

Transcending memory barrier achieves long-term sustainable prediction of complex dynamics

Tianbao Sun

Business School, University of Shanghai for Science and Technology <https://orcid.org/0009-0008-0849-2931>

Qunxi Zhu

Research Institute of Intelligent Complex Systems and MOE Frontiers Center for Brain Science, Fudan University

Wenxin Zheng

Research Institute of Intelligent Complex Systems and MOE Frontiers Center for Brain Science, Fudan University

Changgui Gu

guchanggui@usst.edu.cn

Business School, University of Shanghai for Science and Technology

Haiying Wang

Business School, University of Shanghai for Science and Technology

Jing Wang

Business School, University of Shanghai for Science and Technology

Jack Murdoch Moore

MOE Key Laboratory of Advanced Micro-Structured Materials, and School of Physical Science and Engineering, Tongji University

Ziwei Liu

Research Institute of Intelligent Complex Systems and MOE Frontiers Center for Brain Science, Fudan University

Wei Lin

wlin@fudan.edu.cn

Research Institute of Intelligent Complex Systems and MOE Frontiers Center for Brain Science, Fudan University

Research Article

Keywords:

Posted Date: June 24th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10123570/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Transcending memory barrier achieves long-term sustainable prediction of complex dynamics

Tianbao Sun^{1,+}, Qunxi Zhu^{3,4,+}, Wenxin Zheng^{3,4}, Changgui Gu^{1,2,*}, Haiying Wang¹, Jing Wang⁵, Huijie Yang¹, Jack Murdoch Moore^{6,7}, Ziwei Liu³, and Wei Lin^{3,4,8,*}

¹Business School, University of Shanghai for Science and Technology, Shanghai 200093, China.

²School of Intelligent Emergency Management, University of Shanghai for Science and Technology, Shanghai 200093, China.

³Research Institute of Intelligent Complex Systems and MOE Frontiers Center for Brain Science, Fudan University, Shanghai 200433, China.

⁴School of Mathematical Sciences, SCMS, SCAM, and CCSB, Fudan University, Shanghai 200433, China.

⁵Institute of Photonic Chips, University of Shanghai for Science and Technology, Shanghai 200093, China.

⁶MOE Key Laboratory of Advanced Micro-Structured Materials, and School of Physical Science and Engineering, Tongji University, Shanghai 200092, China.

⁷National Key Laboratory of Autonomous Intelligent Unmanned Systems, MOE Frontiers Science Center for Intelligent Autonomous Systems, Tongji University, Shanghai 200092, China.

⁸Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.

*Corresponding author. Email: @gu.changgui@163.com; wlin@fudan.edu.cn

+These authors contributed equally to this work

ABSTRACT

Reservoir computing (RC) has emerged as a compelling framework for modeling complex dynamical systems. When applied to state-complete data generated by deterministic dynamical systems, its inherent memory dependence becomes a structural constraint. This memory redundancy confines the model to local trajectory fitting, thereby limiting its capacity for global long-term prediction. To overcome this limitation, we introduce State-Driven Reservoir Computing (SDRC), an architecture that actively decouples temporal history from state evolution. By resetting the reservoir state to block the temporal propagation of input information, SDRC collapses the functional equivalence of the model from a Volterra series to a Taylor expansion, effectively eliminating the constraints imposed by memory in classical RC. We validate SDRC across a diverse set of benchmarks, ranging from chaotic and coupled dynamical systems to real-world climate data. The framework demonstrates exceptional capability in global predictive calibration and latent variable inference while maintaining robustness against data sparsity and parameter perturbations. Compared to conventional temporal models, SDRC achieves superior predictive fidelity while requiring an order-of-magnitude reduction in both the number of parameters and training data, establishing an efficient, history-independent paradigm for intelligent dynamical modeling.

Reservoir Computing (RC) establishes a distinct methodology for training Recurrent Neural Networks (RNNs)^{1–4}. This architecture is characterized by the use of a fixed, high-dimensional nonlinear dynamical substrate known as the reservoir, where the training process is strictly limited to optimizing a linear readout layer via convex optimization^{5–8}. Empirical studies on nonlinear chaotic systems confirm the superiority of this architecture. Specifically, RC achieves state-of-the-art prediction accuracy while simultaneously circumventing the computational bottlenecks of traditional deep learning through an orders-of-magnitude reduction in trainable parameters and training time^{9,10}. By striking an exceptional balance between computational efficiency and modeling capability, RC has been widely adopted in complex scientific domains, ranging from spatiotemporal chaos prediction and fluid dynamics reconstruction to neuromorphic computing^{10–17}.

The success of RC is grounded in its universal approximation property. Theoretically, an RC system with a sufficient reservoir size is capable of approximating to arbitrary accuracy any discrete-time nonlinear dynamical system with fading memory^{18–23}. This capability derives from the geometric structure of dynamical systems, where reservoir states driven by a scalar input synchronize with the driving signal to form an embedding representation diffeomorphic to the underlying attractor. This mechanism is functionally equivalent to Takens' embedding theorem^{11,24}. Consequently, even for state-incomplete data where instantaneous observations fail to determine future evolution, RC effectively embeds the system manifold to restore determinism^{25–27}. Mathematically, this reconstruction process functions as an approximation of a discrete Volterra series that generates a rich library of high-order nonlinear memory terms^{19,21,23}. Building on this mathemati-

cal formulation, the recently proposed Next-Generation Reservoir Computing (NGRC) further streamlines the architecture by replacing the dynamical reservoir with explicit polynomials, thereby achieving leaner and more efficient prediction of chaotic attractors²⁸.

RC is also extensively applied to state-complete data, where observed variables at any instant contain full information defining the system's future evolution (corresponding to fully observed trajectories, such as the Lorenz-63 system). Significant advancements in this domain include the attainment of arbitrary prediction horizons via sparse state updates that mitigate trajectory divergence²⁹, and the demonstration of out-of-domain generalization, where multiple-trajectory training enables the reconstruction of dynamics across previously unvisited basins of attraction³⁰. Moreover, theoretical analyses regarding critical hyperparameters, including network size, spectral radius, and connectivity, focus primarily on these systems^{31–33}. From the perspective of approximation theory, the evolution of state-complete data is intrinsically a vector field function dependent solely on the current state. In contrast, standard RC approximates these dynamics using a history-dependent Volterra functional^{18,19,28}. Consequently, existing approaches inherently utilize a history-dependent architecture to model a process physically defined by a history-independent mapping.

This divergence reveals a critical efficiency bottleneck. According to the Information Processing Capacity (IPC) framework, a strict trade-off exists wherein maintaining linear memory directly consumes the computational capacity required for high-order nonlinearities^{26,33}. This implies an inherent inability to simultaneously sustain long-term memory and strong nonlinear mapping. Supporting this, Storm et al. showed that recurrent connections within the reservoir can be removed in state-complete scenarios without performance degradation²⁷. Therefore, when processing state-complete systems, the reservoir's fading memory constitutes structural redundancy, limiting the model's potential to approximate complex dynamics.

In practical applications, this structural redundancy imposes severe limitations: the reliance on history necessitates a prolonged "warm-up" period to washout transient errors, constraining immediate response capabilities^{29,34}. Furthermore, the model exhibits high sensitivity to data continuity; in the presence of missing data or non-periodic sampling, the history-dependent reservoir state fails instantly. Additionally, excessive memory predisposes the model to overfit spurious temporal correlations rather than the intrinsic physical vector field, thereby degrading robustness against parameter drift. To overcome these drawbacks from "unnecessary memory," we propose State-Driven Reservoir Computing (SDRC). By eliminating memory redundancy, SDRC reconfigures the reservoir into a state-driven, high-dimensional nonlinear kernel optimized for nonlinear dynamical systems. This architectural alignment fundamentally improves the modeling of complex vector fields, thereby achieving superior predictive performance compared to memory-constrained baselines.

Results

A framework of SDRC

The SDRC framework is conceptually illustrated with the Lorenz-63 system (see Fig. 1). Consider $\mathbf{v}(t)$, an time series observed with sampling period Δt (see Fig. 1(a)). Both the classical RC (see Fig. 1(b)) and the proposed SDRC (see Fig. 1(c)) project the data into a high-dimensional space followed by an output layer that performs a linear transformation on the feature vectors. The fundamental distinction lies in the architectural paradigm: while the classical RC employs history-dependent mechanisms, the SDRC framework implements a history-independent encoding principle. More detailed differences are systematically examined below.

For a classical RC, the dynamics of the nodes are based on leaky integrate neurons and basic sigmoid activation functions. Notably, each node retains memory information from previous time steps, known as the hidden state of the reservoir⁶. The dynamics of the hidden state is described by:

$$\mathbf{r}(t + \Delta t) = (1 - \alpha) \cdot \mathbf{r}(t) + \alpha \cdot \tanh[\mathbf{W}_{\text{in}}\mathbf{v}(t) + \mathbf{W}_{\text{R}}\mathbf{r}(t)], \quad (1)$$

where α is the leakage rate, $\mathbf{r}(t)$ represents the hidden state in the reservoir at time t , $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{R}}$ are, respectively, the input and the weighted adjacency matrices. Typically, these two matrices are chosen randomly and kept fixed in both training and test stages. Then, it follows an output layer as a linear transformation of the feature vector:

$$\hat{\mathbf{v}}(t + \Delta t) = \mathbf{W}_{\text{out}}\mathbf{R}(t), \quad (2)$$

where the feature vector is set as $\mathbf{R}(t) = [b_{\text{out}}, \mathbf{v}^\top(t), \mathbf{r}^\top(t)]^\top$ with b_{out} as a bias.

