Clique-based Topology Enhances Reservoir Computing Performance

A. E. Hramov,[1] A. V. Andreev,[1] N. D. Kulagin,[1] A. A. Badarin,[1] S. A. Kurkin,[1] H. Gao,[2] L. Minati,[3,4] and S. Boccaletti[5,6,7]

[1)]*Research Institute for Applied Artificial Intelligence and Digital Solutions, Plekhanov Russian University of Economics, 115054 Moscow, Russia*

[2)]*Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin, 150001, Heilongjiang, China*

[3)]*School of Life Science and Technology, University of Electronic Science and Technology of China, 610054 Chengdu, Sichuan, China*

[4)]*Center for Mind/Brain Sciences (CIMeC), University of Trento, 38123 Trento, Italy*

[5)]*Research Institute of Interdisciplinary Intelligent Science, Ningbo University of Technology, 315104 China*

[6)]*Sino-Europe Complexity Science Center, North University of China, 030051, Taiyuan, China*

[7)]*CNR - Institute of Complex Systems, Via Madonna del Piano 10, I-50019, Sesto Fiorentino, Italy*

(*Electronic mail: Corresponding author: lminati@uestc.edu.cn)

(*Electronic mail: Corresponding author: hjgao@hit.edu.cn)

(*Electronic mail: Corresponding author: hramovae@gmail.com)

(Dated: 30 June 2026)

Reservoir computing represents a robust machine-learning approach displaying high performance in predicting the behavior of natural and artificial complex systems. However, its full potential has been so far hindered by the limitations of traditional architectures. Here, we give the practical demonstration of the fundamental role that the specific network topology may play in machine learning systems, and we propose a novel reservoir computer whose hidden layer comprises nodes that form cliques of varying sizes. The result is a reservoir with far more advanced data representation and processing capabilities, which consistently outperforms conventional architectures in tasks including predicting stochastic neuron dynamics, modeling the evolution of chaotic systems, and reconstructing electroencephalogram signals. Furthermore, our approach addresses a pivotal challenge in the physical implementation of reservoirs: it mitigates the sensitivity to the choice of specific topologies, improving generality and facilitating the deployment of more compact and efficient architectures. Through a systematic analysis, we elucidate the functional significance of each topological element and offer practical design strategies.

Reservoir computing is a powerful machine learning framework for predicting complex dynamics, yet its performance is often limited by randomly wired hidden layers. In this work, we demonstrate that network topology is not merely a technical detail but a fundamental determinant of predictive capability. We introduce a reservoir computer whose hidden layer is built from cliques (triangles, tetrahedrons, and larger cliques) combined with isolated nodes. This clique-based architecture consistently outperforms traditional reservoirs across multiple benchmarks: forecasting stochastic FitzHugh–Nagumo neuron dynamics, predicting chaotic Lorenz and Rössler systems, and reconstructing hidden EEG signals. Beyond accuracy, our design reduces sensitivity to random initialization, lowers memory requirements, and simplifies physical implementation. Using information-theoretic analysis, we identify why certain cliques enhance performance and provide practical guidelines for constructing efficient, high-performance reservoirs with minimal connectivity.

I. INTRODUCTION

Predicting the behavior of natural systems (such as weather evolution, neural activity, and other physiological dynamics) as well as engineered systems operating under uncertain conditions (such as process controllers and power converters) is one of the most important problems at the intersection of nonlinear science and machine learning[1]. Reservoir computing[2,3] has emerged as a promising model-independent machine learning paradigm that is well suited for predicting the behavior of dynamical systems, even in the face of environmental uncertainties[4–6]. A reservoir computer (RC) is characterized by a resource-efficient training process, significantly less demanding than deep convolutional networks, and a simple architecture with a fixed hidden layer of interacting neurons with predetermined connections. The deployability and adaptability of RC methods has been demonstrated across a wide range of scenarios, including electronic, mechanical, optical, and biological systems, which have been successfully used as physical computation platforms[7]. Even when dealing with chaotic or complex spatiotemporal behaviors (hallmarking some of the most significant challenges in prediction), RCs have often provided predictions beyond the horizon of other known methods in the literature. Moreover, recent

work has validated the effectiveness of RCs in the context of stochastic systems, successfully predicting the dynamics of a FitzHugh-Nagumo stochastic neuron over different noise control parameters, including stochastic resonance effects[8].

The core of the RC paradigm, shown in Fig. 1, is a neural network comprising three layers: an input layer that processes the observed data $\mathbf{g}$, a hidden layer of randomly interconnected neurons governed by nonlinear activation functions φ, and an output layer. The hidden layer forms a random network according to an adjacency matrix $\mathbf{W}$, characterized by an average node degree $\langle k \rangle$. To ensure stability, the hidden-to-hidden adjacency matrix $\mathbf{W}$ is rescaled so that its spectral radius (i.e., the absolute value of the largest eigenvalue) is equal to a predefined value λ. While the weights of the input-to-hidden matrix $\mathbf{G}$ and the hidden-to-hidden matrix $\mathbf{W}$ are randomly assigned at reservoir creation, the hidden-to-output weights $\tilde{\mathbf{G}}$ are iteratively optimized during a training phase, typically using regularized linear optimization, such as ridge regression[9], to minimize overfitting.

The theoretical foundations of reservoir computing are still not fully understood, although several influential theoretical works have substantially clarified its underlying principles. Among the earliest and most important contributions, Boyd and Chua showed that any causal time-invariant operator with fading memory can be approximated by a finite-dimensional linear dynamical system followed by a nonlinear readout. This result provides a conceptual template in which linear internal dynamics are combined with a nonlinear readout function, a structure that closely anticipates what is now known as reservoir computing[10]. Building on this foundational insight, Grigoryeva and Ortega demonstrated that echo-state networks satisfy a universal approximation property for all causal fading-memory filters, even when the reservoir is finite-dimensional and the readout is purely linear[11]. This line of development was further extended by Gonon and Ortega, who showed that ESNs retain universality in the L^p sense for stochastic inputs without requiring boundedness of the underlying processes[12].

In addition, Bollt addressed the question of why a reservoir computer can work at all by analyzing a simplified model with linear activation[13]. He showed that such a reservoir is mathematically equivalent to a vector autoregressive (VAR) model, which brings reservoir computing directly under the scope of the Wold representation theorem. Since the Wold theorem guarantees that any stationary process admits a representation in terms of deterministic linear structure and a moving-average innovation sequence, the VAR form provides a principled explanation for why even a randomly constructed linear reservoir can reproduce meaningful short-term dynamics. Bollt further noted that adding quadratic readout terms yields a nonlinear VAR model, extending this connection to settings that require capturing nonlinear behavior.

Although universality theorems for reservoir computing require sufficiently large hidden-state dimension, practical implementations often rely on much smaller reservoirs that nonetheless achieve high accuracy; in such low-dimensional settings, the influence of topology is no longer suppressed by scale, making the specific connectivity structure a critical determinant of the reservoir's dynamics. Empirical studies show that different topologies exhibit markedly different dynamical properties. In particular, Dale et al. showed that different reservoir topologies exhibit systematically different ranges of attainable dynamical behaviors, with more structured networks displaying a more restricted behavioral repertoire compared to architectures with higher connection complexity[14]. Kawai et al. found that adding long-range shortcuts to form a small-world topology improves the echo-state property and supports better signal propagation and prediction accuracy[15]. Rathor et al. further showed that asymmetric random connectivity provides the highest information-processing capacity and outperforms symmetric and structured networks in chaotic time-series forecasting tasks[16]. These results indicate that reservoir topology plays a crucial role in practical implementations of reservoir computing.

Despite the broad utility of RCs, there remains a critical challenge in designing optimal connectivity patterns within the reservoir. The performance of RCs can vary widely depending on the random configuration of the hidden layer[17], fluctuating significantly even for a fixed choice of $\langle k \rangle$ and λ (referred to as hyperparameters in the RC jargon adopted here). Recent works indicate that the optimal configuration (where the internal reservoir dynamics achieve generalized synchronization with the input signal $\mathbf{g}$) can be identified by tuning the hyperparameters $\langle k \rangle$ and λ[18,19], but finding universal design rules or principles for this type of synchronization state remains elusive[20–24].

Previously, the advantages of sparse reservoirs with small number of connections have been previously discussed in[21]. We suggest constructing a specific topological architecture based on predefined rules – namely, forming small all-to-all connected groups (Q-cliques) of the fixed size Q based on the initial adjacency matrix $\mathbf{W}$: Q-clique in such case corresponds to a principal submatrix of size $Q \times Q$ where all off-diagonal entries are 1 (assuming no self-loops). Interaction inside each clique is introduced via a mean-field principle. These structures, such as triangles (3-cliques) and tetrahedrons (4-cliques), provide a considerably richer framework for representing multi-node dependencies[25]. Fig. 1c illustrates the distinctions between edges (2-cliques), triangles (3-cliques), and tetrahedrons (4-cliques). In this context, one can isolate all Q-cliques in the hidden layer, as shown in Fig. 1b, and then consider an RC, where the hidden layer consists of a combination of Q-cliques and isolated nodes. Examples of such reservoirs are one based on 3-cliques, shown in Fig. 1d, and another with 4-cliques, shown in Fig. 1e. Both configurations will be studied in detail below.

