Chaos
An Interdisciplinary Journal
of Nonlinear Science

REVIEW ARTICLE | MARCH 01 2023

Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology

Eugene Tan ; Shannon Algar ; Débora Corrêa ; Michael Small ; Thomas Stemler ;
David Walker

Check for updates

Chaos 33, 032101 (2023)
https://doi.org/10.1063/5.0137223

View Online Export Citation

Articles You May Be Interested In

Erratum to selecting embedding delays: An overview of embedding techniques and a new method using persistent homology [Chaos: Interdiscip. J. Nonlin. Sci. 33(3), 24 (2023)]

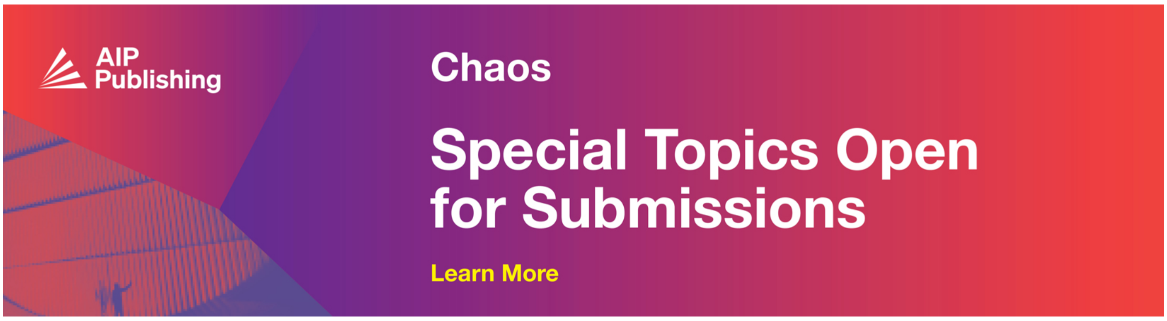
Chaos (September 2024)

Model adaptive phase space reconstruction

Chaos (July 2024)

Grading your models: Assessing dynamics learning of models using persistent homology

Chaos (December 2021)



AIP Publishing

Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology

Cite as: Chaos **33**, 032101 (2023); doi: 10.1063/5.0137223
Submitted: 1 December 2022 · Accepted: 7 February 2023 ·
Published Online: 1 March 2023

Eugene Tan,[1,a)] Shannon Algar,[1,2] Débora Corrêa,[1,3] Michael Small,[1,3,4] Thomas Stemler,[1] and David Walker[1]

AFFILIATIONS

[1] Complex Systems Group, Department of Mathematics and Statistics, The University of Western Australia, Crawley, Western Australia 6009, Australia
[2] Forrest Research Foundation, The University of Western Australia, Crawley, Western Australia 6009, Australia
[3] ARC Centre for Transforming Maintenance Through Data Science, The University of Western Australia, Crawley, Western Australia 6009, Australia
[4] Mineral Resources, CSIRO, Kensington, Western Australia 6151, Australia

[a)] **Author to whom correspondence should be addressed:** eugene.tan@uwa.edu.au

ABSTRACT

Delay embedding methods are a staple tool in the field of time series analysis and prediction. However, the selection of embedding parameters can have a big impact on the resulting analysis. This has led to the creation of a large number of methods to optimize the selection of parameters such as embedding lag. This paper aims to provide a comprehensive overview of the fundamentals of embedding theory for readers who are new to the subject. We outline a collection of existing methods for selecting embedding lag in both uniform and non-uniform delay embedding cases. Highlighting the poor dynamical explainability of existing methods of selecting non-uniform lags, we provide an alternative method of selecting embedding lags that includes a mixture of both dynamical and topological arguments. The proposed method, *Significant Times on Persistent Strands* (SToPS), uses persistent homology to construct a characteristic time spectrum that quantifies the relative dynamical significance of each time lag. We test our method on periodic, chaotic, and fast-slow time series and find that our method performs similar to existing automated non-uniform embedding methods. Additionally, n-step predictors trained on embeddings constructed with SToPS were found to outperform other embedding methods when predicting fast-slow time series.

Embedding methods are commonly used to analyze time series whose full system state cannot be fully or directly observed. However, most embedding methods require the careful selection of parameters to achieve a faithful reconstruction of the system dynamics. One common class of embedding methods—time delay embedding—requires careful selection of embedding lags. In this paper, we provide an outline of embedding theory and a collection of existing methods and principles for guiding the selection of embedding lags. Finally, we present an argument for the usage of non-uniform embedding and propose a new persistent homology-based method, SToPS, to inform the selection for multiple embedding lags.

I. THE CASE FOR EMBEDDING

Since the significant results proposed by Whitney[1] and Takens,[2] the ideas of mathematical embedding have pervaded through almost all aspects of the nonlinear dynamics literature. The related theorems were then subsequently formalized, codified, and discussed in the seminal paper "Embedology" by Sauer, Yorke, and Casdagli.[3] This paved the way for the development of numerous embedding techniques such as the method of derivatives,[4] time delay embedding,[2,5] and PCA embedding,[6] among others, that have been subsequently applied to a wide variety of study areas. Today, the embedding approach remains an invaluable tool in the study of nonlinear time-series analysis.

A time series $x(t) \in \mathbb{R}^m$ may be generically viewed as the product of a data-generating process consisting of successive, though not always regular, observations of some dynamical process with state $\vec{s}(t) \in \mathbb{R}^n$ via a measurement function $h : \mathbb{R}^n \to \mathbb{R}^m$. Typically, $m < n$ as the full state of the underlying dynamical process cannot be observed. For the purposes of illustration, we will consider in this paper the simplest case where $x(t)$ is scalar (i.e., $m = 1$).

The main goals in the time series analysis often fall into two main categories: system identification or classification and prediction. The aims of the former are focused on characterizing and understanding the dynamics and operating mechanisms of the underlying dynamical system. This can range from the study of system invariants such as Lyapunov spectra[7,8] and correlation dimension[9–11] to the bifurcation analysis.[11,12] The latter task of time-series prediction has a more practical aim that is clearly stated in its name: "given some history $x(t_0) \cdots x(t)$, find the best predicted estimates of $x(t + \tau)$." It is worth clarifying that these two areas are not exclusive and often work synergistically. However, we will focus on the latter problem of time series prediction in this paper.

Though simple in its aim, practically fulfilling the task of time series prediction presents many challenges. Outside the study of simple toy models, many interesting systems exhibit high-dimensional and also chaotic behavior.[13–15] For the cases of time delay systems, the dimension of the system dynamics may not even be finite. The potential inaccessibility of the full system state also adds to the challenge. As such, it is common to reframe the time series prediction problem in terms of the measurement function $h : \mathbb{R}^n \to \mathbb{R}^m$ where $m < n$. Thus, time series prediction can be rewritten: "given some observed history $h(x(t_0)), \ldots, h(x(t))$, predict the future state $x(t + \tau)$." This can be framed in terms of probability theory with aim to calculate the conditional probability $P(x(t + \tau)|h(x(t_0)), \ldots, h(x(t)))$.

Given observational restrictions, it is desirable for any time-series predictor to extract and utilize as much information that is contained within the observed time series $x(t)$. Therein lies the value of embedding, vis-à-vis dimension augmentation. Fortunately, under sufficient precision and noise-free assumptions, Takens' embedding theorem guarantees that a time delay embedding with dynamics Φ defined in a space of sufficiently high dimension $\mathbb{R}^d$ constructed from scalar observed time series is generically diffeomorphic to the full state space dynamics of the underlying dynamical system.[2] In essence, one can reconstruct a proxy image of the full state dynamics using only part of the observed system variable. The usual approach to achieve this is to embed the observed time series into a sufficiently large dimension containing the "full" state dynamics, learn the state space dynamics or vector field using one's favored modeling tool (e.g., neural networks,[16,17] reservoir computing[18–20] support vector machines,[21,22] etc.), and predict forward, performing all required calculations in the new embedded space.

Despite the elegance and utility afforded by embedding theorems, their reliance on infinite data precision, length, and noise-free signals pose practical problems. The effectiveness of embedding is highly dependent on the choice of embedding parameters used to augment the data.[23,24] For example, in the case of time delay embedding, the selection of time lag[25] and embedding dimensions[3,26] can have a profound effect on the quality of resulting reconstruction.

Thus, we may summarize the main challenge of embedding as the following question: "How do we select good embedding parameters?" This will be the focus of our discussion. Numerous methods have been proposed to tackle this problem ranging from purely statistical and dynamical arguments such as mutual information[11,25] and continuity statistic[27] to more purely topological arguments like distortion,[28] noise amplification,[28,29] and fill-factor.[30] In many cases, each of these methods only performs well for specific types of systems and not others.

This paper has three main objectives. First, to provide a simple overview of the challenges of selecting good embedding parameters. Second, to collate and compare the various popular methods across the dynamics-topology spectrum that have been proposed to tackle the problem of embedding parameter selection. We will focus on the particular case of optimizing time delay embedding. Finally, to present a different approach based on the growing field of persistent homology—the significance score—that attempts to incorporate both dynamical and topological arguments into selection of embedding parameters.

This paper is structured as follows. We begin in Sec. II by providing an overview of embedding theory and various common embedding methods. This is followed by a short discussion on guiding principles on the selection of embedding parameters in Sec. III. Sections IV–V introduce several embedding parameter selection methods for both uniform and non-uniform delay embedding. Finally, we present SToPS in Sec. VIII, a persistent homology approach to embedding parameter selection, which is our contribution to the embedding parameter selection problem.

II. EMBEDDING METHODS

Multiple embedding methods exist to perform state space reconstruction. However, they all aim to perform a similar task, to augment an observed time series into a high enough dimension that is useful for describing the underlying dynamics. For completeness, we provide a brief overview on embedding and three common embedding methods, time delay embedding, derivatives embedding, and global principal value embedding. However, the ideas present in this paper will focus on the selection of time delay embedding parameters. A deeper discussion on other embedding methods can be found elsewhere.[28]

A. Embedding theory

For a given dynamical system with state $\vec{s}(t)$ with dynamics on state space $S \subseteq \mathbb{R}^d$ and evolution operator f such that,

$$\vec{s}(t + T) = f_T(\vec{s}(t)), \tag{1}$$

we can define a measurement function $h : \mathbb{R}^d \to \mathbb{R}$ that simulates the process of observing the system and extracting information evaluated at given time steps to produce a time-series,

$$x(t) = h(\vec{s}(t)). \tag{2}$$

An embedding can be defined as a transformation $\Psi : \mathbb{R} \to \mathbb{R}^m$ that augments the dimension of the time series using observed values across some window of time $[t_1, t_2]$. For example, uniform delay embedding is given by the following equation:

$$\Psi(x(t)) = \vec{x}(t) = (x(t), x(t-\tau), \ldots), \tag{3}$$

where $\vec{x}(t)$ is the embedding vector with dynamics defined in a reconstructed state space $X \subseteq \mathbb{R}^m$ and a transformed evolution operator F. If Ψ is a valid embedding, Takens' embedding theorem guarantees that generically, there exists a diffeomorphism $\Phi : S \rightarrow X$ that preserves the dynamics of the system such that,

$$F = \Phi \circ f \circ \Phi^{-1}. \tag{4}$$

Learning the dynamics along the reconstructed state space F is equivalent to learning the true system dynamics f (see Fig. 1). Therefore, the task of time series prediction using embedding simplifies to learning the evolution operator F where

$$x(t+T) = \Psi^{-1} \circ F_T \circ \Psi(x(t)). \tag{5}$$

B. Time delay embedding

First described by Packard *et al.*,[4] time delay embedding involves the augmentation of a scalar time series $x(t)$ into a higher dimension through the construction of delay vector $\vec{x}(t)$ given as

$$\vec{x}(t) = (x(t), x(t-\tau), \ldots, x(t-(m-1)\tau)), \tag{6}$$

where the embedding parameters to be selected are the delay lag τ and embedding dimension m. According to the guarantees of Takens' theorem, any value of τ will yield a valid embedding given sufficiently large m and measurement values of infinite precision. However, this is not achievable in practice, and different selections of delay lag and embedding dimension can yield varying results. A further discussion of this is given in Sec. III A.

We also note that the task of selecting ideal delay lag τ and embedding dimension m is not unique to time delay embedding. Selecting values of τ and m is also key decision in other time series analysis methods such as permutation entropy[31] and ordinal partition networks.[32,33] In both of these instances, a delay vector is constructed and represented by an encoding based on the size order of each component. The time series may then be viewed as transitions between different encoding states and used for further analysis.

Many automated time-series prediction methods such as recurrent neural networks[16,17] and reservoir computing[18–20,34] may also be related to delay embedding. In both cases, input time series is fed into a dynamical network that contains some notion of memory. The forward propagation of this memory of past states on future states effectively acts as a time delay embedding with small delay lag τ and large embedding dimension m.

C. Derivatives embedding

The embedding method of derivatives reconstructs an embedding vector using successively increasing order of time derivatives from the observed time series.[4] This is given by

$$\vec{x}(t) = \left(x(t), \frac{dx(t)}{dt}, \ldots, \frac{d^m x(t)}{dt^m}\right). \tag{7}$$

Derivatives are taken via numerical approximations. The derivatives embedding method is a valid embedding for sufficiently large m if one is able to accurately calculate the required derivatives.

D. Integral-differential embedding

One weakness of the derivatives embedding approach is the need to evaluate numerical derivatives from data. While this may be acceptable for the first derivative, approximations of successive higher order derivatives are generally inaccurate as the signal to noise ratio tends to be negatively impacted. This is true even for the cases of very clean datasets.

An alternative to derivatives embedding is integral-differential embedding.[35] This approach avoids the calculation of successive higher order derivatives by replacing the second-order derivative with an integral instead. This yields the following embedding construction:

$$\vec{x}(t) = \left(\int_{-\infty}^{t} x(t) - \langle x(t)\rangle_t \, dt, x(t), \frac{dx(t)}{dt}\right), \tag{8}$$

where the first component is first set to zero mean before integration. The usage of a first-order integral and numerical derivative results in a degradation of the signal-to-noise ratio by only one order each for first and third embedded components. This is in contrast with the derivatives embedding approach where each successive numerical derivative has a signal-to-noise ratio that is degraded with increasing orders of magnitude. However, the integral-differential embedding approach suffers from same noise effects as the pure derivatives method for higher dimensional embedding. This limits its applicability to systems where system dynamics are presumed to be high dimensional.

E. Global principal value embedding

The method of principal value embedding was proposed by Broomhead and King[6] as a modified alternative to time delay embedding using the theorems by Takens. This method draws upon the ideas of the principal component analysis[36] to find an ideal rotation of time delay embedding with a sufficiently high dimension. Given a time series $x(t)$ of length N_T and a sliding window of length M, we can construct a collection of $N = N_T - (M-1)$ delay vectors,

$$\mathbf{X} = N^{-1/2} \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_N \end{bmatrix}, \tag{9}$$

where $\vec{x}_i$ is the delay vector constructing using the ith value in the time series as the first component,

$$\vec{x}_i = (x(t_i), x(t_{i-1}), \ldots, x(t_{i-(M-1)})). \tag{10}$$

An $m \times m$ covariance matrix C can be calculated from $\mathbf{X}$. The elements C_{ij} of this matrix can be simply given as

$$C_{ij} = \langle x(t)\, x(t + (i-j))\rangle_t, \tag{11}$$

where $\langle \cdots \rangle_t$ denotes a time average. The principal components of C are then found by calculating its respective eigenvalues and eigenvectors. Taking the first d principal components corresponding to the desired number of embedding dimension, the eigenvector

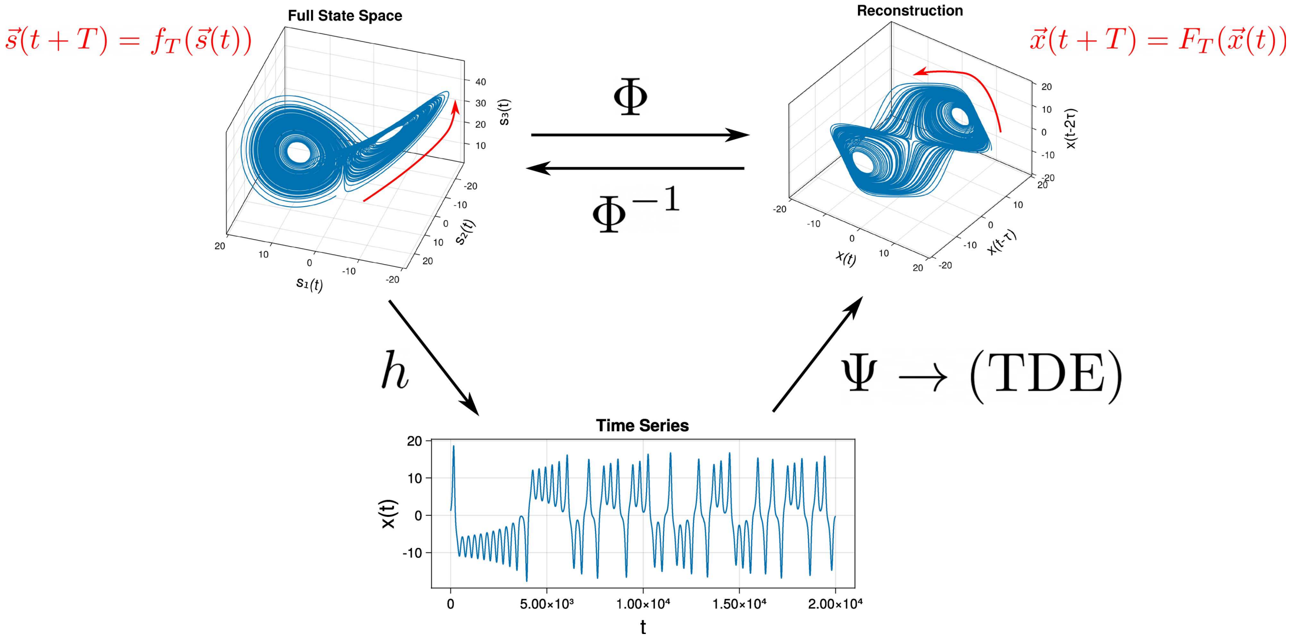


FIG. 1. Schematic of the embedding process and the relationship between its components.

matrix can be used to calculate a projection of **X** corresponding to the final embedded coordinates. Readers are advised to refer elsewhere[6,28] for more details.

Principal component value embedding essentially aims to distill and simplify high-dimensional delay embedding (usually obtained by taking a large number of lagged components) into a lower dimensional subspace. The remaining subspaces are argued corresponding to component directions with little dynamical variation and importance. One application of this method was as an attempt to simplify the selection of the optimal embedding dimension,[6] where the ideal embedding dimension m corresponds to the number of singular values that are distinctly greater than some "noise floor." However, this approach has received several criticisms.[37,38] The main of which arguing that the onset of a plateau noise floor can be attributed to precision and noise strength in the data, rather than the importance of the corresponding eigenvector direction.

Within the general context of embedding, Paluš and Dvorák test the quality of delay embedding with reduced dimension obtained using the first k principal eigenvector directions.[38] The authors show that reduced dimension embeddings' estimates of dynamical invariants such as the correlation dimension vary with different time delays and the number of components. They argue that the usage of principal components of the covariance matrix is restricted to linear correlations. Therefore, while components in embedding may be independent in the linear sense, they may still be nonlinearly dependent. Instead, the truncation of embedding dimensions can result in the exclusion of important nonlinear components.

The inclusion of large time lags within a given principal component direction also may not make much sense for chaotic systems where temporally close observations decorrelate exponentially in time. From the perspective of selecting time lags, each principal component will almost invariably contain contributions from all M possible lagged components. Apart from the dimensional reduction argument[38] (of which care must be taken in its interpretation), global principal value embedding does not present any significant difference to the general time delay embedding with large embedding dimension m.

III. EMBEDDING CONSIDERATIONS

A. Embedding quality

As previously discussed, the theoretical guarantees of Takens' fail in the presence of finite precision and noise[23,25] leading to the concept of "optimal" embedding parameters. The existence of such an "optimal" set implies that not all embeddings are of equal quality. However, this requires a measure by which embedding quality can be compared against. An attempt to quantify embedding quality was studied extensively by Casdagli[28] and Potapov.[39]

There are large variations in the definition of embedding quality such as those based on information theoretic arguments,[25] prediction tasks,[28] and attractor topology.[30,40] It is worthwhile to note that the "optimality" of a set of parameters is dependent on the task that the embedding is being used for. As such, while one may find similar results between methods based on different notions of embedding quality, disagreement between results will likely always be present.[11,23,28] Therefore, it is better to avoid the claim that a particular set of parameters are more favorable unless there are dynamical and topological reasons within the data itself that support it. However, we will highlight in this section the general considerations that are often used in defining the quality of embedding.

The different notions of embedding quality can be summarized in two broad categories or arguments, prediction-based and topological arguments. Prediction-based notions of embedding quality can be seen to be inspired by application of embeddings in the context of time-series prediction. Fundamentally, good embeddings should enable better predictions.[28,39]

In time series prediction tasks, the presence of an unknown measurement function h and noisy data introduces some degree of uncertainty to the inference of the real system state $s(t)$. Casdagli argued that an ideal embedding should minimize the uncertainty of inferring the true state $\vec{s}(t)$ given a position $\vec{x}(t)$ in the reconstructed state space. In essence, the inverse transformation Φ^{-1} applied to constructed delay embedding in the presence of noise should have little ambiguity on the true state of the system, if $\vec{s}(t)$ could be fully known.[28] This robustness to noise and low ambiguity should in theory be beneficial for time series prediction and also forms the basis of the ideas of noise amplification and distortion used to quantify embedding quality. In poor embeddings, such as those whose attractor manifolds are laminar with little separation, the effect of noisy perturbations across manifold layers results in significant uncertainty of the true state $s(t)$, making time series prediction difficult.

The information theoretic arguments for choosing embedding parameters (e.g., autocorrelation, minimum mutual information, and continuity statistics) are also closely related to the ideas of prediction. These methods generally try to maximize the amount of new information incorporated in each delay dimension with the aim that it will provide more information of the true system state $\vec{s}(t)$ and aid in time series prediction.

