

Machine Learning Link Inference of Noisy Delay-Coupled Networks with Optoelectronic Experimental Tests

Amitava Banerjee^{1,2,*}, Joseph D. Hart^{3,†}, Rajarshi Roy^{1,2,4} and Edward Ott^{1,2,5}

¹*Department of Physics, University of Maryland, College Park, Maryland 20742, USA*

²*Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, Maryland 20742, USA*

³*Optical Sciences Division, United States Naval Research Laboratory, Washington, D.C. 20375, USA*

⁴*Institute for Physical Science and Technology, University of Maryland, College Park, Maryland 20742, USA*

⁵*Department of Electrical and Computer Engineering, University of Maryland, College Park, Maryland 20742, USA*

 (Received 28 October 2020; revised 14 April 2021; accepted 19 May 2021; published 20 July 2021)

We devise a machine learning technique to solve the general problem of inferring network links that have time delays using only time series data of the network nodal states. This task has applications in many fields, e.g., from applied physics, data science, and engineering to neuroscience and biology. Our approach is to first train a type of machine learning system known as reservoir computing to mimic the dynamics of the unknown network. We then use the trained parameters of the reservoir system output layer to deduce an estimate of the unknown network structure. Our technique, by its nature, is noninvasive but is motivated by the widely used invasive network inference method, whereby the responses to active perturbations applied to the network are observed and employed to infer network links (e.g., knocking down genes to infer gene regulatory networks). We test this technique on experimental and simulated data from delay-coupled optoelectronic oscillator networks, with both identical and heterogeneous delays along the links. We show that the technique often yields very good results, particularly if the system does not exhibit synchrony. We also find that the presence of dynamical noise can strikingly enhance the accuracy and ability of our technique, especially in networks that exhibit synchrony.

DOI: [10.1103/PhysRevX.11.031014](https://doi.org/10.1103/PhysRevX.11.031014)

Subject Areas: Complex Systems,
Computational Physics,
Interdisciplinary Physics,
Nonlinear Dynamics

I. INTRODUCTION

Dynamically evolving complex networks are ubiquitous in natural and technological systems [1]. Examples include food webs [2], biochemical [3,4] and gene interaction [5,6] networks, neural networks [7], human interaction networks [8], and the Internet, to mention a few. Inference of the structure of such networks from observation of their dynamics is a key issue with applications such as

determination of the connectivity in nervous systems [9–11], mapping of interactions between genes [12] and proteins in biochemical networks [13], distinguishing relationships between species in ecological networks [14,15], understanding the causal dependencies between elements of the global climate [16], and charting of the invisible dark web of the Internet [17]. In many of these problems, we can only passively observe time series data for the states of the network nodes and cannot actively perturb the systems in any way. Network inference for these cases has led to several different computational and statistical approaches, including Granger causality [18,19], transfer entropy [20], causation entropy [21], event timing analysis [22], Bayesian techniques [12,15,23,24], inversion of response functions [25,26], random forest methods [27], and feature ranking methods [28], among others.

In this work, we are interested in the common situation of dynamics that evolves through interactions mediated by the network links along which information transfer is subject to time delay and dynamical noise. We propose

*Corresponding author.
amitava8196@gmail.com, he/him or they/them, amitavab@umd.edu, he/him or they/them

†Corresponding author.
joseph.hart@nrl.navy.mil

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

and test, both experimentally and computationally, a machine learning methodology to infer these time-delayed network interactions. For this purpose, we use only the sampled time series data of the network nodal states. We find that our method is successful in both experimental and computational tests, for a wide variety of network topologies, and for networks with either identical or heterogeneous delay times along their links, provided the time series we use contains sufficient information for the networks to be inferred.

Applications of machine learning techniques for network inference have recently begun to be explored [27,29–32]. However, all of these treat networks without link delay. For example, Ref. [32] uses machine learning but considers inference of generalized synchronization, rather than inference of links, between two systems. In particular, to our knowledge, there is no paper in which the common general situation considered in our paper is treated by machine learning, namely, link inference in delay-coupled networks with arbitrary topology and noise. Furthermore, a key feature of our work is that, in contrast with Refs. [27,29–32], we present an experimental validation with *known ground truth*. Based on the surprising success of machine learning across a wide variety of data-based tasks [33,34], we believe machine learning is a particularly promising approach to the general network inference problem that we address.

Our approach is based on the demonstrated ability of machine learning for the prediction and analysis of dynamical time series data. In particular, we shall use a specific neural network architecture called reservoir computing (RC) [35,36], which has previously been used to analyze time series data from complex and chaotic systems for such tasks as forecasting spatiotemporally chaotic evolution [37–40], determination of Lyapunov exponents and replication of chaotic attractors [41], chaotic source separation [42,43], and inference of networks (without link time delays) [29]. Reservoir computers are implemented in a variety of platforms [35,44,45], e.g., in photonic [46–48], electronic, and optoelectronic [49,50] systems. In our technique, we first use time series data from an unknown delay-coupled network to train an RC to predict the future evolution of the network. We then employ this trained network to predict how the effect of imagined applied perturbations would spread through the network, thus enabling us to deduce the network structure. This approach allows us to retain the noninvasive nature of computational tools like the transfer entropy while also retaining the conceptual advantage of invasive methods [51–53].

We test our network inference method on both simulated and experimental time series data from delay-coupled optoelectronic oscillator networks. An optoelectronic oscillator with time-delayed feedback is a dynamical system that can display a wide variety of complex behaviors, including periodic dynamics [54], breathers [55], and broadband

chaos [56]. Optoelectronic oscillators find applications in highly stable microwave generation for frequency references [54], neuromorphic computing [57,58], chaotic communications [59], and sensing [60]. The nonlinear dynamics of individual [61–63] and coupled optoelectronic oscillators [64–67] are well understood, making networks of optoelectronic oscillators an excellent test bed for network inference techniques.

We find that our method accurately reconstructs the network from experimentally measured time series data, as long as the coupling is sufficiently strong and the network does not display strong global synchronization. We also find that the presence of dynamical noise, and heterogeneity of delays along network links, may have a significant positive effect on the ability to infer links. Our results provide a clear demonstration that reservoir computing, and possibly other related machine learning methods, can provide accurate network inference for real networks, including situations where complications like noise and time delays in the coupling are present.

This paper is organized as follows. In Sec. II, we introduce our network inference method for a general delay-coupled network dynamical system. In Sec. III, we present the optoelectronic oscillator networks that we use for testing our method, along with a brief description of the collective dynamics of these networks in different parameter regimes. Section IV presents results of our tests of the effectiveness of our method for both simulated and experimental time series data. Finally, we conclude in Sec. V with further discussion, suggested future directions, and possible generalizations of our method.

II. RESERVOIR COMPUTING METHODOLOGY FOR NETWORK INFERENCE

A. The general delay-coupled network

In this section, we present the principles of our RC-based network link inference method. We consider a system of D_n nodes, with the interactions among them mediated by a network of time-delayed links. For simplicity of presentation, in this section, we restrict ourselves to networks with identical delays along different links. Later in this paper (Sec. IV C), we consider application of our framework to networks with heterogeneous link delays. For the present purpose, we assume that the state of the i th node in the network is given by a time-dependent vector $\mathbf{X}_i[t]$ of dimension D_s , with $i = 1, 2, 3, \dots, D_n$. We denote the components of this vector by X_i^μ with $\mu = 1, 2, 3, \dots, D_s$. The coupled dynamics of the full system is governed by a general delay differential equation of the form

$$\frac{dX_i^\mu(t)}{dt} = F_i^\mu[\mathbf{X}_i(t); \mathbf{X}_1(t-\tau), \mathbf{X}_2(t-\tau), \dots, \mathbf{X}_{D_n}(t-\tau); t]. \quad (1)$$

Here, F_i^μ is the function governing the dynamics of the μ th component of the state vector of the i th node. F_i^μ is a function of X_j^ν if and only if there is a network link (connection) from the ν th component of the state vector of the j th node to the μ th component of the state vector of the i th node. Note that, as previously noted, for simplicity, in the above equation, we assume that the couplings have only a single time delay τ , which is the time it takes a signal to propagate from one component of the system to another. In the experiments we consider here, the time series data from the above dynamics is sampled at a time interval $\Delta t = \tau/k$ (with k being an integer) and are denoted by $\{\mathbf{X}_i[\Delta t]\}$, $\{\mathbf{X}_i[2\Delta t]\}$, $\{\mathbf{X}_i[3\Delta t]\}$, and so on.

The problem we wish to address can be formalized as follows: *If the observed time series data $\{\mathbf{X}_i[t]\}$ is the only information from the system we have, can we infer the connections of the network assuming that the underlying dynamical equations are of the general form as in Eq. (1)?* We note that we lack any explicit knowledge of the functions $\{F_i^\mu\}$. However, except in Sec. IV C, we henceforth assume that we know the delay $\tau = k\Delta t$, which, as shown in Appendix A, can, in the case of our optoelectronic test system, be inferred from the available time series data. Furthermore, we note that the performance of our method is not strongly dependent on the accuracy with which we infer the delay time τ . For example, in our cases where we typically have $k = 34$ (corresponding to a delay time $\tau = 1.4$ ms) in the simulations and experiments, setting the inferred value of k anywhere between 34 and 37 (delay time of 1.4 and 1.5 ms, respectively) gives us essentially the same link inference results. (We choose the delay time in order to work in a regime where the dynamics for our particular experimental test network is well characterized [65–68]). While the dynamics of the network depend on the delay time, we do not expect any change in the efficacy of our link inference technique for different delay times. In those cases, we need only to change the value of k [as in Eqs. (4)–(9)] in our inference procedures. Finally, we note that, in a general situation, it might not be feasible to sample all the components of the state vectors. So, henceforth, we assume that we may sample only a subset of the components of each of the nodal state vectors $\{\mathbf{X}_i[t]\}$, which, without loss of generality, we designate as the first $D'_s (\leq D_s)$ components. We now turn to a description of RC machine learning.

B. Time series prediction with a reservoir computer

In this work, our first step is to train an RC to predict the time evolution of the node states one delay time $\tau = k\Delta t$ into the future, following which we use that training information to extract the network structure of the system. A schematic of our RC implementation [29,36,37] is shown in Fig. 1. We consider an RC network consisting of a large number of nodes D_r (such that $D_r \gg D_n \times D'_s \equiv D_X$). Each of the nodes has a time-dependent scalar state, all of which are collected in a column vector $\mathbf{R}$ of length D_r .

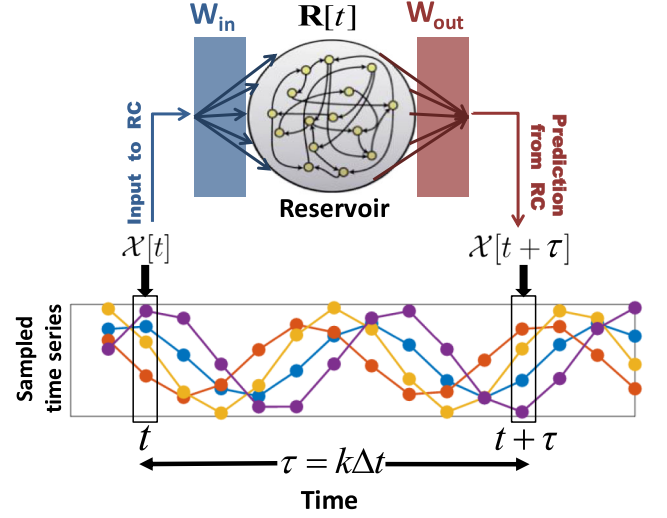


FIG. 1. Schematics of the RC trained for predicting the time series k time steps ahead. Lower: the four time series represent scalar components of $\mathbf{X}[t]$.

These reservoir nodes receive measured inputs of the unknown network system states $\{\mathbf{X}_i[t]\}$. We concatenate the sampled input measurements of the time-dependent node state vectors $\{\mathbf{X}_i[t]\}$ and place them in a single time-dependent column vector $\mathbf{X}[t]$ of length D_X , such that the components of $\mathbf{X}[t]$ are arranged as follows:

$$\mathbf{X}[t] = (X_1^1[t], X_1^2[t], \dots, X_1^{D'_s}[t], X_2^1[t], X_2^2[t], \dots, X_2^{D'_s}[t], \dots, X_{D_n}^{D'_s}[t])^T. \quad (2)$$

This vector is fed into the reservoir via a $D_r \times D_X$ input-to-reservoir coupling matrix $\mathbf{W}_{\text{in}}$ (Fig. 1). Furthermore, the reservoir nodes are coupled among themselves with a $D_r \times D_r$ adjacency matrix $\mathbf{H}$. The time evolution of the reservoir node states $\mathbf{R}$ are given by the equation

$$\mathbf{R}[n\Delta t] = \sigma\{\mathbf{H}\mathbf{R}[(n-1)\Delta t] + \mathbf{W}_{\text{in}}\mathbf{X}[n\Delta t]\}, \quad (3)$$

which maps the reservoir state at time $(n-1)\Delta t$ to the reservoir state at the next time step $n\Delta t$, where n is a positive integer and σ is a sigmoidal activation function acting componentwise on its vector argument (which has the same dimension D_r as $\mathbf{R}$).