The results indicate that the introduction of such specific topological architecture markedly enhances the data-processing capabilities of reservoirs. In particular, we demonstrate that an RC system which contains only isolated nodes and Q-cliques in its hidden layer (referred to henceforth as a clique-based RC (CB-RC)) consistently outperforms conventional RCs in several benchmark tasks, including predicting the dynamics of stochastic neurons, predicting chaotic systems, and reconstructing hidden data from electroencephalography (EEG) recordings. It is also noteworthy that CB-

Chaos
An Interdisciplinary Journal of Nonlinear Science
AIP Publishing

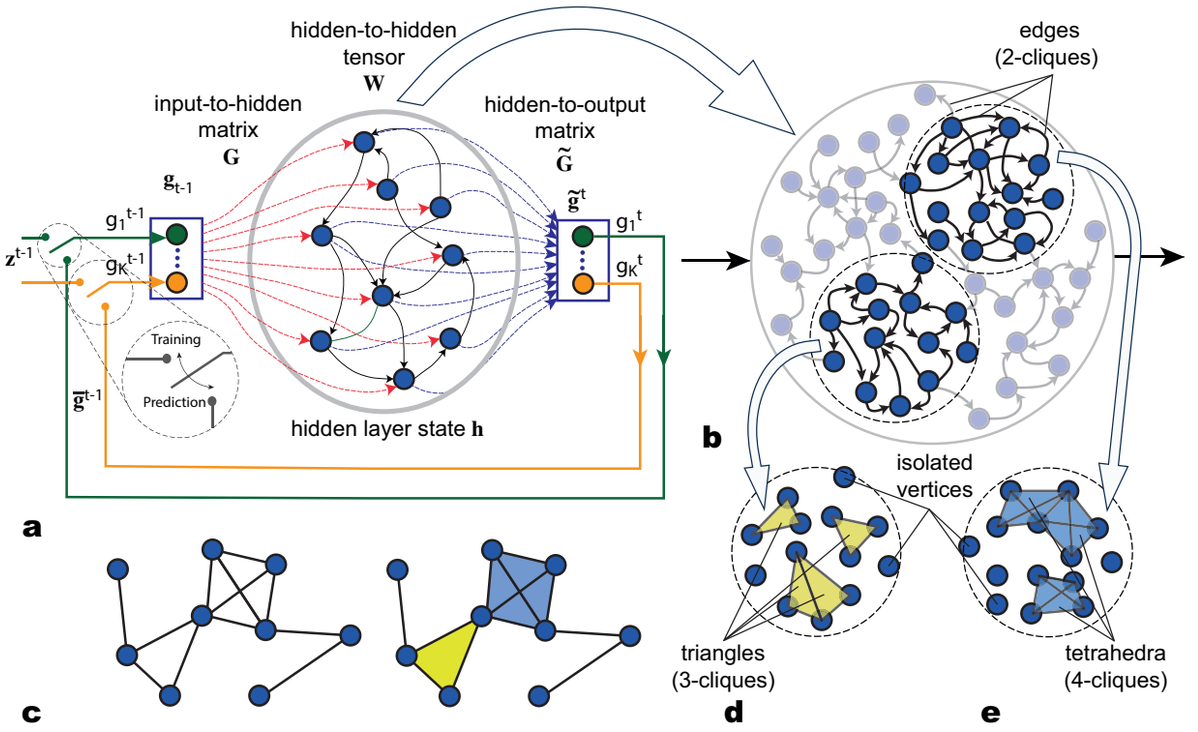

FIG. 1. **Reservoir computer for predicting temporal dynamics with traditional topology and cliques with all-to-all connected nodes in the hidden layer.** **(a)** A traditional RC processes time series data as a vector $\mathbf{g}$ with dimension K. In the training mode, the feedback loop is open, whereas in the prediction mode the output is fed back to the input. **(b)** Structure of the hidden reservoir layer as an artificial recurrent neural network with traditional topology. **(c)** Examples of cliques with different size: edges (2-cliques, black), triangles (3-cliques, yellow), and tetrahedrons (4-cliques, blue). **(d)** Formation of the hidden layer containing 3-cliques and a set of isolated nodes. **(e)** Formation of the hidden layer containing 4-cliques and a set of isolated nodes.

RCs exhibit a reduced dependence on specific interconnection topologies and memory usage, simplifying the physical implementation, particularly in hardware-constrained environments such as low-power integrated circuits. Through a comprehensive examination, we map the functional role of each topological component within the hidden layer, leading to a practical design strategy. In the following sections, we consider three benchmark problems, namely predicting the dynamics of the FitzHugh-Nagumo stochastic neuron[8], predicting the evolution of the Lorenz and Rössler chaotic systems[17,26,27], and reconstructing hidden data from EEG recordings of human brain activity.

II. RESERVOIR COMPUTERS WITH CLIQUE-BASED TOPOLOGY

As illustrated in Fig. 1, the reservoir computing paradigm involves feeding input data $\mathbf{g}$ into a high-dimensional network of N interconnected nodes. This network, or reservoir, generates a state vector $\mathbf{h}$, which is subsequently transformed into an output $\tilde{\mathbf{g}}$ that attempts to closely approximate the desired output $\mathbf{z}$. The connections between nodes, represented by the adjacency matrix $\mathbf{W}$, are randomly assigned and remain fixed throughout the process. The input data $\mathbf{g}$ is introduced into the reservoir through an input layer characterized by fixed random coefficients, denoted by the matrix $\mathbf{G}$. The final output $\tilde{\mathbf{g}}$ is generated in the output layer using the matrix $\tilde{\mathbf{G}}$, whose coefficients are fine-tuned to reproduce the desired output state $\mathbf{z}$ as accurately as possible.

The reservoirs under consideration operate as discrete dynamical systems, with their dynamics represented by[3]

$$\begin{cases} \mathbf{h}^t = F(\mathbf{g}^{t-1}, \mathbf{h}^{t-1}, \mathbf{W}, \mathbf{G}) \\ \tilde{\mathbf{g}}^t = U(\mathbf{h}^t, \tilde{\mathbf{G}}), \end{cases} \tag{1}$$

where F is the vector field governing the dynamics within the hidden layer, and U is the output function encapsulating the functional couplings between the reservoir variables. Once the vector field F, the output function U, and the matrices $\mathbf{W}$ and $\mathbf{G}$ are established, the RC is ready to perform learning tasks.

For the specific problem of discrete time series forecasting, represented by $\mathbf{z}^t$, the learning process involves the formulation of an optimization problem. This problem seeks to determine the optimal coefficients of the output matrix $\tilde{\mathbf{G}}$, with the goal of minimizing the loss function, which is defined as the L_2 error between the estimated states $\tilde{\mathbf{g}}^t$ and the target states

$\mathbf{z}^t$:

$$L_2 = \sum_{t=1}^{T_{\text{opt}}} \left(||U(\mathbf{h}^t, \tilde{\mathbf{G}}) - \mathbf{z}^t||^2 + R(\tilde{\mathbf{G}}) \right), \tag{2}$$

where $R(\tilde{\mathbf{G}})$ represents the regularization term, and T_{opt} is the length of the time interval during which the optimization is performed.

Accordingly, two modes of operation can be distinguished. In the first mode, the training phase (during which the switch in Fig. 1a is set to "training"), the input signal $\mathbf{g}^t = \mathbf{z}^t$ is fed to the RC and the corresponding output $\tilde{\mathbf{g}}^{t+1}$ is generated. During this phase, the output layer $\tilde{\mathbf{G}}$ is trained by numerically solving the optimization problem described in Eq. (2). After the training phase, the prediction phase begins (with the switch in Fig. 1a set to "prediction"), during which the RC autonomously predicts the time series. The same reservoir equations [Eq. (1)] are applied during this phase, using the output layer weights $\tilde{\mathbf{G}}$ determined during training.

To summarize, each node $i \in (1,\ldots,N)$ in the network receives inputs from the other nodes $j \in (1,\ldots,N)$, with $i \neq j$ and from the input layer. The connections between nodes i and j in the hidden layer are defined by the weights ω_{ij}, which are encapsulated in the adjacency matrix $\mathbf{W}$. As mentioned above, the accuracy achievable by the RC is significantly affected by the topology of the hidden layer, i.e., the structure of the matrix $\mathbf{W}$.

In traditional RCs, the fixed hidden layer matrix $\mathbf{W}$ is typically generated as a random network with a predetermined density, corresponding to an average node degree $\langle k \rangle$. Each link between nodes is assigned a pairwise coupling strength ω_{ij}, which is randomly chosen within the interval $[0,1]$ (see Fig. 1a,b). To ensure the stability of the network, the adjacency matrix $\mathbf{W}$ is rescaled so that the spectral radius, defined as the magnitude of the largest eigenvalue, is equal to a given value λ. This approach results in a random matrix $\mathbf{W}$ characterized by hyperparameters $(\langle k \rangle, \lambda)$. A necessary preparatory step is the optimization of the hidden layer of the reservoir, which involves identifying the optimal values for the hyperparameters, denoted as $(\langle k \rangle_0, \lambda_0)$, by applying techniques such as parameter sweeping or iterative search algorithms. It is important to note that each attempt to generate a matrix $\mathbf{W}$ for a given pair of hyperparameters $(\langle k \rangle_0, \lambda_0)$ results in a new random matrix, which can lead to significant fluctuations in performance. Currently, this problem is largely addressed empirically: multiple RCs are generated and trained, and the best-performing candidates are then selected[17,21].

In the following, we propose an algorithm for constructing a reservoir computer with clique-based topology. The process begins by generating a parent random network determined by the hyperparameter $\langle k \rangle$, as is done in conventional RCs (see Fig. 1). This step results in a random binary undirected graph $\mathbf{C}$, described by an adjacency matrix with elements $c_{ij} = 0, 1$.

Next, we define the dimension Q, where $Q = 2$ corresponds to a link – clique of 2 nodes. Using the algorithm proposed in Ref.[28] to identify all cliques of size Q in the undirected graph $\mathbf{C}$ [see Section A 3], we identify all cliques with size Q (the so-called Q-cliques). As a result, the nodes in the graph representing the hidden layer are divided into two subsets: (i) the set $\mathscr{D}$ of nodes that form Q-cliques, and (ii) the set $\mathscr{I}$ of nodes that are not part of any Q-clique. We then focus on a network consisting exclusively of the identified Q-cliques and the set $\mathscr{I}$ of isolated nodes. Figures 1d, e illustrates this partitioning process for networks divided into the 3- and 4-cliques and isolated nodes, respectively. The total number of nodes in the hidden layer remains unchanged, equal to N, as in the case of traditional topology.

