

RESEARCH ARTICLE | DECEMBER 26 2019

Using machine learning to assess short term causal dependence and infer network links ✓

Special Collection: [When Machine Learning Meets Complex Systems: Networks, Chaos and Nonlinear Dynamics](#)

Amitava Banerjee; Jaideep Pathak; Rajarshi Roy; ... et. al



Chaos 29, 121104 (2019)

<https://doi.org/10.1063/1.5134845>

CHORUS



CrossMark

Articles You May Be Interested In

Research on power planning under the green card and carbon trading process

AIP Conference Proceedings (December 2019)

Some aspects of the performance of refrigerating thermojunctions with radial flow of current

Journal of Applied Physics (August 2008)

AIP Advances

Why Publish With Us?

25 DAYS
average time
to 1st decision

740+ DOWNLOADS
average per article

INCLUSIVE
scope

[Learn More](#)

Using machine learning to assess short term causal dependence and infer network links

Cite as: Chaos 29, 121104 (2019); doi: 10.1063/1.5134845

Submitted: 4 November 2019 · Accepted: 5 December 2019 ·

Published Online: 26 December 2019



View Online



Export Citation



CrossMark

Amitava Banerjee,¹ Jaideep Pathak,¹ Rajarshi Roy,^{1,2} Juan C. Restrepo,³ and Edward Ott^{1,4}

AFFILIATIONS

¹Department of Physics and Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, Maryland 20742, USA

²Institute for Physical Science and Technology, University of Maryland, College Park, Maryland 20742, USA

³Department of Applied Mathematics, University of Colorado, Boulder, Colorado 80309, USA

⁴Department of Electrical and Computer Engineering, University of Maryland, College Park, Maryland 20742, USA

Note: This paper is part of the Focus Issue, “When Machine Learning Meets Complex Systems: Networks, Chaos and Nonlinear Dynamics.”

ABSTRACT

We introduce and test a general machine-learning-based technique for the inference of short term causal dependence between state variables of an unknown dynamical system from time-series measurements of its state variables. Our technique leverages the results of a machine learning process for short time prediction to achieve our goal. The basic idea is to use the machine learning to estimate the elements of the Jacobian matrix of the dynamical flow along an orbit. The type of machine learning that we employ is reservoir computing. We present numerical tests on link inference of a network of interacting dynamical nodes. It is seen that dynamical noise can greatly enhance the effectiveness of our technique, while observational noise degrades the effectiveness. We believe that the competition between these two opposing types of noise will be the key factor determining the success of causal inference in many of the most important application situations.

Published under license by AIP Publishing. <https://doi.org/10.1063/1.5134845>

The general problem of determining causal dependencies in an unknown time evolving system from time-series observations is of great interest in many fields. Examples include inferring neuronal connections from spiking data, deducing causal dependencies between genes from expression data, discovering long spatial range influences in climate variations, etc. Previous work has often tackled such problems by consideration of correlations, prediction impact, or information transfer metrics. Here, we propose a new method that leverages the potential ability of machine learning to perform predictive and interpretive tasks and uses this to extract information on causal dependence. We test our method on model complex systems consisting of networks of many interconnected dynamical units. These tests show that machine learning offers a unique and potentially highly effective approach to the general problem of causal inference.

I. INTRODUCTION

The core goal of science is often described to be a generalization from observations to understanding,¹ commonly embodied in

predictive theories. Related to this is the desire to use measured data to infer necessary properties and the structure of any description consistent with a given class of observations. On the other hand, it has recently emerged that machine learning (ML) is capable of effectively performing a wide range of interpretive and predictive tasks on data.² Thus, it is natural to ask whether machine learning might be useful for the common scientific goal of discovering structural properties of a system from data generated by this system. In this paper, we consider an important, widely applicable class of such tasks. Specifically, we consider the use of machine learning to address two goals.

Goal (i): Determine whether or not a state variable of a time evolving system causally influences another state variable.

Goal (ii): Determine the “strength” of such causal influences.

In the terminology of ML, Goal (i) is referred to as “classification ML,” and Goal (ii) is referred to as “regression ML.” These goals have previously been of great interest in many applications (e.g., economics,³ neuroscience,⁴ genomics,⁵ climate,⁶ etc.). Many past approaches have, for example, been based upon the concepts of prediction impact,^{3,4} correlation,^{7–9} information transfer,^{10,11} and direct

physical perturbations.^{12,13} Other previous works have investigated the inference of network links from time series of node states assuming some prior knowledge of the form of the network system and using that knowledge in a fitting procedure to determine links.^{9,14–17} In addition, some recent papers address network link inference from data via techniques based on delay coordinate embedding,¹⁵ random forest methods,¹⁸ network embedding algorithms,¹⁹ and feature ranking.²⁰ In this paper, we introduce a technique that makes the use of an ML training process in performing predictive and interpretive tasks and attempts to use it to extract information about causal dependencies. In particular, here, we use a particular type of machine learning (ML) called reservoir computing, an efficient method of time-series analysis, which has previously been successfully used for different tasks, e.g., prediction of chaotic dynamics^{21–23} and speech recognition,^{24,25} to mention a few. In our case, a “reservoir” dynamical system is trained such that it becomes synchronized to a training time series data set from the unknown system of interest. The trained reservoir system is then able to provide an estimation of the response to perturbations in different parts of the original system, thus yielding information about causal dependencies in the actual system. We will show that this ML-based technique offers a unique and potentially highly effective approach to determining causal dependencies. Furthermore, the presence of dynamical noise (either naturally present or intentionally injected) can very greatly improve the ability to infer causality,^{14,15} while, in contrast, observational noise degrades inference.

