i-1=0$ is unphysical and hence eliminated. Therefore, a recurrent layer simply consists of one hardware nonlinearity, one linear delay line and its fixed connections to previous or consecutive layer. This has multiple consequences. First, interlayer coupling is instantaneous and constant in time. Training of the interlayer connections, a long-time open question for deep reservoirs [10] and significant challenge for full hardware integration [21], is therefore avoided. Second,

such a minimal complexity architecture [22] can readily be implemented in hardware [8]. Finally, it allows establishing a clear mapping from deep TDRs onto deep convolutional neural networks.

The fact that TDR layers can be termed *convolutional* originates from a nonlinear dynamical node's response to perturbations. The state of a nonlinear node in layer i is given by the convolution between its impulse response function $h_i(t)$ and its drive $d_i(t)$. Combined with a normalization of continuous time t by feedback delay τ_{Di} , one can express the dynamical evolution by

$$\frac{t}{\tau_{Di}} = n + \sigma_i / N_i, \sigma_i \in \{1, N_i\}, n \in \{1, 2, \dots\}, \quad (5)$$

$$x_i^{\sigma_i}(n) = \int_{-\infty}^{n+\sigma_i} h_i(n + \sigma_i - \xi) \sin^2[d_i(\xi - 1) + b_i] d\xi, \quad (6)$$

with N_i as the number of neurons in layer i , see Fig. 1. First, Eqs. (5) and (6) map continuous time t onto discrete time n and node σ_i 's position relative to delay time τ_{Di} . Details of this temporal embedding technique can be found in [11,12,23]. Second, expressing the dynamical evolution via the convolution operation shows that a node's impulse response function corresponds to the convolution kernels of a CNN layer. Crucially, coupling created with such a dynamical convolution can directly be translated to the convolution kernel of spatiotemporal networks [12,24].

The analogy between cascaded TDRs and deep convolutional networks goes further. Layers of a CNN commonly feature convolution kernels whose width increases the further back in the cascaded hierarchy a layer is located [2]. This operation is often associated with generalization: convolution with wider filters reduces the importance of local features in their input, while more general aspects are highlighted. The cascaded arrangement of layers in CNNs therefore produces layers which accentuate different input information features. In TDRs, increasing the convolution kernel's width corresponds to widening $h_i(t)$, see the Supplemental Material [25]. Here this is realized by an additional low-frequency cutoff according to timescale δ_i in Eqs. (1) and (2), and we enforce widening kernels. In Fig. 2 we show the response of a three-layer deep TDR driven by the chaotic Mackey-Glass sequence. Each sample corresponds to $\delta t = 1$ time step of the Mackey-Glass system, for which we used the same parameters as in Ref. [3]. The parameters are $\beta_{2,3} = 1.1$, $\tau_1 = 6 \times 10^{-3}$, $\tau_{2,3} = 7 \times 10^{-3}$, $\tau_{Di} = 17.85$, $\Phi_0 = 0.2$, $\delta_{2,3} = 0.01$, $w_{1,2} = 0.7$, $w_{2,3} = 0.8$, $w_{2,1} = w_{3,2} = 0$. Responses are plotted in spatiotemporal representation [23,26], where nodes are arranged along σ_i and the temporal evolution is along discrete time n , with n typically close to a system's delay τ_D [12]. As we move into higher layers, from (a) to (c) in Fig. 2, dynamics do highlight different spatiotemporal scales. Our deep TDR

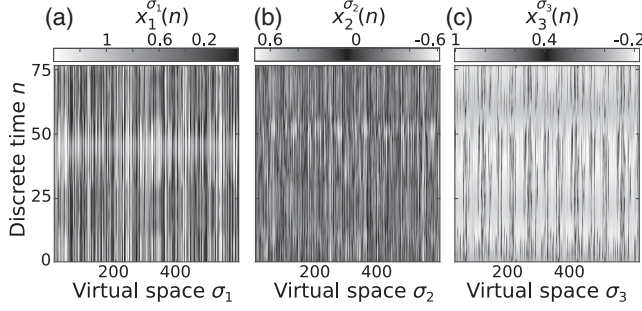


FIG. 2. Neuron responses $[x_i^{\sigma_i}(n)]$ found in layers $i = 1$ (a), $i = 2$ (b), and $i = 3$ (c), illustrated in a spatiotemporal (σ_i, n) representation. The spatial frequency along virtual space σ_i continuously decreases for the higher layers. Comparable functionality is implemented in deep convolutional networks or deep reservoirs.

therefore hosts features much like those taken into consideration in the design of CNNs.