The other broad category of defining embedding quality are those based on topological and geometrical arguments. Many of these methods focus on the study of the attractor structure and distribution of the manifold in its ambient state space. In essence, good embedding with respect to topology and geometry should aim to be well spaced out and unfolded in its ambient space.[11,30,40] This notion of quality has parallels with noise amplification arguments of Casdagli. Some methods based on geometrical arguments include statistics such as the fill factor[30] and displacement from diagonal.[40] Ultimately, many of the considerations outlined above for determining the ideal lag and embedding dimension for time delay embedding can be summarized with the concepts of irrelevance and redundancy.[28,41]

B. Irrelevance and redundancy

The selection of time lag τ and embedding dimension m are the main challenges when constructing time delay embedding. There is uncertainty on the relative importance between embedding lag and dimension. Furthermore, it has also been proposed that these embedding parameters may not be independent. Instead the quantity $\tau_w = m\tau$, termed the embedding window, has been proposed as a more important parameter to optimize.[28,41] However, for sufficiently large embedding dimension, it could be argued that selection of τ_w may be simplified to an appropriate selection of τ.

The selection of the embedding window τ_w (and by extension embedding lag τ) may be summarized by a notion that it must be neither too short (redundance) nor too long (irrelevance). This explanation applies for chaotic or aperiodic signals. However, periodic signals may be successfully embedded with large lags τ where the effective lag τ^* is related to the period T,

$$\tau = nT + \tau^*, \quad n \in \mathbb{N}. \tag{12}$$

Embeddings with high redundance result in trajectories that lie in layers roughly parallel to each other (e.g., close to the diagonal). In the presence of sufficient noise, the clear separation between layers is affected. This results in a greater degree of uncertainty of the true system state $\vec{s}(t)$, given some noisy observation $\vec{x}(t)$ in reconstructed state space.

Similarly, embeddings with high irrelevance contain components that are highly decorrelated with the true state.[28] This is also unfavorable as it may introduce unwanted crossings between trajectories in the reconstructed manifold. Therein lies the Goldilocks problem of selecting an embedding window τ_w that is neither too large or too small.

IV. UNIFORM DELAY EMBEDDINGS

The simplest form of delay embedding is the case of uniform delays where single constant values for τ and m are selected. In this case, embedding vectors are selected with uniformly increasing time delays as given in the following equation:

$$\vec{x}(t) = (x(t), x(t-\tau), \dots, x(t-(m-1)\tau)). \tag{13}$$

Due to the debate between the selection priority of τ and m, multiple methods have been proposed to simultaneously estimate both parameters. Some methods include those of Gao and Zheng,[42,43] characteristic lengths,[23] and Schuster (wavering product).[44] An overview of these methods is provided in Sec. V.

Other common methods attempt to simplify the problem by assuming the independence of τ and m and choose to estimate both values separately. Generally, an embedding τ is first determined using a choice of various measures. Once selected, uniform delay vectors of increasing dimensions are constructed and tested with algorithms such as the Grassberger–Procaccia[10] algorithm or false nearest neighbors[26,45,46] until convergence is achieved. The length of delay vector when stability is reached is used to decide the embedding dimension m.

The methods that are used for determining embedding lag τ vary from simple heuristics to more complex statistical arguments. One common heuristic is the selection of embedding lag as one quarter of the signal period (or quasi-period for chaotic signals). Delving into more statistically grounded arguments, autocorrelation[11] and its nonlinear generalization, mutual information,[47] continuity statistics,[27] and L-statistics[29] are occasionally used to determine good values for τ. A comprehensive overview and further discussion on these methods is provided in Secs. VI–VII.

V. SIMULTANEOUS OPTIMIZATION OF UNIFORM EMBEDDING PARAMETERS

In contrast to many of the current methods that involve the selection of τ and m independently, several embedding methods have been proposed that simultaneously estimate both values using a single measure. This measure is often calculated across multiple lags and repeated for increasing m. A value for m is first selected according to some criterion followed by selection of τ. Detailed steps on implementation of these methods are outlined by Celluci.[23]

A. Method of Gao and Zheng and characteristic lengths

The embedding method proposed by Gao and Zheng is based on the incidence of false nearest neighbors.[42,43] False nearest neighbors can be attributed to either redundancy (insufficiently unfolded) and irrelevancy (spurious intersections in the attractor). The method proposed by Gao and Zheng operates on the notion that the separation distance and proportion of false nearest neighbors should be minimized in an ideal embedding.

Consider a pair of points in embedded space $\vec{x}_i, \vec{x}_j$ and their evolution k steps into the future $\vec{x}_{i+k}, \vec{x}_{j+k}$. Points that are false nearest neighbors will tend to separate faster than real neighbors as the attractor unfolds in a time delay embedding. As a result, the ratio between their distances $|\vec{x}_{i+k} - \vec{x}_{j+k}|/|\vec{x}_i - \vec{x}_j|$ will be larger for pairs of false nearest neighbors and approximately equal to 1 for real neighbors. Gao and Zheng then propose the following measure Λ to optimize the embedding parameters:

$$\Lambda(k, m, \tau) = \frac{1}{N_{ref}} \sum_{i,j} \ln \frac{|\vec{x}_{i+k} - \vec{x}_{j+k}|}{|\vec{x}_i - \vec{x}_j|}, \tag{14}$$

where N_{ref} is the number of randomly sampled point pairs over which the distance ratio is averaged. There are several additional restrictions on the selection of point pairs $\vec{x}_i, \vec{x}_j$. First, the initial separation of these points should satisfy $|\vec{x}_i - \vec{x}_j| \leq r$, where r is a small selected threshold, i.e., the initial separation of points should be small enough such that the calculation of growing separation is sensible. Second, the selection of pairs of points should not have an intersecting Theiler window $|i - j| > l_{\text{Theiler}}$, where $l_{\text{Theiler}} \in \mathbb{N}^+$. This is done to prevent unwanted correlations between points on the same local trajectory.[48,49] Finally, the constant k should not be too large and selected with respect to the natural time scale of the system dynamics.

To identify good embedding parameters, profiles of $\Lambda(\tau)$ are calculated for increasing values of embedding dimensions m. The value of m that corresponds to the largest decrease across the profile $\Lambda(\tau)$ is selected as the embedding dimension. The embedding lag τ is then selected as the first minimum of $\Lambda(\tau)$.

B. Characteristic lengths

The method of characteristic lengths is an extension of Gao and Zheng[23] that attempts to solve the problem of selecting an evolution time k. Instead of arbitrarily selecting k, a characteristic length $J(m, \tau)$ describing the natural spatial scale of the system attractor is calculated,

$$J(m, \tau) = \langle |\vec{x}_i - \vec{x}_j| \rangle, \tag{15}$$

where $\langle \cdots \rangle$ denotes an average over sampled pairs of points of the attractor. The characteristic length is then used to calculate the separation time $T_J(\vec{x}_i, \vec{x}_j)$ defined as the time taken for pairs of nearest neighbors to diverge by some proportion of the characteristic length $J(m, \tau)$. For real neighbors, T_J will converge to a value related to the Lyapunov exponent of the system with increasing embedding dimension m, while false nearest neighbors will result in a smaller value T_J as trajectories quickly separate. The new measure that is used to determine the embedding parameters is given by

$$C(m, \tau) = \frac{1}{N_{ref}} \sum_{i,j} T_J(\vec{x}_i, \vec{x}_j), \tag{16}$$

where N_{ref} is the number of sampled pairs of nearby neighbors. The values for m and τ that maximize $C(m, \tau)$ are selected as the embedding dimension and lag.

C. Wavering product

The wavering product[44] is similar to that of Gao and Zheng and characteristic lengths in that all are based around the concepts of nearest neighbors. The authors propose that good embeddings should preserve the correspondence between the order of nearby neighbours of a given reference point (i.e., the order of neighbors sorted according to distance from some reference point $\vec{x}_i$ should be preserved). This is done by comparing the order of N nearby neighbors of a point $\vec{x}_i$ between a given embedding Φ_k, whose ordered sequence neighbours are given by

$$X_{\Phi_k} = \{\vec{x}_{i,1}, \ldots, \vec{x}_{i,N}\}_{\Phi_k}, \tag{17}$$

and its projection onto its next order embedding Φ_{k+1} (by increasing m or τ) with the sequence given by

$$Z_{\Phi_{k+1}} = \{\vec{z}_{i,1}, \ldots, \vec{z}_{i,N}\}_{\Phi_{k+1}}. \tag{18}$$

Here, $\vec{x}_{i,n}$ corresponds to the nth nearest neighbor of the ith reference point $\vec{x}_i$. The projection $\vec{z}_{i,n}$ corresponds to the same neighbor data point $\vec{x}_{i,n}$ whose position is recalculated from next order embedding Φ_{k+1}.

Similarly, comparisons can also be made into a projection into an embedding of lower order (by decreasing m or τ) giving a new set of ordered points,

$$V_{\Phi_{k-1}} = \{\vec{v}_{i,1}, \ldots, \vec{v}_{i,N}\}_{\Phi_{k-1}}. \tag{19}$$

Ideally, a good embedding should preserve a one to one correspondence in these ordered sequences. This will yield a value equal to 1 for the following ratios:

$$\frac{|\vec{x}_i - \vec{z}_{i,n}|}{|\vec{x}_i - \vec{x}_{i,n}|}, \text{ and } \frac{|\vec{x}_i - \vec{x}_{i,n}|}{|\vec{x}_i - \vec{v}_{i,n}|}. \tag{20}$$

The method presented by Schuster and Liebert propose the following measure as the product of the above two ratios,

$$W_i(m\tau) = \prod_{n=1}^{N} \left\{ \left(\frac{|\vec{x}_i - \vec{z}_{i,n}|}{|\vec{x}_i - \vec{x}_{i,n}|} \right) \cdot \left(\frac{|\vec{x}_i - \vec{x}_{i,n}|}{|\vec{x}_i - \vec{v}_{i,n}|} \right) \right\}. \tag{21}$$

The measure to be optimized is given by the average over N_{ref} randomly sampled reference reference points,

$$W(m,\tau) = \ln\left(\frac{1}{N_{ref}}\sum_{i=1}^{N_{ref}} W_i(m,\tau)\right), \tag{22}$$

with m being selected as the dimension that achieves the limiting behavior of W and τ corresponds to the first minimum of the resulting profile.

VI. SELECTING UNIFORM EMBEDDING LAG

In contrast to embedding methods in Sec. V that utilize a single measure to optimize the selection of embedding lag and dimension, the most practiced approach still selects the embedding lag τ and dimension m independently according to separate metrics. Here, we will focus on the methods used to select the lag for time delay embedding. This lag may then be used to construct delay embeddings of arbitrary dimension.

The simpler case of uniform delay embedding requires the selection of a singular value of τ that is increased in multiples to construct the required delay vector. The methods used to inform the selection of delay lag τ mirror the broad dichotomy in notions of embedding quality outlined in Sec. III A. In this section, we divide the various methods for optimizing embedding lag into two categories: the first based on dynamical and information theoretic arguments and the second based on topological arguments. We also discuss the topic of non-uniform embedding in Sec. VII.

Methods rooted in a dynamical approach can be interpreted as focusing on the mechanism behind the data-generating process and statistical relationships between measurements. Information theoretic and statistical approaches are also included in this category. The methods that we will review in this category include the autocorrelation, minimum mutual information, and quarter period. This is in contrast to those in the second category whose methods are more topological. Some examples include the fill-factor[30] and noise amplification.[28,29]

A brief note on embedding dimension: There are several methods that are used to identify the required embedding dimension such as the false nearest neighbors (FNNs)[26,45] and the Grassberger–Procaccia algorithm[10] used to estimate the correlation sum and dimension. Other invariants similar to correlation dimension such as the Kaplan–Yorke dimension[50] and box-counting dimension[51] are also often used to determine the embedding dimension. These results are usually used in conjunction with Whitney's theorem stating that any d-dimensional manifold (such as an attractor) can be embedded in at least $m > 2d + 1$ dimensions.[1] However, it has been noted that this direct application has its flaws as Whitney's theorem is only proved for integer dimensions d, which is rarely the case for a majority of systems of interest, such as those exhibiting fractal and chaotic behavior. However, we note that an extension of Whitney's theorem to generalize the inequality to the box counting dimensions ($m > 2D_F$) was given by Sauer *et al.*[11,52]

A. Dynamical approaches

1. Autocorrelation and minimum mutual information

Many commonly analyzed dynamical systems tend to exhibit chaotic behavior where nearby trajectories rapidly diverge and quickly become uncorrelated. The method of autocorrelation is based on the idea that each component in reconstruction should include as much new information regarding the true state as possible. It has been suggested that components in the delay vector should aim to minimize the correlation.[11] This has similar effects to minimizing the redundancy of reconstruction.

For delay embedding, the lag corresponding to the first minimum of the autocorrelation function is taken as the embedding time lag. Alternatively, the decay time to $1/e$ of the autocorrelation signal has also been proposed.[11] A variation of this approach based on the first root of the mean local autocovariance has also been proposed as a robust alternative to the minimum autocorrelation approach.[53]

One weakness of autocorrelation is its inability to account for non-stationarities in the time series (e.g., drifts in phase, frequency, and magnitude). Additionally, its application is limited to linear signals.[54] In all but simplest cases, dynamical systems exhibit some level of non-stationary and nonlinear behavior. Fraser and Swinney[25] proposed that the mutual information between the system be used in place of autocorrelation. In their original paper, Fraser and Swinney first provide a geometrical interpretation to complement the theoretic arguments for mutual information. Namely, consider a set of points whose values in one component x lie within some fixed window. From this set, track their positions τ steps into the future and calculate the distribution of values $p_\tau(x)$ in the same component for the same set of points. A value of τ that results in a wider distribution $p_\tau(x)$ should correspond to a good lag, which also corresponds to small values in the mutual information.

The mutual information $I(\tau)$ can be interpreted as the nonlinear analog of the autocorrelation function,

$$I(\tau) = \int P(x(t), x(t+\tau)) \log_2\left(\frac{P(x(t), x(t+\tau))}{P(x(t))P(x(t)+\tau)}\right) dt,$$

where $P(x(t))$ is the probability of observing a state $x(t)$ at any given time and $P(x(t), x(t+\tau))$ is the joint probability defined similarly for both time t and a future time $t+\tau$. Drawing from information theory, mutual information $I(\tau)$ aims to quantify the amount of information about a future state at time $t+\tau$ that is contained in an observation at time t. High levels of mutual information for a given lag τ imply a high degree of correlation between states and will result in higher redundancy for delay reconstruction.

The strengths of the minimum mutual information and autocorrelation lies in its ability to provide reasonable estimates for lag with relatively simple and quick computation. However, there are no guarantees for the existence of a clear minimum for a given mutual information profile $I(\tau)$.[55] Additionally, calculating mutual information requires the numerical estimation of probability density functions $P(x(t))$ and $P(x(t), x(t+\tau))$ and, thus, requires consideration regarding optimal histogram bin size and data length requirements.[56,57] Numerous alternative methods to more effectively estimate mutual information have been proposed including the usage of adaptive binning,[25,58] kernel density estimators,[59] and k-nearest neighbors.[56]

2. Quarter of period

A commonly used heuristic for selecting an embedding lag is to set τ to be quarter of the most dominant period in the signal.[60] This approach allows the natural time scale of the system dynamics to be encoded within the embedding procedure. This heuristic is inspired from the problem of embedding a sine wave in 2D $x(t) = \sin(\omega t)$. In this case, $\tau = \frac{2\pi}{4\omega}$ yields a 2D delay embedding that is the most circular with other values resulting in elliptical trajectories instead. However, this heuristic cannot be directly applied to chaotic systems where signals are aperiodic. Instead, an estimation of some form of pseudo-period is required, which will be the focus of our proposed method in this paper.

B. Topological approaches

1. Fill-factor

The fill-factor approach first proposed by Buzug *et al.*[30] is an entirely geometrical approach to calculate the quality of a given embedding. This method assumes that an ideal embedding should be able to unfold an attractor and maximize the separation between the trajectories. The authors argue that such an embedding optimally utilizes the ambient space and reduces the ambiguity of the true state of the system for different points in the reconstructed state space.

The fill-factor is calculated by first sampling $m+1$ random points from an m dimensional delay embedding of the data. A reference point $\vec{x}_r$ is then selected from this collection and the corresponding relative distance vectors can be calculated,

$$\vec{d}_i(\tau) = \begin{bmatrix} x_i(t) - x_r(t) \\ x_i(t-\tau) - x_r(t-\tau) \\ \vdots \\ x_i(t-(m-1)\tau) - x_r(t-(m-1)\tau) \end{bmatrix}. \tag{23}$$

The corresponding $m \times m$ matrix can then be expressed as

$$M(\tau) = (\vec{d}_1, \vec{d}_2, \ldots, \vec{d}_m), \tag{24}$$

and the volume of the resulting parallelepiped is given by calculating the determinant of M,

$$V(\tau) = \det(M(\tau)). \tag{25}$$

The final expression for the fill-factor is given by calculating the average volume over a collection of randomly sampled parallelepipeds $V_i(\tau)$, normalized by the range of the sampled data points,

$$f = \log\left(\frac{\frac{1}{N}\sum_{i=1}^{N} V_i(\tau)}{(\max_k x(t_k) - \min_k x(t_k))^m}\right). \tag{26}$$

The authors recommend the selection of τ that maximises the fill-factor f over the interval $\tau \in (0, T_c/2)$, where T_c is the characteristic recurrence time. The value of T_c is given by

$$T_c = \frac{1}{\omega_c}, \tag{27}$$

where ω_c is the most dominant frequency from the power spectrum of the time series.

2. Noise amplification

Noise amplification was a measure proposed by Casdagli in an attempt to quantify the quality of an embedding.[28,39] This is supported by the notion that a good embedding should be useful in performing predictions. Additionally, good embeddings should be able to still perform relatively well even in the presence of noise. Noise amplification for a given embedding $\vec{x}(t)$ is defined with respect to predictability of the system T steps into future under the presence of noise. Generally, this is given by

$$\sigma(T, \vec{x}) = \lim_{\epsilon \to 0} \sigma_\epsilon(T, \vec{x}), \tag{28}$$

where

$$\sigma_\epsilon(T, \vec{x}) = \frac{1}{\epsilon}\sqrt{\mathrm{Var}[x(T)|B_\epsilon(\vec{x})]}. \tag{29}$$

Here, $\mathrm{Var}[x(T)|B_\epsilon(\vec{x})]$ corresponds to the conditional variance of T step predictions into the future in $\mathbb{R}$ from an initial condition $\vec{x}$ in embedding space $\mathbb{R}^m$ contaminated with added small observation noise ϵ. In this case, it is assumed that predictions have no model errors. This condition may be fulfilled by choosing nearby neighbors in the embedding $\mathbb{R}^m$ as a proxy for noisy initial conditions.[29]

Finally, the noise amplification quantity $\sigma(T, \vec{x})$ is averaged over a collection of reference points sampled across the time series in order to calculate the noise amplification value σ. Embeddings with high noise amplification imply that nearby neighbors in embedded space $\mathbb{R}^m$ tend to have future trajectories that rapidly diverge because they do not correspond to real neighbors in the true manifold $\mathcal{M}$ state space. Therefore, the impact of noise is greatly amplified as small perturbations in the reconstructed space $\mathbb{R}^m$ result in large uncertainties in the true state of the system.

3. L-statistic

One weakness of the noise amplification measure is its requirement to define T, the prediction horizon over which to calculate noise amplification. This was addressed by Uzal[29] by modifying the definition of noise amplification to the following equation:

$$\sigma_\epsilon^2(\vec{x}) = \frac{1}{T_M}\int_0^{T_M} \sigma_\epsilon^2(T, \vec{x})\, dT. \tag{30}$$

This definition calculates the noise amplification with respect to a range of prediction horizons up to a maximum value of T_M and is found to be relatively robust for sufficiently large T_M.

The algorithm used to calculate σ relies on using k nearest neighbors from a reference point $\vec{x}_i$ as a proxy. Based on the distribution of points, this can result in effective noise levels ϵ of different sizes for each point. Therefore, Uzal proposed a normalization constant α_k accommodate for this variation given by

$$\alpha_k^2 = \left[\sum_i \epsilon_k^{-2}(\vec{x}_i)\right]^{-1}. \tag{31}$$

Combining these two ideas, the authors propose that noise amplification σ measures some notion of redundancy, and α_k measures some notion of irrelevance. The L-statistic is then described as a cost function to minimize both of these values simultaneously,

$$L = \log(\sigma\alpha_k) = \log(\sigma) + \log(\alpha_k). \tag{32}$$

VII. NON-UNIFORM EMBEDDING

The popularity of uniform delay embedding can be attributed to its ease of implementation and optimization. In a direct application, uniform delays only require the selection of two parameters, τ and m. However, the convenience of such an approach comes at the cost of reduced versatility and limitations, particularly when analyzing systems with dynamics occurring on multiple disparate time scales.[60,61]

First, the choice to use a single delay limits the ability for the reconstruction to highlight features across multiple disparate timescales.[27] For example, a fast-slow system with characteristic time scales τ_1 and τ_2 where $\tau_1/\tau_2 \gg 1$, the choice of selecting τ_1 (i.e., slow dynamics) as the embedding lag can limit the reconstruction's ability to fully unfold attractor topologies corresponding to the fast dynamics. The dynamics of the time scale of τ_2 (i.e., fast dynamics) will appear as noisy fluctuations within the reconstructed state space.

Second, reconstruction from uniform delay embedding that is sufficient is not necessarily optimal. Here, we must clarify that the definition of optimal presumes some criterion or notion of quality. Casdagli noted that the quality of an embedding, defined as the reconstruction's robustness to noisy data for prediction, can vary locally throughout different regions of the attractor.[28] This behavior was also highlighted by Uzal in his extension of Casdagli's noise amplification and distortion methods.[29] Additionally, we should also consider that invariant measures such as the Lyapunov exponent also vary locally.[62,63] Hence, the selection of a single embedding lag implies that all these variations may be averaged.