II. SHORT TERM CAUSAL DEPENDENCE (STCD)

We begin by considering the very general case of an evolving, deterministic, dynamical system whose state at time t is represented by the M -dimensional vector $\mathbf{z}(t) = [z_1(t), z_2(t), \dots, z_M(t)]^T$, where $\mathbf{z}(t)$ evolves via a system of M differential equations, $d\mathbf{z}(t)/dt = \mathbf{F}(\mathbf{z}(t))$, and has reached a statistically steady dynamical state (perhaps chaotic). In this context, we frame the issue of causality as follows: Will a perturbation at time t applied to a component z_i of the state vector $\mathbf{z}(t)$ [i.e., $z_i(t) \rightarrow z_i(t) + \delta z_i(t)$] lead to a subsequent change at a slightly later time, $t + \tau$, of another scalar component z_j [i.e., $z_j(t + \tau) \rightarrow z_j(t + \tau) + \delta z_j(t + \tau)$]; and how can we quantify the strength of this dependence? This formulation might suggest comparison of the evolutions of $\mathbf{z}(t)$ that result from two identical systems, one with, and the other without, the application of the perturbation. However, we will be interested in the typical situation in which such a comparison is not possible, and one can only passively observe (measure) the state $\mathbf{z}(t)$ of the (single) system of interest. Aside from knowing that the dynamics of interest evolves according to a system of the form $d\mathbf{z}/dt = \mathbf{F}(\mathbf{z})$, we assume little or no additional knowledge of the system and that the available information is a limited-duration past time series of the state evolution $\mathbf{z}(t)$. Nevertheless, we still desire to deduce causal dependencies, where the meaning of causal is in terms of responses to perturbations as defined above. Since, as we will see, accomplishment of this task, in principle, is not always possible, our approach will be to first propose a heuristic solution and then numerically test its validity. The main message of this paper is that our proposed procedure can be extremely effective for a very large class of important problems. We will also delineate situations where our procedure is expected to fail. We emphasize that,

as our method is conceptually based on consideration of responses to perturbations, in our opinion, it provides a more direct test of what is commonly of interest when determining causality than do tests based on prediction impact, correlation, or entropy metrics.

Furthermore, although the setting motivating our procedure is for deterministic systems, $d\mathbf{z}/dt = \mathbf{F}(\mathbf{z})$, we will also investigate performance of our procedure in the presence of both dynamical noise [i.e., noise added to the state evolution equation, $d\mathbf{z}/dt = \mathbf{F}(\mathbf{z})$] and observational noise [i.e., noise added to observations of $\mathbf{z}(t)$ used as training data for the machine learning]. Both types of noise are, in practice, invariably present. An important result from our study is that the presence of *dynamical* noise can very greatly enhance the accuracy and applicability of our method (a similar point has been made in Refs. 14 and 15), while observational noise degrades the ability to infer causal dependence.

To more precisely define causal dependence, we consider the effect of a perturbation on one variable on the other variables as follows. Taking the j th component of $d\mathbf{z}/dt = \mathbf{F}(\mathbf{z})$, we have

$$dz_j(t)/dt = F_j(z_1(t), \dots, z_2(t), \dots, z_M(t)),$$

for $j = 1, 2, \dots, M$. Perturbing $z_i(t)$ by $\delta z_i(t)$, we obtain, for small τ , that the component of the orbit perturbation of z_j at time $(t + \tau)$ due to δz_i is

$$\delta z_j(t + \tau) = \tau \left\{ \frac{\partial F_j(\mathbf{z})}{\partial z_i} \Big|_{\mathbf{z}=\mathbf{z}(t)} \right\} \delta z_i(t) + \mathcal{O}(\tau^2).$$

We define the *Short Term Causal Dependence* (STCD) metric, f_{ji} , of z_j on z_i by

$$f_{ji} = \left\langle G \left(\frac{\partial F_j(\mathbf{z})}{\partial z_i} \right) \right\rangle, \quad (1)$$

where $\langle (\dots) \rangle$ denotes a long time average of the quantity $(\dots)$ over an orbit and the function G is to be chosen in a situation-dependent manner. For example, later in this paper, we consider examples addressing Goal (i) [where we want to distinguish whether or not $\partial F_j(\mathbf{z})/\partial z_i$ is always zero] for which we use $G(q) = |q|$, while, when we consider an example addressing Goal (ii) and are concerned with the time-averaged signed value of the interaction strength, we then use $G(q) = q$. In either case, we view f_{ji} as quantifying the causal dependence of z_j on z_i , and the key goal of this paper will be to obtain and test a machine learning procedure for estimating f_{ji} from observations of the state evolution $\mathbf{z}(t)$. For future reference, we will henceforth denote our machine learning estimate of f_{ji} by $\hat{f}_{ji}$. In the case of our Goal (i) experiments, where $G(q) = |q|$, we note that f_{ji} defined by (1) is an average of a non-negative quantity and thus, $f_{ji} \geq 0$, as will be our estimate, $\hat{f}_{ji} \geq 0$. Furthermore, for this case, we will define STCD of z_i on z_j by the condition, $f_{ji} > 0$, and, when using our machine learning estimate $\hat{f}_{ji}$, we shall judge STCD to likely apply when $\hat{f}_{ji} > \epsilon$ where we call $\epsilon > 0$ the *discrimination threshold*. In the ideal case ($\hat{f}_{ji} = f_{ji}$), the discrimination threshold ϵ can be set to zero, but, in practice, due to the error in our estimate, we consider ϵ to be a suitably chosen positive number. We note that, in the ideal case, $\epsilon = 0$ can be regarded as a test for whether or not $F_j(\mathbf{z})$ is independent of z_i .

As a demonstration of a situation for which the determination of STCD from observations of the motion of $\mathbf{z}(t)$ on its attractor is

not possible, we note the case where the attractor is a fixed point (a zero-dimensional attractor). Here, the measured available information is the M numbers that are the coordinates of the fixed point, and this information is clearly insufficient for determining STCD. As another problematic example, we note that in certain cases, one is interested in a dynamical system that is a connected network of identical dynamical subsystems and that such a network system can exhibit exact synchronization of its component subsystems²⁶ (including cases where the subsystem orbits are chaotic). In the case where such a synchronized state is stable, observations of the individual subsystems are indistinguishable, and it is then impossible, in principle, for one to infer causal relationships between state variables belonging to different subsystems. More generally, in addition to the above fixed point and synchronization examples, we note that the dimension of the tangent space at a given point $\mathbf{z}^*$ on the attractor is, at most, the smallest embedding dimension of the part of the attractor in a small neighborhood of $\mathbf{z}^*$. Thus, the full $M \times M$ Jacobian of $\mathbf{F}(\mathbf{z})$ at $\mathbf{z}^*$ cannot be precisely determined from data on the attractor when the local attractor embedding dimension at $\mathbf{z}^*$ is less than M , which is commonly the case. Thus, these examples motivate the conjecture that to efficiently and accurately infer STCD, the orbital complexity of the dynamics should be large enough so as to encode the information that we seek. Note that these considerations of cases where inference of STCD is problematic do not apply to situations with dynamical noise, e.g., $d\mathbf{z}/dt = \mathbf{F}(\mathbf{z}) + (\text{noise})$, as the addition of noise may be roughly thought of as introducing an infinite amount of orbital complexity. Alternatively, the addition of noise increases the embedding dimension of the data to that of the full state space, i.e., M .

