Persistent homology of coarse-grained state-space networks

Audun D. Myers, Max M. Chumley, Firas A. Khasawneh, and Elizabeth Munch

Phys. Rev. E **107**, 034303 — Published 2 March 2023

DOI: 10.1103/PhysRevE.107.034303

Persistent Homology of Coarse Grained State Space Networks

Audun D. Myers,[1, a)] Max M. Chumley,[1, b)] Firas A. Khasawneh,[1, c)] and Elizabeth Munch[2, d)]
[1)] *Department of Mechanical Engineering, Michigan State University, East Lansing, MI.*
[2)] *Dept. of Computation Mathematics Science and Engineering; and Dept. of Mathematics, Michigan State University, East Lansing, MI*

(Dated: 15 February 2023)

This work is dedicated to the topological analysis of complex transitional networks for dynamic state detection. Transitional networks are formed from time series data and they leverage graph theory tools to reveal information about the underlying dynamic system. However, traditional tools can fail to summarize the complex topology present in such graphs. In this work, we leverage persistent homology from topological data analysis to study the structure of these networks. We contrast dynamic state detection from time series using CGSSN and TDA to two state of the art approaches: Ordinal Partition Networks (OPNs) combined with TDA, and the standard application of persistent homology to the time-delay embedding of the signal. We show that the CGSSN captures rich information about the dynamic state of the underlying dynamical system as evidenced by a significant improvement in dynamic state detection and noise robustness in comparison to OPNs. We also show that because the computational time of CGSSN is not linearly dependent on the signal's length, it is more computationally efficient than applying TDA to the time-delay embedding of the time series.

Keywords: Topological Data Analysis, Complex Networks, Coarse Grained, Persistent Homology

I. INTRODUCTION

Signal processing has been successfully and widely utilized to extract meaningful information from time series of dynamical systems including dynamic state detection[1–6], structural health monitoring for damage detection[7–11], and biological health monitoring[12–18]. A promising direction for signal processing is through studying the shape of signals. This is done by implementing tools from Topological Data Analysis (TDA)[19,20] to study the shape of the attractor of the underlying dynamical system. This field of signal processing is known as Topological Signal Processing (TSP)[21], which has had many successful applications, including biological signal processing[22,23], dynamic state detection[24,25], manufacturing[26–30], financial data analysis[31–34], video processing[35,36], bifurcation detection[37], and weather analysis[38,39].

The standard pipeline for TSP constructs a filtration of simplicial complexes (called the Vietoris-Rips complex) based on point cloud data generated from the State Space Reconstruction (SSR) of an input time series[29,40–42]. Given a uniformly sampled signal $x = [x_1, x_2, \ldots, x_L]$, the SSR (also called the delay embedding) consists of n-dimensional delayed vectors

$$\mathbf{X} = \{v_i = [x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+\tau(n-1)}] \mid i \in \{1, \cdots, L - \tau(n-1)\}\}. \tag{1}$$

A simplicial complex is formed by including simplices for all collections of points which are within distance r of each other. We can measure the shape of the simplicial complex by forming simplicial complexes at increasing

---

[a)]Electronic mail: myersau3@msu.edu; audunmyers.com
[b)]Electronic mail: chumleym@msu.edu; maxchumley.com
[c)]Electronic mail: khasawn3@egr.msu.edu; firaskhasawneh.com; Corresponding Author
[d)]Electronic mail: muncheli@msu.edu; elizabethmunch.com

values of r, and tracking the changing homology through a linear mapping. This allows for quantifying when specific topologies form and disappear throughout the filtration giving a sense of shape. The persistence diagram encodes this information for various dimensions, e.g., connected components (dimension zero), loops (dimension one), voids (dimension two). For example, one can examine the one dimensional homology to track loop structures in the SSR that are related to the periodicity of the signal. A problem with this pipeline is its computational demand having complexity $O(N^3)$, where $N = \binom{n}{d+1}$ is the size of the simplicial complex with n as the number of points in the simplical complex and d as the maximum dimension of the used homology. For long signals, this makes this standard pipeline computationally infeasible. A common solution is to subsample the point cloud, but it can be challenging to select an appropriate subsampling rate that preserves the topology of interest.

An alternative, promising direction for signal processing is analyzing time series via representations as complex networks[43–45]. Network representations of time series generally fall within three categories: proximity networks, visibility graphs, and transitional networks. Proximity networks are formed from closeness conditions in the reconstructed state space. Examples include the k-Nearest Neighbors (k-NN)[46] and recurrence networks[47], where the recurrence network is the network underlying the Vietoris-Rips complex of the point cloud data. For proximity networks, the graph representation includes all points in the state space reconstruction as part of the vertex set and does not reduce the computational complexity. Additionally, these networks require choosing a proximity parameter dependent on the signal, where careful consideration is needed in selecting the number of neighbors k or proximity distance ε to generate a graph that captures the correct topology. The visibility graphs[48] are formed by adding vertices for each data point and adding connecting edges if a visibility line can be drawn between the two vertices which do not pass below any other data point between the two. As our focus in this work is on building upon the strong theory developed for the SSR embedding, we will not utilize the visibility graph constructions at this stage. Instead, in this work we focus on transitional networks.

Transitional networks partition a time series x such that it has a vertex set of states $\{s_i\}$ for each visited state and an edge for temporal transitions between states. The resulting transitional network constitutes a finite state space $\mathcal{A}$ as the alphabet of possible states. One interpretation of a topological system on a finite state space is as a finite graph where the edges describe the action of a function φ, i.e., if there is a directed edge from vertex a to vertex b, then $\varphi(a) = b$. Therefore, the transitional networks we obtain from a time series are topological systems, and they yield themselves to further analysis within the framework of topological dynamics. The two most common transitional networks for time series analysis are the Ordinal Partition Network (OPN)[49] and the Coarse Grained State Space Network (CGSSN)[50–53]. In Fig. 1 we demonstrate the rich topological structure of the CGSSN for periodic and chaotic dynamics from the Rossler system. This example shows the periodic dynamics corresponding to an approximate cycle graph while the network of the chaotic signal is highly intertwined.

To date, the majority of evaluation of these complex network representations is through standard graph theory tools[45,49,54,55], but the results can only provide local structural measurements based on the node degree distribution or shortest path measurements. In our previous work[25], we studied the global shape of these networks using persistent homology for dynamic state detection using the ordinal partition network. However, we only used the shortest unweighted path to define distances between nodes, which discarded edge weight and direction information. In our recent work[56] we investigate the use of weighted edge information based on the number of edge transitions. We found that this improved dynamic state detection performance.

However, we show here that there is an issue with the OPN; namely, amplitude information is discarded because the ordinal partition network is built from permutations. Permutations can be thought of as partitioning the state space via intersections of hyperplanes of the form $x_i \leq x_j$. As such, the resulting OPN can have reduced dynamic state detection performance and extreme sensitivity to additive noise for some signals. This can be partially explained by noting that proximity of the trajectory to the hyperdiagonal can cause failures in network construction, particularly when there is noise in the signal (details of this issue are

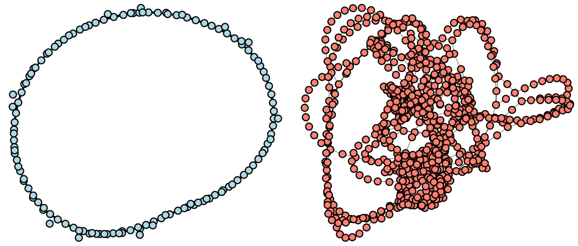


FIG. 1. Example periodic and chaotic CGSSNs generated from the $x(t)$ solution to the Rössler system.

provided in Section IV D). Further, due to the hyperdiagonal intersection issue, we cannot guarantee the stability of the persistence diagram for all signals. Therefore, we turn our attention to the CGSSN to bypass the limitations in OPN.

We investigate the applicability of the CGSSN for enhanced noise robustness and dynamic state detection compared to the OPN. The results presented are based on analyzing the complex networks using persistent homology and tools from information theory and machine learning. Our results show an improvement in dynamic state detection performance with 100% separation between periodic from chaotic dynamics for noise-free signals using a nonlinear support vector machine compared to at most 95% for the OPN. Additionally, we show an improved noise robustness with the CGSSN functioning down to a signal-to-noise ratio of 22 dB compared to 29 dB for the OPN.

Organization

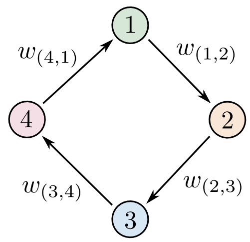
In Section II we overview the necessary background information. We begin with an introduction to the two transitional networks we study—OPN and CGSSN—and an overview of how they are related to state space reconstruction. Next, we introduce four standard methods for measuring the distance between nodes in a weighted graph. We subsequently describe persistent homology and how it is applied to study the shape of the weighted complex networks. In Section III, we demonstrate how to apply our pipeline for studying the shape of complex transitional networks for a simple periodic example. In Section IV, we show results for studying the persistent homology of both the OPN and CGSSN. We begin with results for dynamic state detection for the Lorenz system with a periodic and chaotic response. We then apply the method to 23 continuous dynamical systems, and utilize machine learning to quantify the dynamic state detection performance over a broad range of signals. Lastly, we show results on the noise robustness of the CGSSN in comparison to the OPN. In Section V, we provide conclusions future work on applying persistent homology to study the structure of transitional networks.

II. BACKGROUND

A. Transitional Complex Networks

A graph $G = (V, E)$ is a collection of vertices V and edges $E = (u, v) \subseteq V \times V$. We assume all graphs are simple (no self-loops or hypergraphs) and undirected. Additional stored information comes as a weighted

$$S = [s_1, s_2, s_3, \ldots, s_{L-\tau(n-1)}]$$
$$= [1, 2, 3, 4, 1, 2, 3, \ldots]$$



$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & w_{(1,2)} & 0 & 0 \\ 0 & 0 & w_{(2,3)} & 0 \\ 0 & 0 & 0 & w_{(3,4)} \\ w_{(4,1)} & 0 & 0 & 0 \end{bmatrix} \end{array}$$

FIG. 2. Example formation of a weighted transitional network as a graph (middle figure) and adjacency matrix (right figure) given a state sequence S (left figure).

graph, $G = (V, E, \omega)$ where $\omega : E \to \mathbb{R}_{\geq 0}$ gives a non-negative weight for each edge in the graph. Given an ordering of the vertices $V = \{v_1, \cdots, v_n\}$, a graph can be stored in an adjacency matrix $\mathbf{A}$ where the weighting information is stored by setting $\mathbf{A}_{ij} = w_{(v_i,v_j)}$ if $(v_i, v_j) \in E$ and 1 otherwise.

Transitional networks are graphs formed from a chronologically ordered sequence of symbols or states derived from the time series data. In our construction, these states are mapped from the measurement signal by first creating an SSR $\mathbf{X}$ from Eq. (1) and then assigning a symbolic representation for each vector $v_i \in \mathbf{X}$. To form a symbolic sequence from the time series data, we implement a function to map the SSR to symbol in the alphabet $\mathcal{A}$ of possible states as $f : v_i \to s_j$, where $s_j \in \mathcal{A}$ is a symbol from the alphabet. In this work, we consider the symbols from the alphabet as integers such that $s_i \in \mathbb{Z} \cap [1, N]$, where N is the number of possible symbols. Applying this mapping over all embedding vectors we get a symbol sequence as $S = [s_1, s_2, \ldots, s_{L-\tau(n-1)}]$. This work investigates two methods for mapping SSR vectors v_i to symbols s_j. The first is the OPN which is defined in Section II A 1 and is based on permutations. The second method is the CGSSN defined in section II A 2 which uses an equal-sized hypercube tessellation.

