

Detecting causality using symmetry transformations

Subhradeep Roy, and Benjamin Jantzen

Citation: *Chaos* **28**, 075305 (2018); doi: 10.1063/1.5018101

View online: <https://doi.org/10.1063/1.5018101>

View Table of Contents: <http://aip.scitation.org/toc/cha/28/7>

Published by the [American Institute of Physics](#)

Articles you may be interested in

[Detecting directional couplings from multivariate flows by the joint distance distribution](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 075302 (2018); 10.1063/1.5010779

[Transient and equilibrium causal effects in coupled oscillators](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 075303 (2018); 10.1063/1.5017821

[State space reconstruction of spatially extended systems and of time delayed systems from the time series of a scalar variable](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 075504 (2018); 10.1063/1.5023485

[Attractor reconstruction by machine learning](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 061104 (2018); 10.1063/1.5039508

[Inter-scale information flow as a surrogate for downward causation that maintains spiral waves](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 075306 (2018); 10.1063/1.5017534

[Space-time nature of causality](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 075509 (2018); 10.1063/1.5019917



Detecting causality using symmetry transformations

Subhradeep Roy^{a)} and Benjamin Jantzen^{b)}

Department of Philosophy (0126), Major Williams 229, 220 Stanger St, Virginia Tech, Blacksburg, Virginia 24061, USA

(Received 03 December 2017; accepted 29 April 2018; published online 12 July 2018)

Detecting causality between variables in a time series is a challenge, particularly when the relationship is nonlinear and the dataset is noisy. Here, we present a novel tool for detecting causality that leverages the properties of symmetry transformations. The aim is to develop an algorithm with the potential to detect both unidirectional and bidirectional coupling for nonlinear systems in the presence of significant sampling noise. Most of the existing tools for detecting causality can make determinations of directionality, but those determinations are relatively fragile in the presence of noise. The novel algorithm developed in the present study is robust and very conservative in that it reliably detects causal structure with a very low rate of error even in the presence of high sampling noise. We demonstrate the performance of our algorithm and compare it with two popular model-free methods, namely transfer entropy and convergent cross map. This first implementation of the method of symmetry transformations is limited in that it applies only to first-order autonomous systems. *Published by AIP Publishing.* <https://doi.org/10.1063/1.5018101>

Real world systems are composed of a number of interacting units. Identifying causality refers to detecting how these units are coupled and quantifying the strength of that coupling from time series data. Detecting causality is particularly important in understanding real-world systems, and hence has attracted researchers from different disciplines to develop a number of causality detecting tools. Here, we present a novel causality detecting algorithm that utilizes the property of dynamical symmetry. The application of the present version of the algorithm is limited to first-order autonomous systems. The performance of the algorithm is demonstrated with toy models from ecology and a real-world electronic system and is found to be very reliable in correctly identifying the coupling between variables. In other words, it successfully discovers the presence and direction of causal edges. We further discuss the scope of improvement of the present version of the algorithm that may find application in more general scenarios.

I. INTRODUCTION

Detecting causality between observed time series refers to the identification of dominant coupling direction and the quantification of coupling strengths between the interacting time series variables. Identifying such causal relationships from observational or experimental data is also often referred to as “causal discovery.”¹ Identifying causality is of great importance as it has applications in unraveling various real-world phenomena, ranging from climate networks² to brain networks.³ The challenge of learning causal relations for such a diversity of systems has attracted researchers from various disciplines, and a variety of methods have been proposed. One early method is Granger causality,⁴ which relies on the idea

of separability. This refers to the removal of information of the causative factors from the effects.^{5,6} Originally, the application of Granger causality was limited because it assumed linear dependence,³ whereas most real-world systems are nonlinear. But recently, Granger causality has been extended to nonlinear systems.⁷ Another approach for detecting causality in linear systems is by computing cross-correlation^{8,9} between time series for a range of time lags, and identifying the maximum in correlation value at some non-zero lag. The leading time series variable is then inferred to be the cause.

To infer causality in non-linear systems, model-free methods (that do not assume a background model), such as transfer entropy,¹⁰ convergent cross map,⁵ recurrence plots,^{3,11} causation entropy,¹² and phase dynamics^{13,14} are becoming increasingly popular. A tool to detect the direction of coupling in interacting oscillators is developed in Refs. 13 and 14 and works by observing an asymmetric failure of invariance in the time evolution of oscillator phases. Roughly, if oscillator 1 drives oscillator 2, then the change in phase of oscillator 2 over a fixed interval of time will depend upon the phase of oscillator 1 at the outset, but not vice versa. On the other hand, transfer entropy (TE) is based on an information-theoretic concept and employs the idea that causes predict their effects. In particular, it quantifies the information flow between time series variables and identifies the direction of causation via the dominant direction of information flow. Transfer entropy has already found applications in neuroscience,¹⁵ finance,¹⁶ social media,¹⁷ animal behavior,^{8,18,19} and public policy.²⁰ The main limitation of TE is that the method requires long time series which may not be available in practical situations.

A relatively recent but popular method is the convergent cross map (CCM),⁵ which is based on a state space reconstruction technique. CCM introduces a new criterion for detecting causality. In this case, the effects cross-estimate the causes contrary to the Granger and TE sense.²¹ In Ref. 5, CCM is identified as a superior method to both TE and Granger

^{a)} Electronic mail: sdroy@vt.edu

^{b)} Electronic mail: bjantzen@vt.edu

causality in correctly detecting causality.⁵ CCM has already found application in diverse settings.^{5,22–25} The major limitation of CCM is that it can only detect low to moderate coupling and fails in the presence of strong coupling and for noisy data.²⁶

Despite their broad applicability, TE and CCM continue to struggle with noisy data or small datasets. We present a first version of an algorithm that, though of limited range of application, is robust for small, noisy datasets. This method detects the direction(s) of causal influence by utilizing the concept of dynamical symmetries.²⁷

The paper is organized as follows. In Sec. II, we describe dynamical symmetries, and how properties of these symmetries can be exploited in detecting causality. In Sec. III, we present the algorithm. In Sec. IV, we explain TE and CCM. In Sec. V, we introduce Lotka-Volterra models as toy target systems and demonstrate the performance of our algorithm in detecting unidirectional and bidirectional causality. We further compare the performance of our algorithm with that of TE and CCM on real world systems. Finally, we present the limitations of our algorithm and the scope of improvement in Sec. VI.

II. METHOD

The concept of dynamical symmetry was first introduced in Ref. 28 and has proved to be a powerful tool for classifying dynamical systems according to higher-order properties of their causal structure.²⁷ For purposes of time series analysis, we use the following definition of the concept:

Dynamical symmetry: Let V be the set of states of a system of n variables, and let Λ_{t_1, t_0} be the state evolution operator that takes the system from some state $v_0 \in V$ at time t_0 to the corresponding evolved state $v_1 \in V$ at t_1 . The transformation $\sigma : V \rightarrow V$ is a dynamical symmetry with respect to time if and only if it has the following property: $\forall v \in V \forall t_0 \forall t_1 > t_0 [\Lambda_{t_1, t_0}(\sigma(v)) = \sigma(\Lambda_{t_1, t_0}(v))]$. In other words, the set of dynamical symmetries is the set of transformations of the system that commute with its time evolution.

For deterministic dynamics described by a system of differential equations, dynamical symmetries correspond to automorphisms on the space of solutions, which in the case of dynamical systems are trajectories. Consider two arbitrary trajectories such that at t_0 , the system is in state s_{1, t_0} on trajectory 1 and state s_{2, t_0} on trajectory 2. Then the map that pairs states $s_{1, t_0 + \delta}$ with $s_{2, t_0 + \delta}$ for any $\delta \geq 0$ is equivalent to a dynamical symmetry. To see this, consider any paired states $s_{1, t_0 + \delta}$ and $s_{2, t_0 + \delta}$. If one evolves the state s_{1, t_0} for time δ and then applies the map, one obtains $s_{2, t_0 + \delta}$, exactly as if one had applied the map at t_0 and then evolved through an interval of δ . The map in question is also a function (generally a vector function) since, under the assumption that the system is deterministic, no trajectory can repeat a state (i.e., no trajectories intersect with themselves or others).