Keeping in mind the form of Eq. (1), our first step is to predict the future values of the sampled components of $\{\mathbf{X}_i[(n+k)\Delta t]\}$ in the concatenated form $\mathbf{X}[(n+k)\Delta t]$ [Eq. (2)] from their current observed values $\mathbf{X}[n\Delta t]$ using the reservoir state vector $\mathbf{R}[n\Delta t]$. In our case, this prediction is done by using a suitable linear combination of the reservoir node states with a $D_X \times D_r$ reservoir-to-output coupling matrix $\mathbf{W}_{\text{out}}$ (Fig. 1) according to the equation

$$\mathbf{X}^P[(n+k)\Delta t] = \mathbf{W}_{\text{out}}\mathbf{R}[n\Delta t], \quad (4)$$

where the superscript P indicates that the vector is a prediction from the RC, as opposed to being sampled from the actual system. During the training time, we measure the system training time series data $\{\mathcal{X}[t]\}$ from the unknown system of interest for a large number of time steps. We use these data along with Eq. (3) to generate the time series data for the RC nodal states $\{\mathbf{R}[n\Delta t]\}$, which we store. For the training of the RC, we then find the elements of the matrix $\mathbf{W}_{\text{out}}$ by doing a linear regression from these stored reservoir states $\mathbf{R}[n\Delta t]$ to the measured time-advanced system states $\{\mathcal{X}[(n+k)\Delta t]\}$, such that Eq. (4) provides a best mean squared fit of the prediction $\mathcal{X}^P[(n+k)\Delta t]$ to the measured state $\mathcal{X}[(n+k)\Delta t]$. This fit amounts to the minimization of a cost function $\mathcal{C}$ given by

$$\mathcal{C} = \sum_{\text{training steps } n} \|\mathcal{X}[(n+k)\Delta t] - \mathbf{W}_{\text{out}}\mathbf{R}[n\Delta t]\|^2 + \lambda \|\mathbf{W}_{\text{out}}\|^2, \quad (5)$$

where the last term ($\lambda \|\mathbf{W}_{\text{out}}\|^2$) is a “ridge” regularization term [69] used to prevent overfitting to the training data and λ is typically a small number.

To completely specify the training procedure that we use, we now specify the structures of the different associated matrices. The elements of the input matrix $\mathbf{W}_{\text{in}}$ are chosen randomly from a uniform distribution in the interval $[-w, w]$. The reservoir connectivity matrix $\mathbf{H}$ is a sparse random matrix, corresponding to an average in-degree d_{av} of the reservoir nodes. The nonzero elements of $\mathbf{H}$ are chosen randomly from a uniform distribution $[-h, h]$, and h is chosen such that the spectral radius of $\mathbf{H}$ (i.e., the maximum magnitude of the eigenvalues of $\mathbf{H}$) is equal to some predefined value ρ . The hyperparameters w and d_{av} are chosen using a Nelder-Mead optimization procedure where we minimize $\mathcal{C}$ for a representative training data and the corresponding output matrix $\mathbf{W}_{\text{out}}$ found from the training data. [We use $w = 1.17$ and $d_{\text{av}} = 2.38$ for tests on simulated data and $w = 1.19$ and $d_{\text{av}} = 2.56$ for tests on experimental data. We typically use values $\lambda = 10^{-4}$ and $\rho = 0.9$ for the other two hyperparameters, 3×10^4 steps (about 880 delay times) for training, and the sigmoidal activation function σ is taken to be the hyperbolic tangent function. The reservoirs we use typically have 3000 nodes.] After a successful training of the RC with these specifications, Eq. (4) can be seen as an *in silico* model for the dynamics of the actual system. Explanations for the special properties of trained RCs, which allow us to use RCs for our purpose, can be found in Refs. [41,70–72].

C. Our network inference procedure

We now describe how we use the training results of the previous subsection to obtain the network structure of the unknown system. We first briefly discuss how the form of Eq. (1) allows us to relate the network structure to the

spread of the effect of small perturbations to the system. Suppose that, at a time point $t = n\Delta t$, we perturb the ν th component of the state of node j by an infinitesimal amount $\delta X_j^\nu[n\Delta t]$. Differentiating both sides of Eq. (1), we see that this perturbation changes the μ th component of the state of node i ($i \neq j$) at a later time $(n+k)\Delta t = t + \tau$ via the corresponding change in the time derivative:

$$\delta \left(\frac{dX_i^\mu}{dt} \right) \Big|_{t+\tau} = \frac{\partial F_i^\mu}{\partial X_j^\nu} \delta X_j^\nu[t] + \mathcal{O}\{(\delta X_j^\nu[t])^2\}. \quad (6)$$

This equation shows that, to lowest order, the effect of the small perturbation on component ν of the state of node j is propagated to the component μ of the state of node i with a delay of $\tau = k\Delta t$ provided that there is a corresponding network link between them, i.e., if $\partial F_i^\mu / \partial X_j^\nu \neq 0$. In particular, propagation of a perturbation from component ν of the state of node j to component μ of the state of node i with a delay of k time steps implies that a directional network link, $(j, \nu) \rightarrow (i, \mu)$, exists between them.

While the above discussion is predicated on application of an active perturbation, we see that the result is essentially determined by the partial derivative $\partial F_i^\mu / \partial X_j^\nu$. Thus, we wish to determine whether this partial derivative is zero (corresponding to the absence of a link) or not (corresponding to the presence of a link). We attempt to do this determination by use of the trained RC (which, as we emphasize, is obtained solely from observations, i.e., noninvasively). Indeed, when Eq. (4) is approximately true for a well-trained RC with $\mathcal{X}^P[(n+k)\Delta t] \approx \mathcal{X}[(n+k)\Delta t]$, we can use that equation to consider the RC-predicted dynamics as a proxy for the dynamics of the actual system. In that case, within this assumed RC proxy model, we can analytically assess the effects of small perturbations and compare them to Eq. (6). To do so, we first combine Eqs. (3) and (4) for the RC and assume the relation $\mathcal{X}^P[(n+k)\Delta t] = \mathcal{X}[(n+k)\Delta t]$ for the training data to obtain the equation

$$\mathcal{X}[(n+k)\Delta t] = \mathbf{W}_{\text{out}}\sigma\{\mathbf{H}\mathbf{R}[(n-1)\Delta t] + \mathbf{W}_{\text{in}}\mathcal{X}[n\Delta t]\}, \quad (7)$$

where the time points belong to the training time series. In order to evaluate the effect of a perturbation to one node on another, we desire to eliminate reservoir variables $\mathbf{R}$ from this equation. Naively, this elimination could be done by solving Eq. (4) for $\mathbf{R}[(n-1)\Delta t]$ in terms of $\mathcal{X}[(n+k-1)\Delta t]$. However, the number of components of $\mathbf{R}$ is large compared to the number of components of $\mathcal{X}$, and so there are many solutions of Eq. (4) for $\mathbf{R}$. As in our previous work [29], we hypothesize (and subsequently test) that, for our purpose, the Moore-Penrose pseudoinverse [73] (symbolically denoted by $\hat{\mathbf{W}}_{\text{out}}^{-1}$) provides a useful solution of the equation $\mathcal{X}[(n+k-1)\Delta t] = \mathbf{W}_{\text{out}}\mathbf{R}[(n-1)\Delta t]$ for $\mathbf{R}[(n-1)\Delta t]$. With this result, Eq. (7) becomes

$$\mathcal{X}[(n+k)\Delta t] = \mathbf{W}_{\text{out}} \sigma\{\mathbf{H}\hat{\mathbf{W}}_{\text{out}}^{-1} \mathcal{X}[(n+k-1)\Delta t] + \mathbf{W}_{\text{in}} \mathcal{X}[n\Delta t]\}, \quad (8)$$

yielding a putative dynamical model for the system from which we now study the effect of small perturbations. Thus, an infinitesimal amount of change in the network node states at time step $n\Delta t$, written as $\delta\mathcal{X}[n\Delta t]$, propagates to a change at time $(n+k)\Delta t$ as described by differentiating Eq. (8):

$$\delta\mathcal{X}[(n+k)\Delta t] = (\mathbb{1} - \mathbf{W}_{\text{out}}\hat{\mathbf{H}}[n\Delta t]\hat{\mathbf{W}}_{\text{out}}^{-1})^{-1} \mathbf{W}_{\text{out}}\hat{\mathbf{W}}_{\text{in}}[n\Delta t]\delta\mathcal{X}[n\Delta t] \equiv \mathbf{M}[n\Delta t]\delta\mathcal{X}[n\Delta t], \quad (9)$$

where we use Eq. (3), assume that Δt is sufficiently small that $\delta\mathcal{X}[(n+k)\Delta t] \approx \delta\mathcal{X}[(n+k-1)\Delta t]$, and define the new matrix elements $\hat{H}_{ij}[n\Delta t] = H_{ij}\sigma'\{R_i[n\Delta t]\}$ and $(\hat{W}_{\text{in}})_{ij}[n\Delta t] = (W_{\text{in}})_{ij}\sigma'\{R_i[n\Delta t]\}$, where $\sigma'\{u\} = d\sigma\{u\}/du$. We now employ this equation in component form in the place of Eq. (6) as a proxy approximating how small perturbations spread across the network. In particular, just like the partial derivative $\partial F_i^\mu/\partial X_j^\nu$ determines whether a change at the ν th component of the state of node j results in a change of the μ th component of the state of node i after k time steps in Eq. (6), $M_{i,j}[n\Delta t]$ determines the same in Eq. (9), when used in conjunction with our definition Eq. (2) for $\mathcal{X}[t]$.

We now describe how we use our determination of $\mathbf{M}$ in Eq. (9) to recover the network structure. For this purpose, based on our knowledge of $M_{i,j}[n\Delta t]$, we are interested only in determining whether $\partial F_i^\mu/\partial X_j^\nu$ is zero [no link $(j, \nu) \rightarrow (i, \mu)$] or not. In the true Jacobian, the absence of a link would imply $\partial F_i^\mu/\partial X_j^\nu = 0$ exactly. However, in our procedure, there are errors, and, thus, the elements of $\mathbf{M}[n\Delta t]$ are never zero. These errors are due to finite reservoir size, finite training data length, noise in the training data, and the Moore-Penrose inversion, which, as we hypothesize, is useful but not exact.

Generally speaking, in past link inference methods, the common approach is to somehow obtain a time-independent, continuous-valued score S_{ij} hopefully accurately reflecting the likelihood that node i is linked to j . Once the score is found, as explained below, one can then choose an appropriate statistical technique for translating the score into a good binary choice of whether or not $i \rightarrow j$ corresponds to an actual link. In essence, the key goal of our paper is to use machine learning to obtain good scores, and for that purpose we use our machine learning determination of $\mathbf{M}$.

To form an appropriate score corresponding to each of the time-dependent elements M_{ij} , we use $S_{ij} \equiv \langle |M_{ij}| \rangle_t$, where $\langle \rangle_t$ denotes time averaging over a time sufficiently long that the scores S_{ij} do not change appreciably upon, e.g., doubling the averaging time. In our tests (Sec. IV), the averaging time is $1000\Delta t$. Here, the absolute value of M_{ij} is to be taken so that the positive and negative values do not cancel each other while doing the time averaging. If this assigned score S_{ij} is above a threshold for a particular element M_{ij} , we assume that there is a network link corresponding to that element, and if the score is less than

a threshold, we assume that the corresponding network link is absent.

With the scores S_{ij} defined and calculated, choosing the threshold is a well-known problem of binary categorization of a collection of continuous numerical scores. Since obtaining a useful score for the network link inference is the goal of our machine-learning-based methodology, we regard a detailed discussion of thresholding or other follow-up statistical analysis of the obtained scores to be beyond the scope of this work. We comment only that, once the scores S_{ij} are found, one can then choose an appropriate statistical technique for translating the score into a good binary choice. This choice depends on circumstances that are specific to the situation at hand (e.g., the cost of a false-positive link choice versus the cost of a false-negative link choice). This problem is basic in statistics and addressed extensively in earlier works with methods such as receiver operating characteristic curves [74,75], precision-recall curves [74,75], fitting to mixtures of statistical distributions [74], Bayesian techniques [74], etc. A recent paper [76] proposes a binary classification technique specifically designed for network inference purposes.