Non-uniform delay embedding has been proposed as an natural extension of uniform embedding that aims to address some of the latter's limitations. Non-uniform delay embedding requires the selection of multiple delay lags $\{\tau_1, \tau_2, \dots \tau_{m-1}\}$ in order to construct a delay vector,

$$\vec{x}(t) = (x(t), x(t-\tau_1), \dots, x(t-\tau_{m-1})). \tag{33}$$

The selection of delay lags represents a combinatorially hard problem that grows with increasing embedding dimension. The methods proposed for constructing non-uniform delay embedding often involve the iterative selection of time lags to gradually construct a delay vector until the required embedding dimension is reached. In this section, we give an overview of various methods that have been proposed to solve and automate this problem. These methods include the continuity statistic,[27] PECUZAL,[64] maximizing derivatives on projections (MDOP),[40] reduced autoregressive models,[60,61] and search optimization algorithms such as ant colony optimization[65] and Monte Carlo decision tree search (MTCDS).[66]

A. Garcia and Almeida

One of the earliest proposed methods of choosing non-uniform delays was proposed by Garcia and Alemeida.[47] They proposed a variation in the nearest neighbor methods of Kennel and Hegger applied to the problem of selecting time delays. Their method also recursively selected lags using a proposed N-statistic over multiple embedding cycles. At the end of each cycle, the false nearest neighbors algorithm is used to assess the quality of newly constructed embedding. This process is repeated until the false nearest neighbor statistic F decreases below a critical threshold.

For the selection of the first time lag τ_1, a 2D delay embedding $\vec{x}(t)$ is first done with respect to some prospective time lag τ^* to be tested,

$$\vec{x}(t) = (x(t), x(t-\tau^*)). \tag{34}$$

The closest neighbor $\vec{x}(t_j)$ for each point $\vec{x}(t_i)$ in the embedding reconstruction is identified. Neighbors should be chosen such that they are not temporally close (i.e., with respect to some Theiler window).[48,49] This is to ensure that their spatial proximity is not purely due to their temporal proximity. The two Euclidean distances $d_{1,\tau^*}(t_i), d_{2,\tau^*}(t_i)$ between any given two points are then calculated as follows:

$$d_{1,\tau^*}(t_i) = ||\vec{x}(t_i) - \vec{x}(t_j)|| \tag{35}$$

$$d_{2,\tau^*}(t_i) = ||\vec{x}(t_i + \delta t) - \vec{x}(t_j + \delta t)||, \tag{36}$$

where δt is the sampling time of data. Simply put, d_{1,τ^*} is the spatial separation between pairs of nearest neighbors in the reconstructed state space and d_{2,τ^*} is the resulting separation one step forward in time. The N-statistic is taken as the proportion of points whose distance ratio $d_{2,\tau^*}/d_{2,\tau^*} > 10$,

$$N(\tau^*) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\frac{d_{2,\tau^*}(t_i)}{d_{1,\tau^*}(t_i)} > 10\right), \tag{37}$$

where N is the length of the time series and $\mathbb{1}$ is the indicator function. The threshold of 10 was heuristically selected by the authors based on the numerical calculations of Kennel *et al.*[46] The time lag corresponding to the first minimum in $N(\tau^*)$ is taken to be the embedding lag.

For non-uniform delay embedding, the selection of additional lags for each subsequent embedding cycle is done using a similar procedure. However, the reconstructed space used to calculate nearest neighbors and pairwise distances are calculated conditionally on previously selected lags. Therefore, the selection of the mth embedding lag in a non-uniform embedding procedure will require neighbors and distances $d_{2,\tau^*}, d_{2,\tau^*}$ to be calculated using embedding with $m-1$ lags that have already been chosen and the new candidate lag τ^*,

$$\vec{x}(t) = (x(t), x(t-\tau_1), \dots, x(t-\tau_{m-1}), x(t-\tau^*)). \tag{38}$$

B. Continuity statistic

The continuity statistic was first proposed by Pecora *et al.* as a way to procedurally construct non-uniform delay vectors based on the idea of functional independence between vector coordinates.[27] Takens' and Sauer both discussed the requirement that an embedding reconstruction requires vectors whose coordinates are independent.[3,67] Pecora *et al.* proposed using a test for calculating the functional dependence between the components of a delay vector's components in order to assess the quality of embedding. A functional dependence between vector coordinates implies

$$x(t-\tau_m) = F(x(t), x(t-\tau_1), \dots, x(t-\tau_{m-1})), \tag{39}$$

where F is some arbitrary function. Constructing non-uniform delay embedding requires iterative building of a collection of time lags

25 October 2024 06:12:41

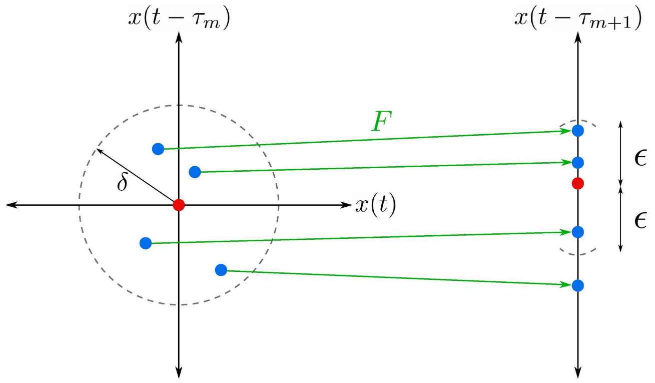


FIG. 2. Illustration of the proposed functional dependence F that is tested with the continuity statistic. The reference point (red) has four neighbors (blue) within a ball of radius δ. Under mapping F, only three of these neighbors lie within a distance ϵ from the image of the reference point. The value p is the fraction of neighbors in embedded space (left) that are also neighbors in the potential new lagged component (right). The value of ϵ is adjusted until p is insufficient to reject the null hypothesis that the probability of neighbors in embedded space (left) are also neighbors in the new lagged component (right) is binomially distributed with $p^* = 0.5$.

$\vec{\tau} = \{\tau_1, \dots \tau_{m-1}\}$ that minimizes the likelihood of functional dependence between components. In each iteration, a prospective lag τ_i is tested for functional dependence with the existing lagged components corresponding to $\vec{\tau}$. If there is no significant functional dependence, then τ_i may be added to the collection of lags. To test the equality of Eq. (39), the authors assume that F is smooth and use the property of continuity to quantify functional dependence.

Consider an existing m-dimensional embedding $\vec{x}_m(t) \in \mathbb{R}^m$ constructed from lag $\tau = \{\tau_1, \dots \tau_m\}$ and a potential new embedding lag to be tested τ_{m+1}. To test the functional dependence of a new lag, select a reference reference point $\vec{x}_m(t_0)$ in embedded space. If a smooth functional dependence exists, then the continuity condition states that points $\vec{x}_{m,i}$ nearby the reference point ($||\vec{x}_{m,i} - \vec{x}_m(t_0)|| < \delta$) in reconstructed space $\mathbb{R}^m$ should have lagged $m + 1$th components that are also close by to each other ($|x_i(t - \tau_{m+1}) - x(t_0 - \tau_{m+1})| < \epsilon$) (see Fig. 2).

The proportion p of points $\vec{x}_{m,i}$ whose lagged components lie within ϵ of the reference point's lag component can be calculated. This proportion is then compared against a null hypothesis; that correspondence between these sets is purely by chance. Large values of p suggest a strong relationship between the m-dimensional reconstruction and the new τ_{m+1} lagged component. Pecora *et al.* suggest the usage of a binomial distribution with a critical value of $p^* = 0.5$ in order to decide if a functional dependence exists with respect to some chosen ϵ due to its simplicity and robustness to noise.[27]

For a given τ_{m+1} to be tested and a sample of points, the continuity test is applied with decreasing values of ϵ until the null hypothesis fails to be rejected. The smallest possible value for rejecting the null hypothesis is given as ϵ^*. This value is averaged over a collection of reference points sampled from the data to calculate the continuity statistic $\langle \epsilon^* \rangle(\tau_{m+1})$.

During each iteration of choosing a candidate lag τ_{m+1} for an existing collection of lags $\vec{\tau} = \{\tau_1, \dots, \tau_m\}$, the continuity statistic profile $\langle \epsilon^* \rangle(\tau_{m+1})$ is calculated. The new lag τ_{m+1} is taken as the lag corresponding to the relative maxima of the continuity statistic profile. This is repeated until the desired embedding dimension (as per Whitney's theorem) is reached. Pecora *et al.* also propose an undersampling statistic that can be used as a termination criterion for iterative selection of time delays. Further details can be found in the original paper.

This method was applied to a 2-torus, yielding embedding lags and embedding dimensions that were matching with theoretical expectations.[27] The resulting reconstructed attractor was also found to be visually optimal. Similar results were gained when applied to the Lorenz chaotic time series. However, the resulting reconstruction appeared to be visually overfolded.

C. PECUZAL

A criticism of the continuity statistic method is the ambiguity in selecting the optimal lag τ at each embedding iteration.[64] In the original paper of Pecora *et al.*, the definition of "relative maxima" is unclear, and there is no objective criterion for selecting the best lag between multiple prospective local maxima.[64] Additionally, the method also does not consider the effects of selecting different distances δ used to define nearby points in the reconstruction. Finally, the undersampling statistic originally proposed as a breaking condition for the embedding algorithm is computationally intensive and does not inform on which of the prospective lags should be selected. A more detailed critique is provided by Kraemer *et al.*[64]

Kraemer *et al.* suggested that the continuity statistics approach could be combined with Uzal's L-statistic[29] in order to provide a fully automated method of constructing non-uniform embedding delays. In their paper, they provide a workflow that uses the continuity statistic to perform a coarse search of multiple lag times and identify a small set of potential lags. These usually correspond to the various local maxima of the continuity statistic profile $\langle \epsilon^* \rangle(\tau_{m+1})$.

The L-statistic is then used as an assessment criteria to select which of the prospective lags should be selected in each embedding cycle. This addresses the problem of ambiguity of selecting lags that is present when using continuity statistics. The prospective lag whose new extended delay embedding resulted in the largest decrease in the L-statistic is selected in each embedding cycle. The L-statistic also provides a breaking condition for the embedding algorithm. The embedding cycles end when there is no achievable decrease in the L-statistic from the collection of prospective lags, i.e., $\Delta L > 0$ between successive embedding cycles.

D. Maximum derivatives on projection

The maximum derivatives on projection (MDOP) method was first proposed by Nichkawde as a geometrical alternative to the statistics and information theoretic approaches of mutual information and continuity statistics.[40] MDOP optimizes an embedding based on the criteria that a reconstructed attractor should be maximally unfolded and minimize redundancy in the delay components. Similar to the majority of non-uniform embedding methods, MDOP recursively constructs the delay vector through embedding cycles.

Each cycle identifies a new time lag that maximizes the directional derivative $\phi'_d(\tau)$ of points in reconstructed state space.

Like Pecora's approach in continuity statistics, MDOP begins with the criterion of functional dependence between each new prospective lag and an existing time delay reconstruction [see Eq. (40)]. However, unlike in continuity statistic, Nichkawde suggests using the directional derivative of the functional dependence F [see Eq. (39)] to quantify the degree of redundancy in embedding and unfolding of the reconstructed attractor. This directional derivative is given by

$$F'_m = \left|\frac{\Delta F_m}{\Delta x_m}\right|, \tag{40}$$

where Δx_m corresponds to the spatial distance between a pair of nearby neighbor points $\vec{x}(t_i), \vec{x}(t_j)$ in reconstructed space with $m - 1$ dimensions,

$$\Delta x_{m,ij} = ||\vec{x}(t_i) - \vec{x}(t_j)|| \tag{41}$$

$$= \sqrt{\sum_{k=1}^{m-1} [x(t_i - \tau_k) - x(t_j - \tau_k)]^2} \tag{42}$$

$$\tau_0 = 0. \tag{43}$$

The sampled pair of points should also be chosen such that spatial closeness is not due to them being virtually close in time.[48,49] This is easily achieved by allowing for a Theiler window l_{Theiler} where $|i - j| > l_{\text{Theiler}}$.

Testing the inclusion of a new time lag τ_m requires evaluating the spatial variation in the prospective new lagged component, $\Delta F_{m,ij}$, and is given by the following equation:

$$\Delta F_{m,ij} = |x(t_i - \tau_m) - x(t_j - \tau_m)|. \tag{44}$$

This quantity is used to evaluate the directional derivative of a small region on the reconstructed attractor,

$$F'_{m,ij} = \left|\frac{\Delta F_{m,ij}}{\Delta x_{m,ij}}\right|. \tag{45}$$

The directional derivative is evaluated with respect to each prospective new time lag τ_m and is averaged across randomly sample close pairs of points across the entire reconstructed attractor,

$$\beta_d(\tau_m) = \left\langle F'_{m_{i,j}} \right\rangle_{i,j}, \tag{46}$$

where $\langle \cdots \rangle_{i,j}$ corresponds to the geometric mean across all sampled pairs of points. The author proposes using a geometric mean due to its robustness in the presence of outliers. In each recursive embedding cycle, the lag τ_m that maximizes the directional derivative $\beta_d(\tau_m)$ is selected to be used for the reconstruction in the next cycle. This process is repeated until the desired number of embedding dimensions m is reached, where m is chosen via a number of different embedding dimension estimation methods such as false nearest neighbors, etc.

E. Reduced autoregressive models

The reduced autoregressive model for non-uniform embedding was proposed by Judd and Mees[60,68] as a proposed method of constructing ideal models with respect to some information criterion. This method involves the construction of a pseudo-linear autoregressive predictive model with all k possible lagged components as inputs or basis functions,

$$x(t+1) = a_0 + \sum_{i=1}^{k} a_i x(t - \tau_i) + \epsilon(t), \tag{47}$$

where $\lambda = (a_0, \ldots, a_k)^T$ are the coefficients of each input to be determined and ϵ_t are the model prediction errors. For a time series of length N, construct a matrix V_τ with each row containing a vector of lags at a given time t,

$$V_\tau = \begin{pmatrix} x(\tau_k + 1) & x(\tau_k) & \ldots & x(1) \\ x(\tau_k + 2) & x(\tau_k + 1) & \ldots & x(2) \\ \vdots & \vdots & \ddots & \vdots \\ x(N) & x(N-1) & \ldots & x(N - \tau_k) \end{pmatrix}. \tag{48}$$

The matrix V_τ has dimensions $(N - k) \times k$ and is defined with respect to a set of lags $B = \tau_1, \ldots, \tau_k$ and $\tau_1 < \tau_2 < \cdots < \tau_k$.

Estimates for the coefficients of $\hat{\lambda}$ can be calculated using least squares regression,

$$\hat{\lambda} = (V_\tau^T V_\tau)^{-1} V_\tau^T \xi, \tag{49}$$

where $\xi = (x(\tau_k + 1), x(\tau_k + 2), \ldots, x(N))$. Therefore, the resulting model errors utilizing the set of all possible lagged components B can be calculated as $e_\tau = V_\tau \hat{\lambda}_\tau$.

In order to reduce the number lagged components to a smaller selection, Judd and Mees propose the method of minimum description length. The principle of minimum description length is an application of Occam's razor to the context of model selection. It defines that the best model for a given time series prediction task is one that achieves the most concise description of the data. For model selection, this would require achieving a compromise between model accuracy and model complexity (i.e., model description length). Model description length L may be approximated by

$$L = L_\epsilon + L_a = N \log(\bar{\epsilon}) + L_\lambda,$$

where L_ϵ is the description length of the model errors which is a function of the length of the time series N and the mean square prediction error $\bar{\epsilon}$ and L_λ is the description length of the model parameters.

The algorithm for reducing the number of lagged components is as follows:

1. Construct an empty set B' of chosen lags, and B of candidate lags
2. Define the prediction error with respect to the chosen set of lags B' as $e_{B'} = \xi - V_{B'}^T \lambda_{B'}$ where $V_{B'}^T$ and $\lambda_{B'}$ are defined with respect to the smaller set of lags B'. If B' is empty, then $e_{B'} = \xi$.

3. Calculate the vector $\mu = V_\tau^T e_{B'}$ and identify the index p of the largest magnitude element corresponding to the most significant lag component. Add this lag $\{B' \cup \tau_p\} \to B'$
4. Recalculate μ with respect to the new set of chosen lags B' and verify that least significant lag component was the lag τ_p that was most recently added. Otherwise, return to step 2.
5. Evaluate the model description lengths L_B and $L_{B'}$ sets of lags B and B'. If $L_{B'} < L_B$, return to step 2. Otherwise, end the algorithm and return the set of chosen lags $B' \to B$.

An implementation of the minimum description length criterion for optimal embedding lag and window was done by Small and Tse.[69,70] An extension of this method was proposed by Hirata *et al.*[61] where a normalized maximum likelihood $\mathcal{L}$ is used in place of minimum description length L for model selection.[71,72] Hirata *et al.* also propose a variation in the above algorithm by Judd and Mees—cross-validation—that utilizes the radial basis modesl instead of pseudo-linear models.

F. Search optimization algorithms

Many of the common non-uniform embedding methods involve a single optimization step in each embedding cycle. In contrast, search optimization algorithms attempt to search across state space of possible lags to identify ideal combination embedding lags without necessarily selecting the first local optima encountered. Two examples of such approaches are the ant colony optimization (ACO) method[65] and Monte Carlo decision tree search (MTCDS).[66]

Ant colony optimization (ACO) is a swarm intelligence method first proposed by Dorigo[73,74] that is inspired by the foraging behavior of ant colonies. Similar to other swarm optimization methods, ACO initializes a number of agents (“ants”) that simultaneously perform an initial search of the solution space. The quality of each attempted solution is assessed according to an objective function and a “pheremone” is assigned to the corresponding search path. These pheremones are able to accumulate and fade over time. This biases the search direction of subsequent iterations of the algorithm and is reminiscent of the optimal path finding behaviors of real world ant colonies.

The ant colony optimization method applied to non-uniform embedding (ACO-NE) builds upon this framework in a few ways. (i) By using an objective function based on various notions of embedding quality (mean neighborhood distance, minimum false nearest neighbors, and minimum description length[69,70]) to optimize parameters. (ii) Incorporating heuristics into the algorithm to speed up convergence. Interested readers are encouraged to refer to the original paper for more details.[65]

Another search optimization algorithm, MTCDS, proposed by Kraemer *et al.*[66] reframes the non-uniform lag selection problem into a decision tree search. Each embedding cycle is represented by a collection leaves or nodes stemming from a root (the original time series) where each leaf is the selection of a particular candidate embedding lag. A Monte Carlo approach is used to randomly sample the tree and identify various local optima for a given objective function and backpropagation is then used to decide on the best selection of lags in each step.

VIII. PERSISTENT STRANDS AND CHARACTERISTIC TIMES

The focus of this paper is on the problem of selecting time delays for delay embedding, and, in particular, for non-uniform delay embedding. Many of the proposed methods for optimizing embedding delay in both the uniform and the non-uniform case generally fall into the broad categories of dynamics (e.g., mutual information, continuity, etc.) or topology (fill factor, MDOP). With the exception of the PECUZAL automated embedding framework, non-uniform delay embedding strategies only focus on one of these two broad aspects in their definition of good embedding.

Another weakness in non-uniform embedding is that they do not always provide a full picture on the relative significance of each delay. When operating under the iterative construction of delay vectors, each prospective new time delay must be reevaluated with respect to the most recently updated embedding. Hence, the significance of each subsequent delay is conditional on the previous selected sequence of delays. This is a weakness particularly in fully automated algorithms such as MDOP and PECUZAL where ideal embedding lags are selected automatically, with little to no reference on their relative impact on embedding quality. Additionally, there is often a lack of consistency between the results of different methods. This can be attributed to the fact that optimization is done with respect to different notions of embedding quality. Often, these methods do not provide a dynamical explanation for each time lag’s significance and ability to improve a given embedding.

In view of this, we argue that a good non-uniform embedding method should have two main qualities. First, the embedding criteria should utilize both dynamical (irrelevance, periodicity, independence of coordinates) and topological (attractor unfolding) features in their selection of embedding lag. Second, the significance of each selected embedding lag should be explainable.

As our contribution, we propose a new method, “significant times on persistent strands” or SToPS, of identifying non-uniform embedding time lags using techniques drawn from persistent homology and recurrence analysis. We introduce the idea of a characteristic time scale spectrum of a signal based on the periodicities of time series and show how this may be used to identify ideal time delays. The selection of multiple time delays are also treated independently, marking a contrast to the iterative approach of constructing delay vectors that is common in most non-uniform delay embedding methods. We demonstrate the performance of our method on a collection standard periodic and chaotic time series from the literature. Additionally, we explore its performance on experiment neuron data containing fast-slow dynamics and show that SToPS is sensitive in identifying explainable time delays.

A. Introduction to persistent homology

Persistent homology has seen a recent growth in popularity particularly in the fields of dynamical systems.[75–79] We also note that recent work has also been done attempting to automate the selection of delay embedding parameters with persistent homology.[80] In its essence, persistent homology aims to quantify and track the evolution of the topological properties of an object (network, point cloud data, etc.) under an increasing notion of distance.[76] The process of gradually increasing distances is referred to as "thickening." Topological features that persist for a large interval of distance under the thickening process are observed to be significant. For simplicity, we refer to the coordinates or increasing distance ϵ as analogous to increasing time. Conversely, short-lived topological changes are typically perceived to be noise.

We describe the thickening process as follows. Consider a point cloud $\mathcal{A} = \{\boldsymbol{x}_i\}$ arranged in the pattern of a circle. Place an open ball $B_\epsilon(\boldsymbol{x}_i)$ of radius ϵ centered at each point and let the union of all open balls be the set of interest,

$$\mathcal{B}(\epsilon) = \bigcup_{i=1}^{N} B_\epsilon(\boldsymbol{x}_i), \tag{50}$$

where we are interested in calculating the homology of the set $\mathcal{B}(\epsilon)$ whose complex is given by $K(\mathcal{B}(\epsilon))$. As ϵ is increased, the set of open balls also increase in size, forming a filtration shown below,

$$K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n, \tag{51}$$

where $K_n = K(\mathcal{B}(\epsilon_n))$ and $\epsilon_0 \le \epsilon_1 \le \cdots \le \epsilon_n$. More precisely, each set K_n represents a collection of simplicial complexes, with subsequent ϵ_n yielding a nested sequence called filtration. Computation of the thickening process is well documented with two main algorithms employed, Vietoris-Rips[81] and Čech,[82] of which we will employ the former.

For this case, we are concerned with changes in the homology of the set as ϵ increases. Simply, persistent homology aims to enumerate and track the number of n-dimensional "holes" in the set.[83] Namely, H_0, H_1, and H_2 for cases of low-dimensional homology. The zero-, one-, and two-dimensional holes correspond to disjoint components, cycles, and voids, respectively. Simplices (triangles, tetrahedrons, etc.) are considered solid components and are, therefore, not holes.

