



POTSDAM-INSTITUT FÜR
KLIMAFOLGENFORSCHUNG

Originally published as:

Dhadphale, J. M., [Krämer, K.-H.](#), [Gelbrecht, M.](#), [Kurths, J.](#), [Marwan, N.](#), Sujith, R. I.
(2024): Model adaptive phase space reconstruction. - Chaos, 34, 7, 073125.

DOI: <https://doi.org/10.1063/5.0194330>

RESEARCH ARTICLE | JULY 10 2024

Model adaptive phase space reconstruction

Jayesh M. Dhadphale ; K. Hauke Kraemer ; Maximilian Gelbrecht ; Jürgen Kurths ;
Norbert Marwan ; R. I. Sujith 



Chaos 34, 073125 (2024)

<https://doi.org/10.1063/5.0194330>



Chaos
Special Topic:
Anomalous Diffusion and Fluctuations
in Complex Systems and Networks
[Submit Today](#)



Model adaptive phase space reconstruction

Cite as: Chaos 34, 073125 (2024); doi: 10.1063/5.0194330

Submitted: 26 December 2023 · Accepted: 20 June 2024 ·

Published Online: 10 July 2024



View Online



Export Citation



CrossMark

Jayesh M. Dhadphale,^{1,a)}  K. Hauke Kraemer,²  Maximilian Gelbrecht,^{2,3}  Jürgen Kurths,² 
Norbert Marwan,^{2,4}  and R. I. Sujith^{1,b)} 

AFFILIATIONS

¹Department of Aerospace Engineering, Indian Institute of Technology Madras, Chennai, Tamil Nadu 600036, India

²Potsdam Institute for Climate Impact Research, Member of the Leibniz Association, 14473 Potsdam, Germany

³School of Engineering & Design, Technical University of Munich, 80333 Munich, Germany

⁴Institute of Physics and Astronomy, University of Potsdam, 14476 Potsdam, Germany

^{a)}Author to whom correspondence should be addressed: jayeshmdhadphale@gmail.com

^{b)}Electronic mail: sujith@iitm.ac.in

ABSTRACT

Phase space reconstruction (PSR) methods allow for the analysis of low-dimensional data with methods from dynamical systems theory, but their application to prediction models, such as those from machine learning (ML), is limited. Therefore, we here present a model adaptive phase space reconstruction (MAPSR) method that unifies the process of PSR with the modeling of the dynamical system. MAPSR is a differentiable PSR based on time-delay embedding and enables ML methods for modeling. The quality of the reconstruction is evaluated by the prediction loss. The discrete-time signal is converted into a continuous-time signal to achieve a loss function, which is differentiable with respect to the embedding delays. The delay vector, which stores all potential embedding delays, is updated along with the trainable parameters of the model to minimize prediction loss. Thus, MAPSR does not rely on any threshold or statistical criterion for determining the dimension and the set of delay values for the embedding process. We apply the MAPSR method to uni- and multivariate time series stemming from chaotic dynamical systems and a turbulent combustor. We find that for the Lorenz system, the model trained with the MAPSR method is able to predict chaotic time series for nearly seven to eight Lyapunov time scales, which is found to be much better compared to other PSR methods [AMI-FNN (average mutual information-false nearest neighbor) and PECUZAL (Pecora-Uzal) methods]. For the univariate time series from the turbulent combustor, the long-term cumulative prediction error of the MAPSR method for the regime of chaos stays between other methods, and for the regime of intermittency, MAPSR outperforms other PSR methods.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0194330>

Often, the limited observed variables pose the challenge of characterizing and, in turn, modeling the dynamical system. Due to incomplete observations, the “true,” potentially high dimensional, state of the dynamical system cannot be obtained. Phase space reconstruction (PSR) addresses the problem of reconstructing the higher dimensional representation of the system dynamics, which has a one-to-one correspondence with the actual high-dimensional dynamics of the dynamical system. Several techniques address the issue of PSR where delay embedding is one of the commonly used techniques. The delay embedding technique uses the time series of observed variables and finds a set of suitable time delays based on various criteria, such as average mutual information-false nearest neighbor (AMI-FNN) and noise amplification. Those delays are then used to make an embedding by using lagged copies of the observed variables. The existing methods use delay values that are integer multiple of sampling

time and make the process of delay embedding combinatorial in nature. We propose a novel technique for PSR where we use machine learning (ML) to optimize the embedding delays. We use the prediction error of the model as the objective function. For this, we convert the observed variables, which are discrete in time, into continuous-time variables using interpolation techniques. We, therefore, ensure the differentiability of the objective function with respect to the delays used for PSR. The optimization of the objective function with respect to the time delays and parameters of the model gives the optimal phase space for the assumed model; hence, we name the method *model adaptive phase space reconstruction* (MAPSR). The MAPSR method fits perfectly in the optimization framework (based on automatic differentiation) commonly used for machine learning and can take advantage of advances in ML for developing prediction models.

I. INTRODUCTION

The evolution of deterministic dynamical systems is governed by a set of rules.¹ The quest to discover these rules has led to various discoveries in science. These rules can be identified by deriving the mathematical expressions starting with the first principles.¹ This approach is tedious for dynamical systems with large degrees of freedom, and obtaining predictions is computationally expensive. However, dynamical systems with such large degrees of freedom often exhibit dynamics in a much smaller subset of the entire state space.¹⁻³

A vector in the system's state space, the state vector, defines the dynamical state of the system. In practice, the inaccessibility of a dynamical system often limits the number of measured state variables and, therefore, results in an incomplete state vector. In those cases, it is nevertheless possible to reconstruct the attractor of the unknown state space according to the embedding theorems of Whitney,⁴ Mañé,⁵ and Takens⁶ using different techniques, such as derivative coordinates,^{7,8} Legendre coordinates,⁹ and delay coordinates.¹⁰

The attractor reconstructed from the measured time series data has a similar topology as that of the measured dynamical system,¹¹ i.e., is diffeomorphic to it. The properties, such as the Lyapunov exponent, eigenvalues of fixed points or the fractal dimension, can be preserved under the phase space reconstruction¹¹ (PSR). PSR attempts to create an attractor with a sufficient embedding dimension to avoid the intersection of the trajectories and guarantee diffeomorphic mapping.

Whitney⁴ showed that a generic smooth map F from a D -dimensional compact manifold M to \mathbb{R}^{2D+1} is actually a diffeomorphism on M ; that is, M and $F(M)$ are diffeomorphic. Takens⁶ showed that for the generic dynamical system and the observed quantity, the delay-coordinate map from a D -dimensional smooth compact manifold M to \mathbb{R}^{2D+1} is a diffeomorphism. Sauer *et al.*¹² stated the conditions that ensure that the map from the attractor into the reconstruction space is an embedding, meaning that it is one-to-one and preserves differential information. Thus, for a compact invariant subset A of \mathbb{R}^k , under mild conditions on the dynamical system, almost every delay-coordinate map F from \mathbb{R}^k to \mathbb{R}^n is one-to-one on A provided that $n > 2D_0$,¹² with D_0 being the box-counting dimension of A (see [Appendix A](#)).

For real-world time series, time-delay reconstruction methods try to balance too small delay values, which lead to *redundancy*, and too large delay values, which lead to *irrelevance* of coordinates.^{11,13-15} Since noise is present in real-world time series data, the choice of appropriate delay values is important to avoid amplification of the noise and to keep the complexity of the attractor within limits.^{11,13}

The time series $\vec{s}(t) = [s_i(t); i = 1, \dots, m]$ measured from the dynamical system can be univariate ($m = 1$) or multivariate ($m > 1$). For univariate time series, the delay coordinates with a dimension of d can be represented as $\vec{x}(t) = [s_1(t + \tau_j); j = 1, \dots, d]$. The set of delays can be uniformly spaced, i.e., $\Delta\tau = \tau_{i+1} - \tau_i = \text{const. } \forall i$ [known as uniform time-delay embedding (UTDE)] or non-uniformly spaced (NUTDE). In UTDE, delays and embedding dimension are usually estimated, e.g., using average mutual information (AMI) and false nearest neighbor (FNN) criteria.¹⁶⁻²⁰ For NUTDE, recent work by Kraemer *et al.*²¹ proposes

a method (*PECUZAL*) that unifies the continuity statistic of Pecora *et al.*²² and Pecora *et al.*²³, which quantifies functional dependence, with the L -statistic of Uzal *et al.*,¹³ which quantifies noise amplification. The former can be seen as a delay estimator and the latter as a dimension estimator given those estimated delays (we refer the interested reader to Kraemer *et al.*²⁴). NUTDE and UTDE techniques usually optimize an objective function that quantifies the goodness of the reconstruction, such as the L - or false nearest neighbor-statistic. Kraemer *et al.*²⁴ proposed to solve this optimization with a decision tree search. Most recently, Tan *et al.*²⁵ proposed a method based on persistent homology intending to get delay values for NUTDE, which are independently selected and allow for dynamical explainability. In addition, these authors provide a brief overview of the current embedding techniques.

Brunton *et al.*²⁶ proposed to model the dynamics of the chaotic system as an intermittently forced linear system that combines delay embedding and Koopman theory.²⁷ The intermittent forcing is required when the dynamics is strongly nonlinear and needs to be determined from the time series. Bakarji *et al.*²⁸ use UTDE to get an interpretable closed-form expression of the dynamical system. An encoder maps the reconstructed attractor to low-dimensional space, where a closed-form model is obtained for this encoded attractor using the SINDy method.²⁹

Ultimately, one goal of PSR approaches is to predict the system in question given its incomplete observations. Predictions based on PSR can, e.g., use models that extrapolate based on neighborhoods in the reconstructed phase space. Several approaches have been made that differ in the exact way a local neighborhood-based model is built.³⁰⁻³⁴ Alternatively, Dhadphale *et al.*³⁵ used a delay embedding technique along with a neural ODE approach in order to yield a suitable model for a thermoacoustic system.³⁶ Neural ODEs³⁷ are a natural candidate for data-driven modeling of dynamical systems. They integrate artificial neural networks (ANNs) into the right-hand side of differential equations. As ANNs are universal function approximators, a neural ODE is trivially a universal dynamical system approximator. What kind of PSR is optimal in this case? Kraemer *et al.*²⁴ were optimizing the PSR for predictions with a decision tree search. However, this approach is costly.

In this article, we, therefore, propose an alternative: a differentiable variant of a time-delay embedding that makes use of ANNs and optimizes for the prediction loss. The method simultaneously updates the time delays $\vec{\tau}$ and the trainable parameters (excluding hyperparameters) of the mathematical model for the system to minimize the assumed objective function. Hence, we name the method as model adaptive phase space reconstruction (MAPSR). We showcase this approach with neural ODEs, but the MAPSR method fits perfectly in machine learning frameworks and can readily be used with other data-driven models.

Three other frameworks can be thought of as related to our proposal. The reservoir computing (RC) framework can also be used to model the dynamical system from available low-dimensional data.^{38,39} For the resemblance of RC with the delay embedding, we refer to Duan *et al.*⁴⁰ The latent ODE framework⁴¹ is a method for sequence-to-sequence learning that tries to learn an ODE in an adaptively learned latent space. Augmented neural ODEs⁴² add unobserved, latent dimensions to an otherwise unmodified neural

ODE. Compared to these methods, our approach, MAPSR, also achieves an interpretable PSR that we can investigate further.

The remainder of this paper is organized as follows: The proposed methodology of the MAPSR method is described in Sec. II. Section III presents the results and discusses how the MAPSR performs on standard dynamical systems and on time series obtained from real-world dynamical systems. The key features and limitations of the method are summarized in Sec. IV. The brief algorithm of the MAPSR is presented in Appendix C.

II. DESCRIPTION OF THE METHOD

The first step in nonlinear time series analysis is often the PSR from available time series data.⁴³ The PSR is conventionally performed by targeting the independence of the selected coordinates by optimizing an objective function, which reflects such an independence.^{16–20,22,23} At the same time, the obtained reconstructed trajectory is not optimized for a specific application or analysis. Kraemer *et al.*²⁴ have discussed this issue, provided a modular way to choose the statistic and objective function for the delay selection according to the research question, and used the MCDTS (Monte Carlo Decision Tree Search) method to obtain global minima. The objective of the current work is to make the parameters of the phase space reconstruction differentiable to avoid a combinatorial selection of delay values. This step allows us to use a common optimization framework to determine the optimal parameters for the phase space and trainable parameters of the assumed mathematical model of the dynamical system. Specifically, we intend to create an initial delay vector $\vec{\tau}$ of a certain initial dimension D_{init} , which gives us an initial trajectory in our reconstruction phase space. We will use this trajectory for training a mathematical model, specifically an ANN, which can approximate the underlying ordinary differential equation from which the trajectory can be obtained by integration. The prediction error for a certain prediction horizon will serve as the loss function. The model will, therefore, depend on $\vec{\tau}$ and optimize this vector along with its own parameters via a gradient descent method. We allow the delay vector to reduce in size, i.e., reduction of D_{init} , during training.

Conventional methods for PSR attempt to find the delays that are multiples of the sampling time Δt .^{16–20} Here, we remove this restriction by converting the discrete-time measurements into continuous-time variables using interpolation, which allows delays to take continuous values. The advantage of this conversion is that a continuous variable is now piecewise differentiable. Suppose the measured time series $\vec{s}(t)$ is a multivariate vector time series with m components or variables, i.e., $\vec{s}(t) = [s_i(t)|i = 1, \dots, m]$. We define the delay vector

$$\vec{\tau} = [\tau_{1,1}, \tau_{1,2}, \dots, \tau_{1,d_1}, \tau_{2,1}, \dots, \tau_{2,d_2}, \dots, \tau_{m,d_m}], \quad (1)$$

where $\tau_{i,j}$ is the j th delay associated with the i th measured variable. Here, for the i th measured variable, there are d_i delay values; i.e., $\vec{\tau}$ has $D = \sum_i d_i$ components, and D corresponds to the dimension of the phase space. The vector $\vec{\tau}$ is initialized such that the first delay value associated with all the time series is set to zero; i.e., $\tau_{i,1} = 0, \forall i$. The remaining components of the $\vec{\tau}$ are initialized assuming UTDE for individual time series. For example, the time delay ($\tau_{i,AMI}$) and

dimension ($d_{i,AMI}$) obtained from the first minimum of the auto-mutual information $AMI(\tau)$ together with a dimension estimator, such as the FNN-statistic, provide estimates for the order of the common difference $\Delta\tau_i$ and the initial dimension (d_i) for each time series. This means that while initializing $\vec{\tau}$, the common difference between successive delay values, $\Delta\tau_i = \tau_{i,j+1} - \tau_{i,j}$, for the i th time series is maintained constant such that $\Delta\tau_i \sim O(\Delta\tau_{i,AMI})$. Note that UTDE is only used to initialize the $\vec{\tau}$ vector before training. $\vec{\tau}$ is initialized with $(d_i)_{init} = d_{init}$, and the common difference $\Delta\tau$ is the same for all time series. For simplicity, we will refer to d_{init} in the subsequent plots; i.e., the initial dimension of the delay vector is $D_{init} = \sum_{i=1}^m (d_i)_{init} = d_{init}m$, where m is the number of time series. In practice, we train the model for different initial dimensions $d_{init} = 1, \dots, d_{max}$ and eventually, the model with the initial dimension d_{init} with the minimum loss after the training, i.e., $\mathcal{L}_{min}(d_{init})$, gives the optimal set of delays. d_{max} can be set to an arbitrary; yet, a large enough value or can be of the order of the dimension estimated using the AMI-FNN method for an educated guess.