The set of dynamical symmetries of a system is determined by its causal structure. Furthermore, dynamical symmetries can be directly estimated from data without first learning a dynamical model.²⁷ For these reasons, dynamical symmetry can be used to probe causal structure.

In the present study, we exploit the nature of dynamical symmetries to identify the causal influences in a system composed of two time-dependent variables that may be coupled in either or both directions. To illustrate our procedure, consider a unidirectionally coupled, autonomous, first-order system of the form:

$$\dot{x} = h_1(x, \dot{y}, y), \quad (1a)$$

$$\dot{y} = h_2(y). \quad (1b)$$

Note that the dynamics of x is dependent upon the evolution of y , but not vice versa. This sort of coupling is what we have in mind when speaking of causal influence in such a system. In this case, the coupling is “unidirectional” from y to x .

A common representational tool in causal discovery is the structural equation model (SEM)²⁹ (Chaps. 1 and 5). If a system such as Eq. 1 has solutions for the given h_1 and h_2 , then we can equivalently represent it with a structural equation model.²⁷ This is because one can express a flow of the dynamical system as a function of time and the present and past values of the dynamical variables. In this case, we have

$$t := t, \quad (2a)$$

$$x := f(t, y; x_0), \quad (2b)$$

$$y := g(t; y_0). \quad (2c)$$

In the SEM, $f(t, y)$ and $g(t)$ capture the dynamics of the system, while x_0 and y_0 are initial conditions for the variables x and y , respectively. Note that g is strictly a function of time, but x depends on y , again reflecting the asymmetry of influence.

Our method depends upon examining the dynamical symmetries of a system. More specifically, we look at differences in their one-variable projections. Given any two trajectories of a deterministic system of two dynamical variables (x and y), there is (at least one) dynamical symmetry that maps one into the other. In general, these symmetries can be represented by two-component vector functions of system states. Each component of such a vector function can be viewed as a surface whose distance above or below a point in the x, y phase plane is the value to which that component of the dynamical symmetry maps the point. Sequences of points in the phase plane are thus mapped by a dynamical symmetry to curves on this surface. Our method relies in effect on detecting differences in the shape of these surfaces along different basis directions. We do so by looking at the projection of curves on these surfaces onto the plane corresponding to one or the other dynamical variable. For example, the x -component of the vector function representing a symmetry may be written $\sigma_x(x, y)$. Consider a curve in the phase plane given by $\langle r(t), s(t) \rangle$. This is mapped by a dynamical symmetry to $\sigma_x(r(t), s(t)) = \langle u(t), v(t) \rangle$, a curve on the surface dictated by $\sigma_x(x, y)$. If we project this curve onto the plane containing the x -axis, we obtain a new curve, denoted $\sigma(x)$. In this case, when parameterized by time, this curve is given by $\sigma(x(t)) = u(t)$. In some cases, the projected curve $\sigma(x)$ is in fact a proper function of x and thus represents a dynamical symmetry of the system that involves intervention on only the variable x . However, it is often the

TABLE I. Symmetry transformation for $x: \sigma(x)$.

$\Lambda_{\Delta,0}(\sigma(x))$		
t	x	y
0	x_0	y_0
0	$\sigma(x_0)$	y_0
Δ	$f(\Delta, g(\Delta; y_0); \sigma(x_0))$	$g(\Delta; y_0)$
$\sigma(\Lambda_{\Delta,0}(x))$		
t	x	y
0	x_0	y_0
Δ	$f(\Delta, g(\Delta; y_0); x_0)$	$g(\Delta; y_0)$
Δ	$\sigma(f(\Delta, g(\Delta; y_0); x_0))$	$g(\Delta; y_0)$

TABLE II. Symmetry transformation for $y: \sigma(y)$.

$\Lambda_{\Delta,0}(\sigma(y))$		
t	x	y
0	x_0	y_0
0	$f(0, \sigma(y_0); x_0)$	$\sigma(y_0)$
Δ	$f(\Delta, g(\Delta; y_0); x_0)$	$g(\Delta; \sigma(y_0))$
$\sigma(\Lambda_{\Delta,0}(y))$		
t	x	y
0	x_0	y_0
Δ	$f(\Delta, g(\Delta; y_0); x_0)$	$g(\Delta; y_0)$
Δ	$f(\Delta, \sigma(g(\Delta; y_0)); x_0)$	$\sigma(g(\Delta; y_0))$

case that $\sigma(x)$ is not a function. For this reason, we call these projections *pseudo-symmetries*.

To derive implicit equations for the pseudo-symmetries involving only x [denoted $\sigma(x)$] and those involving only y [denoted $\sigma(y)$] given the above assumptions, consider the series of operations summarized in Tables I and II, respectively. To find $\sigma(x)$ we start from $t = 0$ then apply the transformation on x followed by time evolution. Each row of the center column of the table indicates the state of the system after each of these operations as dictated by Eq. (2). Next, we start from $t = 0$, but change the order of time evolution and the transformation on x . The resulting system states are shown in the right-hand column. This procedure yields a condition that must be satisfied for any transformation, $\sigma(x)$, to qualify as a pseudo-symmetry. From the final row of Table I, we can see that any transformation of x that is a pseudo-symmetry must satisfy

$$f(\Delta, g(\Delta; y_0); \sigma(x_0)) = \sigma(f(\Delta, g(\Delta; y_0); x_0)). \quad (3)$$

Importantly, this condition entails that any symmetry $\sigma(x)$ is dependent upon both initial conditions, x_0 and y_0 . That is, the set of pseudo-symmetries of x are determined in part by the initial value of y .

We perform a similar exercise to obtain the condition that $\sigma(y)$ must satisfy. From Table II, it is clear that the pseudo-symmetries involving y alone must satisfy

$$g(\Delta; \sigma(y_0)) = \sigma(g(\Delta; y_0)), \quad (4)$$

which is independent of the initial condition for x . The reason $\sigma(y)$ is independent of x_0 is because the evolution of y as given in Eq. (2) is not influenced by x . A similar argument to the above shows that if the causal influence is bidirectional—if x influences y and vice versa—then $\sigma(x)$ would depend upon

TABLE III. Notation for trajectories.

Initial conditions	Notation for x	Notation for y
$[\gamma, \alpha]$	\vec{x}_α	\vec{y}_γ
$[\gamma, \beta]$	\vec{x}_β	\vec{y}_γ
$[\delta, \alpha]$	\vec{x}_α	\vec{y}_δ
$[\delta, \beta]$	\vec{x}_β	\vec{y}_δ

the initial value of y and $\sigma(y)$ would depend upon the initial value of x .

The central idea of the present study is to exploit this pattern of dependence on initial conditions to infer the direction(s) of causal influence. By estimating $\sigma(y)$ for two different initial conditions of x , one can ascertain whether $\sigma(y)$ depends upon x_0 . If it does, then x is a cause of y . Conversely, if $\sigma(x)$ is found to depend upon y , then y is a cause of x , as in the example above. And if both one-variable pseudo-symmetries depend upon the initial values of the other variable, then the causal influence is mutual (i.e., x and y influence each other). Note, however, that this decision procedure does not indicate the degree to which one influences the other.

This approach depends upon comparing pairs of pseudo-symmetries, $\sigma(x)$ and $\sigma(y)$, for the dynamical system of interest. In practice, the information to do so can be gleaned from a single time series of sufficient length, but we assume here that multiple replicates of a dynamical system are used, as depicted in Fig. 1. In the first (System 1), we start by choosing two initial conditions for x ($x_0 = \gamma$ and $x_0 = \delta$) while keeping the initial condition for y fixed at α . System 2 is set up similarly, except that y_0 is fixed at β . The pseudo-symmetries of interest are the functions $\vec{x}_\alpha(\vec{x}_\alpha)$ and $\vec{x}_\beta(\vec{x}_\beta)$ that, respectively, map values of x in the time series \vec{x} to the corresponding (same time) value in the time series vector, \vec{x} (the subscripts α and β refer to the initial value of y for each system). These two functions are denoted $\sigma_1(x)$ and $\sigma_2(x)$ in Fig. 1(a). Similarly, we set up two additional systems, System 3 and System 4, to produce two time series for the variable y , each starting from a different initial condition, while the initial value of x is set to either δ (for System 3) or γ (for System 4). The relevant pseudo-symmetries of y in this case are the functions denoted $\sigma_3(y)$ and $\sigma_4(y)$ in Fig. 1(b). The subscripts δ and γ refer to initial value of x in Systems 3 and 4, respectively. Table III presents a complete notation for the respective trajectories.