III. USING RESERVOIR COMPUTING TO DETERMINE STCD

We base our considerations on a type of machine learning called reservoir computing, originally put forward in Refs. 27 and 28 (for a review, see Ref. 29). We assume that we can sample the time-series data $\mathbf{z}(t)$ from our system at regular time intervals of length τ so that we have a discrete set of observations $\{\mathbf{z}(0), \mathbf{z}(\tau), \mathbf{z}(2\tau), \dots\}$. To begin, we first describe a reservoir-computer-based machine learning procedure in which the reservoir computer is trained to give an output $\hat{\mathbf{z}}(t + \tau)$ in response to an M -dimensional input $\mathbf{z}(t)$ as illustrated in Fig. 1.

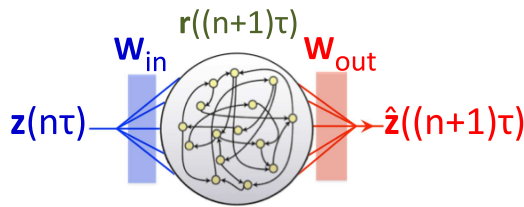


FIG. 1. Schematic of the reservoir computing architecture used in this work. The input-to-reservoir coupling matrix $\mathbf{W}_{in}$ couples the input time series for the vector $\mathbf{z}$ to the reservoir state vector $\mathbf{r}$. The reservoir-to-output coupling matrix $\mathbf{W}_{out}$ generates the output vector $\hat{\mathbf{z}}$ from the reservoir. $\hat{\mathbf{z}}$ is found to be a good estimate of $\mathbf{z}$ after training.

For our numerical tests, we consider a specific reservoir computer implementation (Fig. 1) in which the reservoir consists of a network of $R \gg M$ nodes whose scalar states, $r_1(n\tau), r_2(n\tau), \dots, r_R(n\tau)$, are the components of the R -dimensional vector $\mathbf{r}(n\tau)$.

The nodes interact dynamically with each other through an $R \times R$ network adjacency matrix $\mathbf{A}$, and their evolution is also influenced by coupling of the M -dimensional input $\mathbf{z}(n\tau)$ to the individual nodes of the reservoir network by the $M \times R$ input coupling matrix $\mathbf{W}_{in}$ according to the neural-network type of evolution equation (e.g., Refs. 21–23 and 29–31),

$$\mathbf{r}((n+1)\tau) = \tanh(\mathbf{A}\mathbf{r}(n\tau) + \mathbf{W}_{in}\mathbf{z}(n\tau)), \quad (2)$$

where $\tanh(\mathbf{v})$ for a vector $\mathbf{v} = (v_1, v_2, v_3, \dots)^T$ is defined as $(\tanh v_1, \tanh v_2, \tanh v_3, \dots)^T$. For proper operation of the reservoir computer, it is important that Eq. (2) satisfies the “echo state property”^{21,27,29} (in nonlinear dynamics, this condition is also known as “generalized synchronization”^{32–34}): given two different initial reservoir states, $\mathbf{r}_{1*}$ and $\mathbf{r}_{2*}$, for the same input time series of $\mathbf{z}$, the difference between the two corresponding reservoir states converges to zero as they evolve in time [that is, $|\mathbf{r}_1(t) - \mathbf{r}_2(t)| \rightarrow 0$ as $t \rightarrow \infty$, implying that, after a transient initial period, $\mathbf{r}(t)$ essentially depends only on the past history of $\mathbf{z}$, $\mathbf{z}(t')$ for $t' \leq t$, and not on the initial condition for $\mathbf{r}$].

Using measured input training data over a training interval of length $T\tau$, which begins after the initial transient period mentioned above, we use Eq. (2) to generate $\mathbf{r}(\tau), \mathbf{r}(2\tau), \dots, \mathbf{r}(T\tau)$. We also record and store these determined values $\mathbf{r}(n\tau)$ along with the corresponding inputs, $\mathbf{z}(n\tau)$, that created them. The matrices $\mathbf{A}$ and $\mathbf{W}_{in}$ are regarded as fixed and are typically chosen randomly. In contrast, the $R \times M$ output coupling matrix $\mathbf{W}_{out}$, shown in Fig. 1, is regarded as an adjustable linear mapping from the reservoir states $\mathbf{r}$ to an M -dimensional output vector $\hat{\mathbf{z}}$,

$$\hat{\mathbf{z}}((n+1)\tau) = \mathbf{W}_{out}\mathbf{r}((n+1)\tau). \quad (3)$$

“Training” of the machine learning reservoir computer then consists of choosing the RM adjustable matrix elements (“weights”) of $\mathbf{W}_{out}$ so as to make $\hat{\mathbf{z}}(n\tau)$ a very good approximation to $\mathbf{z}(n\tau)$ over the time duration $(\tau, 2\tau, \dots, T\tau)$ of the training data. This is done by minimization with respect to $\mathbf{W}_{out}$ of the quantity, $\left\{ \sum_{n=1}^T \|\mathbf{z}(n\tau) - \mathbf{W}_{out}\mathbf{r}(n\tau)\|^2 \right\} + \beta \|\mathbf{W}_{out}\|^2$. Here, $\beta \|\mathbf{W}_{out}\|^2$, with β small, is a “ridge” regularization term³⁵ added to prevent overfitting, and $(\mathbf{r}(n\tau), \mathbf{z}(n\tau))$ are the previously recorded and stored training data. In general, $R \gg M$ is required in order to obtain a good fit of $\hat{\mathbf{z}}$ to $\mathbf{z}(t)$. For illustrative purposes, we now consider the ideal case where $\hat{\mathbf{z}} = \mathbf{z}$ (i.e., the training perfectly achieves its goal).