Further below, we describe how the delay values may merge during training (cf. Sec. II B). Thus, the final number of delay values for each time series ($d_i)_{final}$ and D_{final} might be less than D_{init} . The initially chosen dimension of $\vec{\tau}$ must be chosen sufficiently large. When set too small, we do not expect the model to perform well, which will be reflected in the training loss $\mathcal{L}(d_{init})$ not achieving a minimum with respect to other training based on larger D_{init} . Our expectation is that for a sufficiently high initial dimension D_{init} , the training loss $\mathcal{L}(d_{init})$ is minimal.

A. Training the model

With MAPSR, we now allow $\tau_{i,j}$ to be a non-integer multiple of the sampling time Δt . Then, the time series $\vec{s}(t)$ is interpolated to get the vector of delay coordinates with these non-integer delays. At time, $t = n\Delta t$ where n is a non-negative integer, the vector of delay coordinates is $\vec{x}(n\Delta t, \vec{\tau}) = [s_1(\tau_{1,1} + n\Delta t), \dots, s_1(\tau_{1,d_1} + n\Delta t), s_2(\tau_{2,1} + n\Delta t), \dots, s_2(\tau_{2,d_2} + n\Delta t), \dots, s_m(\tau_{m,d_m} + n\Delta t)]$. In this paper, we have used linear interpolation to compute $s_i(t + \tau_{i,j})$, but other interpolation techniques can be easily incorporated as well.

For example, $s_i(t + \tau_{i,j})$ can be computed using linear interpolation as

$$s_i(t + \tau_{i,j}) = (1 - \beta)s_i(h\Delta t) + \beta s_i((h + 1)\Delta t), \quad (2)$$

where $\beta = (t + \tau_{i,j} - h\Delta t)/\Delta t$ and time instance $(t + \tau_{i,j})$ lies between h th and $(h + 1)$ th sampling instances, i.e., $h\Delta t \leq (t + \tau_{i,j}) < (h + 1)\Delta t$. The interpolation gives the time series of $\vec{x}(n\Delta t, \vec{\tau})$ with the sampling time Δt being identical to the sampling time of the original time series.

After the first step of defining the delay vector and obtaining the delay coordinates, the next step in MAPSR is modeling the dynamics. The modeling aims to determine the function \vec{f} such that $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t, \vec{\tau}), W)$, where $\dot{\vec{x}}$ is the time derivative of the reconstructed state vector \vec{x} and W is the set of parameters that governs the dynamical behavior of the model. The model can be linear or nonlinear where \vec{f} is differentiable with respect to W and $\vec{\tau}$. The functional form of \vec{f} is usually unknown, but can be approximated by universal

function approximators, such as ANNs.⁴⁴ ANNs that approximate the right-hand side of an ODE are known as neural ODEs.³⁷ Thus, the differential equation for the trajectories yielded from the given time series time-delay embedded with the delay vector $\vec{\tau}$, Eq. (1), can be approximated with a neural ODE. For the rest of the paper, \vec{f} is approximated with neural ODE, but the proposed method is not limited to neural ODE and allows for other methods as well.

The performance of the trained model and the goodness of the reconstructed phase space can be tested by time integrating the model and quantifying the prediction loss. The time series of the delay coordinates obtained after interpolation gives the initial state of the system as $\vec{x}(t_0)$ (Fig. 1). The future state of the system after r time steps, i.e., $\vec{x}_{r,pred} = \vec{x}_{pred}(t_0 + r\Delta t, \vec{\tau})$, is predicted as

$$\vec{x}_{r,pred} = \vec{x}_{true}(t_0, \vec{\tau}) + \int_{t_0}^{t_0+r\Delta t} f(\vec{x}_{pred}(t, \vec{\tau}), W) dt. \quad (3)$$

Let the prediction horizon be $T_R = R\Delta t$, where $R = \max(r)$, is fixed to be of the order of a characteristic time scale of the system, such as the Lyapunov time scale or the period in the case of limit cycle oscillations (LCOs).

Suppose K points are randomly chosen from the reconstructed trajectory, and the k th selected point corresponds to the time instance t_k . The state of the dynamical system at each of these K points is treated as the initial condition, and future states are predicted R time steps ahead. The variable K is commonly referred to as batch size. The loss in the prediction is computed by averaging the

prediction error as

$$Loss = \mathcal{L}(\vec{\tau}, W, R, d_{init}) = \left\langle \sum_{r=1}^R \left\| \vec{x}_{r,true} - \vec{x}_{r,pred} \right\|_p^p \right\rangle_k \quad (4)$$

and depends on the chosen prediction horizon $T_R = R\Delta t$. Here, $\|\cdot\|_p$ is the p th norm, which is raised to the power p as and $\langle \cdot \rangle_k$ indicates the average over the batch. The loss function quantifies the prediction error of the model by comparing the predicted states $\vec{x}_{r,pred}$ using the model with the future states obtained with interpolation of the measured time series $\vec{x}_{r,true}$.

The above description shows the steps in the forward pass, i.e., how the delay vector is initialized, the model is used to perform prediction, and how the loss is computed. The MAPSR method aims to optimize $\mathcal{L}(R, d_{init})$ with respect to $\vec{\tau}$ and W . This optimization needs the gradient of the loss function with respect to $\vec{\tau}$ and W as $\nabla_{\vec{\tau}}(\mathcal{L})$ and $\nabla_W(\mathcal{L})$, respectively. The derivative of \mathcal{L} with respect to τ_i , which is the i th component of $\vec{\tau}$, can be computed as

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = \left\langle \sum_{r=1}^R \left[p \left[(\vec{x}_{r,true})_i - (\vec{x}_{r,pred})_i \right]^{p-1} \frac{\partial (\vec{x}_{r,true})_i}{\partial \tau_i} - \sum_{q=1}^D p \left[(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q \right]^{p-1} \frac{\partial (\vec{x}_{r,pred})_q}{\partial (\vec{x}_{0,true}(\vec{\tau}))_i} \frac{\partial (\vec{x}_{0,true}(\vec{\tau}))_i}{\partial \tau_i} \right] \right\rangle_k, \quad (5)$$

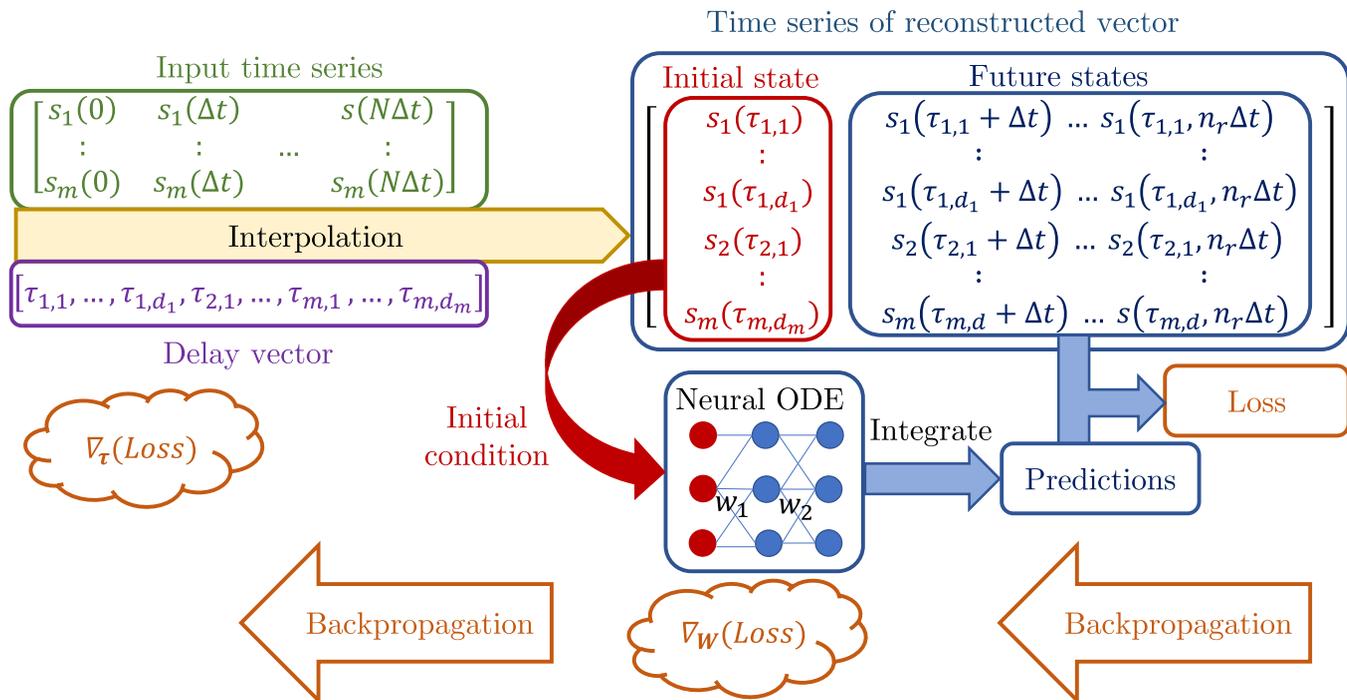


FIG. 1. Flow chart of the model adaptive phase space reconstruction during training. Arrows from left to right (forward pass) show the steps involved in the computation of the loss function. During the backpropagation, the gradient of the loss is computed with respect to W and $\vec{\tau}$.

where $(\vec{x}, \cdot)_i$ indicates the i th component of a vector (\vec{x}, \cdot) . Interpolation is used to calculate $\vec{x}_{r,true} = \vec{x}_{true}(t_0 + r\Delta t, \vec{\tau})$ using the time series of $\vec{s}(t)$. Therefore, the derivative of $\vec{x}_{r,true}$ with respect to τ_i depends on the interpolation scheme. For linear interpolation, Eq. (2), this leads to $\partial(\vec{x}_{r,true})_q / \partial \tau_i = \delta_{iq} [(\vec{x}_{r+1,true})_q - (\vec{x}_{r,true})_q] / \Delta t$, with δ_{iq} being the Kronecker delta function.

Furthermore, the derivative of \mathcal{L} with respect to $W_{ij}^{(l)}$, which is the (i, j) element of the (l) th layer of the weight matrix W , can be computed as

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \left\langle - \sum_{r=1}^R \sum_{q=1}^D p[(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^{p-1} \frac{\partial (\vec{x}_{r,pred})_q}{\partial W_{ij}^{(l)}} \right\rangle_k. \quad (6)$$

Here, $\partial(\vec{x}_{r,pred})_q / \partial W_{ij}^{(l)}$ and $\partial(\vec{x}_{r,pred})_q / \partial(\vec{x}_{0,true}(\vec{\tau}))_i$ from Eq. (5) are the derivatives of integration with respect to the neural ODE parameter and the initial condition. These derivatives can be computed using the backpropagation method^{45,46} or the adjoint method.³⁷ Backpropagation is one of the methods to compute the derivative of the loss function with respect to the differentiable parameters by back-tracing all the operations starting from the loss.⁴⁷ The introduction of the interpolation method to compute delay coordinates allows the backpropagation algorithm to calculate $\nabla_{\vec{\tau}}(\mathcal{L})$ and also allows to apply this method to non-equidistantly sampled time series or time series with (small) gaps, which is the crucial idea we introduce in this paper. After computing $\frac{\partial \mathcal{L}}{\partial \tau_i}$ and $\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}}$,

the elements of W and $\vec{\tau}$ can be updated using different optimizers. The results presented in this paper use the RMSprop algorithm for optimization (details of which can be found in Tieleman *et al.*⁴⁸).

The minimum component of the delay vector is subtracted from itself to maintain the minimum delay value as zero and all other delay values as non-negative. Thus, $\vec{\tau}$ is redefined as

$$\vec{\tau} := \vec{\tau} - \min(\vec{\tau}) \quad (7)$$

after each optimization step. That is, the coordinates of $\vec{\tau}$ in Eq. (7) are most likely non-integer multiples of Δt values. After each optimization step, a new trajectory is constructed from the interpolated time series using those delay values, as shown in the first step in Fig. 1.

B. Dimension reduction while training

During the iterative update of the delay vector, if two delay values associated with the i th time series, say, τ_{ij} and τ_{ik} , are closer than a certain threshold τ_{th} , i.e., $|\tau_{ij} - \tau_{ik}| < \tau_{th}$, one of the delay values is removed from the $\vec{\tau}$. Removing the delay values is equivalent to removing the associated delay coordinates from the PSR, thus decreasing the dimension D of the reconstructed state space. We term this as *delay merging*. Very close delay values are associated with *redundant* delay coordinates, which do not convey any additional information about the state of the system and can, therefore, be removed. After merging the delays, the dimension of the embedding vector \vec{x} reduces. Suppose the weight matrix associated with the first layer of the neural ODE is W_1 , which maps the input vector \vec{x} as $\vec{z}_1 = W_1^T \vec{x}$. If x_i and x_j merge, to avoid restarting the training from scratch, the i th row in the weight matrix W_1 is replaced by

the addition of the i th and the j th rows, and furthermore, the j th row is removed from the matrix. The j th column from the weight matrix associated with the last layer of neural ODE is also removed to reduce the dimension of the output vector. This modification of the weight matrices safely reduces the number of nodes in the input and output layer, i.e., reduces the embedding dimension without affecting the neural ODE model. During training, the delay values may merge, and the final number of delay values $(d_i)_{final}$ might be less than d_{init} . As mentioned earlier, the minimum loss that can be achieved during training does not only depend on the prediction horizon R , but also on the initial dimension of $\vec{\tau}$. The optimal delay embedding is selected based on the minimum of the $\mathcal{L}(d_{init})$,

$$\mathcal{L}_{\min}(d_{init}^{(opt)}) = \min_{d_{init} \in [1, d_{\max}]} \mathcal{L}(d_{init}). \quad (8)$$

The optimal initial dimension $d_{init}^{(opt)}$ for which loss is minimum gives the optimal delay embedding,

$$\vec{\tau}_{init}^{(opt)} = [\tau_{1,1}, \tau_{1,2}, \dots, \tau_{1,d_{init}^{(opt)}}, \tau_{2,1}, \dots, \tau_{2,d_{init}^{(opt)}}, \dots, \tau_{m,d_{init}^{(opt)}}], \quad (9)$$

and the delay vector after training can be written as

$$\vec{\tau}_{final}^{(opt)} = [\tau_{1,1}, \tau_{1,2}, \dots, \tau_{1,d_{1,final}^{(opt)}}, \tau_{2,1}, \dots, \tau_{2,d_{2,final}^{(opt)}}, \dots, \tau_{m,d_{m,final}^{(opt)}}]. \quad (10)$$

From Eq. (9), we can see that initially for all time series $(d_i)_{init} = d_{init}^{(opt)}$ and from Eq. (10), we can see that after training, $(d_i)_{final} = d_{i,final}^{(opt)}$. Thus, if there are m time series, then due to merging of the delays, $\sum_i d_{i,final}^{(opt)} \leq m d_{init}^{(opt)}$. Algorithm 1 gives the step-by-step description of the MAPSR method.

The expected behavior of the loss function is that it decreases with increasing the initial dimension d_{init} until it is sufficiently large so that trajectories do not intersect anymore. Increasing d_{init} further is not expected to result in any further decrease of the loss function. Contrastingly, our investigation reveals that the loss function increases after the optimal dimension. For the D dimensional PSR, all the D components of the $(\vec{x}_{r,true} - \vec{x}_{r,pred})$ contribute to the loss function [Eq. (4)]. Below the optimal dimension, the trajectories cannot be sufficiently resolved, and the loss decreases initially with the initial dimension. Once the trajectories are well resolved and can be captured by the model, the loss function attains minima. There is a discrepancy between the trajectories predicted by the model and true trajectories due to imperfect modeling or noise in the data. Adding the new delay coordinate after the optimal dimension might just increase this discrepancy in the loss function and cause it to increase. This contribution to the loss function keeps increasing upon adding new delay coordinates, and therefore, the loss keeps growing after the optimal dimension. We discuss this in more detail in Sec. III E.