Differences in input layers. In a classical RC, the random matrices $\mathbf{W}_{\text{in}}$, $\mathbf{W}_{\text{R}}$ must be specified to achieve feature dimensionality expansion of the input data. This introduces some instability to the model, and currently, there is no uniformly reliable guideline for setting the random matrix⁴. Recent studies have firmly established the theoretical link between RC and nonlinear vector autoregressive models^{18,28}. Specifically, it has been demonstrated that a linear RC is structurally equivalent to a model operating on polynomial feature vectors of arbitrary order constructed from the input history. While the theoretical framework supports arbitrary degrees, practical implementations often prioritize quadratic terms to balance expressivity with computational efficiency.

Building on this general theoretical foundation, the NGRC framework replaces the reservoir with explicit polynomial feature vectors typically constructed from delayed inputs²⁸. Inspired by this explicit feature construction, our SDRC framework also enhances data dimensionality through polynomial function construction. A fundamental distinction between SDRC and NGRC lies in the handling of temporal dependency. For state-complete datasets, temporal delay embedding is theoretically redundant. Instead, a feature vector composed of polynomials (up to a sufficient order) derived strictly from instantaneous states is adequate to replace the random matrix.

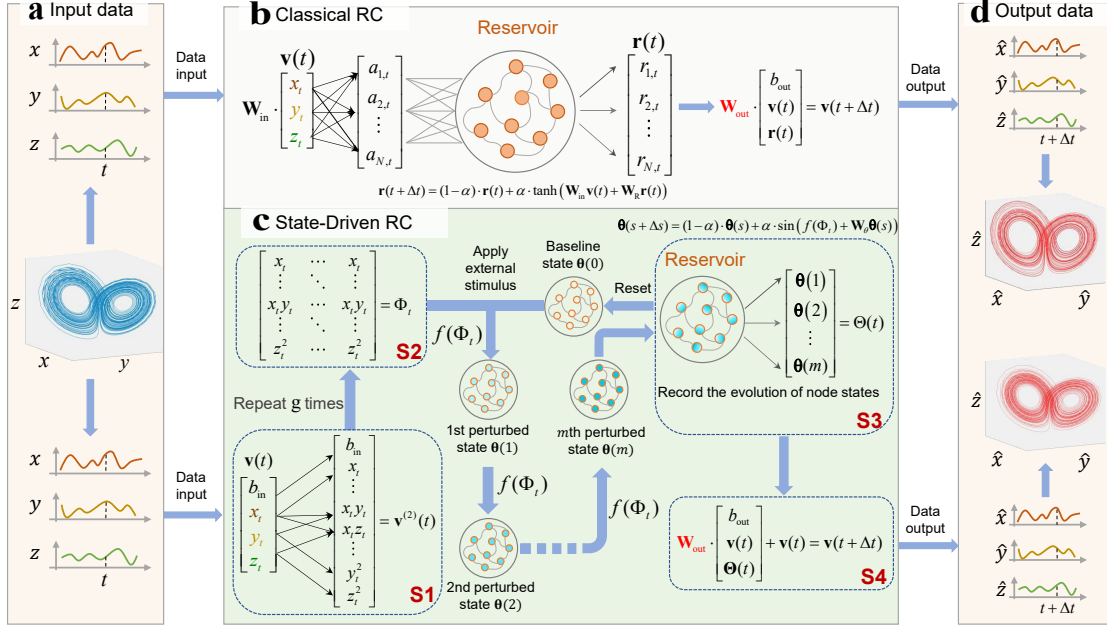


Figure 1. Schematic diagrams for illustrating the proposed SDRC framework. **a** The data simulated by the Lorenz63 system as the input for different frameworks of RC. **b** The classical RC framework. **c** The SDRC framework with four modules from “S1” to “S4”. “S1” follows the NGRC methodology. **d** The output data predicted by the model. Here, using the data simulated by the Lorenz63 system as an example (blue, left), the SDRC employs a quadratic nonlinear function to derive the monomial feature vector of its time series data. This vector is then broadcast and introduced into the reservoir as external perturbations. The predicted data (red, right) is a combination of three parts, specifically, the reservoir’s evolving states, the input values, and the bias term.

For a clearer illustration, we take the Lorenz-63 system³⁵ as an example. Our SDRC constructs a quadratic nonlinear feature vector for the d -dimensional input $\mathbf{v}(t)$, which is directly fed into Eqs. (1)-(2) of the classical RC. First, we compute the outer product of $[b_{\text{in}}; \mathbf{v}(t)]$, where b_{in} represents the input bias, resulting in a symmetric matrix with $(d+1)^2$ elements. Next, we use an operator denoted $[\otimes]$ defined in NGRC²⁸ to collect all unique polynomials from the matrix, yielding a second-order nonlinear feature vector $\mathbf{v}^{(2)}(t)$ with $(d+1)(d+2)/2$ elements (see module S1 of Fig. 1(c)). Here, a polynomial feature vector of order p is defined as:

$$\mathbf{v}^{(p)}(t) = \mathbf{v}(t) [\otimes] \mathbf{v}(t) [\otimes] \cdots [\otimes] \mathbf{v}(t), \quad (3)$$

where $\mathbf{v}(t)$ appears p times on the right-hand side.

In module S2 of Fig. 1(c), we illustrate the *replicate* operation of the feature vectors. The D -dimensional (b_{in} excluded) p -order feature vector from the input layer is replicated g times to form a matrix Φ_t containing $g \times D$ elements. To enhance the model's generalization capability, we introduce a bias node in the reservoir by adding 1 to the elements of matrix Φ_t (denoted by $N = g \times D + 1$). Thus, the resulting number of elements in Φ_t corresponds to the configured reservoir nodes, where each node can receive external stimuli, minimize information transmission time within the network to the greatest extent, and enhance information coupling.

The dimensionality expansion methodology, described above, can be generalized to high-dimensional coupled complex systems. Let $\mathcal{G}$ be a complex system composed of n mutually interacting subsystems, denoted by $\mathbf{u} = \{u_1, u_2, \dots, u_n\}$. For each subsystem $u_i \in \mathbf{u}$, characterized by d internal variables, its state vector is defined as $\mathbf{v}_i(t) = [v_{i1}(t), \dots, v_{id}(t)]^\top \in \mathbb{R}^d$ for $i = 1, \dots, n$. Thus, the collective state of the system, given by $\mathbf{v}(t) = [\mathbf{v}_1^\top(t), \dots, \mathbf{v}_n^\top(t)]^\top$, generates the expanded feature vector $\mathbf{v}_{\mathcal{G}}(t) = [\mathbf{v}_1^{(p)\top}(t), \dots, \mathbf{v}_n^{(p)\top}(t)]^\top$ through the input layer dimension expansion, where $\mathbf{v}^{(p)}$ is defined in Eq. (3). Notably, the SDRC framework can be implemented in parallel by assigning each subsystem an independent model. Specifically for each subsystem, the feature vector:

$$\mathbf{v}_{u_i}(t) = [\dots, \omega_{ij} \mathbf{v}_i^{(p)\top}(t), \dots]^\top, \quad i, j = 1, \dots, n, \quad (4)$$

where each $\omega_{ij} \in \mathbb{R}^+$ quantifies the coupling weight between subsystems u_j and u_i . We treat the coupling strength as a structural hyperparameter. Its optimal value is identified via grid search on a held-out validation set, minimizing prediction error to dynamically balance subsystem independence with synchronization requirements. A combination of the feature vectors in Eq. (4) emerges as a weighted global feature vector. Continuing the broadcast operation, we can obtain subsystem u_i 's broadcast matrix $\Phi_{t,i}$, comprising $N = n \times g \times D + 1$ elements.

In parallel reservoir computing (PRC), each sub-reservoir receives the full input $\mathbf{v}_i$ associated with a subsystem u_i , establishing a direct one-to-one correspondence between the

number of sub-reservoirs and subsystems. While these units process data autonomously, coupling interactions are synthesized via the output layers, where the prediction for each subsystem aggregates reservoir states from all parallel units. The architectural layout of PRC is detailed in Supplementary Note 1.

Differences in reservoir layers. In our SDRC, the reservoir is also a network consisting of N nodes with the modified dynamics. Different from the classical RC, our SDRC, as seen in module S3 of Fig. 1(c), utilizes high-dimensional states generated by a single external input at time t to continuously perturb these nodes, generating transient disturbance responses that serve as the hidden states of the reservoir. Such a configuration, using only a few dozen nodes, enables our SDRC to produce dynamic information comparable to that of the classical RC using hundreds of nodes. As a result, our SDRC does not rely on a specific topological structure. Moreover, unlike NGRC which replaces the dynamical substrate with static polynomial features, SDRC preserves the reservoir to function as a high-dimensional transient kernel. The transient dynamics of SDRC are thus designed as:

$$\theta(s + \Delta s) = (1 - \alpha) \cdot \theta(s) + \alpha \cdot \sin[f(\Phi_t) + \mathbf{W}_\theta \theta(s)], \quad (5)$$

where $\theta(s)$ is the node state of the SDRC reservoir at the transient iteration time s ($= 1, \dots, m$) of the global time t , $\mathbf{W}_\theta$ is the adjacency matrix with all entries equal to 1, and $\Delta s = 1$ is the transient time increment for the inner iteration process, and $f(\Phi_t)$ denotes the activation of the above-defined broadcast matrix Φ_t at time t through the function $\sin(\cdot)$, followed by a column-wise reshaping of its elements into a column vector. As for the high-dimensional system $\mathcal{G}$, the reservoir node state $\theta_i(s)$ of the subsystem u_i is also obtained by inputting the broadcast matrix $\Phi_{t,i}$. Here, we set the sinusoidal function as the activation function, because a non-monotonic function has been demonstrated to have chaos dynamics and a stronger memory capacity in artificial neural networks³⁶. In Supplementary Note 2, we also use the Kuramoto oscillator network model as reservoir dynamics to demonstrate the reliability of the network's transient disturbance process as an information encoding mechanism.

