




RESEARCH ARTICLE | OCTOBER 06 2023

Optimizing the combination of data-driven and model-based elements in hybrid reservoir computing

Dennis Duncan  ; Christoph R ath  



Chaos 33, 103109 (2023)

<https://doi.org/10.1063/5.0164013>



View
Online



Export
Citation

CrossMark

Articles You May Be Interested In

Geomagnetic transport in the solar wind driven nightside magnetosphere–ionosphere system

Physics of Plasmas (August 2002)

The solar-wind driven magnetosphere–ionosphere as a complex dynamical system

Physics of Plasmas (November 1999)

Chaos and the limits of predictability for the solar-wind-driven magnetosphere–ionosphere system

Physics of Plasmas (June 2001)

AIP Advances

Why Publish With Us?

-  **25 DAYS**
average time to 1st decision
-  **740+ DOWNLOADS**
average per article
-  **INCLUSIVE**
scope

[Learn More](#)

 AIP
Publishing

Optimizing the combination of data-driven and model-based elements in hybrid reservoir computing

Cite as: Chaos 33, 103109 (2023); doi: 10.1063/5.0164013
Submitted: 20 June 2023 · Accepted: 11 September 2023 ·
Published Online: 6 October 2023





View Online



Export Citation



CrossMark

Dennis Duncan^{1,a)}  and Christoph R ath^{2,b)} 

AFFILIATIONS

¹Department of Physics, Ludwig-Maximilians-Universit t, Schellingstra e 4, 80799 Munich, Germany

²Deutsches Zentrum f r Luft- und Raumfahrt (DLR), Institut f r KI Sicherheit, Wilhelm-Runge-Stra e 10, 89081 Ulm, Germany

^{a)}Electronic mail: duncdennis@gmail.com

^{b)}Author to whom correspondence should be addressed: christoph.raeth@dlr.de

ABSTRACT

Hybrid reservoir computing combines purely data-driven machine learning predictions with a physical model to improve the forecasting of complex systems. In this study, we investigate in detail the predictive capabilities of three different architectures for hybrid reservoir computing: the input hybrid (IH), output hybrid (OH), and full hybrid (FH), which combines IH and OH. By using nine different three-dimensional chaotic model systems and the high-dimensional spatiotemporal chaotic Kuramoto–Sivashinsky system, we demonstrate that all hybrid reservoir computing approaches significantly improve the prediction results, provided that the model is sufficiently accurate. For accurate models, we find that the OH and FH results are equivalent and significantly outperform the IH results, especially for smaller reservoir sizes. For totally inaccurate models, the predictive capabilities of IH and FH may decrease drastically, while the OH architecture remains as accurate as the purely data-driven results. Furthermore, OH allows for the separation of the reservoir and the model contributions to the output predictions. This enables an interpretation of the roles played by the data-driven and model-based elements in output hybrid reservoir computing, resulting in higher explainability of the prediction results. Overall, our findings suggest that the OH approach is the most favorable architecture for hybrid reservoir computing, when taking accuracy, interpretability, robustness to model error, and simplicity into account.

  2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0164013>

The forecasting of physical systems has traditionally relied on a system of equations derived to model the process to be predicted. This approach reaches its limitations when the underlying processes are nonlinear, particularly in situations where only imperfect models are at hand. On the other hand, the application of purely data-driven artificial intelligence (AI) based forecasting methods has led to great progress in forecasting complex dynamical systems. In recent studies using reservoir computing as the data-driven machine learning element, it was shown that a hybrid ansatz combining both approaches improves the prediction results. Here, we compare different ways of combining the model-based and data-driven elements within the hybrid method. We find that the prediction quality significantly depends on how the data-driven

and model-based elements are combined with each other. Our experimental findings show that the most accurate, reliable with respect to inaccuracies in the model, and interpretable results are obtained when the hybrid architecture combines the data-driven and model-based elements solely at the output layer.

I. INTRODUCTION

The application of AI-based forecasting methods has led to great progress in the prediction of complex systems.^{1,2} Among the AI methods being used so far, reservoir computing (RC) turns out

to be a highly promising approach as it combines superior prediction results with little CPU-needs for training,^{3–5} as well as its ability to be realized by physical devices leading to novel, energy-efficient unconventional computing methods.⁶ The software-based realization of RC, called an echo state network,⁷ has proven to be a valuable machine learning technique, overcoming the difficulties connected with training conventional recurrent neural networks.

The reservoir computing approach in its conventional form is purely data-driven, which is a useful approach if there is no knowledge about the governing equations producing the data. In many cases though, when machine learning is applied to the sciences, some knowledge about the underlying system is available, and its inclusion into the data-driven machine learning methods can be beneficial in improving the prediction accuracy as well as leading to greater interpretability. These combinations of machine learning and physical knowledge are attributed to the important emerging field called *theory-guided data science*, which “seeks to exploit the promise of data science without ignoring the treasure of knowledge accumulated in scientific principles.”⁸ There are several ways to combine physical knowledge-based models with machine learning techniques, with the general aim to create highly accurate, physically consistent models while being resource efficient (see Willard *et al.*⁹ for a review of different possibilities and applications). For example, a common way to include physical knowledge into machine learning is to add a physical loss to the overall loss function, in order to introduce soft constraints for physical consistency. In contrast to this physics-informed loss function, where the physical knowledge is only introduced during the training stage, another approach is to employ hybrid models, where the knowledge-based model becomes an integral part of the machine learning architecture that operates during training and inference.

One example area of science that benefits greatly from hybrid models is the field of extreme events in complex dynamical systems, which studies phenomena such as epidemics, tsunamis, droughts, and more.¹⁰ Due to the high complexity of these systems, which consist of a multitude of different interacting parts, a detailed model-based description of the full dynamics is often not feasible. Overcoming these issues is usually achieved by employing approximate reduced-order models or using novel data-driven machine learning methods. However, both approaches have significant limitations, as extreme events are often inherently nonlinear phenomena that occur only rarely within the available data. Combining reduced-order models with data-driven machine learning approaches often leads to improved predictions and generalizability.¹⁰

In the field of reservoir computing, different ways to include physical knowledge in to the RC framework have been explored. Doan *et al.*^{11,12} introduced the physics-informed echo state network (PI-ESN), which was later extended by Racca and Magri.¹³ By incorporating a physics-based loss term, the PI-ESN utilizes the time series’ true physical equations in order to fine-tune the RC output matrix W_{out} in a second training phase, following the usual RC training phase using only data. Consequently, the PI-ESN approach falls into the category of loss-based physics-ML methods. Their exemplary predictions on the Lorenz-63 system and chaotic flows related to the Navier–Stokes equation demonstrated improved robustness to noise and overall more accurate predictions compared to data-driven RC.

The object of the studies in this work is the hybrid RC approach introduced by Pathak *et al.*,¹⁴ where, in contrast to the PI-ESN, physical knowledge becomes an integral part of the RC architecture. The authors demonstrated significant improvements in forecasting the three-dimensional chaotic Lorenz-63 system and high-dimensional spatiotemporal chaotic Kuramoto–Sivashinsky system compared to the usual data-driven RC approach. The physical knowledge of the system is given as an imperfect knowledge-based model (KBM), $K(\cdot)$, which is a function of the current state of the time series $\mathbf{u}(t)$ that outputs an approximation of the next state: $K(\mathbf{u}(t)) \approx \mathbf{u}(t + \Delta t)$. In the original implementation, the KBM is inserted at two independent points within the RC architecture: first, as an additional input (which we refer to as input hybrid), and second, directly at the RC output (which we refer to as output hybrid). The simultaneous usage of input and output hybrid, as in the original implementation, in this work is referred to as full hybrid. In the subsequent works that employed the hybrid RC approach, all three combinations (input, output, and full hybrid) were utilized, yet a comparative study between the three options is still missing.

Wikner *et al.*¹⁵ implemented a parallelized hybrid reservoir computing approach, combining the output hybrid method with local states, in order to forecast large spatiotemporal chaotic systems in a scalable manner. This method is then applied in Arcomano *et al.*¹⁶ to atmospheric modeling, using a low-resolution atmospheric global circulation model as the imperfect knowledge-based model. In Wikner *et al.*,¹⁷ data assimilation is used to train the output hybrid model for the task of predicting chaotic dynamical systems. Racca and Magri¹⁸ also used the output hybrid approach alongside a normal data-driven RC approach to investigate robust optimization and validation techniques for learning chaotic dynamics. In Köster *et al.*,¹⁹ a delay-based RC was combined with a model element discovered from data, employing the output hybrid architecture. Shahi *et al.*²⁰ compared different recurrent neural network approaches, including the input hybrid RC approach, for the prediction of arrhythmic cardiac action potentials. Huhn and Magri^{21,22} used the full hybrid approach to learn and optimize ergodic averages of chaotic acoustics, employing a reduced-order model as the knowledge-based model. In most cases, the hybrid approach was shown to be superior in terms of prediction quality, compared to the usual data-driven RC approach.

In this paper, the three different hybrid RC approaches are systematically compared for the task of forecasting chaotic dynamical systems. Here, we use nine different three-dimensional chaotic systems and the high-dimensional spatiotemporal chaotic Kuramoto–Sivashinsky system as exemplary systems. As the effectiveness of the hybrid approach is strongly affected by the accuracy of the knowledge-based model, different knowledge-based models are considered, whose correspondence with the true model varies strongly. Our overall results suggest that the output hybrid architecture achieves the optimal combination of data-driven and model-based elements in hybrid reservoir computing in terms of accuracy, robustness to model error, interpretability, and simplicity.

The remainder of this paper is organized as follows. Section II introduces the forecasting methods, namely, data-driven and hybrid reservoir computing. Section III describes the implementation details, including the chaotic dynamical systems used as time series data, the different knowledge-based models, and the methods used

for model evaluation. The results are presented and discussed in Sec. IV, and the work of this study is concluded in Sec. V.

II. FORECASTING METHODS

A. Reservoir computing

This section introduces a slightly generalized RC architecture, which allows for a joint description of the usual data-driven RC approach and of the hybrid RC approach developed by Pathak *et al.*¹⁴ Depending on whether the hybrid methods are employed or not, the following functions f_{inp} and f_{out} have different forms, but the training and prediction scheme remains the same.

The state of the true time series, used as training data, will be denoted as a u_{dim} -dimensional vector $\mathbf{u}(t)$. From a bird's eye view, during the training phase, the reservoir—a high-dimensional and nonlinear dynamical system—is fed with the true time series $\mathbf{u}(t)$, one discrete time step Δt at a time. By encoding the true time series $\mathbf{u}(t)$ nonlinearly in the high-dimensional reservoir state $\mathbf{r}(t)$, the reservoir's output $\mathbf{y}_r(t)$ is fitted using ridge regression (RR) to approximate the time series next state: $\mathbf{y}_r(t) \approx \mathbf{u}(t + \Delta t)$. During the prediction phase, the reservoir's output $\mathbf{y}_r(t)$ is used as its next input, allowing the reservoir to run autonomously in a loop for an arbitrarily long number of prediction steps.

1. Architecture

In this section, the RC architecture defining the flow of quantities from the input to the output is described. The architecture relies on the usual echo state network implementation.⁷ The time series input is denoted as $\tilde{\mathbf{u}}(t)$ referring to the true time series $\mathbf{u}(t)$ during synchronization and training, as well as the previous prediction $\mathbf{y}_r(t - \Delta t)$ during the prediction phase. The training and prediction phases will be described in detail in Secs. II A 3 and II A 4.