To avoid the details of the statistical methodologies, we adopt a procedure which is simple but sufficient for the purpose of evaluating the goodness of the scores of candidate links resulting from our link inference method. Thus, we henceforth assume that we know the total number of links (denoted by L) in the unknown network and designate the largest L scores S_{ij} as corresponding to inferred network links, while those M_{ij} with scores below the largest L scores are inferred to not correspond to network links. The performance of this link inference technique is measured by the corresponding number of falsely inferred links (false positives). Since we assume that we know the total number of links, the number of falsely inferred links is also equal to the number of missed links (false negatives). As we show below, this method, applied to our machine-learning-determined scores, can produce excellent results in link inference tasks over a wide range of coupling strengths, network topologies, and noise levels. While, in practice, L may be unknown, and, depending on the situation, a user may wish to employ an appropriate statistical technique for making the above binary choice from our score, we claim that the results we obtain with L assumed known indicate the value of our technique for

score determination, without having to introduce and discuss more involved methods and their appropriateness in different situations (e.g., precision recall is more appropriate for sparse networks than receiver operating characteristic [75]). Later in this paper, at the end of Sec. IV A, we present results that support this claim.

III. OPTOELECTRONIC OSCILLATOR NETWORKS

A. Description of the experiment

In this section, we introduce our optoelectronic network that is used as a platform to test our network inference procedure. An individual optoelectronic oscillator is a nonlinear, time-delayed feedback loop. Our network consists of four nominally identical optoelectronic oscillators with time-delayed optical coupling between them. The individual and coupled dynamics of optoelectronic oscillators have been studied extensively [56,62,63,65–67, 77,78].

An optoelectronic oscillator essentially consists of a nonlinearity whose output is fed back into its input with some feedback time delay τ^f . This feedback delay τ^f is inherent to the oscillator, and, without it, the system would have no dynamical behavior. A description of one of our optoelectronic oscillators follows.

A fiber-coupled continuous-wave laser emits light of constant intensity. The light passes through an intensity modulator, which serves as the nonlinearity in the feedback loop and can be modeled by $\cos^2(\pi v/2V_\pi)$, where v is the voltage applied to the modulator. For our modulators, $V_\pi = 3.4$ V. The feedback optical signal is converted to an electrical signal by a photodiode, which is then delayed and filtered by a digital signal processing (DSP) board (Texas Instruments TMS320C6713). This signal is output by the DSP board, amplified, and fed back to drive the modulator. The voltage $v(t)$ applied to the intensity modulator is measured, and the normalized voltage $x(t) \equiv \pi v(t)/2V_\pi$ serves as our dynamical variable. If there were no DSP board, the delay would be controlled by the optical fiber length, and the filtering would be done by the analog electronic components, such as the amplifier. The DSP board simply provides enhanced control over the delay and filter parameters.

In general, when one couples two oscillators together, a coupling delay is induced by the finite propagation time of the signal. That coupling delay becomes important when it is not too much shorter than the fastest system timescale, which is the case for our network of coupled optoelectronic oscillators. In general, for a network of oscillators with L links, there are L different coupling delays. We use the notation τ_{ij}^c to refer to the coupling delay in the link from node j to node i . In our experiments, we choose all the coupling delays to be identical, such that $\tau_{ij}^c = \tau^c$ (heterogeneous delays are considered in Sec. IV C).

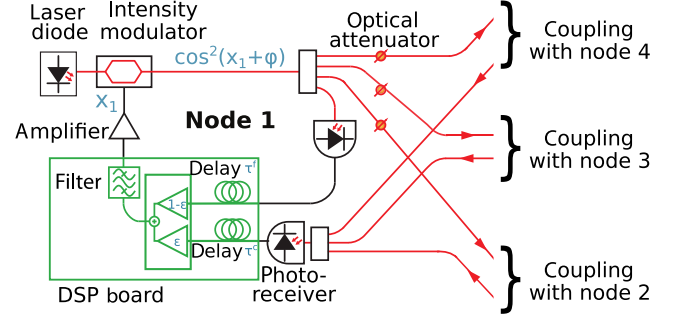


FIG. 2. Illustration of optoelectronic oscillator and coupling scheme. Red lines indicate signal paths in optical fibers. Black lines are used to indicate electronic signals, and green indicates the DSP board.

An illustration of a single networked optoelectronic oscillator is shown in Fig. 2. In order to couple the optoelectronic oscillator to other nodes in the network, a 1×4 fiber optic splitter is introduced after the intensity modulator. One of the optical outputs of the splitter is converted to an electrical signal by a photodiode, and then fed back into that node's DSP board, creating a self-feedback electrical signal. The other three splitter outputs are sent to the other three nodes in the network. Incoming optical signals from the other nodes pass through variable optical attenuators, which control the link strengths, and then are combined by a 3×1 fiber optic combiner. This combined optical signal is converted into a coupling electrical signal by a second photodiode.

The feedback and coupling signals are delayed independently in the DSP board such that τ^f and τ^c represent the feedback and coupling delay times, respectively. The outputs of the two delay lines are weighted and combined in the DSP board. This combined signal is output by the DSP, amplified, and used to drive the modulator. The ratio of amplification factors of the coupling signal and feedback signal is given by the coupling strength ϵ .

The amplifier gain is set such that each feedback loop has identical round-trip gain $\beta = 3.8$, phase bias $\phi_0 = \pi/4$, and feedback time delay $\tau^f = 1.4$ ms such that a single uncoupled node behaves chaotically. The digital filter implemented by the DSP board is a two-pole Butterworth bandpass filter with cutoff frequencies $\omega_H/2\pi = 100$ Hz and $\omega_L/2\pi = 2.5$ kHz and a sampling rate of 24 kSamples/s. These parameters are chosen because the experimental system with these parameters is well characterized [65–68].

For each set of measurements, the nodes are initialized from noise from the electrical signal into the DSP board. Then feedback is turned on without coupling, and the optoelectronic oscillators are allowed to oscillate independently until transients die out. At the end of this period, the coupling to the other nodes is turned on, and the voltage reading $x(t)$ of each optoelectronic oscillator is recorded on an oscilloscope.

B. Mathematical model and numerical simulations of the optoelectronic network

The equations governing the dynamics of our network of optoelectronic oscillators are derived in Ref. [77] and are given by

$$\begin{aligned} \frac{d\mathbf{X}_i(t)}{dt} = & \mathbf{E}\mathbf{X}_i(t) - \beta\mathbf{G}\cos^2[X_i^1(t - \tau^f) + \phi_0] \\ & - \epsilon\beta\mathbf{G}\sum_j A_{ij}\{\cos^2[X_j^1(t - \tau_{ij}^c) + \phi_0] - \cos^2[X_i^1(t - \tau^f) + \phi_0]\} + \xi_i(t), \end{aligned} \quad (10)$$

where

$$\mathbf{E} = \begin{bmatrix} -(\omega_L + \omega_H) & -\omega_L \\ \omega_H & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \omega_L \\ 0 \end{bmatrix}. \quad (11)$$

Here, $\mathbf{X}_i = [X_i^1(t), X_i^2(t)]^T$ [corresponding to $D_s = 2$ in Eq. (2)] is the state of the digital filter of node i (with $i, j \in \{1, 2, 3, 4\}$, corresponding to $D_n = 4$). By virtue of the second component of $\mathbf{G}$ being zero, coupling between nodes occurs only between X_i^1 and X_j^1 ($i \neq j$), where $X_i^1(t)$, the normalized voltage of the electrical input to the intensity modulator, is also the only observed variable [i.e., $X_i^1(t) = x(t)$, corresponding to $D'_s = 1$]. The nodes are coupled via the adjacency matrix A_{ij} , such that $A_{ij} = 1$ if there is a link to the first component of the state vector of node i from the first component of the state vector of node j and $A_{ij} = 0$ otherwise. Since the coupling is between only the first components of the vectors $\mathbf{X}_i$, we drop the component indices in the adjacency matrix in Eq. (10). The coupling strength is given by ϵ , and $\mathbf{E}$ and $\mathbf{G}$ are matrices that describe the filter. Finally, $\xi_i(t) = [\xi_i^1(t), \xi_i^2(t)]^T$ is a vector corresponding to white noise acting independently at each oscillator, and its components have the property that $\langle \xi_i^\mu[s] \xi_j^\nu[t] \rangle = 2\zeta\delta(s - t)\delta_{ij}\delta_{\mu\nu}$ with ζ denoting the strength of the noise.

In our experiments, we choose all the feedback delays and coupling delays to be nominally equal (i.e., $\tau^f = \tau_{ij}^c = \tau$). In this case, Eq. (10) describes a network with Laplacian coupling:

$$\begin{aligned} \frac{d\mathbf{X}_i(t)}{dt} = & \mathbf{E}\mathbf{X}_i(t) - \beta\mathbf{G}\cos^2[X_i^1(t - \tau) + \phi_0] \\ & - \epsilon\beta\mathbf{G}\sum_j L_{ij}\cos^2[X_j^1(t - \tau) + \phi_0] + \xi_i(t). \end{aligned} \quad (12)$$

In this case, the nodes are coupled via the Laplacian connectivity matrix L_{ij} , defined so that $L_{ij} = 1$ if there is a link to the first component of the state vector of node i from the first component of the state vector of node j , $L_{ij} = 0$ if there is no such link, and $L_{ii} = -\sum_{j \neq i} L_{ij}$. Laplacian coupling tends to lead to global synchronization, which is a

particularly challenging case for link inference, as we show in the following sections.

Since the coupling is between only the first components of the vectors $\mathbf{X}_i$, we drop the component indices in the Laplacian adjacency matrix in Eq. (10). Comparison with Eq. (1) shows that, in our example, $\mathbf{F}_i = \mathbf{E}\mathbf{X}_i(t) - \beta\mathbf{G}\cos^2[X_i^1(t - \tau) + \phi_0] - \epsilon\beta\mathbf{G}\sum_j L_{ij}\cos^2[X_j^1(t - \tau) + \phi_0] + \xi_i(t)$, where we drop the component superscripts. The oscillators are identical, so these functions are independent of i , except for the noise term. The relevant partial derivative controlling the propagation of perturbation is $\partial F_i^\mu / \partial X_j^\nu \propto L_{ij}\delta_{\mu 1}\delta_{\nu 1}$, for $i \neq j$.

While Eq. (10) accurately describes the behavior of our network of optoelectronic oscillators, numerical simulations are inherently discrete in time. Instead of discretizing Eq. (10) directly, our simulations use a discrete-time model based on the discrete-time filter equations implemented on the DSP board, which can be found in Ref. [79] and is explained in Appendix B. In particular, for this case, we characterize the noise strength by the variable κ , so that $\kappa = \zeta\Delta t$. For the discrete equation that we simulate, the time step is 0.04 ms, which corresponds to the 2.4×10^4 samples per second sampling rate used by the digital filter in our experiment.

Our model is verified by comparison with the experiment, as shown in Fig. 3 for two sets of examples. The upper panel shows experimentally measured (left) and simulated (right) time series of the four optoelectronic oscillators arranged in the six-link network shown. The dynamics are synchronized and are also the dynamics of an individual uncoupled optoelectronic oscillator, since the effect of the Laplacian coupling vanishes for global synchronization. The lower panel shows a measurement (left) and simulation (right) of the four optoelectronic oscillators coupled in the nine-link network shown. In this case, the optoelectronic oscillators do not synchronize even though the coupling is strong ($\epsilon = 0.6$). In both cases, the simulations are in good agreement with the experiment.