After collecting the transient perturbed states corresponding to each input data, the node states need to be reset. Before model training, a baseline state (the initial state of the nodes before an external stimulus) is randomly assigned to the system nodes (as tested in Supplementary Note 3, we find that a random selection of the baseline does not affect the model's predictive performance). After collecting the hidden state for the current global time step t , the reservoir is reset to the baseline state to ensure that the state changes caused by each input perturbation begin from the same reference point. In each transient, the network is iterated m times according to Eq. (5), and the states of all nodes are recorded, as $\theta(s)$ with $s = 1, \dots, m$. Then, the recorded node states are used to construct the hidden state vector $\Theta(t) = [\theta^\top(1), \dots, \theta^\top(m)]^\top$. Finally, together with the hidden state vector $\Theta(t)$, the cur-

rent input $\mathbf{v}(t)$, is also fed into the output layer to enrich the system's dynamic information, and an additional bias term is introduced to enhance the model's generalization ability. During this dynamical process, memory accumulates solely through the transient perturbation following uniform input data, and the information between consecutive input data remains independent, not contributing to the state update at the next global time step.

To summarize, we formulate how the information in the output layer is mapped to the target state space through the output weights $\mathbf{W}_{\text{out}}$ as follows:

$$\hat{\mathbf{v}}(t + \Delta t) = \mathbf{v}(t) + \mathbf{W}_{\text{out}} \mathbf{R}(t), \quad (6)$$

where, as depicted in module S4 of Fig. 1(c), $\hat{\mathbf{v}}(t + \Delta t)$ is the predicted value at time $t + \Delta t$, $\mathbf{R}(t) = [b_{\text{out}}^\top, \mathbf{v}(t)^\top, \Theta^\top(t)]^\top$ is the composite feature vector received by the output layer, and b_{out} is the output layer's bias. The term $\mathbf{W}_{\text{out}} \mathbf{R}(t)$ can be interpreted as the residual between $\mathbf{v}(t + \Delta t)$ and $\mathbf{v}(t)$, effectively representing the discretized integration step. This precise mapping capability is reminiscent of the precise spike timing observed in sensory processing³⁷, ensuring accurate reconstruction of the vector field.

Within the parallel architecture of SDRC, the mapping relationship of each subsystem u_i is governed by its corresponding output weight matrix $\mathbf{W}_{\text{out},i}$. Similarly to Eq. (6), the output dynamics are described by:

$$\hat{\mathbf{v}}_i(t + \Delta t) = \mathbf{v}_i(t) + \mathbf{W}_{\text{out},i} \mathbf{R}_i(t), \quad (7)$$

where $\mathbf{R}_i(t) = [b_{\text{out},i}^\top, \mathbf{v}_i(t)^\top, \Theta_i^\top(t)]^\top$ and $\Theta_i(t) = [\theta_i^\top(1), \dots, \theta_i^\top(m)]^\top$ is the reservoir hidden state vector of u_i .

Differences in approximation. To understand the representational bias of SDRC compared to classical approaches, we examine its approximation properties using operator theory. It is well-established that standard reservoir computing, operating as a time-invariant fading-memory filter, functions as a universal approximator for discrete-time Volterra series¹⁸. Our analysis demonstrates that the state-resetting mechanism in SDRC fundamentally alters this representation, effectively imposing a sparsity constraint that leads to a collapse of the general Volterra functional into a multivariate Taylor expansion of the instantaneous input.

Consider the reservoir as an operator H mapping inputs $\mathbf{v}_t \in \mathbb{R}^d$ to outputs $\mathbf{y}_t$ (where $\mathbf{y}_t$ targets the future state $\mathbf{v}_{t+1}$). Under the standard fading memory assumption, the Boyd-Chua theorem implies that this input-output relationship admits a discrete Volterra series representation¹⁹:

$$\mathbf{y}_t = \mathbf{h}_0 + \sum_{k=1}^{\infty} \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_k=0}^{\infty} \mathbf{H}_k(\tau_1, \dots, \tau_k) \cdot (\mathbf{v}_{t-\tau_1} \otimes \dots \otimes \mathbf{v}_{t-\tau_k}), \quad (8)$$

where $\mathbf{h}_0$ is the bias, $\mathbf{H}_k$ represents the k -th order Volterra kernel tensor, and $\tau_i \in \mathbb{Z}_{\geq 0}$ denotes the time lag. In this general form, the output $\mathbf{y}_t$ is computed via a convolutional integration of the entire input history trajectory.

In the proposed SDRC architecture, however, the reservoir state $\theta(s)$ is explicitly reset to a baseline $\theta(0)$ at each time step t . This mechanism decouples the transient state evolution from previous time steps, ensuring that the hidden state depends exclusively on the current input $\mathbf{v}_t$. Mathematically, this enforces a strict **history-independence constraint**, rendering the output $\mathbf{y}_t$ conditionally independent of any historical inputs $\mathbf{v}_{t-\tau}$ for $\tau \geq 1$. This independence forces the Volterra kernels associated with non-zero lags to vanish. Specifically, for any kernel tensor $\mathbf{H}_k$, we have:

$$\mathbf{H}_k(\tau_1, \dots, \tau_k) = \mathbf{0}, \quad \text{if } \exists i, \tau_i \neq 0. \quad (9)$$

Applying this constraint to Eq. (8) eliminates all cross-temporal terms. Consequently, the infinite nested summation over time lags collapses to a single term at the current time index ($\tau_1 = \dots = \tau_k = 0$), simplifying the series to:

$$\mathbf{y}_t = \mathbf{h}_0 + \sum_{k=1}^{\infty} \mathbf{H}_k(0, \dots, 0) \cdot (\mathbf{v}_t \otimes \mathbf{v}_t \otimes \dots \otimes \mathbf{v}_t). \quad (10)$$

By defining the instantaneous coefficient tensor as $\mathbf{C}_k = \mathbf{H}_k(0, \dots, 0)$ and letting $\mathbf{C}_0 = \mathbf{h}_0$, the mapping degenerates into a power series of the instantaneous input:

$$\mathbf{y}_t = \mathbf{C}_0 + \sum_{k=1}^{\infty} \mathbf{C}_k \cdot (\mathbf{v}_t)^{\otimes k}. \quad (11)$$

Eq. (11) is structurally identical to the multivariate Taylor series expansion of a nonlinear function $\mathbf{F}(\mathbf{v}_t)$. This derivation highlights a critical theoretical divergence: while classical RC approximates a functional of the input trajectory (history-dependent)^{18,21}, SDRC strictly approximates an instantaneous function of the vector field (history-independent). The reservoir in SDRC thus functions as a high-dimensional kernel machine that constructs the higher-order Taylor features required to approximate the governing nonlinear dynamics $\mathbf{v}_{t+1} = \mathbf{F}(\mathbf{v}_t)$.

Training method. Similarly to classical RC, our SDRC employs the ridge regression to train the output weights $\mathbf{W}_{\text{out}}$. Ridge regression addresses the multicollinearity problem by introducing an L_2 regularization term into the traditional linear regression. In the entire reservoir, the output weight $\mathbf{W}_{\text{out}}$ is the only parameter that needs to be trained. Its analytic solution in a matrix form is given by:

$$\mathbf{W}_{\text{out}} = \mathbf{U} \mathcal{R}^\top (\mathcal{R} \mathcal{R}^\top + \lambda \mathbf{I})^{-1}, \quad (12)$$

where the k -th column of $\mathcal{R}$ (resp., $\mathbf{U}$) is given by $\mathbf{R}(k\Delta t)$ (resp., $\mathbf{v}((k+1)\Delta t) - \mathbf{v}(k\Delta t)$), λ is the regularization parameter used to prevent overfitting, and $\mathbf{I}$ denotes the identity matrix. For the parallel implementation, once obtaining the $\mathbf{R}_i$ and $\mathbf{v}_i$, one can estimate $\mathbf{W}_{\text{out},i}$ as well.

Performances in representative dynamics

Description of input data. To evaluate the performance of the SDRC framework, we consider three tasks. The first

task examines the predictive performance of SDRC in low-dimensional state-complete and state-incomplete datasets in comparison to traditional temporal models; the second task assesses the model's global prediction correction capability; and the third task tests its robustness.

The experimental data required for Task 1 are generated by numerically integrating the Lorenz-63 system and the Mackey–Glass equation. The Lorenz-63 system is described as follows³⁵:

$$\begin{aligned}\dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z), \\ \dot{z} &= xy - \beta z,\end{aligned}\quad (13)$$

where the parameters $(\sigma, \rho, \beta) = (10, 28, 8/3)$ are chosen to induce chaotic behavior in the system.