In the last step, we construct the adjacency tensors $\mathbf{W}^Q$ to represent the topology of the hidden layer with Q-cliques. In general, Q-cliques are encoded in the adjacency tensors $\mathbf{W}^Q$ in such a way that if a Q-cliques consists of the vertices $(q_0,\ldots,q_Q)$, the corresponding tensor element $\omega^Q_{q_0\ldots q_Q}$ is set to 1, otherwise 0. For example, the interactions corresponding to 2-, 3-, and 4-cliques are encoded in the adjacency matrix $\mathbf{W}^2$ and the adjacency tensors $\mathbf{W}^3$ and $\mathbf{W}^4$, respectively. Specifically, $\omega^2_{ij} = 1$ if there is a link between nodes i and j, $\omega^3_{ijk} = 1$ if there is a triangle formed by nodes i, j, and k, and $\omega^4_{ijkl} = 1$ if there is a tetrahedron formed by nodes i, j, k, and l. The coupling strength values for non-zero tensor elements $\omega^Q_{q_0\ldots q_{Q-1}} \neq 0$ are randomly chosen over the interval $[0,1]$.

To ensure the stability of the network, the adjacency tensor $\mathbf{W}^Q$ is again rescaled so that the spectral radius of the new adjacency tensor $\hat{\mathbf{W}}^Q$ (the largest absolute eigenvalue obtained from the eigendecomposition of the flattened tensor[29]) is equal to λ, with

$$\hat{\mathbf{W}}^Q = \frac{\lambda}{\lambda_0}\mathbf{W}^Q, \tag{3}$$

where λ_0 is the initial spectral radius of the adjacency tensor $\mathbf{W}^Q$.

It follows that, in the case of the clique-based reservoir computer (CB-RC) with clique's size Q, the dynamical system in Eq. (1) is written as

$$h_i^t = \varphi\left(\sum_{j=1}^{N} G_{ij} g_j^t + \frac{1}{Q-1} \underbrace{\sum_{q_1} \cdots \sum_{q_{Q-1}}}_{Q-1} \omega^Q_{iq_1\ldots q_{Q-1}} \sum_{s=q_1}^{q_{Q-1}} h_s^{t-1} \right), \tag{4}$$

$$\tilde{g}_k^t = \sum_{l=1}^{N} \tilde{G}_{l,k} \hat{h}_l^t, \quad i = 1\ldots N, \quad k = 1\ldots K, \tag{5}$$

where K is the number of forecasted signals in the CB-RC. The function $\varphi(\cdot)$ in Eq. (4) represents the nonlinear activation function of the neurons in the hidden layer, applied element-wise to the corresponding state variables h_i. Although the specific reservoir dynamics illustrated in Eqs. (4)–(5) are based on examples from previous studies[6,8], it is important to note that various alternative schemes have been proposed to describe RC dynamics in different contexts[2,3,30,31].

The first term in parentheses in Eq. (4) corresponds to the signals from the input layer, while the second term generalizes the traditional reservoir equations to the case of Q-cliques in

the hidden layer. It should be noted that in the case of $Q = 2$, Eqs. (4) and (5) reduce to the equations of the traditional RC. The constructed adjacency tensor $\mathbf{W}^Q$ (see also Figs. 1d and e) in Eq. (4) describes the dynamics of a CB-RC with N nodes, which are divided into two subsets: (i) the set $\mathscr{D}$ containing D nodes forming cliques of Q nodes, and (ii) the set $\mathscr{I}$ containing I isolated nodes, where $I + D = N$. For the isolated nodes in $\mathscr{I}$, the second term in Eq. (4) is equal to zero.

Eq. (5) provides an estimate of the predicted vector $\tilde{\mathbf{g}}^t$ at time t using a readout hidden-to-output matrix, $\tilde{\mathbf{G}}$. The N-dimensional vector $\hat{\mathbf{h}}^t$ in Eq. (5) represents an augmented hidden state, where each component is defined as $\hat{h}_i^t = h_i^t$ if i is odd, and $\hat{h}_i^t = (h_i^t)^2$ if i is even. This augmentation, which squares the hidden state in half of the nodes, increases the dynamical complexity of the CB-RC[8,32].

As shown in Fig. 1a, during the training phase, the input signal $\mathbf{g}^t = \mathbf{z}^t$ is fed to the RC and the corresponding output signal $\tilde{\mathbf{g}}^t$ is obtained. To train the output layer $\tilde{\mathbf{G}}$, the L_2 error is minimized according to Eq. (2). This is achieved using ridge regression[9], which serves to mitigate the problem of overfitting by imposing a penalty on the fitting parameters if they are too large. In this case, the term $R(\tilde{\mathbf{G}})$ in the optimization problem is represented as $R(\tilde{\mathbf{G}}) = \beta||\tilde{\mathbf{G}}||^2$, where β is set to 10^{-4} as the L_2 error regularization hyperparameter. Once the training phase is completed, the prediction phase uses the CB-RC equations described above [see Eqs. (4) and (5)] with the selected hyperparameters and the output layer weights $\tilde{\mathbf{G}}$.

To evaluate the prediction accuracy, the root mean square error (RMSE) is calculated as follows

$$\mathrm{RMSE} = \sqrt{\frac{1}{T_{\mathrm{pred}}} \sum_{t=1}^{T_{\mathrm{pred}}} ||\tilde{\mathbf{g}}^t - \mathbf{z}^t||^2}, \tag{6}$$

where T_{pred} is the duration of the prediction time interval.

III. FORECASTING THE STOCHASTIC FITZHUGH-NAGUMO NEURON DYNAMICS

A. Problem specification

As a benchmark problem, we first consider the problem of predicting the dynamics of a stochastic FitzHugh-Nagumo (SFHN) neuron. This is a particularly challenging case, where the emergent properties of the deterministic nonlinear dynamics are simultaneously perturbed and enriched by the stochastic component[33]. The system equations are as follows:

$$\dot{x} = x - \frac{x^3}{3} - y + a, \;\; \dot{y} = 0.08(x - 0.8y + 0.7) + D\xi(t), \tag{7}$$

where x and y denote the excitation and recovery variables. The stochastic term $\xi(t)$ is a zero-mean white Gaussian noise with an autocorrelation function $\langle \xi(t)\xi(t+\tau)\rangle = \delta(\tau)$, and D denotes the noise intensity. The parameter a plays an important role in determining the equilibrium points and, consequently, the excitability threshold of the neuron. In this study we focus on the excitable regime, specifically the pre-critical state that undergoes a Hopf bifurcation at $a \approx 0.3218$, where the neuron remains free of self-sustained oscillations. We thus set $a = 0.3$, a value that falls within the excitable range. With this configuration and in the absence of noise ($D = 0$), the system reaches a steady state. As the noise intensity D increases, spike generation occurs, with the spike characteristics becoming increasingly dependent on the noise intensity. For consistency with our previous work, we set $D = 0.05$[8].

The essential features of the RC are consistent with those of our previous work, which demonstrated the successful prediction of the dynamics of a stochastic FitzHugh-Nagumo neuron model using a reservoir with traditional pairwise interactions in the hidden layer[8]. The architecture of the network is shown in Fig. 2a.

First, the model has three inputs, which determine the SFHN dynamics: a noise term, $D\xi^t$, which excites the neuron, and a two-component input signal vector, $\mathbf{z}(t) = (x(t), y(t))^{\mathrm{T}}$, which contains the dynamic variables of the model. For generality, we denote the input vector as $\mathbf{g}(t) = (D\xi(t), x(t), y(t))^{\mathrm{T}}$. Since it is not possible to predict the noise, the output layer contains only two outputs[8], corresponding to the vector $\tilde{\mathbf{g}}(t) = (\tilde{x}(t), \tilde{y}(t))^{\mathrm{T}}$.

Second, a distinctive aspect of our RC configuration is the complete segregation of reservoir inputs between different neurons within the hidden layer. Fig. 2a illustrates that each input within the reservoir is connected only to a specific subpopulation of hidden layer neurons, specifically to one-third of them, highlighted in the corresponding color. The input vector $\mathbf{g}(t)$ influences the internal high-dimensional hidden state $\mathbf{h}^t$ of the RC with dimension $N = 3n$ (where $n \in \mathbb{N}^+$ is a member of the set of natural numbers excluding zero). The coupling of the input to the hidden state is mediated by the input-to-hidden matrix $\mathbf{G}$, which is an $N \times 3$ matrix with elements taking the values

$$G_{i,j} = \begin{cases} \delta_{i,j} & \text{if} \quad 1 + (j-1)n \le i \le jn, \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

where the values $\delta_{i,j}$ are uniformly sampled from the interval $[-1, 1]$, and the index $j \in (1, 2, 3)$. In other words, the external noise corresponding to the first component of the input vector ($j = 1$) affects the first n neurons of the hidden layer of the RC with weights $\delta_{i,1}$, the variable x ($j = 2$) affects the next n neurons with weights $\delta_{i,2}$, and the variable y ($j = 3$) affects the remaining n neurons with weights $\delta_{i,3}$.

Third, the hidden-to-hidden tensors $\mathbf{W}^Q$ are created according to the algorithm described in Section II. As an illustrative example, Fig. 2b shows the typical structure of the hidden layer in the case of $Q = 4$, containing both 4-cliques and isolated nodes. Each storage node h_i also has an output, which can be described by the Eq. (4). Here we have employed the hyperbolic tangent, denoted as $\varphi(\cdot) = \tanh(\cdot)$, as the activation function.

Fourth, given Eq. (5) with the extended hidden state $\hat{\mathbf{h}}^t$, the output vector $\tilde{\mathbf{g}}^t$ is calculated by a readout hidden-to-output matrix, denoted as $\tilde{\mathbf{G}}$. Since the output vector $\tilde{\mathbf{g}}^t = (\tilde{x}^t, \tilde{y}^t)^{\mathrm{T}}$ has only two components, the output matrix $\tilde{\mathbf{G}}$ is a $2 \times N$ matrix with trainable weights.

