Learn one size to infer all: Exploiting translational symmetries in delay-dynamical and spatiotemporal systems using scalable neural networks

Mirko Goldmann [*], Claudio R. Mirasso, Ingo Fischer [†] and Miguel C. Soriano [‡]
Instituto de Física Interdisciplinar y Sistemas Complejos (IFISC, UIB-CSIC), Campus Universitat de les Illes Balears E-07122, Palma de Mallorca, Spain

(Received 5 November 2021; revised 20 June 2022; accepted 27 September 2022; published 21 October 2022)

We design scalable neural networks adapted to translational symmetries in dynamical systems, capable of inferring untrained high-dimensional dynamics for different system sizes. We train these networks to predict the dynamics of delay-dynamical and spatiotemporal systems for a single size. Then, we drive the networks by their own predictions. We demonstrate that by scaling the size of the trained network, we can predict the complex dynamics for larger or smaller system sizes. Thus, the network learns from a single example and by exploiting symmetry properties infers entire bifurcation diagrams.

DOI: 10.1103/PhysRevE.106.044211

I. INTRODUCTION

Due to their many degrees of freedom and potentially multiple timescales, the prediction and analysis of complex system dynamics represent challenging tasks. Tackling unavailable analytical models, machine learning methods emerged [1–3] that learn from data and forecast the dynamics of biological [4,5], climate [6], spatiotemporal [7,8], and other complex systems [9–11]. More recently, incorporating symmetries is considered to guide such data-driven models towards preserving conservation laws [12], improve prediction accuracy [13], and might yield efficient representations of learned processes [14]. Nonetheless, machine learning demands large amounts of data, and yet predictions are mainly restricted to the dynamical regime they observed during training. In recent works [15–18], parametrized neural networks are being studied for the prediction of untrained dynamics. Thereby, a neural network needs to be trained on several examples, often covering different dynamical regimes. After training, changing the parametrization renders the network the ability to generalize and predict untrained dynamics, transformations of chaotic attractors, and closeby bifurcations. However, in real-world applications, system parameters can often not be easily changed, and, consequently, certain parameter regimes are hardly accessible. Here, the question arises—*Is it possible to infer untrained (size-dependent) dynamical regimes of a complex system whereas learning from one example related to a certain system size only?*

In this paper, we exploit symmetries in dynamical systems by designing neural networks that exhibit: (i) excellent generalization properties allowing us to infer untrained dynamical properties for a wide tuning of bifurcation parameters and (ii) high prediction capabilities. Here, we design scalable neural networks that satisfy the same symmetries as the dynamical system under study. We train these networks on a single time series of either delay-dynamical or spatiotemporal systems with a fixed size. After training, notably, the size of the trained network can be scaled up or down by adding or removing neurons exploiting the translational symmetry in these systems. Without further adaptations, the scaled network generalizes from learned to untrained dynamics and enables far-reaching inferences revealing bifurcations to various dynamical regimes. In the following, we first focus on delay-dynamical systems and provide a single neural network that infers the entire bifurcation diagram for changing the delay whereas learning from a single example and exploiting temporal translational symmetry. Based on the analogies between delay-dynamical and spatiotemporal systems can interpret the symmetry in delay-dynamical systems also as a quasispatial symmetry [19–21]. For comparison, in Sec. III, we apply our approach analogously to the real spatial translational symmetry of a Kuramoto-Sivashinsky (KS) model with periodic boundary conditions. Using a parallel network architecture we can train the network on a single spatial size and then infer untrained spatial extensions.

II. INFERENCE OF DELAY SYSTEM DYNAMICS

Dynamical systems with delayed coupling of system variables play an important role in many real-world contexts, such as climate systems [22], epidemiological models [23], biological systems [24], control systems [25–27], and photonic systems [28–31]. Since the evolution of delay systems relies on a continuous history function $h(t),\ t \in [0, -\tau]$, the phase space of these systems is infinite dimensional [32]. Depending on the length of the delay and other system parameters, these systems either converge to fixed points, limit cycles, or evolve on chaotic attractors [29–31,33]. Furthermore, with increasing delay length, these systems exhibit extensive chaotic behavior, i.e., the maximal dimension of their chaotic attractors scales linearly with the delay time [34].

[*]mirko@ifisc.uib-csic.es
[†]ingo@ifisc.uib-csic.es
[‡]miguel@ifisc.uib-csic.es

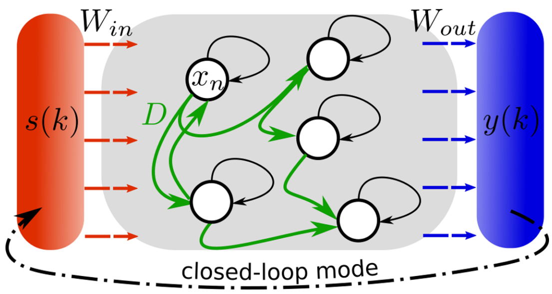

FIG. 1. Scheme of the delayed echo state network. The reservoir contains delayed internode weighted connections (green) where the delay D of the nodes connections can be adjusted.

TABLE I. List of parameters used for learning the Mackey-Glass system.

Parameter	Symbol	Value
Network size	K	1000
Initial steps	N_{init}	5000
Training steps	N_{train}	25 000
Feedback gain	β	0.176
Input gain	γ	1.24
Spectral radius	ρ	0.84
leak rate	α	0.75

In the following, we design echo state networks that incorporate a delay in their topology, as illustrated in Fig. 1. Due to this delay, the network exhibits the same temporal translational symmetry as the delay-dynamical system it is built to predict [19]. During training, the network delays will be tuned to match those of the target system, and the readout weights are adapted to implement one-step-ahead prediction. After learning the time evolution operator of the delay system, the network can be scaled up or down by extending or shortening its inherent delay to infer the untrained dynamics corresponding to shorter or longer delay, respectively. The dynamical evolution of the delayed echo state network (dESN) is governed by the following equation:

$$\vec{x}(n+1) = \alpha\vec{x}(n)+\beta \tanh[\mathbf{W}\vec{x}(n-D) + \gamma\mathbf{W}_{\text{in}}s(n) + \mathbf{W}_b], \tag{1}$$

where $\vec{x}(n) \in \mathbb{R}^K$, K is the network network size, α is the leak term, β is a feedback gain, γ is the input gain, and D is the delay length. The total size of the network scales linearly with D as all previous states $x(n)$ up to $x(n-D)$ must be stored. Thus, the delay can be seen as a signal through a cue of D hidden linear neurons along each delayed connection. However, the read-out dimension of the network is independent of the delay length since $\mathbf{W}_{\text{out}} \in \mathbb{R}^K$. The randomly drawn matrix $\mathbf{W}_{\text{in}} \in \mathbb{R}^K$ gives the connection between the input $s(n) \in \mathbb{R}$ and the network, whereas $\mathbf{W}_b \in \mathbb{R}^K$ gives a random bias to each node. The delayed connections are weighted via $\mathbf{W} \in \mathbb{R}^{KK}$, the elements of which are randomly drawn from a uniform distribution $\mathcal{U}[-1, 1]$ with a sparsity of 1.5%.