III. THE ALGORITHM

The algorithm is based on a technique that was first introduced in Ref. 27, where the dynamical symmetries of two systems are compared to determine whether the systems share a similar dynamics. In case they do not share the same symmetry transformations, they are confidently determined to belong to “different kinds”; otherwise, they are provisionally taken to belong to the “same kind.” In the present study, a similar procedure is adopted to compare pairs of pseudo-symmetries and use the pattern of results to detect causal influences. To detect causality in this way, we proceed in four steps: (i) sampling,

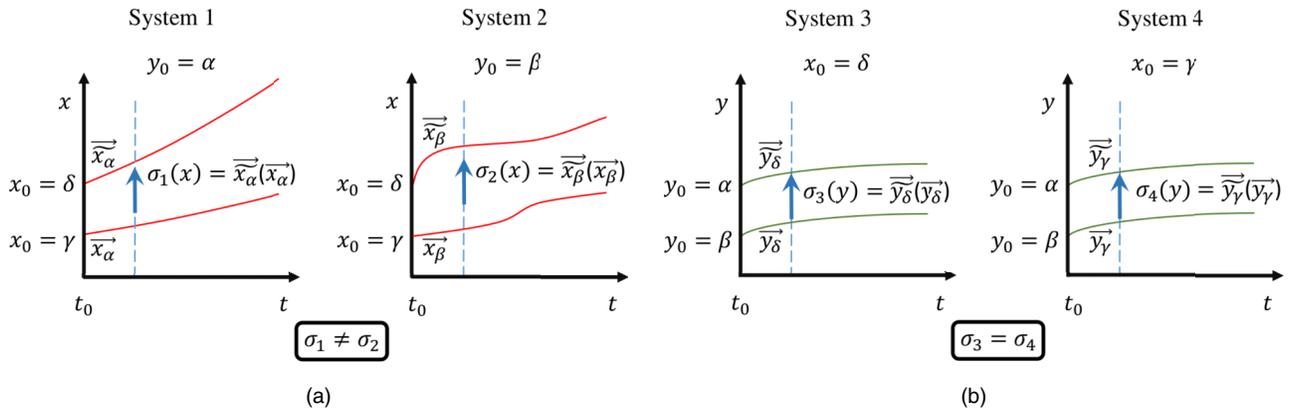


FIG. 1. The schematic shows how dynamical symmetries can be used to identify causal direction. In this example, the variable y is driving x . Each pane depicts a pair of trajectories for four different replicates of a dynamical system. In Systems 1 and 2, initial conditions for y are fixed, and functions σ_1 and σ_2 are the functions that map the time series vector \vec{x}_α to \vec{x}_δ and the time series vector \vec{x}_β to \vec{x}_γ , respectively. Similarly, for Systems 3 and 4, σ_3 and σ_4 are the functions connecting trajectories of y for which the initial values of x are set to $x_0 = \delta$ and $x_0 = \gamma$, respectively. The functions σ_1 and σ_2 depend on the initial values of y , and hence $\sigma_1 \neq \sigma_2$. However, since the variable y is independent of x , $\sigma_3 = \sigma_4$.

(ii) data transformation, (iii) comparison (of σ_1 with σ_2 , and σ_3 with σ_4), and (iv) decision.

In step (i), we sample from four systems, where the governing equations and the parameters remain the same, but the initial conditions vary as depicted in Fig. 1. For System 1, two time series are measured using the same initial condition for the y variable ($y_0 = \alpha$), different initial conditions for x , namely $x_0 = \delta$ and $x_0 = \gamma$. Similarly, for System 2, we set $y_0 = \beta$, and choose two initial conditions for x , $x_0 = \delta$, and $x_0 = \gamma$. The trajectories are obtained by measuring the values of the variables at times t_0, \dots, t_m for each of the systems. Similarly, time series are acquired from System 3 and System 4. In this case, the initial value of x is held constant, and measurements are made on each system for two different initial values of y . In all cases, measurements are made at the same set of times, t_0, t_1, \dots, t_m . We choose segments such that x and y vary monotonically. This guarantees that the pseudo-symmetries are functions over at least this interval, which allows us to use the comparative tools described below.

In step (ii), we combine pairs of time series acquired in step (i) to generate a sample of each of the functions σ_i . Specifically, for System 1 we pair the vector of x -values, \vec{x}_α for the time series beginning with $x_0 = \gamma$ with the vector of x -values \vec{x}_δ from the time series beginning with $x_0 = \delta$. The result is a two-column matrix (represented in Algorithm 1 as input $A_{\sigma_1, x}$) for which the i -th row contains elements from each of the two time series at the time t_i . The function mapping the values in the first column to those of the second [represented by $\vec{x}_\delta(\vec{x}_\alpha)$] is the pseudo-symmetry $\sigma_1(x)$. We similarly pair the time series vectors of x or y variables for Systems 2–4.

In step (iii), we determine whether $\sigma_1 = \sigma_2$ and whether $\sigma_3 = \sigma_4$. To detect whether the pseudo-symmetries differ between Systems 1 and 2, we estimate the error of two competing models, *sep* and *joint*, using 10-fold cross-validation. In the *sep* model, data from System 1 and System 2 are treated separately, whereas for the *joint* model, the data from System 1 and System 2 are combined and treated as a single sample from the same function. In either case, we fitted polynomial

functions to estimate the pseudo-symmetries. For a given dataset, the order of the polynomial functions to be fit is determined by an additional 10-fold cross-validation (represented by FitPolynomial in Algorithm 1), as was done in Ref. 27. This function starts by considering a polynomial of order 1 and computes the mean-squared error of fitting a first-order polynomial over 10 folds of the data. The function continues fitting the data with higher order polynomials and compares the mean-squared error with that of the preceding order. If the error increases, the function terminates and returns the order that minimized the mean square error. The polynomial coefficients are then evaluated by fitting the entire data set with the order as returned by FitPolynomial in Algorithm 1.

It is important to note that mean squared errors computed for the *sep* and *joint* models cannot be directly compared, because the error depends on how the data are partitioned and on the amount of observation noise present in the system. Following the procedure in Ref. 27, we quantify this background contribution to the mean squared errors of *sep* and *joint* models by dividing each of the systems into two subsystems, and using the same 10-fold cross-validation routine on each pair to identify the difference of the mean squared errors between *sep* and *joint* models that one should expect purely by chance. For example, System 1 is divided into two subsystems and mean squared errors are computed for joint and separate models fitted to these subsystems; the absolute difference in these errors is represented by μ_1 . Similarly, μ_2 refers to the absolute difference in mean squared errors between *sep* and *joint* models built from System 2. In Algorithm 1, $\text{threshold}(A_{\sigma_1, x}, B_{\sigma_2, x}) = \max(\mu_1, \mu_2)$. The hypothesis that σ_1 and σ_2 have the same functional form is rejected if the condition $MSE_{\text{joint}} > MSE_{\text{sep}} + \text{threshold}(A_{\sigma_1, x}, B_{\sigma_2, x})$ holds, and hence we infer $\sigma_1 \neq \sigma_2$. The same calculations are done with the data from Systems 3 and 4 to decide whether $\sigma_3 = \sigma_4$.

In the final step (iv), we reach a decision regarding causal influence. Specifically, if we detect $\sigma_1 = \sigma_2$ and $\sigma_3 \neq \sigma_4$, we identify x as the cause of y . Conversely, if we detect $\sigma_1 \neq \sigma_2$ and $\sigma_3 = \sigma_4$, we identify y as the cause of x . If both pairs are unequal, this suggests mutual causal influence, and if both

Algorithm 1. Comparison of σ_1 with σ_2 and comparison of σ_3 with σ_4

Input: $A_{\sigma_1,x}$ —for **System 1**, an $m \times 2$ matrix where each row contains one element of \vec{x}_α and the corresponding element of \vec{x}_α to which σ_1 maps the system state.
 $A_{\sigma_3,y}$ —for **System 3**, an $m \times 2$ matrix where each row contains one element of \vec{y}_δ and the corresponding element of \vec{y}_δ to which σ_3 maps the system state.
 $B_{\sigma_2,x}$ —for **System 2**, an $m \times 2$ matrix where each row contains one element of \vec{x}_β and the corresponding element of \vec{x}_β to which σ_2 maps the system state.
 $B_{\sigma_4,y}$ —for **System 4**, an $m \times 2$ matrix where each row contains one element of \vec{y}_γ and the corresponding element of \vec{y}_γ to which σ_4 maps the system state.