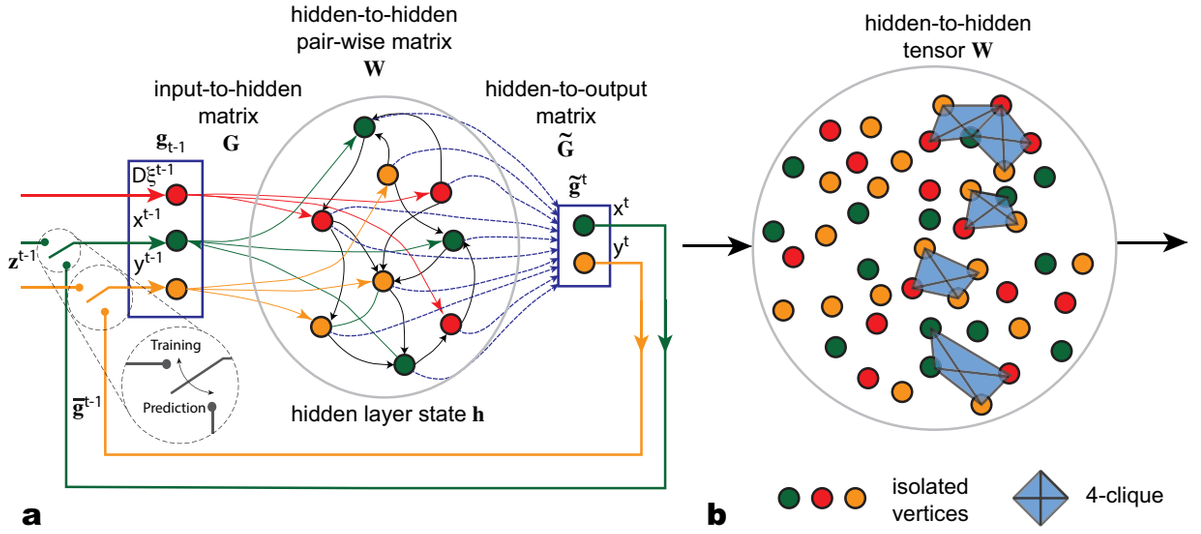

FIG. 2. **Schematic representation of a reservoir computer with fully separated inputs for predicting stochastic neuron dynamics. (a)** General operational scheme of the reservoir computer in training and prediction modes. Color denotes the different inputs and their segregation in the hidden layer: the red, green, and orange nodes represent neurons that receive influence from the stochastic input ξ, the x variable, and the y variable, respectively. **(b)** Transition to a hidden layer that is endowed with a set of 4-cliques and a set of isolated nodes.

In the fifth step, the input signal $\mathbf{g}^t$ in the training mode, which includes the driving noise $D\xi^t$ and the SFHN neuron signal $\mathbf{z}^t$, is fed into the input of the RC, and the corresponding output signal $\tilde{\mathbf{g}}^t$ is obtained. To train the output layer $\tilde{\mathbf{g}}$, the L_2 error between the target states $\mathbf{z}^{t+1}$ and the estimated states $\tilde{\mathbf{g}}^{t+1}$, according to Eq. (2), is minimized with $T_{\text{opt}} = 20000$.

Finally, following the training phase, the prediction phase is initiated, during which the RC attempts to autonomously reproduce the stochastic neuron dynamics, as shown in Fig. 2a. In the prediction phase, the same CB-RC equations described above [see Eqs. (4) and (5) with the above hyperparameters] are employed, with the previously determined output layer weights $\tilde{\mathbf{G}}$. However, the input vector $\tilde{\mathbf{g}}^{t-1}$ now consists of the calculated output vector $\tilde{\mathbf{g}}^t$ from the previous step, while the external noise (noise from the first input $g_1^t = D\xi(t)$) continues to drive the CB-RC, which then retains a continuous stochastic influence.

B. Increased prediction accuracy

Four configurations of CB-RC are studied, differing only in the topology of the hidden layer (the adjacency matrix or tensor $\mathbf{W}^Q$). Specifically, we consider the cases of four sizes of the cliques, Q: (i) $Q = 2$, which represents a conventional reservoir with traditional topology (a system previously studied in Ref. [8]), (ii) $Q = 3$, corresponding to a set of triangles (3-cliques) and a set of isolated nodes, (iii) $Q = 4$, where a set of tetrahedrons (4-cliques) and a set of isolated nodes, and (iv) $Q = 5$ with a set of pentachorons and a set of isolated nodes are present. The architecture of the input and output layers and the procedure for training the output weights are identical for the three cases.

With respect to the hidden layer, its size is the same in all three cases ($N = 501$), implying that the number of hidden neurons connected to each neuron from the input layer is $n = N/3 = 167$. For each CB-RC with clique's size Q, the average node degree $\langle k \rangle$ was varied in the range $[2, 20]$ for $Q = 2, 3, 4$ and in the range $[25, 43]$ for $Q = 5$ with a step size of 1, and the spectral radius λ was varied in the range $[0.5, 1.9]$ with a step size of 0.1 for all Q. The prediction accuracy depends on the given set of hyperparameters and the particular topology of the connections within the internal layer. To account for this, we generated ten random topologies for each set of hyperparameter values, resulting in 2,850 instances of CB-RC with a specific clique's size Q but different hyperparameters and link topologies. The resulting individual reservoirs were trained on the same stochastic neuron data, and the RMSEs according to Eq. (6) were computed to evaluate the prediction quality of each. Fig. 3a shows the distributions of prediction accuracy (RMSE values) for four types of reservoirs: a traditional reservoir with traditional topology ($Q = 2$), and CB-RCs with triangles ($Q = 3$), tetrahedrons ($Q = 4$), and pentachorons ($Q = 5$). It can be observed that the distribution shifts monotonically to the left, i.e., towards smaller RMSE, with increasing clique's size Q.

Fig. 3b shows the time series of the predicted signal $\tilde{x}(t)$, the target signal $x(t)$, and the prediction error $\Delta(t) = x(t) - \tilde{x}(t)$ for the reservoirs with the best (minimum RMSE) and low (RMSE corresponding to the mean of the distribution) prediction accuracy for the traditional reservoir with $Q = 2$ and the CB-RC with $Q = 4$. In the best case, the predicted and target signals are almost indistinguishable. The prediction error is more informative and is influenced by small de-

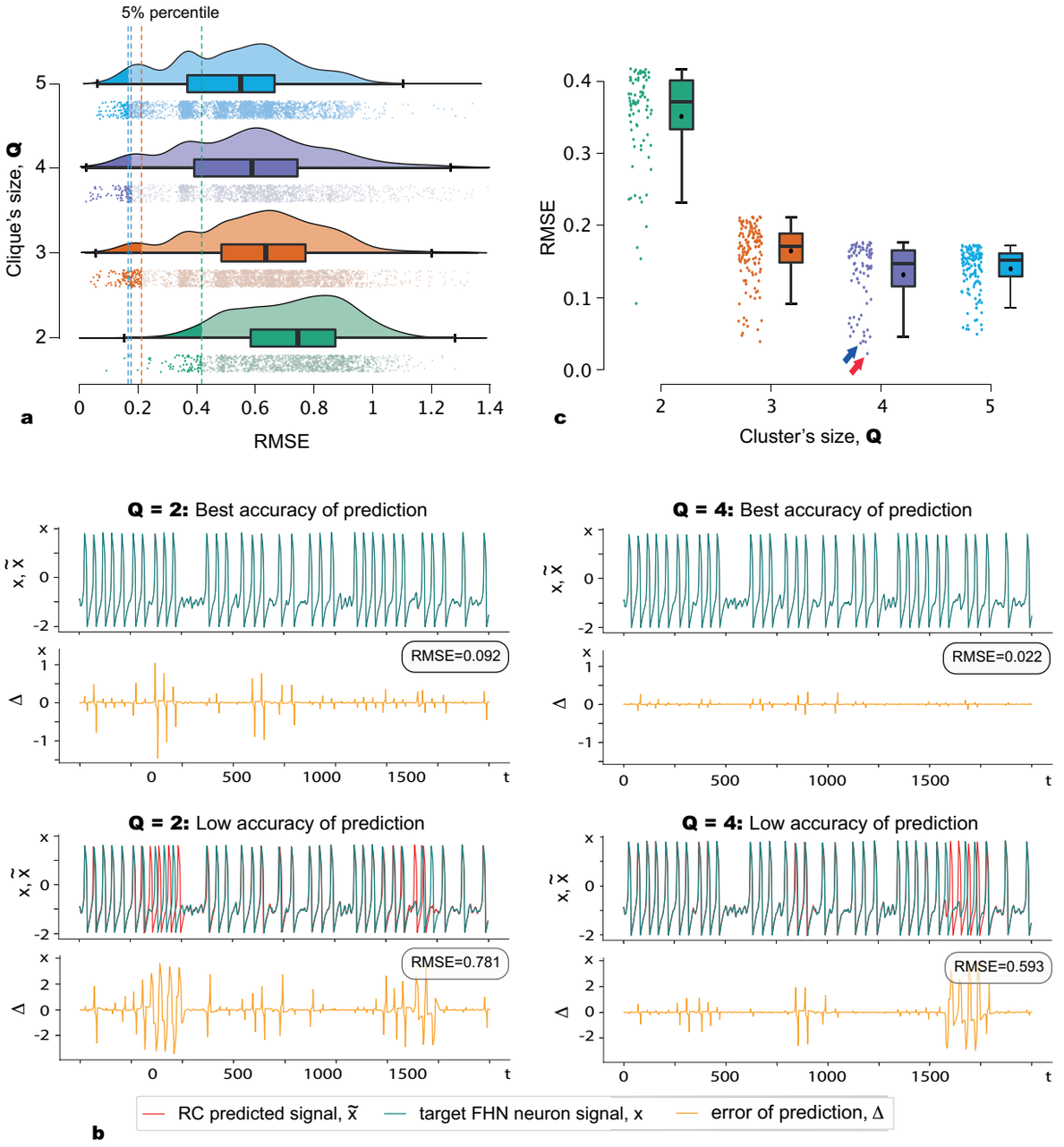

FIG. 3. **Prediction accuracy for RC with clique-based topology.** **(a)** Distributions of prediction accuracy (RMSE values) for the four considered types of RC: a traditional reservoir with traditional topology ($Q = 2$), and CB-RCs with triangles ($Q = 3$), tetrahedrons ($Q = 4$), and pentachorons ($Q = 5$). The vertical lines denote the fifth percentile of the RMSE distributions, representing the best predictions. **(b)** Time series of the target signal $x(t)$, predicted signal $\tilde{x}(t)$, and prediction error $\Delta(t)$ for the RCs with the best prediction quality (minimum RMSE) and with the low prediction quality (mean value of the RMSE distribution) for $Q = 2$ and $Q = 4$. **(c)** RMSE values for the top 5% (dots on the left) of the RCs, represented by the dots on the left, and a box-and-whiskers plot (on the right) of this set of RMSE values across four different values of clique's size Q. The arrows in panel **(c)** indicate the two best reservoirs in terms of minimum RMSE. In the boxplot, the horizontal line represents the median, and the dot represents the mean value of the distribution.