In the first step, the time series input $\tilde{\mathbf{u}}(t)$ is transformed to the x_{dim} -dimensional reservoir input $\mathbf{x}(t)$,

$$\mathbf{x}(t) = f_{\text{inp}}(\tilde{\mathbf{u}}(t)). \quad (1)$$

In the usual data-driven RC approach, the reservoir input is simply given directly as the time series input $\mathbf{x}(t) = \tilde{\mathbf{u}}(t)$. In the input hybrid approach, as will be described in Sec. II B 2, $f_{\text{inp}}(\cdot)$ will additionally contain the knowledge-based model.

Before feeding the reservoir input into the reservoir, it is standardized to zero-mean and unit-variance using the standard-scaler function $S(\cdot)$, which is fitted during the training phase,

$$\hat{\mathbf{x}}(t) = S(\mathbf{x}(t)). \quad (2)$$

The standardized reservoir input $\hat{\mathbf{x}}(t)$ is then coupled to the so-called reservoir, which is realized as a high-dimensional recurrent neural network, whose internal reservoir state is given as a r_{dim} -dimensional vector $\mathbf{r}(t)$,

$$\mathbf{r}(t) = \tanh(\mathbf{W}_{\text{in}}\hat{\mathbf{x}}(t) + \mathbf{A}\mathbf{r}(t - \Delta t) + \mathbf{b}). \quad (3)$$

$\hat{\mathbf{x}}(t)$ is connected to the reservoir through the $r_{\text{dim}} \times x_{\text{dim}}$ input matrix \mathbf{W}_{in} and the recurrent connections between consecutive reservoir states $\mathbf{r}(t)$ are introduced through the $r_{\text{dim}} \times r_{\text{dim}}$ adjacency matrix \mathbf{A} , also known as the reservoir network. The r_{dim} -dimensional node bias vector \mathbf{b} is introduced to break the symmetry

within the reservoir in order to suppress the appearance of mirror attractors.²³ The hyperbolic tangent $\tanh(\cdot)$ is used as the node activation function, introducing the important nonlinearities by acting element-wise on each dimension of its argument. Following the fundamental idea of RC, the internal parameters \mathbf{A} , \mathbf{W}_{in} , and \mathbf{b} are static quantities initialized once before the training and then kept fixed. Their implementation details are discussed in Sec. II A 2. The purpose of Eq. (3) is to encode the time series inputs $\tilde{\mathbf{u}}(t)$, $\tilde{\mathbf{u}}(t - \Delta t)$, ... nonlinearly into the high-dimensional reservoir state $\mathbf{r}(t)$ so that the desired output can be learned by simple ridge regression.

The reservoir state $\mathbf{r}(t)$ is then transformed to the h_{dim} -dimensional ridge regression input vector $\mathbf{h}(t)$,

$$\mathbf{h}(t) = f_{\text{out}}(\mathbf{r}(t), \tilde{\mathbf{u}}(t)). \quad (4)$$

In the simplest data-driven RC approach, the RR input is simply given by the reservoir states directly $\mathbf{h}(t) = \mathbf{r}(t)$. The more general Eq. (4) allows for the inclusion of the knowledge-based model within the output hybrid approach, as will be described in Sec. II B 3.

The reservoir's output $\mathbf{y}_r(t)$ is then calculated as a simple linear function of the RR input $\mathbf{h}(t)$,

$$\mathbf{y}_r(t) = \mathbf{W}_{\text{out}}\mathbf{h}(t) + \mathbf{w}_{\text{out}}. \quad (5)$$

The output matrix \mathbf{W}_{out} and intercept \mathbf{w}_{out} are simply obtained by ridge regression, as will be described in Sec. II A 3. The reservoir's output $\mathbf{y}_r(t)$ aims to predict the next step in the time series.

2. Random initialization of \mathbf{W}_{in} , \mathbf{A} , and \mathbf{b}

Before beginning the training phase, the static quantities \mathbf{W}_{in} , \mathbf{A} , and \mathbf{b} are randomly initialized and then kept fixed.

The $r_{\text{dim}} \times x_{\text{dim}}$ input matrix \mathbf{W}_{in} connects the standardized reservoir input $\hat{\mathbf{x}}(t)$ to the reservoir states $\mathbf{r}(t)$. The elements of \mathbf{W}_{in} are populated in a sparse way,²⁴ where each reservoir node r_i is connected to only one randomly chosen reservoir input variable x_j . Consequently, each x_{dim} -dimensional row in \mathbf{W}_{in} contains exactly one non-zero element, whose value is randomly drawn from a uniform distribution within the range $[-\sigma, \sigma]$. The hyperparameter σ represents the input strength.

The reservoir network \mathbf{A} introduces the recurrent connections between consecutive reservoir states $\mathbf{r}(t)$ and $\mathbf{r}(t - \Delta t)$. Following the usual implementation,²⁴ it is realized by a weighted Erdős-Rényi network, which is generated as follows. Each possible undirected edge between reservoir node i and j with $i, j \in [1, \dots, r_{\text{dim}}]$, $i \neq j$, is generated with a probability p , resulting in an average node degree of $d = p(r_{\text{dim}} - 1)$. For each of the created undirected edges, both of the directed edges constituting the undirected edge are weighted independently with a random number drawn from a uniform distribution within the range $[-1, 1]$. Finally, in order to control the overall scaling applied by the reservoir matrix, each of its elements is uniformly rescaled to a specific spectral radius $\rho > 0$, which is usually chosen to be smaller than one to ensure that the influence of past inputs fades away with time. This property is linked to the *echo state property*, which has encountered some controversy within the reservoir computing literature.²⁵

Each element of the r_{dim} -dimensional node bias vector \mathbf{b} is randomly drawn from the uniform distribution $[-\sigma_b, \sigma_b]$, where the σ_b refers to the node bias scale.

3. Training

For the training phase, knowledge of a $N_{TS} + N_T$ long training sequence of the true time series $\mathbf{u}(t)$ is assumed. During the training phase, the time series input of the RC architecture, as defined in Sec. II A 1, is taken to be the actual time series data: $\tilde{\mathbf{u}}(t) = \mathbf{u}(t)$. The training sequence is split into a N_{TS} long synchronization sequence $\mathbf{U}_{sync} = [\mathbf{u}(t_1), \dots, \mathbf{u}(t_{N_{TS}})]$, a subsequent $N_T - 1$ long input training sequence $\mathbf{U}_T = [\mathbf{u}(t_{N_{TS}+1}), \dots, \mathbf{u}(t_{N_{TS}+N_T-1})]$, and another $N_T - 1$ long output training sequence shifted one time step into the future $\mathbf{Y} = [\mathbf{u}(t_{N_{TS}+2}), \dots, \mathbf{u}(t_{N_{TS}+N_T})]$.

As a first step, the standard-scaler function $S(\cdot)$ is fitted to center the reservoir input $\mathbf{x}(t) = f_{inp}(\mathbf{u}(t))$ and rescale it to unit-variance across the training sequence \mathbf{U}_T . The standard-scaler function acts on each dimension $i = 1, \dots, x_{dim}$ of the reservoir input variable $x_i(t)$ as

$$S(\mathbf{x}(t))_i = \frac{1}{s_i} (x_i(t) - \bar{x}_i), \tag{6}$$

where $\bar{x}_i = \langle x_i(t) \rangle$ and $s_i = \sqrt{\langle (x_i(t) - \bar{x}_i)^2 \rangle}$ represent the dimension-wise mean and standard deviation across the training sequence $f_{inp}(\mathbf{U}_T)$ of reservoir inputs, respectively.

Once the standard-scaler function is fitted, the synchronization and input training sequences \mathbf{U}_{sync} and \mathbf{U}_T are transformed to create the corresponding standardized reservoir input $\hat{\mathbf{x}}(t)$ sequences, which are then used to drive the reservoir via Eq. (3). In order to do so, the initial state of the reservoir is initialized to zero, $\mathbf{r}(t_0) = [0, \dots, 0]^T$, and then synchronized to the time series data after being driven with \mathbf{U}_{sync} , arriving at the synchronized reservoir state $\mathbf{r}(t_{N_{TS}})$. The synchronization aims to remove any dependency on the arbitrarily chosen initial condition $\mathbf{r}(t_0)$. The subsequent reservoir states generated by driving with the training input data \mathbf{U}_T are transformed to the RR input variables $\mathbf{h}(t)$ via Eq. (4) and collected into a $h_{dim} \times (N_T - 1)$ matrix $\mathbf{H} = [\mathbf{h}(t_{N_{TS}+1}), \dots, \mathbf{h}(t_{N_{TS}+N_T-1})]$.

The output matrix \mathbf{W}_{out} and intercept \mathbf{w}_{out} and are then obtained by ridge regression (including an unregularized intercept) by fitting the collection of RR input variables \mathbf{H} to the collection of desired outputs \mathbf{Y} represented as a $u_{dim} \times (N_T - 1)$ matrix. The RR solutions for \mathbf{W}_{out} and \mathbf{w}_{out} are calculated as²⁴

$$\mathbf{W}_{out} = \mathbf{Y}\mathbf{H}_c^T (\mathbf{H}_c\mathbf{H}_c^T + \beta\mathbf{I})^{-1}, \tag{7}$$

$$\mathbf{w}_{out} = \bar{\mathbf{y}} - \mathbf{W}_{out}\bar{\mathbf{h}}. \tag{8}$$

$\bar{\mathbf{y}}$ and $\bar{\mathbf{h}}$ represent the dimension-wise mean across the datasets \mathbf{Y} and \mathbf{H} , respectively. \mathbf{H}_c represents the centered RR input dataset $\mathbf{H}_c = \mathbf{H} - \bar{\mathbf{h}}$. The regularization parameter $\beta > 0$ is a crucial hyperparameter in RC which aims to avoid overfitting.

To summarize the training phase, the reservoir's output Eq. (5) was trained to approximate the next step of the time series, $\mathbf{y}_r(t) \approx \mathbf{u}(t + \Delta t)$, when driven with the true time series $\dots, \mathbf{u}(t - \Delta t), \mathbf{u}(t)$ up to time t .

4. Prediction

Once the output matrix and intercept are fitted, the RC model is able to generate arbitrarily long predictions continuing a given N_{PS} long sequence of the true time series $\mathbf{u}(t)$. In order to do so, the N_{PS}

long sequence of the true time series is used to re-synchronize the reservoir state by driving the reservoir via Eqs. (1)–(3), as was done in the training phase. Having synchronized the reservoir, an initial prediction is obtained from the reservoir's output via Eqs. (4)–(5),

$$\begin{aligned} \mathbf{y}_r(t_{N_{PS}}) &= \mathbf{W}_{out} \cdot f_{out}(\mathbf{r}(t_{N_{PS}}), \mathbf{u}(t_{N_{PS}})) + \mathbf{w}_{out} \\ &\approx \mathbf{u}(t_{N_{PS}+1}). \end{aligned} \tag{9}$$

Future predictions are then simply obtained by using the reservoir's previous prediction as the new time series input: $\tilde{\mathbf{u}}(t + \Delta t) = \mathbf{y}_r(t)$. Consequently, Eqs. (1)–(5) can be run in a loop for an arbitrary number of steps, and the generated outputs $\mathbf{y}_r(t)$ represent the time series forecast.

B. Hybrid reservoir computing

1. Knowledge-based model

In the original implementation by Pathak *et al.*,¹⁴ the knowledge-based model (KBM) takes the form of a function acting on the state of time series and producing an imperfect next-step prediction,

$$K(\mathbf{u}(t)) \approx \mathbf{u}(t + \Delta t). \tag{10}$$

In that implementation, the KBM output dimension K_{dim} equals the time series dimension u_{dim} . In principle, Eq. (10) can be relaxed so that $K(\cdot)$ represents any function that contains physical knowledge about the system, which does not necessarily have to be an approximate next-step predictor. In these cases, $K_{dim} \neq u_{dim}$ is also possible. For instance, in Racca and Magri,¹⁸ a reduced-order KBM is obtained by using a Proper Orthogonal Decomposition, leading to $K_{dim} < u_{dim}$.