In the context of low-dimensional strange attractors, persistent homology tracks the persistence or lifetime (death time minus birth time) homological features (usually H_1 holes) of the attractor using spatial data from an embedding or otherwise. All the information pertaining to the birth and persistence of features can be represented easily in a persistence diagram. In a persistence diagram, each homological feature (i.e., H_0—disjoint components, H_1—holes) are represented by a plotted point with coordinates (ϵ_b, ϵ_d) corresponding to its birth and death times, respectively.

The tracking of birth and death time of homological features presents two useful features. First, it allows the tracking of the locations of holes within the data. When analyzing phase space trajectories, these holes may correspond to short-term pseudo-periods or turning points in the time series (see Fig. 5). By tracking ϵ and representative cycles, one can also identify boundary points of holes. Second, the lifetimes of homological features allow an estimate of the relative size of the feature, which may or may not be related to its significance depending on the type of data. All of these features may be represented in a persistent diagram (see Fig. 3) where points further from the diagonal represent more persistent homological features.

B. Characteristic time scales

One approach to select lags for non-uniform embedding would be to select values related to the natural time scale of the system's dynamics. Picking lags that are much smaller or larger would logically correspond to the cases of high redundancy and irrelevance, respectively. Consider the simplest case of a periodic signal. One can argue that an ideal embedding would require a delay that is related to the time scale of its main dynamics, i.e., its periodicity. For this, we employ the quarter period heuristic in the definition of the characteristic time τ,

$$\tau = \frac{T}{4}.$$

A natural progression of this concept into more complex dynamics would be to take the lag from the collection of natural frequencies,

$$\tau_i = \frac{2\pi}{4\omega_i}.$$

Because τ_i is evaluated at individual frequencies ω_i, which may vary greatly in magnitude, it is possible to capture the dynamics of systems with multiple time scales, such as fast-slow dynamical systems. Relating embedding lag with natural frequencies and periodicity also introduces a degree of explainability to the selection of τ that also directly relates to the dynamics. While this presents a potential advantage over typical non-uniform embedding methods, it requires one to be able to accurately measure the natural frequencies (or equivalent) from any given signal.

If the analyzed signals were relatively smooth and easily decomposable into sinusoids, Fourier transforms would provide an excellent solution to this problem. However, this approach quickly fails when analyzing discontinuous-like signals such as neuron voltages where the time series is characterized by alternating phases of bursting and resting dynamics. Fourier transforms also require the time signal to be stationary on its statistics. This property is not possible for chaotic time series where the phase and period of the signal vary over time. For example, the frequency power spectrum of a chaotic signal such as Lorenz and Chua produces a shape with an exponential decay. Similar arguments may be made when analyzing experimental time series, where it is expected that drifting and oscillations in the phase and/or period may occur.

Another alternative method to detect periodicity would be to use the notion of recurrence distance employed in the recurrence analysis and unstable periodic orbit detection,[84]

$$d(\tau) = \frac{1}{N}\sum_{i=1}^{N} \left\| \begin{pmatrix} x(t_i) \\ x(t_i - \tau) \end{pmatrix} - \begin{pmatrix} x(t_i + 4\tau) \\ x(t_i + 3\tau) \end{pmatrix} \right\|, \tag{52}$$

where τ is a characteristic time if the recurrence distance $d(\tau)$ is below some threshold d^*,

$$d(\tau) < d^*. \tag{53}$$

The recurrence distance tracks the displacement from a point in state space and its future trajectory. For a periodic orbit and correctly selected τ, this will result in a local minima for $d(\tau)$ where the periodic orbit returns close to its initial position. This method of detecting characteristic times τ is unsuitable as it cannot distinguish between cases where $T = \tau/2$ and the resulting embedded trajectory clusters along the diagonal.

We propose a method of identifying and weighting the significance of characteristic times from a time series. The identification

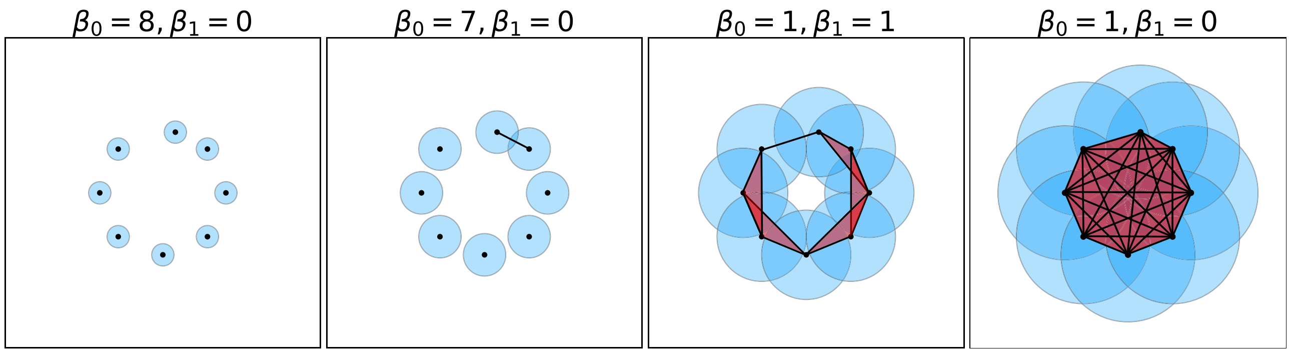

$\beta_0 = 8, \beta_1 = 0$
$\beta_0 = 7, \beta_1 = 0$
$\beta_0 = 1, \beta_1 = 1$
$\beta_0 = 1, \beta_1 = 0$

FIG. 3. Illustration of the filtration process with Betti numbers β_0, β_1 illustrating the number of H_0 (disjoint components) and H_1 (holes) in homology, respectively. Filtration is with increasing open ball radius ϵ. Initially for small ϵ, the constructed balls form eight disjoint components with no holes ($\beta_0 = 8, \beta_1 = 0$). Further increases in ϵ eventually cause the intersection of two balls, resulting in merging of two components. Subsequently, ϵ is increased until all components are merged into a single component with one unfilled center ($\beta_0 = 1, \beta_1 = 1$). Eventually, ϵ is increased until the hole in the center is filled in (shaded in red), resulting in a single solid component with no holes ($\beta_0 = 1, \beta_1 = 0$).

of characteristic times is done by sampling "strands" (short contiguous windows of 2D delay embedded time series) and calculating their persistent homology. The representative cycles of the persistent strands' calculated homologies are used to assign a significance score to each identified characteristic time in order to construct a characteristic time spectrum. This spectrum can then be used to inform the selection of lags for non-uniform time delay embedding. This combined framework is named Significance Times on Persistent Strands (SToPS).

IX. SIGNIFICANCE TIMES ON PERSISTENT STRANDS

A. Persistent strands

The first challenge to tackle is the identification of all pseudo-periodic behavior of period T. For a characteristic time τ, the quarter period heuristic suggests that the corresponding 2D embedding with coordinates $(x(t), x(t-\tau))$ will result in a periodic orbit that is approximately maximally convex in reconstructed space. We note that the precise shape of this orbit is not guaranteed to be circular and varies depending on the signal.

To test for periodic behavior with characteristic time τ for a signal $x(t)$, 2D delay embedding of the time series with a single lagged component is constructed,

$$\vec{x}(t) = (x(t), x(t-\tau)). \tag{54}$$

A collection of N strands of length $l = 4\tau$ with random initial positions t_i are then sampled from the time series, where each strand is given by

$$\vec{x}_i(t) = (x(t), x(t-\tau)), \quad t \in [t_i, t_i + 4\tau]. \tag{55}$$

We argue that strands of this length should be approximately sufficient to detect loop structures in 2D embedding based on the quarter period heuristic. The persistent homology of each strand can be calculated to detect the presence of orbits with period l. For each strand $\vec{x}_i(t)$, extract the maximum persistence from the resulting persistence diagram PD. A sample strand is said to contain an orbit of length l if the maximum persistence of the corresponding diagram exceeds a critical value ρ. The value of ρ is taken to be the average distance between two consecutive observations in phase space. Therefore, a naïve value P_i that quantifies the significance of a characteristic time τ can be defined as

$$P_i = \begin{cases} \text{maxpers}(PD_i), & \text{maxpers}(PD_i) > \rho \\ 0 & \text{maxpers}(PD_i) < \rho \end{cases}. \tag{56}$$

This value can be used to calculate an overall maximum persistence spectrum $P(\tau)$ by averaging over all non-zero scores P_i for each characteristic time τ,

$$P(\tau) = \langle P_i \rangle_{P_i \neq 0}. \tag{57}$$

We also impose an additional constraint that the number of points used to reconstruct the boundary of the hole (H_1 homology) in the orbit should be at least N_{hole}. This avoids the problem of including spurious holes where a small number of points suggest the existence of a hole even though the embedded strand is insufficiently long to close the orbit (Fig. 4). We select a minimum value of $N_{hole} = 8$ for our analyses based on the argument that a minimum of 8 points should be sufficient in at least identifying a hole of a small lag $\tau = 2$ without discounting higher lags. This value was found to work well in our analyses.

The maximum persistence spectrum provides a simple way to quantify the degree to which τ corresponds to a dynamically significant time scale (i.e., periods). However, $P(\tau)$ is biased toward larger features due to its reliance on the lifetime of each homological feature. As a result, spatially small but dynamically significant features over small time scales are underrepresented. We also note that $P(\tau)$ is unable to differentiate between pathologically inefficient embeddings such as those whose loops are not maximally circular or have trailing tails (see Fig. 5).

As we can see from Sec. III A, a good embedding is one that has been maximally unfolded to best utilize the reconstructed state space, while being robust to the effects of noise amplification. Therefore, the significance of each characteristic time τ should be

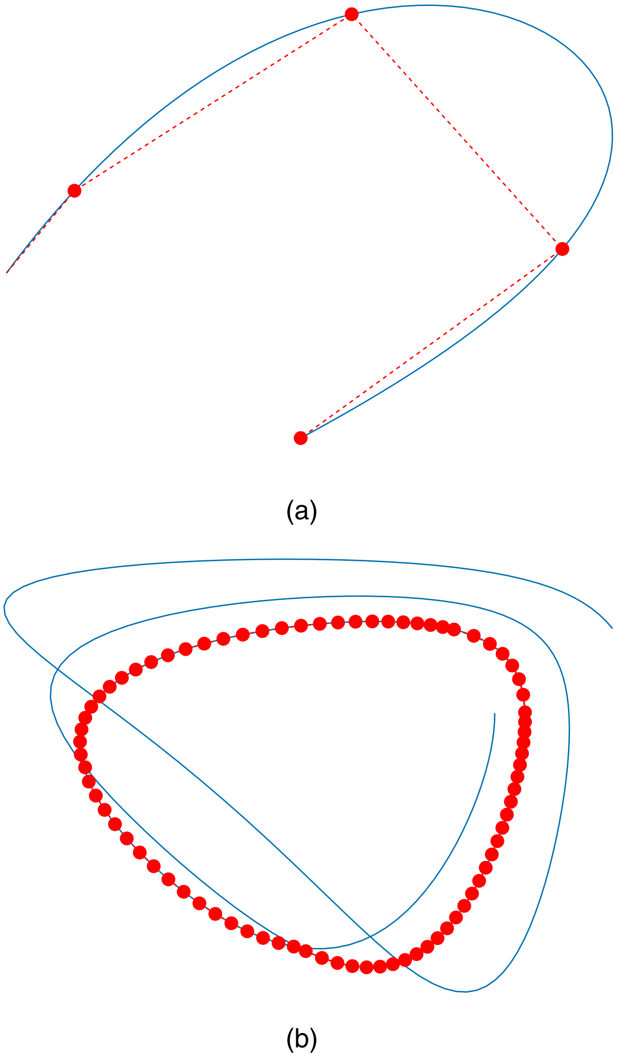


FIG. 4. Pathological problems of the SToPS method. (a) Lags are too small and resulting strand length is too short to fully enclose resulting in the detection of spurious holes with very small number of boundary points. (b) Presence of a hole even at large lags due to overlapping trajectories on a similar orbit. Long strand does not efficiently use all the available points.

weighted according to how well the corresponding strand "unfolds" into a loop structure in the 2D delay embedding with lag τ.

From this, we propose the significance score $S(\tau)$, which is a measure of the dynamical significance of each characteristic time τ that accounts for the quality of unfolding of sampled strands. Using the topological notion of good embedding, the significance score $S_i(\tau)$ for the ith sampled strand is defined as

$$S_i(\tau) = \alpha_i(\tau)\gamma_i(\tau), \tag{58}$$

where $\alpha(\tau)$ and $\gamma(\tau)$ are two separate measures named the circularity and efficiency, respectively. This score is also not biased toward larger homological features, allowing for the detection of both small and large pseudo-periodic dynamics.

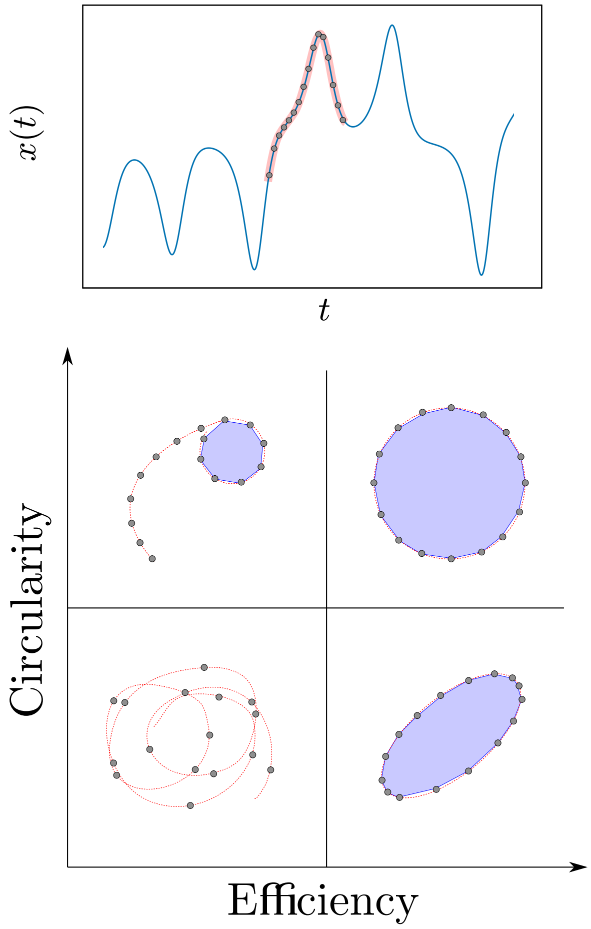


FIG. 5. Illustration of a randomly sampled "strand" (shaded red) taken from a time series and the subsequent pathological cases of embeddings from different time scales τ. Each case corresponds to non-optimal embeddings based on the notions of circularity and efficiency. Significant characteristic times should aim to maximize both metrics (top right quadrant). Blue shaded areas correspond to the polygon formed by homology generators of the hole calculated from the embedded strand's representative homology.

The circularity $\alpha(\tau)$ tries to quantify the quality of unfolding of a persistent strand in embedded space. Embeddings that yield circular loops imply a selected lag τ that maximally unfolds the dynamics (i.e., τ may correspond to a characteristic time). Therefore, loops that are more circular or regular have higher values compared to ellipses and other shapes with eccentricities. This allows circularity to also function as a measure of redundancy with lower α corresponding to high redundancy.

To calculate α_i for a given sampled strand, the boundary points $\vec{b}_i$ corresponding to the birth of the most persistent homological feature are identified by examining representative cycles. Principal component analysis (PCA) is used to identify the major and minor axes of the embedded hole. Because each strand is used to evaluate a single lag in 2D delay embedding, the first and second principal eigenvalues approximately correspond to the relative sizes of the major and minor axes of the embedded points' bounding ellipse. Hence, we define circularity as the ratio between the first and second principal eigenvalues, averaged across all strands,

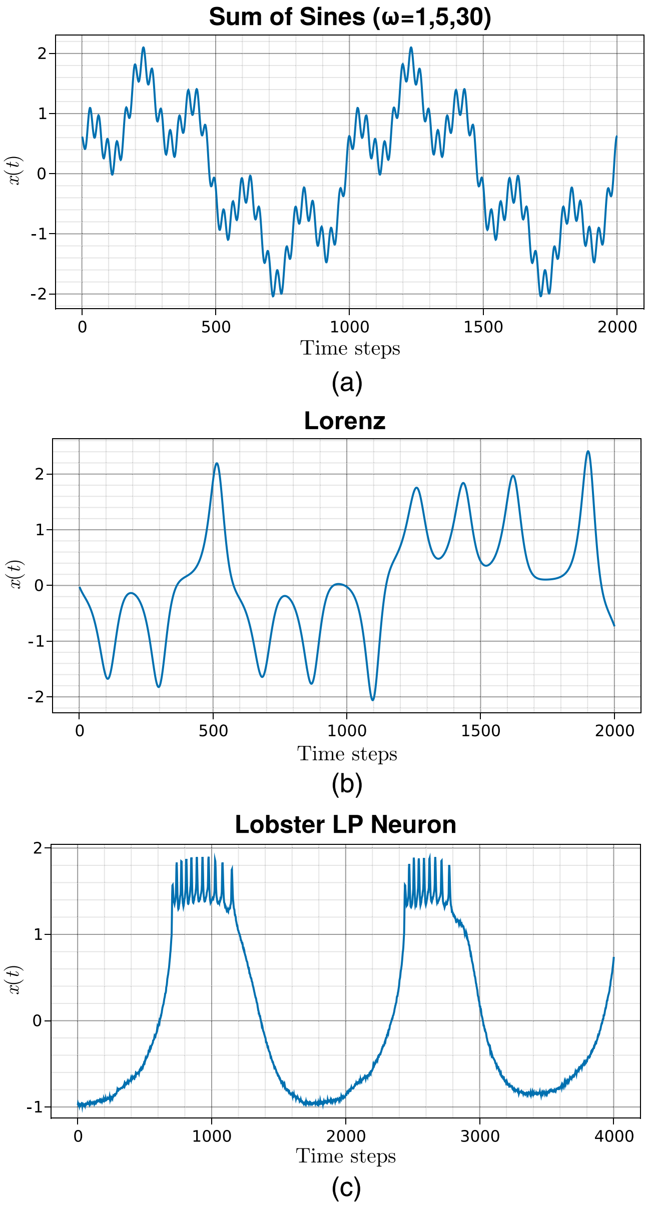
$$\alpha(\tau) = \langle \alpha_i \rangle_{P_i \neq 0} = \left\langle \frac{\lambda_{2,i}}{\lambda_{1,i}} \right\rangle_{P_i \neq 0}, \tag{59}$$

where $\alpha(\tau) \in (0, 1]$. As $\alpha \to 1$, embedded holes are more uniformly circular.

Efficiency $\gamma(\tau)$ is defined by the ratio of two areas,

$$\gamma(\tau) = \frac{A_h}{A_{pc}}, \tag{60}$$

where A_h is the area of the hole given by the ordered set of boundary points $\vec{b}_i$ and A_{pc} is the area of the smallest convex polygon that includes all points in the strand $\vec{x}_i(t)$. The area of the hole A_h can be simply calculated using the shoelace algorithm.[85] The area of the latter A_{pc} can be similarly calculated by using a Graham scan algorithm to first identify the smallest bounding convex polygon.[86] Similar to $\alpha(\tau)$, the efficiency score is also bounded with $\gamma(\tau) \in [0, 1]$. Efficiency is a measure of how well utilized the ambient space of an embedded strand is with respect to the hole. It is used to detect cases where the detected hole is circular but does not utilize the full length of the strand (see Fig. 5).



(a)

(b)

(c)

FIG. 6. Three classes of time series tested: sum of sines (periodic), Lorenz (chaotic), and Lobster LP Neuron (multiple disparate time scales). (a) Artificial time series constructed from sum of three sine terms with $\omega = \{1, 5, 30\}$, $\phi\{0, 0.25, 0.75\}$, and $A_i = \{1, 0.5, 0.2\}$. (b) First component $[x(t)]$ of the chaotic Lorenz time series. (c) Lobster LP neuron time series showing pseudo-periodic dynamics over two scales. Fast (spiking neurons) and slow (bursting neurons).

X. TESTING METHODOLOGY

SToPS was tested on three different types of time series covering periodic, chaotic, and fast-slow dynamics.

First, a sum of sines signal (see Fig. 6) with step size $dt = 0.001$ was used to simulate the case of periodic time series,

$$x(t) = \sum_{i=1}^{N} A_i \sin(2\pi\omega_i t + \pi\phi_i), \tag{61}$$

where $N = 3$ is the number of superimposed of sine waves. Phases $\phi = \{0, 0.25, 0.75\}$ and magnitudes $A = \{1, 0.5, 0.2\}$ were selected to ensure that the spacial scales of the dynamics were distinct (see Fig. 6).

The second time series analyzed was the x-component of the canonical Lorenz chaotic time series. This is used to represent the case of chaotic time series. This time series was numerically integrated with a fourth-order Runge–Kutta scheme with an integration time step of 0.0004 and subsampled to an effective time step of $dt = 0.004$ for 25 000 steps.

The third time series consisted of experimental data measured from a lobster somatogastric ganglion (STG) lateral pyloric (LP) neuron. This time series represents a third class of dynamics corresponding to fast-slow dynamics with two different spatial and temporal scales. This time series was originally analyzed by Abarbanel[87] and includes two characteristic dynamics. These are small magnitude and time scale oscillations corresponding to neuron spiking dynamics and a long time scale periodic behavior for neuron bursting (see Fig. 6). Additionally, the phase of the bursting dynamics also varies slightly over time. This results in a gradual shift of spatial position of the expected loop in a 2D embedding. As a result, strands with lengths T that are too long are penalised as these loops eventually get filled in.

All three input time series were normalized with zero mean and unit standard deviation before applying the embedding algorithms.