viations in the spike onset times. In particular, the RMSE for the case of $Q = 4$ is more than four times smaller than that of the traditional reservoir with $Q = 2$. This is also clearly visible on the $\Delta(t)$ plot. On the other hand, macroscopic errors in the predicted spike sequence are visible for low prediction accuracy, although the RMSE remains consistently lower for the RC with cliques. In summary, it is evident that the transition to the RC with clique-based topology results in generally

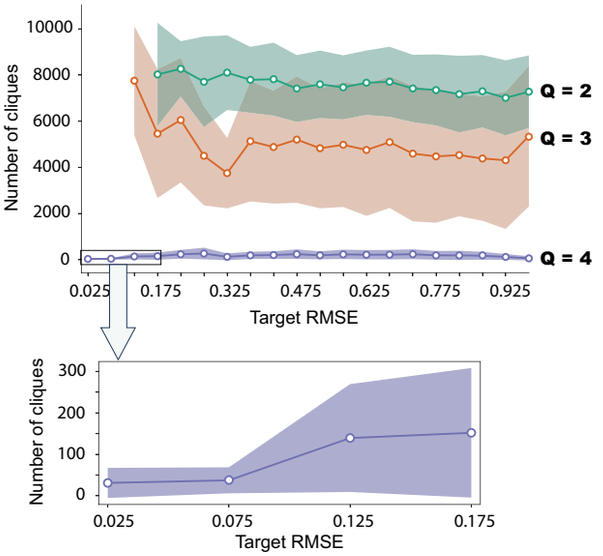

FIG. 4. **The number of cliques with different size $Q = 2, 3, 4$ required to achieve a given target forecasting accuracy (RMSE value).** The x-axis represents the target RMSE value, while the y-axis shows the mean number of cliques ($\pm$ SD) needed to meet that RMSE threshold. The data were obtained by binning the RMSE values from all simulation runs into intervals of ΔRMSE=0.05. The zoomed-in panel for $Q = 4$ highlights that high accuracy (RMSE<0.075) is achievable with only a few cliques.

increased prediction accuracy.

To illustrate this situation more clearly, we selected the fifth percentile of the RMSE distribution, representing the 5% best RCs in terms of minimum RMSE, for each clique's size Q. The corresponding percentiles are highlighted in Fig. 3b by vertical dashed lines and light colored points. Fig. 3c shows the individual RMSE values for each of the top 5% of RCs, represented by dots on the left, and the box-and-whiskers plot of this set of RMSE values, shown on the right. It can be seen that the reservoirs in which the hidden layer is equipped with all-to-all connected structures offer significantly higher prediction accuracy compared to the reservoir with traditional topology. Furthermore, an additional, albeit smaller, increase in accuracy is evident when the clique's size is increased from $Q = 3$ to $Q = 4$. Further increasing of Q up to 5 does not lead to any significant changes: accuracy remains almost the same as for $Q = 4$.

However, increasing the clique's size Q also leads to another notable consequence: a rapid decrease in the number of cliques required to achieve a given accuracy as measured by RMSE. This phenomenon is illustrated by Fig. 4, which shows the number of cliques with size Q in the CB-RC hidden layer for different clique's size $Q = 2, 3$, and 4 (see Appendix A for details). To construct this plot, we discretized the RMSE values from all 2,850 simulation runs into intervals of 0.05 and calculated the mean number of cliques corresponding to each RMSE threshold. It can be observed that the number of cliques shows a minimal dependence on the accuracy (RMSE value), but a strong dependence on the clique's size Q. Therefore, it is difficult to realize the RC on traditional topology or 3-cliques with high accuracy. The highest achievable accuracy of RMSE < 0.075 is obtained using an CB-RC with cliques of size 4. In this case, the number of components required is minimal, averaging about 30 4-cliques. At RMSE ≈ 0.125, common links are insufficient to achieve the required prediction accuracy for an RC; the average number of 3-cliques required is about 8,000, while the number of 4-cliques is less than 150. At lower prediction accuracies, the ratio between the number of links and cliques is such that the required number of 3-cliques is approximately 1.6 times smaller than the number of common links, and the required number of 4-cliques is approximately 50 times smaller than the number of common links.

TABLE I. Structural characteristics of reservoirs with different clique sizes Q. Mean values and standard deviations (std) are presented. P-values were obtained using the Mann-Whitney test (two-tailed).

Measure	**Q=2**		**Q=3**		**Q=4**		**p-value**		
	mean	std	mean	std	mean	std	Q=2 vs Q=3	Q=2 vs Q=4	Q=3 vs Q=4
Clustering coefficient	0.013	0.003	0.025	0.006	0.068	0.023	< 0.001	< 0.001	< 0.001
Transitivity	0.059	0.012	0.092	0.012	0.508	0.239	< 0.001	< 0.001	< 0.001
Modularity	0.178	0.022	0.630	0.289	0.823	0.091	< 0.001	< 0.001	0.005
EDF	453.1	50.5	300.7	166.7	193.5	115.8	< 0.001	< 0.001	< 0.001

To clarify the functional role of our design, we also analyzed how the proposed construction parameter Q changes the structural characteristics of the reservoir graphs. We computed standard network measures and observed a statistically significant increase in the clustering coefficient, transitivity and modularity and decreasing effective degrees of freedom (EDF) as Q increases from 2 to 3 and 4 (Table I).

By extracting fixed-size cliques from the initial dense network, the reservoir topology consistently shifts toward a more locally organized and modular structure[14,34]. This is reflected in an increase in the clustering coefficient and transitivity, that is, an increased proportion of triangles and short connection cycles, as well as in increased modularity. From a dynamic perspective, increased clustering means that the influence of an input persists in the vicinity of connected nodes over several subsequent steps, forming short-lived traces of activity and maintaining short-term memory without requiring global signal propagation throughout the network[15,35].

It's important to emphasize that this reorganization acts as a structural regularization of the reservoir. Due to more pronounced local organization, the states of nodes within clusters become more consistent, and the state space becomes more structured. A decrease in EDF indicates that the representation is becoming more compact: multiple reservoir states contribute similar information and therefore do not increase the effective complexity of the linear reader[10,11,36].

With further increases in order, modularity becomes excessive, and the network increasingly fragments into weakly interacting substructures. Despite the fact that input is fed to all nodes, the emerging modules form partially autonomous dynamics that are less well coordinated with each other. In this regime, local short-term activity traces cease to be effectively combined into a unified representation, and structural regularization devolves into excessive localization, leading to

a decrease in prediction quality at higher orders, despite continued growth in clustering[14,37].

Our results can be summarized as follows. (i) As illustrated in Fig. 3, we observe a significant increase in accuracy (decrease in RMSE value) when moving from traditional topology ($Q=2$) to triangles ($Q=3$), while moving to tetrahedrons ($Q=4$) gives a marginal further increase in accuracy. (ii) As can be seen from Fig. 4, the number of cliques in the hidden layer needed to achieve a given accuracy threshold (RMSE) decreases by a factor of 1.6 when transitioning from traditional topology to triangles, and then drastically decreases by a factor of about 30 when moving from triangles to tetrahedrons. The results suggest that, in terms of accuracy and number of cliques in the hidden layer, it would be beneficial to reconsider the architectural approach used in constructing the hidden layer of RCs, incorporating fully connected components with size $Q=4$. As we have shown on Fig. 3c, increasing Q from 4 to 5 does not improve the prediction quality of the RC, and the maximum accuracy is achieved for $Q=4$. So, the CB-RC with 4-cliques is optimal in terms of both prediction accuracy and the small number of fully connected components required. It is unclear whether an optimal clique's size would be related to the dimensionality of the phase space of the input data. Further investigation is required to tackle this issue.

C. Efficiency of CB-RCs in terms of topological and information characteristics

Let us now examine the features and mechanisms underlying the enhanced RC operation in the presence of fully connected components. For this purpose, we focus on the two most efficient CB-RCs with $Q=4$ and minimal RMSE (marked by red and blue arrows in Fig. 3b for $Q=4$). Fig. 5a illustrates the topology of the hidden layer of an RC based on the aforementioned principles. The topology is fundamentally different from that of a traditional RC. First, as mentioned above, there are many isolated nodes in the hidden layer, specifically $I_1=354$ in the first RC and $I_2=456$ in the second RC. This is due to the relatively low value of $\langle k\rangle$, which limits the density of links and results in many nodes not being part of cliques. Second, the number of cliques formed is relatively small, with $N_{Q1}=58$ for the first reservoir and $N_{Q2}=12$ for the second, and the number of nodes forming these cliques is considerably smaller than the number of isolated nodes, with $D_1=147$ and $D_2=45$. Moreover, the cliques are organized into several unconnected components, enumerated in Fig. 5a as $1,2,\ldots,10$. The most efficient RC, giving the absolute minimum RMSE, is observed to have a giant connected component 1 containing 38 cliques and nine other components containing small numbers of cliques. For the second most accurate CB-RC, the hidden layer is mainly composed of single unconnected cliques, except for component 3, which contains three cliques.