For the purpose of estimating STCD, we now wish to eliminate the quantity $\mathbf{r}$ from the basic reservoir computer system [Eqs. (2) and (3)] to obtain an evolution equation solely for the state variable $\mathbf{z}$. To do this, we would like to solve (3) for $\mathbf{r}$ in terms of $\mathbf{z}$. However, since R , the dimension of $\mathbf{r}$, is much larger than M , the dimension of $\mathbf{z}$, there are typically an infinite number of solutions of (3) for $\mathbf{r}$. To proceed, we hypothesize that it may be useful to eliminate $\mathbf{r}$ by choosing it to be the solution of (3) with the smallest L_2 norm. This condition defines the so-called Moore-Penrose inverse³⁶ of $\mathbf{W}_{out}$, which we denote $\hat{\mathbf{W}}_{out}^{-1}$; i.e., the minimum L_2 norm solution for $\mathbf{r}$ is written as $\mathbf{r} = \hat{\mathbf{W}}_{out}^{-1}\mathbf{z}$. We emphasize that $\hat{\mathbf{W}}_{out}^{-1}\mathbf{z}$ is not necessarily expected to

give the correct $\mathbf{r}$ obtained by solving systems (2) and (3). However, from numerical results to follow, our choice will be supported by the fact that it often yields very useful estimates of f_{ji} .

Now, applying $\mathbf{W}_{out}$ to both sides of Eq. (2) and employing $\mathbf{r} = \hat{\mathbf{W}}_{out}^{-1} \mathbf{z}$ to eliminate $\mathbf{r}(n\tau)$ from the argument of the tanh function in Eq. (2), we obtain a surrogate time $-\tau$ map for the evolution of $\mathbf{z}$, $\mathbf{z}((n+1)\tau) = \mathbf{H}[\mathbf{z}(n\tau)]$, where $\mathbf{H}(\mathbf{z}) = \mathbf{W}_{out} \tanh[(\mathbf{A}\hat{\mathbf{W}}_{out}^{-1} + \mathbf{W}_{in})\mathbf{z}]$. Here, we note that we do not claim that this map in itself can be used for time-series prediction in place of Eqs. (2) and (3), which were commonly used in previous works (e.g., Refs. 21–23, 30, and 31). Rather, we use it as a symbolic representation of the result obtained after eliminating the reservoir state vector $\mathbf{r}$ from Eqs. (2) and (3). In particular, the prediction recipe using Eqs. (2) and (3) is always unique and well defined, in contrast to the above map, where $\hat{\mathbf{W}}_{out}^{-1}$ is clearly nonunique. Therefore, we use this map only for causality estimation purposes, as described below. Differentiating $\mathbf{H}(\mathbf{z})$ with respect to z_i , we have

$$\frac{\partial F_j(\mathbf{z})}{\partial z_i} = \tau^{-1} \left[\frac{\partial H_j(\mathbf{z})}{\partial z_i} - \delta_{ij} \right], \quad (4)$$

where δ_{ij} is the Kronecker delta, and we propose to use Eqs. (1) and (4) to determine STCD.

In our numerical experiments, the number of training time steps is $T = 6 \times 10^4$ for Figs. 2 and 3 and $T = 2 \times 10^4$ for Fig. 4. In each case, the actual training data are obtained after discarding a transient part of 2×10^4 time steps, and the reservoir system sampling time is $\tau = 0.02$. The elements of the input matrix $\mathbf{W}_{in}$ are randomly chosen in the interval $[-0.1, 0.1]$. The reservoir is a sparse random network of $R = 5000$ nodes for Figs. 2 and 3 and of $R = 1000$ nodes for Fig. 4. In each case, the average number of incoming links per node is 3. Each nonzero element of the reservoir adjacency matrix $\mathbf{A}$ is randomly chosen from the interval $[-a, a]$, and $a > 0$ is then adjusted so that the maximum magnitude eigenvalue of $\mathbf{A}$ is 0.9. The regularization parameter is $\beta = 10^{-4}$. These parameters are adapted from Ref. 23. The average indicated in Eq. (1) is over 1000 time steps. The chosen time step τ is sufficiently small compared to the time scale over which $\mathbf{z}(t)$ evolves that the discrete time series $\mathbf{z}(n\tau)$ is a good representation of the continuous variation of $\mathbf{z}(t)$.

Although we use a specific reservoir computing implementation, we expect that, with suitable modifications, our approach can be adapted to “deep” types of machine learning,² as well as to other implementations of reservoir computing^{24,25,37,38} [notably, implementations involving photonics,²⁴ electronics,³⁷ and field programmable gate arrays (FPGAs)³⁵].

IV. TESTS OF MACHINE LEARNING INFERENCE OF STCD

In order to evaluate the effectiveness of our proposed method, we introduce mathematical model test systems that we use as proxies for the unknown system of interest for whose state variables we wish to determine STCD. We next use the test systems to generate simulated training data from which we determine STCD by our ML technique. We then assess the performance of the technique by the correctness of its results determined from the known properties of the test systems.

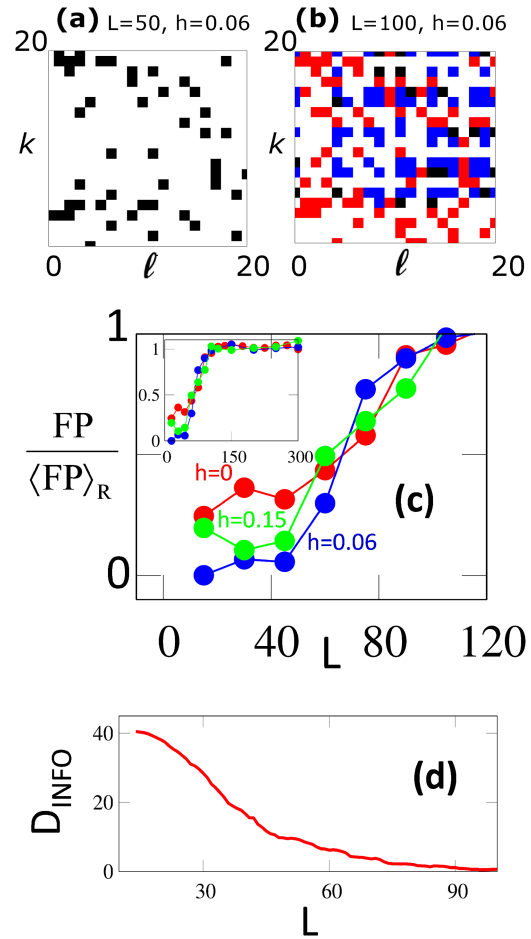


FIG. 2. Results of Experiment 1 (noiseless case). Panels (a) and (b) show the results of link inferences for two noiseless cases for $L = 50$ links and $L = 100$ links. The inference is perfect in (a) but is very bad in (b). (c) $\text{FP} / \langle \text{FP} \rangle_R$ vs L for $h = 0, 0.06$, and 0.15 averaged over 100 random realizations of the system and the reservoir adjacency matrix. (d) The orbital complexity as measured by the attractor information dimension D_{INFO} decreases with increasing L . Note that at each value of L , we compute the D_{INFO} for 10 random realizations of a network with L links with $h = 0$. The Kaplan-Yorke dimension is then averaged over all network realizations, and the resulting plot is further smoothed by applying a moving average filter.