The Mackey–Glass equation describes an infinite-dimensional chaotic system. It is defined by the following time-delay differential equation³⁸:

$$\dot{x} = \beta \frac{x(t - \tau)}{1 + [x(t - \tau)]^n} - \gamma x(t), \quad (14)$$

where the parameters $(\beta, \gamma, n, \tau) = (0.2, 0.1, 10, 17)$ are chosen to induce chaotic behavior in the system.

The experimental data required for Task 2 are numerically generated by integrating a 15-D coupled Lorenz-63 (CL63) system³⁹. Each subsystem is described by the following set of ordinary differential equations (ODEs):

$$\begin{aligned}\dot{x}_i &= -\sigma \left[x_i - y_i + \gamma \sum_{j=1}^n k_{ij} g_{ij}(x_i, x_j) \right], \\ \dot{y}_i &= x_i(\rho - z_i) - y_i, \\ \dot{z}_i &= x_i y_i - \beta z_i,\end{aligned}\quad (15)$$

where n is the number of subsystems, $\gamma = 0.5$ represents the coupling strength, k_{ij} indicates the coupling weight, and $g_{ij} = (x_i - x_j)$ represents the coupling function. We consider the CL63 system in Eq. (15) with 5 subsystems, making it 15-dimensional. Structure and coupling weights are depicted in Fig. 3(a).

To provide an intuitive demonstration of the impact of memory effects on model performance, we further investigate the coupled FitzHugh–Nagumo system (FHNS) and Chen system (ChS)⁴⁰. The FHNS is employed to describe a collection of interacting neurons, and is characterized by the following set of ODEs:

$$\begin{aligned}\dot{x}_i &= x_i(1 - x_i^2) - y_i + \gamma \sum_{j=1}^n w_{ij}(x_i - y_j), \\ \dot{y}_i &= a + bx_i + cy_i.\end{aligned}\quad (16)$$

The experimental data in Fig. 4(a) is generated by setting $\gamma = 0.5$, $a = 0.28$, $b = 0.5$, $c = -0.04$, and $n = 5$ in system Eq. (16). These parameter values are in accordance with the unified dynamics of the FHNS as defined by this equation. They ensure an appropriate equilibrium between the

nonlinear self-dynamics of the system and moderate coupling effects, thereby preventing the dominance of synchronization phenomena.

The coupled ChS of 15-dimension exhibits more complex dynamical behavior, encompassing double-scroll structures and a wealth of chaotic characteristics. The standard coupled form of its ODEs is given by:

$$\begin{aligned}\dot{x}_i &= a(y_i - x_i) + k(x_{i+1} - x_i), \\ \dot{y}_i &= (c - a)x_i - x_i z_i + cy_i + k(y_{i+1} - y_i), \\ \dot{z}_i &= x_i y_i - bz_i + k(z_{i+1} - z_i),\end{aligned}\quad (17)$$

where the parameters $(a, b, c) = (35, 3, 28)$ govern the inherent chaotic dynamics of the ChS. The coupling strength, specified as $k = 0.1$, enables interactions among five adjacent units. As illustrated in Fig. 4(e), the ChS generates complex and highly chaotic trajectories. Additional experiments involving other systems are elaborated upon in Supplementary Note 2.

Real-world validation employed the Jena Climate Dataset from the Max Planck Institute for Biogeochemistry. We analyzed a 14-dimensional multivariate time series of meteorological variables, including temperature, pressure, and humidity, sampled at 10-minute intervals. Spanning the full annual cycle from January 1, 2024, to January 1, 2025, the dataset comprises 52,704 temporal samples per variable.

To evaluate the model's prediction performance, we use the Lyapunov time as the effective prediction time of the SDRC framework and employ the Root Mean Square Error (RMSE):

$$\text{RMSE}(t) = \frac{1}{n} \sum_{i=1}^n \sqrt{\frac{1}{d} \sum_{j=1}^d \left[\frac{\hat{v}_{ij}(t) - v_{ij}(t)}{\sigma_{ij}} \right]^2}, \quad (18)$$

as the evaluation metric for the prediction error. Here, $v_{ij}(t)$ (resp., $\hat{v}_{ij}(t)$) denotes the j -th component of $\mathbf{v}_i(t)$ (resp., $\hat{\mathbf{v}}_i(t)$), and σ_{ij} is the standard deviation of the i -th component of $\mathbf{v}_i(t)$. For a more lucid illustration, we define the valid predictive steps (VPS) as the maximal number of consecutive time steps during which the prediction accuracy remains above a pre-defined threshold. Unless otherwise explicitly stated in this work, the default value of this threshold is set as 0.1.

Task 1 Comparative analysis on nonlinear dynamics with and without state completeness. In Task 1, we address two primary research objectives: (1) to examine the performance differences between SDRC and temporal models in predicting fully observed nonlinear dynamical systems, which inherently possess state completeness; and (2) to explore approaches for extending SDRC to systems without state-completeness. The Lorenz-63 system and the Mackey–Glass equation are selected as representative benchmarks for these respective regimes. All experimental results reported are averages over 10 independent runs to ensure statistical reliability. The comparative models comprise NGRC, RC, LSTM, GRU, and Transformer (see Supplementary Note 4 for detailed configurations). To isolate the contribution of each SDRC module to prediction performance, we conducted an ablation study

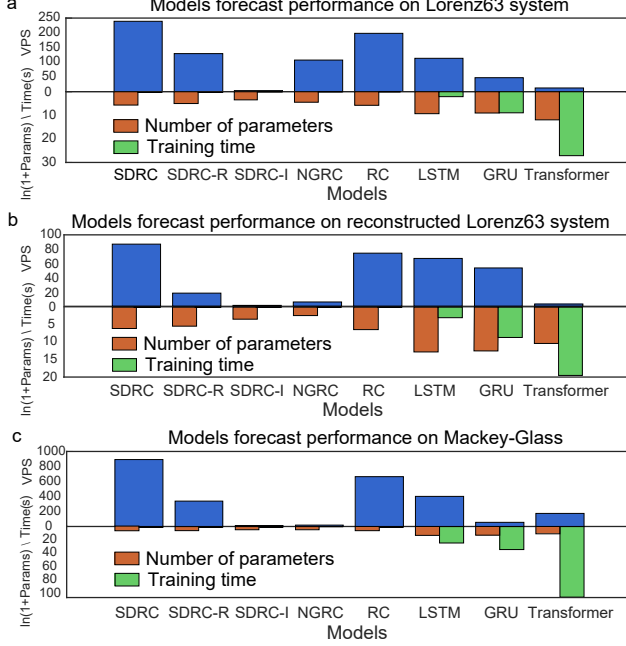


Figure 2. Forecasting performance and efficiency of different models on chaotic systems. **a** Performance on the fully observed Lorenz-63 system Eq. (13). **b** Performance on the reconstructed Lorenz-63 system (using observation of only the x -component). **c** Performance on the time-delay embedded Mackey-Glass system Eq. (14). For each model, the effective prediction time (blue), the number of trainable parameters (orange), and the training time (green) are shown. SDRC denotes the proposed model, which is compared against its ablated variants (SDRC-R, SDRC-I) and several established architectures (NGRC, RC, LSTM, GRU, Transformer).

using two variants: SDRC-R, which retains only the reservoir structure without the polynomial transformation in the input layer, and SDRC-I, which preserves only the input-layer polynomial transformation without the reservoir.

We first evaluate the model on the fully observed Lorenz-63 system Eq.(13) (the training set size is 400), which inherently satisfies state-completeness. The input layer constructs a monomial feature vector of dimension 9×1 (up to second-order). After applying the broadcasting operation with $g = 2$, the size of Φ_t is expanded to 9×2 . Following external stimulation, the nodes iterate with a step size of $\Delta s = 1$. The states of the nodes over the subsequent five iterations are used as the hidden state $\Theta(t)$, corresponding to 90 virtual oscillators in the reservoir. Combined with the current input data and the bias introduced in the output layer, the output weight matrix $\mathbf{W}_{\text{out}}$ has a dimension of 3×94 .

As illustrated in Fig. 2(a), SDRC achieves superior predictive fidelity compared to classical RC while utilizing only 23.8% of the training data and 18% of the reservoir nodes. Ablation studies clarify the underlying mechanism by demonstrating that the standalone polynomial input layer (SDRC-I), structurally analogous to NGRC, exhibits limited capability in modeling complex chaotic attractors. This observation suggests that the polynomial layer provides a coarse-grained approximation of the vector field, whereas the state-driven reservoir functions as a high-dimensional nonlinear kernel designed to capture high-frequency residuals and fine-grained topological features. Through this integration, SDRC outperforms both pure polynomial regression and classical memory-based RC, where augmenting the reservoir baseline (SDRC-R) with the SDRC-I module improves prediction accuracy by over 80%. Furthermore, the elimination of reliance on historical inputs allows this architecture to mitigate recursive error accumulation and facilitate long-term chaotic forecasting, a capability examined in detail in Task 2.

Distinct from the fully observed regime, the isolated observation of the Lorenz-63 x -component exhibits induced state incompleteness, where the scalar time series fails to define a unique future evolution due to the occlusion of latent variables (y, z). To align such dynamics with the SDRC architecture, we employ phase-space reconstruction to recover the attractor topology from the scalar proxy. Specifically, utilizing a limited training set of 400, we lift the scalar series $x(t)$ into a 3-dimensional state vector $\mathbf{v}_t = [x(t), x(t - \tau), x(t - 2\tau)]^\top$ with an embedding delay of $\tau = 6$. While this topological unfolding theoretically restores state completeness, it transforms the functional mapping from a simple polynomial in the original coordinates to a highly nonlinear, non-polynomial function in the delay coordinates. As illustrated in Fig. 2(b), this complexity leads to a universal performance degradation across all models; nevertheless, SDRC maintains its superiority.