As we see, the degree of synchronization of the oscillators in the network is an important factor in the success of our method to infer the network topology. In order to quantify the degree of global synchrony, we define synchronization error as

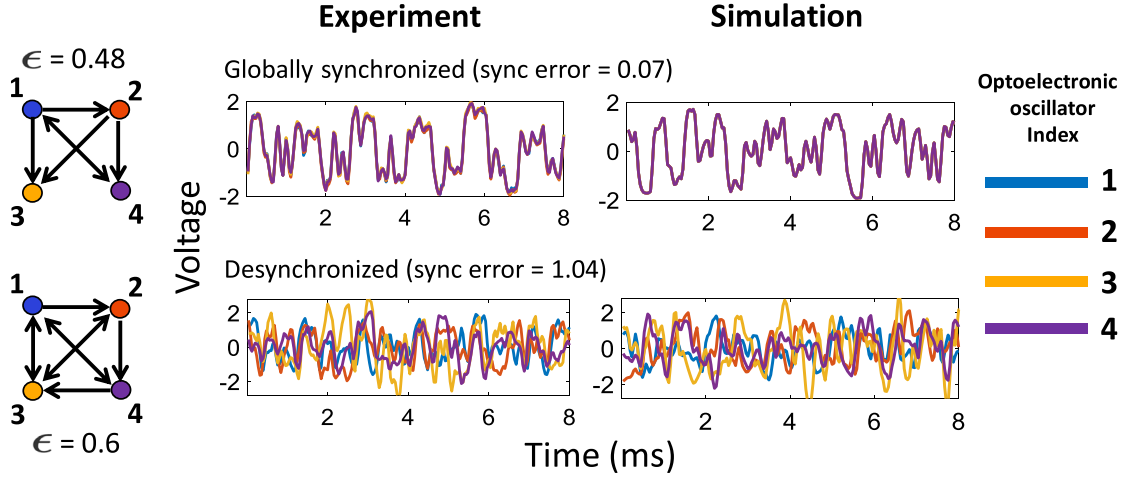


FIG. 3. Examples of experimental and simulated time series from two optoelectronic oscillator networks showing globally synchronous (upper) and completely desynchronized (lower) behavior, respectively. For the simulations, we use the noise strength $\kappa = 10^{-6}$. In the plots, purple overlays orange, which overlays red, which overlays blue.

$$\text{synchronization error} = \frac{1}{D_n(D_n - 1)} \left\langle \sum_{i,j} |x_i(t) - x_j(t)| \right\rangle_t, \quad (13)$$

where $\langle \rangle_t$ means time average over a sufficiently long time. This non-negative measure decreases with the amount of synchronization in the system and is zero for perfect global synchrony. For example, in Fig. 3, the synchronized examples (upper) have synchronization error of approximately 0.07, whereas the desynchronized examples (lower) have synchronization error of approximately 1.04.

L	Possible connected network topologies
12	
11	
10	
9	
8	
7	
6	
5	
4	
3	

FIG. 4. List of possible connected directed four-node networks [65] with different numbers of links (L).

Using computer simulations, we study the dependence of the synchronization error on the network coupling strength ϵ and the number L of network links for all possible directed and connected networks with four optoelectronic oscillator nodes. The list of the 62 possible networks is shown in Fig. 4 and is adapted from Ref. [65]. Figure 5 shows the synchronization behavior of these networks as a function of the coupling strength ϵ for fixed $\beta = 3.8$ and $\phi_0 = \pi/4$. In Fig. 5, the color-coded synchronization error for each of the 62 networks in Fig. 4 is shown as one of the

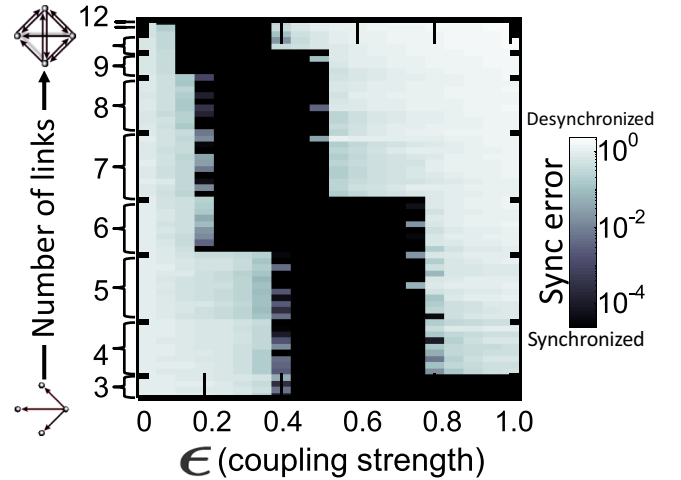


FIG. 5. Synchronization error for simulated time series of the networks in Fig. 4 realized with optoelectronic oscillator nodes with different coupling strengths for random initial conditions. The color-coded synchronization error for each of the 62 networks in Fig. 4 is shown as one of the 62 horizontal bars for each value of the coupling strength. Here, the convention we follow is that, for a fixed number of links (L), moving upward, the horizontal bars correspond to the networks listed in Fig. 4 left to right. The same convention is followed in Figs. 6, 7, and 10.

62 horizontal bars for each value of the coupling strength. Here, the convention we follow is that, for a fixed number of links (L), moving upward, the horizontal bars correspond to the networks listed in Fig. 4 left to right. The same convention is followed in Figs. 6, 7, and 10. The results in Fig. 5 are obtained from numerical simulations without noise ($\kappa = 0$). We see that, for intermediate coupling strengths, the networks synchronize, but for small and large coupling strengths, the networks do not synchronize. The seemingly counterintuitive behavior that large coupling strengths lead to desynchronization has been studied for our network of optoelectronic oscillators [68] and is characteristic of delay-coupled systems in general [80]. Furthermore, for coupling strengths in the range $\epsilon > 0.5$, for a fixed value of coupling strength, sparser networks are seen to synchronize more readily than densely connected networks. This behavior has also been studied and explained in earlier works [68,81].

IV. RESULTS OF LINK INFERENCE TESTS

In this section, we present tests of the efficacy of our machine learning technique. These include numerical simulation tests for simulations with homogeneous link delays (Sec. IV A), optoelectronic experimental tests with nominally homogeneous link delays (Sec. IV B), and numerical simulation tests with inhomogeneous link delays (Sec. IV C).

A. Performance on simulated data—Homogeneous delays

1. When the number of links is known

In this section, we test our methodology on simulated time series for our coupled optoelectronic oscillator networks where links have identical delays. We use these simulation tests to study the effects of noise and coupling strength on the amount of synchrony in the system and their effect on the performance of link inference tasks. In particular, in Sec. III B, we show that our optoelectronic oscillator networks show synchronized dynamics for certain ranges of the coupling strength ϵ . As we now show, our method works excellently when the system dynamics does not show pronounced global synchrony, while it does not work well when there is pronounced global synchrony. Furthermore, we show that our technique gives excellent results when there is a small amount of dynamical noise present so that the global synchrony among the optoelectronic oscillators is appreciably broken.

In order to directly demonstrate the effect of loss of synchrony on link inference, we vary the noise level and coupling strength in the following two sets of examples. In the first example set, the system starts with a random initial condition with no noise for 5×10^4 time steps (about 1470 delay times) and is allowed to settle down to an attractor. Then, we continue the simulation but with the noise

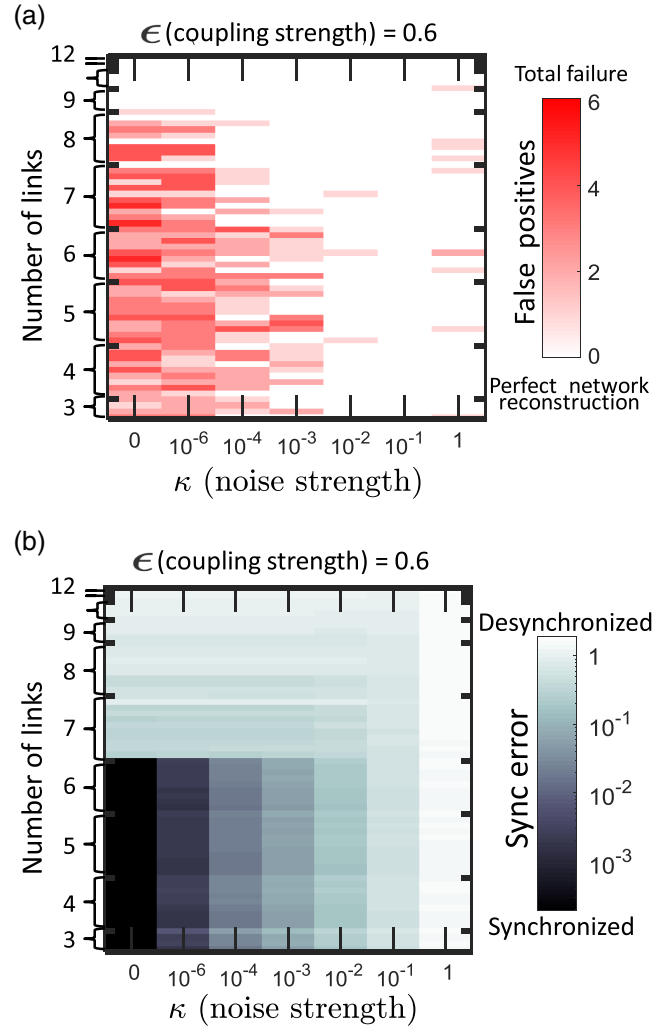


FIG. 6. Simulation test results with varying noise strength κ . (a) Number of false positives and (b) synchronization error for simulated time series from different networks with progressively increasing noise. As described in the text, each horizontal cut of the plots represents a single trajectory of the system, starting from a random initial condition. The convention for the sequence of the networks is the same as in Fig. 5.

strength κ set to 10^{-6} for the next 5×10^4 steps, then with the noise strength set to $\kappa = 10^{-4}$ for the next 5×10^4 steps, and so on, keeping the coupling strength fixed at $\epsilon = 0.6$ [Figs. 6(a) and 6(b)]. As shown in Fig. 6(b), as the noise strength increases, it drives the system away from the attractor and disrupts its global synchrony, resulting in larger synchronization error, which allows better link inference performance, as shown in Fig. 6(a). In these figures, for each of the time series segments with a fixed noise strength, we use the first 3×10^4 time steps (about 880 delay times) to train our RC and infer links using our procedure described in Sec. II. We repeat this process for each of the 62 possible connected networks of four optoelectronic oscillators [65], each one with a different random initial condition.

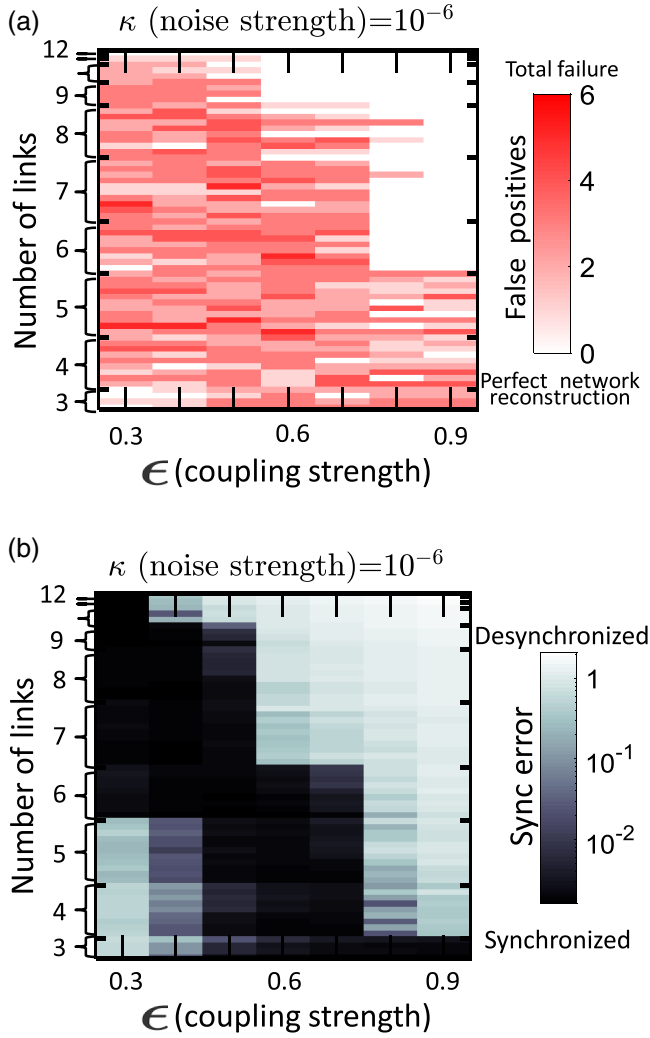


FIG. 7. Simulation test results with varying coupling strength ϵ . (a) Number of false positives and (b) synchronization error for simulated time series from different networks with progressively increasing coupling strength. As described in the text, each horizontal cut of the plots represents a single trajectory of the system, starting from a random initial condition. The convention for sequence of the networks is the same as in Fig. 5.

We use the same procedure in the next set of examples [Figs. 7(a) and 7(b)], but this time we keep the noise level fixed at its nominal experimental value of $\kappa = 10^{-6}$ and vary the coupling strength ϵ stepwise. The results are summarized in Fig. 7. For both Figs. 6 and 7, we simultaneously plot the number of false positives and synchronization error and follow the same convention as in Fig. 5.