III. RESULTS AND DISCUSSION

We test the MAPSR method on the time series from selected dynamical systems, such as univariate time series from a harmonic oscillator and the Lorenz system, and also on univariate and multivariate time series acquired from a real system, here a turbulent

combustor. Subsections III A–III E present the results obtained using the MAPSR method for univariate and multivariate time series data.

A. Application of the MAPSR method to univariate time series from the harmonic oscillator

A univariate time series for the harmonic oscillator is generated by the expression, $s_1(t_n) = \sin(2\pi t_n) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ represents noise, i.e., ϵ is normally distributed with zero mean and variance σ^2 . We use six noise levels, i.e., clean signal ($\sigma = 0$) and noisy signals with σ equal to 1%–5% in steps of 1% of the amplitude of s_1 . The amplitude is computed as the mean of $|\max(s_1)|$ and $|\min(s_1)|$. The time series is normalized to have zero mean and a maximum absolute value of one. The time series is evenly sampled with the sampling period $\Delta t = 0.01$ s.

The dynamical system, $\dot{\vec{x}} = \vec{f}(\vec{x}, W)$, is known to be linear for a harmonic oscillator; hence, a neural ODE with direct linear mapping from the input to the output is used without any nonlinear activation function. Hence, $\dot{\vec{x}} = W_1^T \vec{x} + \vec{b}_1$; here, W_1 is the weight matrix and \vec{b}_1 is the bias. For the D dimensional phase space, the weight matrix W_1 has a dimension of $D \times D$. The weight matrix is randomly initialized before training.

For the time series of the harmonic oscillator considered in the current study, the AMI-FNN method predicts the optimal

delay as $\Delta\tau_{AMI} = 25\Delta t = 0.25$ s. To initialize the delay vector for the MAPSR method, we use a common difference of $\Delta\tau = 0.20$ s $\sim O(\Delta\tau_{AMI})$. For example, for three-dimensional phase space, $\vec{\tau}_{init} = [0.00, 0.20, 0.40]$ s = $\Delta\tau[0, 1, 2]$. The delay vector for each of the six cases (with different noise levels) is initialized as a d dimensional vector $\vec{\tau} = \Delta\tau[0, 1, \dots, d - 1]$ here, due to the time series being univariate, the dimension of the delay vector D is the same as that of d ; i.e., $D = d$. Each case is tested by varying d from 1 to 5 in steps of 1. Linear interpolation is used to obtain the time series of the reconstructed state vector from the input time series based on the delay vector $\vec{\tau}$.

Training of the neural ODE is performed on 100 trajectories (batch size = 100) randomly chosen from the reconstructed attractor. Here, each trajectory is of 0.5 s duration; i.e., batch time is 0.5 s or $t_{batch} = 0.5$ s. A new batch is chosen for each training iteration. The loss is computed using Eq. (4). The learning rate for W is chosen as $\alpha_W = 10^{-3}$ and for $\vec{\tau}$ as $\alpha_{\vec{\tau}} = 10^{-5}$. The training of the neural ODE is performed for fixed 20 000 iterations [it is observed but not shown here that within 20 000 iterations, the loss function converges to a steady value and the components of the delay vector reach a steady value; refer Fig. 2(b)]. The average loss for the last 100 iterations of training is shown in Fig. 2(a). The figure shows that the training loss is minimum for the time series with 0% noise and increases with the noise level. The minimum of the loss occurs for the phase space of dimension two for all the time series with different noise levels. This might seem contradictory to the expected behavior

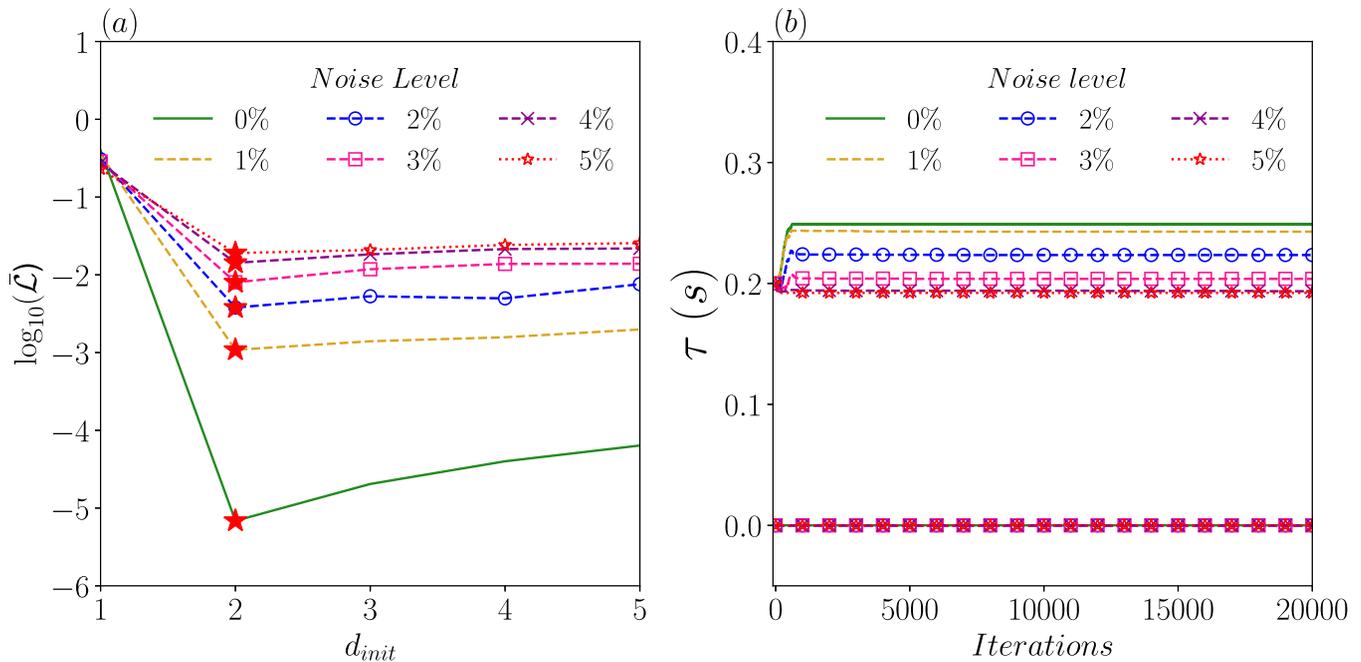


FIG. 2. Application of the MAPSR method to time series of a harmonic oscillator with time period $T = 1$. (a) Variation of the prediction loss in log scale $\log_{10}(\mathcal{L})$ with the initial dimension of the delay vector. The MAPSR is applied to time series with six noise levels, i.e., $\sigma = 0\%$ – 5% of the amplitude of the signal. The dimension at minima of $\log_{10}(\mathcal{L})$ is selected as the embedding dimension (red star). (b) The evolution of the selected delay vector for the optimal embedding dimension. Here, $d = 2$ is the optimum dimension for all noise levels. The delay vector was initialized as $[0, 0.20]$ s for all the noise levels and updated during training. For low noise levels, the delay vector approaches $[0, 0.25]$ s, which corresponds to linearly independent coordinates.

for the loss function; that is, on increasing the dimension, the loss should decrease until trajectories get well resolved and should stay low afterward. The reasons for this observed behavior are discussed in Sec. III E.

The evolution of the delay values for the phase space with optimal dimension (with minimum loss) is shown in Fig. 2(b). The optimal dimensions predicted by the MAPSR method are 2 for all the time series with different noise levels. As discussed before, the figure shows the evolution of the delay vector for different time series with the delay vector initialized to $\Delta\tau[0, 1]$. For all the time series with different noise levels, the first delay value is zero and always maintained at zero throughout the training in accordance with Eq. (7). Hence, the lines showing the evolution of the first delay value, which is zero, overlap for all the cases. For the clean time series without noise, the second component of the delay vector approaches 0.25 s, which is similar to the delay predicted by the AMI-FNN method. As the noise level increases, the predicted value of the second component of the delay vector is lower than 0.25 s; i.e., the delay value stays closer to its initial value. Thus, we can see that the linear independence of the coordinates is not strictly enforced, but the linear model for the clean time series naturally leads to linearly independent coordinates.

The results from the MAPSR method are compared with the AMI-FNN method and the PECUZAL method in Table I. For a clean signal with the noise level of 0%, AMI-FNN predicts the dimension as 7 and delay vector with $\Delta\tau = 0.25$ s. Knowing the dynamics of the harmonic oscillator, the dimensions predicted are indeed quite high. The MAPSR and PECUZAL methods predict the dimension as 2 and estimate the same delay vector. The PECUZAL method is based on the concept of noise amplification and fails

TABLE I. Embedding dimension and delay values for the harmonic oscillator with different noise levels, estimated with different phase space reconstruction methods, i.e., AMI-FNN, MAPSR, and PECUZAL.

Case	Method	Dimension	Delay (s)
Noise level: 0%	AMI-FNN	7	[0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5]
	MAPSR	2	[0.0, 0.2488]
	PECUZAL	2	[0.0, 0.25]
Noise level: 1%	AMI-FNN	4	[0.0, 0.25, 0.5, 0.75]
	MAPSR	2	[0.0, 0.2428]
	PECUZAL	2	[0.0, 0.25]
Noise level: 2%	AMI-FNN	4	[0.0, 0.25, 0.5, 0.75]
	MAPSR	2	[0.0, 0.2235]
	PECUZAL	3	[0.0, 0.25, 0.23]
Noise level: 3%	AMI-FNN	4	[0.0, 0.25, 0.5, 0.75]
	MAPSR	2	[0.0, 0.2038]
	PECUZAL	4	[0.0, 0.25, 0.23, 0.02]
Noise level: 4%	AMI-FNN	4	[0.0, 0.25, 0.5, 0.75]
	MAPSR	2	[0.0, 0.1936]
	PECUZAL	2	[0.0, 0.25]
Noise level: 5%	AMI-FNN	4	[0.0, 0.25, 0.5, 0.75]
	MAPSR	2	[0.0, 0.1922]
	PECUZAL	3	[0.0, 0.25, 0.23]

for the 0% noise level; however, the addition of a minute level of noise solves the problem. On addition of the noise, the AMI-FNN method estimates the same $\Delta\tau = 0.25$ s and the dimension of 4 for all the cases. The MAPSR method estimates a dimension of 2 for all the noise levels, but the second component of the delay vector starts deviating from 0.25 s on increasing the noise. The PECUZAL method estimates the dimension of 2 for 1% and 4% noise level, 3 for 2% noise level, and 4 for 3% noise level. Thus, the MAPSR method estimates the same dimensions for all noise levels, whereas the AMI-FNN predicts the highest dimensions for all the situations; on the other hand, the PECUZAL method estimates a higher number of dimensions with the increase in the noise level. Here, we observe that the MAPSR has estimated the least dimensions compared to the AMI-FNN and the PECUZAL methods. Also, the estimated delay value of 0.25 s for clean data agrees with both of the methods. The delay value gets less modified as the noise level increases.

The comparison of the phase portrait reconstructed using the true input time series and the trajectories predicted using a model trained with the MAPSR method are shown in Fig. 3. For the 0% noise level, Fig. 3(a), the true and predicted trajectories overlap exactly with each other, and the attractor shape is nearly circular. For other noise levels, Figs. 3(b)–3(f), the model is able to capture the attractor of an elliptical shape.

B. Application of the MAPSR method to univariate time series from the Lorenz system

The MAPSR method is further tested with the time series obtained from the Lorenz system.⁴⁹ The x time series of the Lorenz system,

$$\begin{aligned}\dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z,\end{aligned}\quad (11)$$

is selected for the analysis, with parameters $(\rho, \sigma, \beta) = (28, 10, 8/3)$, integrated with a DOPRI5⁵⁰ solver at a step size $\Delta t = 0.01$ s. Similar to the harmonic oscillator, noise is added to the $x(t)$ time series of the Lorenz system to test the method for six noise levels, from 0% to 5%, with a step size of 1%.

For the MAPSR method, a neural ODE is used with three hidden layers and four weight matrices. Each hidden layer has 50 nodes. This configuration of neural ODE is arbitrarily chosen (other modeling techniques can be used in place of neural ODE). For non-linearity, we used the \tanh activation function. The state of the last hidden layer is linearly mapped to the output vector \vec{x} of dimension d without any nonlinearity.

We randomly choose 300 different initial conditions (i.e., batch size = 300), and the model is used for predicting the next 25 states with a time step of $\Delta t = 0.01$ s or for the time of 0.25 s (i.e., batch time = 0.25 s). The weight matrix and delay vector are trained with a learning rate of $\alpha_w = 10^{-3}$ and $\alpha_{\vec{\tau}} = 10^{-5}$ respectively. Training is performed with a fixed 20 000 iterations.

The AMI-FNN method estimates the $\Delta\tau_{AMI}$ as 0.17 s and the dimension as 3 for the Lorenz system without noise. Thus, for the MAPSR method, the delay vector is initialized with a common difference of $\Delta\tau = 0.2 \text{ s} \sim O(\tau_{AMI})$. Each time series is tested for the

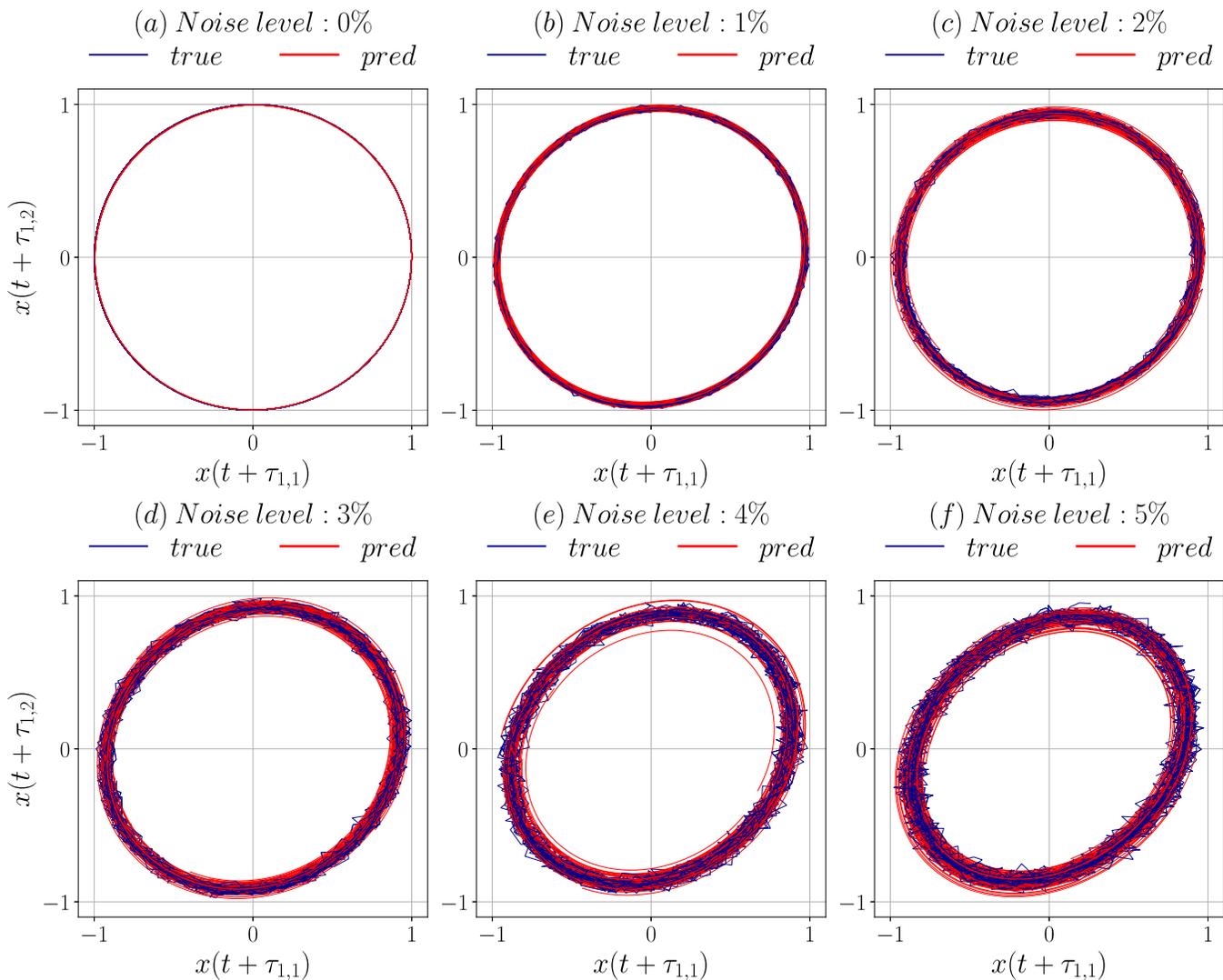


FIG. 3. Comparison of the true and predicted trajectories using the MAPSR method for six different noise levels 0%–5% for the harmonic oscillator.

initial dimension from 1 to 6. The average loss (in log scale) for the last 100 iterations for all the time series with different embedding dimensions is shown in Fig. 4(a). The configuration with minimum loss is marked with a red star and is chosen as optimal embedding. For a clean time series with 0% noise, MAPSR predicts the optimal embedding dimension as 5, whereas for all other noise levels, the embedding dimension of 3 leads to the minimum loss. The evolution of the components of the delay vector during the training for optimal configuration is shown in Fig. 4(b). Similar to the harmonic oscillator, the delay values adjust for the first few iterations and then attain a steady value except for the time series with a 1% noise level for which delays update till (approx.) 10 000 iterations, after which they become steady. The first three delay values (approx.) attain the

same value for 0% and 2% and also for 3% to 5% noise levels. The exact values for the delay are given in Table II.