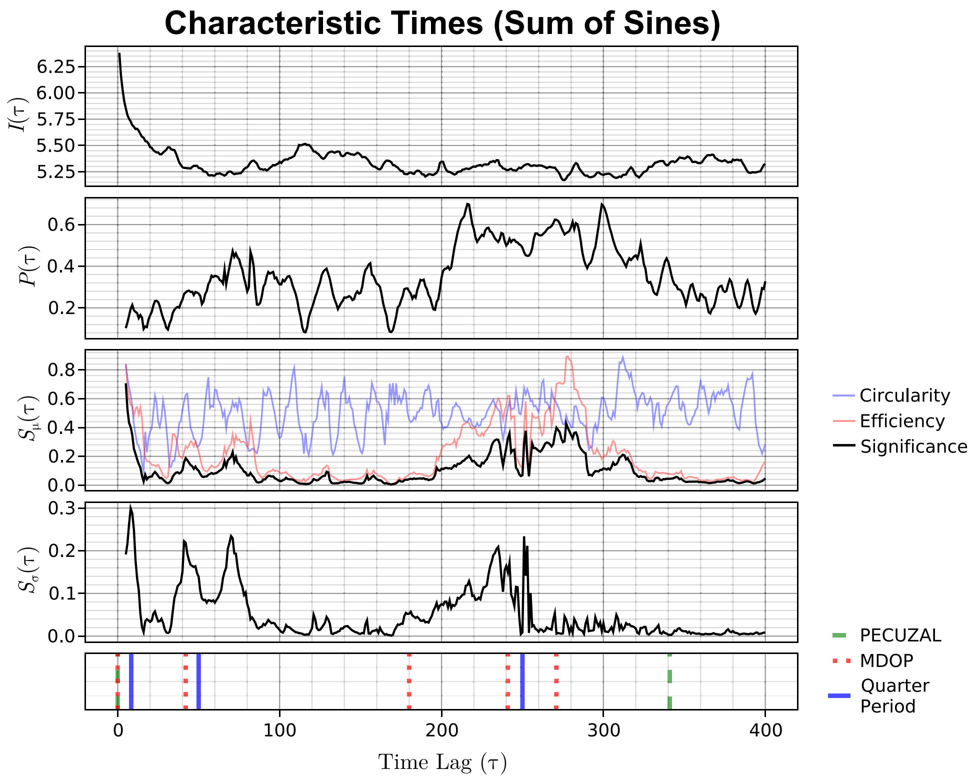


FIG. 7. Comparison of various methods for estimating embedding delay for the sum of sines time series with component frequencies $\omega = (1, 5, 30)$. Top to bottom: mutual information, max persistence, significance score (black) with circularity (blue) and efficiency (red), and standard deviation of significance score. Theoretical characteristic times based on one quarter of the period of component frequencies given by solid blue vertical lines. Comparison non-uniform delays calculated from PECUZAL and MDOP given in green and red vertical lines.

A small amount of additive noise $\xi \sim N(0, 0.001^2)$ was applied to the sum of sines and the digitized experimental Lobster LP neuron data to ensure there were not exact overlaps in values. In all cases, the input data were limited to 25 000 points when calculating embedding lags to ensure the computational time was sufficiently short. This is due to the use of a k-means++ random sampler,[88] whose computation increases with the number of points, to select strand locations that uniformly explore the reconstructed state space.

Different ranges of characteristic times were tested to calculate the efficiency, circularity, and significance score profile depending on the type of time series. Each τ consists of sampling 250 strands of length $T = 4\tau$. Due to the poor computational scaling of the Vietoris-Rips filtration, long strands where T is large are subsampled by a factor of k,

$$k = \left\lfloor \frac{4\tau}{N_p} \right\rfloor, \tag{62}$$

where $N_p = 250$ is the approximate scale of the maximum allowable strand length. This value was chosen in order to accommodate the slow computational efficiency of calculating the Vietoris-Rips filtration and associated persistent homology. The threshold ρ^* used to define the minimum lifespan needed to classify a homological feature as significant was chosen to be the average distance between temporally adjacent points,

$$\rho^* = \langle ||\vec{x}(t_i) - \vec{x}(t_{i+1})||\rangle_{i,k}, \tag{63}$$

where $\langle \cdots \rangle_{i,k}$ is the average across all points in the 2D delay embedded strand $\vec{x}(t)$ subsampled with a factor k. The circularity, efficiency, and significance score profiles were calculated using SToPS. Both mean $S_\mu(\tau)$ and standard deviation $S_\sigma(\tau)$ profiles are calculated and compared against a selection of embedding delay optimization measures. In our analyses, the peaks of each profile are selected by observation. However, automatic identification of peaks may be implemented by using a search algorithm to identify all local maxima in the significance score profiles.

The calculated significance scores were compared against the mutual information used to select τ for uniform delay embedding. Two automated methods, MDOP[40] and PECUZAL,[64] were also used as a comparison benchmark for non-uniform delay embedding. These calculations were done using implementations provided by the "`DynamicalSystems.jl`" package.[89]

In addition to compare the output lags from each method, it is of interest to find how different non-uniform embeddings impact

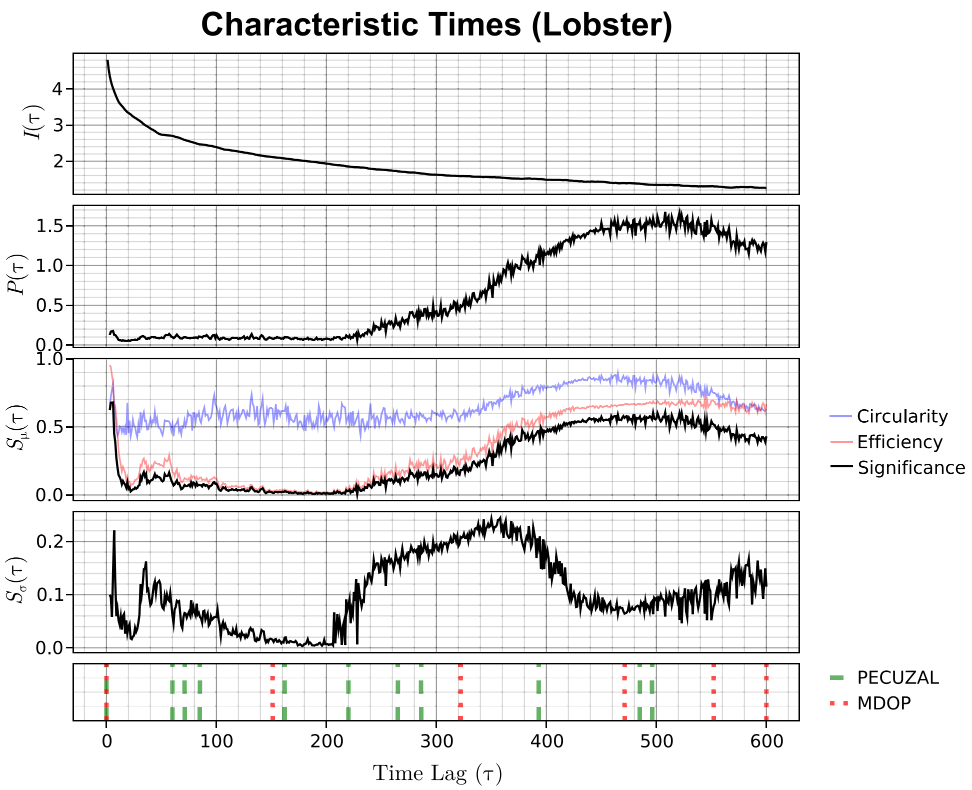


FIG. 8. Comparison of various methods for estimating embedding delay for the chaotic Lorenz time series. Top to bottom: mutual information, max persistence, significance score (black) with circularity (blue) and efficiency (red), and standard deviation of significance score. Comparison non-uniform delays calculated from PECUZAL and MDOP given in green and red vertical lines.

performance in prediction tasks. For each of the non-uniform embedding methods, the first two dominant lags are taken to construct 3D delay embedding. This is done by visually inspecting the $S_\mu(\tau), S_\sigma(\tau)$ profile in the case of SToPS. Time lags for PECUZAL and MDOP were taken as the first two detected timelags in their respective iterative procedures.

The resulting delay embeddings used to train a simple four-layer feed forward neural network consisting of two hidden layers, one input and one output layer. Each hidden layer has 128 nodes with a ReLU activation function with an overall network architecture of 3:128:128:3.

The neural network is trained to output one-step prediction,

$$\vec{x}(t+\delta t) = \mathrm{NN}(\vec{x}(t)). \tag{64}$$

The learning rate was set to 0.001 with a batch size of 512 and run for 30 epochs. In each instance, only the first half of the time series data is used for training. The second half is reserved for validation. Validation is done by calculating n-step freerun predictions. This is calculated by providing an initial condition and feeding back the neural network outputs n times to get the final prediction [see Eq. (65)], which is then used for calculating prediction error.

$$\vec{x}(t+n\delta t) = \mathrm{NN}^{(n)}(\vec{x}(t)). \tag{65}$$

XI. RESULTS AND DISCUSSION

A. Significant times

1. Periodic dynamics—Sum of sines

The periodic sum of sines time series represents the case of periodic dynamics with multiple time scales. The component frequencies were selected as $\omega = \{1, 5, 30\}$, corresponding to three different characteristic times at lags $\tau = (250, 50, 8)$. The sum of sines times series was a major challenge for the baseline mutual information $I(\tau)$ measure (see Fig. 7), where the minima were only able to identify the highest frequency periodicity in the data. The max persistence $P(\tau)$ was also not useful in identifying any significant time lags from the time series.

For automated non-uniform embedding methods, PECUZAL found a single time lag at $\tau = 341$. MDOP returned four time lags at $\tau = (241, 42, 271, 180)$, two of which are close to the expected lags $\tau = (250, 50, 8)$. The other remaining time lags detected by both methods did not bear any clear relation to the expected characteristic times.

In contrast, the proposed significance score measures, $S_\mu(\tau)$ and $S_\sigma(\tau)$, both showed peaks around the expected time lags corresponding to approximately one quarter of the component periodicities (i.e., $T = 1, 1/5, 1/30$) (see Fig. 7). However, the peak

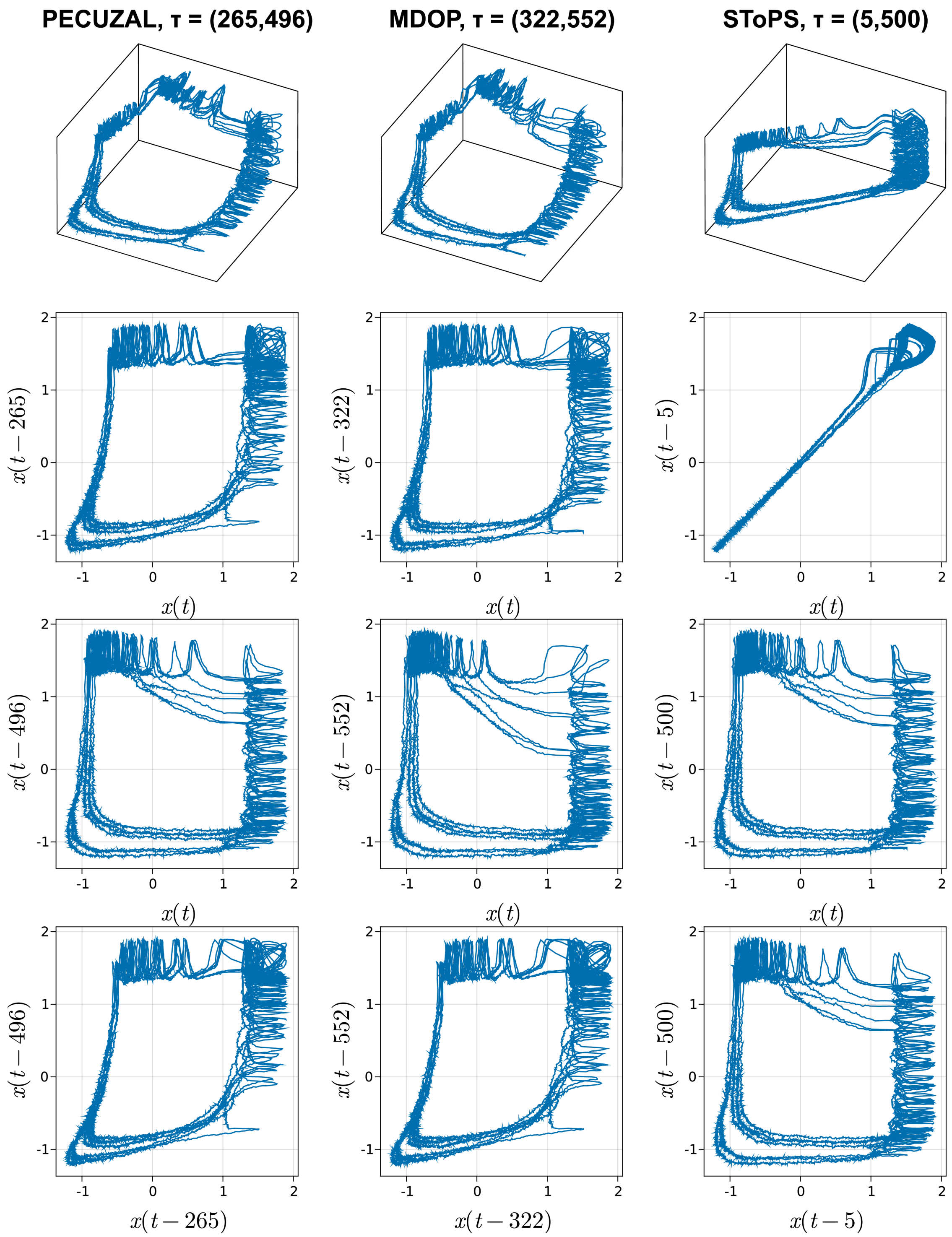


FIG. 9. Comparison of 3D time delay embedding between PECUZAL, MDOP, and SToPS with the corresponding 2D projections. PECUZAL and MDOP time delays are selected from the first two detected delays. SToPS delays are visually selected from $S_\mu(\tau)$. The fast, small-scale oscillations for spiking neurons are captured by SToPS but not by PECUZAL and MDOP.

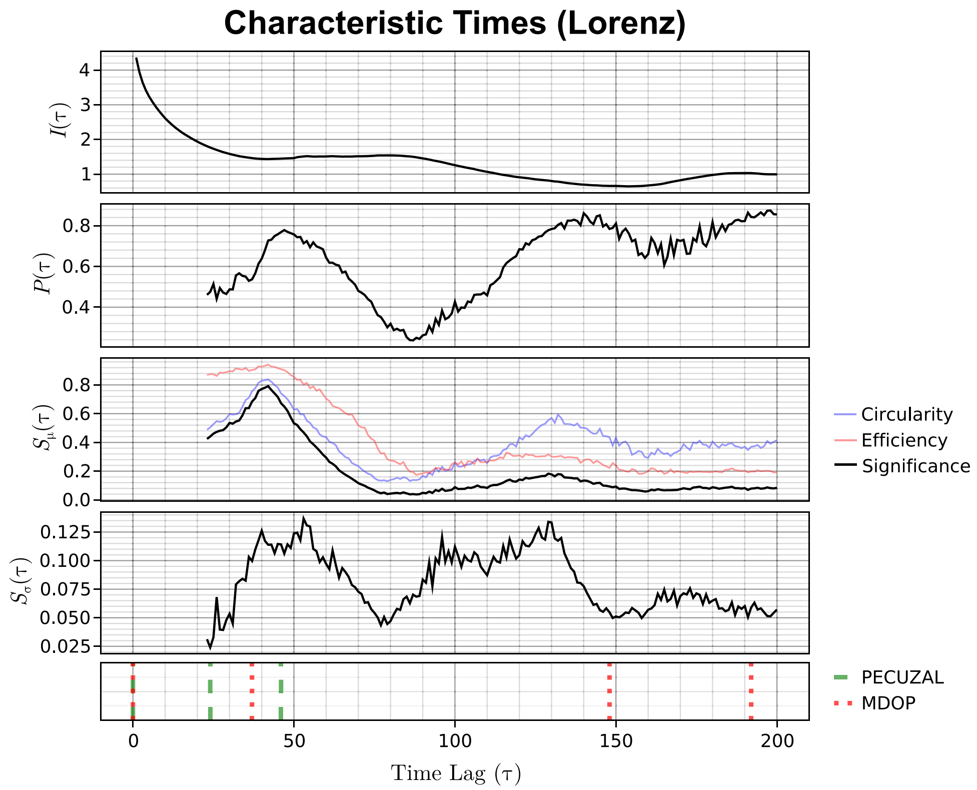


FIG. 10. Comparison of various methods for estimating embedding delay for the chaotic Lorenz time series. Top to bottom: mutual information, max persistence, significance score (black) with circularity (blue) and efficiency (red), and standard deviation of significance score. Comparison non-uniform delays calculated from PECUZAL and MDOP given in green and red vertical lines.

corresponding to the $\omega = 5$ component (i.e., $\tau = 45$) is not clear with two peaks occurring at nearby time lags instead. This anomaly may be because the spatial and time scales of the second frequency are not dynamically distinct enough from the large time scale. When calculating the 2D embedding, this can cause persistent strands to form spirals instead of circular holes at the characteristic times. We note that circularity $\alpha(\tau)$ and efficiency $\gamma(\tau)$ provide quite different profiles with the latter heavily influencing the shape of the resulting profile $S_\mu(\tau)$. A comparison of the phase space reconstructions between MDOP, PECUZAL, and SToPS is provided in the Appendix.

2. Chaotic dynamics—Lorenz

For Lorenz, the minima of mutual information yielded lags at $\tau \approx (40, 150)$. This result matches closely with the maxima taken from the maximum persistence $P(\tau)$ profile. Of the two lags detected by PECUZAL $\tau = (46, 24)$, one was similar to a minima from the mutual information curve. Similarly, MDOP yielded three different lags $\tau = (148, 192, 37)$, one of which approximately matches the minima of the mutual information.

The significance measures calculated using SToPS yielded two distinct maxima across $S_\mu(\tau), S_\sigma(\tau)$ with lags at $\tau = (41, 130)$. The first lag is similar to calculated lags from PECUZAL and MDOP. However, the second lag $\tau = 130$ results in an overembedding of the time series and does not directly correspond to any lag output by either PECUZAL or MDOP. Closer inspection of the embedded time series at $\tau = 120$ reveals the re-emergence of the lobes of the Lorenz attractor at 3/4 of the period with boundaries created by multiple dense loops. While this may produce well-defined holes in persistent homology near the lobes, it does not efficiently utilize all the points in the sampled strand (i.e., $4\tau = 520$ points) (see Fig. 4) and, hence, should not be classified as a characteristic time lag of the time series and is reflected in the much lower significance score. The usage of this time lag results in an overembedding of the time series. However, closer inspection of the sampled strands show that part of the increase in $S_\mu(\tau)$ is attributed to the hole formed from the spread of trajectories near the saddle point of the attractor.

There is also an apparent correspondence between successive minima of mutual information and detected time lags in non-uniform embedding methods. This suggests that mutual information $I(\tau)$ may be useful for informing the selection of lags for non-uniform embedding. However, one difficulty is assessing whether the lag of a minimum is within an acceptable embedding window $m\tau$ such that irrelevance is not a problem.

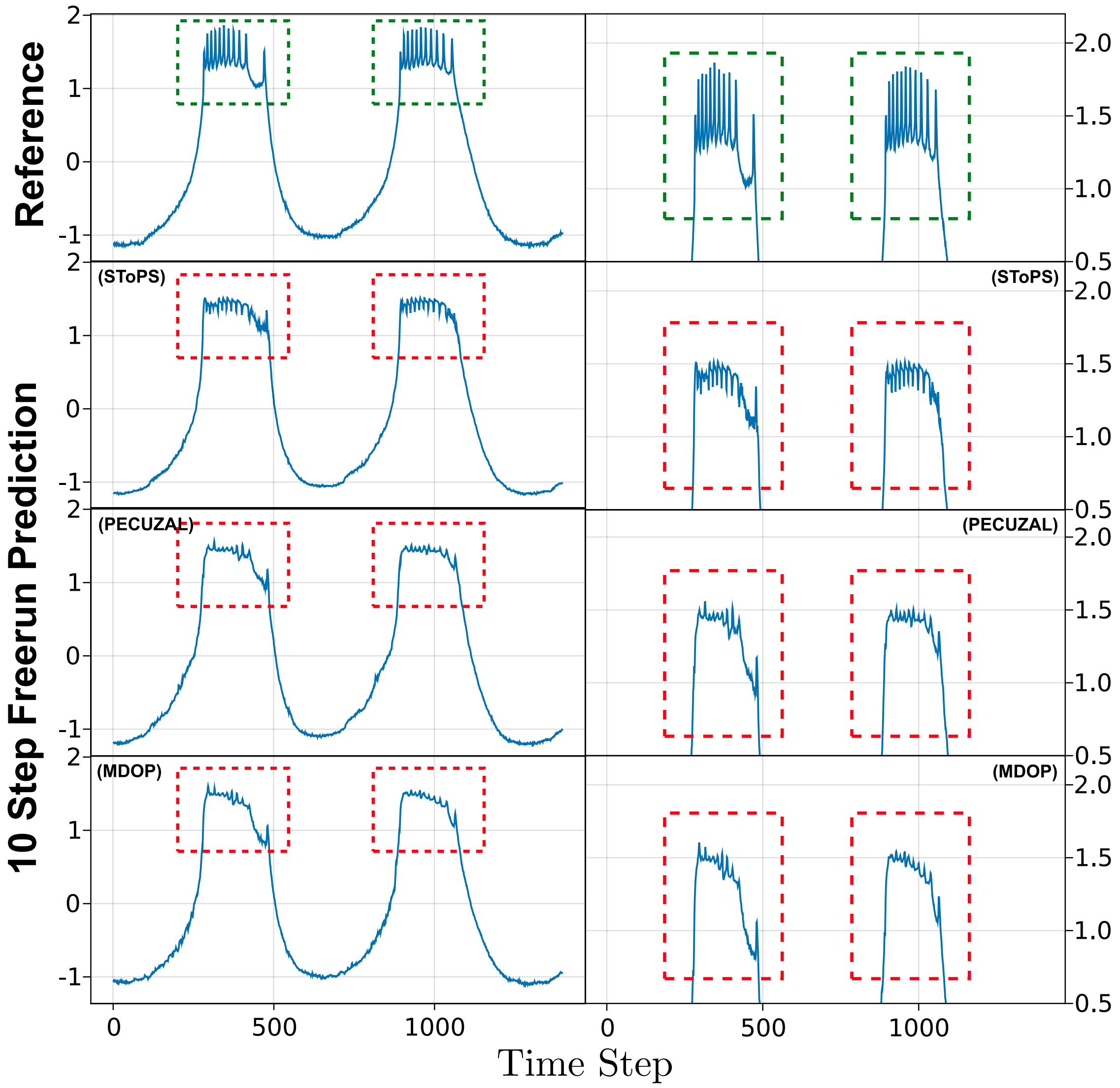


FIG. 13. Comparison of 10-step freerun predictions of the lobster LP neuron time series across various embedding methods. The persistent strands method appears to be able to better detect and replicate the fast neuron spiking dynamics. Magnitude of predictions are normalized with zero mean and unit standard deviation.