We first consider examples addressing our Goal (i) [$G(q) = |q|$ in Eq. (1)], and for our simulation test systems, we consider the case of a network of N nodes and L links, where each node is a classical Lorenz system³⁹ with heterogeneity from node to node, additive dynamical noise, and internode coupling,

$$dx_k/dt = -10 \left[x_k - y_k + c \sum_{l=1}^N a_{kl}^{(x,y)} (y_l - y_k) \right] + \sigma_{\text{Dyn}} n_{kx}(t), \quad (5)$$

$$dy_k/dt = 28(1 + h_k)x_k - y_k - x_k z_k + \sigma_{\text{Dyn}} n_{ky}(t), \quad (6)$$

$$dz_k/dt = -(8/3)z_k + x_k y_k + \sigma_{\text{Dyn}} n_{kz}(t). \quad (7)$$

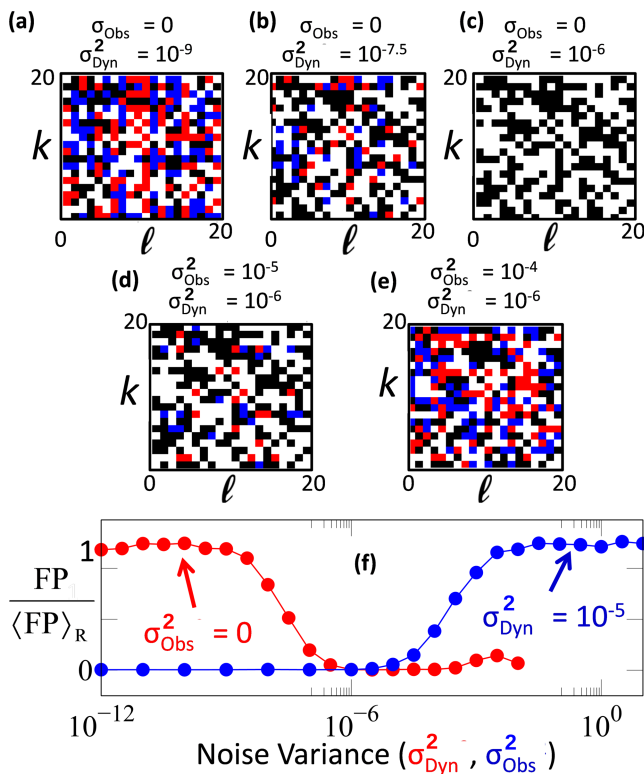


FIG. 3. The effect of noise on STCD inference. Panels (a)–(c) show the effect of increasing the dynamical noise variance σ_{Dyn}^2 to greatly enhance the effectiveness of link identification even at the rather low noise level of $\sigma_{\text{Dyn}}^2 = 10^{-6}$. In contrast, as shown in panels (d)–(f), starting with the situation (c) and increasing the observational noise variance σ_{Obs}^2 degrades link identification. $L = 200, h = 0$ for all the subfigures here.

The state space dimension of this system is $M = 3N$. The coupling of the N nodes is taken to be only from the y variable of one node to the x variable of another node with coupling constant c , and $a_{kl}^{(x,y)}$ is either 1 or 0 depending on whether or not there is a link from l to k . The adjacency matrix $a_{kl}^{(x,y)}$ of our Lorenz network (not to be confused with the adjacency matrix A of the reservoir) is constructed by placing directed links between L distinct randomly chosen node pairs. For each node k , h_k is randomly chosen in the interval $[-h, +h]$, and we call h the heterogeneity parameter. Independent white noise terms of equal variance σ_{Dyn}^2 are added to the left-hand sides of the equations for dx/dt , dy/dt , and dz/dt , where, for example, $\langle n_{kx}(t)n_{k'x}(t') \rangle = 2\delta_{kk'}\delta(t-t')$. For $\sigma = c = h = 0$, each node obeys the classical chaotic Lorenz equation with the parameter values originally studied by Lorenz.³⁹ Furthermore, denoting the right-hand side of Eq. (5) by F_{xk} , we have $\partial F_{xk}/\partial y_l = 10c$ or 0, depending on whether there is, or is not, a link from y_l to x_k .

Since in this case, the derivative $\partial F_{xk}/\partial y_l$ is time independent, $|\partial F_{xk}/\partial y_l|$ is also either $10c$ or 0, and adopting the notation $f_{kl}^{(x,y)} = |\partial F_{xk}/\partial y_l|$, we denote its machine learning estimate by our

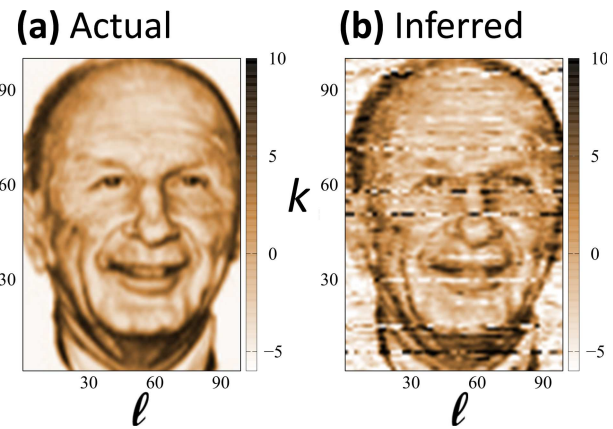


FIG. 4. Results of Experiment 3. Panel (a) shows a 100×100 pixelated, shade-coded portrait of Edward N. Lorenz and (b) reconstruction of (a) by our ML link inference technique. Note that, in (b), we plot all the values greater than or equal to 10 as black and all the values less than or equal to -5.5 as white.