The dimensions and delays estimated for time series with different noise levels are given in Table II. For clean time series, the AMI-FNN and PECUZAL predict the embedding dimension as 3, whereas the MAPSR method predicts the dimension as 5, which is indeed quite large. The first two delay values for all three methods are the same. For higher noise levels, the embedding dimension estimated with the AMI-FNN method increases, whereas PECUZAL and MAPSR methods estimate the same embedding dimension as 4 and 3, respectively. The first two delay values estimated by all the methods are very close. For the third delay value, only the estimates from AMI-FNN are closer to MAPSR. For the fourth delay value,

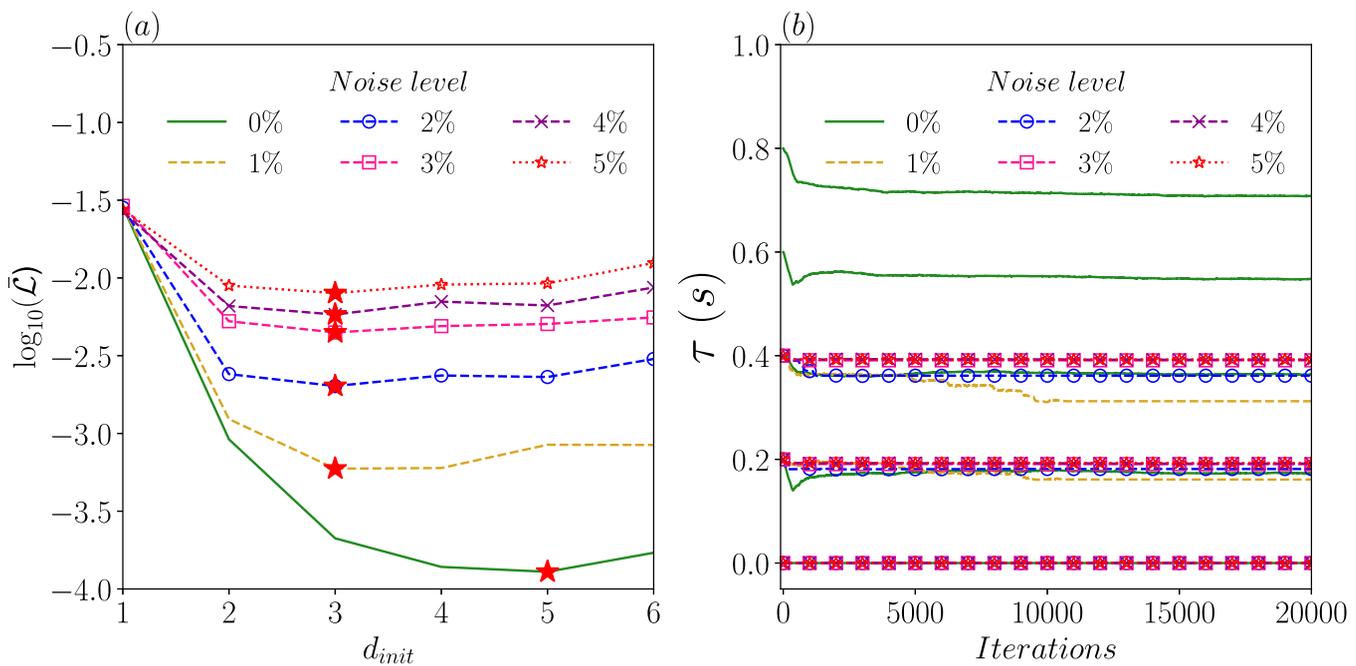


FIG. 4. Parameters of phase space reconstruction for the time series data from Lorenz system with different noise levels using MAPSR method. (a) The plot of the average loss ($\bar{\mathcal{L}}$) in log scale for different initial dimensions d_{init} . The point with minimum loss is shown with a red star for each noise level. For 0% noise, the loss is minimum for a dimension of 5. For the remaining noise levels, the method estimates the dimension as 3. (b) Shows the evolution of the delay vector for optimal dimension with training iterations for time series with different noise levels. The lines showing the evolution of the first three delay values (approximately) overlap, for the 0% and 2% and also for 3% to 5% noise levels.

there is a significant difference in the values predicted by the AMI-FNN and PECUZAL methods. The higher dimensions predicted by the AMI-FNN method can be attributed to increased false nearest neighbors (FNNs) detected due to noise. The MAPSR predicts the embedding dimension of 3, which might be due to the estimation of trajectories using neural ODE, which is smooth and reduces the effect of noise on the embedding dimension estimation. We can see here that increasing the noise level decreases the number of dimensions required for the modeling as compared to the clean data. The same has been conjectured by Ref. 51, which states that “a full formal embedding, although mandatory for detailed dynamical analysis, is not necessary for the purposes of prediction”; this especially holds for noisy data and has been discussed by Bradley and Kantz.⁵²

The three-dimensional phase portrait of the reconstructed attractor using x time series from the Lorenz attractor (*true*) is compared to the trajectories predicted using the neural ODE model (*pred*) in Fig. 5 for different noise levels. For the clean time series [Fig. 5(a)], the true and predicted time series closely overlap. For other noise levels [Figs. 5(b)–5(f)], the true trajectories are distorted due to noise, but the trajectories predicted using the MAPSR method are smooth and able to follow true noisy trajectories.

To compare the prediction horizon of the MAPSR method with the AMI-FNN and PECUZAL methods, we train the neural ODE on the trajectories reconstructed using AMI-FNN and

PECUZAL methods independently. Here, we used the same set of hyperparameters for neural ODE except for the number of input and output nodes, which are PSR method-specific. The neural ODE is used with 50 nodes in each of the 3-hidden layers. We used the clean time series of x from the Lorenz system for PSR using the AMI-FNN and PECUZAL methods. The model is trained to predict for the duration of $T_R \sim 1.4T_\lambda$, where $T_\lambda = 1/\lambda$ is the Lyapunov time scale and λ is the Lyapunov exponent. The MAPSR estimates a five-dimensional phase space (this estimated dimension is high due to large T_R). The comparison of the true time series of x obtained using Eq. (11) with the time series of x estimated using the AMI-FNN, PECUZAL, and MAPSR method is shown in Fig. 6(a). The predicted time series follows the true time series for the duration close to $2T_\lambda - 3T_\lambda$ for AMI-FNN and PECUZAL methods and $8T_\lambda - 9T_\lambda$ for the MAPSR method. The normalized cumulative deviation ($\delta_{cum}/\delta_{max}$) of the predicted time series from the true time series is shown in Fig. 6(b); here, $\delta_{cum}(t) = \sum_{0 \leq t_i \leq t} |x_{pred}(t_i) - x_{true}(t_i)|$ and $\delta_{max} = \max(x_{true}) - \min(x_{true})$. The normalized cumulative deviation for AMI-FNN and PECUZAL methods shows a similar trend where for the MAPSR method, δ_{cum} grows gradually compared to the AMI-FNN and PECUZAL method.

One can improve the predictions of the neural ODE trained using trajectories from AMI-FNN and PECUZAL methods by adjusting neural ODE configuration and hyperparameters. We find that the MAPSR method, which uses delay embedding optimized for

TABLE II. Embedding dimension and delay values estimated for the x time series of Lorenz system with noise levels from 0% to 5% in the step of 1%. The estimates from the MAPSR method are compared with the AMI-FNN and PECUZAL methods.

Case	Method	Dimension	Delay (s)
Noise level: 0%	AMI-FNN	3	[0.0, 0.17, 0.34]
	MAPSR	5	[0.0, 0.17, 0.36, 0.55, 0.71]
	PECUZAL	3	[0.0, 0.17, 0.09]
Noise level: 1%	AMI-FNN	5	[0.0, 0.17, 0.34, 0.51, 0.68]
	MAPSR	3	[0.0, 0.16, 0.31]
	PECUZAL	4	[0.0, 0.17, 0.9, 0.74]
Noise level: 2%	AMI-FNN	5	[0.0, 0.17, 0.34, 0.51, 0.68]
	MAPSR	3	[0.0, 0.18, 0.36]
	PECUZAL	4	[0.0, 0.18, 0.88, 0.71]
Noise level: 3%	AMI-FNN	6	[0.0, 0.17, 0.34, 0.51, 0.68, 0.85]
	MAPSR	3	[0.0, 0.19, 0.39]
	PECUZAL	4	[0.0, 0.18, 0.87, 0.44]
Noise level: 4%	AMI-FNN	7	[0.0, 0.19, 0.38, 0.57, 0.76, 0.95, 1.14]
	MAPSR	3	[0.0, 0.19, 0.39]
	PECUZAL	4	[0.0, 0.19, 0.85, 0.38]
Noise level: 5%	AMI-FNN	7	[0.0, 0.19, 0.38, 0.57, 0.76, 0.95, 1.14]
	MAPSR	3	[0.0, 0.19, 0.39]
	PECUZAL	4	[0.0, 0.18, 0.8, 0.98]

considered neural ODE configuration and set of hyperparameters, performs better than AMI-FNN and PECUZAL methods.

C. Application of the MAPSR method to univariate time series from a turbulent combustor

After the above two theoretical examples, we demonstrate the MAPSR method on experimental data: time series data obtained from a turbulent combustor in this section. The experiments conducted on the turbulent combustor aim to study the transitions in thermoacoustic systems.⁵³ Thermoacoustic systems involve the interaction of the heat source and the acoustic field within the confining chamber.³⁶ The positive feedback between the acoustics and the heat source can lead to high amplitude limit cycle oscillations (LCOs) and is well known as thermoacoustic instability in the field of gas turbines and rocket engines.³⁶ To study how the dynamical state of the thermoacoustic system changes with the air flow rate (control parameter), the experiments are conducted at a constant fuel (liquefied petroleum gas: butane 40% and propane 60% by volume) flow rate of 28 SLPM (standard liter per minute) and by quasi-steadily varying the air flow rate from 448 to 878 SLPM in steps of 28 SLPM. For each mass flow rate of air, 3 s long time series of pressure fluctuations p' and heat release rate \dot{q} are recorded at a sampling rate of 10 kHz. For an air flow rate near 448 SLPM, the time series of p' shows chaotic behavior,^{54,55} and near 878 SLPM shows the dynamical regime of limit cycle oscillations

(LCOs). The detailed description of the experiment is reported by Unni and Sujith.⁵³ The transition from chaos to LCO occurs via intermittency.⁵⁶ Here, we test the proposed methodology to obtain delay embedding and model the different dynamical regimes of the turbulent thermoacoustic system.

The time series data from the turbulent combustor during three different regimes, (a) chaos, (b) intermittency, and (c) LCO, is used to assess the behavior of MAPSR on real-world time series data. The AMI-FNN method predicts the dimension of the phase space as 5 and $\Delta\tau_{AMI} = 0.26 \times 10^{-2}$ s for the regime of chaos. Thus, the delay vector for the MAPSR method is initialized with a common difference of $\Delta\tau = 0.2 \times 10^{-2}$ s $\sim O(\Delta\tau_{AMI})$ with the first delay value equal to zero and d_{init} delay components. The MAPSR method compares the training loss for $d_{init} = 1$ to 12. The neural ODE used for modeling the dynamics has three hidden layers and four weight matrices. The learning rate for the weight matrices is initially maintained at 10^{-3} , which is smoothly changed to 10^{-4} while training. The delay vector is trained with a fixed learning rate of 10^{-6} . Using the delay vector and time series data from experiments, the attractor is reconstructed using linear interpolation. Randomly, 60 points are chosen on the attractor as the initial conditions, i.e., batch size = 60, and neural ODE is integrated for $0.72T$ s; i.e., batch time = $0.72T$ s, with each of these as initial conditions. Here, T s is the period of the oscillations during the regime of LCO. The loss is computed using Eq. (4) with L_1 -norm.

The loss incurred after training for dynamical regimes of chaos, intermittency, and LCO is shown in Fig. 7(a). The horizontal axis shows the initial dimensions (d_{init}), i.e., at the start of training. The dimension at which loss is minimum is shown with a star mark accompanying the value within parentheses (d_{final}), where d_{final} is the optimal embedding dimensions after the training. For all three dynamical regimes, d_{init} and d_{final} are different due to the merging of the delays as discussed in Sec. II. For the regime of chaos, the MAPSR method estimates the dimension of the delay vector as $d_{final} = 5$, which was initialized to $d_{init} = 9$ before training. Similarly, for the dynamical regimes of intermittency and LCO, the MAPSR method estimates the dimension as 4 and 2, which were initialized to 6 and 3, respectively. The development of the delay vector for the optimal cases marked by the star in Fig. 7(a) is shown in Fig. 7(b). The first delay value for all the cases is zero. The merging of the delay occurs during the first few iterations, and a zoomed view is shown in Fig. 8. For the regime of chaos, the initial delay vector is $\Delta\tau[0, 1, 2, 3, 4, 5, 6, 7, 8]$. During training, the first delay component stays zero. The second–third, fourth–fifth, sixth–seventh, and eighth–ninth components merge to give a five-dimensional delay vector. For the regime of intermittency, the initial delay vector is $\Delta\tau[0, 1, 2, 3, 4, 5]$ whose first-second and third-fourth components merge and give four-dimensional phase space. Similarly, for the regime of LCO, the initial delay vector is $\Delta\tau[0, 1, 2]$, whose first and second components merge and give a two-dimensional phase space.