How would the proposed hidden layer architecture deliver a performance benefit? To answer this question, we consider its constituent elements separately. For comparison, Fig. 5b shows the signals obtained by excluding different parts of the internal layer. The top two time series correspond to the signal of neuron $x(t)$ (green line) and the signal $\tilde{x}(t)$ (red line) predicted by the CB-RC with the topologies depicted in Fig.5a. What is the role of the isolated nodes? To answer this question, we can exclude all the cliques from the hidden layer and train the RC using only the isolated nodes. The resulting time series realization, $x_{\mathrm{IN}}(t)$ (blue color), is shown in the third panel of Fig. 5b. It is clear that the reservoir based only on isolated nodes learns to reproduce the spike shape quite well, generating a sequence of spikes at the output, but whose order does not correspond correctly to that in the predicted neuronal signal $x(t)$. Thus, we conclude that the RC with cliques separates the processing of information for prediction. We postulate that a significant number of isolated nodes reproduce the shape of the spike, while the connected network components containing the simplices ensure the precise timing of generation for these spikes through enhanced integration and processing of the input information, including the processing of the noise signal $D\boldsymbol{\xi}^t$ within the simplices, where the information from each node is averaged according to Eq.(4). A comprehensive study of these representation mechanisms is reserved for future studies.

What is the role of each component in increasing the accuracy of the prediction of the system dynamics? To answer this question, we apply the following procedure. We iteratively remove each component, retrain the output weights, and evaluate the prediction accuracy by calculating the RMSE of the reservoir with the remaining components. The left panels of Fig. 5c show the RMSE value of the prediction accuracy of the CB-RC when the corresponding component is removed. If the RMSE value is relatively small and comparable to the prediction accuracy of the original CB-RC with all components, this would indicate that the component in question does not have a significant influence on the overall prediction accuracy. Conversely, if the RMSE increases significantly when the component is removed, it can be inferred that the component contributes to effective prediction and should therefore be considered influential. In Fig. 5c, all the components are ranked based on their influence, quantified by the magnitude of the increase in RMSE observed when a given component is removed. It can be seen that there are only a few influential components in both reservoir configurations, three in the first case and only one in the second.

Concurrently, the alterations in entropy characteristics $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$ are estimated for each component as illustrated in Fig. 5c (right panels in both columns). Here, S_R represents the redundant entropy, while S_S denotes the synergistic entropy[38,39]. The superscript "0" denotes the case of isolated nodes (absence of links), whereas the superscript "*" denotes the case of presence of links [the detailed description of calculating the entropy measures is presented in A 5]. In the context of networks, redundancy refers to information duplicated across many network nodes, such that the same information could be obtained by observing any of the nodes. In contrast, synergy refers to information that is accessible only by considering the joint states of multiple nodes, rather than simpler combinations of sources. The color of the circles in

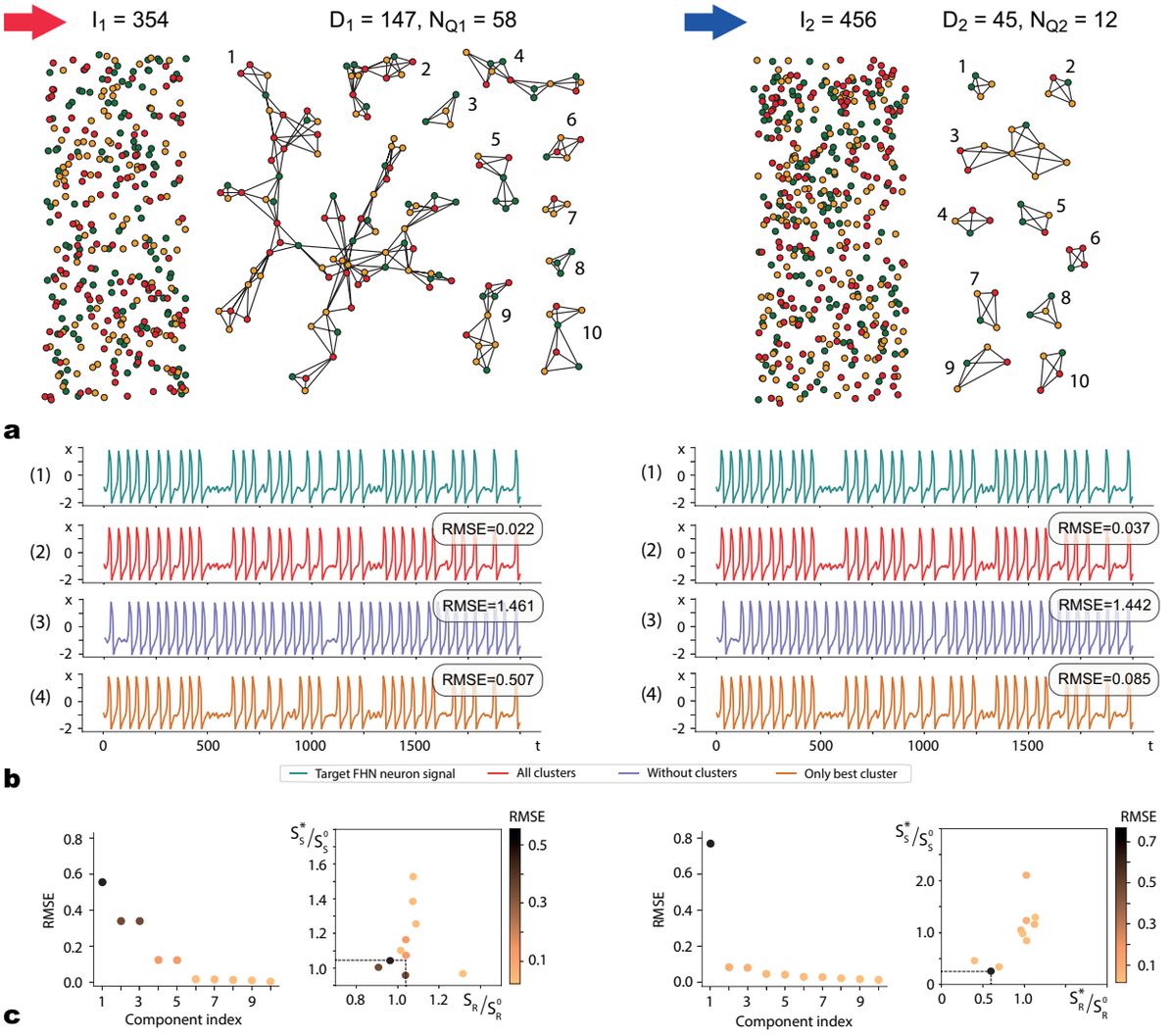

FIG. 5. **Characteristics of the two most efficient CB-RCs with 4-cliques.** **(a)** The topologies of the hidden layer of the two most efficient CB-RC with 4-cliques (marked by red and blue arrows in Fig. 3b). The color of each node, as in Fig. 2a, denotes the reservoir input to which it is connected: red nodes receive the noise $D\xi^t$, green nodes receive the variable x^t and orange nodes receive y^t. The hyperparameters are $\lambda_1 = 1.9$, $\langle k\rangle_1 = 13$ and $\lambda_2 = 1.9$, $\langle k\rangle_2 = 10$. **(b)** The $x(t)$ time series of (1) the predicted (target) signal of a stochastic neuron, (2) a trained reservoir with I isolated nodes and N_Q cliques including D nodes, (3) an RC with only isolated nodes remaining and all cliques removed, and (4) an RC with all isolated nodes remaining and the best-performing clique. **(c)** The left panel in both columns shows the RMSE value of the prediction accuracy of the CB-RC when the component is removed. The components are ordered by their influence, namely, by the magnitude of the increase in RMSE when a given component is removed. The right panels in both columns illustrate the changes in entropy characteristics $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$. The color of the circles corresponds to the RMSE value when this component is removed. The dashed lines indicate the most influential components, whose removal has the greatest impact on the accuracy of the CB-RC prediction. Compared to isolated nodes with the same inputs, these components are characterized by the smallest increments of redundant entropy and synergistic entropy measures during component formation.

Fig. 5c corresponds to the RMSE value when this component is removed (see the color bars in Fig. 5c). These RMSE values correspond to the values shown next to each other in the left panels in Fig. 5c. It can be observed that the most influential components of the hidden layer, whose removal significantly affects the accuracy of the CB-RC prediction, exhibit the smallest increments of redundant entropy and synergistic entropy measures during component formation compared to isolated nodes with the same input influences given by the corresponding elements of the matrix $\mathbf{G}$. Thus, in both examples under consideration, there is one influential component that is distinguished by simultaneous minimum values of S_R^*/S_R^0 and S_S^*/S_S^0, which fall within the approximate range of $[0.6, 1.0]$. At the same time we should note, that other cliques with similar entropy ratios are far less significant. Thus, we observe a one-way correlation between a clique's importance and its

entropy: while most significant cliques have entropies within certain ranges, having such entropies does not guarantee significance.

Can a viable reservoir be constructed using only isolated nodes and the most important component? Fig. 5b (bottom panel) illustrates this situation, where a reservoir configuration includes only the first most influential component and isolated nodes. In this case, the prediction accuracy is lower than a reservoir comprising the complete set of components but still acceptable and higher than in the case of a reservoir with traditional topology. It is important to note that in the second case, the first component is a 4-clique, and the prediction accuracy is considerably higher compared to the first case, where a complex component harbors a large number of cliques.

In light of these results, we hypothesize that it is possible to purposefully design a CB-RC based on a single Q-clique and an appropriate number of isolated nodes in the hidden layer. This proposition will be elaborated in the next section.

D. Designing a CB-RC based on a single Q-clique

Fig. 6 illustrates the flowchart for designing a CB-RC containing a single Q-clique. First, we choose the size of the hidden-to-hidden reservoir layer, which contains $N = I + Q$ nodes, among which I isolated nodes and Q nodes form a Q-clique. Second, we generate different configurations of the Q-clique. These include different input variables affecting the nodes forming the Q-clique, as well as different weights of the input-to-hidden matrix $\mathbf{G}$ describing the effects of the inputs on the Q-clique nodes. Third, we compute the change in the entropy characteristics $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$ and find the variant of the Q-clique that is characterized by the minimum change in the entropy characteristics. We then train the corresponding reservoir (i.e. the one with the identified Q-clique). Another possibility is to minimize the number I of isolated nodes in the reservoir, possibly by an iterative approach (illustrated in the last box of the flowchart in Fig. 6), while retraining the initial nodes, as long as the CB-RC prediction accuracy RMSE(I) remains acceptable for the problem at hand.