Creating a computational result requires to connect the deep TDR to an output via weights adjusted during learning. Our readout layer has access to all virtual nodes of all network layers, and the system's output is created according to

$$y_j^{\text{out}}(n) = \sum_i^I \sum_{\sigma_i}^{N_i} W_{i,\sigma_i,j}^{\text{out}} x_i^{\sigma_i}(n). \quad (7)$$

Here, j is the dimension of the system's output, which depends on the particular task. Common methods to obtain \mathbf{W}^{out} are based on linear or ridge regression, and \mathbf{W}^{out} is optimized using a representable set of training data [3,12]. In experimental systems, these methods can be implemented in auxiliary hardware like field-programmable gate arrays [27], or can to a degree be replaced by Boolean learning algorithms [14]. Recurrent neural networks are primarily relevant for processing temporal information. We therefore task the system to predict chaotic sequences Δn time steps into the future. Training optimizes \mathbf{W}^{out} for

$y^{\text{out}}(n)$ to approximate target $y^T(n) = s(n + \Delta n)$, $n \in \{1, n^T\}$, where $n^T = 5000$ are the number of samples used for training. We quantify the prediction's quality for $n > n^T$, hence on testing data not used for training the system, according to the normalized mean square error $\text{NMSE} = 1/n^T \sum_{n=1, \dots, n^T} [y^T(n) - y^{\text{out}}(n)]^2 / (\sigma^T)^2$, where σ^T is the target-signal's standard deviation.

First, we predict the chaotic Mackey-Glass delay equation, which features a delay of 17 time steps. By predicting ahead twice its delay ($\Delta n = 34$), the objective is long-term prediction. We establish a systematic interpretation by cascading only two TDR layers ($N_i = 600$) and display the performance dependence on the exhaustively scanned system parameters in Fig. 3. We keep $\tau_1 = 0.6 \times 10^{-3}$, $\tau_2 = 0.6 \times 10^{-3}$, $\tau_{D1,2} = 12$, $b_{1,2} = 0.2$, $\rho_1 = 8$, and $\delta_2 = 0.01$ constant, with their values mostly based on empirical observations. In order to provide a baseline reference for other topologies, we evaluate uncoupled layers ($w_{1,2} = w_{2,1} = 0$) and scan the bifurcation parameter plane (β_1, β_2) , see Fig. 3(a). Importantly, for this test we set $\rho_2 = \rho_1$ and hence couple the BP layer to the same input as the LP layer. We find a clear optimum for β_1 , while performance dependence on β_2 is less pronounced. The lowest error (NMSE = 8.3×10^{-6}) is obtained at $\beta_1 = 1.4$ and $\beta_2 = 1.2$.

We now turn to different coupling topologies and disconnect the second layer from the system's input information ($\rho_2 = 0$, $w_{1,2} = 0.7$, $w_{2,1} = 0.6$). Figure 3(b) shows that bidirectional coupling significantly alters the optimal bifurcation parameters and results in a equally pronounced β_2 dependency. We obtain $\text{NMSE} = 8.8 \times 10^{-6}$ at $\beta_1 = 1.4$ and $\beta_2 = 1.2$, and the performance benefit of bidirectional coupling is negligible. Continuing with the optimized value of β_i , we focus on the coupling topology by exhaustively scanning $w_{1,2}$ and $w_{2,1}$, see Fig. 3(c). The NMSE reveals some performance sensitivity upon the coupling-strength from the first to the second layer. The most important finding is, however, that there is a systematic dependency upon $w_{2,1}$: the clear global performance optimum is found for