Output: $Comp_1$ —a boolean value indicating whether σ_1 and σ_2 are identical.
 $Comp_2$ —a boolean value indicating whether σ_3 and σ_4 are identical.

- 1: randomize the rows of $A_{\sigma_1,x}, A_{\sigma_3,y}, B_{\sigma_2,x}, B_{\sigma_4,y}$
- 2: $Partition^1 \leftarrow$ divide the rows of $A_{\sigma_1,x}$ into 10 segments
- 3: $Partition^2 \leftarrow$ divide the rows of $B_{\sigma_2,x}$ into 10 segments
- 4: $Partition^3 \leftarrow$ divide the rows of $A_{\sigma_3,y}$ into 10 segments
- 5: $Partition^4 \leftarrow$ divide the rows of $B_{\sigma_4,y}$ into 10 segments
- 6: **for** $i = 1$ **to** 10 **do**
- 7: **for** $k = 1$ **to** 4 **do**
- 8: $Train^k \leftarrow \bigcup_p \{Partition^k[p] | p \neq i\}$
- 9: **end for**
- 10: **for** $j = 1$ **to** 2 **do**
- 11: $Model^{2j-1} \leftarrow$ FitPolynomial($Train^{2j-1}$)
- 12: $Model^{2j} \leftarrow$ FitPolynomial($Train^{2j}$)
- 13: $Model^{2j-1,2j} \leftarrow$ FitPolynomial($Train^{2j-1} \cup Train^{2j}$)
- 14: $SE^{2j-1} \leftarrow$ concatenate(SE^{2j-1} , squared errors of $Model^{2j-1}$ with respect to $Partition^{2j-1}[i]$)
- 15: $SE^{2j} \leftarrow$ concatenate(SE^{2j} , squared errors of $Model^{2j}$ with respect to $Partition^{2j}[i]$)
- 16: $SE^{2j-1,2j} \leftarrow$ concatenate($SE^{2j-1,2j}$, squared errors of $Model^{2j-1,2j}$ with respect to $Partition^{2j-1}[i] \cup Partition^{2j}[i]$)
- 17: **end for**
- 18: **end for**
- 19: $MSEsep^1 \leftarrow$ mean($SE^1 \cup SE^2$)
- 20: $MSEjoint^1 \leftarrow$ mean($SE^{1,2}$)
- 21: $MSEsep^2 \leftarrow$ mean($SE^3 \cup SE^4$)
- 22: $MSEjoint^2 \leftarrow$ mean($SE^{3,4}$)
- 23: **if** $MSEjoint^1 > MSEsep^1 + \text{threshold}(A_{\sigma_1,x}, B_{\sigma_2,x})$ **then**
- 24: return TRUE and $Comp_1 = 1$
- 25: **else**
- 26: return FALSE and $Comp_1 = 0$
- 27: **end if**
- 28: **if** $MSEjoint^2 > MSEsep^2 + \text{threshold}(A_{\sigma_3,y}, B_{\sigma_4,y})$ **then**
- 29: return TRUE and $Comp_2 = 1$
- 30: **else**
- 31: return FALSE and $Comp_2 = 0$

end if

pairs are identical, then x does not appear to influence y or vice versa.

Note that the algorithm relies on randomizing the rows of $A_{\sigma_1,x}, \dots, B_{\sigma_4,y}$, and hence the decision may vary on different runs using the same data. Thus, for a given time series, we need to run Algorithms 1 and 2 multiple times, and select the majority decision.

IV. OTHER CAUSALITY DETECTING TOOLS: TRANSFER ENTROPY (TE) AND CONVERGENT CROSS MAP (CCM)

In our present study, we choose two model-free methods, namely TE and CCM, to provide a benchmark for the performance of our algorithm in detecting causality. In drawing comparisons, it is important to note that the method introduced here requires four replicate time series data each starting from four different pairs of initial conditions, whereas

both TE and CCM require a single time series data, without any prerequisite on the number of initial conditions needed.

A. Transfer entropy

The idea of TE was first formalized in Ref. 10 to measure information flow between two time series variables. TE uses the concept of Shannon entropy. Shannon entropy quantifies the expected value of the information associated with the occurrence of an event and is defined as follows:

$$H(X) = - \sum_{x \in \mathbb{X}} \Pr[x] \log_2 \Pr[x], \quad (5)$$

where $\Pr[x]$ is the probability mass function for a time series variable X taking the value x and \mathbb{X} refers to the set containing all possible realizations of X . Now, given two time series variables X and Y , the TE from Y to X ($T_{Y \rightarrow X}$), quantifies information transfer from Y to X by measuring the reduction in entropy of X when conditioned on Y . Hence, transfer

Algorithm 2. Decision

Input: $Comp_1$ and $Comp_2$ from Algorithm 1.
Output: Value 1 if $x \rightarrow y$, value 2 if $y \rightarrow x$, value 3 if $y \leftrightarrow x$ and value 0 indicating no causal relationship detected.

- 1: **if** $Comp_1 = 1$ and $Comp_2 = 0$ **then**
- 2: return 2
- 3: **else if** $Comp_1 = 0$ and $Comp_2 = 1$ **then**
- 4: return 1
- 5: **else if** $Comp_1 = 1$ and $Comp_2 = 1$ **then**
- 6: return 3
- 7: **else**
- 8: return 0
- 9: **end if**

entropy from Y to X is defined as

$$T_{Y \rightarrow X} = \sum_{\substack{x(t+1) \in \mathbb{X}(t+1), \\ x(t) \in \mathbb{X}(t), \\ y(t) \in \mathbb{Y}(t)}} \Pr[x(t+1), x(t), y(t)] \\ \times \log \frac{\Pr[x(t+1)|x(t), y(t)]}{\Pr[x(t+1)|x(t)]},$$

where $\Pr[x(t+1)|x(t)]$ and $\Pr[x(t+1)|x(t), y(t)]$ denote the probability of $x(t+1)$ conditioned on $x(t)$ alone, and on both $x(t)$ and $y(t)$, respectively; $\Pr[x(t+1), x(t), y(t)]$ denotes joint probability. If Y does not provide any additional information then $\Pr[x(t+1)|x(t), y(t)] = \Pr[x(t+1)|x(t)]$, and hence $T_{Y \rightarrow X}$ equals zero. In general, transfer entropy is an asymmetric quantity, and comparing $T_{Y \rightarrow X}$ with $T_{X \rightarrow Y}$ detects the dominant direction of information flow, thus identifying the direction of causation.

We implement TE using the open-source JIDT (Java Information Dynamics Toolkit),³⁰ which implements various information-theoretic measures. We used the Kraskov, Stogbauer, and Grassberger (KSG) method to calculate the probability mass functions (PMF) used in the above definitions because it uses dynamically altered kernel width in terms of K nearest-neighbors (k_{nn}) and it has been shown to decrease errors in PMF estimation.³¹

B. Convergent cross map

CCM, first introduced in Ref. 5, is a technique for determining causality between sets of time series using a neighborhood method. It relies on the idea that if time series $X(t)$ causally influences $Y(t)$, then signatures (or imprints) of $X(t)$ inherently exist in $Y(t)$, and hence historical records of $Y(t)$ can be used to estimate the current state of $X(t)$. The estimated state of $X(t)$ is denoted by $\hat{X}(t)|M_Y$, where M_Y refers to a shadow version of the original manifold M reconstructed from the projection of times series $Y(t)$. Finally, the accuracy of that estimation is evaluated using the correlation coefficient, calculated between the original time series and their cross-map estimates. If the correlation coefficient between the true values ($X(t)$) and the estimated values ($\hat{X}(t)|M_Y$) is high, then it is concluded that there is “enough” information stored in Y that came from X to indicate that X is among the causes of Y . However, if $Y(t)$ does not influence $X(t)$, then estimate of $Y(t)$ using $X(t)$, denoted by $\hat{Y}(t)|M_X$,

will be inferior. A more detailed description of the algorithm can be found in Ref. 32, and in the supplementary materials of Ref. 5.