previously described procedure by $\hat{f}_{kl}^{(x,y)}$. For a reasonably large network, the number $N^2 - N$ of ordered node pairs (k, l) of distinct nodes is large, and we consequently have many values of $\hat{f}_{kl}^{(x,y)}$. Bayesian techniques (see Ref. 40 and references therein) can be applied to such data to obtain an estimate $\hat{L}$ for the total number of links L , and one can then set the value of ϵ so that there are $\hat{L}$ values of $\hat{f}_{kl}^{(x,y)}$ that are greater than ϵ . Less formally, we find that making a histogram of the values of $\hat{f}_{kl}^{(x,y)}$ often reveals a peak at zero and another peak at a higher positive value with a large gap or discernible minimum in between. One can then estimate ϵ by a value in the gap or by the location of the minimum between the peaks, respectively. For simplicity, in our illustrative numerical simulations to follow, we assume that L is known [approximately equivalent to the case that L is unknown, but that a very good estimate ($\hat{L}$) has been obtained].

Example 1 (A heterogeneous noiseless case): We consider the parameter set $c = 0.3$, $h = 0.06$, $\sigma_{\text{Dyn}} = \sigma_{\text{Obs}} = 0$, $N = 20$, and we vary the number of links L . Figures 2(a) (for $L = 50$) and 2(b) (for $L = 100$) each show an array of 20×20 boxes where each of the boxes represents an ordered node pair (k, l) of the 20-node network, and the boxes have been colored (see Table I) according to whether the results for our procedure predict a link from l to k (“positive”) or not (“negative”) and whether the prediction is correct (“true”) or wrong (“false”).

We see that for a typical case with $L = 50$ [Fig. 2(a)], all the boxes have been correctly labeled, corresponding to all boxes being

TABLE I. Color-coding scheme for Figs. 2 and 3.

TP (true positive)	Black square
TN (true negative)	White square
FP (false positive)	Blue square
FN (false negative)	Red square

either black or white. In contrast to this perfect result at $L = 50$, at $L = 100$ [Fig. 2(b)], the method fails terribly, and the fraction of correct inferences is small. In fact, we find excellent performance for $L \leq 50$, but that, as L increases past 50, the performance of our method degrades markedly. This is shown in Fig. 2(c) where we give plots of the number of false positives (FPs) normalized to the expected value of FP that would result if L links were randomly assigned to the $N^2 - N = 380$ node pairs (k, l) . [We denote this normalization $\langle \text{FP} \rangle_R$; it is given by $\langle \text{FP} \rangle_R = L(380 - L)/380$.] Note that, with this normalization, for the different heterogeneities plotted in Fig. 2(c), the curves are similar and that they all begin increasing at around $L = 60$ and $\text{FP} / \langle \text{FP} \rangle_R$ becomes nearly 1 (i.e., inference no better than random) past $L \sim 100$. In our earlier discussion, we have conjectured that, for inference of STCD to be possible, the orbital complexity should not be too small. To test this conjecture, we have calculated the information dimension D_{INFO} of the network system attractor corresponding to the parameters, $c = 0.3$, $h = 0$, $\sigma = 0$, $N = 20$, as a function of L . We do this by calculating the Lyapunov exponents of the system [Eqs. (5)–(7)] and then applying the Kaplan-Yorke formula for D_{INFO} in terms of the calculated Lyapunov exponents.^{41,42} The result is shown in Fig. 2(d), where we see that D_{INFO} decreases with increasing L . Regarding D_{INFO} as a measure of the orbital complexity, this is consistent with our expectation that the ability to infer STCD will be lost if the orbital complexity of the dynamics is too small. As we next show, the above negative result for L increasing past about 60 does not apply even when small dynamical noise is present.

Example 2 (The effects of dynamical and observational noise): We first consider the effect of dynamical noise of variance σ_{Dyn}^2 for the parameters $h = 0$ (homogeneous), $c = 0.3$, $N = 20$, and $L = 200$. Results [similar in style to Figs. 2(a) and 2(b)] are shown in Figs. 3(a)–3(c). For extremely low dynamical noise variance, $\sigma_{\text{Dyn}}^2 = 10^{-9}$ [Fig. 3(a)], the result is essentially the same as for zero noise, and about one quarter of the boxes are classified as TP, TN, FP, and FN each (since there are 200 links and 400 boxes, this is no better than random assignment). As the noise variance is increased to $\sigma_{\text{Dyn}}^2 = 10^{-7.5}$ [Fig. 3(b)], the results become better, with a fraction of 0.75 of the boxes either TP or TN [as opposed to 0.52 for Fig. 3(a)]. Upon further increase of the dynamical noise variance to the still small value of $\sigma_{\text{Dyn}}^2 = 10^{-6}$ [Fig. 3(c)], the results typically become perfect or nearly perfect. Furthermore, excellent results, similar to those for $\sigma_{\text{Dyn}}^2 = 10^{-6}$, continue to apply for larger σ_{Dyn}^2 . This is shown by the red curve in Fig. 3(f), which shows $\text{FP} / \langle \text{FP} \rangle_R$ vs σ_{Dyn}^2 ($N = 20$; $L = 200$). Importantly, we also note that our normalization of FP by $\langle \text{FP} \rangle_R$ essentially makes the red curve L -independent over the range we have tested, $50 \leq L \leq 200$. Our interpretation of this dynamical-noise-mediated strong enhancement of our ability to correctly infer links is that the dynamical noise allows the orbit to explore the state space dynamics off the orbit's attractor and that the machine learning is able to make appropriate good use of the information it thus gains.