The symbol sequence S forms a transitional network by considering a graph $G = (V, E)$, where the vertices V are the collection of the used symbols, and the edges are added based on transitions between symbols in S. We represent the graph using the adjacency matrix $\mathbf{A}$ data structure of size $N \times N$. We add edges to the adjacency matrix $\mathbf{A}$ via the symbolic transitions with an edge between row s_i and column s_{i+1} for each i. This is represented in the adjacency matrix structure by incrementing the value of $\mathbf{A}_{s_i,s_j}$ by one for each transition between s_i and s_{s+1}, where $\mathbf{A}$ begins as a zero matrix. We set the total number of transitions between two nodes as the edge weight $w_{(s_i,s_j)}$. We ignore self-loops by setting the diagonal of $\mathbf{A}$ to zero. To better illustrate the transitional network formation process, consider the simple cycle shown in Fig. 2. In this example, we take the state sequence S on the left side of Fig. 2 with symbols in the alphabet $\mathcal{A} = [1, 2, 3, 4]$ and create the network shown network in the middle of the figure. This network is represented as a directed and weighted adjacency matrix, as shown on the right side of Fig. 2. In this paper, we discard the directionality information and make $\mathbf{A}$ symmetric by adding its transpose, $\mathbf{A} + \mathbf{A}^T$.

1. Ordinal Partition Network

To form an OPN, the SSR $\mathbf{X}$ must first be constructed requiring the choice of two parameters: the delay τ and dimension n. We select the delay τ using the method of multi-scale permutation entropy[57,58] and the dimension as $n = 7$ as suggested for permutation entropy[58]. For the OPN, the vector v_i is assigned to a permutation π based on its ordinal partition. For dimension n there are $n!$ permutations (e.g., 6 possible permutations for dimension $n = 3$ shown in Fig. 3) which can order arbitrarily $\pi_1, \cdots \pi_{n!}$. Then v_i is assigned to a permutation π_k following that π_k satisfies $v_i(\pi_k(0)) \leq v_i(\pi_k(1)) \leq \ldots \leq v_i(\pi_k(n-1))$. An example of this for the vector $v_i = [-0.08, 0.48, -0.34]$ is shown on the top Ordinal Partition (OP) route of Fig. 3 where

v_i is mapped to permutation π_5 and state $s_i = 5$.

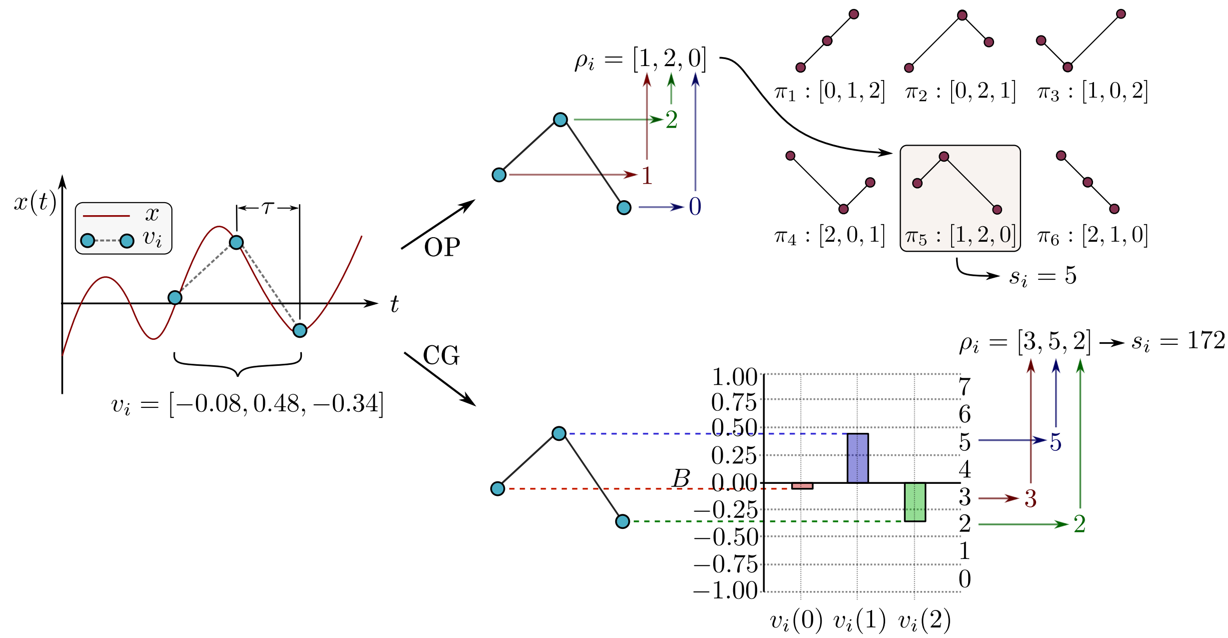


FIG. 3. Example state assignment using the Ordinal Partition (OP) method (top) and Coarse Graining (CG) method (bottom). The state for the OP method is based on the assigned permutation number with $s_i = 5$ for the example. The state assignment for the CG method is based on the number of bins where $s_i = 1 + \sum_{j=0}^{n-1} \rho_i(j) b^j$, ρ_i is the digitization of vector v_i based on binning into b equal-sized bins spanning $[\min(x), \max(x)]$. For this example, $s_i = 3(8^0) + 5(8^1) + 2(8^2) + 1 = 172$ with $b = 8$ bins.

2. Coarse Grained State Space Network

The CGSSN begins by constructing the SSR, where we select the delay τ using the multi-scale permutation entropy method[57,58] and dimension n using the false nearest neighbors[59] based on only needing a dimension great enough for periodic orbits to not self-intersect. For the CGSSN, the vector $v_i \in \mathbf{X}$ is assigned to a state based on which partitioned region the vector v_i lies within. We define the domain $\mathbb{D}$ of the SSR as the non-empty connected, open set that encloses all vectors of the SSR. Specifically, we use an n-dimensional hypercube domain bounded by the intervals $[\min(x), \max(x)]$ for each dimension. In this work we cover this domain using a tessellation of $N = b^n$ hypercubes with side length $(\max(x) - \min(x))/b$, where b is the number of bins per dimension. We assign each n-dimensional hypercube in the tessellation a unique symbol by converting it to a decimal representation denoted as s_i. An introductory example formation of the entire CGSSN for a sinusoidal function is provided in Section III. Some generalizations exist to the described method where instead of assigning symbols to the individual hypercubes, we could assign words of length m which would allow for studying a sequence of coarse grained states of the system which reduces the information load in the process[60]. For the purpose of this paper, a symbolic representation was sufficient.

B. Vertex similarity and dissimilarity measures

To study the structure of the complex network we define functions of the form $V \times V \to \mathbb{R}_{\geq 0}$ combining information about path lengths and weights from the graph in various ways. Some of these definitions are distances, but not all. Despite this, the framework can still be used to define a filtered simplicial complex in the spirit of the Vietoris Rips complex which will be required in the next section.

The measures are encoded in a matrix $\mathbf{D}$, where $\mathbf{D}(a, b)$ is the similarity or dissimilarity between vertices a and b. Note that $\mathbf{D}$ can optionally be normalized by dividing all entries by its maximum value to contain values between 0 and 1. We investigated the use of four choices of measures: the unweighted shortest path distance, the shortest weighted path dissimilarity, the weighted shortest path distance, and the diffusion distance.

1. Shortest Path Distances and Dissimilarities

Commonly used in graph theory, the *shortest path distance* is based on minimizing the cost of taking a path from node a to b. This assumes a path $P = [n_0, n_1, \ldots, n_s]$ consisting of s nodes where $a = n_0$ and $b = n_s$ exists, but we note that all graphs in this paper are connected by construction. The path P can alternatively be represented as the sequence of connected edges between a and b: $P = [e_{0,1}, e_{1,2}, \ldots, e_{s-1,s}]$. The shortest path is determined based on minimizing the path cost function

$$C(P) = \sum_{e \in P} w(e). \tag{2}$$

In the case of an weighted graph, we then define $D(a, b) = \min_P C(P)$. Note that in the case of an unweighted graph, we have all weights equal to 1 and thus the cost of a path is simply the number of edges included in it.

We next define two variations on this idea, although they are not quite distances but are useful for the kinds of input graph data we study. In particular, the weights on edges are higher for those that are more highly traversed with the transitional networks. We thus want these paths to be considered more important than those only traversed a few times. To that end, we will focus on paths whose length using the reciprocal of the weights is as small as possible.

The first variation, called the *weighted shortest path* measure, is defined as follows. First, we find the path from a to b with the minimum total path weight in terms of the reciprocal weights. That is, P such that

$$C'(P) = \sum_{e \in P} 1/w(e). \tag{3}$$

is minimized. We then define $D(a, b) = \sum_{e \in P} w(e)$. For this definition, D encodes information about frequency of traversal of the edges.

The second variation, called the *shortest weighted path*, still uses the path P for which $C'(P)$ is minimized. However, in this case, we define $D(a, b)$ to be the length of the path; i.e. the number of edges in P. For this variant, we are essentially giving higher priority to well traveled paths, but using a measurement of this path related to the number of regions of state space are traversed.

2. Diffusion Distance

The final vertex similarity measure we use is the diffusion distance for graphs[61]. The diffusion distance leverages the transition probability distribution matrix $\mathbf{P}$ of the graph, where $\mathbf{P}(a, b)$ is the probability of transitioning to b when at a in a single step based on the random walk framework. Specifically, given the weighted, undirected adjacency matrix $\mathbf{A}$ with no self-loops (i.e., zero diagonal), the transitional probability matrix is

$$\mathbf{P}(i,j) = \frac{\mathbf{A}(i,j)}{\sum_{k=1}^{|V|} \mathbf{A}(i,k)}. \tag{4}$$

Equation (4) can be extended to calculate the transition probabilities for non-adjacent neighbors by raising them to higher powers. For example, transitioning to vertex b from vertex a in t random walk steps is $\mathbf{P}^t(a,b)$. A common modification of Eq. (4) is to include a probability that a random walk can stay at the current vertex, which is commonly referred to as the lazy transition probability matrix. This is given by

$$\widetilde{\mathbf{P}} = \frac{1}{2}\left[\mathbf{P}(a,b) + \mathbf{I}\right], \tag{5}$$

where $\mathbf{I}$ is the identity matrix matching the size of $\mathbf{P}$. The diffusion distance measures how similar two nodes are based on comparing their t-step random walk probability distributions. This is done by taking the degree-normalized ℓ_2 norm of the probability distributions between nodes and is calculated as

$$d_t(a,b) = \sqrt{\sum_{c \in V} \frac{1}{\mathbf{d}(c)} \left[\widetilde{\mathbf{P}}^t(a,c) - \widetilde{\mathbf{P}}^t(b,c)\right]^2} \tag{6}$$

where $\mathbf{d}$ is the degree vector of the graph with $\mathbf{d}(i)$ as the degree of node i. Applying the diffusion distance to all node pairs results in the distance matrix $\mathbf{D}_t$.

C. Persistent Homology of Complex Networks

A simplicial complex is a generalization of a graph to higher dimensions, which are collections of simplices at various dimensions (e.g., points are zero-dimensional, edges are one-dimensional, and faces are two-dimensional simplices). These simplices are subsets of a vertex set $\sigma \subset V$, and we require for the complex that if $\sigma \in K$ and $\tau \subseteq \sigma$, then τ is also in K. Using a distance matrix to describe similarity between nodes, or indeed any function of the form $d : V \times V \to \mathbb{R}$ where $d(v,v) = 0$ although we still call this a distance matrix for simplicity, we can construct simplicial complex representations from graphs at a distance level r. This idea is related to the Vietoris Rips complex, where we build a simplicial complex K_r for any fixed parameter $r \geq 0$ by including all simplices with pairwise relationships at most r; i.e. $K_r = \{\sigma \subseteq V \mid d(u,v) \leq r \text{ for all } u,v \in \sigma\}$. Zero-dimensional simplices, the vertices of the complex, are all added at $r = 0$. An edge uv, which is a 1-dimensional simplex, is present in K_r for any r value above $d(u,v)$. Higher dimensional simplices such as triangles are included when all subedges are present; equivalently this means a simplex is added for every clique in the complex. For example, consider Fig. 4 which shows a graph with four nodes, and the associated distance matrix $\mathbf{D}$. For each $r \in [0.0, 0.5, 1.0, 1.5, 2.0]$ the associated simplicial complex is shown as K_r in the bottom row.