The trajectories reconstructed using the true time series of p' from the turbulent combustor are compared with the predicted time series using the MAPSR method with the neural ODE model in Fig. 9. For the dynamical regimes of chaos, intermittency, and LCO, the predicted trajectories closely follow the true trajectories reconstructed using true time series of p' . For the dynamical regimes of chaos and intermittency, Figs. 9(a) and 9(b), only the first

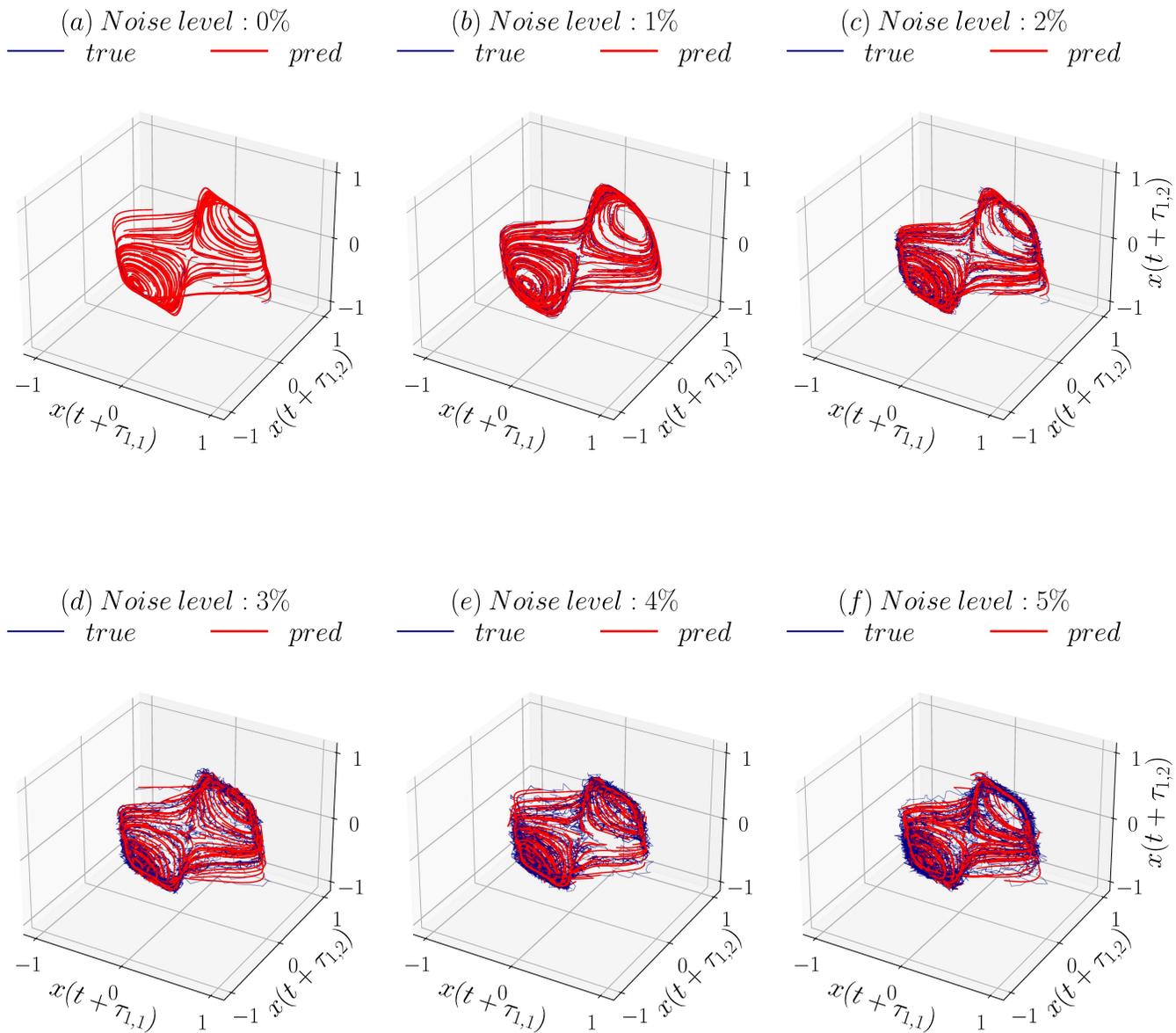


FIG. 5. Comparison of the true trajectories reconstructed from the x time series of the Lorenz system with the predicted trajectories using the MAPSR method for six different noise levels 0%–5% with neural ODE as the model.

three delay coordinates are used for visualization of the attractor in three-dimensional space. The delay embedding for the dynamical regime of LCO is estimated to be two-dimensional using the MAPSR method [Fig. 9(c)].

The delay vector estimated with AMI-FNN, MAPSR, and PECUZAL methods for the data obtained from a turbulent combustor is given in Table III. For the dynamical regime of chaos, all the methods estimate the embedding dimension as 5, but the estimated delay values are different. For the dynamical regime of intermittency, the MAPSR method estimates the least dimension

as 4, whereas AMI-FNN and PECUZAL estimate the dimension as 5 and 6, respectively. Also, for the regime of intermittency, the estimates of the first two delay values are approximately the same, i.e., $[0.0, 0.24] \times 10^{-2}$ s for all the methods. For the dynamical regime, LCO, AMI-FNN, and PECUZAL methods estimate the same embedding dimension of 5, whereas the MAPSR method estimates two dimensional delay embedding, which is expected for LCO. Here, we can see that for all dynamical regimes, the dimension estimated by the MAPSR is less than or equal to that predicted using other methods.

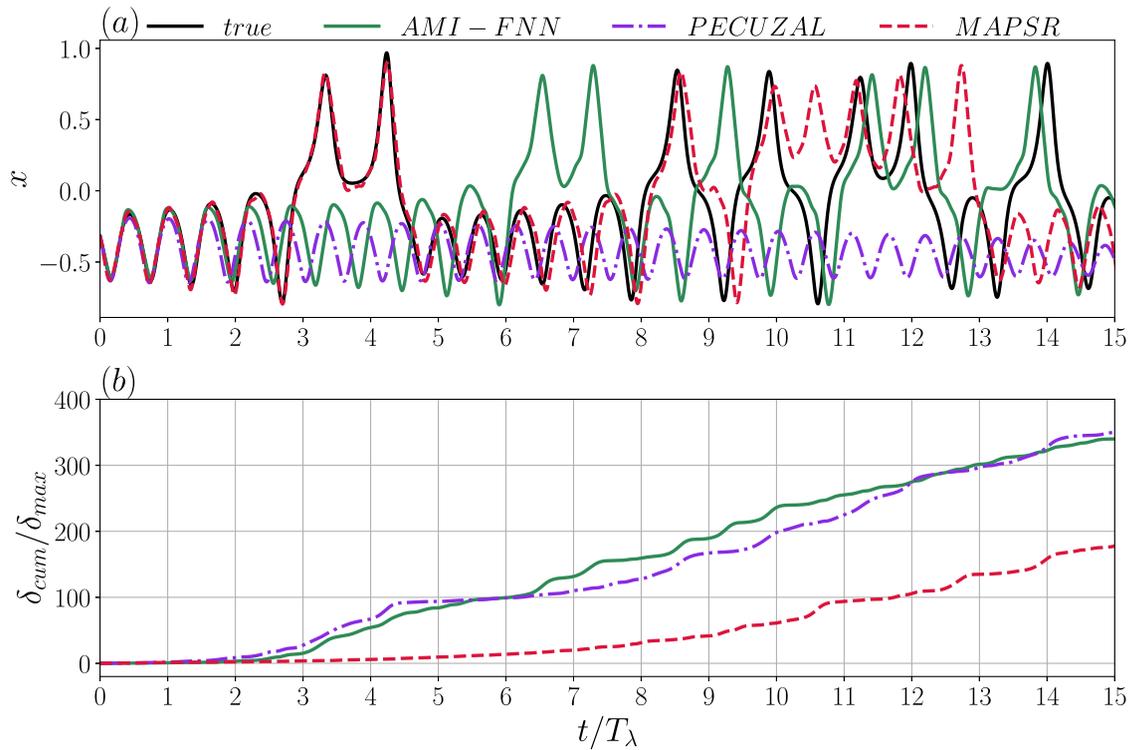


FIG. 6. (a) Comparison of the true x time series of the Lorenz system [Eq. (11)] with the time series predicted using the neural ODE trained on the trajectories reconstructed using AMI-FNN, PECUZAL, and MAPSR methods. The predicted time series follows the true time series close to $2T_\lambda - 3T_\lambda$ for the AMI-FNN and PECUZAL method and $8T_\lambda - 9T_\lambda$ for the MAPSR method. (b) The evolution of the cumulative normalized deviation ($\delta_{cum}/\delta_{max}$) with normalized time (t/T_λ).

Figure 10 compares the prediction results for the AMI-FNN, MAPSR, and PECUZAL methods. The evolution of the normalized average cumulative deviation ($\bar{\delta}_{cum}/\delta_{max}$) with normalized time t/T_λ for the dynamical regimes of chaos and intermittency is, respectively, shown in Figs. 10(a) and 10(b). Here, $\bar{\delta}_{cum}(t)$ is the cumulative deviation up to time t averaged over the batch, and T_λ is the Lyapunov time scale for the time series of p' . For the dynamical regime of chaos, ($\bar{\delta}_{cum}/\delta_{max}$) grows similarly for all the methods for nearly $7T_\lambda$, after which ($\bar{\delta}_{cum}/\delta_{max}$) grows faster for the MAPSR method and slowest for the PECUZAL method. After $16T_\lambda$, the deviation for the PECUZAL method grows rapidly beyond other methods. For the dynamical regime of intermittency, ($\bar{\delta}_{cum}/\delta_{max}$) is the same for all the methods up to $5T_\lambda$. Furthermore, the deviation grows fastest for the PECUZAL and slowest for AMI-FNN. The deviation for MAPSR stays closer to AMI-FNN and drops below AMI-FNN after $10T_\lambda$.

D. Application of the MAPSR method to multivariate time series from the turbulent combustor

The MAPSR is also tested with multivariate time series data $\vec{s} = [s_1, s_2] = [\dot{q}', p']$ from the turbulent combustor (\dot{q}' is the heat release rate fluctuation, which is mean subtracted \dot{q}). The delay vector is initialized with $2d_{init}$ components with d_{init} components for each time series. Similar to the univariate case, for both the

time series, delay values are initialized with a common difference of $\Delta\tau = 0.2 \times 10^{-2}$ s with the first delay value as zero; e.g., for $d_{init} = 2$, $\vec{\tau} = \Delta\tau [0, 1, 0, 1] = [\tau_{1,1}, \tau_{1,2}, \tau_{2,1}, \tau_{2,2}]$. For the application of the MAPSR method to multivariate time series, the same configuration of neural ODE and the learning rates used are the same as that of the univariate case.

The variation of the loss with initial dimension d_{init} is shown in Fig. 11(a). For the three dynamical regimes of chaos, intermittency, and LCO, the loss is minimum for initial dimension d_{init} of 2, 2, and 5 for each time series \dot{q}' and p' ; i.e., the net initial dimension D_{init} of the delay vector is 4, 4, and 10 for these three dynamical regimes, respectively. In Fig. 11(a), next to this minima (red star), the parentheses show the final dimension estimated using the MAPSR method; i.e., $(d_{1,final} + d_{2,final} = D_{final})$. Thus, the MAPSR method, respectively, estimates D_{final} of 9, 4, and 2 for the dynamical regime of chaos, intermittency, and LCO, respectively.

The evolution of delay components for the time series of \dot{q}' and p' is shown in Figs. 11(b) and 11(c), respectively. For the dynamical regime of chaos, Fig. 11(b) shows that initially, there are five components, and there is no merging of delay values for $s_1 = \dot{q}'$ time series, whereas first-second delay values merge in the case of p' time series as shown in Fig. 11(c). Hence, the final dimension of delay embedding for the regime of chaos is $D_{final} = 5 + 4 = 9$. For the dynamical regime of intermittency, Fig. 11(b) shows that two

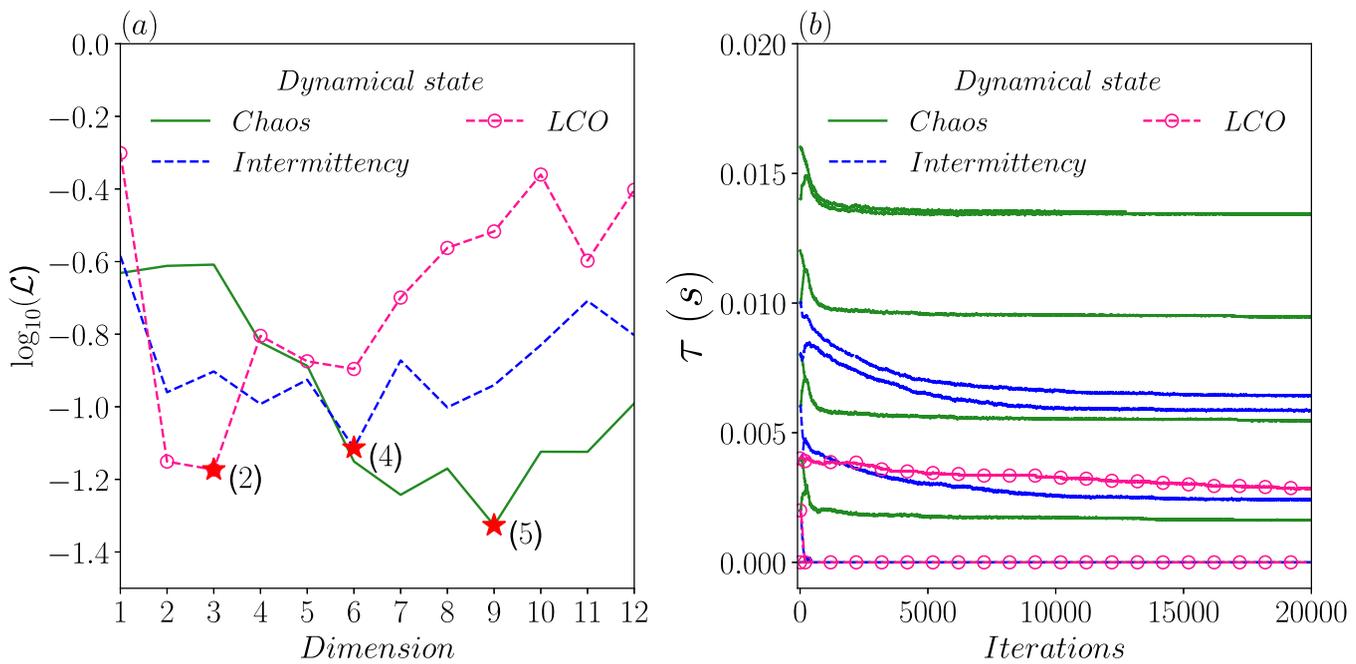


FIG. 7. Application of the MAPSR method to the time series of p' from the turbulent combustor. (a) The variation of the average loss (\mathcal{L}) for different initial dimensions d_{init} in a \log scale. The point with minimum loss is shown with a red star for each dynamical regime. The parentheses next to a red star show the dimensions (d_{final}) estimated using the MAPSR method. For the time series of chaos, the training starts with three-dimensional phase space, which shrinks to two-dimensional space during training. For the time series data with intermittency and LCO, the initial dimension of 6 and 9 shrinks to 4 and 5, respectively. (b) Shows the evolution of the delay vector with training iterations. The delays that have moved closer get merged and can be observed for the first few iterations (The enlarged view is shown in Fig. 8). Furthermore, the delay values converge with iterations and stay nearly constant.

components of the delay vector come closer but do not merge for \dot{q}' time series. Figure 11(c) shows that for the time series of p' , the two delay components adjust initially but stay separated. Hence, the final dimension estimated for the regime of intermittency is $D_{final} = 2 + 2 = 4$. For the dynamical regime of LCO, Fig. 11(b) shows that two components of the delay vector associated with the time series of \dot{q}' merge and similar behavior is observed in Fig. 11(c). Thus, the resultant dimension of the estimated phase space is $D_{final} = 1 + 1 = 2$ for the regime of LCO.