We will now apply this approach to the benchmark for predicting stochastic neuron dynamics introduced in Section III. We consider a reservoir designed a priori as a single 4-clique in combination with a set of $I = 500$ isolated nodes. We evaluate the feasibility of an approach to optimal clique selection based on the study of the entropy characteristics of individual cliques. For this purpose, 1,200 CB-RCs based on a single 4-clique were randomly generated. This involved selecting four nodes within a reservoir hidden layer at random and their connection into a 4-clique with random link strength from a uniform distribution $[0.5, 2.5]$. The set of input links was fixed and selected from the top two CB-RCs with 12 cliques. For each reservoir generated, the entropy characteristics of the clique and the root mean square error (RMSE) are calculated. The 5% of the reservoirs with the lowest RMSE values were selected as the best performing reservoirs (threshold of RMSE $= 0.18$). Fig. 7a illustrates the distribution density of the entropy features for the selected top 5% clique configurations. It can be observed that the majority of CB-RCs with high classification quality have values of S_R^*/S_R^0 and S_S^*/S_S^0 within the range $(0.6, 1.4)$. This suggests that any clique can be chosen as a working clique in an RC with such entropy change characteristics.

Based on the found optimal values of S_R^*/S_R^0 and S_S^*/S_S^0 we suggest an algorithm to faster construction of optimal RC's topology with only one fully-connected component and a number of isolated nodes which produces the desired RMSE which was chosen as 0.18. Standard algorithm will be as follows: (i) obtain hidden state of RC which consists of only isolated nodes; (ii) randomly create a Q-clique and obtain the corresponding signals from Q hidden neurons; (iii) train the model; (iv) test the model; (v) compare the resulting RMSE with the desired one and finish the algorithm or return to step (ii). We suggest to add an intermediate step (iia) "calculate the entropy measures and check for inclusion in the optimal range", which plays the role of filter: we chose the cliques with the desired entropy measures and train and test the RC with only them. The calculation of the entropy measures takes only 0.716 ± 0.067 seconds (over 10000 iterations) while to perform the regression for model training and then test it we need to spend 5.14 ± 0.414 seconds (over 3000 iterations). We test the two approaches for $Q = 4$ for 10 random seeds and find that our algorithm is on average 1.72 times faster than the standard one.

We then investigated the issue of CB-RC prediction accuracy as the number I of isolated nodes decreases. The results are visible in Fig. 7b, which shows that the initial number of isolated nodes was excessive, with a significant proportion being redundant. The prediction accuracy remains relatively stable when the number is reduced to 100. For example, if we choose 0.1 as the RMSE threshold, a reservoir consisting of a 4-clique and 89 isolated nodes is sufficient. This model is remarkably more compact than a traditional RC, yet it offers comparable prediction accuracy.

E. Predictive capability of a CB-RC with a single Q-clique

In this section, we introduce two benchmark models that are used to demonstrate the predictive performance of CB-RCs with a single Q-clique. As a first prediction benchmark problem, we generate training and test data by numerically integrating the Lorenz system[40], which is a well-known double-scroll chaotic system, consisting of a set of three coupled nonlinear differential equations (for the system's variables x, y, and z) given by

$$\dot{x} = 10(y - x), \;\; \dot{y} = (28 - z)x - y, \;\; \dot{z} = xy - \frac{8}{3}z. \qquad (9)$$

As a second benchmark problem, we investigate the use of the CB-RC with a single 4-clique to predict the dynamics of a Rössler system[41], which is another paradigmatic chaotic system with a single-scroll topology. The behavior of this system is governed by

$$\dot{x} = -(y + z), \;\; \dot{y} = x + 0.2y, \;\; \dot{z} = 0.2 + z(x - 5.7). \qquad (10)$$

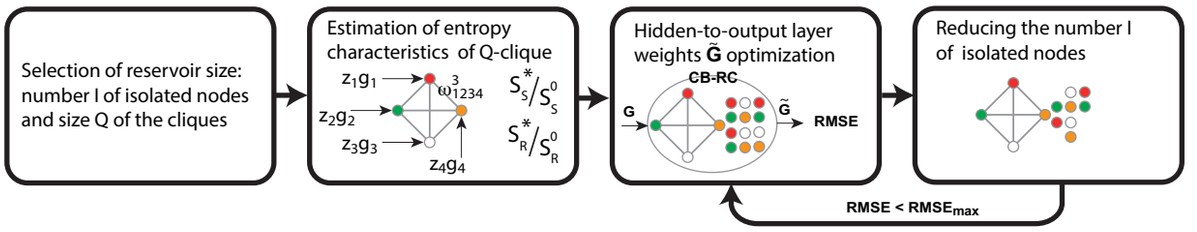

FIG. 6. **The schematic flowchart of designing a CB-RC based on a single Q-clique.**

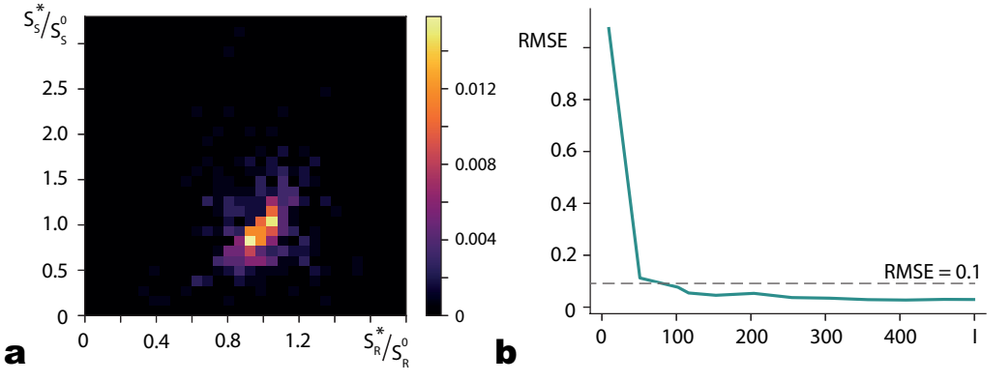

FIG. 7. **Stochastic neuron dynamics prediction by a CB-RC based on hidden layer containing a single 4-clique and I isolated nodes. (a)** The distribution density of entropy characteristics $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$ of the selected top 5% clique configurations. To obtain this distribution, the range of redundant entropy relations $[0, 1.8]$ is divided into 30 segments of length 0.06, the range of synergetic entropy relations $[0, 3.3]$ is divided into 30 segments of length 0.11, and the number of cliques falling into each segment is normalized to the total number (1,200) of simplices. **(b)** Prediction accuracy (RMSE) of the stochastic neuron dynamics as a function of the number of isolated nodes, I.

For the chosen parameter values, these systems generate deterministic chaos, resulting in the formation of attractors with markedly distinct topologies, as shown in Fig. 8a,b for the Lorenz system and Fig. 8e,f for the Rössler system.

CB-RCs containing a single 4-clique and 500 isolated nodes were generated to predict the dynamics. Specifically, a set of 15,000 clique configurations was formulated based on 30 sets of input weights and 500 cliques for each set with random coupling strengths drawn from a uniform distribution $[0.5, 2.5]$. We then evaluated the entropy characteristics of each clique S_R^*/S_R^0 and S_S^*/S_S^0 and searched for those characterized by a minimal change in the redundant entropy and synergistic entropy measures.

The results obtained are shown in Fig. 8d for the Lorenz system and in Fig. 8h for the Rössler system. Figs. 8d and 8h show the distribution of the top 5% of the best CB-RCs in the coordinates of the entropy characteristics of a single clique $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$.

Reservoirs with the clique, selected for minimum entropy change, and 500 isolated nodes were trained using 8,000 and 25,000 samples, respectively, for the Lorenz and Rössler systems. Fig. 8a,e show the dynamics of the target and predicted x, y and z variables for both chaotic systems. For a short interval reflecting the Lyapunov time, the predicted and actual signals behave almost identically, and then the trajectories diverge inexorably. A qualitative inspection of the predicted strange attractors (Fig. 8c for the Lorenz and Fig. 8g for the Rössler system) and the actual strange attractors (Fig. 8b and f for the two systems, respectively) reveals a striking similarity between the two, indicating that the CB-RC is capable of reproducing the long-term dynamics of both. Thus, our approach, which involves the purposeful design of the CB-RC interior based on the entropy approach, has been demonstrated to be effective in predicting the dynamics of low-dimensional chaotic systems. The results indicate that a simple hidden layer comprising a small number of isolated nodes and a single 4-clique allows an efficient and reasonably accurate prediction that is comparable to that of traditional RCs.