We now turn to the effect of observational noise by replacing the machine learning time-series training data formerly used, $[x_k(n\tau), y_k(n\tau), z_k(n\tau)]$, by $[x_k(n\tau) + \hat{\sigma}_{\text{Obs}} \hat{n}_{kx}(n\tau), y_k(n\tau) + \hat{\sigma}_{\text{Obs}} \hat{n}_{ky}(n\tau), z_k(n\tau) + \hat{\sigma}_{\text{Obs}} \hat{n}_{kz}(n\tau)]$, where the parameter $\hat{\sigma}_{\text{Obs}}$ is the observational noise variance and the $\hat{n}_{kx}$, $\hat{n}_{ky}$, $\hat{n}_{kz}$ are independent Gaussian random variables with, e.g., $\langle \hat{n}_{kx}(n\tau) \hat{n}_{k'x}(n'\tau) \rangle = 2\delta_{kk'} \delta_{nn'}$. The blue curve in Fig. 3(f) shows the effect of adding observational

noise of variance σ_{Obs}^2 on top of dynamical noise for the situation $\sigma_{\text{Dyn}}^2 = 10^{-5}$ of Fig. 3(c). We see from Figs. 3(d)–3(f) that, when σ_{Obs}^2 is below about 10^{-5} , it is too small to have much effect, but, as σ_{Obs}^2 is increased above 10^{-5} , the observational noise has an increasing deleterious effect on link inference. This negative effect of observational noise is to be expected, since inference of characteristics of the unknown system is necessarily based on the part of the signal that is influenced by the dynamics of the unknown system, which the observational noise tends to obscure.

Example 3 (Inferring continuous valued dependence strengths): We now wish to address Goal (ii) [for which we take $G(q) = q$ in Eq. (1)], and we, accordingly, consider the case where $f_{kl}^{(x,y)}$ for each (k, l) takes on a value in a continuous range [rather than the case of Examples 1 and 2 where $f_{kl}^{(x,y)}$ is either 10c or zero for all (k, l)]. For this purpose, we replace Eq. (5) by

$$dx_k/dt = -10(x_k - y_k) + \sum_l f_{kl}^{(x,y)} y_l \quad (8)$$

and consider Eqs. (6)–(8) as our new test system, with $h = 0.9$, $\sigma_{\text{Dyn}}^2 = \sigma_{\text{Obs}}^2 = 0$, and $N = 100$ nodes (corresponding to $100 \times 100 = 10^4$ possible connection strength values). We choose the connection strength values as follows. A photographic portrait of Edward N. Lorenz is divided up into $100 \times 100 = 10^4$ boxes, and by using a shading scale from dark (coded as +10) to light (coded as -5), Fig. 4(a) is obtained, with the shading scale given to the right of Fig. 4(b). Setting $f_{kl}^{(x,y)}$ equal to the color scale value of box (k, l) , we next numerically solve Eqs. (6)–(8). We then use this orbit as the training data for input to our ML determination of causal strength dependence, $\hat{f}_{kl}^{(x,y)}$, and employing the same shading scale, we use the thus determined values of $\hat{f}_{kl}^{(x,y)}$ to reconstruct the original portrait, as shown in Fig. 4(b). We see that, although the reproduction is not exact, the overall picture is still clearly recognizable, indicating the effectiveness of the method for Goal (ii). For a more quantitative comparison of the actual and estimated Jacobian elements, we calculate the normalized Frobenius norm of their difference matrix $f^{(x,y)} - \hat{f}^{(x,y)}$. We first apply upper and lower cutoffs equal to 10 and -5.5, respectively, to $\hat{f}^{(x,y)}$, in order to eliminate some extreme values. Then, we calculate the ratio,

$$\delta = \frac{\|f^{(x,y)} - \hat{f}^{(x,y)}\|_F}{\left\langle \left\| f^{(x,y)} - \tilde{f}^{(x,y)} \right\|_F \right\rangle}, \quad (9)$$

where $\|M\|_F = \sqrt{\text{Trace}(M^\dagger M)} = \sqrt{\sum_{ij} |M_{ij}|^2}$ is the Frobenius norm

of the matrix M . Here, $\tilde{f}^{(x,y)}$ denotes a matrix constructed by randomly permuting the elements of the matrix $f^{(x,y)}$, and the angled brackets denote an average over such random permutations. Therefore, this ratio compares the total error in the inferred Jacobian with the variation in the original matrix elements of $f^{(x,y)}$. For example, for a perfect estimation of $f^{(x,y)}$, we will have $\delta = 0$. In contrast, $\delta = 1$ means that the prediction error is equal to the average error when the elements of $f^{(x,y)}$ are randomized. For the example shown in Fig. 4, we find that δ is approximately equal to 0.37.

V. DISCUSSION

In this paper, we have formulated and tested a new, highly effective, machine-learning-based approach for inferring causal dependencies of state variables of an unknown system from time-series observations of these state variables. A key finding is that the effectiveness of our approach is greatly enhanced in the presence of sufficient dynamical noise, provided that the deleterious effect of observational noise is not too great. *The competition between the opposing effects of these two types of noise will likely be the essential key factor determining the success or failure of causality inference in many of the most important situations of interest (e.g., in neuroscience and genomics).* Much work remains to be done to more fully address the utility of our method. In particular, further numerical tests on diverse systems, and, especially, experimental studies in real world applications, will ultimately determine the circumstances under which the method developed here will be useful.

ACKNOWLEDGMENTS

This work was supported by the U.S. National Science Foundation (NSF) (Grant No. DMS 1813027). The authors acknowledge useful discussion with Sarthak Chandra, Amitabha Sen, Woodrow Shew, Nuno Martins, Adrian Papamarcou, Erik Bollt, and, especially, Brian Hunt.