Validation was subsequently extended to the Mackey-Glass system defined in Eq. (14) with a training set of 3000 samples, which exhibits intrinsic state incompleteness arising from in-

herent time delays. We employed Takens' embedding with $\delta = 17$ and $D = 5$ to explicitly unfold the infinite-dimensional history into a state-complete spatial vector. Figure 2(c) illustrates that SDRC achieves superior predictive accuracy on this reconstructed manifold. This result suggests that explicitly restoring state completeness via embedding, rather than relying on the implicit memory of RNNs, enables the efficient application of the SDRC framework to diverse dynamics characterized by state incompleteness. Consequently, subsequent analyses concentrate primarily on systems exhibiting state completeness.

Task 2.1 Global prediction calibration test using coupled high-dimensional dynamical systems. Beyond eliminating structural redundancy, the advantages of modeling based on state completeness are further reflected in enhanced predictive flexibility and robustness. In Task 2.1, we have innovatively introduced the concept of global prediction calibration. This refers to a process in which, once prediction accuracy exceeds a threshold starting from a certain time point t_a , the predicted value at any subsequent time point t_p (where $t_p \geq t_a$) is calibrated using the true value. Starting from t_p , the model's predictions return to within the accuracy threshold and continue to operate for a period of time. This calibration capability is effective globally across the target system, indicating that prediction and calibration can still be achieved even for time points prior to the training set.

For the prediction of coupled chaotic systems, all models are implemented using parallel architectures. To ensure a fair comparison, LSTM, GRU, and Transformer models are selected following the node-level modeling approach described in reference [10]. In these models, variables directly involved in subsystem computations are provided as inputs to their respective submodels to maximize predictive performance. In contrast, SDRC constructs its model without any prior knowledge of the coupling relationships among variables.

In the CL63 system Eq. (15), chaotic data are generated with a time interval of $\Delta t = 0.01$. The dimensionality of the unique monomial feature vector of the subsystem u_i is 9×1 (up to the second-order). Upon performing the broadcast operation with $g = 2$, the size of $\Phi_{t,i}$ is transformed to 9×2 . Following the introduction of the bias node, the number of reservoir nodes n , which is determined by the five subsystems, is set as 91. After the nodes receive external stimuli, they iterate with a step size of $\Delta s = 1$. The states of the nodes over the next 5 iterations are used as the hidden state $\Theta(t)$, corresponding to 455 virtual oscillators in the reservoir. With the current input data and the bias introduced into the output layer, the output weight matrix $\mathbf{W}_{\text{out}}$ has dimensions 15×459 .

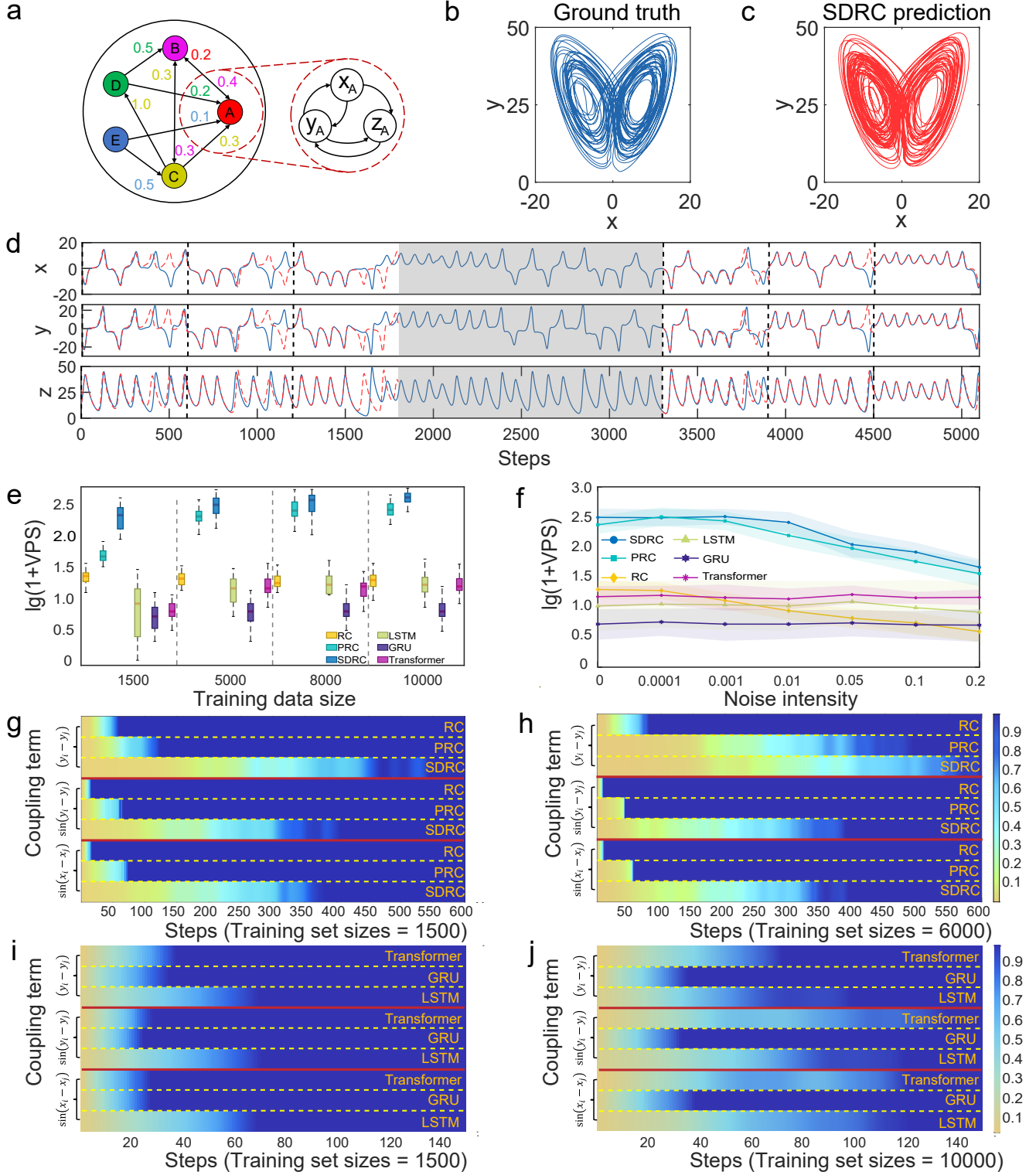
As shown in Figs. 3(b) & 3(c), the SDRC framework achieves high-fidelity reproduction of the subsystem A attractor topology. During the prediction phase (see Fig. 3(d)), our SDRC demonstrates a global prediction capability that no traditional temporal models can achieve. Specifically, after training, our SDRC can start the prediction task from any position in the target state space, whereas the temporal models,

including the classical RC, can only begin the prediction from the next time step adjacent to the training set. This also indicates that our SDRC possesses global calibration ability. As the prediction accuracy exceeds the threshold at any time step, the output can be corrected using the corresponding real value at that time (as shown by the black dashed line in Fig. 3), ensuring that subsequent predictions remain within the required accuracy threshold for a period of time. As detailed in Supplementary Note 5, our analysis of weight value distributions revealed pseudo-periodic patterns. Notably, weights with larger magnitudes exhibited strong correlations with higher-order terms in the dynamical equations, a phenomenon that may provide mechanistic insights into the robustness of SDRC.

We select 30 random initial points from the test dataset for multi-step prediction and evaluate the predictive performance of each method based on its effective prediction horizon. As illustrated in Fig. 3(e), which presents statistical results of predictable steps for each method under varying training set sizes, our SDRC framework demonstrates superior performance relative to both benchmark approaches. This advantage persists even when our SDRC utilizes an order of magnitude fewer computational nodes than conventional methods, particularly pronounced in small-sample training scenarios. In Supplementary Note 6, we provide the predicted results of the remaining CL63 subsystems.

Noise robustness was evaluated by adding zero-mean Gaussian white noise with variance σ_n to the training dataset ($N = 6000$), where results presented in Fig. 3(f) demonstrate that SDRC sustains predictive fidelity even under stochastic perturbations. The analysis was subsequently extended to generalized coupling dynamics characterized by sinusoidal and linear interaction terms, including $\sin(x_i - x_j)$ and $(y_i - y_j)$. As illustrated in Figs. 3(g)-3(j), SDRC maintains a distinct performance advantage over baselines in data-scarce regimes ($N = 1,500$) comparable to that observed in data-rich settings ($N = 6,000$). Conversely, conventional LSTM, GRU, and Transformer architectures exhibit marked degradation relative to RC-based approaches, a divergence that emphasizes the superior data efficiency of the proposed framework. Moreover, although recent methodologies have integrated attention mechanisms into reservoir computing to enhance performance⁴², current findings suggest that for systems with full state observability, structural simplification achieved by explicitly removing memory yields comparable or superior computational efficiency by aligning the model architecture with the intrinsic state completeness of the system.