The training data set used only contains a time series of a delay system with a single fixed delay. Here, we consider inferring the dynamics of a Mackey-Glass (MG) delay system [35], however, the same holds for other delay systems as we demonstrate in Appendix D by performing a similar study for an Ikeda-type delay system. The dynamics of the MG delay system is given by the following equation:

$$\dot{s}(t) = -\frac{s(t)}{T_0} + \frac{0.2s(t-\tau)}{1+s(t-\tau)^{10}}. \tag{2}$$

We set the characteristic relaxation time $T_0 = 10$ and the system delay $\tau = 100$ and generate a time series $s(t)$ that is sampled with $\Delta t = 1$ to obtain the training data set $s(n)$. For these parameters, the system evolves along a chaotic attractor as depicted in Fig. 3. Before training the reservoir on the MG time series, the delay D of the dESN must be adjusted to the delay of the target system. We find the optimal performance when we match the delay of the reservoir connections directly with the delay of the MG system, i.e., by setting $D = \tau/\Delta t = 100$. In most real-world systems, however, the delay is not known *a priori* and needs to be extracted from the data. As we show in Appendix A, we can optimize the reservoir delay D by determining the best performance in a one-step-ahead prediction that reveals the delay underlying the training data. All other hyperparameters of the dESN are optimized using Bayesian optimization and are given in Table I.

During training, the dESN is driven by the input $s(n)$ with added white noise. The addition of noise during training was shown to improve the stability of the reservoir in the closed-loop mode [36]. The network's output layer is optimized using linear regression to predict the next step in the training time series. Once the output weights W_{out} are computed, the reservoir is decoupled from the input $s(n)$ and the prediction of the reservoir $\hat{y}(n) = \mathbf{W}_{\text{out}}\vec{x}(n)$ is returned as the new input. Consequently, the reservoir evolves autonomously in the closed-loop mode [37–39]. Reservoir computing [2,40–42] is applied here as it offers fast and efficient training; nevertheless, we expect that other training methods yield comparable results.

In the closed-loop mode, the reservoir evolves along the attractor on which it was trained as depicted in Figs. 2(a) and 2(f). When resetting the delay D of the dESN connections, we do not need any additional data for initialization, no further retraining, and the output weights can be kept fixed. As shown in Figs. 2(g)–2(i), the dESN generates high-dimensional chaotic attractors at $D = 17,\ 200$, and a limit cycle at $D = 39$. These inferred dynamical states are similar to those of the original MG system with delay $\tau = 17,\ 39$, and 200 shown in Figs. 2(b)–2(d). Consequently, it is sufficient that the dESN learned from a single time series of the MG system with delay $\tau = 100$ to infer the dynamics for other delay lengths. By scanning the delay of the dESN from $D = 1$ to 120 we generate a bifurcation diagram of the inferred dynamics as shown in Fig. 2(j). Depending on D, the dESN exhibits chaotic behavior, transitions to intermittent limit cycles, for example, at $D = 34$ and 39, period-doubling bifurcations at $D =$ 14 and 16, and transitions to a stable fixed point at $D = 5$. The observed transitions in the dESN coincide with those found for the original MG system shown in Fig. 2(e). Furthermore, in Appendix C we show that the dESN can infer multistabilities in the delay range $\tau,\ D = 70$–80. Röhm *et al.* [43] recently presented that reservoir computing is capable of

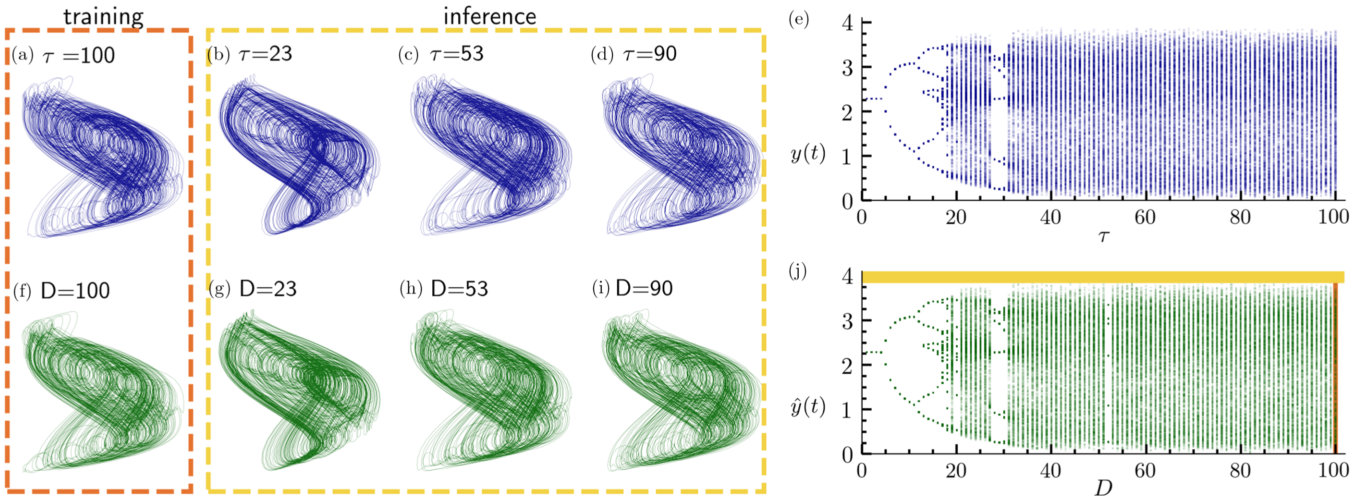

FIG. 2. Two-dimensional projection [x-axis $y(t)$, y-axis $y(t-\tau)$] of attractors of the chaotic Mackey-Glass system for different delay lengths in (a)–(d). (e) Bifurcation diagram generated using the Mackey-Glass delay system. Inferred attractors by the dESN trained on a single example of the Mackey-Glass system with $\tau = 100$ shown in (f)–(i). (j) Bifurcation diagram inferred by the dESN trained on data of a Mackey-Glass system with $\tau = 100$.

predicting multistabilities in a dynamical system even if the reservoir is trained only around one of the attractors. Here, our dESN infers infers coexisting attractors, whereasit was trained in a dynamical regime where there was no multistability.