As we see from Fig. 6, a greater degree of global synchronization generally corresponds to a larger number of false positives, consistent with the hypothesis that global synchrony is detrimental to the performance of link inference. This result is expected, because exact synchronization makes the time series from the four optoelectronic oscillators indistinguishable, and, hence, the observed dynamics yields no information about their underlying

causal interactions. In particular, we see that networks with eight or more links do not synchronize sufficiently even in the absence of the noise, and we are indeed able to infer the links well, with, at most, only one false positive. In contrast, for networks with smaller numbers of links, which are strongly synchronized for noise levels $\kappa \lesssim 10^{-3}$, we have many false-positive link inferences. Again, as we see from Fig. 6, all the networks show a loss of global synchrony for sufficiently strong noise levels ($\kappa \gtrsim 10^{-2}$), and this loss results in almost perfect link inference until the noise strength becomes significant compared to the noiseless optoelectronic oscillator signal amplitudes ($\kappa \gtrsim 10^0$). Other examples of a similar beneficial role of noise in link inference can be found at earlier works as well, e.g., in Refs. [25,29,82–86]. Of these, Ref. [86] describes a network inference technique based solely on noise correlations. However, Ref. [86] proposes a technique applicable only in the case of Laplacian coupling, unlike our work, which does not employ this model-specific restriction. Furthermore, unlike our method, their method does not work in the absence of dynamical noise and is not validated using experimental data.

In the second set of examples [Figs. 7(a) and 7(b)], we fix the noise at a particular strength $\kappa = 10^{-6}$ and progressively increase the coupling strength ϵ . We estimate that this noise level approximates that for the experimental tests reported in the next subsection (Sec. IV B). As in the previous examples, Fig. 7 shows that our link inference method performs well when the global synchrony is not too strong. A difference in this set of results from the previous ones is the nontrivial relationship between the coupling strength ϵ and global synchrony, which we discuss in the last section (Fig. 5). Furthermore, we notice that, even in the absence of global synchrony, the coupling strength needs to exceed a minimum value (about 0.1) for successful link inference. For smaller coupling, the off-diagonal elements of the matrix $\mathbf{M}[n\Delta t]$ could be so small in magnitude that the values corresponding to actual links are of the same order as those corresponding to absent links. Thus, sufficiently large coupling strength ϵ is beneficial for our link inference technique because of the better contrast among the elements of $\mathbf{M}[n\Delta t]$ and diminished global synchrony.

2. When the number of links is unknown

Finally, to show the effectiveness of our procedure in situations where the number of links L is unknown, in Fig. 8, we plot the distribution of the scores S_{ij} calculated from the matrix $\mathbf{M}[n\Delta t]$, for all the 62 networks listed in Fig. 4 with coupling strength $\epsilon = 0.6$ and noise strength $\kappa = 10^{-3}$, where, in determining the statistics, for each of the networks, we use a single random realization of initial condition and reservoir couplings. Figure 8 also shows the numerical values and properties of the scores we typically get in our method. In Fig. 8(a), we label individual scores into two types [those corresponding to actual network links

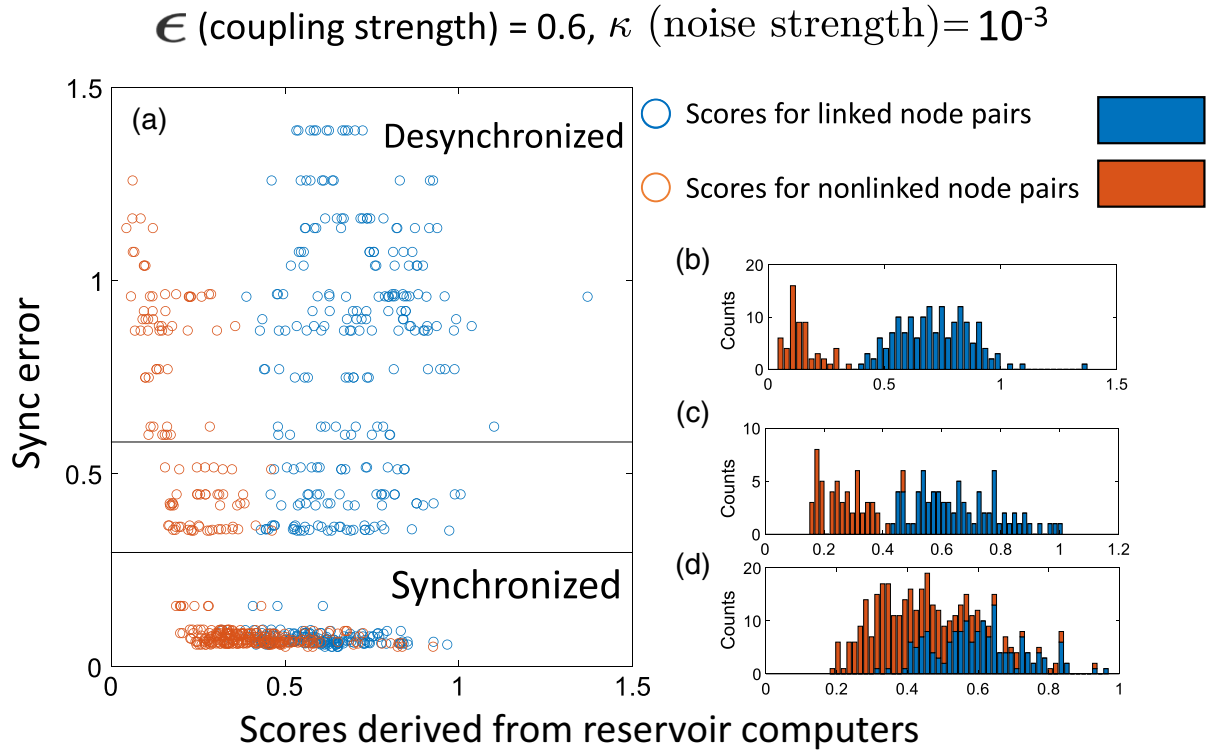


FIG. 8. This figure shows statistics of our results for the link scores $S_{i,j}$ of all the possible networks (listed in Fig. 4) with $\epsilon = 0.6$ and $\kappa = 10^{-3}$, where, in determining the statistics, for each of the networks, we use a single random realization of the initial condition and reservoir couplings. In (a), for each individual directed node pair (i, j) , i.e., each candidate link, a point is plotted in the sync-error or score plane with true links plotted in blue and link absences denoted in red, with red overlaying blue. The two black horizontal lines divide the sync-error or score plane into three regions corresponding to networks that are highly desynchronized, moderately synchronized, and strongly synchronized. (b)–(d) show histograms of the node scores for each of the three levels of network synchronization demarcated in (a). Bins with scores that all correspond to true links (absence of links) are colored blue (red). Bins with scores corresponding to both true links and link absence are vertically stacked into upper and lower pieces, where the lower piece (blue) corresponds to the number of true links in the bin and the upper piece (red) corresponds to the number of missing links.

(colored blue) and those corresponding to absence of links (colored red)] and plotted the scores with the respective synchronization errors in the network. We see that, in the cases with complete desynchronization, where we obtain perfect network inference with known L [as seen in Fig. 6(a)], it is also easy to predict a binary decision threshold on our scores [the top histogram, Fig. 8(b)]. These scores separate into two distinct populations according to their magnitudes, with a gap in between them [Fig. 8(b)], and the populations correspond to actual links or absence of links. In the cases for which Fig. 6(a) shows a finite but small number of false positives with known L , the two populations overlap, but there are still two distinct histogram peaks [the middle histogram, Fig. 8(c)], so that one can expect good inference results in cases where L is not known by choosing a suitable threshold based on the shape of the histogram. However, when the histogram overlap is so great that two peaks are not discernible, we expect that the ability to infer links no longer exists [the bottom histogram, Fig. 8(d)]. In Fig. 6(a), this scenario corresponds to cases with a large number of false positives,

even with a known L , because the network exhibits global synchrony.

B. Performance on experimental data

Having established the usefulness of our network inference method on simulated time series, we now report our experimental tests on the optoelectronic oscillator networks described in Sec. IV A. In Fig. 9, we show some representative examples of the performance of our method on experimental time series. Each column in the figure corresponds to a time series from a distinct network indicated above the column, with the respective global synchronization error indicated on the horizontal axis. The height of the columns gives the total number of links in the corresponding network. The columns are each divided into three parts (colored in red, green, and black in the plot). The height of the red portion indicates the number of falsely inferred links [“false positives: (FP)”. This portion is absent in the many cases where we have perfect network inference. The number of correctly inferred links [“true positives” (TP)] is indicated by the total height of the green plus black

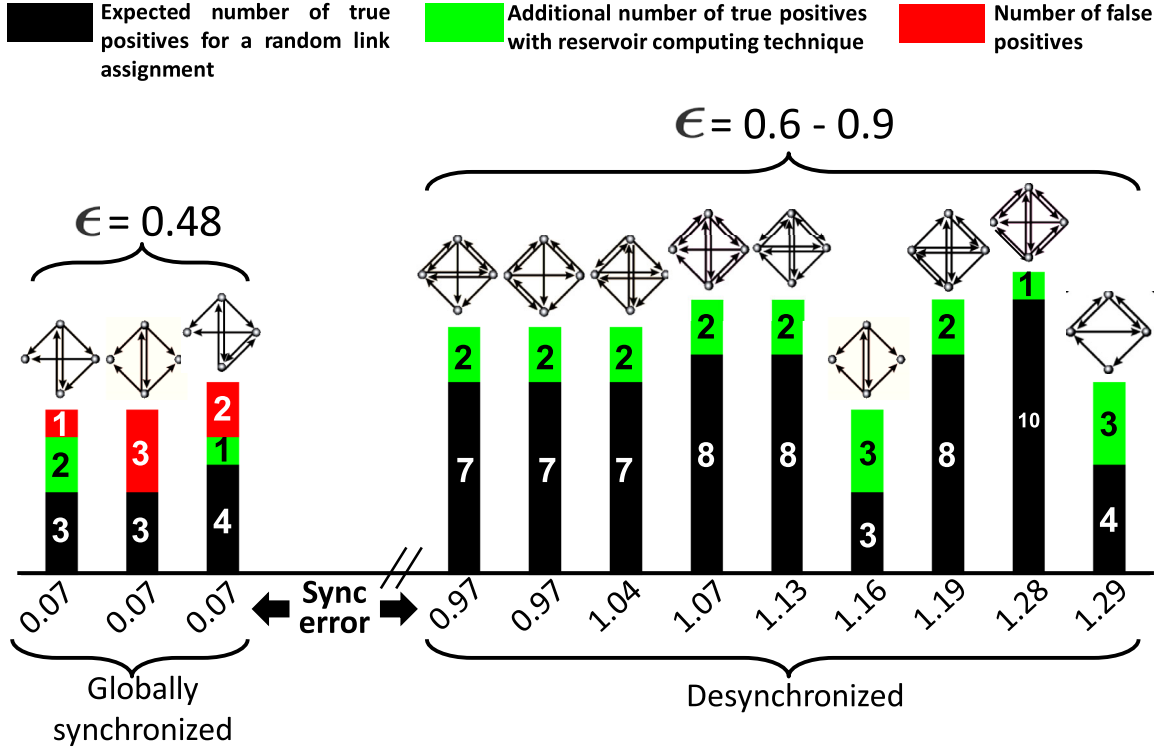


FIG. 9. Typical performance of our network inference method on experimental data. Each of the vertical bars corresponds to the experimental realization of a distinct optoelectronic oscillator network, with the height indicating the corresponding number of links. Each bar is separated into three segments as shown, but in many of them only two are seen if there is a perfect link inference. The numbers inside the segments indicate the corresponding heights of the segments, rounded to the nearest integers. Ranges of coupling strength (ϵ) and synchronization error for all the examples are indicated as well.

portions of a column. The height of the black portion indicates the expected number of true positives on average (to nearest integer) that would be obtained if all L links were to be guessed randomly, while the height of the green portion indicates the increase of true positives over what would result from random selection. To evaluate the expected number of randomly selected true positives, we note that, for L links randomly and uniformly assigned among the $D_n(D_n - 1)$ ordered pairs of D_n nodes, the expected number of false-positive links is $L\{1 - [L/D_n(D_n - 1)]\}$, and the expected number of true-positive links is, thus, $L^2/D_n(D_n - 1)$. If our method yields more true positives than $L^2/D_n(D_n - 1)$, then we consider our method to be successful, even if it gives some false positives.