To elucidate the challenges posed by memory dependence in dynamics characterized by state completeness, we conduct experiments on the datasets of the coupled FHNS and the coupled ChS. The prediction process is categorized into four stages: (1) No Prediction Stage (NPS): The model does not perform any predictions but directly forecasts the next time step adjacent to the current stage; (2) Accurate Prediction Stage (APS): Predictions within our predefined error threshold; (3) Inaccurate Prediction Stage (IPS): Predictions ex-



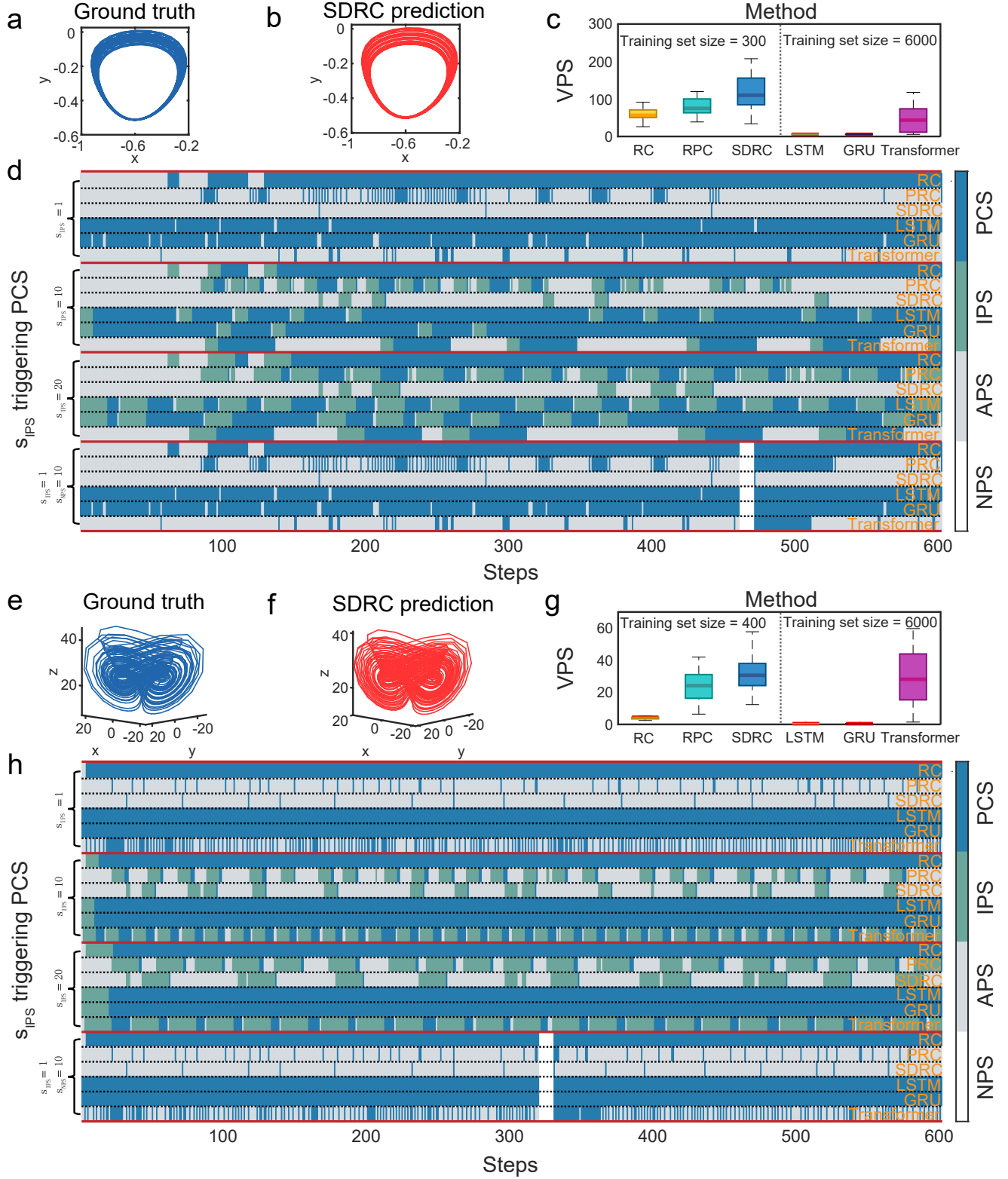


Figure 4. Investigating the limitations of model memory using coupled dynamic systems. Prediction results for coupled FHNS in Eq. (16) (a-d) and coupled ChS in Eq. (17) (e-h) attractor trajectories using SDRC and comparison models. Here in c and g, the line in the middle of the box represents the median of the VPS, and the upper and lower boundaries of the box represent the upper and lower quartiles, respectively. Furthermore, the boundaries of the upper and lower whiskers represent the maxima and minima, respectively. d, h The prediction processes of different models under various conditions that trigger the calibration operation. The training set sizes for the experimental data of these systems are set to 300 and 400, with Δt values of 0.25 and 0.02, respectively.

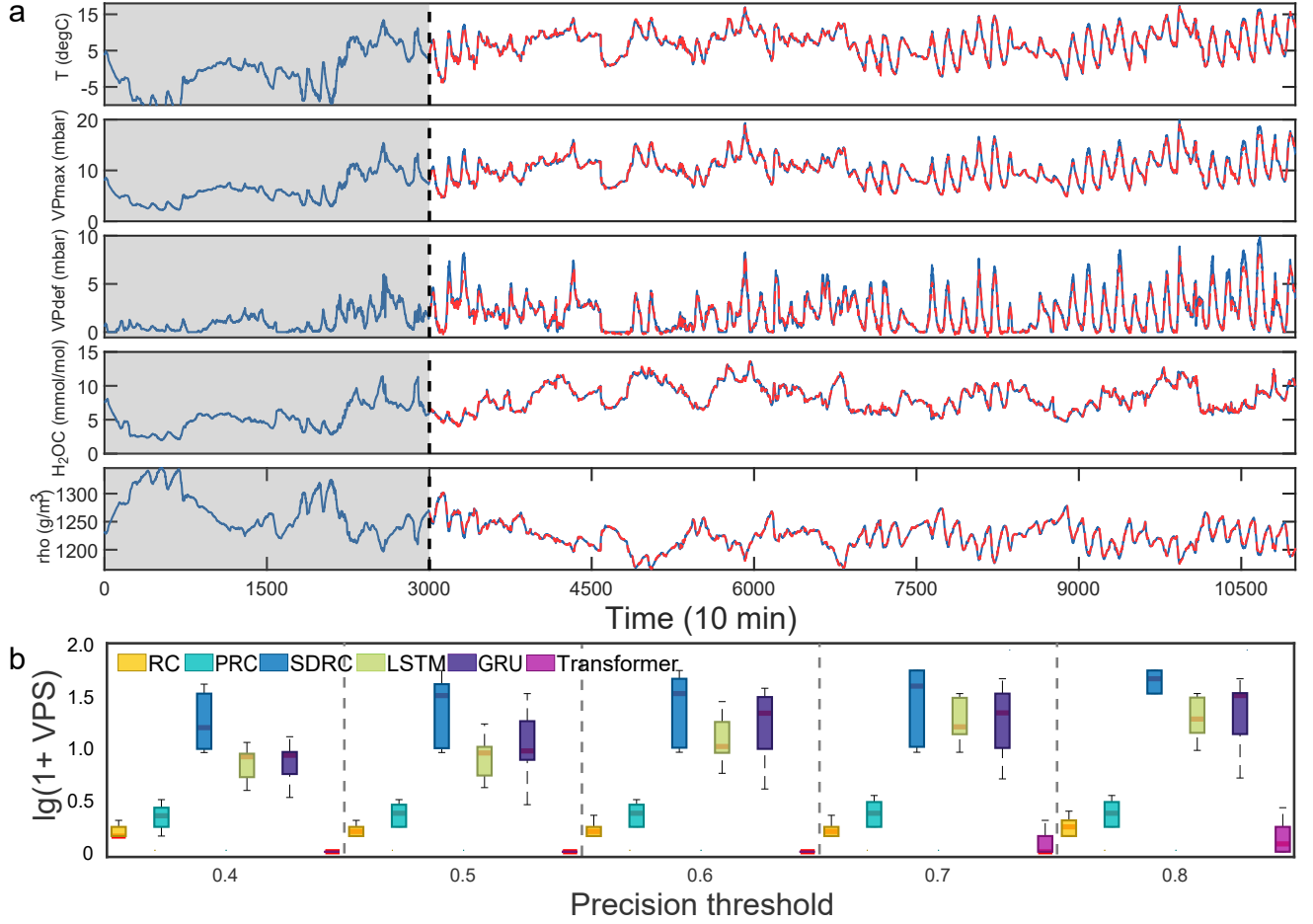


Figure 5. Inferring missing variables in the real climate system using SDRC. **a** Randomly selected five missing variables. The data in the gray region represents the true values (blue) from the training set, while the data outside the gray region are the true values (blue) and the predicted values (red) from the test set. **b** The VPS of different method under different threshold accuracy is calculated using the Jena climate dataset. Here, the line in the middle of the box represents the median, and the upper and the lower boundaries of the box represent the upper and the lower quartiles, respectively. Moreover, the boundaries of the upper and the lower whiskers represent the maxima and the minima, respectively.

ceeding error thresholds; and (4) Prediction Correction Stage (PCS).

Figures 4(a)-4(d) and 4(e)-4(h) present results for the datasets of the coupled FHNS and the coupled ChS, respectively. As shown in Figs. 4(d) & 4(h), calibration triggers are activated when IPS steps exceed the s_{IPS} threshold. By controlling NPS and IPS step sizes (abbreviated as s_{NPS} and s_{IPS} , respectively), we regulated the error accumulation in memory units and observe the memory effects on model performance.