The comparison of delay values estimated using MAPSR and PECUZAL methods for multivariate time series from a turbulent combustor is shown in Table IV. The delay values estimated for different dynamical regimes are shown separately for each of the time series of \dot{q}' and p' . For the dynamical regime of chaos, the MAPSR method estimates the embedding dimension as 9, where five delay coordinates are from the time series of \dot{q}' and four coordinates are from the time series of p' . On the other hand, for the regime of chaos, the PECUZAL method estimates the embedding dimension of 5 as that of the univariate case with a single delay coordinate from \dot{q}' time series and four delay coordinates from time series of p' . For the regime of intermittency, the MAPSR estimates the four-dimensional delay embedding with two delay coordinates from \dot{q}' and p' time series individually. For the same dynamical regime, the PECUZAL method estimates six-dimensional delay embedding with

no delay coordinate from the time series of \dot{q}' , and all the delay values estimated for p' time series are the same as that of the univariate case. For the regime of LCO, the MAPSR method estimates the delay embedding of dimension 2 with a single delay coordinate from each time series of \dot{q}' and p' , whereas the PECUZAL method estimates five dimensional phase space with no delay coordinate from \dot{q}' time series and the delay values for p' time series are the same as that of the univariate case.

The discrepancy in the estimated delay coordinates can be understood based on the idea behind the MAPSR and PECUZAL method. The PECUZAL method is based on noise amplification, whereas the MAPSR method optimizes the delay embedding for modeling. In the case of multivariate time series data, the PECUZAL method might not include the time series causing noise amplification. This behavior of the PECUZAL method can be observed for the dynamical regimes of intermittency and LCO where there is no delay coordinate from the time series of \dot{q}' . On the other side, the MAPSR method optimizes the loss function that quantifies the prediction error. Here, the loss function gives equal weightage to the prediction of all the delay coordinates. Hence, none of the time series is dropped out as the modeling is being performed to improve the prediction for input time series. Thus, one should apply the MAPSR method to those time series for which modeling is intended.

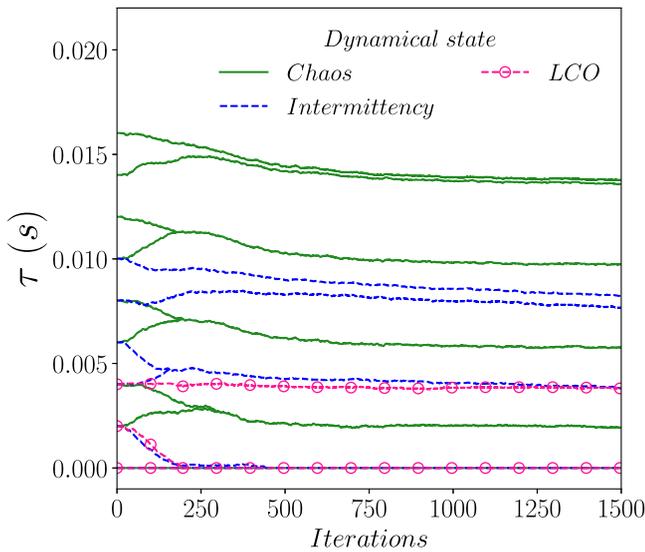


FIG. 8. Variation of the delay vector during the training with experimental data from different dynamical regimes. The plot shows an enlarged view of Fig. 7(b) for the first few iterations. The delay values merge if they are closer than a threshold $\tau_{th} = \Delta t/2$. For the chaos (green lines), the delay vector of initial length 9 shrinks to 5 during the training. The delay vector with an initial length of 6 and 3 shrinks to 4 and 2 for the regime of intermittency (blue dashed lines) and LCO (magenta dashed lines with circle marker), respectively.

The comparison of the true trajectories constructed using time series of (\dot{q}, p') with the trajectories predicted using the MAPSR method with neural ODE model is shown in Fig. 12. For the dynamical regimes of chaos and intermittency, Figs. 12(a) and 12(b), the attractor is visualized in three-dimensional space with one delay component from the time series of \dot{q} and two other components

TABLE III. Embedding dimension and delay values for three dynamical regimes of the turbulent combustor (a) chaos, (b) intermittency, and (c) LCO, estimated with different phase space reconstruction methods, i.e., AMI-FNN, MAPSR, and PECUZAL.

Case	Method	Dimension	Delay ($\times 10^{-2}$) s
Chaos	AMI-FNN	5	[0.0, 0.26, 0.52, 0.78, 1.04]
	MAPSR	5	[0.0, 0.1623, 0.5448, 0.9474, 1.3439]
	PECUZAL	5	[0.0, 0.29, 0.14, 0.22, 0.07]
Intermittency	AMI-FNN	5	[0.0, 0.24, 0.48, 0.72, 0.96]
	MAPSR	4	[0.0, 0.2425, 0.5831, 0.6434]
	PECUZAL	6	[0.0, 0.24, 0.12, 0.18, 0.48, 0.35]
LCO	AMI-FNN	5	[0.0, 0.2, 0.4, 0.6, 0.8]
	MAPSR	2	[0.0, 0.2845]
	PECUZAL	5	[0.0, 0.21, 0.11, 0.33, 0.27]

from the time series of p' . Figure 12(c) shows the two-dimensional phase portrait for the dynamical regime of LCO, with one delay coordinate from the time series of \dot{q} and another from the time series of p' . We can see in Fig. 12 that the predicted trajectories are able to follow the true trajectories.

The comparison of the MAPSR and PECUZAL method using the evolution of the normalized average deviation $\bar{\delta}_{cum}/\delta_{max}$ for the time series of \dot{q} and p' with normalized time t/T_{max} for the dynamical regimes of chaos and intermittency is, respectively, shown in Figs. 13(a) and 13(b). δ_{max} is computed separately for each time series, and T_{λ} is computed using the time series of p' . The neural ODE with the optimal configuration obtained using the MAPSR method is used to predict for nearly $25T_{\lambda}$ duration for the dynamical regimes of chaos and intermittency. The deviation for the dynamical regime of chaos for the MAPSR method initially grows slower compared to the PECUZAL method for both \dot{q} and p' . For the

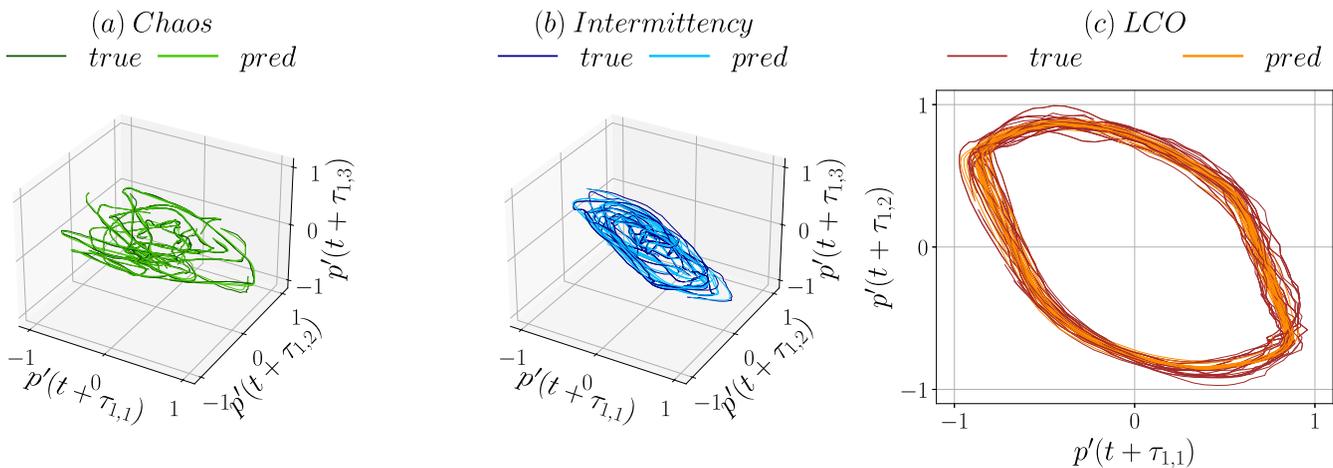


FIG. 9. Comparison of the true trajectories reconstructed from the univariate time series of p' with the trajectories predicted using the MAPSR method using neural ODE as a model.

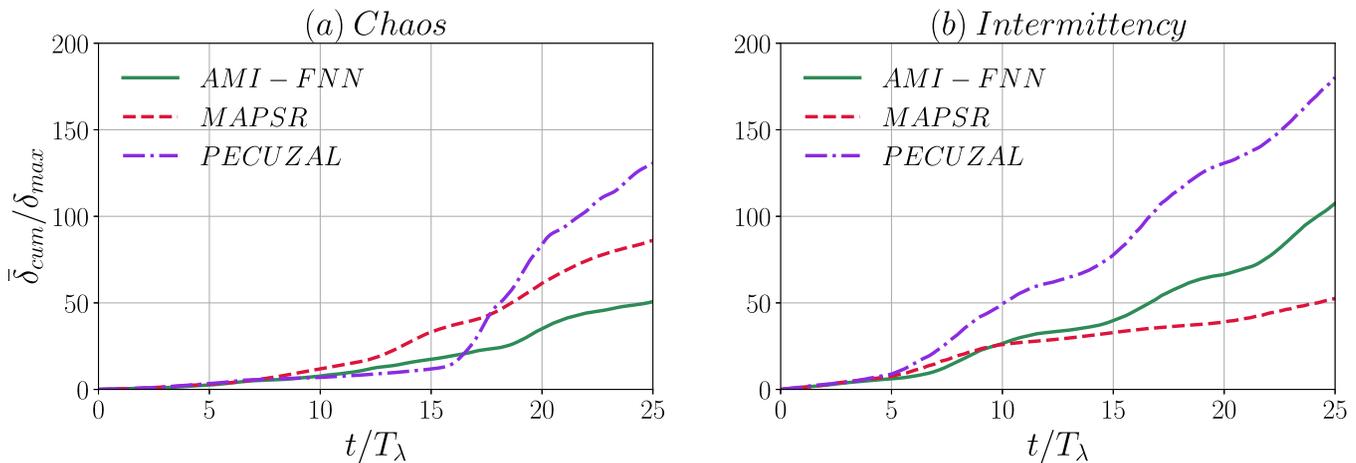


FIG. 10. Comparison of the evolution of the normalized average cumulative deviation ($\bar{\delta}_{cum}/\delta_{max}$) of the predicted time series of p' for AMI-FNN, MAPSR, and PECUZAL methods with normalized time (t/T_λ) for the dynamical regime of (a) chaos and (b) intermittency.

regime of intermittency, the deviation for the MAPSR method grows faster compared to the PECUZAL method where the PECUZAL method only considers time series of p' from input time series of q' and p' . Comparing the time scales over which the deviation grows

for multivariate time series data (Fig. 13) with the univariate time series (Fig. 10), we can see that for the univariate case, the deviation stays lower for a longer duration compared to the multivariate case. This might be due to the noise amplification caused by the inclusion

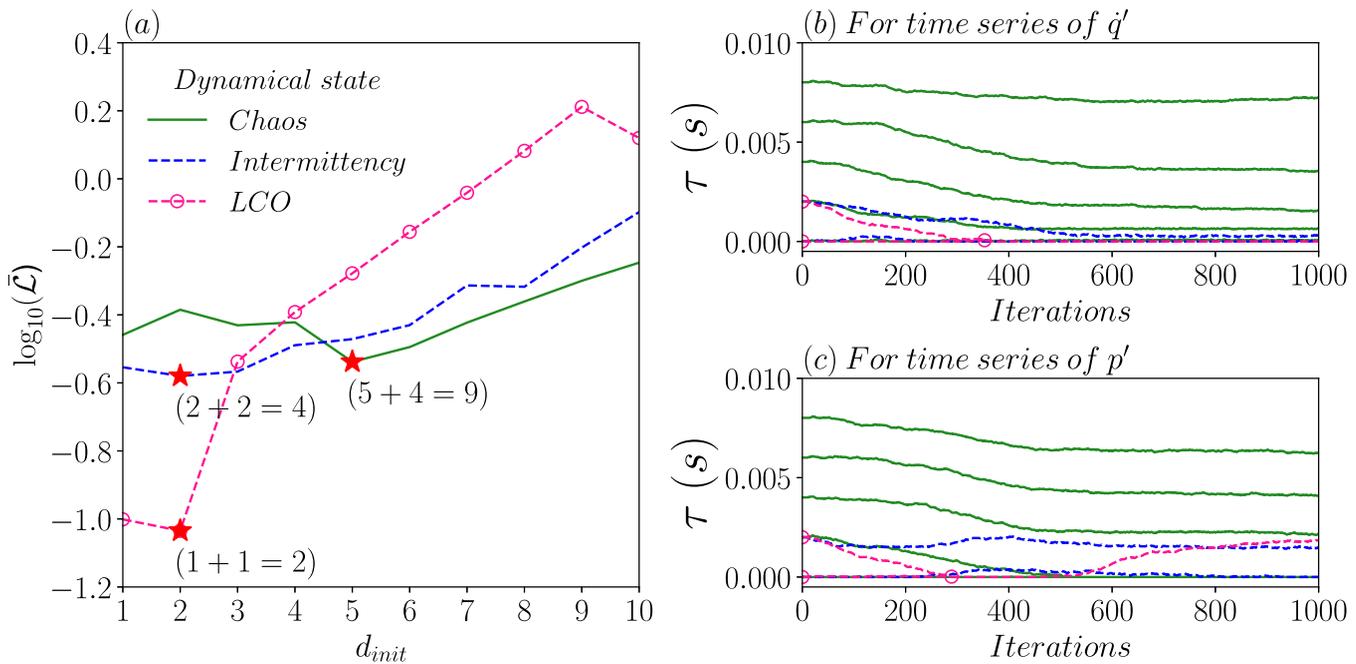


FIG. 11. Variation of loss and delay vectors for multivariate time series (q' , p') from a turbulent combustor. (a) The horizontal axis shows the initial dimension for each time series. Here, there are two time series; hence, the initial dimension (d_{init}) of 4 on the horizontal axis means that the initial embedding dimension is 4 for each time series; hence, the net embedding dimension is $4 + 4 = 8$. The points with minimal loss are marked by a red star. For the chaotic time series (green lines), the delay vector of initial length $5 + 5 = 10$ shrinks to $5 + 4 = 9$ during the training. The delay embedding with an initial dimension of $2 + 2 = 4$ stay as it is for intermittency (blue dashed lines), whereas the initial dimension of $2 + 2 = 4$ shrinks to $1 + 1 = 2$ for the regime of LCO (magenta dashed lines with circle marker). (b) and (c) Shows the evolution and merging of the delay values for the time series of q' and p' . The same color and markers are used as Fig. (a).

23 July 2024 10:06:14

TABLE IV. Embedding dimension and delay values for three dynamical regimes of the turbulent combustor (a) chaos; (b) intermittency; and (c) LCO, using multivariate time series data $[\dot{q}', p']$, estimated with different phase space reconstruction methods; i.e., MAPSR and PECUZAL.

Case	Method	Dimension	Delay ($\times 10^{-2}$) s
Chaos	MAPSR	9	$\dot{q}' : [0.006, 0.0643, 0.1445, 0.3748, 0.7351]$ $p' : [0.0, 0.1835, 0.4128, 0.6031]$
	PECUZAL	5	$\dot{q}' : [0.0]$ $p' : [0.0, 0.28, 0.14, 0.21]$
Intermittency	MAPSR	4	$\dot{q}' : [0.0, 0.0163]$ $p' : [0.006, 0.2363]$
	PECUZAL	6	$\dot{q}' : []$ $p' : [0.0, 0.24, 0.12, 0.18, 0.48, 0.35]$
LCO	MAPSR	2	$\dot{q}' : [0.0]$ $p' : [0.188]$
	PECUZAL	5	$\dot{q}' : []$ $p' : [0.0, 0.21, 0.11, 0.33, 0.27]$

of time series of \dot{q}' , which has been discarded by the PECUZAL method for the dynamical regimes of intermittency and LCO (refer Table IV).