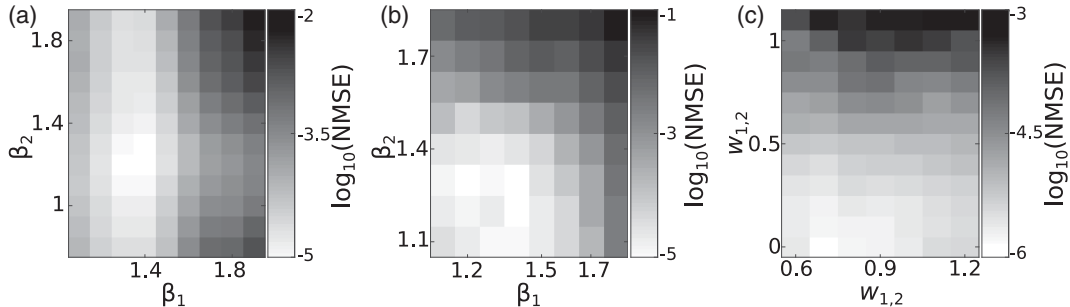


FIG. 3. Coupling strongly enhances the network's performance for predicting the chaotic Mackey-Glass time series by $\Delta n = 34$ steps into the future. (a) The uncoupled system [both systems receive the input sequence $u(t)$] ($w_{1,2} = w_{2,1} = 0$) achieves $\text{NMSE} = 8.3 \times 10^{-6}$. (b) Bidirectional coupling ($w_{1,2} = 0.7$, $w_{2,1} = 0.6$) results in no improvement ($\text{NMSE} = 8.8 \times 10^{-6}$). (c) The decisively best performing architecture is the unidirectional coupling between the recurrent layers, i.e., feed-forward connections ($w_{1,2} = 1.4$, $w_{2,1} = 0$): $\text{NMSE} = 1.3 \times 10^{-6}$.

TABLE I. Comparison for different architectures with identical total number of neurons $N = 1200$. Layers are lp = low pass, bp = bandpass. LZ: Lorenz chaotic time series one step prediction parameters: $\tau_1 = 0.006$, $\tau_2 = \tau_3 = 0.007$, $\delta_2 = \delta_3 = 0.01$, $\beta_1 = 1.5$, $\beta_2 = \beta_3 = 1.2$.

Nodes per layer	Coupling strength	LZ NMSE
1200 lp	...	7.6×10^{-7}
600 lp, 600 bp	$w_{1,2} = 1.1$	5.7×10^{-7}
400 lp, 400 bp, 400 bp	$w_{1,2} = w_{2,3} = 1.1$	2.5×10^{-7}

unidirectional coupling with $w_{2,1} = 0$. The achieved prediction error (NMSE = 1.3×10^{-6}) is ~ 3 times smaller than for the bidirectional and the uncoupled systems, confirming the benefit of the hierarchical arrangement between consecutive network layers also for TDRs.

To further generalize our finding, we turn to predicting the chaotic Lorenz system. The Lorenz system is a three-dimensional set of ordinary differential equations. Each sample corresponds to $\delta t = 0.02$ time steps, and we used the same parameters as in Ref. [28]. The input information was the Lorenz system's first dimension $x(n)$, and the prediction target was $y^T(n) = x(n+1)$; hence, $\Delta n = 1$. Results are listed in Table I. Prediction performance is again enhanced by the addition of two layers in a unidirectional configuration. However, on a first glance the positive benefit appears to be smaller.

Until now prediction only evaluated the system via predicting ahead by distance Δn . A more suited approach to determine the capacity of approximating a chaotic system's behavior is based on the so-called teacher forcing [3]. After training using $\Delta n = 1$, the system's input becomes its own output, $s(\tilde{n}) = y^{\text{out}}(\tilde{n}-1)$, $\tilde{n} = n - n^T$, $n > n^T$. The TDR becomes an autonomous predictor of the learned system [3], and the autonomous evolution enables comparison to the original chaotic sequence over long intervals. Crucially, this corresponds to predicting until \tilde{n} only relying information of the original signal at n^T ; the prediction autonomously advances from there. This reveals how well the chaotic system as a whole is approximated by the neural network.

Figures 4(a) and 4(c) show autonomous evolution for Lorenz and Mackey-Glass prediction using three cascaded TDR layers with unidirectional coupling. The prediction targets are the black solid data. The positive impact of deep TDRs is apparent, and particularly striking when predicting the Lorenz system, see Fig. 4(a). Rather than chaotic excursions along an attractor, the autonomous single layer TDR quickly converges to a dynamical state resembling a limit cycle and therefore fails to reproduce its target system. Only with the three layers coupled in a deep, unidirectional topology the network is capable of an excellent approximate of Lorenz chaos. This is also visible from the temporal divergence measured as the Euclidean distances between the Taken's reconstructed attractors of $\mathbf{y}^{\text{out}}(\tilde{n})$ and $\mathbf{y}^T(\tilde{n})$, see Figs. 4(b) and 4(d). The solid black lines indicate the divergence according to the maximum Lyapunov exponent