Increasing IPS duration correlates with declining prediction accuracy in RC, PRC, and Transformer, accompanied by extended calibration requirements and reduced APS duration post-calibration. This demonstrates how memory error accumulation degrades model performance. It is worth noting that, in the Transformer architecture, which is based on a self-attention mechanism, the range of error accumulation is constrained by the size of the constructed look-back window. The predictive capabilities of LSTM and GRU models in the tested coupled systems are nearly exhausted. Furthermore, NPS execution causes original APS steps to transition to PCS (evident in control groups with $s_{IPS} = 1$ and $s_{IPS} = s_{NPS}^{-1} = 10$), indicating that disrupted memory architectures induce memory confusion requiring prolonged calibration to restore APS functionality. In contrast, our SDRC exhibits the prediction stability unaffected by the error accumulation, demonstrating capacity for arbitrary position prediction.

Task 2.2 Inferring missing variables in real-world and high-dimensional systems. We assessed the capability of the SDRC framework to infer unobserved dynamics using a 14-dimensional Jena climate dataset. During the training phase, all 14 variables were provided as inputs, while five of these (Temperature [T], Saturated vapor pressure [VPmax], Vapor pressure deficit [VPdef], Water vapor concentration [H₂O], and Air density [rho]) were randomly selected to serve as prediction targets. In the testing phase, the inputs consisted of the ground-truth values of the remaining nine variables combined with the predicted values of the five missing variables. This configuration effectively enabled the inference of the unobserved components of the system from a subset of observed variables.

Considering the pronounced nonlinearity of the climate system, each variable in the comparative models was treated as an independent subsystem for node-level modeling within a parallel configuration. As illustrated in Fig. 5(a), the SDRC achieved stable and consistent predictive performance with only 3000 training samples. Figure 5(b) further demonstrates that, compared with the classical RC and PRC, the SDRC maintained superior accuracy and robustness while employing a reservoir size one order of magnitude smaller. Although the LSTM and GRU exhibited notable performance improvements relative to previous tasks, their predictive accuracy remained considerably lower than that of the SDRC.

Task 3.1 Robustness testing with missing training sets. We extend the analysis from Task 2.1 by evaluating the robustness of the SDRC model when trained on missing datasets

generated from the CL63 system Eq. (15). As shown in Fig. 3(d), the gray region represents a complete training set, hereafter termed the training window (range from 1,000 to 6,000). To simulate real-world data imperfections, for each window size, we randomly remove portions of data within this window to create missing training sets. The SDRC model is then retrained using these incomplete datasets under identical parameter configurations. In addition, to ensure statistical reliability, each experiment is repeated 20 times, and the average VPS is used as the performance metric (Fig. 6(a)).

We define the minimal training set as the smallest subset of data within a given training window that achieves prediction performance equivalent to that of a complete training set of the same size. From Fig. 6(b), it can be observed that approximately 12% of the training window can still maintain the same predictive performance as the complete training window even when the data volume is less than 10%. Notably, the model demonstrates a remarkable tolerance for data deficiency (up to 60%) in the optimal training windows required for peak performance, which collectively highlights the exceptional robustness of the SDRC framework. Detailed proportional relationships are provided in Supplementary Note 7.

Parallel missing data experiments were conducted on the phase-space reconstructed Mackey-Glass system, as detailed in Supplementary Note 8. In these scenarios, SDRC exhibited robust predictive performance, achieving a level of stability unattainable by conventional temporal models. This outcome further underscores the efficiency and resilience of the proposed framework.

Task 3.2 Robustness testing under system parameter drift. Generalization capability and structural stability were evaluated by subjecting the SDRC model, trained exclusively on the canonical Lorenz-63 system with $\sigma = 10$, to unseen dynamical regimes characterized by parametric drift. With readout weights $\mathbf{W}_{out}$ and hyperparameters held constant to preclude retraining, the system parameter σ was varied across the range [5, 15] in increments of 0.1. For each parameter value, the average prediction horizon was calculated using 20 randomly generated test samples. Results illustrated in Fig. 7(a) show that predictive fidelity follows a gradual decay profile rather than a sharp cutoff as the system deviates from the training condition. This trend indicates that SDRC captures the intrinsic topological features of the underlying attractor manifold, thereby avoiding overfitting to the local vector field associated with the specific training regime.

To further evaluate predictive robustness under parameter drift, all comparative models were initially trained on data generated with a fixed system parameter $\sigma = 10$. During the testing phase, the system parameter was varied while the model weights remained frozen. To rigorously probe the potential performance of the temporal models (e.g., standard RC and LSTMs), we implemented a "memory washout" mechanism to mitigate the impact of initial state discontinuities. Specifically, prior to initiating the prediction sequence, the models were "warmed up" by feeding a sequence of 100 true

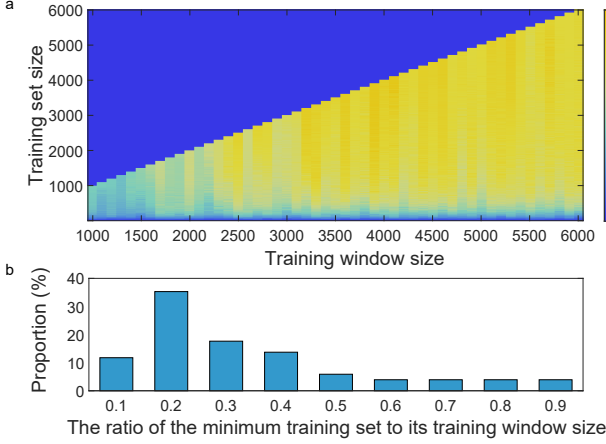


Figure 6. Prediction results of the CL63 system in Eq. (15) with missing training datasets. **a** Within training windows of sizes ranging from 1,000 to 6,000, a random subset of data points is removed to generate missing datasets, which are subsequently used as the training sets for SDRC. The colors at different positions represent the average VPS obtained using the corresponding missing dataset within the training window. **b** The horizontal axis represents the ratio of the minimum training set size to its corresponding window size for each training window, while the vertical axis displays the proportional distribution of these ratios across all windows.

historical data points (The warm-up sequence length for the Transformer matches its specified lag (or lookback window) size). This procedure ensures the internal states are synchronized with the immediate past trajectory, maintaining memory continuity. In stark contrast, SDRC requires no such warm-up phase, as its transient architecture enables instantaneous state synchronization, allowing it to initiate valid predictions immediately from any given point in phase space.

As shown in Fig. 7(b), traditional temporal models, including LSTM, GRU, and Transformer, exhibit predictive performance at least an order of magnitude lower than that of SDRC and PRC. Due to their inherently weak baseline predictive capacity, the performance fluctuations of these temporal models following the warm-up period appear relatively flat. While PRC retains a degree of robustness by leveraging the warm-up mechanism, SDRC consistently demonstrates superior performance across the entire parameter range. The state-mapping structure of SDRC preserves functional integrity by maintaining stable predictive fidelity despite parameter drift and evolving system behaviors, all without requiring historical data for initialization.

Discussion

This study presents the SDRC framework, designed to address the inherent inefficiencies of traditional RC when modeling nonlinear dynamical systems. The framework is specifically

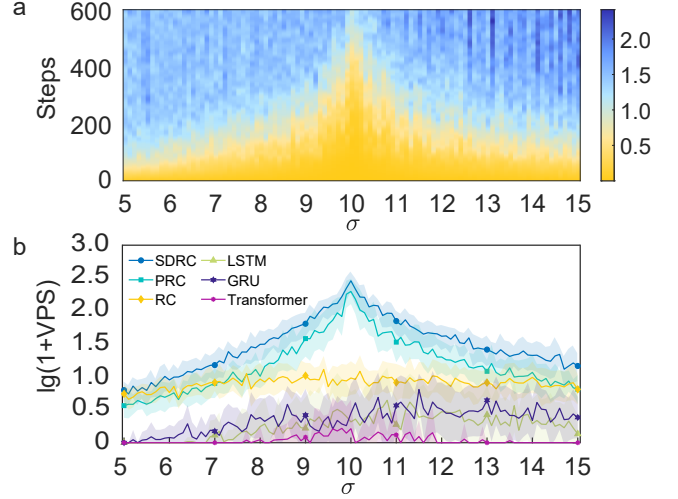


Figure 7. Predictive results for different datasets under parameter drift. **a** The CL63 system in Eq. (15) generates test sets by varying σ in a step-size 0.1 within the range [5, 15] while training the model using only the training set generated at $\sigma = 10$. The colors at different positions represent the average prediction error at the corresponding average prediction steps. **b** The predictive ability of different methods under parameter drift conditions.

optimized for state-complete representations, encompassing both inherent full observations and dynamics restored via phase space reconstruction. In classical RC, the internal recurrence enforces a fading memory of past inputs. While beneficial for temporal pattern recognition, this memory dependence is redundant for state-complete dynamics, where the future evolution is strictly determined by the current state²⁷. SDRC resolves this misalignment by constructing polynomial feature vectors to eliminate stochastic initialization and resetting the reservoir state at each step to block historical propagation. Through this reformulation, the functional equivalence between RC and the Volterra series collapses into a Taylor expansion, effectively replacing historical dependence with a purely state-driven prediction mechanism.

The applicability of SDRC extends beyond systems characterized by state completeness to those exhibiting state incompleteness arising from partial observations or latent variables. Through the application of phase space reconstruction techniques, including time-delay embedding, hidden dynamics are unfolded into a reconstructed state space²⁴. This transformation effectively restores state completeness to the trajectory and renders the dynamics compatible with the instantaneous mapping capabilities of SDRC. Distinct from conventional methodologies that entangle temporal memory with state transition, SDRC isolates the deterministic law of evolution to ensure that the model captures the intrinsic vector field of the system rather than limiting itself to temporal sequence fitting.