3. Fast-slow dynamics—Lobster LP neuron

From observing the lobster neuron data (see Fig. 6), it is possible to infer two dominant time scales corresponding to expected lags of approximately $\tau = (12\,400)$. Temporal variations in these can be attributed to observational noise or potentially chaotic dynamics. The uniform embedding measures of mutual information and max persistence were found to be poor in identifying lags, although max persistence $P(\tau)$ begins to quickly increase when approaching the expected characteristic lag $\tau = 400$. This is unsurprising as 2D embedding at those lags begins to unfold large orbits from the large time scale dynamics corresponding to the slow bursting phase of the neuron (see Fig. 9).

For non-uniform embedding, both PECUZAL and MDOP yielded a large number of potential embedding lags. However, both methods failed to successfully recover the lags for the fast dynamics (see Fig. 8). Additionally, apart from the lag at $\tau \approx 500$ corresponding to the slow dynamics, both PECUZAL and MDOP produce multiple additional time lags with no obvious explainable relation to the time scales of the data.

The significance score using SToPS was able to retrieve the two main time lags present in the data at approximately $\tau = (5, 500)$. There appears to be a slight disagreement on the location of the larger time lag with $S_\mu(\tau)$ and $S_\sigma(\tau)$ producing slightly different lag times. The $S_\mu(\tau)$ also shows a small peak at approximately $\tau = (40, 60)$. Further inspection into the representative homology

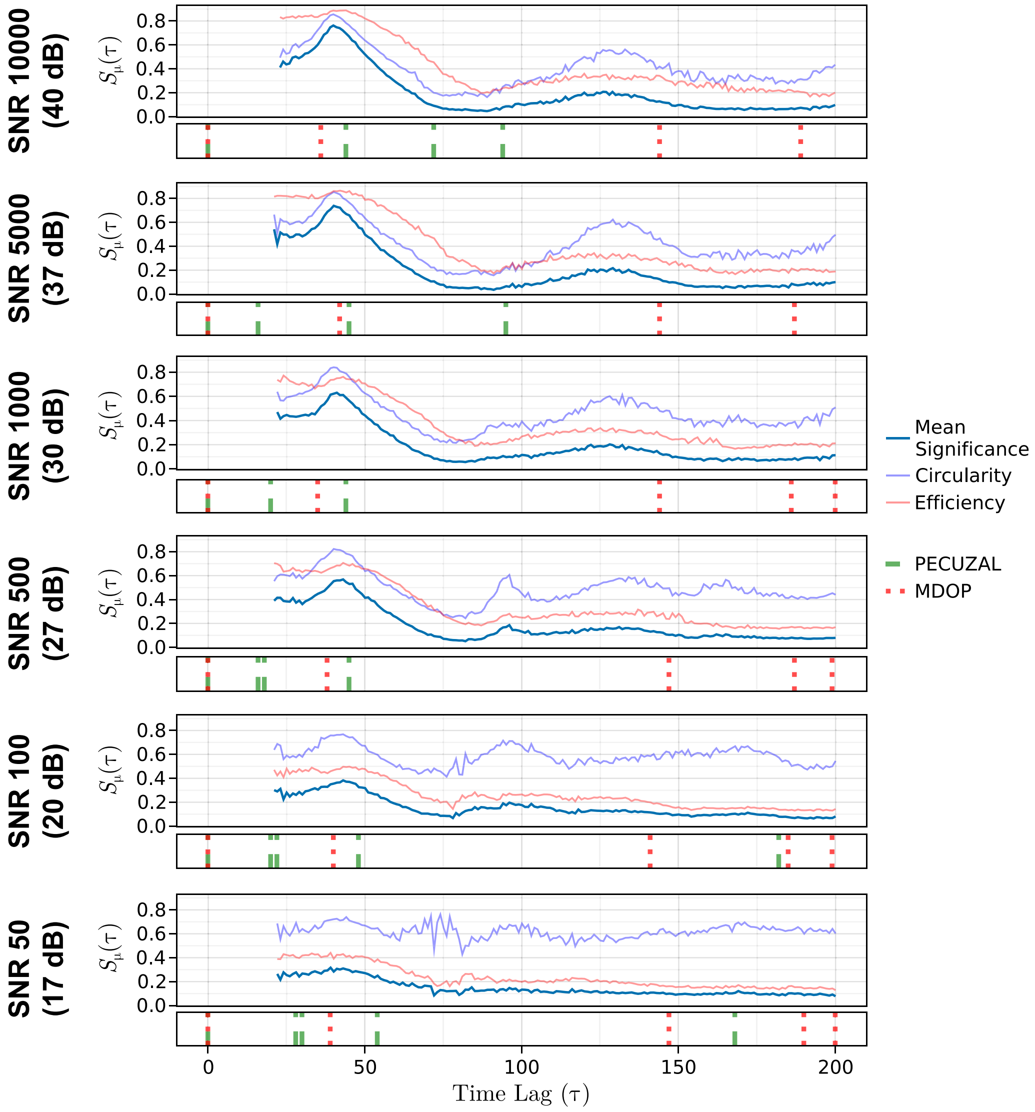


FIG. 14. Comparison of significance score $S_\mu(\tau)$ against PECUZAL and MDOP lags at various additive noise levels. SToPS significance score is relatively stable with peak locations being the same for all tested noise levels. PECUZAL and MDOP are stable for low levels of noise but gradually drift for increasing noise levels.

of the sampled strands reveal that this is the result of multiple overlapping orbits from the fast spiking dynamics. However, this feature is captured by $\tau = 5$ and the peaks at $\tau = (40, 60)$ are an artifact of overembedded time series lying on similar orbits (see Fig. 4).

Based on these results, only SToPS was able to detect both dominant time scales in the data. The lags from PECUZAL and MDOP are conditional on the selection of previously detected lags due to the iterative approach employed by the algorithm. In contrast, SToPS produces a single characteristic time spectrum $S(\tau)$ from which the significance of each potential lag can be assessed and selected independently. We visually compare the resulting 3D delay embeddings of these three methods in Fig. 9. Similar comparisons for the sum of sines and Lorenz data are provided in the Appendix. For PECUZAL and MDOP, we select the first two non-zero delays in order of detection at $\tau = (265, 496)$ and $\tau = (322, 552)$, respectively. The lags for SToPS were selected visually from $S_\mu(\tau)$ and $S_\sigma(\tau)$ with lag times $\tau = (5, 500)$. From the projections of the reconstructed state space, we find that SToPS associates different projections with dynamics of different time scales. This results in unfolding that is visually easier to interpret. In contrast, the PECUZAL requires a large number of dimensions before all dynamical components can be visually detected. Restricting the number of dimensions results in the fast dynamics being obscured at the expense of slow dynamics.

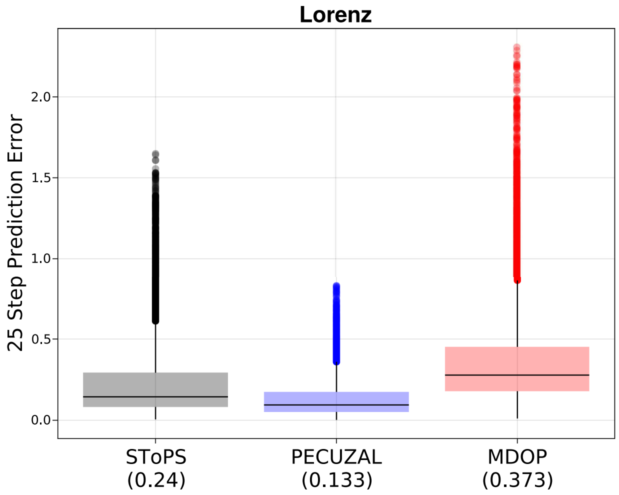


FIG. 11. Distribution of the 25-step freerun prediction error for the Lorenz time series over 25 000 steps using a neural network trained on the 3D delay embedding for one step prediction.

B. Freerun predictions

Freerun prediction errors with models constructed from different non-uniform embeddings were calculated for the non-periodic Lorenz and lobster LP neuron time series. In each case, 3D delay embedding was constructed using the first two lags detected with PECUZAL and MDOP. For SToPS, the first two visually dominant maxima of the $S_{\mu}(\tau)$ and $S_{\sigma}(\tau)$ profiles were selected. In the case for Lorenz where only one relevant time lag exists (Fig. 10) (i.e., $\tau = 41$), we use a uniform embedding scheme where the second lag is a multiple of the first.

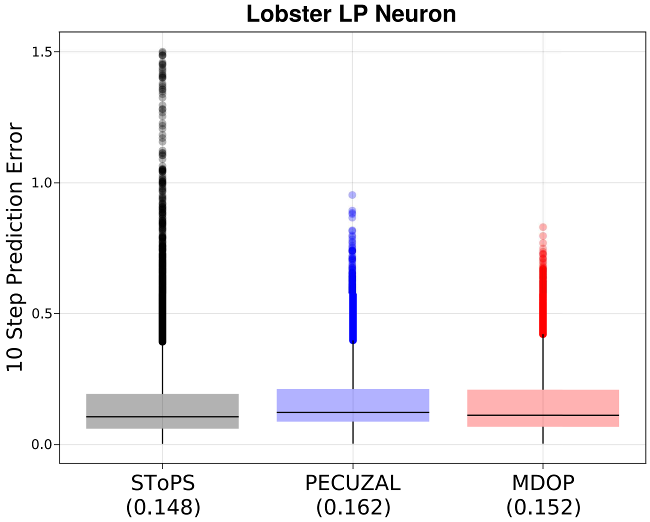


FIG. 12. Distribution of the 10-step freerun prediction error for the lobster LP neuron time series over the entire subsampled dataset (approximately 200 000 steps) using a neural network trained on the 3D delay embedding for one-step prediction.

The prediction error for Lorenz was calculated using a 25-step freerun prediction with a trained feedforward neural network (see Sec. X). For the lobster LP neuron, training was done on a subsampled dataset that included every third point. This subsampling was done to reduce the training time. A 10-step freerun prediction was then evaluated with the subsampled data (i.e., equivalent to 30-step freerun prediction). The lower number of freerun prediction steps was used to allow a more accurate evaluation of the prediction performance within the fast neuron spiking regime of the time series. In both cases, the first half of the data was used for training and the second half used for testing and evaluation. Additional analyses for freerun prediction with non-subsampled data are provided in the Appendix.

The error of each prediction was calculated as the magnitude of the error between the predicted delay vector n-steps ahead,

$$E(t) = ||\mathrm{NN}^{(n)}(\vec{x}(t)) - \vec{x}(t+n)||. \tag{66}$$

The resulting distributions of $E(t)$ for both time series with the three different embedding methods are given in Figs. 11 and 12. For Lorenz, we see that SToPS (persistent strands) provides a mean error in between PECUZAL and MDOP. In the experimental data case (lobster LP neuron), SToPS outperforms both measures with a lower error. These findings are also reflected in corresponding medians in both cases. The median prediction error in the Lorenz case was 0.106 (SToPS), 0.122 (PECUZAL), and 0.112 (MDOP). The lobster LP neuron median prediction errors were 0.106 (SToPS), 0.122 (PECUZAL), and 0.112 (MDOP). Additionally, SToPS shows an error distribution with a heavier tail for freerun predictions with the lobster LP neuron compared to PECUZAL and MDOP. Despite the potentially lower prediction error, we note that this improvement is not significant and should not be the targeted benefit of SToPS. Instead, we argue that the main advantage of SToPS is that it provides lags that are explainable in the context of the observed dynamics of the time series.

One advantage of using SToPS is the deliberate inclusion of both fast and slow time scales within the embedding of the time series. Therefore, it is expected that models trained on this embedding should be able to better resolve the fast dynamics that would have otherwise be missed if larger delays were selected. This is verified in Fig. 13 where SToPS produced a 10-step freerun prediction that is able to better replicate the small-scale, fast spiking dynamics characteristic of the neuron time series. This is in contrast to models trained on the same number of lags from PECUZAL and MDOP where freerun predictions are not able to capture the same level of detail in the spiking dynamics. We note that this advantage is not as apparent when the full dataset without subsampling is used to train a model for prediction. However, it was found that the prediction errors were slightly lower for SToPS in this case.

XII. NOISE EFFECTS

One often cited benefit of persistent homology is its robustness to noise.[90,91] Because only geometric features (i.e., holes) are tracked,

TABLE I. Computation times for PECUZAL, MDOP, and SToPS on the Lorenz and lobster LP neuron time series with 25 000 steps with varying maximum number of candidate lags (i.e., $\tau \in [1, \tau_{max}]$). Computation times are given in seconds. The computed lags for PECUZAL and MDOP are provided. SToPS does not have computed lags as results are given as a significance score profile.

Time series	τ_{max}	PECUZAL	MDOP	SToPS
Lorenz	50	6.0 (45,23)	0.19 (42,23)	148
	100	10.2 (45,23)	0.23 (100,62)	982
	150	15.8 (45,23)	0.28 (147,40)	1689
	200	21.2 (45,23)	0.42 (147,189,40)	2396
Lobster	50	7.1 (50,24)	0.39 (50,31,11,44)	95.9
	100	46.9 (100,63,81,11,35)	0.48 (100,51,19,81)	881
	300	112 (300,251,114)	0.82 (300,149,55,245)	3376
	600	2705 (515,304,132,505,32,20,253,394,81,232,55)	1.62 (544,274,599,114,423)	7266

as long as the magnitude of the noise does not destroy the underlying structure of the embedded time series, the calculated homologies should be stable. To test this property in our method, we repeat analyses with the Lorenz dataset for varying signal-to-noise ratios and observe the changes in the resulting significance score profile $S_{\mu}(\tau)$. Five noise levels of additive Gaussian noise with signal-noise ratios $SNR = (1000, 500, 100, 50)$ were applied after normalizing the input Lorenz time series with zero mean and unit standard deviation. Results are shown in Fig. 14. We found that each of the profiles was relatively stable and robust to increasing noise levels with little to no change in the observed significant lags. For high levels of noise (> 20 dB), the spurious lag artifact at $\tau = 120$ disappeared. This is likely due to the effects of noise across multiple trajectories along a similar orbit destroying the underlying homology structure.

Similarly, PECUZAL and MDOP revealed similar lags for low noise levels (< 37 dB). However, higher levels of noise resulted in gradual drifts in both methods. The lags from PECUZAL also differed significantly for small τ for noise levels above 30 dB. In contrast, MDOP was relatively unaffected by noise, and drift effects were relatively small. Additionally, MDOP was also the only method able to produce any lag predictions for signal-to-noise ratios below 17 dB. Both SToPS and PECUZAL were unable to produce any results.

XIII. COMPUTATION COMPLEXITY

The computation of persistent homology is not parsimonious and still suffers from poor computational scaling. While computation of persistent homologies for datasets with few points is relatively quick, the computation time grows rapidly for even moderately sized datasets. This presents a challenge of the SToPS method as it requires the computation of persistent homology of multiple sampled strands across a large collection of time series. From the proposed algorithm, the time complexity is approximately

$$\mathcal{O}_{SToPS} \approx \mathcal{O}(N_{\tau} N_s \phi_{PH}(N_p)),$$

where N_{τ} and N_s and N_p are the number of lags tested and the number of sampled strands and strand length, respectively, and $\phi_{PH}(N_p)$ is the time complexity for persistent homology computation. This value varies depending on the implementation of the persistent homology algorithm and the type of filtration used (e.g., Čech, Rips, Delaunay, etc.). The Vietoris-Rips filtration has a simplicial complex size K that scales exponentially with $2^{\mathcal{O}(N_p)}$ in current formulations.[92] The computation of the Vietoris-Rips filtration can be split into two phases. However, calculating the computational complexity bounds is not straightforward.[93] We note that there is ongoing work aimed at improving and optimizing the persistent homology algorithm, which has resulted in significant gains in performance.[92,94]

A comparison of the approximate run times for SToPS, PECUZAL, and MDOP is provided in Table I. Computation was done on a Ryzen 7 4800HS with 16 GB of RAM using only one thread in order to allow results to be comparable. Despite having a much longer computational time compared to other methods, we note that SToPS allows multiple lags to be considered in parallel as the significance of each potential time lag is evaluated independently. This is in contrast to PECUZAL and MDOP where the iterative embedding cycles approach is used, and each new lag is selected conditional upon previously selected lags. However, there is a new method proposed by Krämer *et al.* based on Monte Carlo tree search that attempts to tackle this problem.[66] The flexibility to assess embedding lags independently may provide large gains in computational speed where the computation for multiple lags may be distributed across multiple threads. Our current implementation of the algorithm does not yet provide support for this.

The selection of new lags conditional on previously selected lags also means that the results of PECUZAL and MDOP are not robust to changes in the maximum allowable lag τ_{max}. Changes to the range of potential lags can affect the order of selection of future lags if a new candidate lag that better optimizes the objective statistic is introduced. For complex data, this can result in widely varying results as the maximum lag changes. The presence of noise in the data also results in different lags as shown by drifting lags found in Fig. 14. Calculated lags for the lobster LP neuron time series were found to vary even across trials with different realizations of identically distributed noise.

We also note that the computation time for these embedding algorithms can vary depending on the complexity of the input time series. For example, successive embedding cycles in PECUZAL typically increase in computation time. For complex or noisy time series, PECUZAL may produce multiple lags of varying significances. This is seen by the jump in estimated lags for the lobster data between max lags of 300 and 600. In contrast, the SToPS algorithm assesses the significance of each score and should grow linearly proportional to the number of lags. An exception is for small lags τ where strands are too short and have no holes' homology to track.

XIV. CONCLUSIONS

One of the aims of this paper is to provide an overview of the embedding fundamentals and review existing methods for optimizing embedding parameters. We first provide in Sec. I a rough overview on the fundamental concepts of embedding theorems and its applications in the context of the time series analysis. A simple case for the usage of embedding in time series prediction tasks is also given for new or uninitiated readers. In this paper, we focus on the problem of identifying good embedding parameters, specifically on the selection of embedding lags in non-uniform embedding. An overview on the considerations when selecting embedding parameters is provided in Sec. III followed by a comprehensive review of various uniform delay embedding methods in Secs. IV–VI.

We argue that a non-uniform embedding approach provides more flexibility in reconstructing fast-slow dynamical systems. Following this, an overview of existing methods that attempt to automate the selection of non-uniform embedding lags is provided in Sec. VII. However, while many of these automated non-uniform embedding methods reliably return a collection of lags, they do not necessarily agree with each other or provide a satisfactory dynamical explanation for their selection. Furthermore, due to the iterative process used to select delays, the choice of each subsequent delay is conditional on previous selections.

In Secs. VIII and IX A, we propose a new method of selecting non-uniform embedding lags, SToPS, that aims to produce lags that have more dynamical explainability and are independently selected. SToPS utilizes persistent homology to detect loops formed by 2D delay embeddings of sampled windows of the time series, which we call "persistent strands." This is done over multiple different lengths of sample windows to produce a characteristic time spectrum $S_\mu(\tau)$ where larger values of the significance score correspond to time scales that are dynamically significant (i.e., they relate to some notion of periodicity in the time series). The structure of each persistent strand loop is characterized by two quantities, circularity and efficiency, which are combined to give the significance score $S_i(\tau)$. Selection of time lags τ is done based on the mean and standard deviation profiles of the significance score $S_\mu(\tau), S_\sigma(\tau)$.

The SToPS embedding method is tested on three different classes of time series: periodic (sum of sines), chaotic (Lorenz), and fast-slow (lobster LP neuron). In all cases, SToPS was found to detect dynamically explainable time scales that were not reflected in other reference non-uniform embedding methods PECUZAL and MDOP. Additionally, SToPS was found to outperform PECUZAL and MDOP in identifying dominant time scales for the lobster LP neuron where fast-slow dynamics are present.

The impact of each different embedding method on the time series prediction performance was also tested. Embedded time series were used to train a one-step neural network predictor. It was found that the resulting models performed similarly across all embedding methods for both the Lorenz and lobster LP neuron time series. However, freerun predictions of the lobster LP neuron time series with models trained using SToPS embedding lags were found to be able to replicate the fast neuron spiking dynamics better than reference embedding methods. We also provide a brief discussion and analysis on the computational efficiency of SToP as well as its robustness to noisy input data.

Overall, while the performance of SToPS is only marginally better than the existing non-uniform embedding methods PECUZAL and MDOP, we argue that SToPS provides lags that are more dynamically explainable compared to its counterparts. The assessment individual time lags also allow the method to be applied to multivariate time series by considering each component as an independent scalar time series and identifying their respective time lags. This may then be used to construct a delay vector that utilizes all components of the time series. The performance of SToPS in this context has not yet been tested and presents as an avenue of further research. Additionally, the independent selection of lags via the characteristic time spectrum provides a clearer picture of the relative importance of each time lag when compared to existing iterative methods for automated non-uniform delay embedding where an explicit collection of lags is provided.

However, the pursuit of more dynamically explainable delay lags introduces a level of subjectivity in the interpretation of the characteristic time spectrum. Nevertheless, we argue that focusing on selecting dynamically relevant and explainable delay lags is potentially a more meaningful approach to construct models that are more relatable to observed system dynamics. This advantage is especially evident in systems with multiple disparate time and spatial scales, as demonstrated by the dynamics of the Lobster LP neuron. Therefore, we propose dynamical relevance and explainability to be a key additional consideration in the future development of time delay embedding methods.

ACKNOWLEDGMENTS

D.C. was supported by the Australian Research Council through the Centre for Transforming Maintenance through Data Science (Grant No. IC180100030), funded by the Australian Government. D.W. and M.S. were supported by the Australian Research

Council (Grant No. DP200102961). S.A. was supported by the Forrest Foundation. E.T. was supported by the Robert & Maude Gledden Foundation and the A.F. Pillow Applied Mathematics Trust. We would also like to thank H. Abarbanel and his team for providing us access to the experimental LP neuron data.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Eugene Tan: Conceptualization (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (lead). **Shannon Algar:** Conceptualization (supporting); Validation (supporting); Writing – review & editing (supporting). **Débora Corrêa:** Investigation (supporting); Supervision (supporting); Writing – review & editing (supporting). **Michael Small:** Investigation (supporting); Methodology (supporting); Project administration (supporting); Supervision (lead); Writing – review & editing (supporting). **Thomas Stemler:** Investigation (supporting); Supervision (supporting); Writing – review & editing (supporting). **David Walker:** Data curation (lead); Validation (supporting); Writing – review & editing (supporting).