E. Analysis of the observed results

The plot of the loss with initial dimension d_{init} shows that loss initially decreases, attains minima at $d_{init}^{(opt)}$, and then increases. Similar behavior has also been observed by Young and Graham,⁵⁷ where the first minima of $AMI(\tau)$ were used as a common difference $\Delta\tau_{AMI}$, and the dimensions for which prediction loss is minima

was used to construct *UTDE*. The increase in the loss function for the clean data can be attributed to numerical error or the discrepancy between the underlying model of the dynamical system and the model obtained after training. For noisy signal, the reconstructed vector can be decomposed into a clean signal (\vec{x}_{clean}) and noise ($\vec{\epsilon}$), $\vec{x}_{true} = \vec{x}_{clean} + \vec{\epsilon}$. To understand the behavior of the loss function with a dimension of the phase space the loss function, Eq. (4) can be rewritten as, with adjustments,

$$\mathcal{L} = \left\langle \sum_{r=1}^R \sum_{q=1}^D [(\vec{x}_{r,true})_q - (\vec{x}_{r,clean})_q - ((\vec{x}_{r,pred})_q - (\vec{x}_{r,clean})_q)]^p \right\rangle_k \tag{12}$$

Here, we have added and subtracted $\vec{x}_{r,clean}$ where clean indicates that it is a time series without any noise, and the model is expected to capture this time series. For $p = 2$,

$$\mathcal{L} = \left\langle \sum_{r=1}^R \sum_{q=1}^D \delta_{r,q}^2 \right\rangle_k + \left\langle \sum_{r=1}^R \sum_{q=1}^D \epsilon_{r,q}^2 \right\rangle_k - \left\langle \sum_{r=1}^R \sum_{q=1}^D 2\delta_{r,q}\epsilon_{r,q} \right\rangle_k \tag{13}$$

Here, $\epsilon_{r,q}$ is the q th component of $\vec{\epsilon}_r = \vec{x}_{r,true} - \vec{x}_{r,clean}$, which is noise present in the data. Similarly, $\delta_{r,q}$ is the q th component of $\vec{\delta}_r = \vec{x}_{r,pred} - \vec{x}_{r,clean}$ which measures the deviation of the model prediction from the clean state,

$$\mathcal{L} = \Delta_{model} + \Delta_{noise} - \Delta_{comb}. \tag{14}$$

Equation (14) shows the terms that contribute to the loss \mathcal{L} , which uses a l^2 -norm. Δ_{model} accounts for deviation of model prediction from a clean or an expected trajectory for each delay coordinate, Δ_{noise} accounts for the noise present in each delay coordinate, and Δ_{comb} contributes negatively to the loss function, and if $\delta_{r,q}$ and $\epsilon_{r,q}$ have the same sign, it contributes to reduce \mathcal{L} and vice versa. Figure 14 shows the effect of dimension on the terms that contribute to the loss. For the harmonic oscillator [see Fig. 14(a)], Δ_{model}

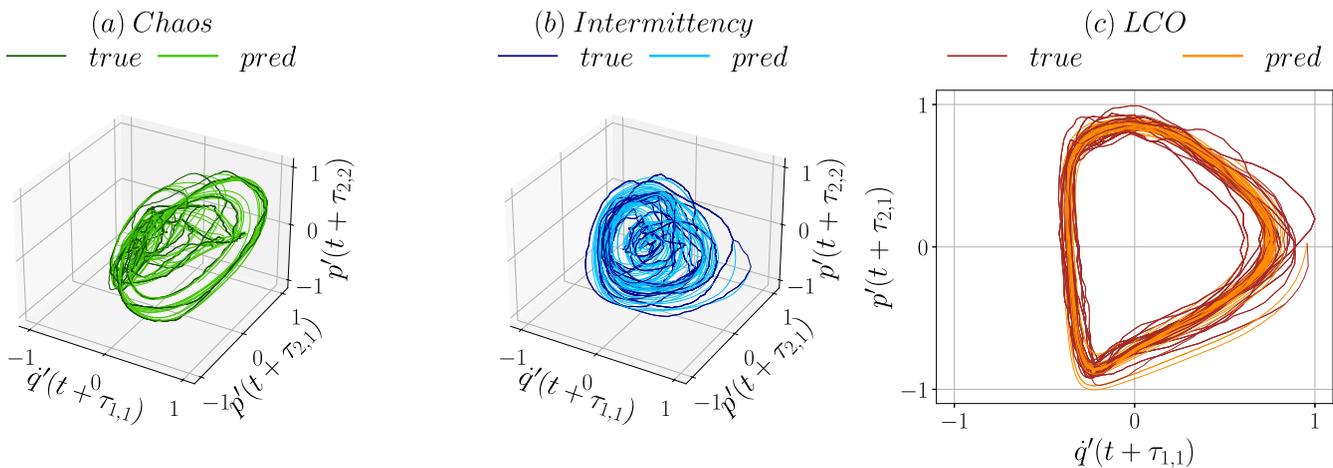


FIG. 12. Comparison of the true trajectories reconstructed from the multivariate time series of (\dot{q}', p') , with the trajectories predicted using MAPSR method using neural ODE as a model.

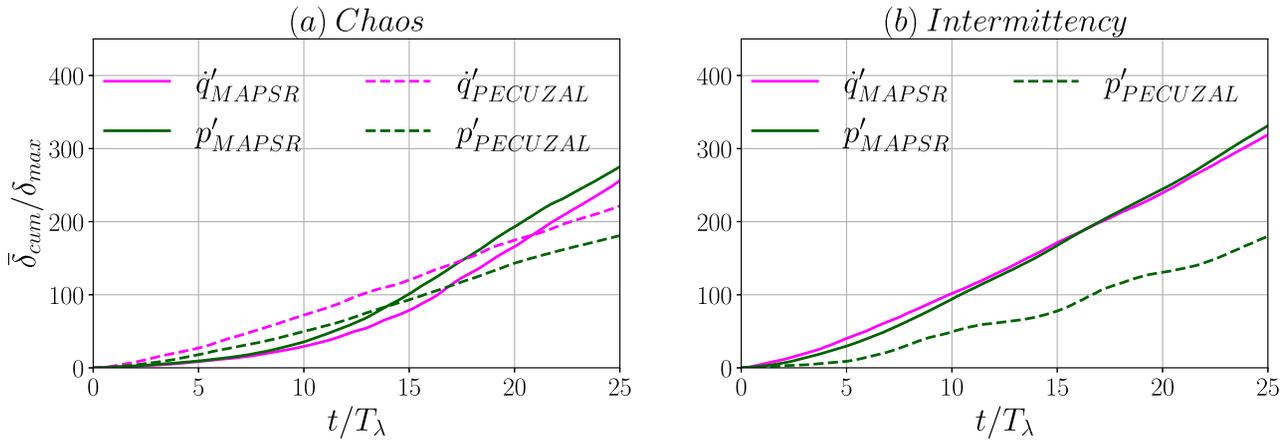


FIG. 13. Evolution of the normalized average cumulative deviation ($\bar{\delta}_{cum}/\delta_{max}$) of the predicted time series of q' and p' for MAPSR and PECUZAL methods with normalized time (t/T_λ) for the dynamical regimes of (a) chaos and (b) intermittency. Here, T_λ is obtained using p' time series, and δ_{max} is obtained separately for individual time series. The PECUZAL method rejects q' time series, hence not shown for the case of intermittency.

decreases initially with d_{init} until $d_{init} = 2$ and increases further. The loss function gets contribution from each delay coordinate and the rise in Δ_{model} can be due to different factors, such as noisy initial conditions, and an imperfect model. The term Δ_{noise} increases with dimension; it shows the contribution of noise present in the data. Δ_{comb} carries the effect of deviation of model and noise; hence, it initially decreases with d_{init} as $\delta_{r,q}$ decreases with the improved model and the phase space reconstruction. Furthermore, an increase in Δ_{comb} can be partly attributed to overfitting that is $\delta_{r,q}$ and $\epsilon_{r,q}$ have a positive correlation indicating that the model has started capturing

the noisy behavior. The resultant loss function initially reduces as model prediction accuracy increases, i.e., Δ_{model} decreases and rises after an optimal dimension due to contribution from different terms as discussed before. The terms contributing to the \mathcal{L} show similar behavior for the Lorenz system [Fig. 14(a)]. Here, loss attains minima for d_{init} of four, whereas in Fig. 4(a), the minima of loss occur at d_{init} of three; this discrepancy is due to considering the average loss for the last few iterations (100 iterations) \mathcal{L} . The average loss is used to reduce fluctuations in loss during the training. To summarize, the loss function is the collective effect of how closely the

23 July 2024 10:06:14

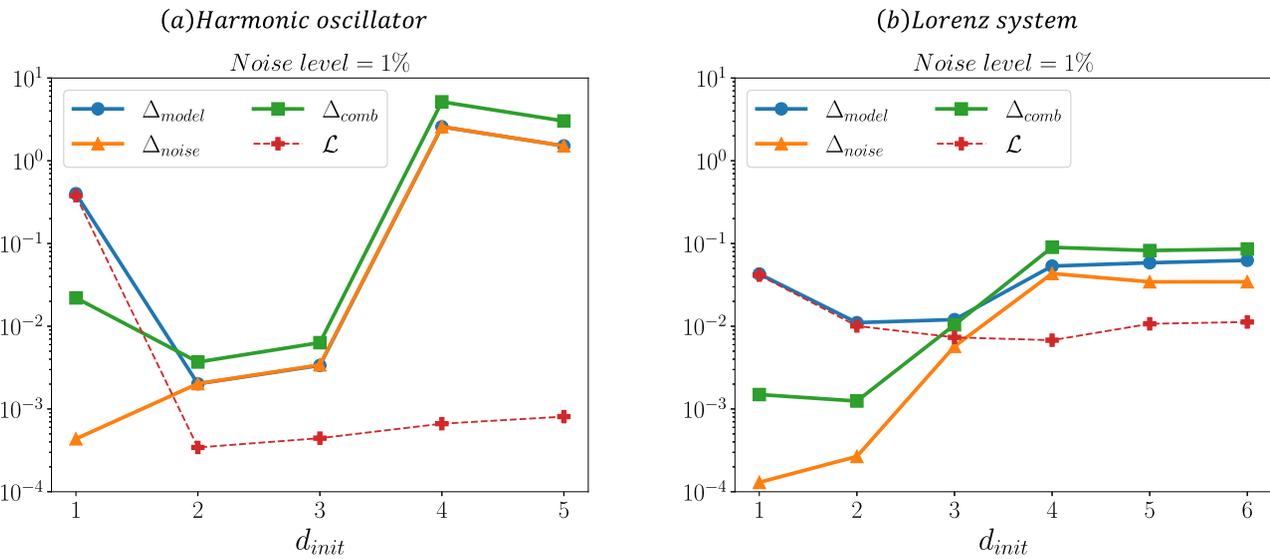


FIG. 14. Variation of different terms contributing to the loss function $\mathcal{L} = \Delta_{model} + \Delta_{noise} - \Delta_{comb}$. For a harmonic oscillator, the minima of loss occur for $d_{init} = 2$ and $d_{init} = 4$ for the Lorenz system. In the paper, the dimension of the Lorenz system for the time series with a 1% noise level is estimated as $d_{init} = 3$. This discrepancy is due to the use of \mathcal{L} .

model approximates the expected clean trajectory, noise present in the data, and overfitting. Δ_{model} decreases with an improvement in model accuracy and phase space reconstruction, attains minima, and increases further. Δ_{noise} increases with the dimension. Δ_{comb} decreases initially as the model improves and further increases. The loss function attains minima due to the combined effect of these terms.

IV. CONCLUSION

The proposed MAPSR method combines phase space reconstruction with reduced-order modeling. It is a differentiable version of a time-delay embedding that can be jointly optimized with data-driven models, such as neural ODEs. The minimization of the loss function with respect to model parameters and delay vector provides a model for the dynamical system as well as optimizes the delay embedding. The delay values can take values that are non-integer multiples of the sampling time as opposed to the existing methods that can take only integer multiples of the sampling time. For all the univariate cases that we tested, MAPSR has predicted the least embedding dimension except for the clean time series from the Lorenz system. From the Lorenz system, we can see that the MAPSR estimates the expected embedding dimension as 3 with the addition of noise. Though the addition of noise is not a requirement, the MAPSR method estimates smaller dimensions for time series with slight noise, which is mostly the case with real-world time series data. With the application of the MAPSR to different dynamical regimes of the turbulent combustor, we demonstrated the generalizability of the method to real-world time series data. For multivariate time series from the same turbulent combustor, the MAPSR method predicted the same number of dimensions as that of univariate time series data, except for the regime of chaos. With the objective of modeling, the dynamics, which is equivalent to capturing the trajectories reconstructed using input time series, the MAPSR method optimizes the delay values for all input time series, whereas the PECUZAL method can drop some of the time series to reduce noise amplification. The neural ODE model trained using the MAPSR method is able to predict the true signal for nearly seven to eight Lyapunov time scales for the Lorenz system, which is much better compared to the AMI-FNN and PECUZAL method for the same set of hyperparameters. For the univariate time series from the turbulent combustor, the average cumulative deviation initially grows faster for the MAPSR method but then stays in between the PECUZAL and AMI-FNN methods. However, as for the dynamical regime of intermittency, MAPSR performs best. For the multivariate time series from the turbulent combustor, the average cumulative deviation for the MAPSR method is lower than PECUZAL, whereas the PECUZAL method estimates the delay embedding with a single time series and performs better than the MAPSR method for the intermittency regime.

Here, we solely presented the combination of MAPSR with neural ODEs. However, it is a flexible approach that could be combined with other machine learning methods as well. The differentiability of MAPSR will result in an optimal phase space reconstruction in each of these cases. As such, we presented a strong building block for a data-driven approximation of dynamical systems. As we can inspect the learned PSR, this can also give further insights into the learned dynamics of the observed data.

ACKNOWLEDGMENTS

J. Dhadphale is indebted to the Ministry of Human Resource Development for providing the fellowship under The Prime Minister's Research Fellows (PMRF) scheme and the International Immersion Experience Award, Office of Global Engagement, IIT Madras. R. I. Sujith acknowledges the funding from the IOE initiative (No. SB/2021/0845/AE/MHRD/002696), IIT Madras, India. M. Gelbrecht acknowledges funding from the Volkswagen Foundation.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Jayesh M. Dhadphale: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (lead). **K. Hauke Kraemer:** Conceptualization (supporting); Investigation (supporting); Methodology (supporting); Writing – original draft (equal); Writing – review & editing (equal). **Maximilian Gelbrecht:** Conceptualization (supporting); Investigation (supporting); Methodology (supporting); Writing – original draft (equal); Writing – review & editing (equal). **Jürgen Kurths:** Conceptualization (supporting); Methodology (supporting); Resources (supporting); Visualization (supporting); Writing – original draft (supporting); Writing – review & editing (supporting). **Norbert Marwan:** Conceptualization (supporting); Project administration (equal); Resources (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **R. I. Sujith:** Conceptualization (supporting); Funding acquisition (lead); Project administration (lead); Resources (lead); Supervision (lead); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. The code for the MAPSR method is publicly available at <https://github.com/JayeshMD/MAPSR.git>

APPENDIX A: GENERALIZED DIMENSIONS

The box-counting dimension corresponds to $q = 0$ for a spectrum of dimensions D_q .⁵⁸ To compute D_q , the attractor in \mathbb{R}^n is divided into K boxes using the mesh size ϵ . The natural measure P_i (probability measure) is defined for each box, $i = 1, \dots, K$. D_q is calculated as

$$D_q = \lim_{\epsilon \rightarrow 0} \frac{1}{1-q} \frac{\log \sum_{i=1}^K P_i^q}{\log 1/\epsilon}. \quad (\text{A1})$$

The commonly used (D_q)'s are D_0 , D_1 , and D_2 , which are known as box-counting- (capacity-), information-, and correlation dimension, respectively. The correlation dimension can be found using the

methods described by Ding *et al.*,⁵⁹ Ding *et al.*,⁶⁰ and Grassberger and Procaccia.⁶¹ When the correlation dimension is computed for an increasing dimension m of delay embeddings, it attains the plateau for high enough m . Ding *et al.*^{59,60} rigorously showed that the onset of this plateau begins when m exceeds D_2 , i.e., if the used time series is long enough. Ding *et al.*^{59,60} also studied the effect of short and noisy data on the estimation of correlation dimension and showed that there are cases where numerically estimated correlation dimension agrees closely with the underlying attractor despite the time series being short and noisy.