This architectural transformation establishes a state-driven

reservoir capable of initiating predictions from arbitrary phase space coordinates, effectively decoupling forecasting tasks from the limitations of specific training trajectories. SDRC consequently extracts invariant informational content embedded within the dataset, constituting a fundamental advantage over Recurrent Neural Networks (RNNs) and related memory-dependent models. Empirical evaluations demonstrate that SDRC achieves predictive accuracy comparable to or exceeding state-of-the-art benchmarks while utilizing less than 25% of the training samples and 6% of the reservoir nodes. Moreover, the observed order-of-magnitude reduction in training time highlights the computational efficiency derived from aligning the model architecture with the intrinsic state completeness of the target dynamics.

Robustness evaluations further validate the framework's stability. In missing dataset tests, SDRC demonstrated resilience to moderate data loss, with performance improving as the training window expanded to capture global attractor geometry. In parameter perturbation tests, where system parameters were altered to induce regime shifts, the fixed weights of SDRC maintained parametric robustness. This suggests that the framework approximates the underlying global manifold rather than a local linearization. We hypothesize that the synergistic interaction between nonlinear input encoding and reservoir dynamics facilitates a faithful manifold reconstruction, thereby enabling global predictive capability even under structural perturbations.

While SDRC demonstrates superior performance on systems where the state is complete or successfully reconstructed, we must delineate its boundary conditions regarding state incompleteness. For systems governed by intrinsic time delays (e.g., the Mackey-Glass system) or partial observations, the raw time series lacks the state completeness required by SDRC. In such cases, SDRC is not a replacement for embedding theory but rather a powerful downstream predictor that operates after phase-space reconstruction.

As shown in our experiments, applying Takens' embedding restores the state completeness of the vector, thereby re-enabling SDRC's instantaneous mapping capabilities. Future work will focus on integrating automatic embedding dimension estimation with SDRC to create an end-to-end framework capable of resolving latent state incompleteness in real-world data, such as partially observed climate subsystems.

Methods

The precise numerical results provided here, such as RMSE, may vary slightly depending on the software used to compute them. We obtained the results presented in this paper using MATLAB R2023a on an x86-64 CPU running Windows 10.

Data availability

The data generated in this study can be recreated by running the publicly available code as described in the Code availability statement.

Code availability

The code used in this study is freely available in the public Zenodo repository: <https://doi.org/10.5281/zenodo.18514031>

References

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
2. Higgins, I. Generalizing universal function approximators. *Nat. Mach. Intell.* **3**, 192–193 (2021). <https://doi.org/10.1038/s42256-021-00318-x>
3. Manjunath, G. & Jaeger, H. Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural. Comput.* **25**, 671–696 (2013).
4. Yan, M. *et al.* Emerging opportunities and challenges for the future of reservoir computing. *Nat. Commun.* **15**, 2056 (2024).
5. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
6. Jaeger, H. The echo state approach to analysing and training recurrent neural networks—with an erratum note. *GMD Technical Report* **148**, 13 (2001).
7. Jaeger, H. & Haas, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
8. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
9. Pathak, J., Lu, Z., Hunt, B. R., Girvan, M. & Ott, E. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos* **27**, 121102 (2017).
10. Pathak, J., Hunt, B., Girvan, M., Lu, Z. & Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102 (2018).
11. Lu, Z. *et al.* Attractor reconstruction by machine learning. *Chaos* **28**, 061104 (2018).
12. Vlachas, P. R. *et al.* Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. *Neural Netw.* **126**, 191–217 (2020).
13. Carroll, T. L. Using reservoir computers to distinguish chaotic signals. *Phys. Rev. E* **98**, 052209 (2018). doi:10.1103/PhysRevE.98.052209.
14. Zhu Q., Ma H., Lin W. Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing. *Chaos* **29**, 093125 (2019).

15. Li X., Zhu Q., Zhao C., *et al.* Tipping point detection using reservoir computing. *Research* **6**, 0174 (2023).
16. Duan X., Ying X., Leng S., *et al.* Embedding theory of reservoir computing and reducing reservoir network using time delays. *Phys. Rev. Res.* **5**(2), L022041 (2023).
17. Larger, L., Baylón-Fuentes, A., Martinenghi, R., Udaltsov, V. S., Chembo, Y. K. & Jacquot, M. High-Speed Photonic Reservoir Computing Using a Time-Delay-Based Architecture: Million Words per Second Classification. *Phys. Rev. X* **7**, 011015 (2017). doi:10.1103/PhysRevX.7.011015.
18. Bollt, E. On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD. *Chaos* **31**, 013108 (2021).
19. Boyd, S. & Chua, L. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Trans. Circuits Syst.* **32**, 1150–1161 (1985).
20. Grigoryeva, L. & Ortega, J.-P. Echo state networks are universal. *Neural Netw.* **108**, 495–508 (2018).
21. Gonon, L. & Ortega, J. P. Reservoir computing universality with stochastic inputs. *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 100–112 (2019).
22. Gonon, L. & Ortega, J.-P. Fading memory echo state networks are universal. *Neural Netw.* **138**, 10–13 (2021).
23. Gonon, L., Grigoryeva, L. & Ortega, J.-P. Approximation bounds for random neural networks and reservoir systems. *Ann. Appl. Probab.* **33**, 28–69 (2023).
24. Takens, F. Detecting strange attractors in turbulence. in *Dynamical Systems and Turbulence* (Springer, 1981).
25. Lu, Z. *et al.* Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos* **27**, 041102 (2017).
26. Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. *Sci. Rep.* **2**, 514 (2012).
27. Storm, L., Gustavsson, K. & Mehlig, B. Constraints on parameter choices for successful reservoir computing. *Mach. Learn.: Sci. Technol.* **3**, 045021 (2022).
28. Gauthier, D. J., Bollt, E., Griffith, A. *et al.* Next generation reservoir computing. *Nat. Commun.* **12**, 5564 (2021).
29. Fan, H., Jiang, J., Zhang, C., Wang, X. & Lai, Y.-C. Long-term prediction of chaotic systems with machine learning. *Phys. Rev. Res.* **2**, 012080 (2020).
30. Norton, D. A., Zhang, Y. & Girvan, M. Learning beyond experience: Generalizing to unseen state space with reservoir computing. *Chaos* **35**, 103146 (2025).
31. Jiang, J. & Lai, Y.-C. Model-free prediction of spatiotemporal chaotic systems with recurrent neural networks: Role of network spectral radius. *Phys. Rev. Res.* **1**, 033056 (2019).
32. Griffith, A., Pomerance, A. & Gauthier, D. J. Forecasting chaotic systems with very low connectivity reservoir computers. *Chaos* **29**, 123108 (2019).
33. Inubushi, M. & Yoshimura, K. Reservoir computing beyond memory-nonlinearity trade-off. *Sci. Rep.* **7**, 10199 (2017).
34. Weng, T. *et al.* Synchronization of reservoir computers with applications to communications. *Phys. A: Stat. Mech. Appl* **544**, 123453 (2020).
35. Lorenz, E. N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963).
36. Lin, W., Chen, G. *et al.* Large memory capacity in chaotic artificial neural networks: A view of the anti-integrable limit. *IEEE Trans. Neural Netw.* **20**, 1340–1351 (2009).
37. Lestienne, R. Spike timing, synchronization and information processing on the sensory side of the central nervous system. *Prog. Neurobiol.* **65**, 545–591 (2001).
38. Mackey, M. C. & Glass, L. Oscillation and chaos in physiological control systems. *Science* **197**, 287–289 (1977).
39. Banerjee, A. *et al.* Using machine learning to assess short term causal dependence and infer network links. *Chaos* **29**, 121104 (2019).
40. Li, X., Zhu, Q., Zhao, C. *et al.* Higher-order Granger reservoir computing: simultaneously achieving scalable complex structures inference and accurate dynamics prediction. *Nat. Commun.* **15**, 2506 (2024).
41. Sangiorgio, M. & Dercole, F. Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. *Chaos Solitons Fractals* **139**, 110045 (2020).
42. Köster, F. *et al.* Attention-enhanced reservoir computing. *Phys. Rev. Appl.* **22**, 014039 (2024).

Acknowledgements

C.G. is supported by the National Natural Science Foundation of China (No. 12275179). Q.Z. is supported by the National Natural Science Foundation of China (No. 62406072, (No. 12171350) and by the STCSM (Nos. 23YF1402500, 21511100200, 22ZR1407300, and 22dz1200502). W.L. is supported by the National Natural Science Foundation of China (No. 11925103) and by the STCSM (Nos. 22JC1402500, 22JC1401402, and 2021SHZDZX0103). H.W. is supported by the National Natural Science Foundation of China (No. 12305047).

Author contributions

T. B. Sun, C. G. Gu and W. Lin conceived the idea. T. B. Sun and Q. X. Zhu designed the research and helped perform the analysis with constructive discussions. T. B. Sun, W. Lin and C. G. Gu performed the experiments and wrote the initial draft of the manuscript. W. X. Zheng and J. Wang collected the data and carried out additional analyses. J. M. Moore, H.

Y. Wang and H. J. Yang contributed to refining the ideas. All authors contributed to writing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at *.

Correspondence and requests for materials should be addressed to Changgui Gu or Wei Lin.