The CCM estimates are unitless and the values range between 0 and 1. The cross map estimation is evaluated at different library sizes, which refers to the length of the time series used. The cross map estimation converges to a constant value with increasing library size, and the comparison of relative skill of cross-map estimations indicates the direction of causation. As the real-world datasets possess observational and process noise and may lack sufficient data, the asymptotic value of cross map estimation may not converge to a value of one.⁵ The parameters associated with CCM are lag, which is used for the construction of the shadow manifold, and the optimal embedding dimension. In the present study, we implement CCM using the rEDM package³³ in R, where the lag was set to 1 by default, and the optimal embedding dimension was evaluated following the Simplex Projection method.³⁴

In real-world applications of TE, the determination of causal direction can be supported by a variety of statistical significance tests.^{21,35} However, for CCM no statistical test has been developed yet.²¹

V. RESULTS

A. Detecting unidirectional coupling in competitive Lotka-Volterra model

To evaluate the performance of our algorithm, we chose to implement a two-species competitive Lotka-Volterra model. Nonlinear models of this type are used to describe the population dynamics of two species competing for resources. The governing equations are given as follows:

$$\begin{aligned} \dot{x} &= r_1 x (1 - (x + \alpha_{12}y)/K_1), \\ \dot{y} &= r_2 y (1 - (y + \alpha_{21}x)/K_2), \end{aligned} \quad (6)$$

where x and y denote the population sizes, r_1 and r_2 the growth rates, and K_1 and K_2 the carrying capacities of species 1 and species 2, respectively. The parameter α_{12} represents the influence of species 2 on species 1, and similarly, α_{21} represents the influence of species 1 on species 2.

For the first set of tests, we set $r_1 = r_2 = 12$, $K_1 = K_2 = 150$, choose $\alpha_{21} = 0$, and vary α_{12} from 1 to 8. By setting $\alpha_{21} = 0$, we ensure that x has no influence on y , while increasing values of α_{12} correspond to an increasing

strength of influence from y to x . Next, we implement our algorithm on the time series data generated by solving Eq. (6) using ode45 in Matlab. To implement our algorithm, we need four pairs of initial conditions for x and y , which we choose as follows: $[x_0 = \gamma = 20, y_0 = \alpha = 30]$, $[x_0 = \delta = 40, y_0 = \alpha = 30]$, $[x_0 = \gamma = 20, y_0 = \beta = 60]$, and $[x_0 = \delta = 40, y_0 = \beta = 60]$. We generate time series data of length 1000. We further add Gaussian observation noise with 0 mean and varying standard deviation from 5 to 10.

We choose different coupling strengths and standard deviations of observation noise to verify the robustness of our algorithm in correctly detecting y as the cause of x . Figure 2(a) demonstrates the results, where we vary the coupling strength (α_{12}) from 1 to 8, and for each choice of α_{12} , we choose six different amounts of observation noise ranging from standard deviation 5 to 10. Since our algorithm relies on shuffling the data while evaluating σ , the results obtained in the decision phase are not deterministic. Hence, we run the algorithm ten times on the same data and count the number of times x is incorrectly detected as the cause of y , the number of times y is correctly detected as the cause of x , and the number of times the algorithm indicates bidirectional causality. We plot the result as percentage in Fig. 2(a). Though some runs on a given data set incorrectly indicate bidirectional causation, we note that the majority decision is accurate for every combination of parameter values tested. Furthermore, in no case does it miss the influence of y on x .

Next, to provide a benchmark for this performance we implement TE and CCM. Since TE and CCM do not require four different choices of initial conditions, we reuse the data set generated above with initial conditions $[x_0 = \gamma = 20, y_0 = \alpha = 30]$. We evaluate TE for k_{nn} values ranging from 2 to 15, where k_{nn} is the parameter used in estimating the PMF. However, the resolution width used in estimating TE should not be too large nor too small,³¹ so we fix k_{nn} to 4. We find in Fig. 2(b) that many times TE falsely identifies x as the sole cause, particularly as the coupling strength α_{12} and measured noise increase. Next, we implement CCM which is presented in Fig. 2(c). CCM relies on the asymptotic value of convergence at maximum library size, and Fig. 2(c) presents the causal direction as identified by CCM. The results demonstrate that CCM also performs poorly, as it too falsely identifies x as a cause of y for a number of values of α_{12} and noise.

1. Bidirectional causality with equal coupling strength

So far, we only considered the presence of unidirectional coupling. Next, we investigate bidirectional coupling, beginning with the special case where the coupling strength for either direction is the same. Again we modeled a competitive Lotka-Volterra system and kept all parameter values the same, but set $\alpha_{12} = \alpha_{21}$. We vary the coupling strength from values of 1 to 8, and for each choice of coupling strength we vary the standard deviation for the Gaussian noise from 5 to 10. Next, we implement our algorithm on the time series data generated by solving Eq. (6) using ode45 in Matlab. We keep the choice of initial conditions fixed as in the earlier example.

TABLE IV. The number of times out of ten runs the algorithm detects x and y as causal drivers. The result is tabulated below in percentage.

α_{12}	1	2	3	4	5	6	7	8	9	10
$\sigma_1 \neq \sigma_2 (y \rightarrow x)$	0	100	100	100	100	100	100	100	100	100
$\sigma_3 \neq \sigma_4 (x \rightarrow y)$	100	100	100	100	100	100	100	100	100	100

Further, we compare the results of our algorithm with that of TE and CCM.

The performance of all three algorithms is presented in Fig. 3(a), where we observe that our algorithm can reliably detect the bidirectional coupling for all choices of coupling strength and increased measured noise. In this case, the output of all 10 runs was uniform for every combination of parameter values. The performance of TE and CCM in detecting the bidirectional coupling is presented in Figs. 3(b) and 3(c), respectively. This case is clearly more challenging for TE and CCM. At least in a minority of cases, TE correctly identifies bidirectional causation, but CCM never does. Note, however, that these methods output continuous measures—of information exchange in the case of TE and degree of correlation for CCM—for which there is no universal decision threshold. It is likely that adjustment of these thresholds would improve performance, but any such threshold would have to be justifiable in the absence of information about the system or the nature of the sampling noise. It is an advantage of the method presented here that a natural threshold is built into the binary decision process that requires no foreknowledge of the system to determine.

For reference, we have provided sample trajectories for both unidirectional and bidirectional case in Fig. 4.

2. Bidirectional causality with unequal coupling strength

Finally, we investigate the general case of bidirectional causality, when both the variables influence each other but do not necessarily do so with equal coupling strength. We again choose the competitive Lotka-Volterra equations and fix the coupling strength from x to y by setting $\alpha_{21} = 10$, and the standard deviation for the Gaussian noise to 5. Next, we vary the coupling strength from y to x from values 1 to 10, where small values of α_{12} correspond to weak coupling. To measure the performance, we run our algorithm 10 times for each choice of α_{12} (again, on the identical data set) and count the number of times x and y are identified as a causal driver, independently. Bidirectional coupling will be inferred if the algorithm detects both x and y as causal drivers. We present the result as percentage in Table IV. We notice that the algorithm detects the presence of bidirectional coupling for all choices of α_{12} , except when $\alpha_{12} = 1$. In other words, when the value of α_{12} is relatively small in comparison to α_{21} the algorithm correctly infers x as a cause of y , but misses the relatively weak influence of y on x .