We can use homology[62,63] to measure the shape of any such simplicial complex K which is denoted $H_d(K)$. This mathematical object is a vector space, where elements are representative of d-dimensional features (i.e., connected components (zero-dimensional structure), loops (one-dimensional structure), voids (two-dimensional structure), and higher dimensional analogues) in K. In this work we will only utilize the 0-dimensional and 1-dimensional features to measure the connected components and holes in the simplicial complex. For example, consider the simplicial complex K_r at $r = 1.0$ in Fig. 4, which has one H_0 classes with a single connected component and one H_1 class with a single loop or hole in the simplicial complex.

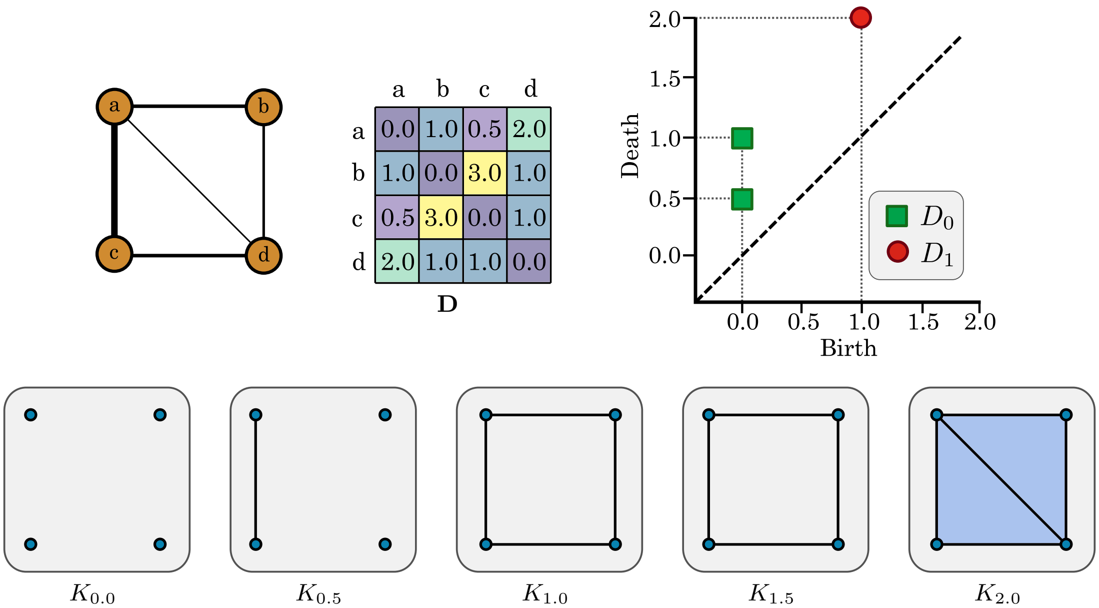


FIG. 4. Example demonstrating persistent homology of a graph using the matrix $\mathbf{D}$ with resulting persistence diagram shown top right. The filtration of simplicial complexes are shown in the bottom row.

An issue with just using homology to measure the shape of a simplicial complex to understand the shape of a graph is that the correct distance value r needs to be selected. Additionally, it does not provide any information on the geometry or size of the underlying graph. To alleviate these issue we use persistent homology[64], which studies the *changing* homology of a sequence of simplicial complexes. We will again use Fig. 4 as an example for demonstrating how the persistent homology is calculated. To calculate the persistent homology we begin with a collection of nested simplicial complexes

$$K_{r_1} \subseteq K_{r_2} \subseteq \cdots \subseteq K_{r_N}.$$

The bottom row of Fig. 4 shows an example of this filtration over the distance parameter r with $K_{r=1.0} \subseteq K_{r=0.5} \subseteq \cdots \subseteq K_{r=2.0}$. We then calculate the homology of each simplicial complex and create linear maps between each homology class for each dimension d as

$$H_d(K_{r_1}) \to H_d(K_{r_1}) \to \cdots \to H_d(K_{r_N}).$$

By studying the formation and disappearance of homology classes we can understand the shape of the underlying graph. Specifically, class $[\alpha] \in H_d(K_{r_i})$ is said to be born at r_i if it is not in the image of the map $H_d(K_{r_{i-1}}) \to H_d(K_{r_i})$. The same class dies at r_j if $[\alpha] \neq 0$ in $H_d(K_{r_{j-1}})$ but $[\alpha] = 0$ in $H_d(K_{r_j})$. In the case of 0-dimensional persistence, this feature is encoding the appearance of a new connected component at K_{r_i} that was not there previously, and which merges with an older component entering K_{r_j}. For 1-dimensional homology, this is the formation (birth) and disappearance (death) of a loop structure. We store this information in what is known as the persistence diagram using the persistence pair $x_i = (b_i, d_i) \in D_d$, where D_d is the persistence diagram of dimension d with a homology class of dimension d being born at filtration value b_i and dying at d_i. We also define the lifetime or persistence of a persistence pair as $\ell_i = \text{pers}(x_i) = d_i - b_i$. The set of lifetimes for dimension d is defined as L_d. For a more detailed roadmap for the calculation of persistent homology we direct the reader to the work of Otter et al[65].

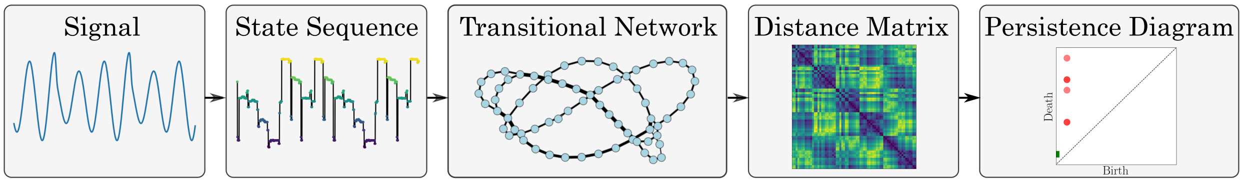


FIG. 5. Pipeline for studying transitional networks using persistent homology. From left to right, we begin with a signal or time series and represent it as a state sequence which is summarized using a transitional network as described in Section II A. A distance between nodes is then used to create a distance matrix (see Section II B for graph distances) which can be directly analyzed using persistent homology shown in Section II C.

Returning to our example, the persistence diagram is shown in Fig. 4 for both D_0 and D_1. For D_0 all four persistence pairs were born at $r = 0.0$ with one dying at $r = 0.5$ and two dying at $r = 1.0$. The fourth persistence pair in D_0, not drawn, is an infinite-class dying at ∞ since there is a single component for $r \geq 1.0$. In this work we do not utilize infinite-class persistence pairs and will not include them in the persistence diagrams. For D_1 there is a single persistence pair born at $r = 1.0$ with the formation of the loop in K_1 and filling in at K_2.

III. METHOD

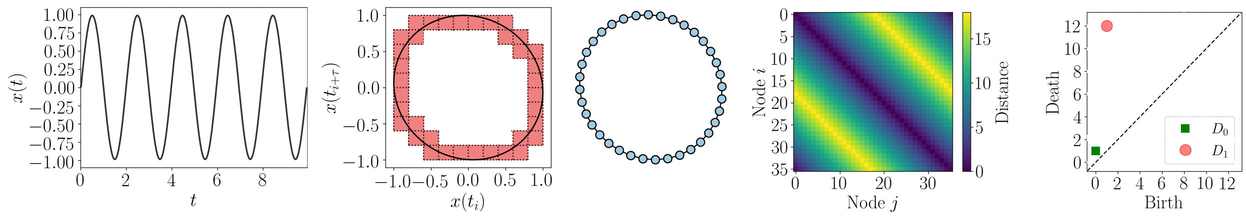
This section describes the method for studying complex transitional networks using persistent homology. The pipeline for doing this is outlined in Fig. 5. We begin with a signal or time series and represent it as a state sequence described in Section II A. The state sequence can be summarized using a weighted transitional network as described in Sec. II A. A distance between nodes (see Section II B) is then used to create a distance matrix which can be directly analyzed using persistent homology as described in Section II C.

To further describe the method we develop here, we use a simple periodic signal example shown in Fig. 6. The signal is defined as $x(t) = \sin(\pi t)$ sampled at a uniform rate of $f_s = 50$ Hz. The SSR was constructed using $n = 2$ and $\tau = 26$. For this example, we create the CGSSN by partitioning the SSR domain into 100 rectangular regions as states, each with a unique symbol. The states visited through the SSR trajectory are highlighted in red. The temporal tracking of the states used creates the state sequence, which is then represented as the cycle graph. This example demonstrates how the periodic nature of the signal is captured by the cycle structure of the corresponding CGSSN.

We define a distance between nodes using the unweighted shortest path distance for this example due to its simplicity. The corresponding distance matrix and resulting persistence diagram are shown. The resulting persistence diagram shows that the periodic structure of the underlying time series and corresponding CGSSN is captured by the single point in the persistence diagram D_1 at coordinate $(1, 12)$ with the loop structure being born at a filtration distance of 1 and filling in 12.

IV. RESULTS

This section shows that the CGSSN outperforms the previously used OPN for both noise robustness and dynamic state detection performance. We first begin in Section IV A where we provide a simple example highlighting improved dynamic state detection performance of the CGSSN over the OPN for a periodic and chaotic Rossler system simulation. We show these results using the persistent entropy summary statistic. The second result in Section IV B quantifies the dynamic state detection, of the OPN and CGSSN using



(a) (b)

FIG. 6. Example demonstrating CGSSN formation procedure ($b = 10$) with the signal $x(t) = \sin(t)$ embedded into $\mathbb{R}^2$ space using an SSR and analysis using persistent homology with the unweighted shortest path distance. (a) Formation of the CGSSN from a time series signal and its delayed signal, (b) The distance matrix and associated persistence diagram using the unweighted shortest path distance.

lower dimensional embedding on 23 continuous dynamical systems with periodic and chaotic simulations. Lastly, in Section IV D, we empirically investigate the noise robustness of the CGSSN compared to the OPN.

A. Dynamic State Detection for Rossler System

Our first result is from a study of the complex network topology of OPNs compared to CGSSNs. To demonstrate the difference and motivate why the CGSSN outperforms the OPN in terms of dynamic state detection, we use an $x(t)$ simulation of the Rossler system defined as

$$\frac{dx}{dt} = -y - z, \qquad \frac{dy}{dt} = x + ay, \qquad \frac{dz}{dt} = b + z(x - c). \tag{7}$$

We simulated Eq. (7) using the *scipy* odeint solver for $t \in [0, 1000]$ with only the last 230 seconds used to avoid transients. The signal was sampled at a rate of $f_s = 22$ Hz. For periodic dynamics we use system parameters of $[a, b, c] = [0.1, 0.2, 14]$ and for chaotic we set $a = 0.15$. These simulated signals are shown in Fig. 7. To create the OPNs for both signals, we used an embedding delay $\tau = 43$ selected using the multi-scale permutation entropy method and dimension $n = 7$. The corresponding networks are shown in the second column of Fig. 7. To form the CGSSNs we similarly chose $\tau = 43$, but used dimension $n = 4$ and $b = 12$ for partitioning the SSR with resulting networks shown in the third column.