2. Input hybrid

In the input hybrid (IH) approach, the reservoir input $\mathbf{x}(t)$ is given as the concatenation of the time series input $\tilde{\mathbf{u}}(t)$ and the KBM output $K(\tilde{\mathbf{u}}(t))$,

$$\mathbf{x}(t) = f_{inp}(\tilde{\mathbf{u}}(t)) = \begin{bmatrix} \tilde{\mathbf{u}}(t) \\ K(\tilde{\mathbf{u}}(t)) \end{bmatrix}. \tag{11}$$

This extends the former input dimension of normal data-driven RC, $x_{dim} = u_{dim}$, to $x_{dim} = u_{dim} + K_{dim}$. Consequently, the input matrix \mathbf{W}_{in} becomes a $r_{dim} \times (u_{dim} + K_{dim})$ matrix, where the first u_{dim} columns connect the pure input $\tilde{\mathbf{u}}(t)$ to the reservoir nodes and the last K_{dim} columns connect the knowledge-based predictor $K(\tilde{\mathbf{u}}(t))$ to the reservoir nodes, $\mathbf{r}(t)$, respectively. The original implementation by Pathak *et al.*¹⁴ additionally introduces a value $\gamma \in [0, 1]$ defining the fraction of reservoir nodes that are exclusively connected to the pure input $\mathbf{u}(t)$ by controlling the generation of the input matrix \mathbf{W}_{in} . Here, the additional KBM inputs and pure inputs are treated equally.

3. Output hybrid

In the output hybrid (OH) approach, the ridge regression input variables, which are given by the pure reservoir states $\mathbf{h}(t) = \mathbf{r}(t)$ in the usual data-driven RC, are extended by the knowledge based

model, $K(\tilde{\mathbf{u}}(t))$. Consequently, the OH approach is implemented by setting

$$\mathbf{h}(t) = f_{\text{out}}(\mathbf{r}(t), \tilde{\mathbf{u}}(t)) = \begin{bmatrix} \mathbf{r}(t) \\ K(\tilde{\mathbf{u}}(t)) \end{bmatrix}. \quad (12)$$

As a result, the output matrix \mathbf{W}_{out} now has the shape of $u_{\text{dim}} \times (r_{\text{dim}} + K_{\text{dim}})$, where the first r_{dim} rows connect the reservoir states and the last K_{dim} rows connect the KBM to the output, respectively. The reservoir output Eq. (5) can then be rewritten as

$$\mathbf{y}_r(t) = \mathbf{W}_{\text{out}} \begin{bmatrix} \mathbf{r}(t) \\ K(\tilde{\mathbf{u}}(t)) \end{bmatrix} + \mathbf{w}_{\text{out}}, \quad (13)$$

$$= \underbrace{\mathbf{W}_{\text{res}} \mathbf{r}(t)}_{\mathbf{y}_{\text{res}}(t)} + \underbrace{\mathbf{W}_{\text{kbm}} K(\tilde{\mathbf{u}}(t))}_{\mathbf{y}_{\text{kbm}}(t)} + \mathbf{w}_{\text{out}}. \quad (14)$$

The partial output matrices, \mathbf{W}_{res} and \mathbf{W}_{kbm} , are of the shape $u_{\text{dim}} \times r_{\text{dim}}$ and $u_{\text{dim}} \times K_{\text{dim}}$, respectively. The partial outputs $\mathbf{y}_{\text{res}}(t)$ and $\mathbf{y}_{\text{kbm}}(t)$ represent the reservoir and KBM contributions to the total output $\mathbf{y}_r(t)$, respectively. As the data-driven and model-informed information are only combined in the linear output layer, investigating $\mathbf{y}_{\text{res}}(t)$ and $\mathbf{y}_{\text{kbm}}(t)$ allows for a straightforward interpretation of their respective influence on the prediction. In Sec. IV, the dimension-wise standard deviations of $\mathbf{y}_{\text{res}}(t)$ and $\mathbf{y}_{\text{kbm}}(t)$, $\text{std}(y_{\text{res}/\text{kbm},j})$, will be presented for selected cases, which are generated when driving the reservoir with the training input.

4. Full hybrid

In the full hybrid (FH) approach, as it was initially introduced by Pathak *et al.*,¹⁴ both the input hybrid approach via Eq. (11), and output hybrid approach via Eq. (12) are employed simultaneously.

5. Remarks

The three possible hybrid RC approaches, IH, OH, and FH, alongside the normal data-driven approach are visualized in Fig. 1. The main idea of these hybrid approaches is to make the physical knowledge, $K(\tilde{\mathbf{u}}(t))$, available within the reservoir's prediction. The output hybrid framework works in a straightforward manner, with ridge regression finding the optimal weights in \mathbf{W}_{out} that combine the reservoir states $\mathbf{r}(t)$ with the KBM output $K(\tilde{\mathbf{u}}(t))$ to learn the next step in the time series. In the input hybrid approach, the KBM output is fed into the reservoir as additional input, making it less straightforward to interpret.

In contrast to the original implementation,¹⁴ the hybrid approach defined in this work employs the standard-scaler function $S(\cdot)$. In the usual data-driven RC approach, the additional standard-scaler layer is not necessarily needed, as the whole training time series dataset can just be rescaled to zero-mean and unit-variance. However, this leads to issues in the hybrid approaches, as the KBMs assume the unscaled original time series $\mathbf{u}(t)$ as its input. This shortcoming is overcome by employing the additional standard-scaler layer.

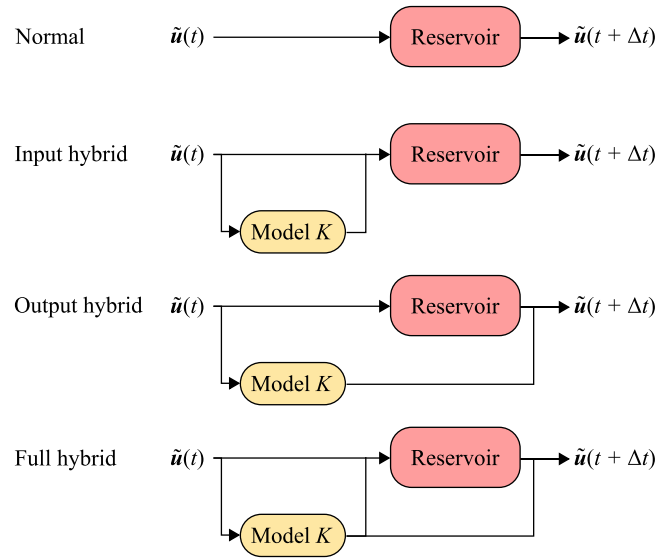


FIG. 1. Different hybrid reservoir computing architectures.

III. IMPLEMENTATION

A. Time series data from dynamical systems

The hybrid reservoir computing approaches are applied to the task of forecasting nine three-dimensional chaotic systems, as well as the spatiotemporal chaotic Kuramoto–Sivashinsky system. All systems are numerically simulated with a time step of Δt , essentially reducing them to a map,

$$\mathbf{u}(t + \Delta t) = K_{\text{true}}(\mathbf{u}(t)). \quad (15)$$

Due to the chaotic nature of the considered time series, any forecast is destined to fail after some finite time, as errors grow exponentially in time. This property is captured by a positive largest Lyapunov exponent, $\lambda_{\text{max}} > 0$, and corresponding Lyapunov time $1/\lambda_{\text{max}}$, after which the error has increased on average by a factor e . The largest Lyapunov exponents of the various systems are calculated from their respective maps [Eq. (15)] using the algorithm described in Sprott,²⁶ as outlined in Appendix B.

1. 3D chaotic dynamical systems

The evolution of the three-dimensional chaotic dynamical systems is described by a flow $F(\cdot)$ acting on the three-dimensional state $\mathbf{u}(t)$ of the dynamical system,

$$\dot{\mathbf{u}}(t) = F(\mathbf{u}(t)). \quad (16)$$

The discrete time series $\mathbf{u}(t_0), \mathbf{u}(t_0 + \Delta t), \dots$ is generated by numerically solving the flow Eq. (16) using the fourth-order Runge-Kutta method (RK4): $k_{\text{true}} = k_{\text{RK4}}$. The flow equations, parameter values, time steps Δt , initial conditions, and largest Lyapunov values of the nine chaotic dynamical systems tested are listed in Appendix A.

2. Spatiotemporal chaotic Kuramoto–Sivashinsky system

In order to investigate the performance of the hybrid RC architectures in forecasting high-dimensional spatiotemporal chaotic systems, the one-dimensional Kuramoto–Sivashinsky (KS) equation^{27,28} is used. The KS system is described by the following partial differential equation (PDE) for the 1D state function $u(x, t)$:

$$\dot{u}(x, t) = -uu_x - au_{xx} - u_{xxx}. \quad (17)$$

As is common, the parameter a is set to one: $a = 1$. Here, the parameter a is explicitly introduced, as it will be modified within the ε -model described in Sec. III B 1, to create an artificially imperfect model that will then be used as the knowledge-based model. The KS system is numerically simulated using the ETDRK4 method.²⁹ $k_{\text{true}} = k_{\text{ETDRK4}}$. The simulation is conducted on a grid of $u_{\text{dim}} = 64$ uniformly spaced grid points, employing periodic boundary conditions with a domain size of $L = 35$ and a time step of $\Delta t = 0.25$. The choice of u_{dim} , L , and Δt corresponds to the choice in the original hybrid RC implementation.¹⁴ As in Kassam and Trefethen,²⁹ the initial condition for the u_{dim} -dimensional state vector is given as

$$\mathbf{u}(t_0) = \cos\left(\frac{2\pi}{u_{\text{dim}}}[1, 2, \dots, u_{\text{dim}}]^T\right) \times \left(1 + \sin\left(\frac{2\pi}{u_{\text{dim}}}[1, 2, \dots, u_{\text{dim}}]^T\right)\right). \quad (18)$$

The largest Lyapunov exponent is calculated as $\lambda_{\text{max}} = 0.07489$ using the method outlined in Appendix B, which corresponds well to the value noted in Pathak *et al.*¹⁴ (≈ 0.07).

B. Tested knowledge-based models

This section introduces the various KBMs $K(\cdot)$ that will be utilized to test the different hybrid approaches. While the flow- and sine-models are only applied to the nine three-dimensional systems, the ε -model is also applied to the spatiotemporal KS system.

1. ε -model K_ε

The ε -model, as was originally used by Pathak *et al.*,¹⁴ is created by modifying one parameter value of the true dynamical system's equation (flow for the three-dimensional systems and PDE for the KS system) and then using the same numerical integration method as for the true data (RK4 or EDTRK4), in order to artificially create an imperfect predictor $K_\varepsilon(\cdot)$. The modified parameter is obtained by multiplying one of the true system's parameters by a factor $(1 + \varepsilon)$. Consequently, the model error ε controls the level of imperfection, where $\varepsilon = 0$ corresponds to the true knowledge-based model: $K_{\varepsilon=0} = K_{\text{true}}$.

The chosen parameters for each of the nine chaotic dynamical systems, as well as for the KS system, which are modified within the ε -model, are displayed in Table I.

TABLE I. Selected parameters of the dynamical systems, which are modified within the ε -model.