B. Accuracy

So far, we have found that the proposed algorithm performs better for a single noisy dataset in comparison to both

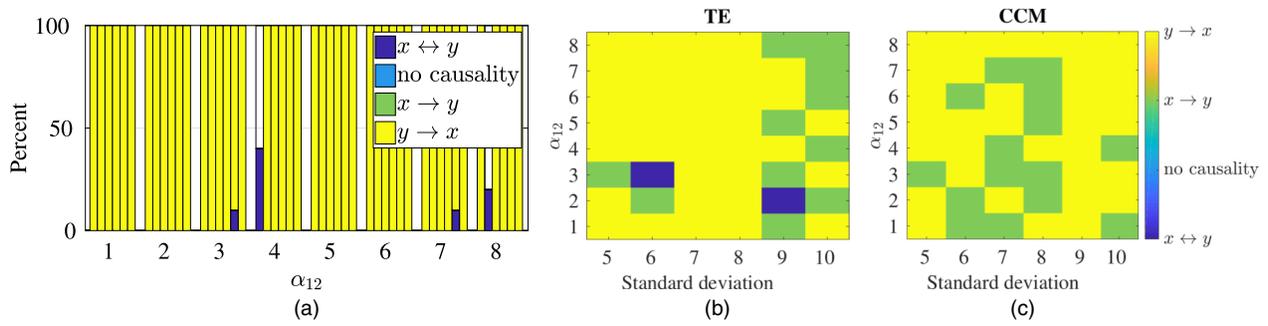


FIG. 2. (a) Results demonstrating performance of our algorithm in detecting unidirectional causality. We set $\alpha_{21} = 0$, and varied the coupling from y to x , by varying α_{12} from 1 to 8. For each value of α_{12} , the six bars correspond to increasing Gaussian measurement noise with mean 0 and standard deviation ranging from 5 to 10 in increments of 1. For each choice of α_{12} , and each choice of standard deviation we run our algorithm ten times on single sample and count the number of times y is correctly detected as the sole cause (colored in yellow), the number of times x is falsely detected as the sole cause (colored in green), the number of times algorithm does not detect a causal relation (colored in light blue), and the number of times the algorithm detects bidirectional causation (colored in deep blue). The results are plotted as percentage. Interestingly, the algorithm never misses the influence of y on x , but at times additionally detects influence of x on y inferring bidirectional coupling. (b) Results demonstrating the performance of TE in detecting y as the cause of x . (c) Results demonstrating the performance of CCM in detecting y as the cause of x . Results show that at times both TE and CCM fail to identify y as the sole cause.

TE and CCM. As noted above, the algorithm is indeterministic for a given dataset; the output may depend upon the shuffling that takes place in cross-validation. As suggested above, this can be mitigated when using the algorithm to decide causal structure by running multiple times on a given dataset and then taking a consensus of the results. For characterizing the accuracy of such a procedure, we accept the majority output as definitive and test against a sample of datasets drawn from the same Lotka-Volterra system. We repeat these tests for a range of systems and sampling noises. Specifically, we vary the standard deviation of the Gaussian sampling noise from 5 to 50. For a fixed value of noise, we vary the coupling strength α_{12} from 1 to 8. We set α_{21} to zero for the unidirectional case and vary both α_{12} and α_{21} for the bidirectional case, where α_{21} is set equal to α_{12} . For each set of system and noise parameters, we conduct ten trials, generating a distinct time series for each. Within each trial, we run our algorithm ten times and identify the causal direction by majority vote. We define the accuracy as the ratio of the number of times the algorithm correctly identifies all and only the correct causal edges to the total number of trials, which is equal to ten. We present the

accuracy results in Fig. 5. We find that the accuracy decays with an increase in the noise, however, the effect of noise on performance reduces when the coupling strength is also increased. In other words, for relatively weak coupling the noise dominates the performance, but the performance gets better if coupling strength is increased proportionally. This pattern is prominent in the bidirectional case [Fig. 5(b)] and weakly present in the unidirectional case Fig. 5(a).

C. Detecting bidirectional coupling in a Dixon system

In this section, we evaluate the performance of our algorithm by implementing it on a system that demonstrates chaos-like behavior. According to the Poincaré-Bendixson theorem, the attractor for any smooth two-dimensional bounded continuous-time autonomous system is either a stable equilibrium or a limit cycle.³⁶ Since our algorithm is limited to first-order two-dimensional continuous-time autonomous systems, we cannot apply it to most of the well-known chaotic systems. However, there exists a system derived by Dixon *et al.* which does not satisfy the smoothness

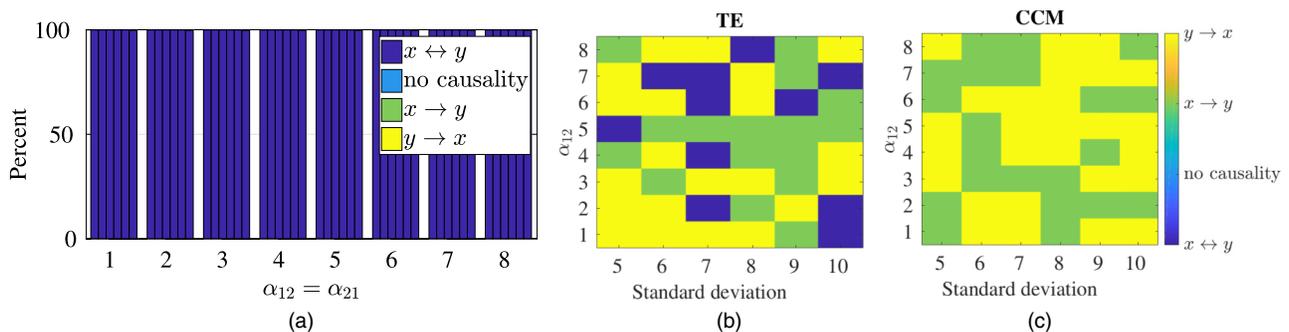


FIG. 3. (a) Results demonstrating the performance of our algorithm in detecting bidirectional causality. We varied α_{12} from 1 to 8, and set $\alpha_{21} = \alpha_{12}$. For each value of α_{12} , the six bars correspond to increasing Gaussian measurement noise with mean 0 and standard deviation ranging from 5 to 10 in increments of 1. For each choice of coupling strength, and each choice of standard deviation, we run our algorithm ten times on the same sample and count the number of times y is detected as the sole cause (colored in yellow), the number of times x is detected as the sole cause (colored in green), the number of times algorithm does not detect a causal relation (colored in light blue), and the number of times bidirectional causality is detected (colored in deep blue). The results are plotted as percentage. The algorithm uniformly detects bidirectional causality for the entire range of coupling strength and noise. (b) Results demonstrating the performance of TE in detecting bidirectional causality. (c) Results demonstrating the performance of CCM in detecting bidirectional causality. Results show that TE misses one or the other direction of causal influence, while CCM never detects bidirectional causality.

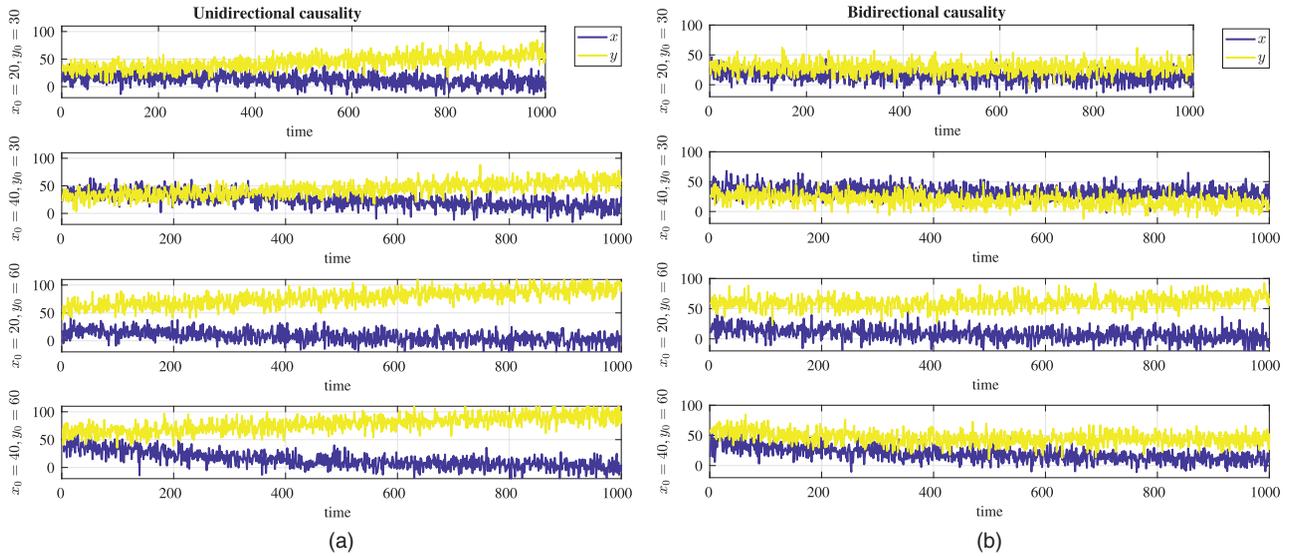


FIG. 4. Sample trajectories of the Lotka-Volterra model used in the analysis. (a) Unidirectional case, with $\alpha_{12} = 8$, and the standard deviation of the Gaussian noise is 10. (b) Bidirectional case, with $\alpha_{12} = \alpha_{21} = 8$, and the standard deviation of the Gaussian noise is 10.