REFERENCES

- ¹R. Feynman, *The Character of Physical Law* (MIT Press, Cambridge, 1965).
- ²I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, 2016).
- ³C. W. J. Granger, "Investigating causal relations by econometric methods and cross-spectral methods," *Econometrica* **37**, 424–430 (1969).
- ⁴M. Ding, Y. Chen, and S. L. Bressler, "Granger causality: Basic theory and applications to neuroscience," in *Handbook of Time Series Analysis* (Wiley-VCH, 2006), pp. 437–460.
- ⁵C. Sima, J. Hua, and S. Jung, "Inference of gene regulatory networks using time-series data: A survey," *Curr. Genomics* **10**(6), 416–429 (2009).
- ⁶J. F. Donges, Y. Zou, N. Marwan, and J. Kurths, "Complex networks in climate dynamics," *Eur. Phys. J. Spec. Top.* **174**, 157 (2009).
- ⁷W. L. Ku, G. Duggal, Y. Li, M. Girvan, and E. Ott, "Interpreting patterns of gene expression: Signatures of coregulation, the data processing inequality, and triplet motifs," *PLoS ONE* **7**, e31969 (2012).
- ⁸J. Ren, W.-X. Wang, B. Li, and Y.-C. Lai, "Noise bridges dynamical correlation and topology in coupled oscillator networks," *Phys. Rev. Lett.* **104**, 058701 (2010).
- ⁹Z. Levnajic and A. Pikovsky, "Untangling complex dynamical systems via derivative-variable correlations," *Sci. Rep.* **4**, 5030 (2014); M. G. Leguia, R. G. Andrzejak, and Z. Levnajic, "Evolutionary optimization of network reconstruction from derivative-variable correlations," *J. Phys. A Math. Theor.* **50**, 334001 (2017).
- ¹⁰T. Schreiber, "Measuring information transfer," *Phys. Rev. Lett.* **85**, 461–464 (2000).
- ¹¹J. Sun and E. M. Bollt, "Causation entropy identifies indirect influences, dominance of neighbors and anticipatory couplings," *Phys. D* **267**, 49–57 (2014).
- ¹²E. J. Molinelli, A. Korkut, W. Wang, M. L. Miller, N. P. Gauthier, X. Jing, P. Kaushik, Q. He, G. Mills, D. B. Solit, C. A. Prtilas, M. Weigt, A. Braunstein, A. Pagnani, R. Zecchina, and C. Sander, "Perturbation biology: Inferring signaling networks in cellular systems," *PLoS Comput. Biol.* **9**(12), e1003290 (2013).
- ¹³M. Timme, "Revealing network connectivity from response dynamics," *Phys. Rev. Lett.* **98**, 224101 (2007).
- ¹⁴M. J. Panaggio, M.-V. Ciocanel, L. Lazarus, C. M. Topaz, and B. Xu, "Model reconstruction from temporal data for coupled oscillator networks," *Chaos* **29**, 103116 (2019).
- ¹⁵M. G. Leguia, C. G. B. Martinez, I. Malvestio, A. T. Campo, R. Rocamora, Z. Levnajic, and R. G. Andrzejak, "Inferring directed networks using a rank-based connectivity measure," *Phys. Rev. E* **99**, 012319 (2019).
- ¹⁶T. Stankovski, T. Pereira, P. V. E. McClintock, and A. Stefanovska, "Coupling functions: Universal insights into dynamical interaction mechanisms," *Rev. Mod. Phys.* **89**, 045001 (2017).
- ¹⁷S. G. Shandilya and M. Timme, "Inferring network topology from complex dynamics," *New J. Phys.* **13**, 013004 (2011).
- ¹⁸S. Leng, Z. Xu, and H. Ma, "Reconstructing directional causal networks with random forest: Causality meeting machine learning," *Chaos* **29**, 093130 (2019).
- ¹⁹R.-M. Cao, S.-Y. Liu, and X.-K. Xua, "Network embedding for link prediction: The pitfall and improvement," *Chaos* **29**, 103102 (2019).
- ²⁰M. G. Leguia, Z. Levnajic, L. Todorovski, and B. Zenko, "Reconstructing dynamical networks via feature ranking," *Chaos* **29**, 093107 (2019).
- ²¹H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science* **304**, 78–80 (2004).
- ²²J. 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).
- ²³Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos* **27**, 041102 (2017).
- ²⁴L. Larger, A. Baylon-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Phys. Rev. X* **7**, 011015 (2017).
- ²⁵B. Schrauwen, M. D'Haene, D. Verstraeten, and J. Van Campenhout, "Compact hardware liquid state machines on FPGA for real-time speech recognition," *Neural Netw.* **21**, 511–523 (2008).
- ²⁶L. M. Pecora and T. L. Carroll, "Master stability function for synchronized chaotic systems," *Phys. Rev. Lett.* **80**, 2109 (1998).
- ²⁷H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," *GMO Report 148*, German National Research Center for Information Technology, 2001.
- ²⁸W. Maass, T. Natschlager, and H. Markham, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.* **14**, 2531–2560 (2002).
- ²⁹M. Lukoševičius and H. Jaeger, "Reservoir computer approaches to recurrent neural network training," *Comput. Sci. Rev.* **3**, 127–149 (2009).
- ³⁰P. 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).
- ³¹P. Antonik, M. Haelterman, and S. Massar, "Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems," *Phys. Rev. Appl.* **7**, 054014 (2017).
- ³²N. F. Rulkov, M. M. Sushchik, L. S. Tsimring, and H. D. T. Abarbanel, "Generalized synchronization of chaos in directionally coupled chaotic systems," *Phys. Rev. E* **51**, 980–994 (1995).
- ³³L. Kocarev and U. Parlitz, "Generalized synchronization, predictability, and equivalence of unidirectionally coupled dynamical systems," *Phys. Rev. Lett.* **76**, 1816–1819 (1996).
- ³⁴B. R. Hunt, E. Ott, and J. A. Yorke, "Differentiable generalized synchronization of chaos," *Phys. Rev. E* **55**, 4029–4034 (1997).
- ³⁵A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics* **12**, 55–67 (1970).
- ³⁶R. Penrose, "A generalized inverse for matrices," *Proc. Camb. Philos. Soc.* **51**, 406–413 (1955).
- ³⁷L. Appeltant, M. C. Soriano, G. van der Sande, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as a complex system," *Nat. Commun.* **2**, 468–473 (2013).
- ³⁸L. Gordon and J.-P. Ortega, "Reservoir computing universality with stochastic inputs," *IEEE Trans. Neural Netw. Learn. Syst.* **23**, 1–13 (2019).

³⁹E. N. Lorenz, “Deterministic nonperiodic flow,” *J. Atmos. Sci.* **20**, 130 (1963).

⁴⁰H. D. Nguyen and G. J. McLachlan, “Maximum likelihood estimation of Gaussian mixture models without matrix operations,” *Adv. Data Anal. Classif.* **9**, 371–394 (2015).

⁴¹J. L. Kaplan and J. A. Yorke, “Chaotic behavior of multidimensional difference equations,” in *Functional Differential Equations and Approximations of Fixed Points* (Springer, Heidelberg, 1979), pp. 204–227.

⁴²J. D. Farmer, E. Ott, and J. A. Yorke, “The dimension of chaotic attractors,” *Phys. D* **7**, 153–180 (1983).