In Fig. 2 we show that the dESN trained at the MG system with $\tau = 100$ can infer dynamics of the MG system with other delay lengths. In the following, we further underline how precise the dESN infers the dynamics even far from the training example. Therefore, in Fig. 3, we show the inference of the dESN with $D = 30$ compared to the original dynamics. We divide the prediction capabilities into two regimes, which we term weather and climate [36]. Here, the weather regime refers to the short-term behavior, when going from externally driven operation to autonomous operation. As shown in Fig. 3(a), the dESN can precisely continue the trajectory of a MG system with $\tau = 30$ for around 1000 steps, which corresponds to approximately nine Lyapunov times. Due to the chaotic behavior of the MG system with $\tau = 30$, two closely initialized trajectories diverge in time where the divergence rate is given by the largest Lyapunov exponent. As depicted in Fig. 3(b), the divergence rate of the predicted trajectory (green line) is similar to the largest Lyapunov exponent of the MG system (blue line). Furthermore, the performance that the dESN provides is robust against randomization. We find similar results (orange line) for 20 different dESNs with varied connection matrices and training data. The climate regime describes the long-term behavior of a dynamic system, e.g., the evolution along the chaotic attractor. In Fig. 3(d), we show the predicted trajectory after it diverges from the initialization trajectory. The dESN reproduces the climate of the delay system by inferring the chaotic attractor of the MG system with $\tau = 30$ as shown in Fig. 3(c) and is the case for other delays D shown in Fig. 2. Accordingly, training the dESN only on a single example is sufficient to precisely infer the dynamical properties of the MG system for various delay lengths.

As presented in Figs. 2 and 3, training the dESN with data from the MG system with a delay of $\tau = 100$ enables inferring untrained dynamics related to shorter and even longer delays. A delay of $\tau = 100$ places the system in the long delay limit where the delay is much longer than the characteristic

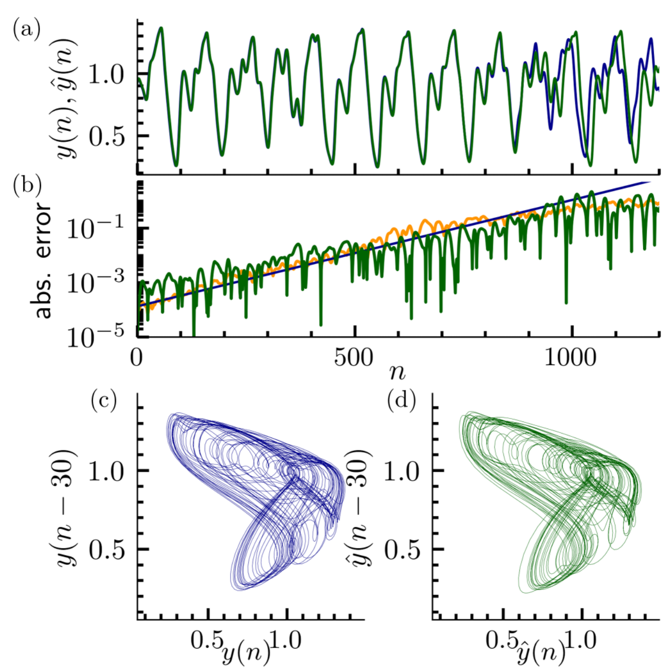

FIG. 3. (a) Time series of the original Mackey-Glass system (blue) and the autonomously continued (green) chaotic attractor using a dESN. (b) Divergence rate of the chaotic system and the time series generated by the dESN (green) as shown in (a), the orange line indicates the average divergence rate of 20 different dESNs initialized at 20 different trajectories of the chaotic Mackey-Glass system. The blue line indicates the divergence rate related to the largest Lyapunov exponent $\lambda = 0.009$ of the Mackey-Glass system with $\tau = 30$. In (c) and (d), the two-dimensional projection of the chaotic attractor of the Mackey-Glass system with a delay of $\tau = 30$ and the dESN prediction with $D = 30$, respectively.

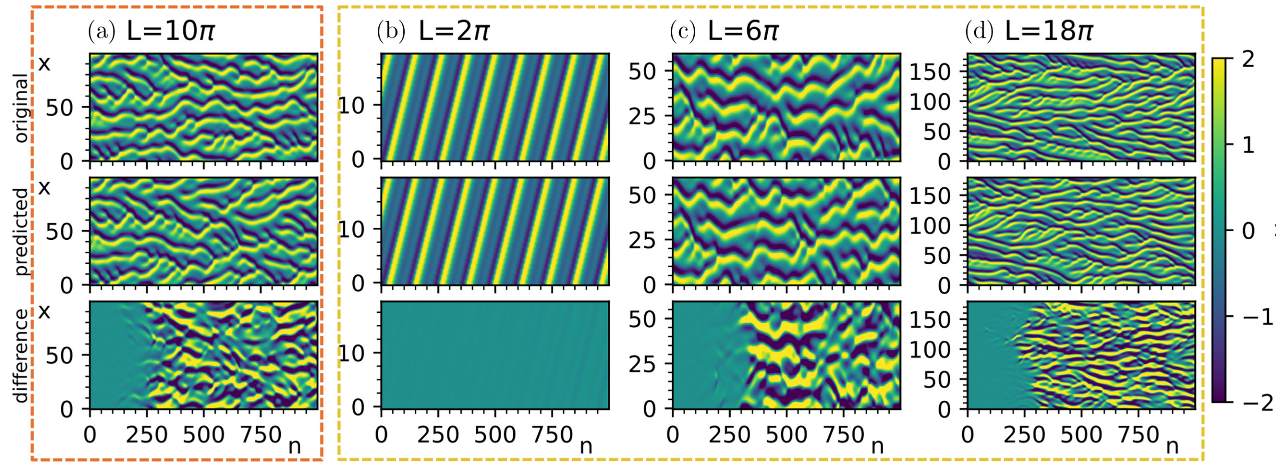

FIG. 4. The dynamics of the spatiotemporal Kuramoto-Sivashinsky system $y(x, n)$ (top row), the predicted dynamics from a parallel network architecture $\hat{y}(x, n)$ (mid row), and their difference $y(x, n) - \hat{y}(x, n)$ (bottom row). The parallel networks are trained with data from the Kuramoto-Sivashinsky system of spatial extension $L = 10\pi$ (red box), by adapting the network architecture it generates the dynamics also for smaller and larger spatial extensions (yellow box). For comparison, both systems were initialized with the same initial conditions.

relaxation time T_0 in Eq. (2). Here, learning benefits from a clear timescale separation between response and delay time. We find that learning in the short delay regime enables inferring towards relatively shorter delays, whereas going towards larger delays, the inference quality deteriorates. For a more detailed discussion, we refer to Appendix B.