The resulting OPN and CGSSN from the Rossler system simulations of periodic and chaotic dynamics both capture the increasing complexity of the signal with the dynamic state change. For the periodic signal, the OPN show overarching large loops relating to the periodic nature of the SSR. However, the CGSSN better captures the periodic nature of the trajectory with only a single loop forming. This characteristic of the CGSSN is due to periodic flows never intersecting in the SSR if the signal is sampled at a high enough frequency, there is no or little additive noise, and an appropriately sized delay and dimension are selected. While correctly choosing the delay and dimension is not a trivial task, there is a broad literature on their selection for the SSR task. This work relies on the multi-scale permutation entropy method for selecting the delay and the false-nearest-neighbors algorithm[59] for selecting an appropriate SSR dimension. However, we found that increasing the dimension one higher than that suggested using false-nearest-neighbors more reliably formed a single loop structure in the CGSSN. Additionally, in Appendix A we demonstrate that for 23 dynamical systems, setting $b \geq 12$ resulted in only a single loop structure for periodic signals while minimizing the computational demand when using the CGSSN. As such, we set $b = 12$ unless otherwise stated.

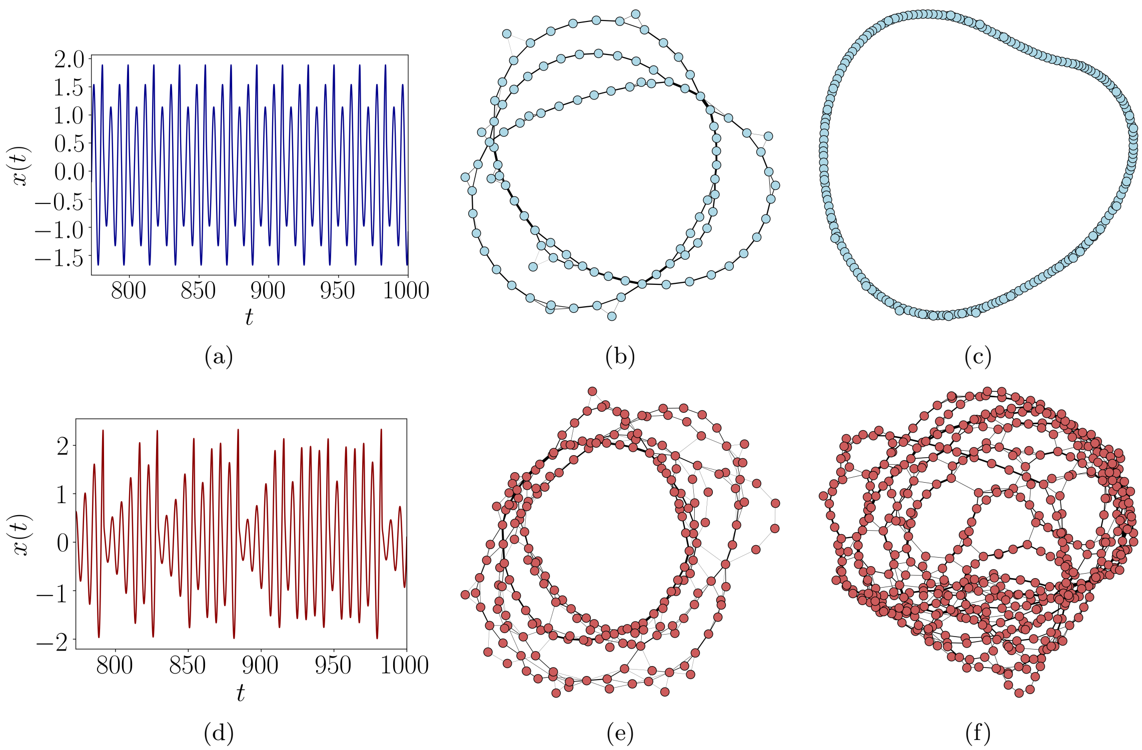


FIG. 7. Transitional complex network topology comparison between OPN and CGSSN for the $x(t)$ simulation of the Rossler system described in Eq. (7). (a) Periodic Rossler Simulation $x(t)$, (b) Periodic OPN ($n = 7$) $E'(D_1) = 0.503.$, (c) Periodic CGSSN ($n = 4$ and $b = 12$). $E'(D_1) = 0.026$, (d) Chaotic Rossler Simulation $x(t)$, (e) Chaotic OPN ($n = 7$). $E'(D_1) = 0.893$, (f) Chaotic CGSSN ($n = 4$ and $b = 12$). $E'(D_1) = 0.905$.

For the chaotic $x(t)$, the OPN and CGSSN both summarize the topology of the attractor with both networks having a high degree of entanglement with nodes being highly intertwined. This is a typical characteristic of complex transitional networks formed from chaotic signals. Furthermore, it should be noted that the CGSSN tends to be more entangled than its OPN counterpart, suggesting that the CGSSN better captures the increase in complexity of the chaotic signal.

To quantify how well the OPN and CGSSN capture the complexity of the signals, we rely on persistent entropy[66], which was previously adapted[25] to study the resulting persistence diagram using the unweighted shortest path distance of complex networks. The normalized persistent entropy[67,68] is defined as

$$E'(D) = \frac{-\sum_{x\in D} \frac{\mathrm{pers}(x)}{\mathscr{L}(D)} \log_2 \left(\frac{\mathrm{pers}(x)}{\mathscr{L}(D)}\right)}{\log_2 \left(\mathscr{L}(D)\right)}, \tag{8}$$

where $\mathscr{L}(D) = \sum_{x\in D} \mathrm{pers}(x)$ with $\mathrm{pers}(x) = |b - d|$ as the lifetime or persistence of point $x = (b, d)$ in a persistence diagram D. For studying the complexity of transitional network we apply this score to the one-dimensional persistent diagram D_1, which measures the loop structures in the network. This score yields a value close to zero for networks with a single loop structure corresponding to periodic dynamics and a value close to one for chaotic dynamics with highly intertwined networks. For our example OPN and CGSSNs in Fig. 7 we get normalized persistent entropy scores of 0.503 and 0.893 for periodic and chaotic OPNs, respectively, and 0.026 and 0.905 for CGSSNs. These statistics show that the CGSSN outperforms the OPN with a significantly larger difference in the entropy values. This is mainly due to the CGSSN having a score near zero for periodic dynamics due to its general loop structure compared to the periodic OPN having several loops. This result comparing the OPN and CGSSN suggests that the CGSSN will outperform the OPN for the dynamic state detection task. With this single case under our belt, we turn our attention to an empirical study of this characteristic over more dynamical systems.

B. Empirical Testing of Dynamic State Detection for 23 Continuous Dynamical Systems

The previous example in Section IV A showed the improved dynamic state detection performance of the CGSSN over the OPN for a single example (Rössler System). However, we want to show that this improvement is present over various systems. To do this, we use 23 continuous dynamical systems listed in the Appendix B with details on the simulation method–each system was simulated for both periodic and chaotic dynamics.

For each periodic and chaotic signal, we calculate the resulting persistence diagram of the OPN and CGSSN using each of the distance methods (unweighted shortest path, shortest weighted path, weighted shortest path, and diffusion distance). We then compare the collection of persistence diagrams for a specific network type (OPN or CGSSN) and distance measure by calculating the bottleneck distance matrix between each persistence diagram. The bottleneck distance $d_{BN}(D, F)$ is a similarity measure between two persistence diagrams (D and F). It is calculated as the sup norm distance between the persistence diagrams, where the persistence diagrams are optimally matched with the distance between matched persistence points being at most d_{BN}. The bottleneck distance matrix $\mathbf{D}_{BN}$ is calculated by finding d_{BN} between all persistence diagrams.

The question we are trying to answer is if periodic and chaotic dynamics result in similar persistence diagrams across multiple systems. To answer this, we first use a lower-dimensional projection of $\mathbf{D}_{BN}$ by implementing the Multi-Dimensional Scaling (MDS) projection to two dimensions. To measure how well the dynamics delineate on the MDS projection, we use a Support Vector Machine (SVM) with a Radial Basis Function (RBF). Note that because the MDS does not allow for the mapping of previously unseen points, we cannot use this procedure for a proper classification test as we cannot approximate training error. However, we can use this procedure to see if the the persistence diagrams of different classes are separated with respect to the bottleneck distance.

We fit the SVM using the default `SKLearn` SVM parameters package. The resulting separations for periodic and chaotic dynamics using the OPN (left) and CGSSN (right) are shown in Appendix C. These separations are for the diffusion distance calculation as it provided the best results for both the OPN and CGSSN. However, we also include similar figures for other choices of distances in Appendix C.

Figure 8 demonstrates the significant improvement in dynamic state detection of the CGSSN over the OPN. This is shown with the periodic and chaotic networks being clustered for the CGSSN (right of Fig. 8) with no overlap compared to the OPN (left of Fig. 8) having some overlap between periodic and chaotic dynamics. This is further shown with the SVM kernel being able to separate the periodic and chaotic regions for the CGSSN easily. To better compare all distance measures and complex network combinations, we quantify the performance of each SVM kernel using the accuracy of the separation. We repeated this accuracy calculation 100 times for each combination using 100 random seeds to generate the SVM kernels. The resulting average accuracies with standard deviation uncertainties are reported in Table I.

Based on the results in Table I, the CGSSN outperforms the OPN for all distance measures. Additionally, we found 100% separation accuracy for both the shortest weighted path and diffusion distances when combined with the CGSSN. We believe this performance improvement is due to the coarse-graining procedure

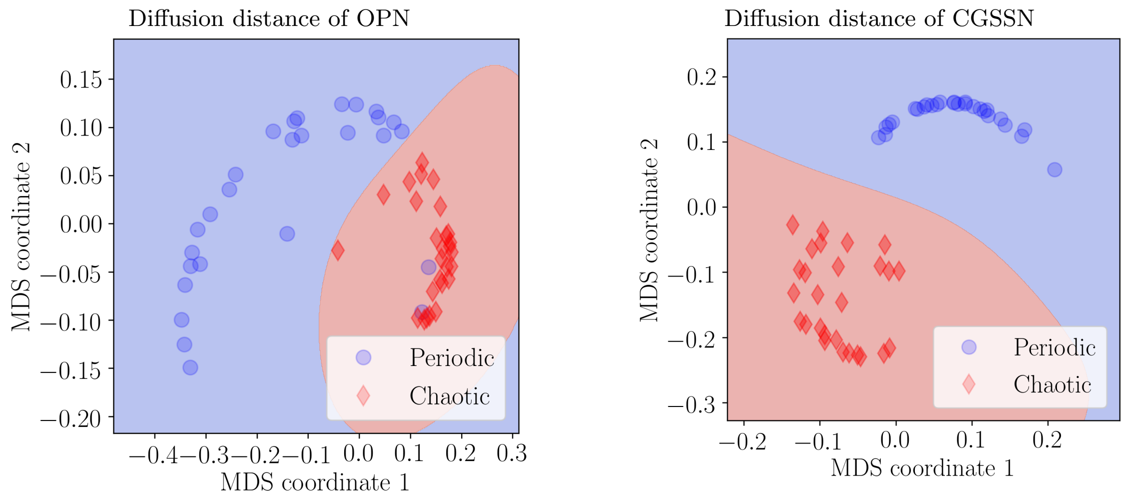


FIG. 8. Two-dimensional MDS projection of the bottleneck distances between persistence diagrams of the chaotic and periodic dynamics with an SVM radial bias function kernel separation. This separation analysis was repeated for the OPNs and CGSSNs using the diffusion distance.

TABLE I. Accuracies for SVM seperation of MDS projections for dynamic state detection. Uncertainties are recorded as one standard deviation for random seeds 1-100.

Network	**Distance**	**Average Separation Accuracy**	**Uncertainty**
OPN	Shortest Unweighted Path Distance	80.7%	1.5%
OPN	Shortest Weighted Path Distance	88.9%	0.0%
OPN	Weighted Shortest Path Distance	88.9%	0.0%
OPN	Lazy Diffusion Distance	95.0%	0.9%
CGSSN	Shortest Unweighted Path Distance	98.1%	0.9%
CGSSN	Shortest Weighted Path Distance	100.0%	0.0%
CGSSN	Weighted Shortest Path Distance	98.1%	0.9%
CGSSN	Lazy Diffusion Distance	100.0%	0.0%

capturing the SSR vector's amplitude information which is discarded when identifying permutations in the OPN.