To summarize our experimental results, consistent with the simulation results in Figs. 5 and 7, the time series from the experimental optoelectronic oscillator networks (Fig. 9) are either globally well synchronized or else are strongly desynchronized, and, when strong desynchronization applies, our method correctly identifies all of the links.

C. Performance on simulated data—Heterogeneous delays

So far, we consider cases where the link time delays τ_{ij}^c in Eq. (1) are the same along all links (i.e., $\tau_{ij}^c = \tau$). We now

present results on simulated data for which the link delays τ_{ij}^c are chosen randomly from a uniform distribution of mean τ_0 and width $2\Delta\tau$, where the mean link delay τ_0 is assumed known. We then apply our previously described method treating all links as if they have the same delay τ_0 , and assess the results as a function of the link delay heterogeneity, as characterized by the fractional spread $2\Delta\tau/\tau_0$ of the link delays. We do this process for all networks listed in Fig. 4. For purposes of discussion, we divide these networks into two categories: (i) networks for which we obtain perfect inference with homogeneous delays and (ii) networks for which synchronization hinders link inference performance with homogeneous delays [Fig. 6(a)]. For a fixed mean delay time τ_0 (corresponding to $k = 34$ with $\tau_0 = k\Delta t$), the results for different amounts of spread of the delay times $\Delta\tau$ are plotted in Fig. 10. The results indicate that, in case (i), we continue obtaining good results, with the average of maximum number of false positives around 1, if the heterogeneity of the spread in delays is not too large [Fig. 10(a)]. In case (ii), we obtain better, and, in some cases, perfect (i.e., zero false positives), results with moderate delay heterogeneity. This improvement can be attributed to a change in network dynamics: The heterogeneity of the link delays inhibits global synchronization, as is evident from the corresponding

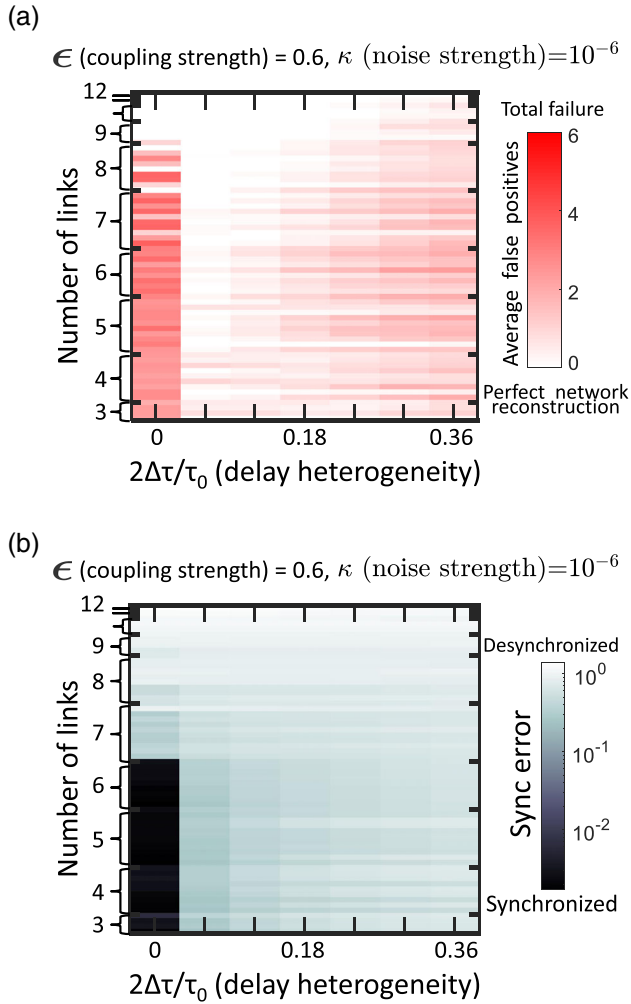


FIG. 10. Tests on networks with heterogeneous link delays. (a) Average number of false positives and (b) the corresponding synchronization error for different networks listed in Fig. 4, with $\epsilon = 0.6$ and $\kappa = 10^{-6}$, and different amounts of link delay heterogeneity, presented as the ratio of link delay variation range $2\Delta\tau$ and mean link delay τ_0 . The results are based on simulated data with different network configurations and different random link delays along the network links. In each case, the averaging is done over 100 different random realizations of reservoir connections, initial conditions, and assignments of interaction delays to different links. The convention for sequence of the networks is the same as in Fig. 5.

synchronization error plot in Fig. 10(b). Thus, moderate delay heterogeneity can be beneficial to the working of our method. This study confirms that our formalism can be applied to realistic networks with distributed link delay time, broadening the scope of our methodology. Further investigations in which we include the delay heterogeneity in the reservoir computer formalism itself are reserved for the future.

V. DISCUSSION

In this work, we develop a reservoir-computer-based technique for the general problem of link inference of noisy

delay-coupled networks from their nodal time series data and demonstrate the success of our method on simulated and experimental data from optoelectronic oscillator networks with identical and distributed link delay times. Our main findings are as follows.

- (i) Testing on experimental and simulated time-series datasets from networks, we find that, in the absence of dynamical noise, our method yields extremely good results, as long as there is no synchrony in the system.
- (ii) We find that dynamical noise and/or a moderate amount of link time delay heterogeneity can greatly enhance the performance of our method when synchrony is present, provided that the noise amplitude or link time delay heterogeneity is large enough to perturb the synchrony.
- (iii) Since dynamical noise is ubiquitous in natural and experimental situations, we anticipate that this technique may be useful in network inference tasks relevant to fields like biochemistry, neuroscience, ecology, and economics.

Among the important issues for future investigation, our work in this paper could be extended to cases when the dynamics of the network nodal states are partially synchronized (e.g., cluster synchronization of nodes [87]) or display generalized synchronization [88–91]. Effects of network symmetries [87,92], nonuniform coupling [93], and nonidentical nodes [94]—all of which can affect the synchrony of nodal states—would also be very interesting to study. Another important issue that awaits study is the effects of incomplete [95] or erroneous nodal state data [96,97] on link inference; e.g., a case of particular interest is that where measured nodal time series is available only from a subset of $N' < N$ of the N network nodes, and the value of N itself is unknown.

ACKNOWLEDGMENTS

The authors thank Thomas E. Murphy for his efforts to enable the experiments to be performed at the University of Maryland, College Park. This work was supported by the U.S. National Science Foundation Grant No. DMS 1813027 (A.B. and E.O.) and the Office of Naval Research Grant No. N000142012139 (R.R.).

APPENDIX A: DETERMINATION OF THE TIME DELAY FROM CROSS-CORRELATION

In this appendix, we demonstrate that the duration of the delay along a link in our network can be accurately estimated from the cross-correlation between the measured time series of the two nodes connected by that link. In particular, we show that the location of the peak of the cross-correlation between the two nodes provides a good estimate of the delay time. We also show that the cross-correlation cannot determine causality, because it cannot

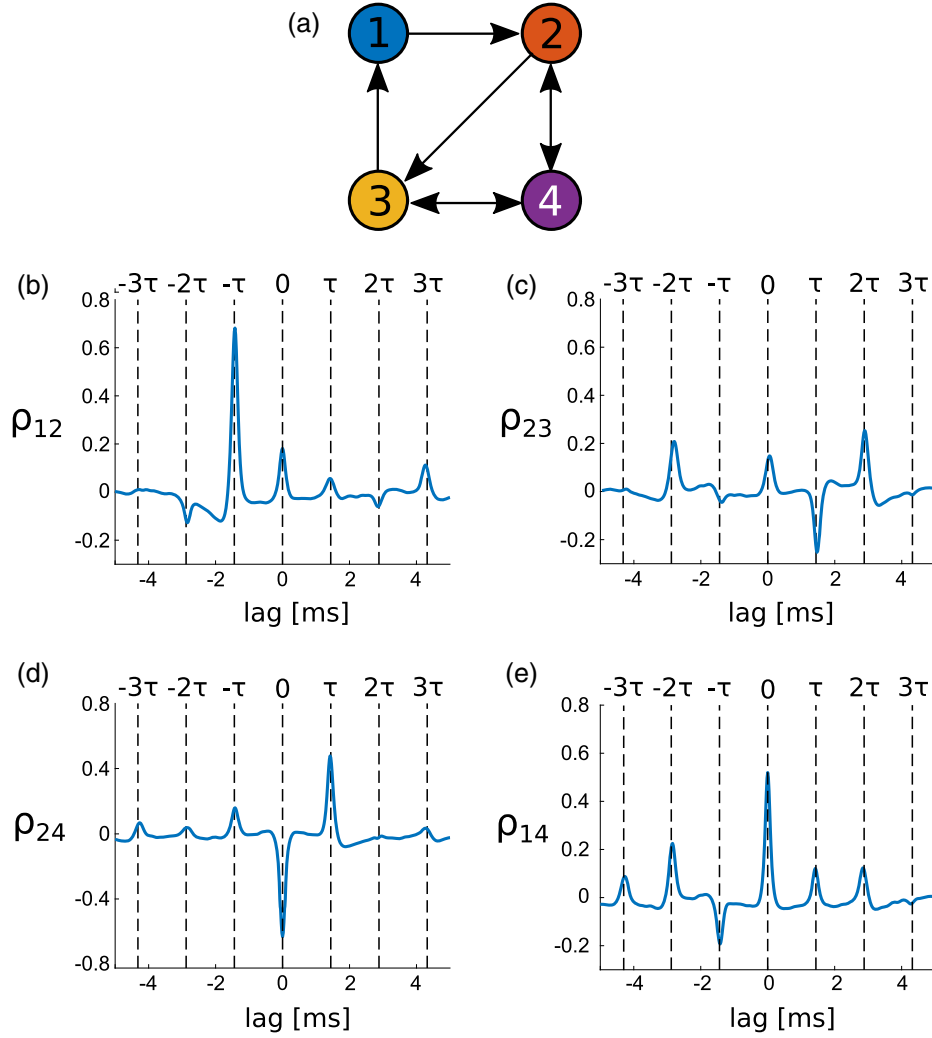


FIG. 11. Time delay determination by cross-correlation of measured time series. In all cases, the coupling delay time is 1.44 ms. (a) The network to be inferred from time series measurements. (b) The cross-correlation between the time series of nodes 1 and 2. A strong peak is observed near -1.44 ms. (c) The cross-correlation between the time series of nodes 2 and 3. A strong negative peak is observed near 1.44 ms and a strong positive peak near -1.44 ms. (d) The cross-correlation between the time series of nodes 2 and 4. A strong peak is observed near 1.44 ms. (e) The cross-correlation between the time series of nodes 1 and 4. Peaks are observed at 1.44 and -1.44 ms, even though there is no link from node 1 to node 4.

determine the direction of a given putative link or if the link even exists at all.

Consider the network depicted in Fig. 11(a), where each node is an optoelectronic oscillator as described in Sec. III. The delay in each link is $\tau = 1.44$ ms. We define the cross-correlation between the sampled time series of two nodes i and j as

$$\rho_{ij}(\text{lag}) = \frac{1}{\sigma_i \sigma_j} \sum_k x_i[k + \text{lag}] x_j[k], \quad (\text{A1})$$

where $x_i[k]$ is the measured time series of node i at discrete time k and σ_i is the rms value of x_i . The time series should be mean subtracted so that $\langle x_i \rangle = 0$. The time series used

here are obtained from experimental measurements of our optoelectronic oscillator network.

First, we compute ρ_{12} , the cross-correlation between node 1 and node 2, shown in Fig. 11(b). A peak is located at -1.44 ms. This peak corresponds to the delay in the link from node 1 to node 2. This correspondence suggests that the dynamics of node 2 lag behind the dynamics of node 1, as one might expect, since the delayed link is from node 1 to node 2.

Next, we compute ρ_{23} , the cross-correlation between node 2 and node 3, shown in Fig. 11(c). The largest (negative, in this case) peak is located at 1.44 ms, correctly identifying the absolute value of the delay time of the link from node 2 to node 3. However, in contrast to ρ_{12} , the peak location in ρ_{23} shows that node 3 leads node 2. There is no

peak at a lag of -1.44 ms. This result shows that the cross-correlation can identify the delay time but not the link direction.

We now consider ρ_{24} . Nodes 2 and 4 have a bidirectional link; however, the cross-correlation ρ_{24} shown in Fig. 11(d) has a prominent peak at 1.44 ms but not at -1.44 ms. There is no indication that the link is bidirectional.

Finally, we consider ρ_{14} . There is no direct link between nodes 1 and 4. Still, the cross-correlation ρ_{24} shown in Fig. 11(e) has peaks at both 1.44 and -1.44 ms.