III. INFERENCE OF SPATIOTEMPORAL SYSTEM DYNAMICS

Taking into account the close relationship of delay and spatiotemporal systems [19–21], we extend our approach to the spatial translational symmetry of a homogeneous KS model with periodic boundary conditions. By analogy, we design reservoirs that can infer attractors for different system sizes when trained for a single size only. The dynamics of the KS model are governed by

$$y_t = -yy_x - y_{xx} - y_{xxxx}, \tag{3}$$

where $y(x, t)$ is a scalar field. We further consider periodic boundary conditions in the interval $[0, L)$. For the training, we generate a data set using a spatial extension of $L = 10\pi$. The generated data is sampled every $\Delta t = 0.25$ in the temporal domain and contains $Q = 100$ equidistant samples in the spatial domain leading to $y(n) \in \mathbb{R}^{100}$. To predict the spatiotemporal evolution of the KS system, we construct a parallel network architecture as described by Pathak *et al.* [8] and train this architecture to perform a one-step-ahead prediction. The parallel architecture contains $G = 10$ subnetworks, each containing $K = 1000$ neurons, respectively. The evolution of the gth subnetwork can be described by the following equation:

$$x^g(n+1) = \tanh[Wx^g(n) + \gamma W_{\text{in}} u^g(n) + 0.2W_b], \tag{4}$$

where $W \in \mathbb{R}^{NN}$ is a randomly drawn adjacency matrix with spectral radius $\rho = 1.3$, $W_{\text{in}} \in \mathbb{R}^{NM}$ are randomly drawn input weights, and the input gain $\gamma = 0.001$, and $W_b \in \mathbb{R}^N$ is a random bias. The input $u^g(n)$ of the subnetworks is generated by dividing the data from the spatial domain into G sections of the same size $Q/G = 10$ (spatial size of π). Each of the G subnetworks receives the ten inputs of one particular section and additionally the three closest inputs from both neighboring sections, leading to an input dimension of $M = 16$. Due to the spatial translational symmetry of the homogeneous KS system along the spatial domain the subnetworks adjacency matrix and hyperparameters can be chosen identical. Similar approaches are mentioned in Refs. [8,44]. During training, each subnetwork is trained to perform a one-step-ahead prediction of the spatial domain section to which it corresponds, resulting in ten outputs per network computed using the output weight matrix $W_{\text{out}} \in \mathbb{R}^{10\times1000}$. Again, due to the translational symmetry the output weights sets can be shared between the subnetworks. After the training phase, we close the loop by feeding the predicted state back to the reservoir. As shown in Figs. 4(a) and 4(f), parallel reservoirs autonomously predict the weather and climate of the chaotic KS system with $L = 10\pi$.

To infer spatial extensions L that are different from the one used during training, we take further advantage of the spatial translation symmetry and similar to changing the delay in the dESN, we adapt the topology of the reservoir. Therefore, either a subnetwork is removed from the parallel architecture or a copy of a subnetwork is inserted into the architecture. By varying the number of subnetworks G in the architecture the output dimension alters in steps of ten which effectively increases the predicted spatial extensions. Due to the spatial length trained on and the used number of subnetworks, we can vary the spatial extension of the predicted system in units of π. In Fig. 4, we present the original KS dynamics (first row), the predicted dynamics of the reservoir (middle row), and the difference between both (bottom row) for different spatial extensions, respectively. Therefore, we initialized both using the same initial conditions. In analogy to the results for the MG delay-dynamical system, we observe that the once-trained reservoir is able to infer the untrained dynamics at significantly shorter and larger spatial extensions. Therefore, the parallel network architecture takes advantage of the spatial translational invariance of the homogeneous KS model with periodic boundary conditions.

IV. CONCLUSION

We demonstrated that exploiting symmetries of dynamical systems by designing neural networks obeying these symmetries improves their prediction ability and further enables far-reaching inference. We, particularly, showed this for translational symmetry in delay-dynamical and spatiotemporal systems. Transformation of the trained networks along this symmetry enabled learning from a single example to infer an entire bifurcation diagram. Thus, we obtain minimal requirements for training data and gain a single model that can infer a wide variety of dynamical behaviors. This represents a very efficient use of resources, such as the required network size, training data, and energy. Therefore, it is also a step in the direction of more sustainable machine learning. In addition, the provided method might be used to analyze real-world systems for which certain parameter settings might not be accessible. Recently, Liu and Tegmark [45] presented a machine learning method to manifest hidden symmetries in physical systems. The discovery of hidden symmetries and the use of our approach to exploit them for far-reaching inferences whereas learning from single examples could give rise to powerful predictive models for a variety of high-dimensional systems, including complex networks [9].

ACKNOWLEDGMENTS

We thank A. Röhm for helpful discussions. We acknowledge support of the Spanish State Research Agency, through the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D (Grant No. MDM-2017-0711 funded by Grant No. MCIN/AEI/10.13039/501100011033) and through the QUARESC Project (Projects No. PID2019-109094GB-C21 and No. -C22/AEI/10.13039/501100011033) and the DECAPH Project (Projects No. PID2019-111537GB-C21 and No. -C22/AEI/10.13039/501100011033). M.G. acknowledges financial support by the European Union's H2020 Excellent Science Program under the Marie Skłodowska-Curie Grant Agreement No. 860360 (POST DIGITAL). This work has been partially funded by the European Union's H2020 Excellent Science Future and Emerging Technologies program under Grant Agreement No. 899265 (ADOPD).

APPENDIX A: DELAY ESTIMATION FROM TIME SERIES USING DELAYED ECHO STATE NETWORKS

As mentioned in Sec. II, during training, we match the delay of the dESN with the delay of the Mackey-Glass system to enhance the prediction abilities of the dESN. As the Mackey-Glass time series is sampled with $\Delta t = 1$ this leads to $D = \tau$. In many real-world systems, however, the underlying delay is not known *a priori* and needs to be determined from a sampled time series of the dynamical system under study. There exist several methods to estimate delays from time series based on, e.g., autocorrelation function, delayed mutual information, local linear fitting in a low-dimensional subspace [46], and even deep learning-based methods [47]. In the following, we show how to estimate the delay that underlies a time series using the dESN given in Eq. (1). Thus, we seek for high one-step-ahead prediction accuracy of dESNs by scanning the delay D of the dESN and optimizing the reservoir hyperparameters using Bayesian optimization, respectively. In Fig. 5, we show the results of this scan for three different delays $\tau = 30, 60,$ and 100 of the Mackey-Glass system that generated the training data. The delay scan of the dESN reveals an increased performance (reduced NRMSE) when its delay is in resonance or equal to the delay $D = \tau$ of the Mackey-Glass system. As it can be seen in Fig. 5, identifying these resonances, in turn, allows to extract the delay that underlies the data. There can be relative offsets of ± 1 in the optimal estimated delay, which here is the smallest possible offset related to the sampling of the Mackey-Glass time series used here ($\Delta t = 1$). In this manner, the optimal setting of the dESN delay might depend on the frequency used for sampling the real-world system.