System	Eq.	ε -parameter	System	Eq.	ε -parameter
Lorenz-63	(A1)	ρ	Roessler	(A6)	c
Chen	(A2)	a	Rucklidge	(A7)	κ
ChuaCircuit	(A3)	α	Thomas	(A8)	b
DoubleScroll	(A4)	a	Windmi	(A9)	a
Halvorsen	(A5)	a	KS	(17)	a

2. Flow-model K_{flow}

The flow-model is simply given as the flow of the true three-dimensional dynamical system: $K_{\text{flow}} = F$. The flow-model is incorporated into this study to investigate whether a KBM that does not function as a next-step predictor, $K_{\text{flow}}(\mathbf{u}(t)) \not\approx \mathbf{u}(t + \Delta t)$, yet still contains information about the system, can be leveraged by the hybrid approaches to improve the prediction qualities when compared to usual data-driven RC.

3. Sine-model K_{sin}

To examine the impact of an extremely inaccurate model that incorporates no knowledge about the underlying system but still produces a nonlinear transformation of the time series state, the sine-model is employed,

$$K_{\text{sin}}(\mathbf{u}(t)) = \sin(\mathbf{u}(t)). \quad (19)$$

The sine function is applied to every dimension of its argument.

C. Experimental setup

In this section, the experimental setup used to evaluate the prediction qualities of the data-driven and hybrid RC approaches is described.

1. Forecast horizon measure

In order to measure the quality of a single prediction, the forecast horizon (also known as valid time) is calculated, which measures the time that the true time series closely matches the predicted time series. As described in Sec. II A 4, the reservoir has to be synchronized with a N_{PS} long sequence of the true time series $\mathbf{u}(t)$ before the forecast can be generated. The following N_{p} long forecast generated by the reservoir's output $\mathbf{y}_{\text{r}}(t)$ can then be compared to the true continuation of the time series, denoted as $\mathbf{y}(t)$. The forecast horizon t_{v} is obtained as the elapsed time before the normalized, time-dependent error $e(t)$ exceeds a threshold value e_{max} .¹⁴

$$e(t) = \frac{\|\mathbf{y}(t) - \mathbf{y}_{\text{r}}(t)\|}{\langle \|\mathbf{y}(t)\|^2 \rangle^{1/2}}. \quad (20)$$

Here, $\langle \cdot \rangle$ represents the average over all prediction steps N_{p} and $\|\cdot\|$ denotes the L2-norm over all state space dimension. The forecast horizon is defined as the time t_{v} when $e(t) > e_{\text{max}}$ for the first time. It is measured in terms of the Lyapunov times $1/\lambda_{\text{max}}$ of the

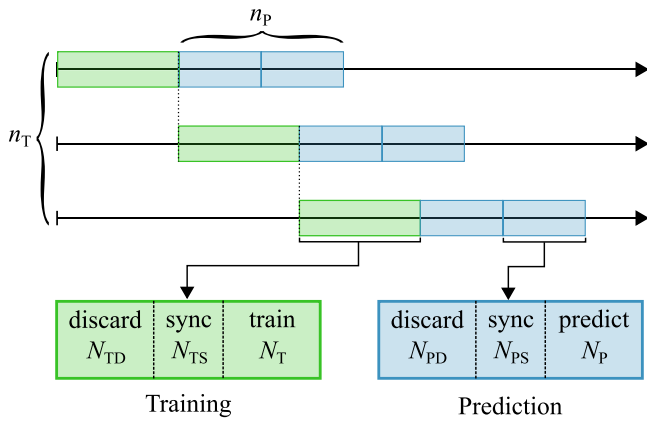


FIG. 2. Training and prediction sections of the ensemble experiment. For demonstration purposes, $n_T = 3$ and $n_p = 2$ are used.

dynamical system to be predicted, in order to ensure greater comparability across different systems. The threshold is chosen to be $\epsilon_{\max} = 0.4$.

2. Ensemble experiment method

As the RC framework contains three sources of randomness, namely, the input matrix \mathbf{W}_{in} , the reservoir network \mathbf{A} , and the node bias vector \mathbf{b} , different random realizations of these quantities result in different predictions of various qualities. Therefore, to properly assess the prediction quality of a particular reservoir architecture defined by its hyperparameters, the prediction must be repeated for an ensemble of reservoir realizations. Furthermore, as the prediction quality also depends on the specific time series sections used for training and prediction, the statistical significance of the results is further improved by training and predicting each reservoir realization on multiple sections of the time series. The following describes the implementation details of the employed ensemble experiment.

To begin, an ensemble of n_{res} random realizations of the quantities $(\mathbf{W}_{in}, \mathbf{A}, \mathbf{b})$ is generated as described in Sec. II A 2, each representing one RC realization i_{res} . Each RC realization is trained on the same n_T consecutive training sections resulting in $n_{res} \times n_T$ trained reservoirs. Each training-section i_T contains $N_{TD} + N_{TS} + N_T$ time steps, which includes N_{TS} train-synchronization steps and N_T train-fitting steps, as described in Sec. II A 3. To ensure that the different training sections are independent of each other, the first N_{TD} steps of each section are discarded. Next, each reservoir realization i_{res} that was trained on the training section i_T predicts on n_p consecutive prediction sections following the training section, resulting in $n_{res} \times n_T \times n_p$ predictions. Each prediction section consists of $N_{PD} + N_{PS} + N_P$ time steps, where the first N_{PD} time steps are again discarded for greater independence and to traverse larger portions of the attractor. The following N_{PS} time steps are used for synchronization, after which the prediction is performed for N_P time steps, as described in Sec. II A 4. Finally, for each i_{res} , i_T , and i_p , the reservoir prediction is compared to the true time series of that prediction section, and the forecast horizon $t_v(i_{res}, i_T, i_p)$ is recorded. As a final

TABLE II. Number of time steps used for the training and prediction sections, as described in Sec. III C 2.

System	Training			Prediction		
	N_{TD}	N_{TS}	N_T	N_{PD}	N_{PS}	N_P
3D chaotic systems	1000	100	2000	1000	100	2000
Kuramoto–Sivashinsky	7000	200	10 000	1000	200	1500

measure of the prediction qualities, the median value from the set of $n_{res} \times n_T \times n_p$ forecast horizons is calculated, with the lower and upper quartiles serving as the range of uncertainty.

The data points for the standard deviations for the reservoir and KBM output contributions in the OH model during the training phase, $\text{std}(y_{res/kbm,j})$, are also obtained as the median value across all $n_{res} \times n_T$ trained reservoirs, with the error bar given by the lower and upper quartiles.

The selection of training and prediction sections is visualized in Fig. 2. The number of realizations, training sections, and prediction sections is chosen to be $n_{res} = n_T = 15$ and $n_p = 10$, resulting in an effective ensemble of $15 \times 15 \times 10 = 2250$ forecast horizon calculations. The number of time steps used per training and prediction section is displayed in Table II.

3. RC hyperparameter values

The hyperparameter values defining the RC setup (data-driven and hybrid) are chosen in a heuristic manner, corresponding to the usual parameter ranges used in RC, and are listed in Table III. Preliminary experiments have demonstrated that a spectral radius of $\rho \approx 0.4$ yields favorable results across most of the tested systems.

IV. RESULTS

In order to investigate the predictive powers of the three hybrid models (IH, OH, and FH), we compare the forecast horizons for these architectures with each other and with a conventional non-hybrid reservoir, utilizing the three knowledge-based models: ϵ -model, flow-model, and sine-model. Additionally, in order to assess the predictive powers of the KBMs alone, the forecast horizons of the linearly fitted KBMs are also shown. This approach corresponds to the OH/FH architecture without a reservoir ($r_{dim} = 0$), and its output reduces to

$$y_{kbm-fitted}(t) = \mathbf{W}_{kbm}K(\tilde{\mathbf{u}}(t)) + \mathbf{w}_{out}. \tag{21}$$

Among the considered KBMs, only the ϵ -model functions as a next-step predictor directly: $\mathbf{u}(t + \Delta t) \approx K_\epsilon(\mathbf{u}(t))$. These unaltered ϵ -model forecasts are additionally presented as a baseline within the

TABLE III. Chosen RC hyperparameter values.

Reservoir		Input		Node bias	Ridge regression
r_{dim}	d	ρ	σ	σ_b	β
Varied	5.0	0.4	1.0	0.4	10^{-7}

ε -model results,

$$y_{\text{kbm-only}}(t) = K_\varepsilon(\tilde{u}(t)). \quad (22)$$

While all KBMs (ε -model, flow-model, sine-model) are tested for the predictions of the nine three-dimensional chaotic systems with a particular focus on the Lorenz-63 system, only the ε -model is applied for the prediction of the high-dimensional spatiotemporal Kuramoto–Sivashinsky system.

A. ε -model results for 3D systems and high-dimensional Kuramoto–Sivashinsky system

1. 3D systems

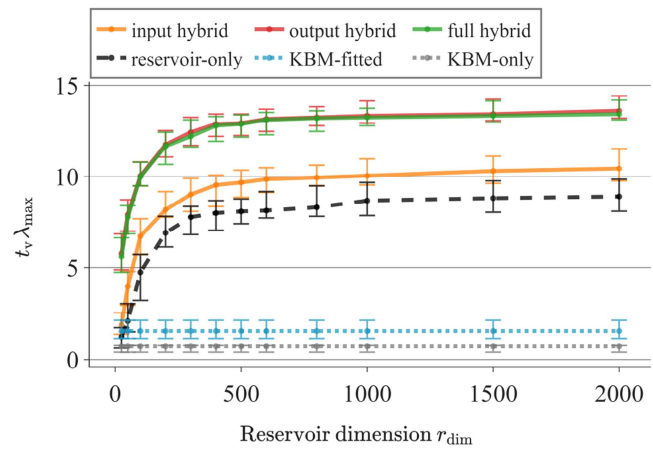
Figure 3 shows the influence of reservoir dimension r_{dim} and model error ε on the forecast horizons of the Lorenz-63 system.

With the model error fixed at $\varepsilon = 0.1$, Fig. 3(a) clearly shows that all three hybrid approaches significantly increase the forecast horizons compared to either the reservoir-only, the KBM-only, or the KBM-fitted prediction for all reservoir sizes tested: $r_{\text{dim}} = 25 \rightarrow 2000$. The pure KBM-only forecast provides the overall worst prediction results. Similar to what is typically observed in reservoir-only approaches, the hybrid approaches also extend their forecast horizons as the reservoir size r_{dim} increases, after which they saturate at one point. Within the hybrid approaches, the predictive results of OH and FH are identical and superior to those of the IH approach. While the predictive powers of the reservoir-only and IH approaches are diminished as the reservoir size decreases, those of the OH and FH approaches remain quite accurate, even for the smallest tested reservoir size of $r_{\text{dim}} = 25$: $t_v \lambda_{\text{max}} \approx 6$. On the other hand, the limited predictive power of the KBM-fitted approach, corresponding to the OH/FH approach with $r_{\text{dim}} = 0$, demonstrates that the reasonably accurate predictions of the OH/FH approaches with a small reservoir are not solely attributed to the KBM, but rather require at least a small reservoir size. Since the IH approach employs the KBM only as an additional reservoir input, it requires a sufficiently large reservoir to effectively leverage the KBM.