1. n-Periodic Systems

Based on the state space embedding structure of a system, one may expect that for a 2 or 3-periodic system that the CGSSN may result in 2 and 3 loops respectively, but this is not the case. In general, for an n-periodic system, we expect the CGSSN to contain only a single loop, and so we caution the user that this method will likely not be able to differentiate differences in the periodicity. We demonstrate this nuance by showing CGSSN results on the Lorenz system for multi-periodic responses. Fig. 9 shows the corresponding CGSSNs for the Lorenz system varying the ρ parameter to obtain multi-periodic responses. The networks are labeled with a sequence of A's and B's where each letter corresponds to a loop in the trajectory around one of the attractors. For example AAB trajectory would be two loops around A and one around B before

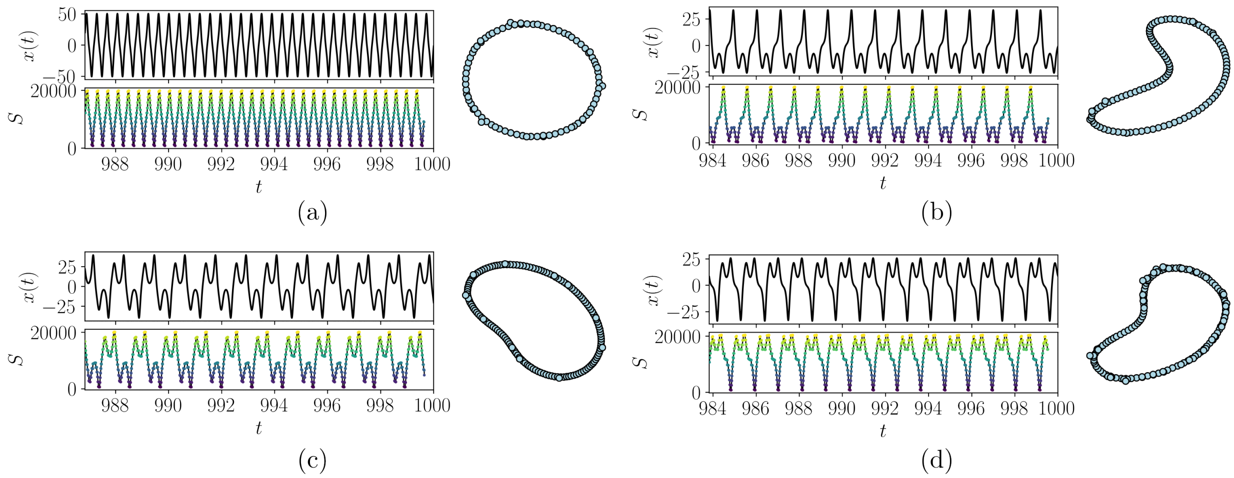


FIG. 9. CGSSN results for four multi-periodic cases of the Lorenz system. The sequence of A's and B's below each image indicates the oribital sequence around the attractors A and B in the system. (a) AB $\rho = 350$, (b) AAB $\rho = 100.5$, (c) AABB $\rho = 160$, (d) ABBABB $\rho = 99.65$. As expected, all four cases result in a single loop CGSSN. These networks were generated using $n = 4$ and $b = 12$.

repeating the cycle. For all four cases shown, a single loop is obtained in the CGSSN even though the system exhibits multi-periodicity.

C. A Remark on Discrete Maps

As we demonstrated in Section. IV B, the CGSSN method allows for efficient and accurate dynamic state detection over a range of continuous dynamical systems. Discrete maps are another subset of dynamical systems where it would be useful to apply these tools; however, care must be taken for this type of system to ensure that the CGSSN is a suitable approach. This is because in discrete systems, there are typically far fewer states that the system can exhibit so in some cases the CGSSN may not contain any loops, but the response is still periodic leading to an incorrect classification in the model. To demonstrate, we show the CGSSNs for the periodic and chaotic logistic map in Fig. 10 where the unweighted shortest path distance was used to compute persistence. We see that the CGSSNs show vastly different structures where the periodic network contains a single loop and the chaotic network is tangled. However, the persistence diagrams for these networks appear to be equivalent because the networks were unweighted and all of the loops in the chaotic network are exactly the same size as the periodic case. Due to only having 4 possible states in the periodic logistic map here, the network loop does not provide enough of a difference to automatically classify it as either dynamic state. We note that the chaotic persistence diagram contains more loops than the periodic case here, but all are the same persistence lifetime. In the case of a continuous system where many more states are possible, these loops will be larger in size and the persistence diagram will reflect those differences allowing for classification of the dynamic state. In this case, when other distances are used such as the shortest weighted path, the resulting persistence diagrams have the forms that we expect for periodic and chaotic behaviors due to the weighting of the edges influencing the persistence lifetime of that loop.

In the case where the system being studied can exhibit many possible states in its periodic response, a single loop will form the CGSSN and the persistence diagram will show a persistence pair with a long

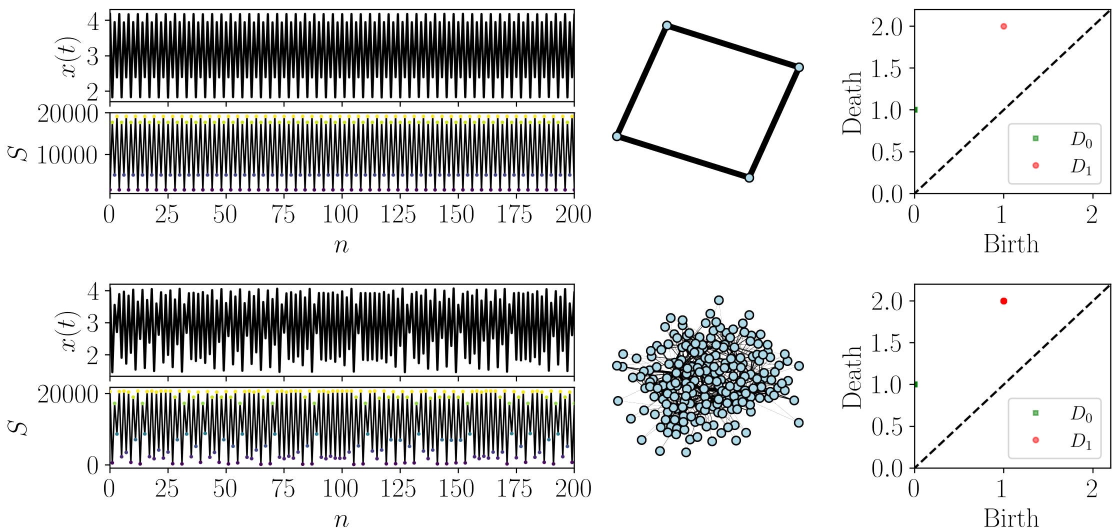


FIG. 10. CGSSN results for the periodic (top row) and chaotic (bottom row) logistic map using the unweighted shortest path. The system responses are shown on the left along with the permutation sequence. The network representations are in the middle with the persistence diagrams on the right. Both networks exhibit the same persistence diagram due to the limited possible system states for the periodic case.

lifetime. For example, we demonstrate this behavior on the 3 periodic linear congruential generator map in Fig. 11. The results in Figs. 10 and 11 demonstrate that this method should be used with caution on discrete systems and for systems with enough states that approach the behavior of a continuous system, the CGSSN persistence diagrams can provide a correct dynamic state detection.

D. Noise Sensitivity

One issue with ordinal partition networks is they are not exceptionally resilient to noise. Indeed, one can think of the ordinal partition network as being the 1-skeleton of the nerve of a particular closed cover of the state space, delineated by the hyperplanes $x_i \leq x_j$. Consequently, when noise is injected into the system, there are superfluous transitions when nearing one of these boundaries. For example, consider the signal and its embedding into $\mathbb{R}^3$ in Fig. 12.

This effect becomes even more prominent near an intersection of multiple hyperplanes. As the distance to the hyperdiagonal d^H becomes small, we see a significant increase in seemingly superfluous transitions between permutations π (highlighted in orange in Fig. 12). This issue is even more exaggerated when the embedded signal is consistently close to the hyperdiagonal, which results in network representations whose shape carries no information on the underlying dynamical system (e.g., see the signal and far-right OPN in Fig. 13). This is particularly detrimental when we attempt to include the weighting information, as the flips can skew the count for the number of times a boundary is crossed.

Certain network representations of time series are naturally more noise-robust than others. For example, Fig. 13 shows the OPN and CGSSN for the signal with and without noise. This example demonstrates that the CGSSN is the best choice for this signal with only minor changes in its shape, while the OPN loses all resemblance to the noise-free network.

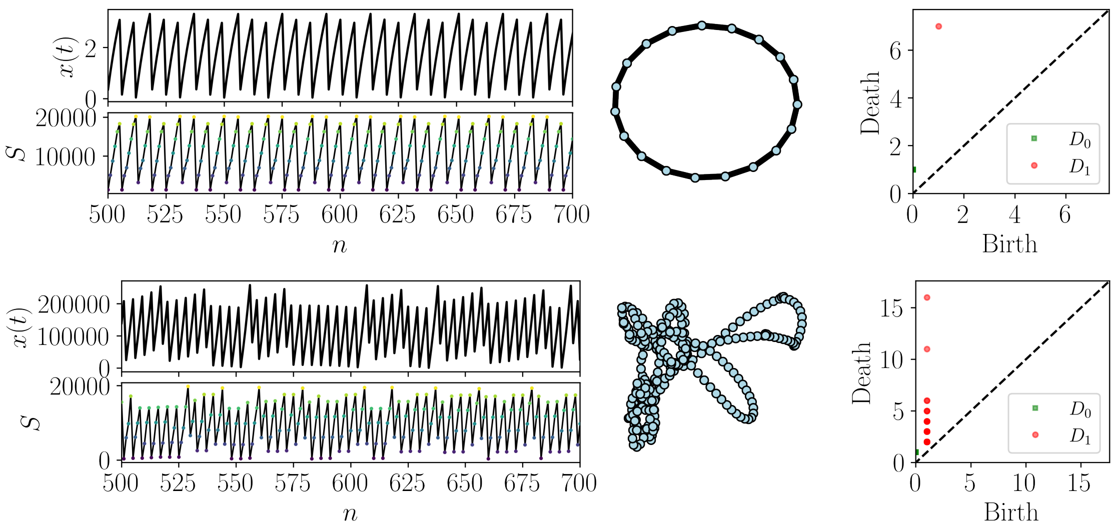


FIG. 11. CGSSN results for the periodic (top row) and chaotic (bottom row) linear congruential generator map using the unweighted shortest path. The system responses are shown on the left along with the permutation sequence. The network representations are in the middle with the persistence diagrams on the right. Both networks exhibit the distinct persistence diagram structures due to the larger loop in the periodic network.

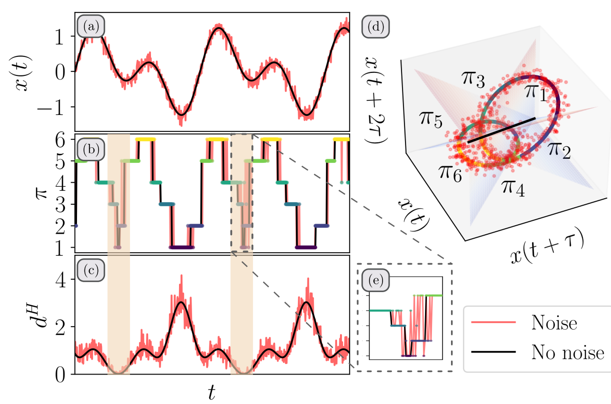


FIG. 12. The three-dimensional state space reconstruction (d) from the signal $x(t)$ with and without additive noise (a) shows as the distance to the hyperdiagonal d^H (c) becomes small, undesired permutation transitions (b)—with zoomed-in section shown in (e)—occur as shown in the orange highlighted regions.