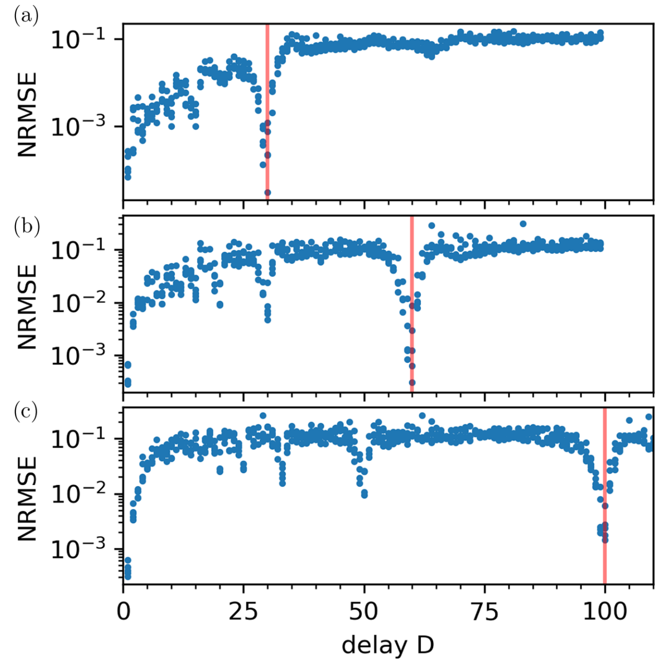

FIG. 5. Estimating the delay a from time series of the chaotic Mackey-Glass system for three different delay lengths τ. The red line marks the delay τ that underlies the data set (a) $\tau = 30$, (b) $\tau = 60$, and (c) $\tau = 100$. The blue dots mark the one-step-ahead prediction accuracy of dESNs with different delays D (x axis) and different sets of hyperparameters. The accuracy is determined by the normalized root mean square error (NRMSE) of the one-step-ahead prediction.

APPENDIX B: DEPENDENCE OF INFERENCE CAPABILITIES ON THE TRAINED CHAOTIC ATTRACTOR

In Sec. II, we show that the dESN with delay $D = 100$ trained only on the Mackey-Glass time series with $\tau = 100$ can predict the entire bifurcation diagram of the Mackey-Glass system by scaling its size (parameter D) after training.

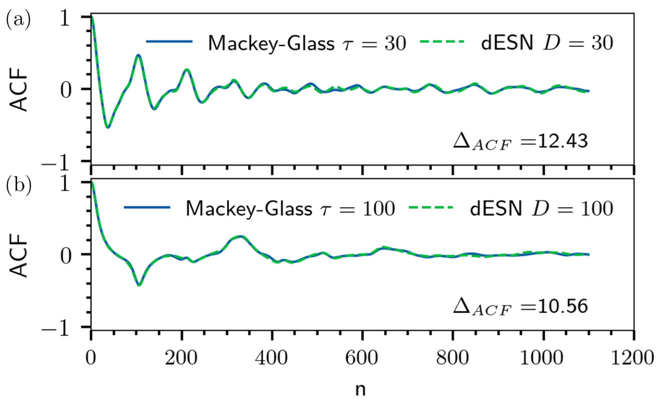

FIG. 6. Comparison of the ACF of the Mackey-Glass system (blue) with (a) $\tau = 30$ and (b) $\tau = 100$ and the predicted time series using the dESN (green). The dESN used in this plot was trained on a chaotic Mackey-Glass system with $\tau = 100$.

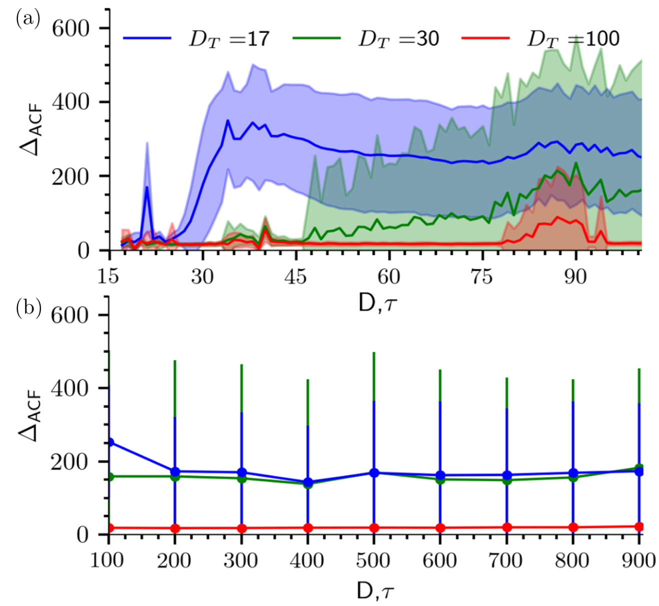

FIG. 7. Absolute difference of the original and the predicted autocorrelation Δ_{ACF} of the chaotic attractors (a) in the range $\tau,\ D \in [17, 100]$ and (b) $\tau, D \in [100, 900]$. The colors refer to different delays of the Mackey-Glass system used in the training of the dESN, $\tau = 17$ (blue), $\tau = 30$ (green), and $\tau = 100$ (red). Solid lines indicate the average over 100 random seeds used to generate the dESN and initialize the Mackey-Glass system that generated the training data. The shaded areas indicate the standard deviation, respectively.