DATA AVAILABILITY

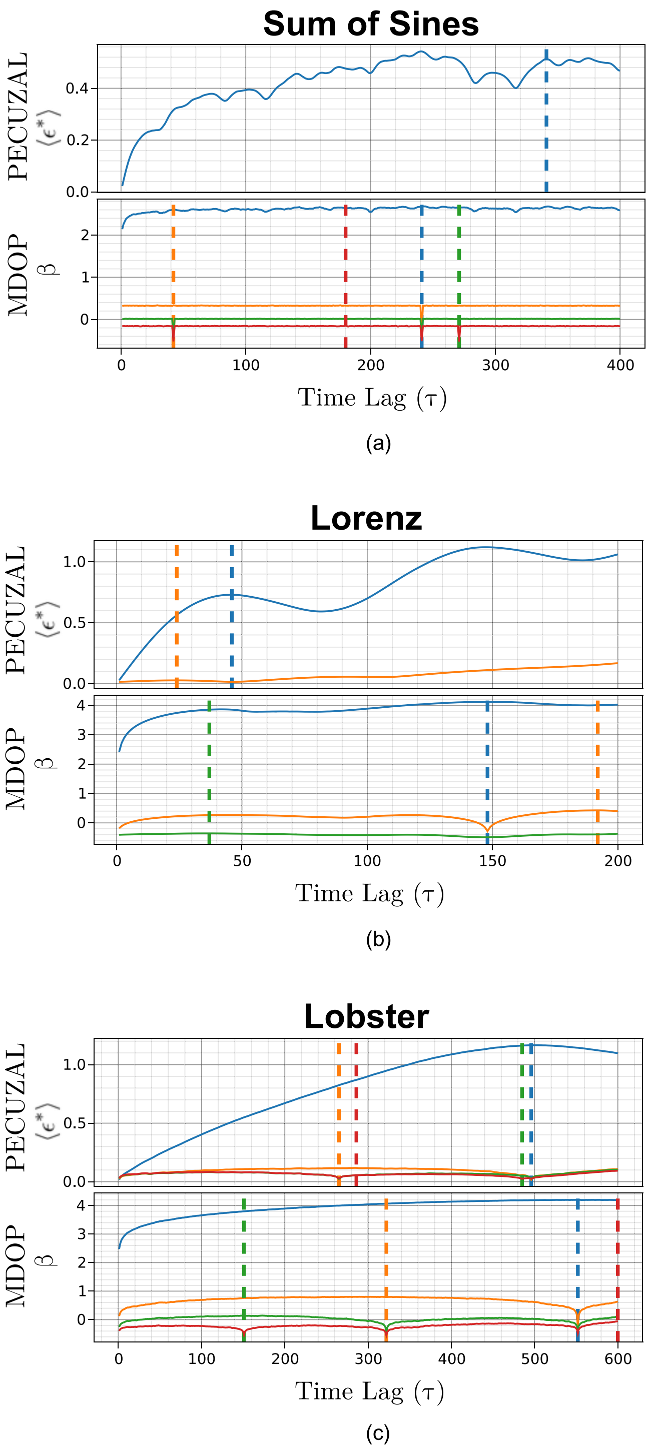
The data that support the findings of this study are available from the corresponding author upon reasonable request and are openly available in Github at https://github.com/eugenetkj98/SToPS˙Public, Ref. 95.

APPENDIX A: NON-UNIFORM EMBEDDING PROFILES

A collection of profiles of the calculated statistics used to select embedding lags the automated non-uniform embedding algorithms PECUZAL and MDOP are given in Fig. 15. Relevant statistics are the continuity statistic $\langle \epsilon^* \rangle$ for PECUZAL and the β statistic for MDOP. For PECUZAL, the τ lags corresponding to local maxima in $\langle \epsilon^* \rangle$ are used as candidate lags. The lag that results in the largest decrease in the L-statistic is chosen as the final embedding lag in each embedding cycle. A similar process is done for MDOP, but the global maxima of β is chosen instead. A termination criterion based on the false nearest neighbor (FNN) statistic is used in conjunction.

APPENDIX B: FREERUN PREDICTION OF NON-SUBSAMPLED DATA

This section contains results of additional freerun prediction tests for models that train on the full lobster LP neuron dataset without any subsampling. This is theoretically an easier task as the model is provided with a larger amount of data and smaller with smaller magnitude predictions in each step. Prediction horizons of 10 steps (Fig. 16) and 30 steps (Fig. 17) were done. The latter's prediction horizon is equal to 10-step prediction horizon models trained on the subsampled data. In the 10-step non-subsampled case, SToPS is



(a)

(b)

(c)

FIG. 15. Resulting embedding profiles for the calculated statistics in the comparison automated non-uniform embedding methods. (a) Sum of three sine terms with $\omega = \{1, 5, 30\}$, $\phi = \{0, 0.25, 0.75\}$, and $A = \{1, 0.5, 0.2\}$. (b) First component of Lorenz. (c) Lobster LP neuron time series.

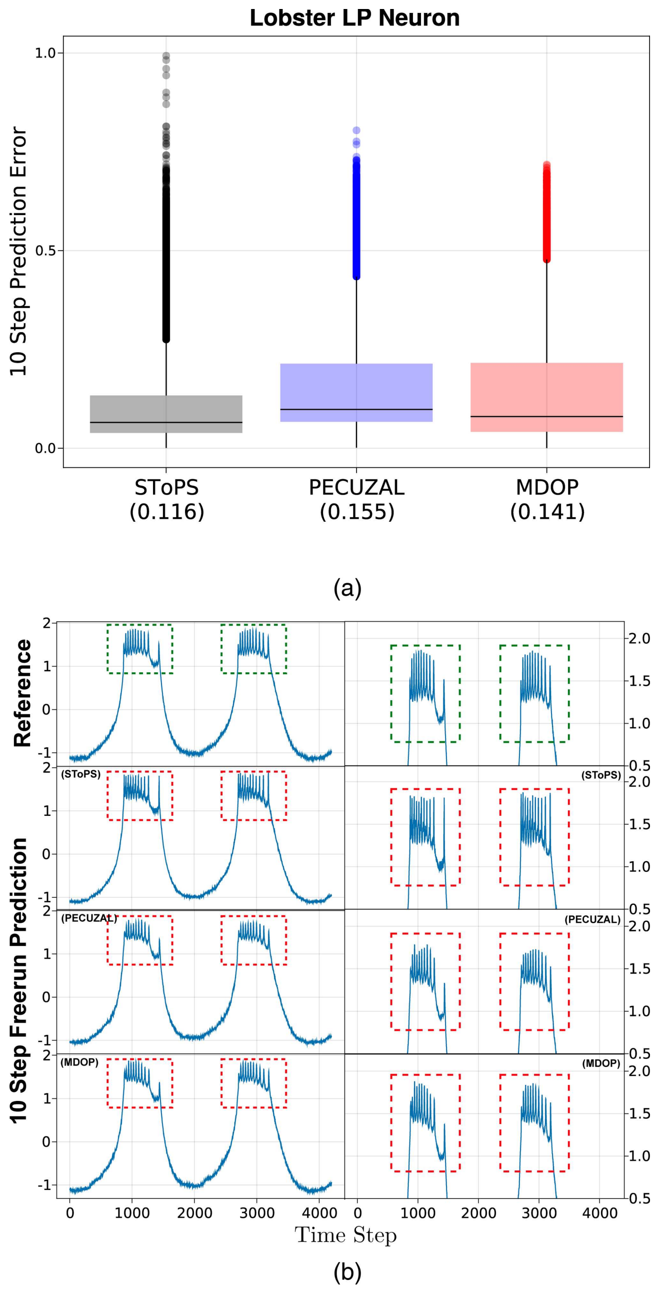


FIG. 16. 10-step freerun prediction results for a neural network model trained on non-subsample data. (a) Comparison of freerun prediction trajectories. (b) Distribution of 10-step mean prediction error.

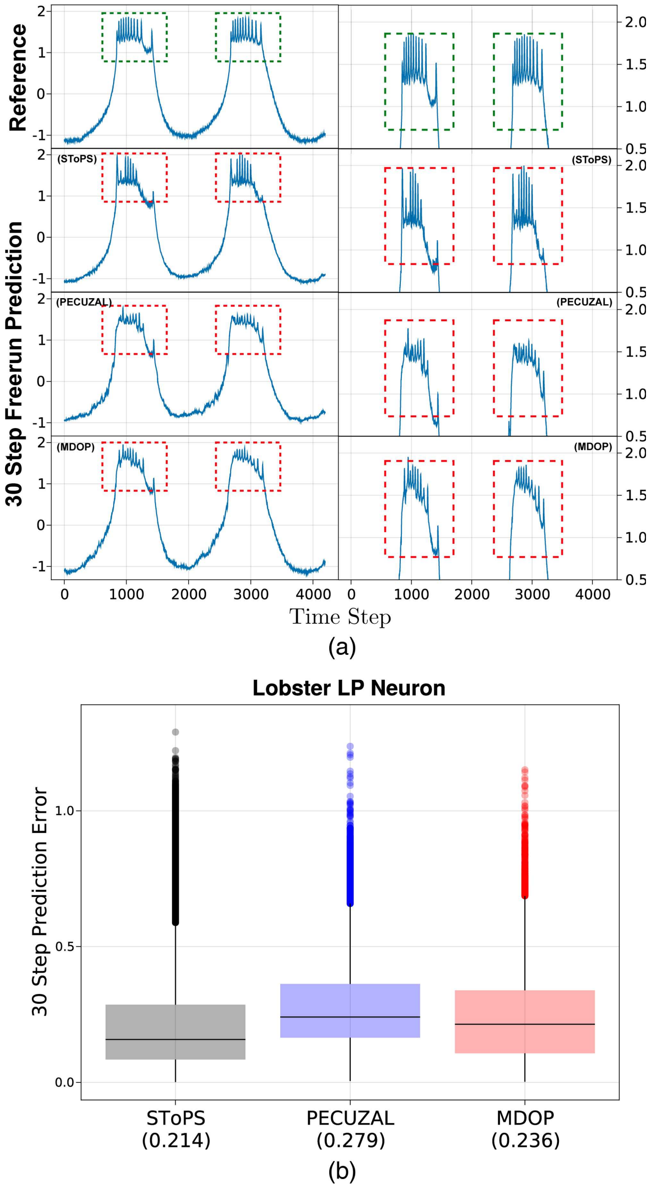


FIG. 17. 30-step freerun prediction results for a neural network model trained on non-subsample data. (a) Comparison of freerun prediction trajectories. (b) Distribution of 30-step mean prediction error.

found to yield a lower prediction error than PECUZAL and MDOP. The replication of the fast spiking dynamics is relatively similar between all methods. Similar results were found for the 30-step prediction case. However, the replication of the fast spiking dynamics is poorer than 10-step prediction cases likely due to the accumulation of errors in successively predicted values.

APPENDIX C: PHASE SPACE RECONSTRUCTION COMPARISONS

A comparison of various reconstructed attractors for the sum of sines and Lorenz time series is provided in Figs. 18 and 19 with the first two detected lags given. For the periodic sum of sines,

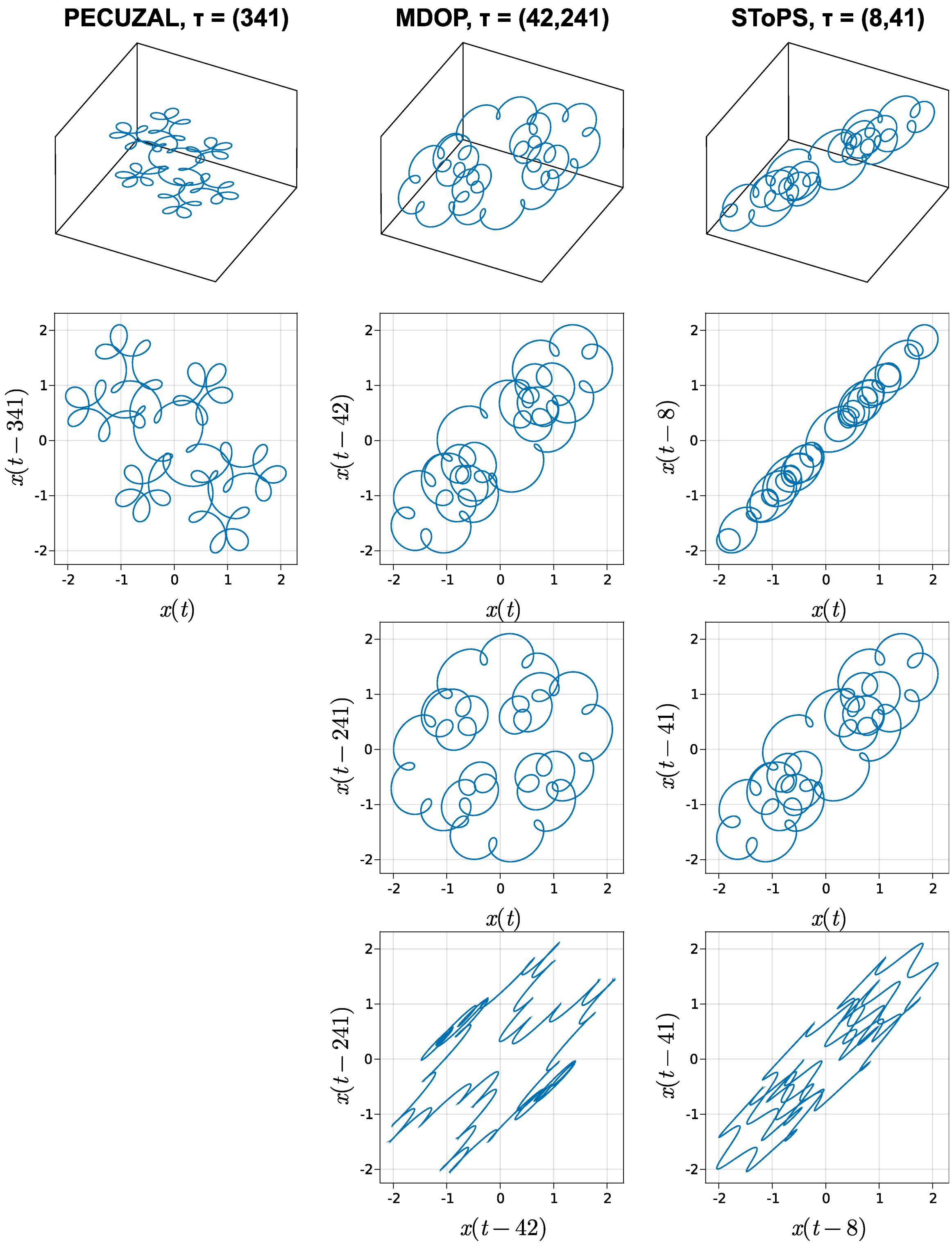


FIG. 18. Sum of sines ($\omega = 1, 5, 30$) phase space reconstructions with PECUZAL, MDOP, and SToPS.

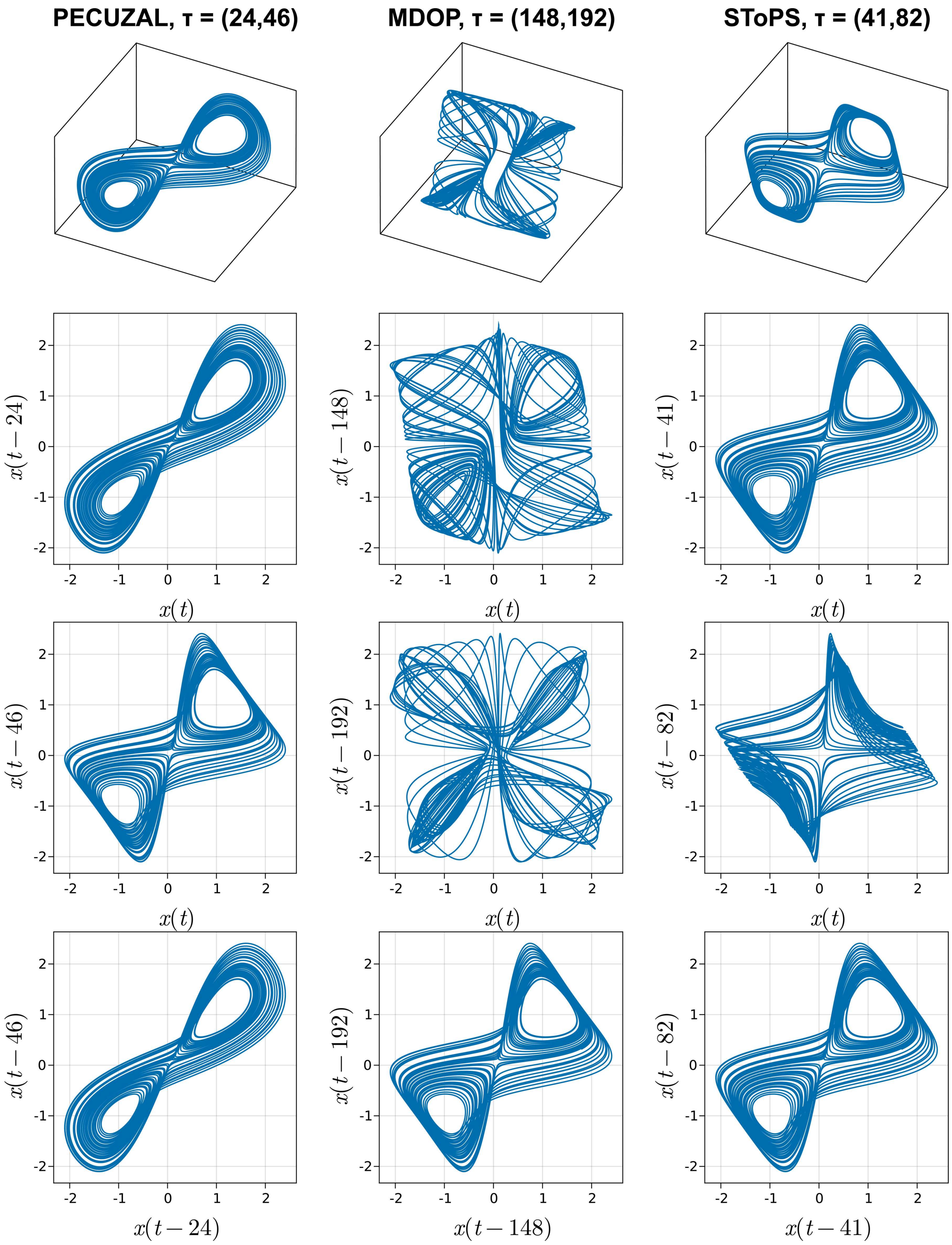


FIG. 19. Lorenz phase space reconstructions with PECUZAL, MDOP, and SToPS.

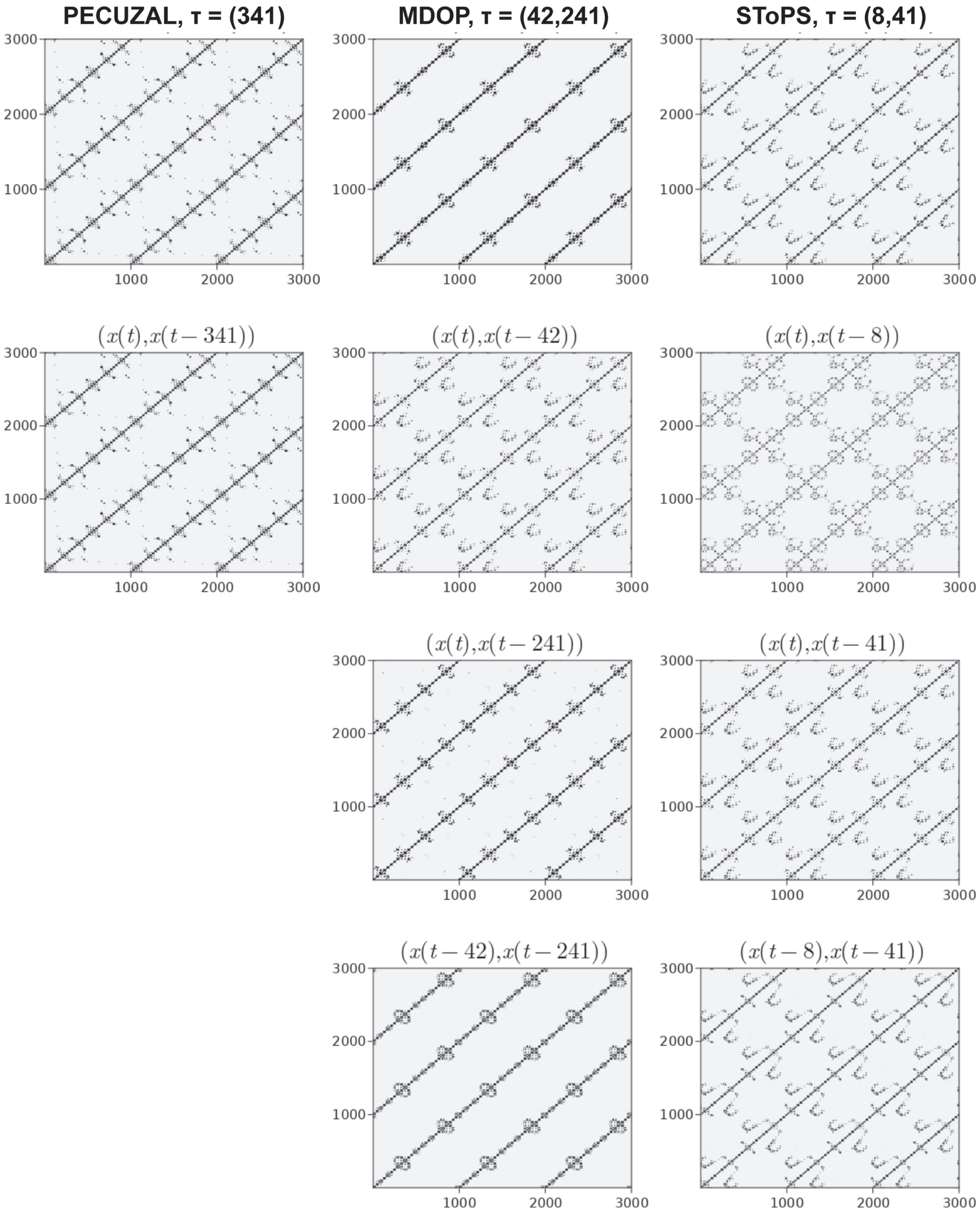


FIG. 20. Sum of sines ($\omega = 1, 5, 30$) recurrence plots of 2D and 3D embeddings.