Figure 3(b) shows the effect of a varying model error ε for a fixed reservoir dimension of $r_{\text{dim}} = 500$. As expected, the forecast horizons of all hybrid approaches as well as the KBM-only and KBM-fitted approaches increase as the model error decreases from an extremely inaccurate model, $\varepsilon = 100$, to an extremely accurate model, $\varepsilon = 10^{-4}$. This effect is particularly prominent for the KBM approaches, both of which approximately reach the forecast horizon of the reservoir-only prediction for $\varepsilon = 10^{-4}$ ($t_v \lambda_{\text{max}} \approx 7.5$). While this increased accuracy of the KBM is clearly mirrored in the OH and FH approaches, enabling them to reach a high forecast horizon of $t_v \lambda_{\text{max}} \approx 15$, the IH approach interestingly saturates for $\varepsilon < 1$, achieving a maximal forecast horizon of $t_v \lambda_{\text{max}} \approx 10$. This demonstrates that the OH and FH approaches, both of which implement the KBM in the linear output layer, benefit more strongly from an extremely accurate model compared to the IH approach, where the KBM has to traverse through the nonlinear reservoir. At a high model error of $\varepsilon = 100$, the OH approach is equivalent to the reservoir-only approach, whereas the FH and IH approaches exhibit even slightly lower forecast horizons.

Figure 4 visualizes the reservoir and KBM contributions of the OH approach for predicting the Lorenz-63 system with a smaller

a) $\varepsilon = 0.1, r_{\text{dim}} = \text{varied}$



b) $\varepsilon = \text{varied}, r_{\text{dim}} = 500$

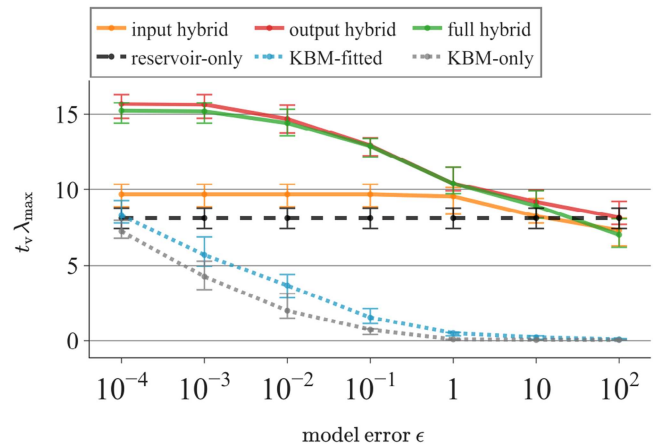


FIG. 3. Forecast horizon of Lorenz-63 system for the three hybrid RC, reservoir-only, KBM-only, and KBM-fitted approaches using the ε -model. (a) The model error is fixed at $\varepsilon = 0.1$, and the reservoir dimension is varied from $r_{\text{dim}} = 25 \rightarrow 2000$. (b) The reservoir dimension is fixed at $r_{\text{dim}} = 500$, and the model error is varied from $\varepsilon = 10^{-4} \rightarrow 10^2$.

model error of $\varepsilon = 0.1$ and a larger error of $\varepsilon = 1$. As described in Sec. II B 3, in the OH approach, the reservoir and KBM contributions [$y_{\text{res}}(t)$ and $y_{\text{kbm}}(t)$] to the overall output $y_r(t)$ can be easily separated since they are simply combined additively according to Eq. (14). With a smaller model error of $\varepsilon = 0.1$, panel (a), most of the total output $y_r(t)$ (“Both”) consists of the accurate KBM contribution, while the reservoir contribution is very small (green dot below “KBM” and “Both” trajectory). Nevertheless, the spatially small reservoir contribution plays an important role in predictive powers, as observed in Fig. 3, where the OH prediction with $\varepsilon = 0.1$ and $r_{\text{dim}} = 500$ achieves a high forecast horizon of $t_v \lambda_{\text{max}} \approx 13$, whereas the forecast horizon of the KBM-fitted approach is as low as $t_v \lambda_{\text{max}} \approx 2.5$. This behavior is consistent across the ensemble

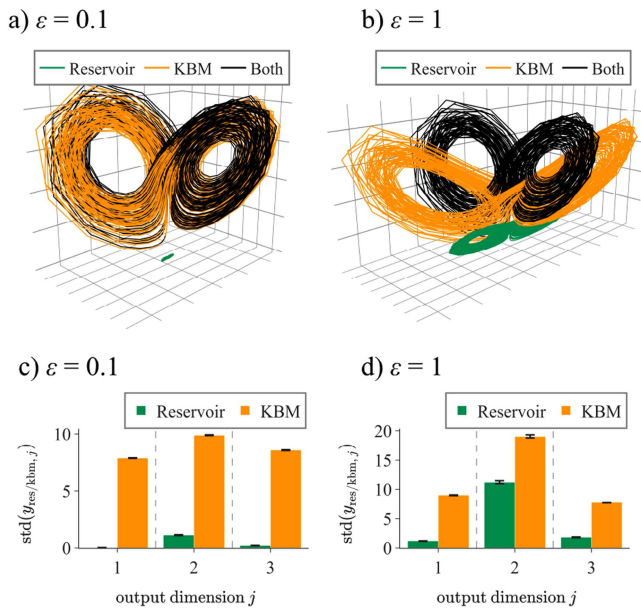


FIG. 4. Partial reservoir outputs of the OH approach for the Lorenz-63 system using the ε -model ($\varepsilon = 0.1$ and $\varepsilon = 1$) as the KBM and a reservoir dimension of $r_{\text{dim}} = 500$. Panels (a) and (b) display the trajectories of the partial outputs $y_{\text{res}}(t)$ (“Reservoir”) and $y_{\text{kbm}}(t)$ (“KBM”), and the resulting full output $y_t(t)$ (“Both”), obtained from the first training fit ($t_{\text{res}} = t_T = 1$). Panels (c) and (d) display the corresponding standard deviations across each dimension (median value and lower/higher quartile over all n_{ens} reservoirs and n_T training sections).

of reservoir realizations n_{ens} and training sections n_T , as evident from the small error bars in panel (c), which indicate the lower and upper quartile around the median value of the respective contributions’ standard deviations. As expected, for a higher model error of $\varepsilon = 1$, panels (b) and (d), the spatial extent of the reservoir contribution increases, with both the KBM and reservoir contributions leading to a distorted Lorenz-63-like attractor. These results demonstrate how the output matrix W_{out} in the OH approach is capable of selecting the optimal combination of the KBM and reservoir nodes based on the model’s accuracy.

Figure 5 illustrates the forecast horizons with a rather small model error of $\varepsilon = 0.1$ for the nine different three-dimensional dynamical systems using a smaller reservoir size of $r_{\text{dim}} = 50$, panel (a), and a larger reservoir size of $r_{\text{dim}} = 500$, panel (b). The forecast horizons obtained from the KBM-fitted and KBM-only approaches (independent of r_{dim}) provide insights into the accuracies of the respective ε -models for the systems when modifying the system parameters listed in Table I. The almost equal predictive abilities of the OH and FH approaches, previously observed for the Lorenz-63 system, can also be confirmed for the other eight three-dimensional chaotic systems and both reservoir sizes tested. Their superior forecast horizons, compared to the IH, KBM, and reservoir-only approaches, once again become most apparent for the small reservoir size ($r_{\text{dim}} = 50$), wherein the IH and reservoir-only approaches struggle to achieve good prediction results. Similarly, for the larger

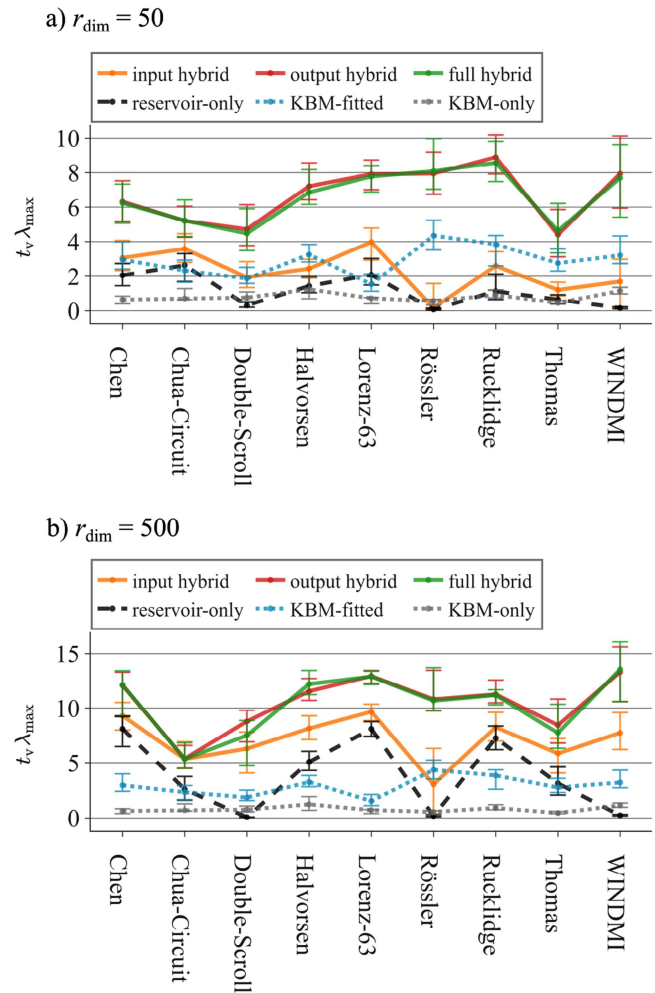


FIG. 5. Forecast horizon for the three hybrid RC, reservoir-only and KBM approaches for nine 3D chaotic model systems using the ε -model with $\varepsilon = 0.1$. Panel (a) small reservoir ($r_{\text{dim}} = 50$) and panel (b) large reservoir ($r_{\text{dim}} = 500$).

reservoir size of $r_{\text{dim}} = 500$, the OH and FH approaches are superior to the IH approach (except for the Chua-Circuit, where all three are equal). However, in this scenario, the IH approach has acquired a significant amount of predictive power, further demonstrating that the IH approach requires a sufficiently large reservoir size to function properly. In any case, the IH approach still achieves better prediction results compared to the reservoir-only approach. However, when considering the smaller reservoir, the input hybrid performs even worse than the simple KBM-fitted forecasts for certain systems. Interestingly, the reservoir-only predictions for the Double-Scroll, Rössler, and WINDMI systems are not able to achieve any accurate predictions, even for the larger reservoir size $r_{\text{dim}} = 500$. On the other hand, the hybrid approaches, particularly the OH and FH approaches, are able to generate very accurate predictions for these three systems.

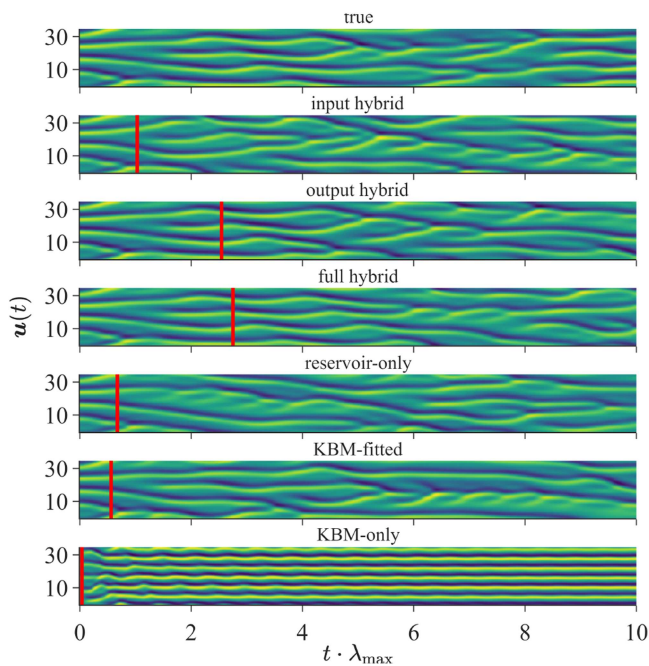


FIG. 6. Forecasts for the Kuramoto–Sivashinsky system. The vertical red lines indicate the forecast horizon. The model error is given as $\varepsilon = 1$ and the reservoir dimension as $r_{\text{dim}} = 4000$.