This means that the scalable dESN is able to infer dynamics of much shorter but also longer delays whereas being trained on data of a single delay length only. In the following, we evaluate how these prediction capabilities depend on the properties of the chaotic attractor used for the training of the dESN. To do this, we use the absolute difference between the autocorrelation function, denoted Δ_{ACF}, of the Mackey-Glass attractor and the attractor predicted by the respective dESN to quantify the quality of the prediction. Here, we computed the autocorrelation function (ACF) of the original and predicted time series for 1100 steps into the past as shown in Fig. 6. Subsequently, the absolute difference Δ_{ACF} between the original and the predicted ACF is calculated and summed in the range $n \in [0, 1100]$. For comparison, we train three networks where $D_T = 17,\ 30,$ and 100 defines the delay during training, respectively, and the corresponding training data set is given by the Mackey-Glass time series with similar corresponding delay $\tau = 17,\ 30,$ and 100. In Fig. 7 we show the autocorrelation difference Δ_{ACF} for two ranges: (a) $\tau,\ D = 17$–100 and (b) $\tau,\ D = 100$–900. Whereas all trained dESN can predict towards smaller delays, the predicted attractors from the dESNs trained at $D_T = 17$ (blue) and $D_T = 30$ (green) start to deviate from the original Mackey-Glass attractor at a delay $D > 30$ and $D > 50$, respectively. In contrast, the one trained at $D = 100$ (red) does not show an increase in the difference between the predicted and the original autocorrelation function Δ_{ACF} even if the delay D of the dESN becomes much larger than the one used during training (deviations in the range $\tau,\ D = 75$–95 are caused by multistability, see Appendix C). These results illustrate that the dESN $D_T = 100$ trained at $\tau = 100$ has the highest capabilities and can predict even attractors with much longer delays.

We relate this ability of the dESN with $D_T = 100$ to the properties of the dynamics observed during training. As mentioned in Sec. II, a delay of $\tau = 100$ sets the Mackey-Glass system in the long delay limit where a clear separation of the local and the delayed dynamics appear. This can be indicated by using the autocorrelation function of the time series as shown in Fig. 6 wherein panel (a) a short delay $\tau = 30$ does not show a clear separation between delay and local dynamics. In contrast, in panel (b) of Fig. 6, one can observe this separation showing a decay of local dynamics until $n = 50$ and the delayed dynamics indicated by the peak around the delay time $n = 100$.

In conclusion, exploiting translational symmetries enables one to infer untrained dynamics whereas learning from a chaotic time series related to a single system size only. The prediction ability of the trained scalable neural network further depends on the (size-dependent) dynamical regimes on which it was trained. Thereby, the prediction ability of the dESN is the strongest if the network is trained in the long delay limit of delay-dynamical systems.

APPENDIX C: INFERRING COEXISTING LIMIT CYCLE AND CHAOTIC ATTRACTOR

As mentioned in the paper, the Mackey-Glass system exhibits a multistability leading to coexistence of a limit cycle and a chaotic attractor in the region $\tau \in [79, 95]$. In Fig. 8(a), we show an overlay of the limit cycle and the chaotic attractor for a delay of $\tau = 85$. During the computation of the bifurcation diagram, depending on their initial conditions, the trajectory of the Mackey-Glass system and the dESN converge either to one or the other of the attractors. This effect causes deviations in the computation of the autocorrelation comparison shown in Fig. 7. In Fig. 8(b), we quantify the frequency with which the Mackey-Glass system and dESN end up in the limit cycle, respectively. Therefore, we generate time series of both systems starting from 100 different initial conditions. Using the autocorrelation function of the time series generated, we discriminate if the observed attractor is either chaotic

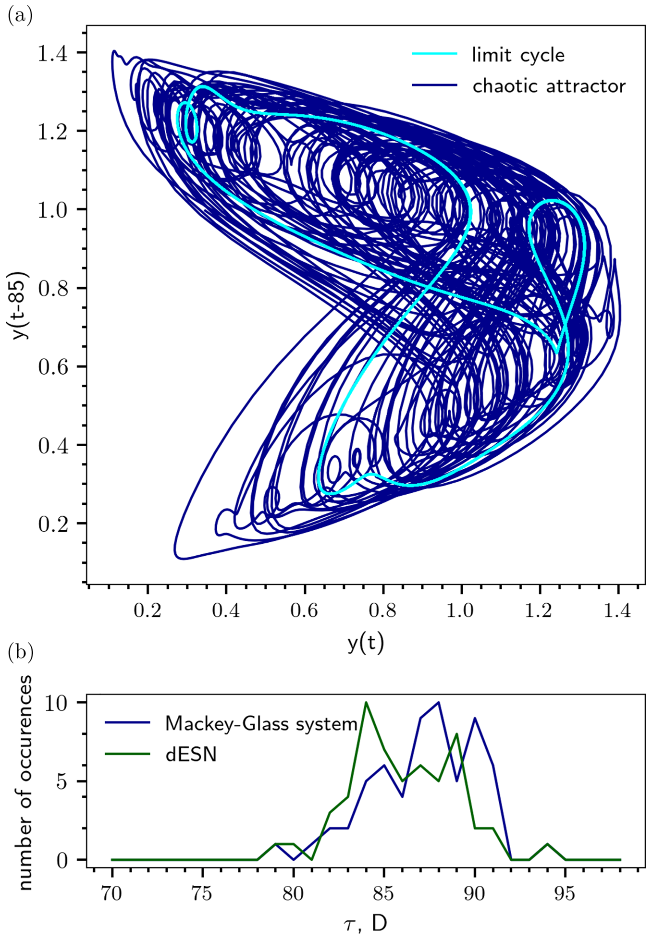

FIG. 8. (a) Two-dimensional representation of the coexisting limit cycle and chaotic attractor generated by the Mackey-Glass system at a delay of $\tau = 85$. (b) Number of occurrences of a limit cycle in the delay range $\tau,\ D \in [70, 100]$ for 100 different initial conditions. The blue colored line indicates the occurrences in differently initialized Mackey-Glass systems, whereas green indicates occurrences in the differently initialized dESN systems. The dESN is trained at $D = 100$ on a Mackey-Glass system with $\tau = 100$ and afterwards delay D was scaled down.

or a limit cycle. As shown in Fig. 8(b), both the Mackey-Glass system and the dESN exhibit comparable probabilities of ending in the limit cycle. It is worth mentioning that the dESN was trained with data of a Mackey-Glass system with $\tau = 100$ where there exists no multistability. The prediction of attractors in a multistable system that was not part of the training using *RC* was shown recently by Röhm *et al.* [43]. The fact that the dESN can predict these multistabilities even if it is trained on a Mackey-Glass system with a different delay illustrates how strong the generalization ability of the dESN is.

APPENDIX D: INFERRING IKEDA DELAY SYSTEM DYNAMICS

In the main text, we show the autonomous continuation of a Mackey-Glass delay system for short, medium, and long delays and the prediction of the bifurcation diagram for arbitrarily long delays. However, the Mackey-Glass system is only a single example of a delay dynamical system, and there exists a variety of such delay systems featuring other nonlinearities. In the following, we show that the proposed method of using a dESN to predict the bifurcation diagram is not only restricted to the Mackey-Glass system, but can be applied to other delay systems. Another well-known delay system, called the Ikeda delay system, was investigated in the field of optics because it can be generated using a Mach-Zehnder interferometer and a cavity [29]. Here, we attempt to learn and infer the dynamics of an Ikeda-type delay system with a sine-square nonlinearity,

TABLE II. Hyperparameters used to continue the time series of the Ikeda delay system.