condition since it has a singularity at the origin. The governing equations are given as follows:

$$\begin{aligned} \dot{x} &= \frac{xy}{x^2 + y^2} - \xi_1 x, \\ \dot{y} &= \frac{y^2}{x^2 + y^2} - \xi_2 y + \xi_2 - 1. \end{aligned} \quad (7)$$

We choose the parameters $\xi_1 = 0$ and $\xi_2 = 0.7$, for which the system displays “chaotic” behavior, which is more precisely referred to as “S-chaos” (singularity-chaos).^{36,37} Next, we implement our algorithm on the time series data generated by solving Eq. (7) using ode45 in Matlab. We choose four pairs of initial conditions for x and y as follows: $[x_0 = \gamma = 0.5, y_0 = \alpha = 1.5]$, $[x_0 = \delta = 0.15, y_0 = \alpha = 1.5]$, $[x_0 = \gamma = 0.5, y_0 = \beta = 2.5]$, and $[x_0 = \delta = 0.15, y_0 = \beta = 2.5]$. We further add Gaussian observation noise with 0 mean and 0.1 standard deviation. Next, we implemented our algorithm, and ran it ten times and noted the dominant causal direction that it detected. Interestingly, the algorithm detected the correct causal direction in each run, indicating bidirectional coupling.

D. Real-world system

To evaluate the performance of our algorithm with respect to a real-world system, we built a nonlinear circuit in which an LC oscillator is used to drive an inverting amplifier. The detailed circuit diagram is provided in Fig. 6(a). We treated the output of the LC oscillator and the output of the inverting amplifier as our two dynamical variables. The signals recorded from these two channels are referred to as X and Y in Fig. 6(a). The input voltage VCC was used to intervene in the system in order to attain the desired initial conditions. We chose four pairs of initial conditions for X and Y as follows: $[X_0 = \gamma = -0.58 \text{ V}, Y_0 = \alpha = 2.6 \text{ V}]$, $[X_0 = \delta = -0.48 \text{ V}, Y_0 = \alpha = 2.6 \text{ V}]$, $[X_0 = \gamma = -0.58 \text{ V}, Y_0 = \beta = 1.94 \text{ V}]$, and $[X_0 = \delta = -0.48 \text{ V}, Y_0 = \beta = 1.94 \text{ V}]$. We recorded a sample of size 2496 at an interval $4\text{E} \times 10^{-8}$ s. In this experimental setup, X drives Y . The signals are plotted in Fig. 6(b) for reference. Next, we implemented our algorithm and ran it ten times and noted the dominant causal direction that it identified. Interestingly, the algorithm detected the correct causal direction in each run, identifying

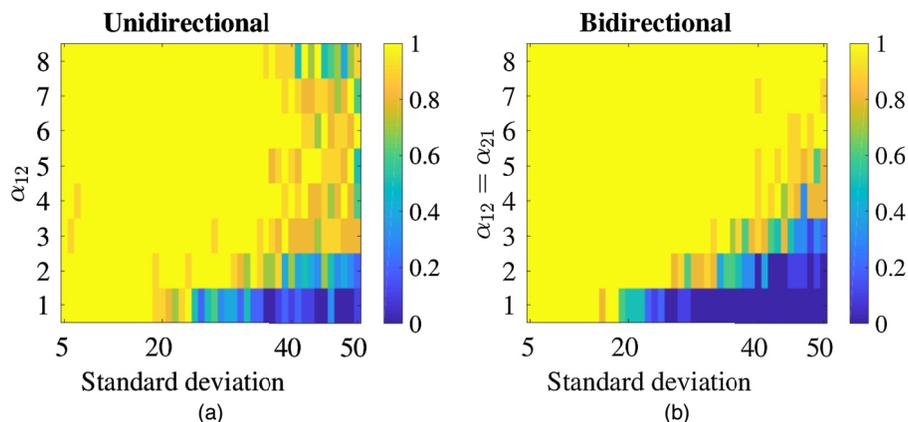


FIG. 5. The performance of the algorithm is measured in terms of accuracy (with a range of 0 to 1 as indicated by color) for both (a) unidirectional and (b) bidirectional case.

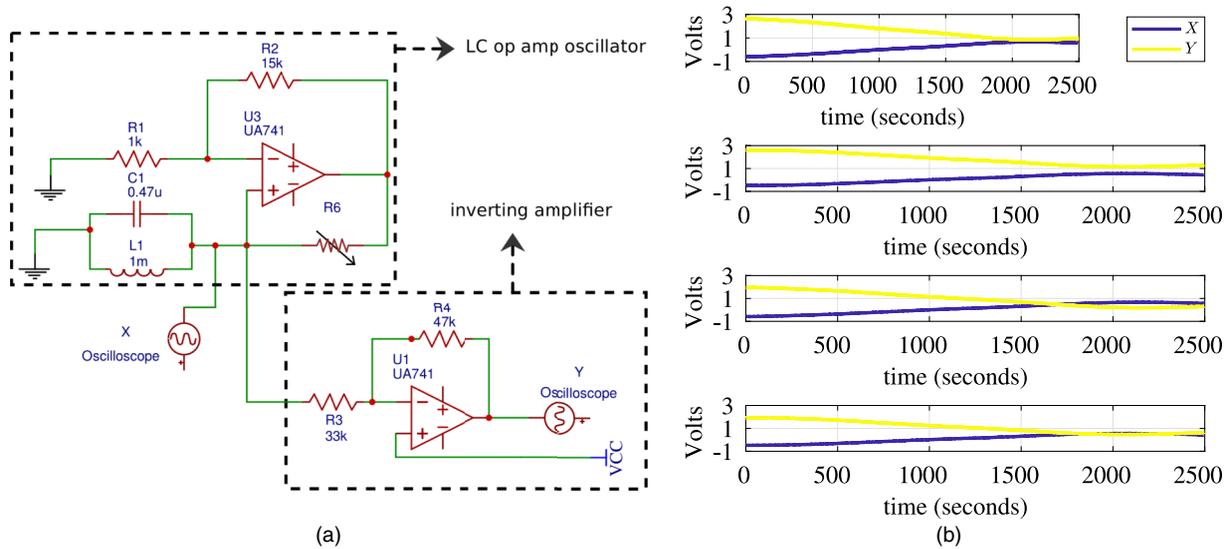


FIG. 6. (a) Circuit diagram used for generating the real-world data. LC op amp oscillator is used to control the inverting amplifier. (b) Two signals, namely X and Y , that are recorded simultaneously at four different initial conditions.

X as the cause. We further implemented TE and CCM, to compare the performance with respect to our tool. TE correctly detected the coupling direction from $X \rightarrow Y$, whereas CCM incorrectly inferred bidirectional coupling.

VI. DISCUSSION AND FUTURE SCOPE

In the present work, we use the concept of pseudo-symmetries to detect unidirectional and bidirectional causality. We compare the performance of our algorithm to that of TE and CCM when applied to noisily sampled two-species competitive Lotka-Volterra systems. In this context, both TE and CCM can in principle detect the causal influences. However, tests demonstrate that both these tools fail often in the presence of significant noise, particularly with bidirectional coupling. On the other hand, the proposed algorithm based on comparisons of pseudo-symmetries is accurate across all of the conditions examined. To demonstrate the application of the present approach to real-world systems, we constructed a circuit in which one voltage drives another. The algorithm accurately identified the correct causal direction. The symmetry approach is therefore a viable alternative to existing tools when dealing with noisy real-world data that may exhibit any combination of causal influences between dynamical variables.