This example demonstrates that the cross-correlation can provide an accurate estimate of the duration of the delay in the coupling between two nodes but that it does not provide sufficient information to determine the existence or directionality of a link. We find similar results in all the networks of optoelectronic oscillators we test.

APPENDIX B: DERIVATION OF THE DISCRETE-TIME EQUATION FOR SIMULATING THE OPTOELECTRONIC SYSTEM

In this appendix, we derive the discrete-time equations implemented by the DSP board in our experimental setup and used in our simulations. These discrete-time equations are derived using standard techniques for approximating an analog filter as a digital filter and are essentially a trapezoid rule approximation to Eqs. (10) and (11).

The derivation here closely follows that presented in Ref. [77]. The missing details from Ref. [77] are filled in here, drawing from Ref. [98] for the details of the z transform and bilinear transform.

The continuous-time filter equations that describe a two-pole bandpass filter are

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{E}\mathbf{u}(t) - \mathbf{F}r(t), \quad (\text{B1})$$

$$x(t) = \mathbf{G}\mathbf{u}(t), \quad (\text{B2})$$

where

$$\mathbf{E} = \begin{bmatrix} -(\omega_L + \omega_H) & -\omega_L \\ \omega_H & 0 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \omega_L \\ 0 \end{bmatrix},$$

$$\text{and } \mathbf{G} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (\text{B3})$$

Here, $\mathbf{u}(t)$ is a 2-vector that describes the state of the filter, $r(t)$ is the filter input, and $x(t)$ is the filter output. In the case of one of our optoelectronic oscillators, $r(t) = \beta \cos^2[x(t - \tau) + \phi_0]$. In order to implement this filter digitally, one derives the digital filter equations by computing the transfer function of the analog filter and then applying the bilinear transform with frequency prewarping to the continuous-time transfer function to obtain the

discrete-time transfer function. From there, the discrete-time digital filter equations can be written down.

The transfer function $H(s)$ of the analog filter can be found by $H(s) \equiv X(s)/R(s)$, where the capital letters X and R indicate the Laplace transform of x and r , respectively. We compute the Laplace transform of Eq. (B1):

$$s\mathbf{U}(s) = \mathbf{E}\mathbf{U}(s) + \mathbf{F}R(s). \quad (\text{B4})$$

Then, performing the Laplace transform of Eq. (B2) and inserting Eq. (B4), we have

$$H(s) \equiv \frac{X(s)}{R(s)} = \mathbf{G}(s\mathbf{I} - \mathbf{E})^{-1}\mathbf{F} = \frac{s\tau_H}{(1 + \tau_L s)(1 + \tau_H s)}, \quad (\text{B5})$$

where $\tau_H = 1/\omega_H$ and $\tau_L = 1/\omega_L$. Equation (B5) is the continuous-time filter transfer function for the filter described by Eqs. (B1)–(B3).

Two standard tools used in the design and analysis of digital filters are the z transform and the bilinear transform. The z transform is the discrete-time analog of the Laplace transform. The bilinear transform is a tool used to turn a continuous-time filter into a discrete-time filter. It can be shown that the result obtained by the bilinear transform method we use here is equivalent to applying the trapezoidal integration rule to Eqs. (B1)–(B3) [98].

The z transform is defined as

$$Z\{x[n]\} \equiv \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad (\text{B6})$$

where z is a continuous complex variable and n is discrete time. One important z -transform relation is that a delay by m time steps in the discrete-time domain is equivalent to multiplication by z^{-m} in the z domain.

The bilinear transform is used to convert our continuous-time filter transfer function [Eq. (B5)] into a discrete-time filter transfer function. An exact conversion is done by discretizing with a time step of T and equating $z = e^{sT}$. Since T is small, we can approximate

$$s = \frac{1}{T} \ln(z) = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}. \quad (\text{B7})$$

Equation (B7) is the bilinear transform. This approximation is equivalent to applying the trapezoid rule to the continuous-time filter equations [98]. When Eq. (B7) is substituted into Eq. (B5), we obtain the transfer function for a discrete-time filter with similar characteristics to the desired analog filter:

$$H(z) = \frac{1}{4}(1 - z_L)(1 + z_H) \frac{1 - z^{-2}}{(1 - z_L z^{-1})(1 - z_H z^{-1})}. \quad (\text{B8})$$

This change of variables is a nonlinear mapping, so frequency warping occurs. This effect is minimal when the frequencies are significantly less than the Nyquist frequency (in this case, $f_L = 2.5$ kHz, and the Nyquist frequency is 12 kHz) and can be further mitigated by prewarping the frequencies of the continuous-time filter by $\Omega = (2/T) \tan(\omega/2)$, where Ω is the discrete-time frequency and ω is the continuous-time frequency [98]. Therefore, we find that

$$z_H = \frac{1 - \tan(T/2\tau_H)}{1 + \tan(T/2\tau_H)}$$

and

$$z_L = \frac{1 - \tan(T/2\tau_L)}{1 + \tan(T/2\tau_L)}.$$

Now, one can use the definition of the transfer function $H(z) \equiv X(z)/R(z)$ to find

$$\begin{aligned} & [1 - (z_L + z_H)z^{-1} + z_L z_H z^{-2}]X(z) \\ &= \frac{1}{4}(1 - z_L)(1 + z_H)(1 - z^{-2})R(z). \end{aligned} \quad (\text{B9})$$

We arrive at the discrete-time filter equation by performing the inverse z transform on Eq. (B9):

$$\begin{aligned} x[n] &= (z_L + z_H)x[n-1] - z_L z_H x[n-2] \\ &+ \frac{1}{4}(1 - z_L)(1 + z_H)(r[n] - r[n-2]). \end{aligned} \quad (\text{B10})$$

For the filter used in this work, $z_L + z_H = 1.4845$, $z_L z_H = 0.4968$, and $\frac{1}{4}(1 - z_L)(1 + z_H) = 0.242$.

-
- [1] R. Albert and A.-L. Barabasi, *Statistical Mechanics of Complex Networks*, *Rev. Mod. Phys.* **74**, 47 (2002).
 - [2] J. A. Dunne, R. J. Williams, and N. D. Martinez, *Food-Web Structure and Network Theory: The Role of Connectance and Size*, *Proc. Natl. Acad. Sci. U.S.A.* **99**, 12917 (2002).
 - [3] N. D. Price and I. Shmulevich, *Biochemical and Statistical Network Models for Systems Biology*, *Curr. Opin. Biotechnol.* **18**, 365 (2007).
 - [4] M. Vidal, M. E. Cusick, and A.-L. Barabasi, *Interactome Networks and Human Disease*, *Cell* **144**, 986 (2011).
 - [5] A. Baryshnikova, M. Costanzo, C. L. Myers, B. Andrews, and C. Boone, *Genetic Interaction Networks: Toward an Understanding of Heritability*, *Annu. Rev. Genomics Hum. Genet.* **14**, 111 (2013).

- [6] M. Costanzo, B. VanderSluis, E. N. Koch, A. Baryshnikova, C. Pons, G. Tan, W. Wang, M. Usaj, J. Hanchard, S. D. Lee *et al.*, *A Global Genetic Interaction Network Maps a Wiring Diagram of Cellular Function*, *Science* **353**, aaf1420-1 (2016).
- [7] D. S. Bassett and O. Sporns, *Network Neuroscience*, *Nat. Neurosci.* **20**, 353 (2017).
- [8] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, *Epidemic Processes in Complex Networks*, *Rev. Mod. Phys.* **87**, 925 (2015).
- [9] I. M. de Abril, J. Yoshimoto, and K. Doya, *Connectivity Inference from Neural Recording Data: Challenges, Mathematical Bases and Research Directions*, *Neural Netw.* **102**, 120 (2018).
- [10] M.-E. Lynall, D. S. Bassett, R. Kerwin, P. J. McKenna, M. Kitzbichler, U. Muller, and E. Bullmore, *Functional Connectivity and Brain Networks in Schizophrenia*, *J. Neurosci.* **30**, 9477 (2010).
- [11] M. Chavez, M. Valencia, V. Navarro, V. Latora, and J. Martinerie, *Functional Modularity of Background Activities in Normal and Epileptic Brain Networks*, *Phys. Rev. Lett.* **104**, 118701 (2010).
- [12] C. Sima, J. Hua, and S. Jung, *Inference of Gene Regulatory Networks Using Time-Series Data: A Survey*, *Curr. Genomics* **10**, 416 (2009).
- [13] R. Albert, *Network Inference, Analysis, and Modeling in Systems Biology*, *Plant Cell* **19**, 3327 (2007).
- [14] E. L. Sander, J. T. Wootton, and S. Allesina, *Ecological Network Inference from Long-Term Presence-Absence Data*, *Sci. Rep.* **7**, 7154 (2017).
- [15] I. Milns, C. M. Beale, and V. A. Smith, *Revealing Ecological Networks Using Bayesian Network Inference Algorithms*, *Ecology* **91**, 1892 (2010).
- [16] J. Runge, V. Petoukhov, J. F. Donges, J. Hlinka, N. Jajcay, M. Vejmelka, D. Hartman, N. Marwan, M. Palu, and J. Kurths, *Identifying Causal Gateways and Mediators in Complex Spatio-temporal Systems*, *Nat. Commun.* **6**, 8502 (2015).
- [17] P. Venegas, *Tracing the Untraceable: AI Network Inference for the Dark Web and Crypto Privacy Coins*, Authorea (2018) <https://doi.org/10.22541/au.154389280.02359798>.
- [18] M. Ding, Y. Chen, and S. L. Bressler, *Granger Causality: Basic Theory and Application to Neuroscience*, in *Handbook of Time Series Analysis* (Wiley, New York, 2006), Chap. 17, pp. 437–460.
- [19] D. Zhou, Y. Xiao, Y. Zhang, Z. Xu, and D. Cai, *Causal and Structural Connectivity of Pulse-Coupled Nonlinear Networks*, *Phys. Rev. Lett.* **111**, 054102 (2013).
- [20] T. Schreiber, *Measuring Information Transfer*, *Phys. Rev. Lett.* **85**, 461 (2000).
- [21] J. Sun, D. Taylor, and E. M. Bollt, *Causal Network Inference by Optimal Causation Entropy*, *SIAM J. Appl. Dyn. Syst.* **14**, 73 (2015).
- [22] J. Casadiego, D. Maoutsa, and M. Timme, *Inferring Network Connectivity from Event Timing Patterns*, *Phys. Rev. Lett.* **121**, 054101 (2018).
- [23] C. Zou and J. Feng, *Granger Causality vs. Dynamic Bayesian Network Inference: A Comparative Study*, *BMC Bioinf.* **10**, 122 (2009).