Parameter	Symbol	Value
Network size	K	1000
Initial steps	N_{init}	1000
Training steps	N_{train}	20 000
Feedback gain	β	0.1
Input gain	γ	1.71429
Spectral radius	ρ	0.6975
Leak rate	α	0

$$\dot{s}(t) = -1/T_0 s(t) + \beta\ \sin^2[s(t-\tau)]. \quad \text{(D1)}$$

The state of the delay system is given by $s(t)$, the delay is indicated by τ, the characteristics relaxation time is fixed at $T_0 = 10$, and the feedback gain is $\beta = 0.4$ in the following. The data set is generated by integrating the delay differential equation in Eq. (D1) and the time series is sampled with $\Delta t = 1$. Due to the delay, this Ikeda system can evolve in different dynamical regimes including chaos [29,30]. In the following, we will train a dESN to continue the time series of the Ikeda delay system with a delay of $\tau = 100$. Afterward, the trained dESN will be used to infer the entire bifurcation of the Ikeda system.

As in the case of the Mackey-Glass system, the delay D of the dESN and the delay τ of the Ikeda system to be learned are matched. Afterwards, the other hyperparameters of the dESN are optimized using a Bayesian optimization approach. The found optimal parameters are given in Table II. The optimal leak rate to predict the Ikeda delay system is found at $\alpha = 0$, which means that the leak term in the equation of the dESN can be neglected.

In Fig. 9(f), we show the results of the dESN trained on an Ikeda system with $\tau = 100$ and compare it to the original attractor in Fig. 9(a). We find that the dESN properly reproduces the chaotic attractor. Similarly as for the Mackey-Glass system, we now use the dESN trained on the Ikeda system with delay $\tau = 100$ to infer the entire bifurcation diagram. In Fig. 9, we show a comparison of the original bifurcation diagram in (e) and the one predicted by the dESN in (j). The dESN is able to predict stable fixed points and limit cycles in the short delay range as well as the route to chaos and the chaotic attractors for longer delays. Furthermore, it precisely infers intermittent limit cycles in the chaotic regime for delays $\tau = 23$–25. The inferred bifurcation diagram is again very similar; we even reproduce the fine details in the bifurcation diagram. Nevertheless, we observe small deviations between

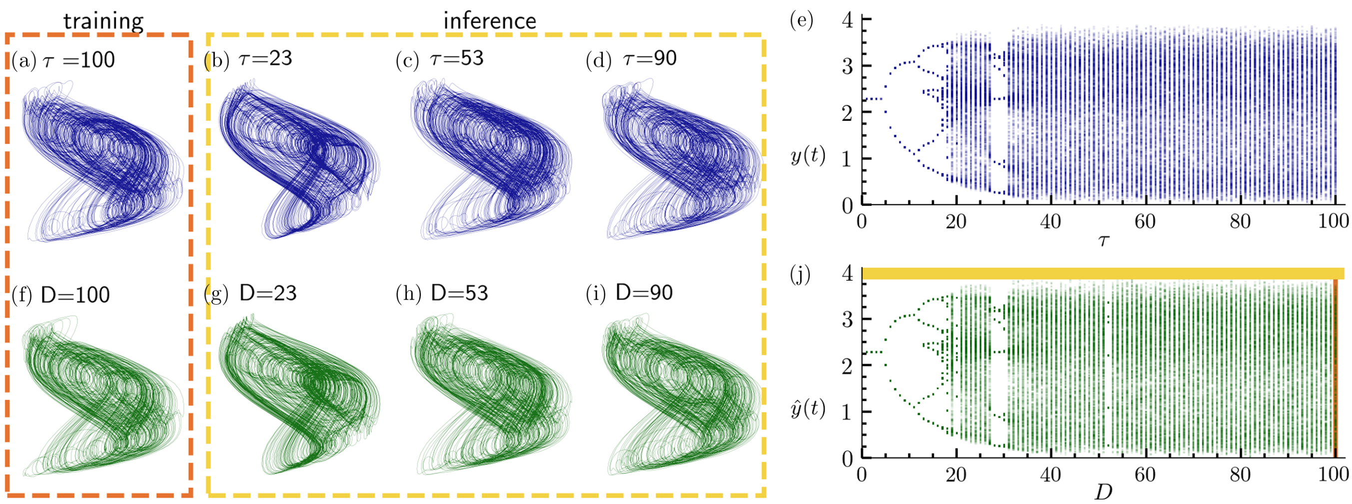

FIG. 9. Two-dimensional projection [x-axis $y(t)$, y-axis] of attractors of the chaotic Ikeda system for different delay lengths in (a)–(d). (e) Bifurcation diagram generated using the Ikeda delay system. Inferred attractors by the dESN trained on a single example of the Ikeda system with $\tau = 100$ shown in (f)–(i). (j) Bifurcation diagram inferred by the dESN trained on data of a Ikeda system with $\tau = 100$.

the inferred and the original attractor. Similarly, as for the Mackey-Glass system, these deviations are caused by the appearance of multistabilities in the range of $\tau,\ D = 50$–55. Depending on the initial conditions of the dESN and Ikeda system, respectively, they either evolve on a limit cycle or on a chaotic attractor.

[1] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[2] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report **148**, 13 (2001).

[3] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, Next generation reservoir computing, Nat. Commun. **12**, 1 (2021).

[4] H. A. Elmarakeby, J. Hwang, R. Arafeh, J. Crowdis, S. Gang, D. Liu, S. H. AlDubayan, K. Salari, S. Kregel, C. Richter *et al.*, Biologically informed deep neural network for prostate cancer discovery, Nature (London) **598**, 348 (2021).

[5] M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold *et al.*, Integrating machine learning and multiscale modeling-perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences, npj Digit. Med. **2**, 1 (2019).

[6] H. A. Dijkstra, P. Petersik, E. Hernández-García, and C. López, The application of machine learning techniques to improve el Niño prediction skill, Front. Phys., **7**153 (2019).

[7] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data, Chaos **27**, 121102 (2017).

[8] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, Phys. Rev. Lett. **120**, 024102 (2018).

[9] K. Srinivasan, N. Coble, J. Hamlin, T. Antonsen, E. Ott, and M. Girvan, Parallel Machine Learning for Forecasting the Dynamics of Complex Networks, Phys. Rev. Lett. **128**, 164101 (2022).