2. Kuramoto–Sivashinsky

The predicted trajectories corresponding to the first reservoir realization, the first training section, and the first prediction section ($i_{\text{res}} = i_{\text{T}} = i_{\text{P}} = 1$) for the spatiotemporal chaotic Kuramoto–Sivashinsky system, alongside the true simulated trajectory, are visualized in Fig. 6. The respective forecast horizons are indicated as vertical red lines for each predictor. From a qualitative standpoint, all predictors except the KBM-only approach are capable of generating KS-like trajectories, while the KBM-only approach seems to converge toward a stationary state. While all hybrid approaches outperform the reservoir-only and KBM approaches, the OH and FH approaches once again demonstrate superior prediction results.

The prediction results of this single trajectory are supported by the ensemble experiment displayed in Fig. 7, where the reservoir dimension (panel a) and model error (panel b) are varied. These results are qualitatively very similar to the previously discussed Lorenz-63 findings (see Fig. 3): the OH and FH forecast horizons are roughly equivalent and superior to the IH forecast horizons, while all hybrid approaches still perform better than the reservoir-only approach (except for a large model error $\varepsilon = 10$ in panel b). Contrary to the Lorenz-63 findings, the KS ε -model with $\varepsilon = 1$ appears to already be quite accurate, as evident from the large forecast horizon of the KBM-fitted approach (panel a), which matches the maximal forecast horizon of the reservoir-only approach with $r_{\text{dim}} = 6000$: $t_v \lambda_{\text{max}} \approx 1$. The sharp increase in forecast horizons of the OH and FH approaches, transitioning from an extremely small

reservoir of $r_{\text{dim}} = 50$ to a reservoir size $r_{\text{dim}} = 1000$ that is still relatively small, in light of the large KS time series dimension $u_{\text{dim}} = 64$, once again illustrates how a relatively small reservoir can significantly benefit from a fairly accurate KBM within the OH and FH approaches. On the other hand, the forecast horizons of the IH and reservoir-only approaches improve more slowly as the reservoir size increases. Similar to the Lorenz-63 findings, in the case of a large model error of $\varepsilon = 10$ (panel b), the OH approach seems to converge toward the reservoir-only approach, whereas the IH and FH approaches exhibit even slightly lower forecast horizons. As the model error is decreased, the predictive capabilities of the IH approach saturate already for $\varepsilon < 1$, whereas the predictive abilities of the OH and FH approaches only reach saturation for $\varepsilon < 10^{-1}$, indicating once again for the high-dimensional KS system that the latter approaches can benefit more strongly from highly accurate KBMs.

B. Flow-model results for 3D systems

In this section, the flow $F(\cdot)$ of the respective three-dimensional dynamical systems (see Appendix A for the flows of all nine 3D systems) will be used as the KBM within the three hybrid approaches, as well as for the KBM-fitted predictor. This experiment represents a case in which the KBM does not function as a next-step predictor but still contains knowledge about the underlying system. Therefore, a KBM-only predictor is not applicable in this case.

Figure 8 shows the influence of the reservoir size r_{dim} on the forecast horizons of the Lorenz-63 system. All hybrid approaches show higher forecast horizons compared to the reservoir-only approach. This demonstrates that the physical knowledge within the flow of the Lorenz-63 system can indeed be leveraged by the hybrid approaches. Unlike the comparatively accurate ε -models, where the forecast horizons of the OH and FH approaches were, in most cases, significantly higher than those of the IH approach, the usage of the flow-model leads to very similar forecast horizons across all hybrid architectures, especially for reservoir dimensions $r_{\text{dim}} > 300$. For reservoirs of smaller size, where $r_{\text{dim}} < 300$, the OH approach demonstrates slightly better prediction results, while both the IH and FH approaches align closely with the predictions of the reservoir-only model. Overall, the improvements in predictions achieved by the hybrid approaches over the reservoir-only approach are only marginal when compared to the ε -model [compare with Fig. 3(a)]. This marginal improvement is also reflected in the unsuccessful KBM-fitted prediction.

Figure 9 shows the forecast horizons for all nine chaotic dynamical systems, with a fixed reservoir size of $r_{\text{dim}} = 500$. The inaccuracy of the flow-models, compared to the ε -models, becomes evident through the unsuccessful KBM-fitted predictions for all nine systems. Nevertheless, for most systems, the hybrid methods can benefit from the flow-models and achieve longer forecast horizons compared to the reservoir-only approach. This is particularly evident in the cases of the Rössler and WINDMI systems, where the reservoir-only predictions are unsuccessful, while the hybrid approaches lead to significant prediction results. Interestingly, the Chua-Circuit and Thomas systems represent two cases in which the OH approach performs slightly worse than the reservoir-only

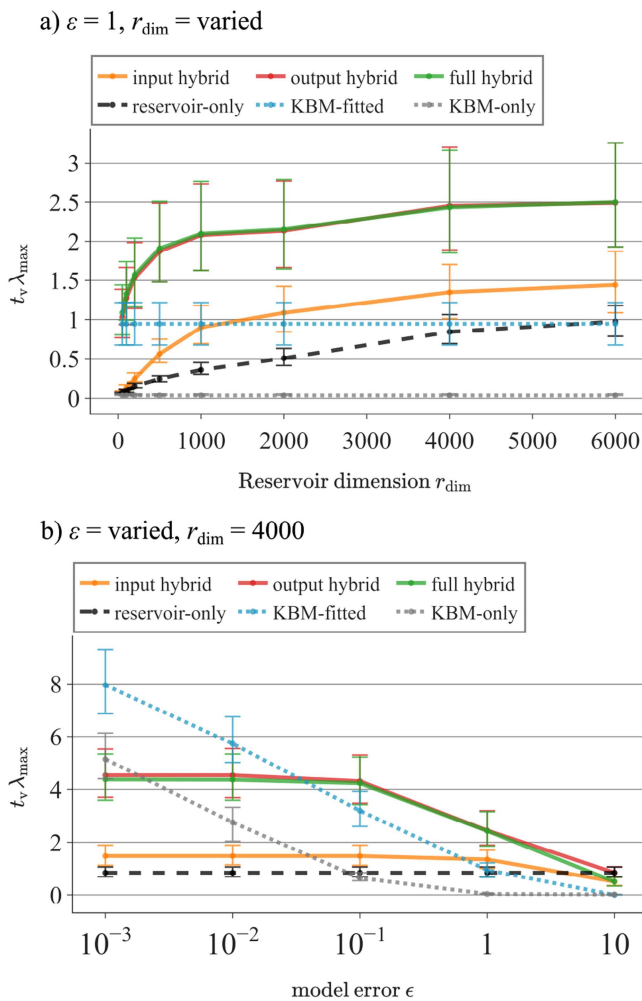


FIG. 7. Forecast horizon of spatiotemporal chaotic Kuramoto-Sivashinsky system for the three hybrid RC, reservoir-only, KBM-only, and KBM-fitted approaches using the ε -model. (a) The model error is fixed at $\varepsilon = 1$, and the reservoir dimension is varied from $r_{\text{dim}} = 50 \rightarrow 6000$. (b) The reservoir dimension is fixed at $r_{\text{dim}} = 4000$, and the model error is varied from $\varepsilon = 10^{-3} \rightarrow 10$.

approach, while the IH and FH approaches show small improvements.

C. Sine-model results for 3D systems

Finally, the sine-model, Eq. (19), is considered which represents a highly inaccurate model containing no information about the dynamical system. As for the other KBMs, Fig. 10 shows the influence of the reservoir dimension r_{dim} on the prediction of the Lorenz-63 system, while Fig. 11 depicts the forecast horizons for all nine 3D dynamical systems that were tested using a reservoir dimension of $r_{\text{dim}} = 500$. As expected, the KBM-fitted approach is not able to generate any successful predictions. For the Lorenz-63 system

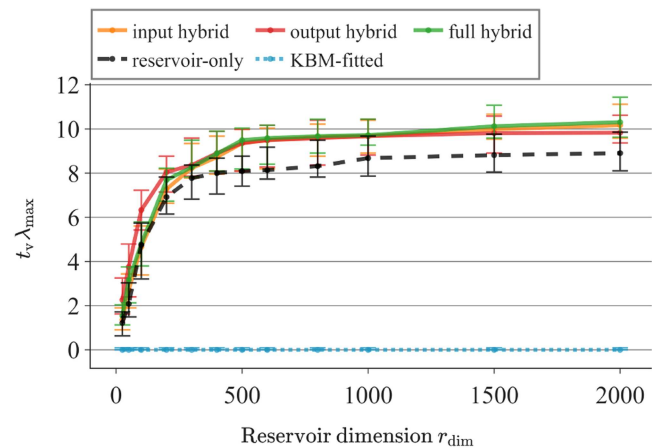


FIG. 8. Forecast horizon of Lorenz-63 system as a function of the number of reservoir nodes for the three hybrid RC, reservoir-only, and KBM-fitted approaches using the flow-model as the KBM.

shown in Fig. 10, the OH and reservoir-only approaches perform equally well and significantly outperform the IH and FH approaches, which also show equal forecast horizons. This holds true regardless of the reservoir dimension. The equivalence in performance between the OH and reservoir-only approaches, along with their superiority over the IH and FH predictions, is consistently observed across most of the other eight dynamical systems, as depicted in Fig. 11. This demonstrates that when the model is highly inaccurate, the OH approach reduces to the often still satisfactory reservoir-only prediction. On the other hand, the IH and FH approaches, which incorporate the inaccurate model into the reservoir states $r(t)$, fail to ignore the inaccurate model, and as a result, the prediction quality decreases. The Thomas system behaves differently than the other

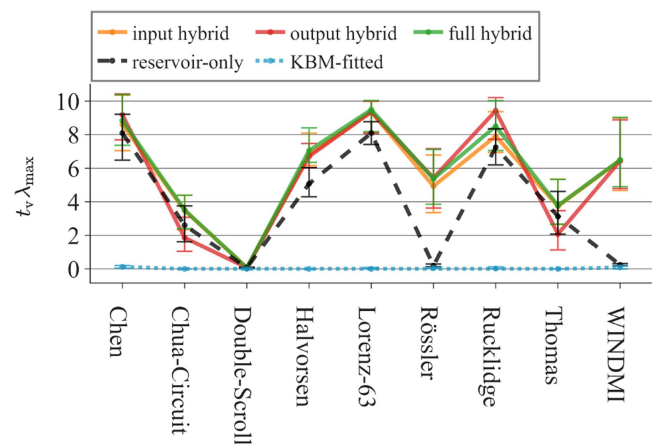


FIG. 9. Forecast horizon for the three hybrid RC, reservoir-only, and KBM-fitted approaches for nine 3D chaotic model systems using the flow-model and a reservoir dimension of $r_{\text{dim}} = 500$.