Outside of this sensitivity to the hyperdiagonal, we also found that the CGSSN is more noise robust than the OPN for other signals. For example, in Fig. 14 we show the normalized persistent entropy statistic from Eq. (8) calculated for the periodic and chaotic simulations of the Rössler system defined in Eq. (7) when additive noise is present in the signal. We incremented the additive noise using the Signal-to-Noise Ratio (SNR). The SNR (units of decibels) is defined as $\mathrm{SNR} = 20\log_{10}(A_{\mathrm{signal}}/A_{\mathrm{noise}})$, where A_{signal} and A_{noise}

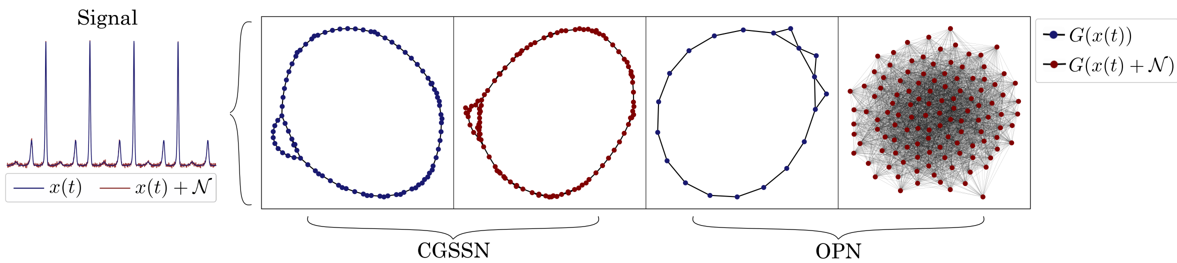


FIG. 13. Example demonstrating the importance of choosing an appropriate network formation method when there is additive noise in the signal. The CGSSN retains the graph structure even with additive noise; in contrast, the OPN network loses all resemblance to the noise-free topological structure even with a small amount of additive noise. $x(t)$ is the signal, $\mathcal{N}$ is additive noise, and $G(x)$ is the graph representation of x.

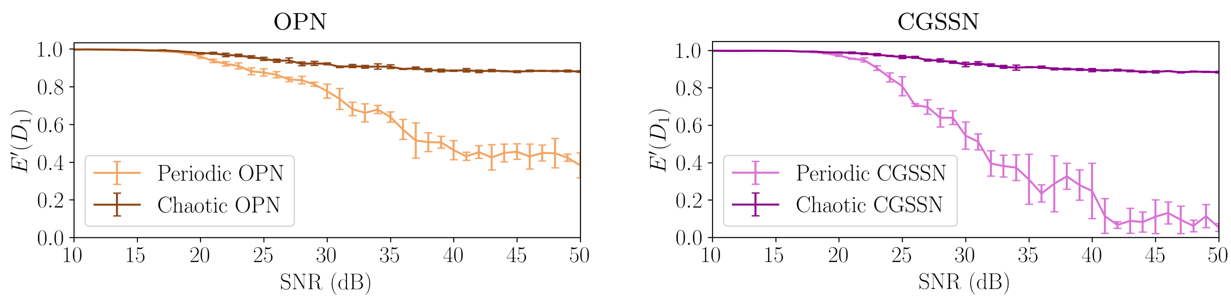


FIG. 14. Noise robustness analysis of dynamic state detection using the summary statistic persistent entropy (see Eq. (8)) for OPN and CGSSN with increasing SNR on a periodic Rossler simulation from Eq. (7).

are the root-mean-square amplitudes of the signal and additive noise, respectively. This result shows that for this signal the OPN network is only robust down to an SNR of approximately 32 dB of additive Gaussian noise, while the CGSSN is able to separate periodic from chaotic dynamics down to approximately 23 dB. We found similar results for the other 22 dynamical systems investigated in this work.

E. Experimental Results

To validate these tools, we apply them to experimental data collected from a base excited magnetic pendulum[69]. This system was shown to exhibit periodic and chaotic behavior under different parameters and the CGSSN persistence diagrams were generated for each case using all 4 distance measures presented in this paper. Figure 15 shows the corresponding time series, permutation sequence, CGSSN, and persistence diagrams for the periodic response. We see that for all of the distance metrics, there is a clear singular cycle that forms with a significant persistence lifetime. Conversely, the same results are presented for the chaotic response in Fig. 16 where we see a drastically different distribution of persistence pairs corresponding to the high number of cycles present in the chaotic CGSSN. The results presented here are in agreement with our work in[69].

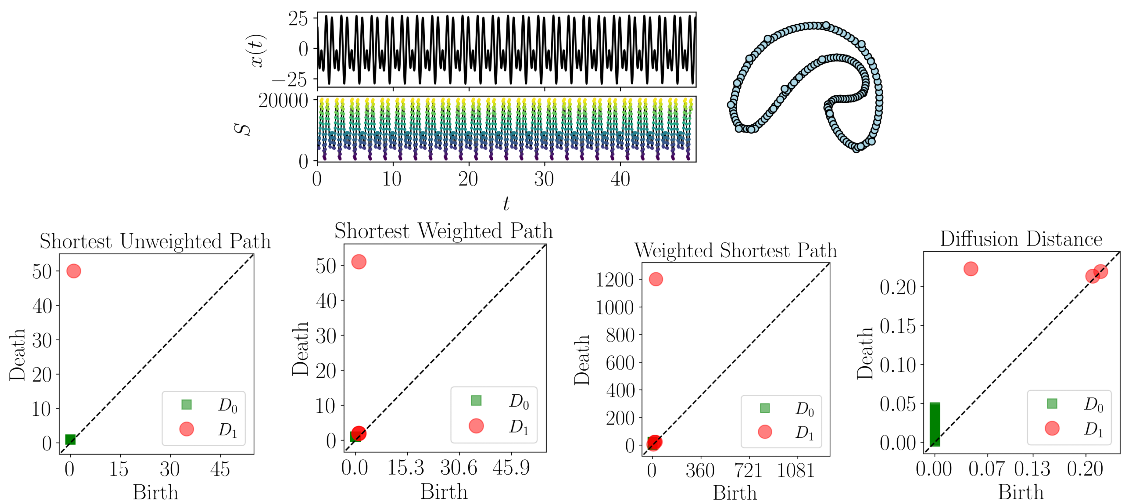


FIG. 15. CGSSN results for the forced single magnetic pendulum under conditions that yield a periodic response. The top left images show the time series and permutation sequence and the top right shows the coarse grained state space network. The bottom row shows the corresponding persistence diagrams for the network under the distance metric in the title of each diagram.

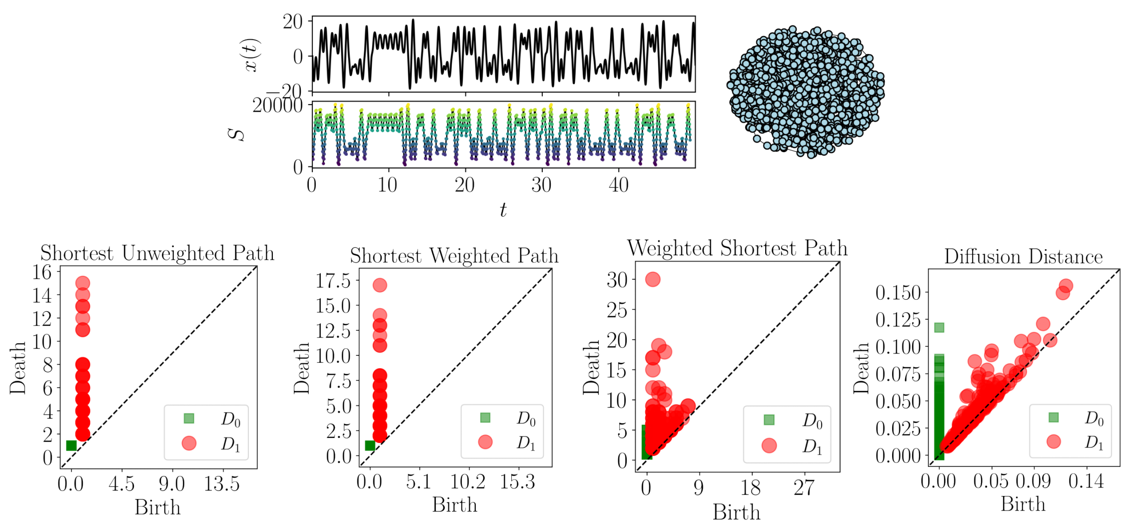


FIG. 16. CGSSN results for the forced single magnetic pendulum under conditions that yield a chaotic response. The top left images show the time series and permutation sequence and the top right shows the coarse grained state space network. The bottom row shows the corresponding persistence diagrams for the network under the distance metric in the title of each diagram.

V. CONCLUSION

In this work, we developed a novel framework for studying CGSSNs using persistent homology. We showed that the CGSSN outperformed the standard ordinal partition network in both noise robustness and dynamic state detection performance, with the CGSSN reaching 100% separation accuracy for dynamic state detection for 23 continuous dynamical systems. This is in comparison to the OPN, which could at most reach 95% accuracy. This approach was validated using data from a magnetic pendulum experiment to show that the topological structure for periodic and chaotic timseries are captured in the resulting persistence diagrams.

In this work, we only investigated the most straightforward construction of the CGSSN. Namely, the equal-sized hyper-cube tessellation cover of the SSR domain. Possible improvements to the CGSSN could be through a data-dependent adaptive cover algorithm. We also suspect that other choices of distances could provide improvements for the given pipeline.

Another future direction would be to prove a stability theorem for the CGSSN. That is, can we show that for a noisy version of a signal, the resulting CGSSN, and subsequently the computed persistence diagram, is similar to the ground truth. It would also be interesting to study how the CGSSN could serve to detect quasiperiodicity. We believe that the torus shape associated to the SSR of quasiperiodic signals could be captured using the CGSSN as it accounts for the signal amplitude.

ACKNOWLEDGEMENTS

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-22-1-0007.

REFERENCES

[1] G. A. Gottwald and I. Melbourne, "On the implementation of the 0–1 test for chaos," *SIAM Journal on Applied Dynamical Systems*, vol. 8, pp. 129–145, jan 2009.

[2] G. A. Gottwald and I. Melbourne, "The 0-1 test for chaos: A review," in *Chaos Detection and Predictability*, pp. 221–247, Springer Berlin Heidelberg, 2016.

[3] M. Melosik and W. Marszalek, "On the 0/1 test for chaos in continuous systems," *Bulletin of the Polish Academy of Sciences Technical Sciences*, vol. 64, no. 3, pp. 521–528, 2016.

[4] M. T. Rosenstein, J. J. Collins, and C. J. D. Luca, "A practical method for calculating largest lyapunov exponents from small data sets," *Physica D: Nonlinear Phenomena*, vol. 65, no. 1-2, pp. 117 – 134, 1993.

[5] C. Froeschlé, R. Gonczi, and E. Lega, "The fast lyapunov indicator: a simple tool to detect weak chaos. application to the structure of the main asteroidal belt," *Planetary and Space Science*, vol. 45, pp. 881–886, jul 1997.

[6] J. Sprott, "Chaos and time-series analysis," *Choice Reviews Online*, vol. 41, pp. 41–3492–41–3492, feb 2004.

[7] S. B. B. Aval, V. Ahmadian, M. Maldar, and E. Darvishan, "Damage detection of structures using signal processing and artificial neural networks," *Advances in Structural Engineering*, vol. 23, pp. 884–897, nov 2019.

[8] N. M. Nejad, S. B. B. Aval, M. Maldar, and B. Asgarian, "A damage detection procedure using two major signal processing techniques with the artificial neural network on a scaled jacket offshore platform," *Advances in Structural Engineering*, vol. 24, pp. 1655–1667, dec 2020.