APPENDIX B: COMPUTATION OF DERIVATIVES FOR TRAINING

The dynamical state predicted using the model and initial condition $\vec{x}_{0,true}$ at time $t_0 + r\Delta t$ is

$$\vec{x}_{r,pred} = \vec{x}_{0,true}(\vec{\tau}) + \int_{t_0}^{t_0+r\Delta t} f(\vec{x}_{pred}(t, \vec{\tau}), W) dt. \quad (B1)$$

For optimizing the loss with respect to parameters, the derivative of the loss function is computed with respect to the components of $\vec{\tau}$ and W . The loss function is defined as

$$\text{Loss} = \mathcal{L}(\vec{\tau}, W, R, d_{init}) = \left\langle \sum_{r=1}^R \|\vec{x}_{r,true} - \vec{x}_{r,pred}\|_p^p \right\rangle_k; \quad (B2)$$

differentiating \mathcal{L} with τ_i gives

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = \left\langle \frac{\partial}{\partial \tau_i} \sum_{r=1}^R \|\vec{x}_{r,true} - \vec{x}_{r,pred}\|_p^p \right\rangle_k, \quad (B3)$$

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = \left\langle \frac{\partial}{\partial \tau_i} \sum_{r=1}^R \sum_{q=1}^D [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^p \right\rangle_k, \quad (B4)$$

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = \left\langle \sum_{r=1}^R \sum_{q=1}^D p [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^{p-1} \frac{\partial [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]}{\partial \tau_i} \right\rangle_k, \quad (B5)$$

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = \left\langle \sum_{r=1}^R \left[p [(\vec{x}_{r,true})_i - (\vec{x}_{r,pred})_i]^{p-1} \frac{\partial (\vec{x}_{r,true})_i}{\partial \tau_i} - \sum_{q=1}^D p [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^{p-1} \frac{\partial (\vec{x}_{r,pred})_q}{\partial \tau_i} \right] \right\rangle_k, \quad (B6)$$

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = \left\langle \sum_{r=1}^R \left[p [(\vec{x}_{r,true})_i - (\vec{x}_{r,pred})_i]^{p-1} \frac{\partial (\vec{x}_{r,true})_i}{\partial \tau_i} - \sum_{q=1}^D p [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^{p-1} \frac{\partial (\vec{x}_{r,pred})_q}{\partial \tau_i} \right] \right\rangle_k, \quad (B7)$$

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = \left\langle \sum_{r=1}^R \left[p [(\vec{x}_{r,true})_i - (\vec{x}_{r,pred})_i]^{p-1} \frac{\partial (\vec{x}_{r,true})_i}{\partial \tau_i} - \sum_{q=1}^D p [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^{p-1} \frac{\partial (\vec{x}_{r,pred})_q}{\partial (\vec{x}_{0,true}(\vec{\tau}))_i} \frac{\partial (\vec{x}_{0,true}(\vec{\tau}))_i}{\partial \tau_i} \right] \right\rangle_k. \quad (B8)$$

Interpolation is used to compute $\vec{x}_{r,true} = \vec{x}_{true}(t_0 + r\Delta t, \vec{\tau})$ using the time series of $\vec{s}(t)$; thus, its derivative with τ_i depends on the interpolation scheme. For linear interpolation, $\partial (\vec{x}_{r,true})_q / \partial \tau_i = \delta_{iq} [(\vec{x}_{r+1,true})_q - (\vec{x}_{r,true})_q] / \Delta t$ where δ_{iq} is a Kronecker delta function.

The derivative of the loss function with respect to the element of weight matrix $W_{ij}^{(l)}$ of layer (l) can be computed as follows:

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \left\langle \frac{\partial}{\partial W_{ij}^{(l)}} \sum_{r=1}^R \sum_{q=1}^D [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^p \right\rangle_k, \quad (B9)$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \left\langle \sum_{r=1}^R \sum_{q=1}^D p [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^{p-1} \frac{\partial [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]}{\partial W_{ij}^{(l)}} \right\rangle_k. \quad (B10)$$

The time series of delay coordinates obtained using delay embedding $\vec{x}_{r,true}$ is independent of $W_{ij}^{(l)}$. Thus, we get

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \left\langle - \sum_{r=1}^R \sum_{q=1}^D p [(\vec{x}_{r,true})_q - (\vec{x}_{r,pred})_q]^{p-1} \frac{\partial (\vec{x}_{r,pred})_q}{\partial W_{ij}^{(l)}} \right\rangle_k. \quad (B11)$$

Here, $\partial (\vec{x}_{r,pred})_q / \partial W_{ij}^{(l)}$ and $\partial (\vec{x}_{r,pred})_q / \partial (\vec{x}_{0,true}(\vec{\tau}))_i$ from Eq. (B8) are the derivative of integration with respect to the neural ODE parameter and the initial condition. These derivatives can be computed using the backpropagation method^{45,46} or the adjoint method.³⁷

After computing $\frac{\partial \mathcal{L}}{\partial \tau_i}$ and $\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}}$, the elements of W and $\vec{\tau}$ can be updated using different optimizers. To demonstrate, the optimization using gradient descent can be written as

$$(\vec{\tau})_i := (\vec{\tau})_i - \alpha_\tau \frac{\partial \mathcal{L}}{\partial \tau_i}, \quad (B12)$$

$$W_{ij}^{(l)} := W_{ij}^{(l)} - \alpha_W \frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}}. \quad (B13)$$

Here, α_τ and α_W are learning rates for $\vec{\tau}$ and W . The results presented in this paper use the RMSprop algorithm for optimization, details of which can be found in Ref. 48.

REFERENCES

- ¹J. A. Y. Kathleen, T. Alligood, and T. D. Sauer, *Chaos: An Introduction to Dynamical Systems* (Springer, 1997).
- ²T. Mullin, "Finite-dimensional dynamics in Taylor-Couette flow," *IMA J. Appl. Math.* **46**, 109–119 (1991).
- ³F. N. Madden and T. Mullin, "An experimental observation of low-dimensional dynamics in an open channel flow," *Phys. Fluids* **7**, 2364–2374 (1995).
- ⁴H. Whitney, "Differentiable manifolds," *Ann. Math.* **37**, 645–680 (1936).
- ⁵R. Mañé, "On the dimension of the compact invariant sets of certain non-linear maps," in *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80* (Springer, Berlin, Heidelberg, 1981), pp. 230–242.
- ⁶F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80* (Springer, Berlin, Heidelberg, 1981), pp. 366–381.
- ⁷D. Broomhead and G. P. King, "Extracting qualitative dynamics from experimental data," *Physica D* **20**, 217–236 (1986).
- ⁸B. Mann, F. Khasawneh, and R. Fales, "Using information to generate derivative coordinates from noisy time series," *Commun. Nonlinear Sci. Numer. Simul.* **16**, 2999–3004 (2011).
- ⁹J. F. Gibson, J. Doyno Farmer, M. Casdagli, and S. Eubank, "An analytic approach to practical state space reconstruction," *Physica D* **57**, 1–30 (1992).
- ¹⁰N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Phys. Rev. Lett.* **45**, 712–716 (1980).
- ¹¹M. Casdagli, S. Eubank, J. Farmer, and J. Gibson, "State space reconstruction in the presence of noise," *Physica D* **51**, 52–98 (1991).
- ¹²T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *J. Stat. Phys.* **65**, 579–616 (1991).
- ¹³L. C. Uzal, G. L. Grinblat, and P. F. Verdes, "Optimal reconstruction of dynamical systems: A noise amplification approach," *Phys. Rev. E* **84**, 016223 (2011).
- ¹⁴M. T. Rosenstein, J. J. Collins, and C. J. De Luca, "Reconstruction expansion as a geometry-based framework for choosing proper delay times," *Physica D* **73**, 82–98 (1994).
- ¹⁵A. Eftekhari, H. L. Yap, M. B. Wakin, and C. J. Rozell, "Stabilizing embeddology: Geometry-preserving delay-coordinate maps," *Phys. Rev. E* **97**, 022222 (2018).
- ¹⁶M. 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).
- ¹⁷M. B. Kennel and H. D. I. Abarbanel, "False neighbors and false strands: A reliable minimum embedding dimension algorithm," *Phys. Rev. E* **66**, 026209 (2002).
- ¹⁸L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D* **110**, 43–50 (1997).
- ¹⁹R. Hegger and H. Kantz, "Improved false nearest neighbor method to detect determinism in time series data," *Phys. Rev. E* **60**, 4970–4973 (1999).
- ²⁰A. Krakovská, K. Mezeiová, and H. Budáčová, "Use of false nearest neighbours for selecting variables and embedding parameters for state space reconstruction," *J. Complex Syst.* **2015**, 1–12.
- ²¹K. H. Kraemer, G. Datsis, 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).
- ²²L. M. Pecora, T. L. Carroll, and J. F. Heagy, "Statistics for mathematical properties of maps between time series embeddings," *Phys. Rev. E* **52**, 3420–3439 (1995).
- ²³L. M. Pecora, L. Moniz, J. Nichols, and T. L. Carroll, "A unified approach to attractor reconstruction," *Chaos* **17**, 013110 (2007).
- ²⁴K. H. Kraemer, M. Gelbrecht, I. Pavithran, R. I. Sujith, and N. Marwan, "Optimal state space reconstruction via Monte Carlo decision tree search," *Nonlinear Dyn.* **108**, 1525–1545 (2022).
- ²⁵E. Tan, S. Algar, D. Corrêa, M. Small, T. Stemler, and D. Walker, "Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology," *Chaos* **33**, 032101 (2023).
- ²⁶S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, "Chaos as an intermittently forced linear system," *Nat. Commun.* **8**, 19 (2017).
- ²⁷I. Mezić, "Analysis of fluid flows via spectral properties of the Koopman operator," *Annu. Rev. Fluid Mech.* **45**, 357–378 (2013).
- ²⁸J. Bakarji, K. Champion, J. N. Kutz, and S. L. Brunton, "Discovering governing equations from partial measurements with deep delay autoencoders," *arXiv:2201.05136* (2022).
- ²⁹S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. U.S.A.* **113**, 3932–3937 (2016).
- ³⁰J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," *Phys. Rev. Lett.* **59**, 845–848 (1987).
- ³¹M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D* **35**, 335–356 (1989).
- ³²G. Sugihara and R. M. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series," *Nature* **344**, 734–741 (1990).
- ³³H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. (Cambridge University Press, 2003).
- ³⁴J. Isensee, G. Datsis, and U. Parlitz, "Predicting spatio-temporal time series using dimension reduced local states," *J. Nonlinear Sci.* **30**, 713–735 (2020).
- ³⁵J. M. Dhadphale, V. R. Unni, A. Saha, and R. I. Sujith, "Neural ODE to model and prognose thermoacoustic instability," *Chaos* **32**, 013131 (2022).
- ³⁶R. I. Sujith and S. A. Pawar, *Thermoacoustic Instability: A Complex Systems Perspective* (Springer International Publishing, 2021).
- ³⁷R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Adv. Neural Inf. Process. Syst.* **31**, 6571–6583 (2018).
- ³⁸J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Phys. Rev. Lett.* **120**, 024102 (2018).
- ³⁹M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.* **3**, 127–149 (2009).
- ⁴⁰X.-Y. Duan, X. Ying, S.-Y. Leng, J. Kurths, W. Lin, and H.-F. Ma, "Embedding theory of reservoir computing and reducing reservoir network using time delays," *Phys. Rev. Res.* **5**, L022041 (2023).
- ⁴¹Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud, "Latent ordinary differential equations for irregularly-sampled time series," in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32.
- ⁴²E. Dupont, A. Doucet, and Y. W. Teh, "Augmented neural ODEs," in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32.
- ⁴³H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. (Cambridge University Press, 2003).
- ⁴⁴H. Kurt, S. Maxwell, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.* **2**, 359–366 (1989).
- ⁴⁵F. Rosenblatt *et al.*, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Spartan Books, Washington, DC, 1962), Vol. 55.
- ⁴⁶D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature* **323**, 533–536 (1986).
- ⁴⁷A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *arXiv:1502.05767* (2018).
- ⁴⁸T. Tieleman, G. Hinton *et al.*, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSE: Neural Netw. Mach. Learn. **4**, 26–31 (2012).
- ⁴⁹E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.* **20**, 130–141 (1963).
- ⁵⁰E. Hairer, G. Wanner, and S. P. Nørsett, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd ed., Springer Series in Computational Mathematics Vol. 8 (Springer, 1993).
- ⁵¹J. Garland and E. Bradley, "Prediction in projection," *Chaos* **25**, 123108 (2015).
- ⁵²E. Bradley and H. Kantz, "Nonlinear time-series analysis revisited," *Chaos* **25**, 097610 (2015).
- ⁵³V. R. Unni and R. I. Sujith, "Multifractal characteristics of combustor dynamics close to lean blowout," *J. Fluid Mech.* **784**, 30–50 (2015).

- ⁵⁴V. Nair, G. Thampi, S. Karuppusamy, S. Gopalan, and R. I. Sujith, "Loss of chaos in combustion noise as a precursor of impending combustion instability," *Int. J. Spray Combust. Dyn.* **5**, 273–290 (2013).
- ⁵⁵J. Tony, E. A. Gopalakrishnan, E. Sreelekha, and R. I. Sujith, "Detecting deterministic nature of pressure measurements from a turbulent combustor," *Phys. Rev. E* **92**, 062902 (2015).
- ⁵⁶V. Nair, G. Thampi, and R. I. Sujith, "Intermittency route to thermoacoustic instability in turbulent combustors," *J. Fluid Mech.* **756**, 470–487 (2014).
- ⁵⁷C. D. Young and M. D. Graham, "Deep learning delay coordinate dynamics for chaotic attractors from partial observable data," *Phys. Rev. E* **107**, 034215 (2023).

- ⁵⁸H. Hentschel and I. Procaccia, "The infinite number of generalized dimensions of fractals and strange attractors," *Physica D* **8**, 435–444 (1983).
- ⁵⁹M. Ding, C. Grebogi, E. Ott, T. Sauer, and J. A. Yorke, "Estimating correlation dimension from a chaotic time series: When does plateau onset occur?," *Physica D* **69**, 404–424 (1993).
- ⁶⁰M. Ding, C. Grebogi, E. Ott, T. Sauer, and J. A. Yorke, "Plateau onset for correlation dimension: When does it occur?," *Phys. Rev. Lett.* **70**, 3872–3875 (1993).
- ⁶¹P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica D* **9**, 189–208 (1983).