- [24] T. P. Peixoto, *Network Reconstruction and Community Detection from Dynamics*, *Phys. Rev. Lett.* **123**, 128301 (2019).
- [25] J. 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).
- [26] D. Dahmen, H. Bos, and M. Helias, *Correlated Fluctuations in Strongly Coupled Binary Networks beyond Equilibrium*, *Phys. Rev. X* **6**, 031024 (2016).
- [27] S. Leng, Z. Xu, and H. Ma, *Reconstructing Directional Causal Networks with Random Forest: Causality Meeting Machine Learning*, *Chaos* **29**, 093130 (2019).
- [28] M. G. Leguia, Z. Levnajić, L. Todorovski, and B. Ženko, *Reconstructing Dynamical Networks via Feature Ranking*, *Chaos* **29**, 093107 (2019).
- [29] A. Banerjee, J. Pathak, R. Roy, J. G. Restrepo, and E. Ott, *Using Machine Learning to Assess Short Term Causal Dependence and Infer Network Links*, *Chaos* **29**, 121104 (2019).
- [30] J. Chen, X. Lin, C. Jia, Y. Li, Y. Wu, H. Zheng, and Y. Liu, *Generative Dynamic Link Prediction*, *Chaos* **29**, 123111 (2019).
- [31] R.-M. Cao, S.-Y. Liu, and X.-K. Xu, *Network Embedding for Link Prediction: The Pitfall and Improvement*, *Chaos* **29**, 103102 (2019).
- [32] N. Frolov, V. Maksimenko, A. Lttjohann, A. Koronovskii, and A. Hramov, *Feed-Forward Artificial Neural Network Provides Data-Driven Inference of Functional Connectivity*, *Chaos* **29**, 091101 (2019).
- [33] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning* (MIT Press, Cambridge, MA, 2016), Vol. 1.
- [34] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborova, *Machine Learning and the Physical Sciences*, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [35] G. Tanaka, T. Yamane, J. B. Heroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, *Recent Advances in Physical Reservoir Computing: A Review*, *Neural Netw.* **115**, 100 (2019).
- [36] H. Jaeger and H. Haas, *Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication*, *Science* **304**, 78 (2004).
- [37] 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).
- [38] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, *Hybrid Forecasting of Chaotic Processes: Using Machine Learning in Conjunction with a Knowledge-Based Model*, *Chaos* **28**, 041101 (2018).
- [39] G. Neofotistos, M. Mattheakis, G. D. Barnparis, J. Hizanidis, G. P. Tsironis, and E. Kaxiras, *Machine Learning with Observers Predicts Complex Spatiotemporal Behavior*, *Front. Phys.* **7**, 24 (2019).
- [40] R. S. Zimmermann and U. Parlitz, *Observing Spatiotemporal Dynamics of Excitable Media Using Reservoir Computing*, *Chaos* **28**, 043118 (2018).
- [41] 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).
- [42] S. Krishnagopal, M. Girvan, E. Ott, and B. R. Hunt, *Separation of Chaotic Signals by Reservoir Computing*, *Chaos* **30**, 023123 (2020).
- [43] Z. Lu, J. Z. Kim, and D. S. Bassett, *Supervised Chaotic Source Separation by a Tank of Water*, *Chaos* **30**, 021101 (2020).
- [44] Y. K. Chembo, *Machine Learning Based on Reservoir Computing with Time-Delayed Optoelectronic and Photonic Systems*, *Chaos* **30**, 013111 (2020).
- [45] D. Markovic, A. Mizrahi, D. Querlioz, and J. Grollier, *Physics for Neuromorphic Computing*, *arXiv:2003.04711*.
- [46] L. 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).
- [47] Q. Vinckier, F. Duport, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar, *High-Performance Photonic Reservoir Computer Based on a Coherently Driven Passive Cavity*, *Optica* **2**, 438 (2015).
- [48] F. Duport, A. Smerieri, A. Akrouit, M. Haelterman, and S. Massar, *Fully Analogue Photonic Reservoir Computer*, *Sci. Rep.* **6**, 22381 (2016).
- [49] B. Schrauwen, M. DHaene, D. Verstraeten, and J. Van Campenhout, *Compact Hardware Liquid State Machines on FPGA for Real-Time Speech Recognition*, *Neural Netw.* **21**, 511 (2008).
- [50] 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).
- [51] E. J. Molinelli, A. Korkut, W. Wang, M. L. Miller, N. P. Gauthier, X. Jing, P. Kaushik, Q. He, G. Mills, D. B. Solit et al., *Perturbation Biology: Inferring Signaling Networks in Cellular Systems*, *PLoS Comput. Biol.* **9**, e1003290 (2013).
- [52] A. Das and I. R. Fiete, *Systematic Errors in Connectivity Inferred from Activity in Strongly Recurrent Networks*, *Nat. Neurosci.* **23**, 1286 (2020).
- [53] C. Olsen, K. Fleming, N. Prendergast, R. Rubio, F. Emmert-Streib, G. Bontempi, B. Haibe-Kains, and J. Quackenbush, *Inference and Validation of Predictive Gene Networks from Biomedical Literature and Gene Expression Data*, *Genomics* **103**, 329 (2014).
- [54] X. S. Yao and L. Maleki, *Optoelectronic Microwave Oscillator*, *J. Opt. Soc. Am. B* **13**, 1725 (1996).
- [55] Y. C. Koumou, P. Colet, L. Larger, and N. Gastaud, *Chaotic Breathers in Delayed Electro-optical Systems*, *Phys. Rev. Lett.* **95**, 203903 (2005).
- [56] K. E. Callan, L. Illing, Z. Gao, D. J. Gauthier, and E. Scholl, *Broadband Chaos Generated by an Optoelectronic Oscillator*, *Phys. Rev. Lett.* **104**, 113901 (2010).
- [57] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, *Photonic Information Processing beyond Turing: An Optoelectronic Implementation of Reservoir Computing*, *Opt. Express* **20**, 3241 (2012).

- [58] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, *Optoelectronic Reservoir Computing*, *Sci. Rep.* **2**, 287 (2012).
- [59] A. Argyris, D. Syvridis, L. Larger, V. Annovazzi-Lodi, P. Colet, I. Fischer, J. Garcia-Ojalvo, C.R. Mirasso, L. Pesquera, and K.A. Shore, *Chaos-Based Communications at High Bit Rates Using Commercial Fibre-Optic Links*, *Nature (London)* **438**, 343 (2005).
- [60] J. Yao, *Optoelectronic Oscillators for High Speed and High Resolution Optical Sensing*, *J. Lightwave Technol.* **35**, 3489 (2017).
- [61] L. Larger and J.M. Dudley, *Optoelectronic Chaos*, *Nature (London)* **465**, 41 (2010).
- [62] L. Larger, *Complexity in Electro-optic Delay Dynamics: Modelling, Design and Applications*, *Phil. Trans. R. Soc. A* **371**, 20120464 (2013).
- [63] Y.K. Chembo, D. Brunner, M. Jacquot, and L. Larger, *Optoelectronic Oscillators with Time-Delayed Feedback*, *Rev. Mod. Phys.* **91**, 035006 (2019).
- [64] L. Illing, C.D. Panda, and L. Shreshian, *Isochronal Chaos Synchronization of Delay-Coupled Optoelectronic Oscillators*, *Phys. Rev. E* **84**, 016213 (2011).
- [65] B. Ravoori, A.B. Cohen, J. Sun, A.E. Motter, T.E. Murphy, and R. Roy, *Robustness of Optimal Synchronization in Real Networks*, *Phys. Rev. Lett.* **107**, 034102 (2011).
- [66] C.R.S. Williams, T.E. Murphy, R. Roy, F. Sorrentino, T. Dahms, and E. Scholl, *Experimental Observations of Group Synchrony in a System of Chaotic Optoelectronic Oscillators*, *Phys. Rev. Lett.* **110**, 064104 (2013).
- [67] J.D. Hart, K. Bansal, T.E. Murphy, and R. Roy, *Experimental Observation of Chimera and Cluster States in a Minimal Globally Coupled Network*, *Chaos* **26**, 094801 (2016).
- [68] J.D. Hart, J.P. Pade, T. Pereira, T.E. Murphy, and R. Roy, *Adding Connections Can Hinder Network Synchronization of Time-Delayed Oscillators*, *Phys. Rev. E* **92**, 022804 (2015).
- [69] A.E. Hoerl and R.W. Kennard, *Ridge Regression: Biased Estimation for Nonorthogonal Problems*, *Technometrics* **12**, 55 (1970).
- [70] T. Lymburn, D.M. Walker, M. Small, and T. Jngling, *The Reservoirs Perspective on Generalized Synchronization*, *Chaos* **29**, 093133 (2019).
- [71] Z. Lu, B.R. Hunt, and E. Ott, *Attractor Reconstruction by Machine Learning*, *Chaos* **28**, 061104 (2018).
- [72] E. Bollt, *On Explaining the Surprising Success of Reservoir Computing Forecaster of Chaos? The Universal Machine Learning Dynamical System with Contrast to VAR and DMD*, *Chaos* **31**, 013108 (2021).
- [73] R. Penrose, *A Generalized Inverse for Matrices*, in *Mathematical Proceedings of the Cambridge Philosophical Society* (Cambridge University Press, Cambridge, England, 1955), Vol. 51, pp. 406–413.
- [74] T. Fawcett, *An Introduction to ROC Analysis*, *Pattern Recogn. Lett.* **27**, 861 (2006).
- [75] Y. Yang, R.N. Lichtenwalter, and N.V. Chawla, *Evaluating Link Prediction Methods*, *Knowl. Inf. Syst.* **45**, 751 (2015).
- [76] G. Cecchini, R. Cestnik, and A. Pikovsky, *Impact of Local Network Characteristics on Network Reconstruction*, *Phys. Rev. E* **103**, 022305 (2021).
- [77] T.E. Murphy, A.B. Cohen, B. Ravoori, K.R. Schmitt, A.V. Setty, F. Sorrentino, C.R. Williams, E. Ott, and R. Roy, *Complex Dynamics and Synchronization of Delayed-Feedback Nonlinear Oscillators*, *Phil. Trans. R. Soc. A* **368**, 343 (2010).
- [78] J.D. Hart, L. Larger, T.E. Murphy, and R. Roy, *Delayed Dynamical Systems: Networks, Chimeras and Reservoir Computing*, *Phil. Trans. R. Soc. A* **377**, 20180123 (2019).
- [79] J.D. Hart, *Experiments on Networks of Coupled Optoelectronic Oscillators and Phys. Random Number Generators*, Ph.D. thesis, University of Maryland, College Park, 2018, <https://doi.org/10.13016/jf05-uw3i>.
- [80] V. Flunkert, S. Yanchuk, T. Dahms, and E. Scholl, *Synchronizing Distant Nodes: A Universal Classification of Networks*, *Phys. Rev. Lett.* **105**, 254101 (2010).
- [81] A. Townsend, M. Stillman, and S.H. Strogatz, *Dense Networks That Do Not Synchronize and Sparse Ones That Do*, *Chaos* **30**, 083142 (2020).
- [82] M.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).
- [83] M.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).
- [84] J. Lipinski-Kruszka, J. Stewart-Ornstein, M.W. Chevalier, and H. El-Samad, *Using Dynamic Noise Propagation to Infer Causal Regulatory Relationships in Biochemical Networks*, *ACS Synth. Biol.* **4**, 258 (2015).
- [85] R.J. Prill, R. Vogel, G.A. Cecchi, G. Altan-Bonnet, and G. Stolovitzky, *Noise-Driven Causal Inference in Biomolecular Networks*, *PLoS One* **10**, e0125777 (2015).
- [86] W.-X. Wang, J. Ren, Y.-C. Lai, and B. Li, *Reverse Engineering of Complex Dynamical Networks in the Presence of Time-Delayed Interactions Based on Noisy Time Series*, *Chaos* **22**, 033131 (2012).
- [87] L.M. Pecora, F. Sorrentino, A.M. Hagerstrom, T.E. Murphy, and R. Roy, *Cluster Synchronization and Isolated Desynchronization in Complex Networks with Symmetries*, *Nat. Commun.* **5**, 4079 (2014).
- [88] N.F. Rulkov, M.M. Sushchik, L.S. Tsimring, and H.D.I. Abarbanel, *Generalized Synchronization of Chaos in Directionally Coupled Chaotic Systems*, *Phys. Rev. E* **51**, 980 (1995).
- [89] L. Kocarev and U. Parlitz, *Generalized Synchronization, Predictability, and Equivalence of Unidirectionally Coupled Dynamical Systems*, *Phys. Rev. Lett.* **76**, 1816 (1996).
- [90] D. Senthilkumar, M. Lakshmanan, and J. Kurths, *Transition from Phase to Generalized Synchronization in Time-Delay Systems*, *Chaos* **18**, 023118 (2008).
- [91] D. Senthilkumar, R. Suresh, M. Lakshmanan, and J. Kurths, *Global Generalized Synchronization in Networks of Different Time-Delay Systems*, *Europhys. Lett.* **103**, 50010 (2013).
- [92] A.J. Whalen, S.N. Brennan, T.D. Sauer, and S.J. Schiff, *Observability and Controllability of Nonlinear Networks: The Role of Symmetry*, *Phys. Rev. X* **5**, 011005 (2015).

- [93] M. Denker, M. Timme, M. Diesmann, F. Wolf, and T. Geisel, *Breaking Synchrony by Heterogeneity in Complex Networks*, *Phys. Rev. Lett.* **92**, 074103 (2004).
- [94] T. Nishikawa, A. E. Motter, Y.-C. Lai, and F. C. Hoppensteadt, *Heterogeneity in Oscillator Networks: Are Smaller Worlds Easier to Synchronize?*, *Phys. Rev. Lett.* **91**, 014101 (2003).
- [95] X. Han, Z. Shen, W.-X. Wang, and Z. Di, *Robust Reconstruction of Complex Networks from Sparse Data*, *Phys. Rev. Lett.* **114**, 028701 (2015).
- [96] T. P. Peixoto, *Reconstructing Networks with Unknown and Heterogeneous Errors*, *Phys. Rev. X* **8**, 041011 (2018).
- [97] D. Napoletani and T. D. Sauer, *Reconstructing the Topology of Sparsely Connected Dynamical Networks*, *Phys. Rev. E* **77**, 026103 (2008).
- [98] A. V. Oppenheim, J. R. Buck, and R. W. Schaffer, *Discrete-Time Signal Processing* (Prentice-Hall, Upper Saddle River, NJ, 2001), Vol. 2.