[10] Y. Tang, J. Kurths, W. Lin, E. Ott, and L. Kocarev, Introduction to focus issue: When machine learning meets complex systems: Networks, chaos, and nonlinear dynamics, Chaos **30**, 063151 (2020).

[11] Z. Lu, B. R. Hunt, and E. Ott, Attractor reconstruction by machine learning, Chaos **28**, 061104 (2018).

[12] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, Nat. Rev. Phys. **3**, 422 (2021).

[13] W. A. S. Barbosa, A. Griffith, G. E. Rowlands, L. C. G. Govia, G. J. Ribeill, M.-H. Nguyen, T. A. Ohki, and D. J. Gauthier, Symmetry-aware reservoir computing, Phys. Rev. E **104**, 045307 (2021).

[14] I. Higgins, S. Racanière, and D. Rezende, Symmetry-based representations for artificial and biological general intelligence, Front. Comput. Neurosci., 28 (2022).

[15] J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, Teaching recurrent neural networks to infer global temporal structure from local examples, Nat. Mach. Intell. **3**, 316 (2021).

[16] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Machine learning prediction of critical transition and system collapse, Phys. Rev. Res. **3**, 013090 (2021).

[17] L.-W. Kong, H. Fan, C. Grebogi, and Y.-C. Lai, Emergence of transient chaos and intermittency in machine learning, J. Phys. Complex. **2**, 035014 (2021).

[18] C. Klos, Y. F. K. Kossio, S. Goedeke, A. Gilra, and R.-M. Memmesheimer, Dynamical Learning of Dynamics, Phys. Rev. Lett. **125**, 088103 (2020).

[19] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.106.044211 for a discussion on the symmetries of delay-dynamical and spatiotemporal systems and

the analogies between both including two illustrative video sequences.

[20] F. T. Arecchi, G. Giacomelli, A. Lapucci, and R. Meucci, Two-dimensional representation of a delayed dynamical system, Phys. Rev. A **45**, R4225(R) (1992).

[21] G. Giacomelli and A. Politi, Relationship between Delayed and Spatially Extended Dynamical Systems, Phys. Rev. Lett. **76**, 2686 (1996).

[22] A. Keane, B. Krauskopf, and C. M. Postlethwaite, Climate models with delay differential equations, Chaos **27**, 114309 (2017).

[23] L. Bauer, J. Bassett, P. Hövel, Y. N. Kyrychko, and K. B. Blyuss, Chimera states in multi-strain epidemic models with temporary immunity, Chaos **27**, 114317 (2017).

[24] X. Shi, Y. Kuang, A. Makroglou, S. Mokshagundam, and J. Li, Oscillatory dynamics of an intravenous glucose tolerance test model with delay interval, Chaos **27**, 114324 (2017).

[25] G. Orosz, R. E. Wilson, and G. Stépán, Traffic jams: Dynamics and control, Philos. Trans. R. Soc., A **368**, 4455 (2010).

[26] G. Stepan, Delay effects in the human sensory system during balancing, Philos. Trans. R. Soc., A **367**, 1195 (2009).

[27] Q. Zhu, H. Ma, and W. Lin, Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing, Chaos **29**, 093125 (2019).

[28] L. Appeltant, M. C. Soriano, G. Van Der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, Information processing using a single dynamical node as complex system, Nat. Commun. **2**, 468 (2011).

[29] L. Larger, B. Penkovsky, and Y. Maistrenko, Virtual Chimera States for Delayed-Feedback Systems, Phys. Rev. Lett. **111**, 054103 (2013).

[30] T. Erneux, L. Larger, M. W. Lee, and J.-P. Goedgebuer, Ikeda hopf bifurcation revisited, Physica D **194**, 49 (2004).

[31] I. Fischer, O. Hess, W. Elsäßer, and E. Göbel, High-Dimensional Chaotic Dynamics of an External Cavity Semiconductor Laser, Phys. Rev. Lett. **73**, 2188 (1994).

[32] T. Erneux, *Applied Delay Differential Equations* (Springer Science & Business Media, Berlin, 2009), Vol. 3.

[33] S. Heiligenthal, T. Dahms, S. Yanchuk, T. Jüngling, V. Flunkert, I. Kanter, E. Schöll, and W. Kinzel, Strong and Weak Chaos in Nonlinear Networks with Time-Delayed Couplings, Phys. Rev. Lett. **107**, 234102 (2011).

[34] M. Le Berre, E. Ressayre, A. Tallet, and H. M. Gibbs, High-Dimension Chaotic Attractors of a Nonlinear Ring Cavity, Phys. Rev. Lett. **56**, 274 (1986).

[35] M. C. Mackey and L. Glass, Oscillation and chaos in physiological control systems, Science **197**, 287 (1977).

[36] I. Estébanez, I. Fischer, and M. C. Soriano, Constructive Role of Noise for High-Quality Replication of Chaotic Attractor Dynamics Using a Hardware-Based Reservoir Computer, Phys. Rev. Appl. **12**, 034058 (2019).

[37] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).

[38] P. Antonik, M. Gulina, J. Pauwels, and S. Massar, Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography, Phys. Rev. E **98**, 012215 (2018).

[39] A. Haluszczynski and C. Räth, Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing, Chaos **29**, 103143 (2019).

[40] W. Maass, T. Natschläger, and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. **14**, 2531 (2002).

[41] *Reservoir Computing: Theory, Physical Implementations, and Applications*, edited by K. Nakajima and I. Fischer, Natural Computing Series (Springer, Singapore, 2021).

[42] M. Lukoševičius, H. Jaeger, and B. Schrauwen, Reservoir computing trends, Künstl Intell **26**, 365 (2012).

[43] A. Röhm, D. J. Gauthier, and I. Fischer, Model-free inference of unseen attractors: Reconstructing phase space features from a single noisy trajectory using reservoir computing, Chaos **31**, 103127 (2021).

[44] W. A. W. Barbosa and D. J. Gauthier, Learning spatiotemporal chaos using next-generation reservoir computing, Chaos An Interdisciplinary Journal of Nonlinear Science **32**, 093137 (2022).

[45] Z. Liu and M. Tegmark, Machine Learning Hidden Symmetries, Phys. Rev. Lett. **128**, 180201 (2022).

[46] R. Hegger, M. J. Bünner, H. Kantz, and A. Giaquinta, Identifying and Modeling Delay Feedback Systems, Phys. Rev. Lett. **81**, 558 (1998).

[47] X. Gao, W. Zhu, Q. Yang, D. Zeng, L. Deng, Q. Chen, and M. Cheng, Time delay estimation from the time series for optical chaos systems using deep learning, Opt. Express **29**, 7904 (2021).