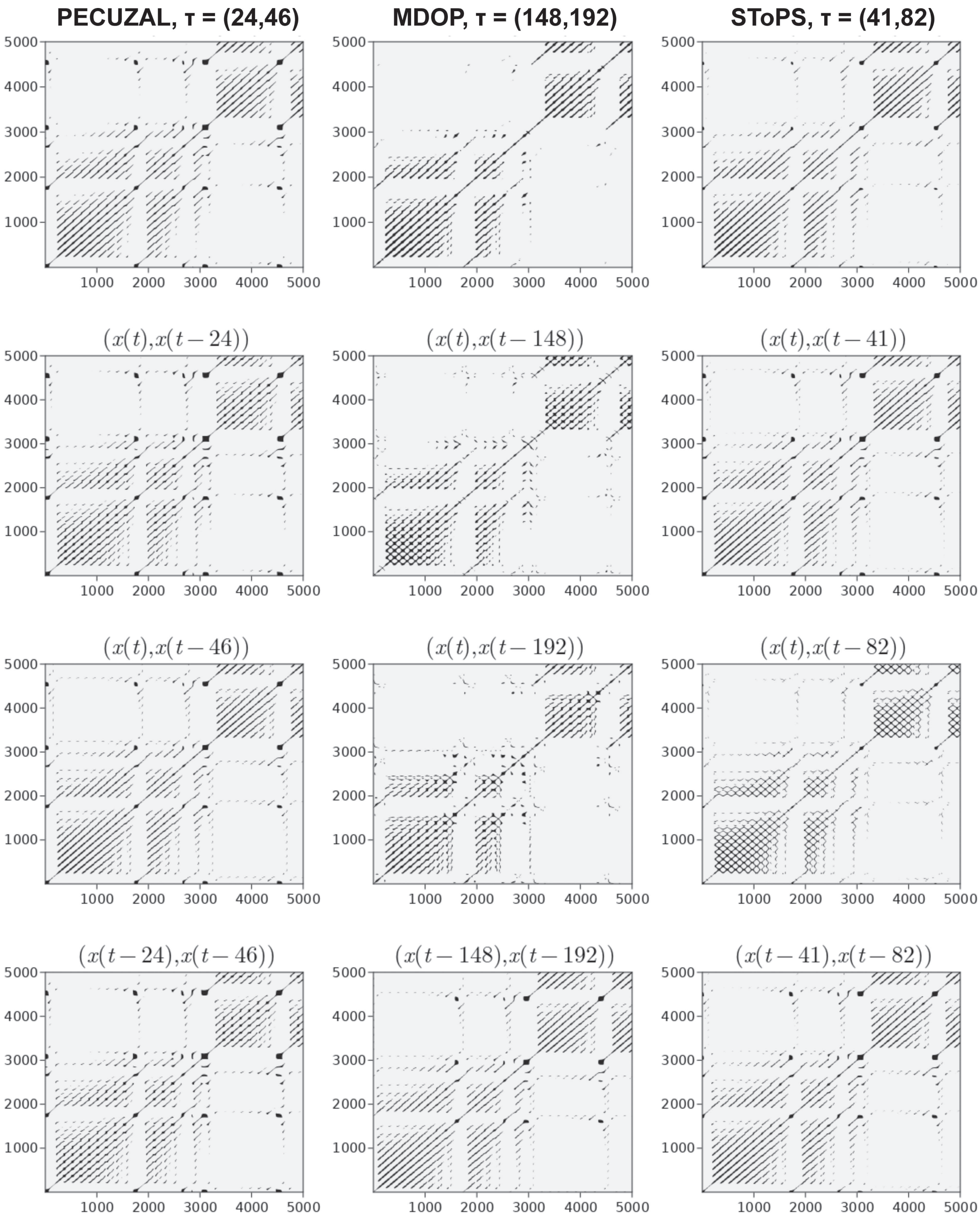


FIG. 21. Lorenz recurrence plots of 2D and 3D embeddings.

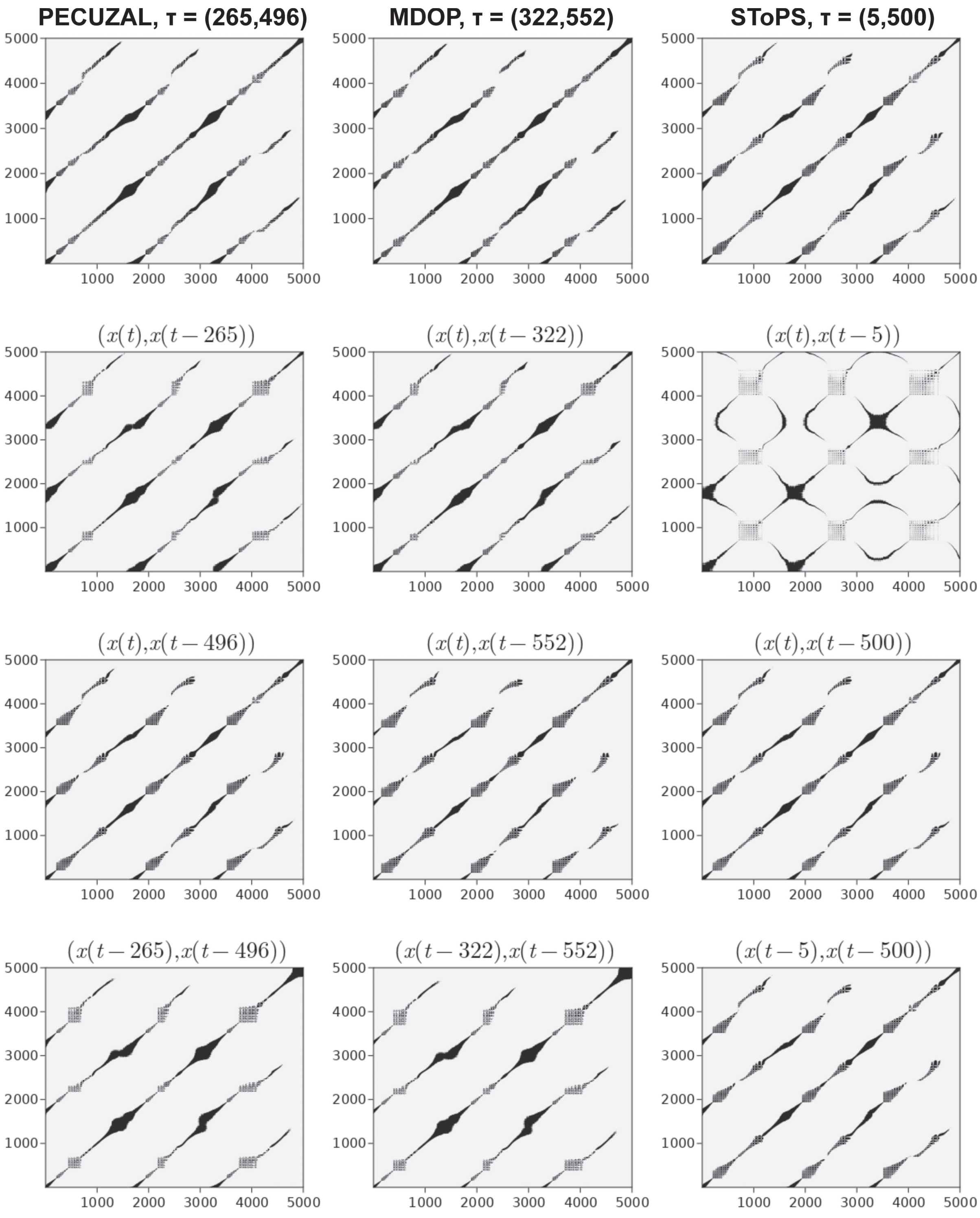


FIG. 22. Lobster LP neuron recurrence plots of 2D and 3D embeddings.

PECUZAL yielded only a single lag. For the Lorenz time series, SToPs only yielded a single peak at $\tau = 41$, which was subsequently used for uniform embedding. Both PECUZAL and SToPS share a similar lag at $\tau \approx 40$. MDOP yields lags that cause overembedding.

Recurrence plots for corresponding 2D and 3D embeddings for the sum of sines, Lorenz, and lobster LP neuron time series are provided in Figs. 20–22. All embedding methods were able to preserve some of the periodic structure for the sum of sines dataset. However, the recurrence plots for MDO and SToPS revealed more small-scale structure than PECUZAL. For Lorenz, both PECUZAL and SToPS yield similar recurrence plots. MDOP recurrence plot loses some detail in comparison and is likely due to overfolding of the attractor caused by large embedding lags. In the lobster LP neuron time series, the selection of a small lag with SToPS reveals the expected periodic behavior in several diagonal regions of the recurrence plot. This is not as clear in PECUZAL and MDOP embeddings where the detected periodic behavior is dominated by the slow neuron dynamics.

REFERENCES

[1] H. Whitney, "Differentiable manifolds," Ann. Math. **37**, 645–680 (1936).

[2] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980* (Springer, 1981), pp. 366–381.

[3] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," J. Stat. Phys. **65**, 579–616 (1991).

[4] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," Phys. Rev. Lett. **45**, 712 (1980).

[5] H. D. Abarbanel, T. Carroll, L. Pecora, J. Sidorowich, and L. S. Tsimring, "Predicting physical variables in time-delay embedding," Phys. Rev. E **49**, 1840 (1994).

[6] D. S. Broomhead and G. P. King, "Extracting qualitative dynamics from experimental data," Physica D **20**, 217–236 (1986).

[7] A. M. Lyapunov, "The general problem of the stability of motion," Int. J. Control **55**, 531–534 (1992).

[8] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," Physica D **16**, 285–317 (1985).

[9] P. Grassberger and I. Procaccia, "Characterization of strange attractors," Phys. Rev. Lett. **50**, 346 (1983).

[10] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," in *The Theory of Chaotic Attractors* (Springer, 2004), pp. 170–189.

[11] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis* (Cambridge University Press, 2004), Vol. 7.

[12] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer Science & Business Media, 2013), Vol. 42.

[13] N. A. Gershenfeld and N. Gershenfeld, *The Nature of Mathematical Modeling* (Cambridge University Press, 1999).

[14] R. Govindan, K. Narayanan, and M. Gopinathan, "On the evidence of deterministic chaos in ECG: Surrogate and predictability analysis," Chaos **8**, 495–502 (1998).

[15] A. S. Weigend and N. A. Gershenfeld, "Results of the time series prediction competition at the Santa Fe institute," in *IEEE International Conference on Neural Networks* (IEEE, 1993), pp. 1786–1793.

[16] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multistep forecasting of chaotic time series," Chaos, Solitons Fractals **139**, 110045 (2020).

[17] J. S. Zhang and X. C. Xiao, "Predicting chaotic time series using recurrent neural network," Chin. Phys. Lett. **17**, 88–90 (2000).

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

[19] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note," Technical Report 148,13 (German National Research Center for Information Technology Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD), Bonn, 2001).

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

[21] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *International Conference on Artificial Neural Networks* (Springer, 1997), pp. 999–1004.

[22] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," IEEE Comput. Intell. Mag. **4**, 24–38 (2009).

[23] C. J. Cellucci, A. M. Albano, and P. Rapp, "Comparative study of embedding methods," Phys. Rev. E **67**, 066210 (2003).

[24] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: The method of surrogate data," Physica D **58**, 77–94 (1992).

[25] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," Phys. Rev. A **33**, 1134 (1986).

[26] M. B. Kennel and H. D. Abarbanel, "False neighbors and false strands: A reliable minimum embedding dimension algorithm," Phys. Rev. E **66**, 026209 (2002).

[27] L. M. Pecora, L. Moniz, J. Nichols, and T. L. Carroll, "A unified approach to attractor reconstruction," Chaos **17**, 013110 (2007).

[28] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, "State space reconstruction in the presence of noise," Physica D **51**, 52–98 (1991).

[29] L. C. Uzal, G. L. Grinblat, and P. F. Verdes, "Optimal reconstruction of dynamical systems: A noise amplification approach," Phys. Rev. E **84**, 016223 (2011).

[30] T. Buzug and G. Pfister, "Comparison of algorithms calculating optimal embedding parameters for delay time coordinates," Physica D **58**, 127–137 (1992).

[31] C. Bandt and B. Pompe, "Permutation entropy: A natural complexity measure for time series," Phys. Rev. Lett. **88**, 174102 (2002).

[32] M. McCullough, M. Small, T. Stemler, and H. H.-C. Iu, "Time lagged ordinal partition networks for capturing dynamics of continuous dynamical systems," Chaos **25**, 053101 (2015).

[33] J. Zhang, J. Zhou, M. Tang, H. Guo, M. Small, and Y. Zou, "Constructing ordinal partition transition networks from multivariate time series," Sci. Rep. **7**, 1–13 (2017).

[34] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proceedings of the 15th European Symposium on Artificial Neural Networks* (Schloss Dagstuhl, Leibniz Center for Informatics, 2007), pp. 471–482.

[35] R. Gilmore, "Topological analysis of chaotic dynamical systems," Rev. Mod. Phys. **70**, 1455 (1998).

[36] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: Principal component analysis," Nat. Methods **14**, 641–642 (2017).

[37] A. I. Mees, P. Rapp, and L. Jennings, "Singular-value decomposition and embedding dimension," Phys. Rev. A **36**, 340 (1987).

[38] M. Paluš and I. Dvořák, "Singular-value decomposition in attractor reconstruction: Pitfalls and precautions," Physica D **55**, 221–234 (1992).

[39] A. Potapov, "Distortions of reconstruction for chaotic attractors," Physica D **101**, 207–226 (1997).

[40] C. Nichkawde, "Optimal state-space reconstruction using derivatives on projected manifold," Phys. Rev. E **87**, 022905 (2013).

[41] J. F. Gibson, J. D. Farmer, M. Casdagli, and S. Eubank, "An analytic approach to practical state space reconstruction," Physica D **57**, 1–30 (1992).

[42] J. Gao and Z. Zheng, "Local exponential divergence plot and optimal embedding of a chaotic time series," Phys. Lett. A **181**, 153–158 (1993).

[43] J. Gao and Z. Zheng, "Direct dynamical test for deterministic chaos," Europhys. Lett. **25**, 485–490 (1994).

[44] W. Liebert, K. Pawelzik, and H. Schuster, "Optimal embeddings of chaotic attractors from topological considerations," Europhys. Lett. **14**, 521 (1991).

[45] R. Hegger and H. Kantz, "Improved false nearest neighbor method to detect determinism in time series data," Phys. Rev. E **60**, 4970–4973 (1999).

[46] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," Phys. Rev. A **45**, 3403–3411 (1992).

[47] S. P. Garcia and J. S. Almeida, "Nearest neighbor embedding with different time delays," Phys. Rev. E **71**, 037204 (2005).

[48] J. Theiler, "Spurious dimension from correlation algorithms applied to limited time-series data," Phys. Rev. A **34**, 2427–2432 (1986).

[49] J. Theiler, "Estimating fractal dimension," J. Opt. Soc. Am. A **7**, 1055–1073 (1990).

[50] J. L. Kaplan and J. A. Yorke, "Chaotic behavior of multidimensional difference equations," in *Functional Differential Equations and Approximation of Fixed Points* (Springer, 1979), pp. 204–227.

[51] J. D. Farmer, E. Ott, and J. A. Yorke, "The dimension of chaotic attractors," Physica D **7**, 153–180 (1983).

[52] E. Ott, T. Sauer, and J. A. Yorke, *Coping with Chaos: Analysis of Chaotic Data and the Exploitation of Chaotic Systems* (Wiley, 1994), p. 41.

[53] J. M. Moore, D. M. Walker, and G. Yan, "Mean local autocovariance provides robust and versatile choice of delay for reconstruction using frequently sampled flowlike data," Phys. Rev. E **101**, 012214 (2020).

[54] H. D. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, "The analysis of observed chaotic data in physical systems," Rev. Mod. Phys. **65**, 1331 (1993).

[55] S. Wallot and D. Mønster, "Calculation of average mutual information (AMI) and false-nearest neighbors (FNN) for the estimation of embedding parameters of multidimensional time series in matlab," Front. Psychol. **9**, 1679 (2018).

[56] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," Phys. Rev. E **69**, 066138 (2004).

[57] A. Papana and D. Kugiumtzis, "Evaluation of mutual information estimators for time series," Int. J. Bifurcat. Chaos **19**, 4197–4215 (2009).

[58] G. A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," IEEE Trans. Inf. Theory **45**, 1315–1321 (1999).

[59] Y.-I. Moon, B. Rajagopalan, and U. Lall, "Estimation of mutual information using kernel density estimators," Phys. Rev. E **52**, 2318 (1995).

[60] K. Judd and A. Mees, "Embedding as a modeling problem," Physica D **120**, 273–286 (1998).

[61] Y. Hirata, H. Suzuki, and K. Aihara, "Reconstructing state spaces from multivariate data using variable delays," Phys. Rev. E **74**, 026202 (2006).

[62] H. D. Abarbanel, R. Brown, and M. B. Kennel, "Variation of Lyapunov exponents on a strange attractor," J. Nonlinear Sci. **1**, 175–199 (1991).

[63] S. D. Algar, T. Stemler, and B. De Saedeleer, "Noise induced jumping dynamics between synchronized modes," Int. J. Bifurcat. Chaos **25**, 1530034 (2015).

[64] K. H. Krämer, G. Datseris, J. Kurths, I. Z. Kiss, J. L. Ocampo-Espindola, and N. Marwan, "A unified and automated approach to attractor reconstruction," New J. Phys. **23**, 033017 (2021).

[65] M. Shen, W.-N. Chen, J. Zhang, H. S.-H. Chung, and O. Kaynak, "Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization," IEEE Trans. Cybern. **43**, 790–802 (2013).

[66] K. H. Krämer, M. Gelbrecht, I. Pavithran, R. Sujith, and N. Marwan, "Optimal state space reconstruction via Monte Carlo decision tree search," Nonlinear Dyn. **108**, 1525–1545 (2022).

[67] D. A. Rand and L.-S. Young, *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80* (Springer, 2006), Vol. 898.

[68] M. Small and K. Judd, "Detecting periodicity in experimental data using linear modeling technique," Phys. Rev. E **59**, 359–376 (1999).

[69] M. Small and C. K. Tse, "Optimal selection of embedding parameters for time series modelling," in *European Conference on Circuits Theory and Design* (European Circuit Society and the Institute of Electrical and Electronic Engineers, 2003).

[70] M. Small and C. K. Tse, "Optimal embedding parameters: A modelling paradigm," Physica D **194**, 283–296 (2004).

[71] J. Rissanen, "MDL denoising," IEEE Trans. Inf. Theory **46**, 2537–2543 (2000).

[72] T. Nakamura, K. Judd, A. I. Mees, and M. Small, "A comparative study of information criteria for model selection," Int. J. Bifurcat. Chaos **16**, 2153–2175 (2006).

[73] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," IEEE Trans. Syst. Man Cybern. Part B Cybern. **26**, 29–41 (1996).

[74] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Trans. Evol. Comput. **1**, 53–66 (1997).

[75] H. Adams, M. Aminian, E. Farnell, M. Kirby, J. Mirth, R. Neville, C. Peterson, and C. Shonkwiler, "A fractal dimension for measures via persistent homology," in *Topological Data Analysis* (Springer, 2020).

[76] J. Jaquette and B. Schweinhart, "Fractal dimension estimation with persistent homology: A comparative study," Commun. Nonlinear Sci. Numer. Simul. **84**, 105163 (2020).

[77] F. A. Khasawneh and E. Munch, "Chatter detection in turning using persistent homology," Mech. Syst. Signal Process. **70-71**, 527–541 (2016).

[78] A. Myers, E. Munch, and F. A. Khasawneh, "Persistent homology of complex networks for dynamic state detection," Phys. Rev. E **100**, 22314 (2019).

[79] E. Tan, D. Corrêa, T. Stemler, and M. Small, "Grading your models: Assessing dynamics learning of models using persistent homology," Chaos **31**, 123109 (2021).

[80] A. D. Myers and F. A. Khasawneh, "Delay parameter selection in permutation entropy using topological data analysis," arXiv:1905.04329 (2022).

[81] L. Vietoris, "Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen," Math. Ann. **97**, 454–472 (1927).

[82] A. Hatcher, *Algebraic Topology* (Cambridge University Press, 2002).

[83] S. Maletić, Y. Zhao, and M. Rajković, "Persistent topological features of dynamical systems," Chaos **26**, 053105 (2016).

[84] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," Phys. Rep. **438**, 237–329 (2007).

[85] B. Braden, "The surveyor's area formula," Coll. Math. J. **17**, 326–337 (1986).

[86] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," Inf. Process. Lett. **1**, 132–133 (1972).

[87] H. D. Abarbanel, R. Huerta, M. I. Rabinovich, N. F. Rulkov, P. F. Rowat, and A. I. Selverston, "Synchronized action of synaptically coupled chaotic model neurons," Neural Comput. **8**, 1567–1602 (1996).

[88] D. Arthur, and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Technical Report (Stanford, 2006).

[89] G. Datseris, "Dynamicalsystems.jl: A Julia software library for chaos and nonlinear dynamics," J. Open Source Software **3**, 598 (2018).

[90] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier, "Persistence images: A stable vector representation of persistent homology," J. Mach. Learn. Res. **18**, 1–35 (2017).

[91] R. Turkeš, J. Nys, T. Verdonck, and S. Latré, "Noise robustness of persistent homology on greyscale images, across filtrations and signatures," PLoS One **16**, e0257215 (2021).

[92] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, "A roadmap for the computation of persistent homology," EPJ Data Sci. **6**, 1–38 (2017).

[93] A. Zomorodian, "Fast construction of the Vietoris-Rips complex," Comput. Graphics **34**, 263–271 (2010).

[94] U. Bauer, "Ripser: Efficient computation of Vietoris–Rips persistence barcodes," J. Appl. Comput. Topol. **5**, 391–423 (2021).

[95] Dataset: E. Tan, C. Débora, S. Algar, T. Stemler, M. Small, and D. Walker (2023). "Significant times on persistent strands," Github. https://github.com/eugenetkj98/SToPS˙Public.