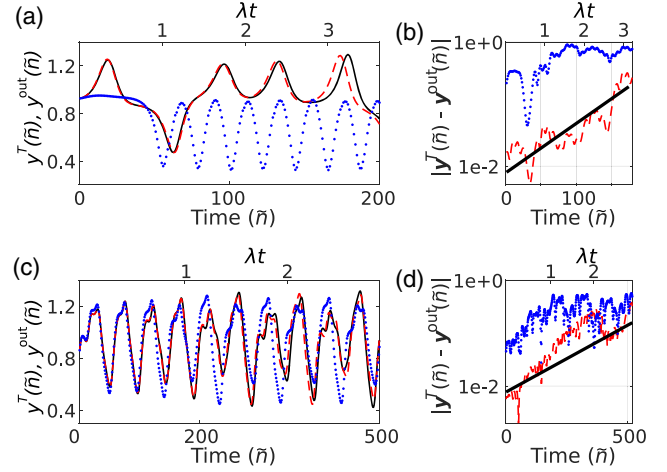


FIG. 4. When connecting the system to its own predicted output at $\tilde{n} = 1$, its dynamical evolution becomes autonomous from the original chaotic time series. The top x axis is in units of the inverse Lyapunov exponent. The long-term prediction performance for predicting the Lorenz (a) and Mackey-Glass system (c) via the connected (not connected) system as red dashed (blue dotted) data. Divergence between the predicted and the original attractors are shown in (b) and (d) for Lorenz and Mackey-Glass, respectively. The solid line indicates divergence according to the largest Lyapunov exponent.

(Mackey-Glass: $\lambda_{\text{max}} = 5.8 \times 10^{-3}$, Lorenz: $\lambda_{\text{max}} = 0.91$). Cascading layer improves prediction by a factor of 20 and 10.5 for Lorenz and Mackey-Glass prediction, respectively. The substantial improvement and fundamental importance of the cascaded, three-layer deep TDR architecture can be further appreciated by inspection of the resulting return maps, see the Supplemental Material [25].

We shall finish our investigation by also discussing limitations of our approach. The range of possible kernel shapes is limited by physical constraints and have not been optimized during training, through this is possible in principle. Also, deep TDRs do not yet reach the accuracy of the original spatiotemporal reservoir [3]. Predicting the Mackey-Glass time series 84 steps into the future results in NMSE = $10^{-4.4}$ with our deep TDR, while the original reservoir achieves NMSE = $10^{-8.4}$ [3]. However, multiple simple additions to the current concept could still significantly improve performance [29,30]. Using current high-performance hardware [31], CNN still run five times slower than TDRs [11]. However CNNs are optimized via back propagation, which will certainly result in lower errors than deep TDRs. If error back propagation can be realized in deep hardware networks remains questionable, while training of our system retains the simplicity and elegance of reservoir computing.

To conclude, we have introduced an elegant scheme for deep convolutional networks in a simple architecture of coupled nonlinear oscillators with delay. Information processing conditions conceptually comparable to deep convolutional neural networks with widening convolution kernels are achieved by cascading TDRs with increasingly

longer internal timescales. Intra- and interlayer connectivity can be adjusted via the oscillators' timescales, providing a practical control mechanism for hardware realizations.

Applied to both, Mackey-Glass and Lorenz chaos prediction, our concept significantly improves the quality of long-term predictions and proofs essential in the case of Lorenz forecasting. Recently, reservoirs have been demonstrated to infer a chaotic oscillator's hidden degrees of freedom [28] and to predict the evolution of chaotic spatiotemporal systems far into the future [32]. Temporal structure found in the divergence between prediction and target, such as in Fig. 4(d), could be addressed via further optimizing timescales τ_i and δ_i .

Finally, we would like to point out the large variety of possible hierarchical TDR networks. Hybrid systems, where for some or all layers self-feedback is removed, would incorporate feed-forward architectures [33]. Layers featuring excitable solitons can potentially create long term memory [34] and, when combined with the reported LP and BP layers, physically implement long-short term memory networks [35]. This opens possibilities in new domains like natural language processing and sequence generation.

This work was supported by the EUR EIPHI program (Contract No. ANR-17-EURE-0002), by the BiPhoProc ANR project (No. ANR-14-OHRI-0002-02), by the Volkswagen Foundation NeuroQNet project and the ENERGETIC project of Bourgogne Franche-Comté. X.P. has received funding from the European Unions Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant Agreement No. 713694 (MULTIPLY).

*Present address: Centre de Nanosciences et de Nanotechnologies, Université Paris-Sud, CNRS, Université Paris-Saclay, 91405 Orsay, France.