15 October 2023 19:07:07

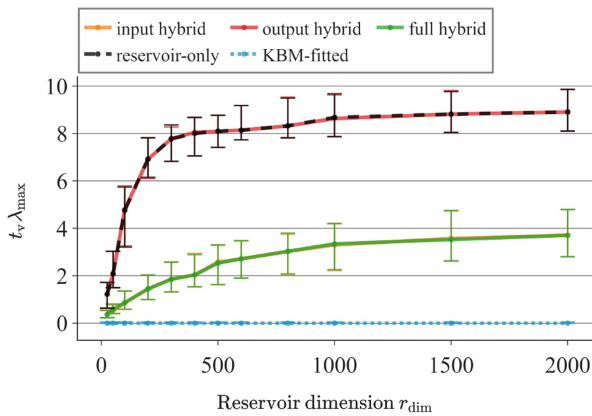


FIG. 10. Forecast horizon of Lorenz-63 system as a function of the number of reservoir nodes for the three hybrid RC, reservoir-only, and KBM-fitted approaches with an inappropriate choice of model (sine-model).

systems, as the OH approach here is actually slightly worse than the reservoir-only prediction, while the IH and FH approaches perform even slightly better. The exceptional behavior for the Thomas system can be attributed to the fact that its flow equation, Eq. (A8), includes the terms $\sin(x)$, $\sin(y)$, and $\sin(z)$. Therefore, in this case, incorporating the sine-model K_{\sin} actually introduces some relevant knowledge about the system. In the case of the OH approach, it appears to generate unstable predictions, while the IH and FH approaches seem to be able to leverage them successfully.

Figure 12 depicts the standard deviations of the partial KBM and reservoir outputs for the OH approach ($r_{\text{dim}} = 500$) for the Lorenz-63 system (panel a) and the Thomas system (panel b). The ability of the OH approach to entirely disregard a highly inaccurate KBM is demonstrated for the Lorenz-63 system, revealing that all output contributions are attributed to the reservoir nodes, without

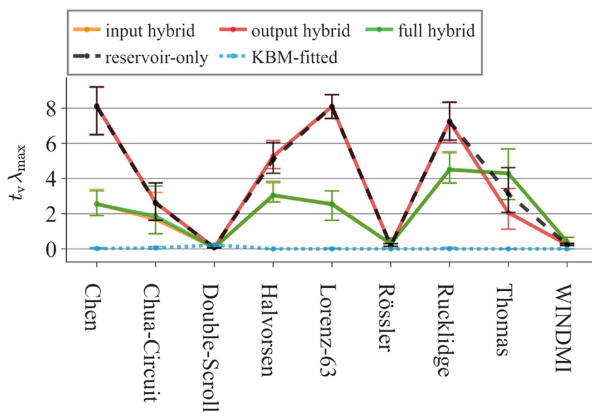


FIG. 11. Forecast horizon for the three hybrid RC, reservoir-only, and KBM-fitted approaches for nine 3D chaotic model systems with an inappropriate choice of model (sine-model) and a reservoir dimension of $r_{\text{dim}} = 500$.

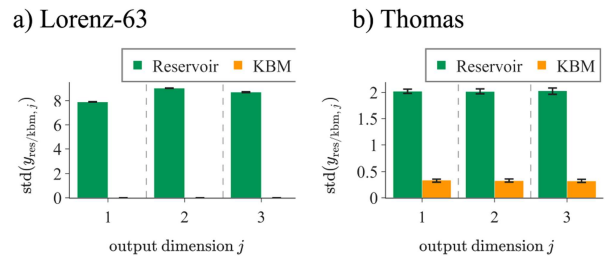


FIG. 12. Standard deviations of partial outputs y_{res} (“Reservoir”) and y_{kbm} (“KBM”) for the OH approach with $r_{\text{dim}} = 500$ and using the sine-model as the KBM. Panel (a) the Lorenz-63 system and panel (b) the Thomas system.

any contributions from the sine-model. The fact that the sine-model actually provides some information about the Thomas system can be observed in panel (b), where the KBM output contributions do not vanish. Nevertheless, as previously mentioned, the incorporation of the sine-model into the Thomas system results in less accurate predictions.

V. CONCLUSIONS

In this paper, we present a first comparative study of the three hybrid reservoir computing approaches: input hybrid (IH), output hybrid (OH), and full hybrid (FH), which combine data-driven reservoir computing with an imperfect knowledge-based model (KBM) for the task of forecasting the time evolution of chaotic dynamical systems.

For KBMs that incorporate knowledge about the underlying system, as provided by the ϵ -models and flow-models, it is demonstrated that all three hybrid approaches outperform both the reservoir-only and KBM baselines. Regarding rather accurate KBMs, as provided by the ϵ -models, distinctions among the three hybrid approaches become evident: The OH and FH approaches show equal performances and significantly outperform the IH approach. This is observed for the task of forecasting three-dimensional chaotic systems, as well as the spatiotemporal chaotic Kuramoto–Sivashinsky system. Furthermore, contrary to the IH approach, the OH and FH approaches are shown to require only very small reservoir sizes to generate fairly accurate predictions, while the IH approach, on the other hand, requires larger reservoir sizes to effectively leverage the KBM. The FH and OH approaches are additionally shown to benefit most strongly from extremely accurate KBMs, whereas the IH approach fails to show improvements after a certain point, with further increases in KBM accuracy.

When the hybrid approaches are confronted with a highly inaccurate KBM, as provided by the sine-model that incorporates no knowledge of the underlying system, the advantages of the OH approach over the FH and IH approaches become apparent. In those cases, the OH approach, which combines the KBM with the reservoir nodes only in the linear output layer, simply reduces to the data-driven reservoir-only approach, while the FH and IH approaches, which include the KBM as additional reservoir input, fail to ignore the contributions of the flawed model, leading to drastically reduced performances in the latter.

One downside of the OH and FH approaches, as noted in Huhn and Magri²² for FH, is the potential for outputs to diverge to infinity within the closed-loop prediction phase when the predictions are unsuccessful. This is due to the fact that the output hybrid KBM-to-output-connection is, in principle, unbounded (depending on the specific KBM used), in contrast to the tanh-bounded reservoir nodes. As this issue may lead to problems in hyperparameter optimization algorithms, Huhn and Magri²² proposed several ways to overcome it, such as saturating the prediction error.

However, the OH approach is the only hybrid architecture that offers the possibility to separate the KBM and reservoir contributions to the overall output, as both of them are only combined additively in the linear output layer. This allows for a higher interpretability of the knowledge-based and data-driven elements within the hybrid prediction.

In essence, the results indicate the following: For accurate KBMs, the FH and OH results are equivalent and significantly outperform the IH results, especially for smaller reservoir sizes. For highly flawed KBMs, the OH approach reduces to the reservoir-only approach, while the IH and FH approaches' performance may drastically reduce even further. This, along with the greater simplicity and interpretability of the OH approach, suggests that the OH approach is the most favorable architecture for hybrid reservoir computing.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Dennis Duncan: Conceptualization (supporting); Formal analysis (lead); Methodology (equal); Software (lead); Validation (lead); Writing – original draft (equal); Writing – review & editing (equal).
Christoph R ath: Conceptualization (lead); Funding acquisition (lead); Methodology (equal); 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.

APPENDIX A: 3D CHAOTIC DYNAMICAL SYSTEMS

This section displays the references and flow equations for the nine three-dimensional chaotic dynamical systems considered in this work. They all represent autonomous dissipative flows and are collected from the appendix in Sprott.²⁶ The original works introducing the systems are also provided in the following. The default parameters, initial states, simulation time steps Δt , and largest Lyapunov exponents (both calculated and reference values) are shown in Table IV. The default parameters and initial states, along with the reference largest Lyapunov exponent, are taken from Sprott.²⁶ The forecast horizon results are obtained using the calculated largest Lyapunov exponents. The resulting attractors of the nine systems are shown in Fig. 13.

1. Lorenz-63 system

The prototypical Lorenz-63 system was introduced by Lorenz³² as a simplified model for atmospheric convection,

$$\begin{aligned}\dot{x}(t) &= \sigma(y - x), \\ \dot{y}(t) &= x(\rho - z) - y, \\ \dot{z}(t) &= xy - \beta z.\end{aligned}\tag{A1}$$

2. Chen's system

Chen's system, which is closely related to the Lorenz-63 system,³³ was originally introduced by Chen and Ueta,³⁴

$$\begin{aligned}\dot{x}(t) &= a(y - x), \\ \dot{y}(t) &= (c - a)x - xz + cy, \\ \dot{z}(t) &= xy - bz.\end{aligned}\tag{A2}$$

3. Chua's circuit

The chaotic dynamics are stemming from a physical electronic circuit involving one nonlinear element, known as Chua's circuit. The chaotic dynamics in Chua's circuit were initially noted by Matsumoto,³⁵ and then rigorously studied by Chua *et al.*,³⁶

$$\begin{aligned}\dot{x}(t) &= \alpha[y - x + bx + 0.5(a - b)(|x + 1| - |x - 1|)], \\ \dot{y}(t) &= x - y + z, \\ \dot{z}(t) &= -\beta y.\end{aligned}\tag{A3}$$

4. Double scroll

Introduced by Elwakil and Kennedy³⁷ as a "simple model which can capture the essential dynamics of double-scroll-like chaotic attractors." It can also be realized by a physical electric circuit,

$$\begin{aligned}\dot{x}(t) &= y, \\ \dot{y}(t) &= z, \\ \dot{z}(t) &= -a[z + y + x - \text{sgn}(x)].\end{aligned}\tag{A4}$$

5. Halvorsen's cyclically symmetric attractor

This cyclically symmetric attractor was originally introduced by Arne Dehli Halvorsen (unpublished as noted in Sprott²⁶),

$$\begin{aligned}\dot{x}(t) &= -ax - 4y - 4z - y^2, \\ \dot{y}(t) &= -ay - 4z - 4x - z^2, \\ \dot{z}(t) &= -az - 4x - 4y - x^2.\end{aligned}\tag{A5}$$

6. R ssler attractor

Originally introduced by R ssler,³⁸ it is like the Lorenz-63 system, a prototypical chaotic dynamical system,

$$\begin{aligned}\dot{x}(t) &= -y - z, \\ \dot{y}(t) &= x + ay, \\ \dot{z}(t) &= b + z(x - c).\end{aligned}\tag{A6}$$

TABLE IV. System parameters and largest Lyapunov values of three-dimensional chaotic systems.

System	Parameter values	Initial state	Δt	λ_{\max} (calculated)	λ_{\max} (Sprott ²⁶)
Lorenz-63	$\rho = 28, \sigma = 10, \beta = 8/3$	$[0, -0.01, 9]$	0.05	0.9041	0.9056
Chen	$a = 35, b = 3, c = 28$	$[-10, 0, 37]$	0.02	2.0138	2.0272
Chua-Circuit	$\alpha = 9, \beta = 100/7, a = 8/7, b = 5/7$	$[0, 0, 0.6]$	0.1	0.3380	0.3271
Double-Scroll	$a = 0.8$	$[0.01, 0.01, 0]$	0.3	0.049 69	0.0497
Halvorsen	$a = 1.27$	$[-5, 0, 0]$	0.05	0.7747	0.7899
Rössler	$a = b = 0.2, c = 5.7$	$[-9, 0, 0]$	0.1	0.069 15	0.0714
Rucklidge	$\kappa = 2.0, \lambda = 6.7$	$[1, 0, 4.5]$	0.1	0.1912	0.0643 ^a
Thomas	$b = 0.18$	$[0.1, 0, 0]$	0.3	0.038 01	0.0349
WINDMI	$a = 0.7, b = 2.5$	$[0, 0.8, 0]$	0.2	0.079 86	0.0755

^aFor the Rucklidge system, the largest Lyapunov exponent noted in Sprott²⁶ does not correspond to the calculated one. In different studies, the largest Lyapunov exponent is noted as $\lambda_{\max} = 0.1877$ (Rusyn³⁰) and $\lambda_{\max} = 0.193$ (Pehlivan *et al.*³¹), which corresponds more closely to the calculated one.