The present approach relies upon observing behavior for four pairs of distinct initial conditions. The simplest way to achieve this is by intervention on the system. But, given the possibility of intervention, an alternate approach to identifying causal direction suggests itself, namely a direct comparison of trajectories. For example, comparing \vec{x}_α with \vec{x}_β and \vec{x}_α with \vec{x}_β as in Figs. 1(a) and 1(b) would enable one to detect that y drives x . The method we use to compare portions of pseudo-symmetries (that can be represented by polynomials) could be used to directly compare trajectories. Differences in, e.g., $x(t)$ for changes in the initial value of y would then be indicative of causal direction (in this case, from y to x). However, that method fails when we do not have

exact control over initial conditions. To test this assertion, we compared the present method with that of such a trajectory comparison method by adding Gaussian noise to the initial conditions, and then letting a Lotka-Volterra system evolve in time. Under these conditions, the symmetry method performs better in detecting the causal direction than a direct comparison of trajectories. Further, we consider a chaotic Dixon system, to verify the performance with respect to noisy initial conditions. The symmetry method detected the correct causal direction. Although it is simpler to intervene, it is often not necessary. For a sufficiently long time series, a purely observational dataset can often be segmented in such a way that, when each segment is treated as a separate time series, the requisite set of initial conditions are at least approximately obtained. This is sufficient for applying the algorithm, albeit at some cost to accuracy.

This first version of the algorithm is, however, limited to first-order autonomous systems. It is also the case that the present version of the algorithm is applicable only to systems having two time-dependent variables. The most straightforward approach to extending this method (or the direct comparison of trajectories mentioned above) to systems with more variables would require a geometric increase in the number of initial conditions as a function of the number of variables. We see two approaches to generalizing our method that avoids this problem. On the one hand, it is possible to assess directional gradients of the surfaces that correspond to components of a dynamical symmetry in ways that do not involve comparing pseudo-symmetries. If this can be done with accuracy from typical time series data, then in principle we would not be limited by the large number of initial conditions demanded by a straightforward generalization of the current approach. On the other hand, it is possible to preserve causal information while reducing many variable systems to approximate two-variable systems.³⁸ Both approaches are left for future work.

Finally, our algorithm, though it reliably detects bidirectional influence, yields only a binary decision; it does not

indicate the relative degree of causal influence in either direction. But such information can be gleaned after a reliable detection is made by applying one or another continuous measure of the degree of causal influence.

Despite its limits, the algorithm presented is a demonstration of the potential of symmetry methods for the reliable detection of causality when data is noisy, causation may be bidirectional, and coupling strengths vary. This work only begins to open a wide scope for improvement, which is left for future studies.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant No. 1454190. We would like to thank Naveed Jantzen for his assistance in collecting data while conducting the experiment.

- ¹J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Scholkopf, *J. Mach. Learn. Res.* **17**, 1103 (2016).
- ²J. Hlinka, D. Hartman, M. Vejmelka, J. Runge, N. Marwan, J. Kurths, and M. Palus, *Entropy* **15**, 2023 (2013).
- ³Y. Hirata, J. M. Amigo, Y. Matsuzaka, R. Yokota, H. Mushiake, and K. Aihara, *PLoS one* **11**, e0158572 (2016).
- ⁴C. Granger, *Econometrica* **37**, 424 (1969).
- ⁵G. Sugihara, R. May, H. Ye, C. Hsieh, E. Deyle, M. Fogarty, and S. Munch, *Science* **338**, 496 (2012).
- ⁶H. Ma, K. Aihara, and L. Chen, *Sci. Rep.* **4**, 7464 (2014).
- ⁷M. Eichler, *Causality: Statistical Perspectives and Applications* (Wiley Online Library, 2012), p. 327
- ⁸S. Butail, V. Mwaffo, and M. Porfiri, *Phys. Rev. E* **93**, 042411 (2016).
- ⁹S. Roy and N. Abaid, *R. Soc. Open. Sci.* **4**, 170130 (2017).
- ¹⁰T. Schreiber, *Phys. Rev. Lett.* **85**, 461 (2000).
- ¹¹Y. Hirata and K. Aihara, *Phys. Rev. E* **81**, 016203 (2010).
- ¹²P. Kim, J. Rogers, J. Sun, and E. Bollt, *J. Comput. Nonlinear. Dyn.* **12**, 011008 (2017).
- ¹³M. G. Rosenblum, L. Cimponeriu, A. Bezerianos, A. Patzak, and R. Mrowka, *Phys. Rev. E* **65**, 041909 (2002).
- ¹⁴M. G. Rosenblum and A. S. Pikovsky, *Phys. Rev. E* **64**, 045202 (2001).
- ¹⁵R. Vicente, M. Wibral, M. Lindner, and G. Pipa, *J. Comput. Neurosci.* **30**, 45 (2011).
- ¹⁶T. Dimpfl and F. J. Peter, *Stud. Nonlin. Dyn. Econom.* **17**, 85 (2013).
- ¹⁷G. Ver Steeg and A. Galstyan, in *Proceedings of the 21st International Conference on World Wide Web* (ACM, 2012), pp. 509–518.
- ¹⁸S. Butail, F. Ladu, D. Spinello, and M. Porfiri, *Entropy* **16**, 1315 (2014).
- ¹⁹N. Orange and N. Abaid, *Eur. Phys. J. Special Topics* **224**, 3279 (2015).
- ²⁰C. Grabow, J. Macinko, D. Silver, and M. Porfiri, *Chaos: Interdiscip. J. Nonlin. Sci.* **26**, 083113 (2016).
- ²¹D. Coufal, J. Jakubik, N. Jajcay, J. Hlinka, A. Krakovska, and M. Palus, *Chaos: Interdiscip. J. Nonlin. Sci.* **27**, 083109 (2017).
- ²²J. C. McBride, X. Zhao, N. B. Munro, G. A. Jicha, F. A. Schmitt, R. J. Kryscio, C. D. Smith, and Y. Jiang, *NeuroImage: Clinical* **7**, 258 (2015).
- ²³F. Dost, *J. Retail. Consum. Serv.* **23**, 49 (2015).
- ²⁴M. C. Maher and R. D. Hernandez, *PeerJ* **3**, e824 (2015).
- ²⁵C. Beauchene, A. Leonessa, S. Roy, J. Simon, and N. Abaid, in *ASME 2017 Dynamic Systems and Control Conference* (American Society of Mechanical Engineers, 2017), pp. V001T37A001–V001T37A001.
- ²⁶D. Monster, R. Fusaroli, K. Tylan, A. Roepstorff, and J. F. Sherson, arXiv preprint [arXiv:1603.01155](https://arxiv.org/abs/1603.01155) (2016).
- ²⁷B. C. Jantzen, in *Proceedings of the UAI 2016 Workshop on Causation: Foundation to Application* (2017), available at <http://ceur-ws.org/Vol-1792/paper2.pdf>.
- ²⁸B. C. Jantzen, *Synthese* **192**, 3617 (2015).
- ²⁹J. Pearl, *Causality: Models, Reasoning and Inference* (Cambridge University Press, 2009), p. 478.
- ³⁰J. T. Lizier, arXiv preprint [arXiv:1408.3270](https://arxiv.org/abs/1408.3270) (2014).
- ³¹A. Kraskov, H. Stogbauer, and P. Grassberger, *Phys. Rev. E* **69**, 066138 (2004).
- ³²J. M. McCracken and R. S. Weigel, *Phys. Rev. E* **90**, 062903 (2014).
- ³³“Sugihara lab- resources (rEDM),” <http://deepeco.ucsd.edu/resources/>.
- ³⁴E. R. Deyle, M. Fogarty, C.-h. Hsieh, L. Kaufman, A. D. MacCall, S. B. Munch, C. T. Perretti, H. Ye, and G. Sugihara, *Proc. Natl. Acad. Sci.* **110**, 6430 (2013).
- ³⁵R. Vicente, M. Wibral, M. Lindner, and G. Pipa, *J. Comput. Neurosci.* **30**, 45 (2011).
- ³⁶J. C. Sprott, *Elegant Chaos: Algebraically Simple Chaotic Flows* (World Scientific, 2010).
- ³⁷D. Dixon, F. Cummings, and P. Kaus, *Phys. D* **65**, 109 (1993).
- ³⁸K. Chalupka, F. Eberhardt, and P. Perona, “Multi-level cause-effect systems,” in *Artificial Intelligence and Statistics* (2016), pp. 361–369.