[9] Z. Guo and Z. Sun, "Multiple cracked beam modeling and damage detection using frequency response function," *Structural Longevity*, vol. 5, no. 2, pp. 97–106, 2011.

[10] H. Sohn and C. R. Farrar, "Damage diagnosis using time series analysis of vibration signals," *Smart Materials and Structures*, vol. 10, pp. 446–451, 2001.

[11] M. S. Cao, G. G. Sha, Y. F. Gao, and W. Ostachowicz, "Structural damage identification using damping: a compendium of uses and features," *Smart Materials and Structures*, vol. 26, p. 043001, mar 2017.

[12]S.-H. Lee, J. S. Lim, J.-K. Kim, J. Yang, and Y. Lee, "Classification of normal and epileptic seizure EEG signals using wavelet transform, phase-space reconstruction, and euclidean distance," *Computer Methods and Programs in Biomedicine*, vol. 116, pp. 10–25, aug 2014.
[13]F. M. Roberts, R. J. Povinelli, and K. M. Ropella, "Identification of ECG arrhythmias using phase space reconstruction," in *Principles of Data Mining and Knowledge Discovery*, pp. 411–423, Springer Berlin Heidelberg, 2001.
[14]T. Liu, W. Yao, M. Wu, Z. Shi, J. Wang, and X. Ning, "Multiscale permutation entropy analysis of electrocardiogram," *Physica A: Statistical Mechanics and its Applications*, vol. 471, pp. 492–498, 2017.
[15]B. Frank, B. Pompe, U. Schneider, and D. Hoyer, "Permutation entropy improves fetal behavioural state classification based on heart rate analysis from biomagnetic recordings in near term fetuses," *Medical and Biological Engineering and Computing*, vol. 44, no. 3, p. 179, 2006.
[16]A. A. S. Khan, U. Mumtahina, and N. Yeasmin, "Heart rate variability analysis using approximate entropy and detrended fluctuation for monitoring heart condition," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, IEEE, may 2013.
[17]V. Millette and N. Baddour, "Signal processing of heart signals for the quantification of non-deterministic events," *BioMedical Engineering OnLine*, vol. 10, no. 1, p. 10, 2011.
[18]M. Omidvar, A. Zahedi, and H. Bakhshi, "EEG signal processing for epilepsy seizure detection using 5-level db4 discrete wavelet transform, GA-based feature selection and ANN/SVM classifiers," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 10395–10403, jan 2021.
[19]T. K. Dey and Y. Wang, *Computational Topology for Data Analysis*. Cambridge University Press, 2021.
[20]E. Munch, "A user's guide to topological data analysis," *Journal of Learning Analytics*, vol. 4, pp. 47–61, jul 2017.
[21]M. Robinson, *Topological Signal Processing*. Springer Berlin Heidelberg, 2014.
[22]C. M. Topaz, L. Ziegelmeier, and T. Halverson, "Topological data analysis of biological aggregation models," *PLOS ONE*, vol. 10, p. e0126383, may 2015.
[23]M. R. McGuirl, A. Volkening, and B. Sandstede, "Topological data analysis of zebrafish patterns," *Proceedings of the National Academy of Sciences*, vol. 117, pp. 5113–5124, feb 2020.
[24]A. Myers and F. A. Khasawneh, "Dynamic state analysis of a driven magnetic pendulum using ordinal partition networks and topological data analysis," in *Volume 7: 32nd Conference on Mechanical Vibration and Noise (VIB)*, American Society of Mechanical Engineers, aug 2020.
[25]A. Myers, E. Munch, and F. A. Khasawneh, "Persistent homology of complex networks for dynamic state detection," *Physical Review E*, vol. 100, aug 2019.
[26]M. C. Yesilli, F. A. Khasawneh, and A. Otto, "On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition," *CIRP Journal of Manufacturing Science and Technology*, vol. 28, pp. 118–135, jan 2020.
[27]M. C. Yesilli and F. A. Khasawneh, "On transfer learning of traditional frequency and time domain features in turning," in *Volume 2: Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability*, American Society of Mechanical Engineers, sep 2020.
[28]M. C. Yesilli and F. A. Khasawneh, "Data-driven and automatic surface texture analysis using persistent homology," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, dec 2021.
[29]M. C. Yesilli, F. A. Khasawneh, and A. Otto, "Topological feature vectors for chatter detection in turning processes," *The International Journal of Advanced Manufacturing Technology*, vol. 119, pp. 5687–5713, jan 2022.
[30]M. C. Yesilli, F. A. Khasawneh, and A. Otto, "Chatter detection in turning using machine learning and similarity measures of time series via dynamic time warping," *Journal of Manufacturing Processes*, vol. 77, pp. 190–206, 2022.
[31]M. Gidea and Y. Katz, "Topological data analysis of financial time series: Landscapes of crashes," *Physica A: Statistical Mechanics and its Applications*, vol. 491, pp. 820–834, feb 2018.
[32]M. Gidea, D. Goldsmith, Y. Katz, P. Roldan, and Y. Shmalo, "Topological recognition of critical transitions in time series of cryptocurrencies," *Physica A: Statistical Mechanics and its Applications*, vol. 548, p. 123843, jun 2020.
[33]M. Gidea, "Topological data analysis of critical transitions in financial networks," in *3rd International Winter School and Conference on Network Science NetSci-X 2017* (P. R. Shmueli E., Barzel B., ed.), Springer Proceedings in Complexity, Springer, Cham, 2017.
[34]P. T.-W. Yen and S. A. Cheong, "Using topological data analysis (TDA) and persistent homology to analyze the stock markets in singapore and taiwan," *Frontiers in Physics*, vol. 9, mar 2021.
[35]C.-S. Hu and M.-C. Yeh, "A topological data analysis approach to video summarization," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, sep 2019.
[36]C. J. Tralie and B. McFee, "Enhanced hierarchical music structure annotations via feature level similarity fusion," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2019*, Feb. 2019.
[37]S. Tymochko, E. Munch, and F. A. Khasawneh, "Using zigzag persistent homology to detect hopf bifurcations in dynamical systems," *Algorithms*, vol. 13, p. 278, oct 2020.

[38]S. Tymochko, E. Munch, and F. A. Khasawneh, “Adaptive partitioning for template functions on persistence diagrams,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, IEEE, dec 2019.
[39]G. Muszynski, K. Kashinath, V. Kurlin, and M. W. and, “Topological data analysis and machine learning for recognizing atmospheric river patterns in large climate datasets,” *Geoscientific Model Development*, vol. 12, pp. 613–628, feb 2019.
[40]Y.-M. Chung, C.-S. Hu, Y.-L. Lo, and H.-T. Wu, “A persistent homology approach to heart rate variability analysis with an application to sleep-wake classification,” *Frontiers in Physiology*, vol. 12, mar 2021.
[41]S. Emrani, T. Gentimis, and H. Krim, “Persistent homology of delay embeddings and its application to wheeze detection,” *Signal Processing Letters, IEEE*, vol. 21, pp. 459–463, April 2014.
[42]S. Maletić, Y. Zhao, and M. Rajković, “Persistent topological features of dynamical systems,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, p. 053105, may 2016.
[43]M. Small, J. Zhang, and X. Xu, “Transforming time series into complex networks,” in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 2078–2089, Springer Berlin Heidelberg, 2009.
[44]Y. Yang and H. Yang, “Complex network-based time series analysis,” *Physica A: Statistical Mechanics and its Applications*, vol. 387, pp. 1381–1386, feb 2008.
[45]Z. Gao and N. Jin, “Complex network from time series based on phase space reconstruction,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, p. 033137, sep 2009.
[46]A. Khor and M. Small, “Examining k-nearest neighbour networks: Superfamily phenomena and inversion,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, p. 043101, apr 2016.
[47]R. V. Donner, Y. Zou, J. F. Donges, N. Marwan, and J. Kurths, “Recurrence networks—a novel paradigm for nonlinear time series analysis,” *New Journal of Physics*, vol. 12, p. 033025, mar 2010.
[48]L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño, “From time series to complex networks: The visibility graph,” *Proceedings of the National Academy of Sciences*, vol. 105, pp. 4972–4975, mar 2008.
[49]M. McCullough, M. Small, T. Stemler, and H. H.-C. Iu, “Time lagged ordinal partition networks for capturing dynamics of continuous dynamical systems,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, p. 053101, may 2015.
[50]M. Wang and L. Tian, “From time series to complex networks: The phase space coarse graining,” *Physica A: Statistical Mechanics and its Applications*, vol. 461, pp. 456–468, nov 2016.
[51]T. Weng, J. Zhang, M. Small, R. Zheng, and P. Hui, “Memory and betweenness preference in temporal networks induced from time series,” *Scientific Reports*, vol. 7, feb 2017.
[52]A. S. L. O. Campanharo, M. I. Sirer, R. D. Malmgren, F. M. Ramos, and L. A. N. Amaral, “Duality between time series and networks,” *PLoS ONE*, vol. 6, p. e23378, aug 2011.
[53]G. Nicolis, A. G. Cantú, and C. Nicolis, “Dynamical aspects of interaction networks,” *International Journal of Bifurcation and Chaos*, vol. 15, pp. 3467–3480, nov 2005.
[54]V. F. Silva, M. E. Silva, P. Ribeiro, and F. Silva, “Novel features for time series analysis: a complex networks approach,” *Data Mining and Knowledge Discovery*, mar 2022.
[55]M. Small, “Complex networks from time series: Capturing dynamics,” in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, IEEE, may 2013.
[56]A. Myers, F. A. Khasawneh, and E. Munch, “Topological signal processing using the weighted ordinal partition network,” 2022.
[57]M. Staniek and K. Lehnertz, “Parameter Selection for Permutation Entropy Measurements,” *International Journal of Bifurcation and Chaos*, vol. 17, pp. 3729–3733, oct 2007.
[58]A. Myers and F. A. Khasawneh, “On the automatic parameter selection for permutation entropy,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, p. 033130, mar 2020.
[59]A. Krakovská, K. Mezeiová, and H. Budáčová, “Use of false nearest neighbours for selecting variables and embedding parameters for state space reconstruction,” *Journal of Complex Systems*, vol. 2015, 2015.
[60]Y. Zou, R. V. Donner, N. Marwan, J. F. Donges, and J. Kurths, “Complex network approaches to nonlinear time series analysis,” *Physics Reports*, vol. 787, pp. 1–97, 2019. Complex network approaches to nonlinear time series analysis.
[61]R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006. Special Issue: Diffusion Maps and Wavelets.
[62]A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002.
[63]J. R. Munkres, *Elements of Algebraic Topology*. Addison Wesley, 1993.
[64]A. Zomorodian and G. Carlsson, “Computing persistent homology,” *Discrete & Computational Geometry*, vol. 33, pp. 249–274, Nov. 2004.
[65]N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, “A roadmap for the computation of persistent homology,” *EPJ Data Science*, vol. 6, aug 2017.
[66]N. Atienza, R. Gonzalez-Diaz, and M. Soriano-Trigueros, “A new entropy based summary function for topological data analysis,” *Electronic Notes in Discrete Mathematics*, vol. 68, pp. 113–118, jul 2018.

[67]N. Atienza, R. Gonzalez-Díaz, and M. Soriano-Trigueros, "On the stability of persistent entropy and new summary functions for topological data analysis," *Pattern Recognition*, vol. 107, p. 107509, 2020.

[68]N. Atienza, M.-J. Jimenez, and M. Soriano-Trigueros, "Stable topological summaries for analyzing the organization of cells in a packed tissue," *Mathematics*, vol. 9, p. 1723, jul 2021.

[69]A. Myers and F. A. Khasawneh, "Dynamic state analysis of a driven magnetic pendulum using ordinal partition networks and topological data analysis," in *Volume 7: 32nd Conference on Mechanical Vibration and Noise (VIB)*, American Society of Mechanical Engineers, aug 2020.

[70]A. D. Myers, M. Yesilli, S. Tymochko, F. Khasawneh, and E. Munch, "Teaspoon: A comprehensive python package for topological signal processing," in *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020.