7. Rucklidge attractor

The Rucklidge system was introduced by Rucklidge³⁹ as a simple model related to fluid convection in a two-dimensional plane,

$$\begin{aligned} \dot{x}(t) &= -\kappa x + \lambda y - yz, \\ \dot{y}(t) &= x, \\ \dot{z}(t) &= -z + y^2. \end{aligned} \tag{A7}$$

8. Thomas' cyclically symmetric attractor

The Thomas' attractor is, like the Halvorsen attractor, also cyclically symmetric and was originally introduced by Thomas,⁴⁰

$$\begin{aligned} \dot{x}(t) &= -bx + \sin(y), \\ \dot{y}(t) &= -by + \sin(z), \\ \dot{z}(t) &= -bz + \sin(x). \end{aligned} \tag{A8}$$

9. WINDMI attractor

The WINDMI attractor relates to the dynamics in the solar-wind-driven magnetosphere-ionosphere and was introduced by Horton *et al.*⁴¹

$$\begin{aligned} \dot{x}(t) &= y, \\ \dot{y}(t) &= z, \\ \dot{z}(t) &= -az - y + b - \exp(x). \end{aligned} \tag{A9}$$

APPENDIX B: CALCULATION OF THE LARGEST LYAPUNOV EXPONENT

The algorithm for numerically calculating the largest Lyapunov exponent is taken from Sprott,²⁶ which originated from Benettin *et al.*⁴²

The algorithm iterates the dynamical system's initial state $\mathbf{u}(t_0)$ and a slightly perturbed state $\mathbf{u}_{\text{pert}}(t_0) = \mathbf{u}(t_0) + \delta \hat{\mathbf{e}}$, with $\|\hat{\mathbf{e}}\| = 1$ and $\delta \ll 1$ for a small number of n time steps using Eq. (15), and calculates the exponential rate of divergence between the resulting states via

$$\lambda = \log \left(\frac{\|\mathbf{u}_{\text{pert}}(t_n) - \mathbf{u}(t_n)\|}{\delta} \right) \times \frac{1}{n\Delta t}. \tag{B1}$$

This procedure is repeated for $m_{\text{disc}} + m$ times using $\mathbf{u}(t_n) \rightarrow \mathbf{u}(t_0)$ as the new initial state and $\mathbf{u}(t_n) + \delta \frac{\mathbf{u}_{\text{pert}}(t_n) - \mathbf{u}(t_n)}{\|\mathbf{u}_{\text{pert}}(t_n) - \mathbf{u}(t_n)\|} \rightarrow \mathbf{u}_{\text{pert}}(t_0)$ as the new initially perturbed state that is renormalized to be again a distance of δ apart, and oriented in the direction of largest expansion. The numerical estimation of the largest Lyapunov exponent is then calculated as the average of λ over the last

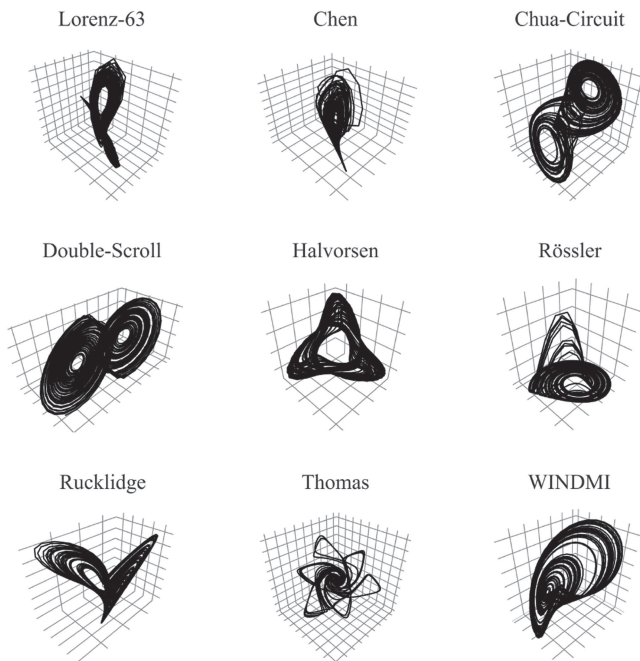


FIG. 13. Visualization of the chaotic attractors of the nine considered three-dimensional chaotic systems using the system parameters, initial states, and Δt as described in Table IV. The data points represent the first training and synchronization section consisting of $N_{TS} + N_T = 2100$ time steps.

m iterations: $\lambda_{\max} \approx \langle \lambda \rangle_m$. The first m_{skip} iterations are skipped in order to only consider points that are on the system's attractor. The parameters are chosen to be $\delta = 10^{-10}$, $n = 15$, $m_{\text{disc}} = 500$, and $m = 3000$.

REFERENCES

- ¹Y. Tang, J. Kurths, W. Lin, E. Ott, and L. Kocarev, "Introduction to focus issue: When machine learning meets complex systems: Networks, chaos, and nonlinear dynamics," *Chaos* **30**, 063151 (2020).
- ²A. Ghadami and B. I. Epureanu, "Data-driven prediction in dynamical systems: Recent developments," *Philos. Trans. R. Soc. A* **380**, 20210213 (2022).
- ³A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian, "Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: Reservoir computing, artificial neural network, and long short-term memory network," *Nonlinear Process. Geophys.* **27**, 373–389 (2020).
- ⁴S. Bompas, B. Georgeot, and D. Guéry-Odelin, "Accuracy of neural networks for the simulation of chaotic dynamics: Precision of training data vs precision of the algorithm," *Chaos* **30**, 113118 (2020).
- ⁵S. Shahi, F. H. Fenton, and E. M. Cherry, "Prediction of chaotic time series using recurrent neural networks and reservoir computing techniques: A comparative study," *Mach. Learn. Appl.* **8**, 100300 (2022).
- ⁶G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks* **115**, 100–123 (2019).
- ⁷H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks—With an erratum note," GMD Technical Report 148, German National Research Center for Information Technology, Bonn, Germany, 2001, p. 13.
- ⁸A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Trans. Knowl. Data Eng.* **29**, 2318–2331 (2017).
- ⁹J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating scientific knowledge with machine learning for engineering and environmental systems," *ACM Comput. Surv.* **55**, 1–37 (2023).
- ¹⁰Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis, "Data-assisted reduced-order modeling of extreme events in complex dynamical systems," *PLoS One* **13**, e0197704 (2018).
- ¹¹N. A. K. Doan, W. Polifke, and L. Magri, "Physics-informed echo state networks for chaotic systems forecasting," in *International Conference on Computational Science* (Springer, 2019), pp. 192–198.
- ¹²N. Doan, W. Polifke, and L. Magri, "Short-and long-term predictions of chaotic flows and extreme events: A physics-constrained reservoir computing approach," *Proc. R. Soc. A* **477**, 20210135 (2021).
- ¹³A. Racca and L. Magri, "Automatic-differentiated physics-informed echo state network (API-ESN)," in *International Conference on Computational Science* (Springer, 2021), pp. 323–329.
- ¹⁴J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model," *Chaos* **28**, 041101 (2018).
- ¹⁵A. Wikner, J. Pathak, B. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, and E. Ott, "Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems," *Chaos* **30**, 053111 (2020).
- ¹⁶T. Arcomano, I. Szunyogh, A. Wikner, J. Pathak, B. R. Hunt, and E. Ott, "A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model," *J. Adv. Model. Earth Syst.* **14**, e2021MS002712 (2022).
- ¹⁷A. Wikner, J. Pathak, B. R. Hunt, I. Szunyogh, M. Girvan, and E. Ott, "Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components," *Chaos* **31**, 053114 (2021).
- ¹⁸A. Racca and L. Magri, "Robust optimization and validation of echo state networks for learning chaotic dynamics," *Neural Networks* **142**, 252–268 (2021).
- ¹⁹F. Köster, D. Patel, A. Wikner, L. Jaurigue, and K. Lüdge, "Data-informed reservoir computing for efficient time-series prediction," *Chaos* **33**, 073109 (2023).
- ²⁰S. Shahi, C. D. Marcotte, C. J. Herndon, F. H. Fenton, Y. Shiferaw, and E. M. Cherry, "Long-time prediction of arrhythmic cardiac action potentials using recurrent neural networks and reservoir computing," *Front. Physiol.* **12**, 734178 (2021).
- ²¹F. Huhn and L. Magri, "Learning ergodic averages in chaotic systems," in *International Conference on Computational Science* (Springer, 2020), pp. 124–132.
- ²²F. Huhn and L. Magri, "Gradient-free optimization of chaotic acoustics with reservoir computing," *Phys. Rev. Fluids* **7**, 014402 (2022).
- ²³J. Herteux and C. R ath, "Breaking symmetries of the reservoir equations in echo state networks," *Chaos* **30**, 123142 (2020).
- ²⁴Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos* **27**, 041102 (2017).
- ²⁵I. B. Yildiz, H. Jaeger, and S. J. Kiebel, "Re-visiting the echo state property," *Neural Networks* **35**, 1–9 (2012).
- ²⁶J. C. Sprott, *Chaos and Time-Series Analysis* (Oxford University Press, Oxford, 2003), Vol. 69.
- ²⁷Y. Kuramoto and T. Tsuzuki, "Persistent propagation of concentration waves in dissipative media far from thermal equilibrium," *Prog. Theor. Phys.* **55**, 356–369 (1976).
- ²⁸G. I. Sivashinsky, "Nonlinear analysis of hydrodynamic instability in laminar flames-I. Derivation of basic equations," *Acta Astronaut.* **4**, 1177–1206 (1977).
- ²⁹A.-K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff PDEs," *SIAM J. Sci. Comput.* **26**, 1214–1233 (2005).
- ³⁰V. Rusyn, "Modeling, analysis and control of chaotic Rucklidge system," *J. Telecommun. Electron. Comput. Eng.* **11**, 43–47 (2019).
- ³¹I. Pehlivan, Y. Uyaroglu, and M. Yogun, "Chaotic oscillator design and realizations of the Rucklidge attractor and its synchronization and masking simulations," *Sci. Res. Essays* **5**, 2210–2219 (2010).
- ³²E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.* **20**, 130–141 (1963).
- ³³J. Lü, G. Chen, and S. Zhang, "The compound structure of a new chaotic attractor," *Chaos, Solitons Fractals* **14**, 669–672 (2002).
- ³⁴G. Chen and T. Ueta, "Yet another chaotic attractor," *Int. J. Bifurcation Chaos* **9**, 1465–1466 (1999).
- ³⁵T. Matsumoto, "A chaotic attractor from Chua's circuit," *IEEE Trans. Circuits Syst.* **31**, 1055–1058 (1984).
- ³⁶L. Chua, M. Komuro, and T. Matsumoto, "The double scroll family," *IEEE Trans. Circuits Syst.* **33**, 1072–1118 (1986).
- ³⁷A. S. Elwakil and M. P. Kennedy, "Construction of classes of circuit-independent chaotic oscillators using passive-only nonlinear devices," *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.* **48**, 289–307 (2001).
- ³⁸O. E. Rössler, "An equation for continuous chaos," *Phys. Lett. A* **57**, 397–398 (1976).
- ³⁹A. M. Rucklidge, "Chaos in models of double convection," *J. Fluid Mech.* **237**, 209–229 (1992).
- ⁴⁰R. Thomas, "Deterministic chaos seen in terms of feedback circuits: Analysis, synthesis, labyrinth chaos," *Int. J. Bifurcation Chaos* **9**, 1889–1905 (1999).
- ⁴¹W. Horton, R. Weigel, and J. C. Sprott, "Chaos and the limits of predictability for the solar-wind-driven magnetosphere-ionosphere system," *Phys. Plasmas* **8**, 2946–2952 (2001).
- ⁴²G. Benettin, L. Galgani, A. Giorgilli, and J.-M. Strelcyn, "Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Part 1: Theory," *Meccanica* **15**, 9–20 (1980).