- [1] Y. LeCun, Y. Bengio, and G. Hinton, *Nature (London)* **521**, 436 (2015).
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Advances in Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, 2012), pp. 1097–1105.
- [3] H. Jaeger and H. Haas, *Science* **304**, 78 (2004).
- [4] Z. Lu, B. R. Hunt, and E. Ott, *Chaos* **28**, 061104 (2018).
- [5] M. Inubushi and K. Yoshimura, *Sci. Rep.* **7**, 10199 (2017).
- [6] S. Marzen, *Phys. Rev. E* **96**, 032308 (2017).
- [7] G. Van der Sande, D. Brunner, and M. C. Soriano, *Nanophotonics* **6**, 561 (2017).
- [8] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, *Neural Netw.* **115**, 100 (2019).
- [9] C. Gallicchio and A. Micheli, *European symposium on artificial neural networks, computational intelligence and machine learning* (2016).
- [10] C. Gallicchio and A. Micheli, [arXiv:1712.04323](https://arxiv.org/abs/1712.04323).
- [11] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, *Phys. Rev. X* **7**, 011015 (2017).
- [12] D. Brunner, B. Penkovsky, B. A. Marquez, M. Jacquot, I. Fischer, and L. Larger, *J. Appl. Phys.* **124**, 152004 (2018).
- [13] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, *Nat. Photonics* **11**, 441 (2017).
- [14] J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, *Optica* **5**, 756 (2018).
- [15] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, M. Jarrahi, and A. Ozcan, *Science* **361**, 1004 (2018).
- [16] M. Nakajima, S. Konisho, K. Tanaka, and T. Hashimoto, in *Cognitive Computing, Merging Concepts and Hardware, Hannover, 2018*, https://drive.google.com/file/d/1Cgg_eKxN_WdQ2C0AyG4DFqAoqn5c59lF.
- [17] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, *Opt. Express* **20**, 3241 (2012).
- [18] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, *Sci. Rep.* **2**, 287 (2012).
- [19] L. Larger, B. Penkovsky, and Y. Maistrenko, *Phys. Rev. Lett.* **111**, 054103 (2013).
- [20] L. Appeltant, M. C. Soriano, G. V. D. Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, I. Fischer, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, *Nat. Commun.* **2**, 468 (2011).
- [21] P. Antonik, F. Duport, M. Hermans, A. Smerieri, M. Haelterman, and S. Massar, *IEEE Trans. Neural Networks Learn. Syst.* **28**, 1 (2016).
- [22] M. C. Soriano, D. Brunner, M. Escalona-Morán, C. R. Mirasso, and I. Fischer, *Front. Comput. Neurosci.* **9**, 68 (2015).
- [23] F. T. Arecchi, G. Giacomelli, A. Lapucci, and R. Meucci, *Phys. Rev. A* **45**, R4225 (1992).
- [24] J. D. Hart, L. Larger, T. E. Murphy, and R. Roy, [arXiv:1808.04596](https://arxiv.org/abs/1808.04596).
- [25] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevLett.123.054101> for impulse response function and Lorenz return map.
- [26] F. Marino and G. Giacomelli, *Phys. Rev. E* **98**, 060201(R) (2018).
- [27] M. Hermans, P. Antonik, M. Haelterman, and S. Massar, *Phys. Rev. Lett.* **117**, 128301 (2016).
- [28] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, *Chaos* **27**, 041102 (2017).
- [29] R. Martinenghi, S. Rybalko, M. Jacquot, Y. K. Chembo, and L. Larger, *Phys. Rev. Lett.* **108**, 244101 (2012).
- [30] L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega, *Neural Netw.* **55**, 59 (2014).
- [31] N. P. Jouppi *et al.*, [arXiv:1704.04760](https://arxiv.org/abs/1704.04760).
- [32] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, *Phys. Rev. Lett.* **120**, 024102 (2018).
- [33] S. Ortín, M. C. Soriano, L. Pesquera, D. Brunner, D. San-Martín, I. Fischer, C. R. Mirasso, and J. M. Gutiérrez, *Sci. Rep.* **5**, 14945 (2015).
- [34] B. Romeira, R. Avó, J. M. L. Figueiredo, S. Barland, and J. Javaloyes, *Sci. Rep.* **6**, 19510 (2016).
- [35] S. Hochreiter and J. Schmidhuber, *Neural Comput.* **9**, 1735 (1997).