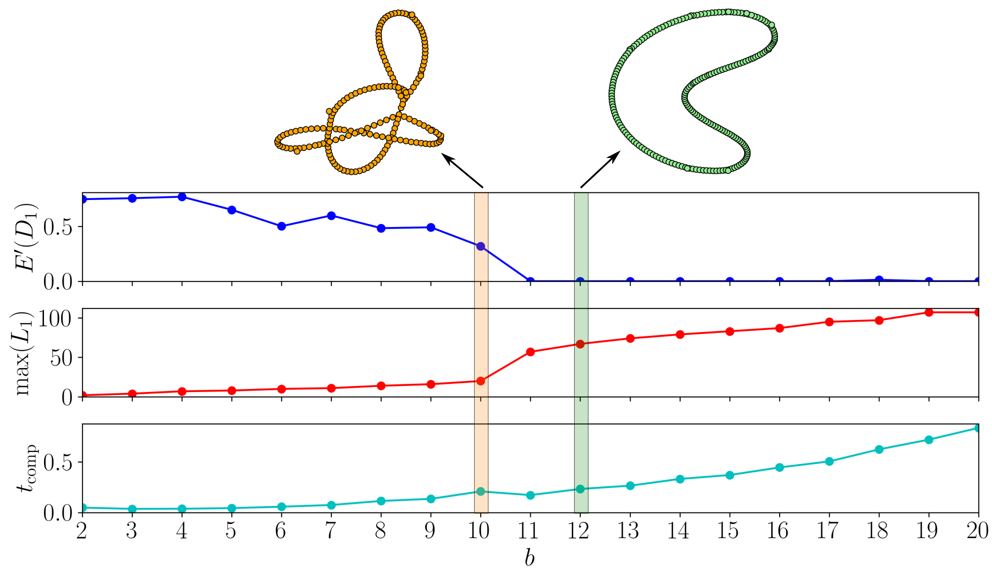


FIG. 17. Normalized persistent entropy $E'(D_1)$, the maximum lifetime $\max(L_1)$, and computation time t_{comp} for the CGSSN formed with dimension $n = 4$ and $b \in [2, 20]$ for the Rossler system in Eq. (7) with example CGSSNs shown at $b = 10$ and $b = 12$.

Appendix A: Coarse Graining Size Analysis

To determine the optimal binning size we we investigate how the structure of the resulting CGSSN changes as more states are used with b increasing. We considered $b \in [2, 20]$ as more than 20 bins per dimension becomes computationally expensive without increasing the performance (see Fig. 17). To summarize the shape of the network we use the maximum lifetime of one-dimensional features (loops) as $\max(L_1)$ and the normalized persistence entropy $E'(D_1)$ defined in Eq. (8) using the shortest unweighted path distance. The goal is to find a fine enough granularity (large enough b) that a periodic, noise-free signal will create a signal loop structure in the CGSSN. This loop structure should result with a persistent entropy of approximately zero. The idea behind this is based on a periodic attractor's SSR never intersecting if a suitably high dimension is selected.

We point the reader to our work in[58] for a comprehensive analysis to choosing a suitable embedding dimension for the problem. It was found that dimensions of $n = 4$ or 5 are suitable for most continuous systems. For the 23 dynamical systems selected a dimension $n = 4$ is greater than the dimension of the attractor and will be used unless otherwise stated. Let us first investigate a suitable number of bins b for the Rossler system defined in Eq. (7) with the $E'(D_1)$, $\max(L_1)$, and computation time t_{comp} calculated as b is increased from 2 to 20 shown in Fig. 17. This result show a sudden drop in $E'(D_1)$ and increase in $\max(L_1)$ from going from 10 to 11 bins. This is due the the granularity of the coarse-graining procedure being fine enough that the hypercubes do not capture multiple segments of the periodic flow. This is shown with the two CGSSNs at $b = 10$ and $b = 12$ where at $b = 10$ we have multiple intersections of the network while at $b = 12$ there are no intersections and we only have a single loop structure. Another characteristic is the exponentially increasing computation time t_{comp} as b increases. As such, we want to optimize the choice of b to capture the necessary complexity of the attractor while also minimizing the computation time. For this

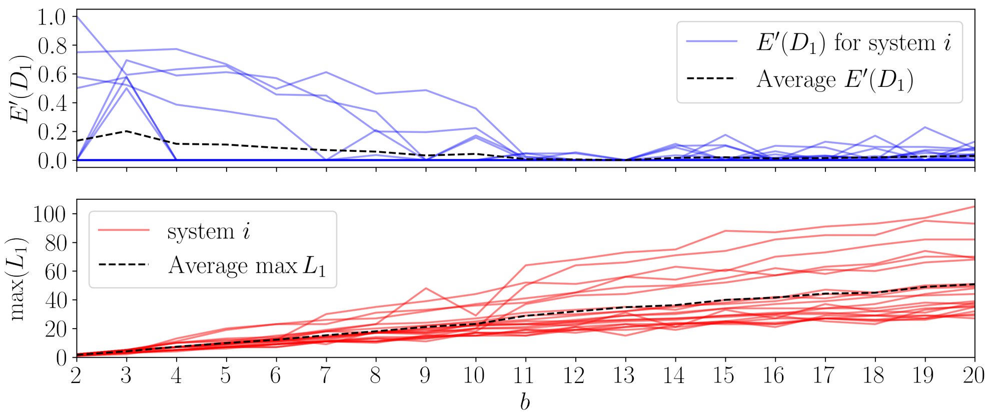


FIG. 18. Binning size analysis using the normalized persistent entropy $E'(D_1)$ and maximum lifetime $\max(L_1)$ for 23 dynamical systems listed in Table II with $b \in [2, 20]$.

example a suitable $b = 12$ would be the best choice.

The next question we want to ask is if $b = 12$ is a good option for other dynamical systems. To test this we again calculate the $E'(D_1)$ and $\max(L_1)$ for $b \in [2, 20]$ for the 23 dynamical systems listed in Table II. Figure 18 shows these statistics for all of the dynamical systems and it demonstrates that a choice of $b \in [11, 13]$ does work well for all of the dynamical systems with a drop in $E'(D_1)$. Based on this seemingly universal choice of b in this work we use $b = 12$ unless otherwise stated.

Appendix B: Data

In this work we heavily rely on a 23 dynamical systems commonly used in dynamical systems analysis. All of these systems are continuous flow opposed to maps. The 23 systems are listed in Table II. The equations of motion for each systems can be found in the python topological signal processing package `Teaspoon` under the module *MakeData* `https://lizliz.github.io/teaspoon/`. Specifically, these systems are described in the dynamical systems function of the make data module[70].

Each system was solved to have a time delay $\tau = 50$, which was estimated from the multiscale permutation entropy method[58]. The signals were simulated for $750\tau/f_s$ seconds with only the last fifth of the signal used to avoid transients. It should be noted that we did not need to normalize the amplitude of the signal since the ordinal partition network is not dependent on the signal amplitude.

Appendix C: Additional Results

Here we provide the additional SVM projections to visualize the dynamic state detection performance of the shortest path distances: unweighted shortest path, shortest weighted path, and weighted shortest path. Table I provides the corresponding average accuracies.

TABLE II. Continuous dynamical systems used in this work.

Autonomous Flows	**Driven Dissipative Flows**
Lorenz	Driven Van der Pol Oscillator
Rossler	Shaw Van der Pol Oscillator
Double Pendulum	Forced Brusselator
Diffusionless Lorenz Attractor	Ueda Oscillator
Complex Butterfly	Duffing Van der Pol Oscillator
Chen's System	Base Excited Magnetic Pendulum
ACT Attractor	
Rabinovich Frabrikant Attractor	
Linear Feedback Rigid Body Motion System	
Moore Spiegel Oscillator	
Thomas Cyclically Symmetric Attractor	
Halvorsen's Cyclically Symmetric Attractor	
Burke Shaw Attractor	
Rucklidge Attractor	
WINDMI	
Simplest Cubic Chaotic Flow	

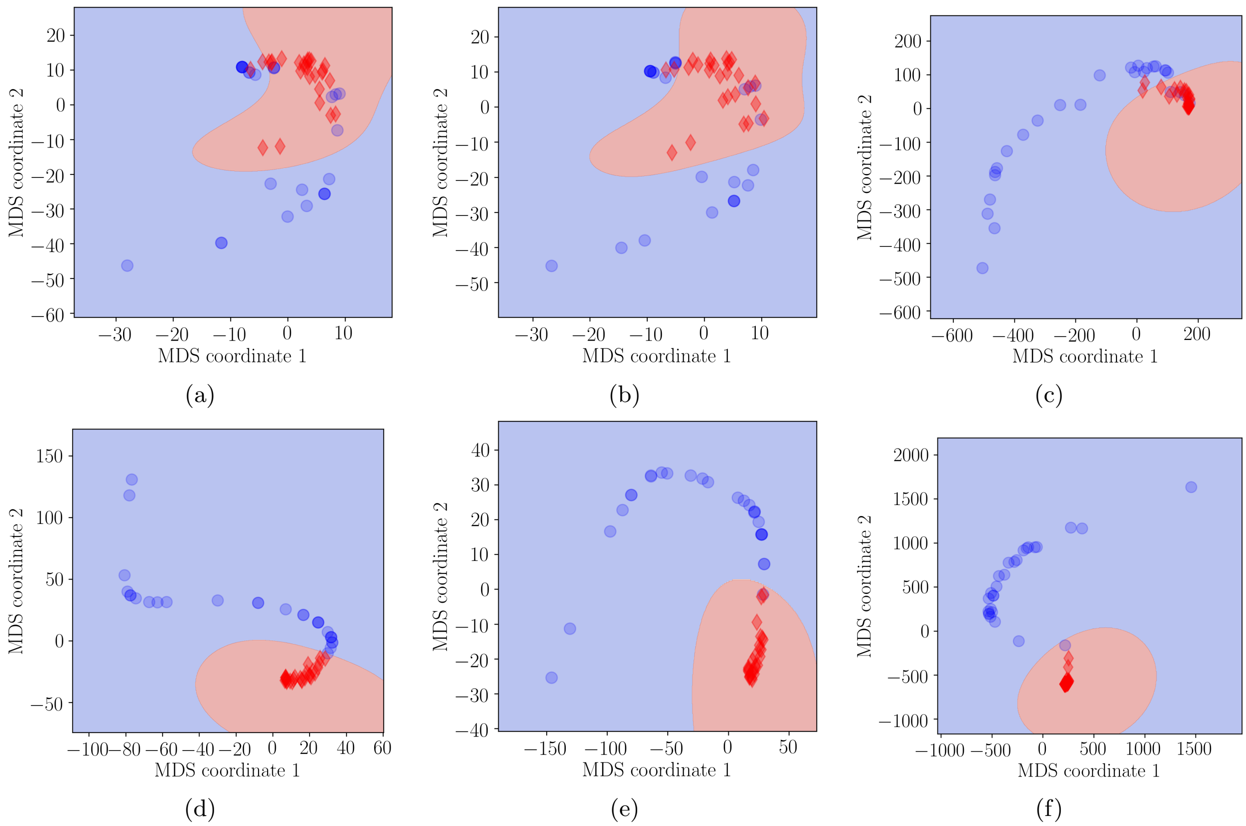


FIG. 19. Two dimensional MDS projection of the bottleneck distances between persistence diagrams of the chaotic and periodic dynamics with an SVM radial bias function kernel separation. This separation analysis was repeated for the OPNs and CGSSNs using the unweighted shortest path, shortest weighted path, and weighted shortest path distances. (a) Unweighted shortest path distance of OPN, (b) Shortest weighted path distance of OPN, (c) Weighted shortest path distance of OPN, (d) Unweighted shortest path distance of CGSSN, (e) Shortest weighted path distance of CGSSN, (f) Weighted shortest path distance of CGSSN.