IV. APPLICATION TO THE RECONSTRUCTION OF EXPERIMENTAL BRAIN SIGNALS

Finally, we use experimental data to illustrate the potential of using CB-RC to reconstruct the signal of one electroencephalography (EEG) channel based on information from adjacent channels. This situation was chosen as an empirical challenge because of the complex interdependencies between EEG channels resulting from the fact that each channel receives weighted contributions from multiple cortical regions, alongside the high-dimensional and stochastic under-

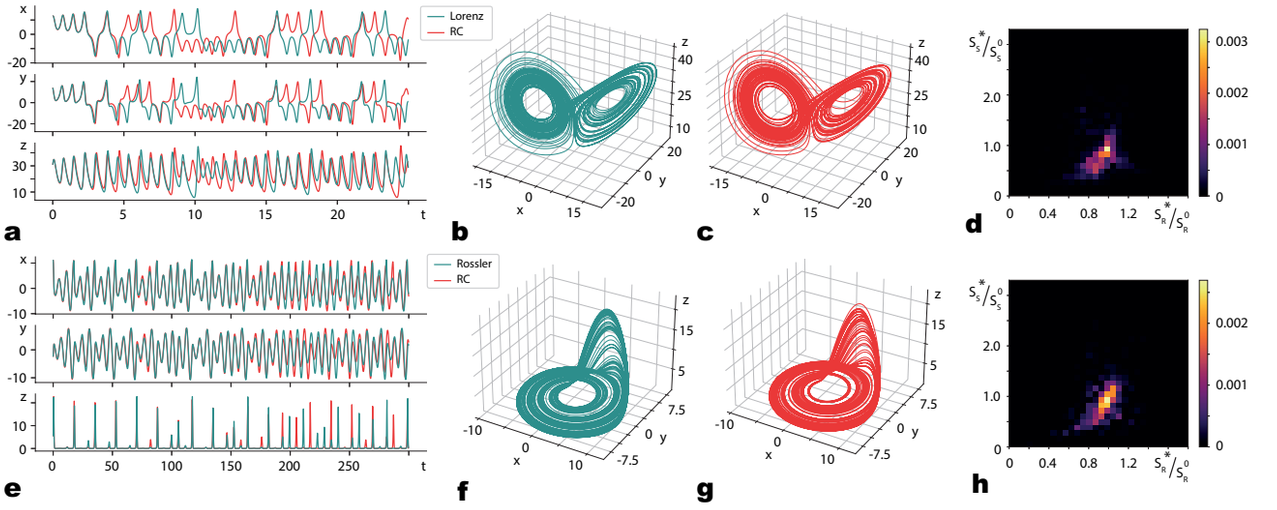

FIG. 8. **Forecasting chaotic systems dynamics using the CB-RC with a single 4-clique (a)** Target (turquoise) and predicted (red) time series of Lorenz system. **(b)** True and **(c)** predicted Lorenz attractors. **(d)** Distribution density of the entropy characteristics $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$ of the selected top 5% cliques for the CB-RC for Lorenz oscillator dynamics prediction. **(e)** Target (turquoise) and predicted (red) time series of Rössler system. **(f)** True and **(g)** predicted Rössler attractors. **(h)** Distribution density of the entropy characteristics $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$ of the selected top 5% cliques for the CB-RC for Rössler system dynamics prediction.

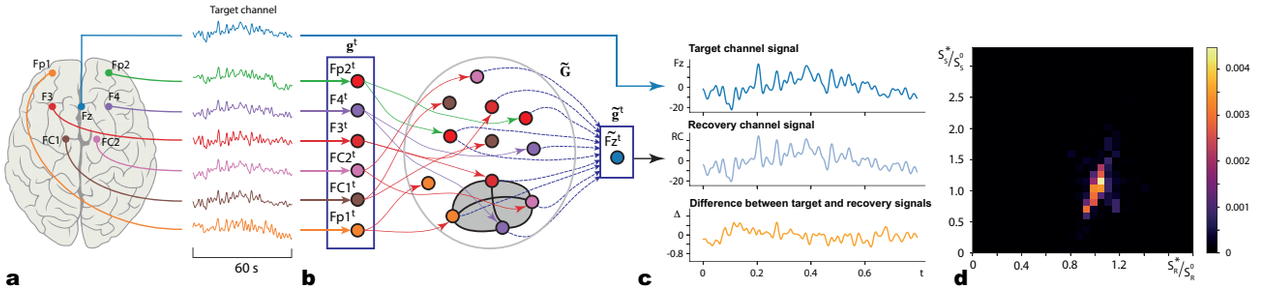

FIG. 9. **Reconstructing the EEG channel signal using the CB-RC with single 4-clique. (a)** Location of the EEG electrodes (namely, Fp1, Fp2, F3, F4, FC1, FC2) whose experimental signals were used to recover a hidden temporal variable (Fz channel signal). **(b)** Schematic representation of the CB-RC that was used to recover the unknown signal. **(c)** Target experimental signal Fz (top), the signal reconstructed with CB-RC (middle), and the difference between the target and recover signals (bottom) in time. **(d)** Density distribution of entropy characteristics $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$ of the selected top 5% cliques.

lying neural dynamics. To this end, we analyzed two-minute recordings of resting-state background neuroelectrical activity made with the eyes open in healthy participants (see details of data acquisition in the Appendix A below).

To limit the dimensionality of the problem (as the objective here is merely to illustrate a concept), we arbitrarily selected seven neighboring channels from the frontal lobe (Fp1, Fp2, F3, F4, FC1, FC2, and Fz, as shown in Fig. 9a). We called the Fz channel a "hidden" channel, with the aim of reconstructing its signal based on data from the remaining six channels. The corresponding configuration is shown in Fig. 9b. The input signals, in the form of a data set, are derived from channels Fp1, Fp2, F3, F4, FC1, and FC2 and are the input to CB-RC. Only one signal, corresponding to the hidden channel Fz, is considered at the output. In this instance, we do not take into account the short-circuit of the input and output of the memory by a feedback loop in the reconstruction mode. After training the output weights of the reservoir by minimizing the loss function in Eq. (2) as described above (see Section II), the CB-RC is then fed with experimental data and the quality of the reconstruction is analyzed using the RMSE metric (6). The training set consisted of 60 s of EEG recordings used to train the output layer of the CB-RC. The subsequent 60 s of data were used to assess the accuracy of the reconstruction.

Fig. 9c illustrates an example of hidden channel reconstruction. The original (target) experimental signal Fz, the signal reconstructed with CB-RC, and the difference between them are plotted. The error at each time point of the hidden variable reconstruction does not exceed 6% with the best RMSE is around 0.004. The CB-RC model had a single 4-clique and

500 isolated nodes.

We calculated the change in entropy characteristics to estimate the best cliques for reconstructing the experimental data $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$. The same binarisation approach to signals used in the case of predicting chaotic signals was used here to calculate the entropies [Eq. (A5)]. We found that, as before, the best-performing cliques are those characterized by a minimal change in the redundant entropy and synergistic entropy measures. This is illustrated in Fig. 9d, which shows the distribution of the top 5% of the most effective CB-RCs in the coordinates of the entropy characteristics of a single clique $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$. The distribution was computed by generating 20 random sets of input weights, for each of which 500 cliques with random coupling strength were generated from a uniform distribution $[0.5, 2.5]$. A total of 10,000 cliques were used to construct the distribution. The distribution in Fig. 9d shows the fifth percentile of the total distribution, corresponding to the threshold RMSE $= 2.09$. Based on this example, we confirm that the proposed paradigm performs remarkably well on experimental data obtained from large complex systems, such as the human brain.

V. DISCUSSION

In recent years, there has been a notable increase in research focused on developing more efficient learning mechanisms and innovative reservoir architectures. The principal objective has been to construct reservoirs with extensive and intricate state spaces, thereby facilitating more efficacious processing of intricate temporal data. For example, see Gauthier *et al.* [17] demonstrated the potential of employing a nonlinear autoregressive vector as a reservoir state space, which markedly reduces the computational burden during training. Similarly, in a further extension of the power of reservoir computing, Kong, Brewer, and Lai [42] explored the use of high-dimensional reservoir states to encode and manipulate complex dynamical attractors. Previously, the advantages of sparse reservoirs with small number of connections have been previously discussed in[21]. We suggest constructing a specific topological architecture based on predefined rules – namely, forming small all-to-all connected groups of the fixed size based on the initial adjacency matrix and introduce the interaction inside it via a mean-field principle. These structures, such as triangles and tetrahedrons, provide a considerably richer framework for representing multi-node dependencies[25]. Our study introduces a CB-RC model that retains the essential characteristics of classical random reservoirs while introducing a robust framework for efficiently modeling and predicting dynamical systems, regardless of their underlying complexity or nature.

A noteworthy feature of the proposed model is its minimalistic configuration, which contrasts with the intricate architectures often associated with high-performance reservoirs. Despite its simplicity, the model attains a level of predictive accuracy that surpasses traditional reservoirs. Moreover, we demonstrate that the effective operation of the reservoir can be maintained by retaining a single fully connected component and a set of isolated neurons, with desirable implications for physical reservoir computing, for example, with regards to the number of metal layers necessary for signal routing in integrated circuits and circuit boards[24,43]. This reduction in complexity not only minimizes the hardware requirements but also ensures the preservation and, in some cases, even the enhancement of the computational capabilities of the reservoir.

As evidenced in Ref. [21], the challenge is determining which elements to link and how to arrange these links to maintain high prediction accuracy. Our analysis indicates that the most effective configurations can be identified by examining the entropy characteristics of connected and unconnected nodes. Our information-theoretical analysis enables the synergistic and redundant structures of these nodes to remain stable when combined into a clique, with entropy ratios consistently falling in the 0.8-1.2 range which means that both characteristics do not change when a clique is formed. Stability is essential, as it enables accommodating more intricate interactions from the input signals through a clique without altering the overall entropy. This explains why even a single clique combined with isolated nodes can outperform traditional dense reservoirs - the clique structure inherently captures the essential nonlinear signal features while eliminating superfluous connections.

In paper[44] Li et al. demonstrated the utility of higher-order interactions in reservoir computing by inferring such structures from data. Our work takes a complementary approach by prescribing specific clique-based topologies (e.g., 3-cliques and 4-cliques) and quantifying their advantages. Our results show that these predefined structures not only enhance prediction accuracy but also enable more compact hardware implementations. In the previous theoretical works[11,45] the authors has shown that pairwise-coupled reservoirs are universal approximators, our results demonstrate that specific simple topology with small number of connections can significantly enhance practical performance. Specifically, clique-based topologies achieve comparable or better accuracy with fewer couplings, making RC advantageous for real-world applications where hardware constraints or training efficiency are critical.

Our architecture, based on cliques of varying sizes, builds on and extends the research on structured topologies in reservoir computing. The seminal work of Jarvis et al.[34] and its predecessors (e.g.,[46]) established that clustered and hierarchical reservoir organization is a powerful means of enhancing its dynamic range and robustness, overcoming the severe spectral radius limitations ($\rho < 1$) inherent in classical random topologies. Our results empirically support this thesis, demonstrating that structured clustering leads to improved data representation and processing. However, unlike approaches where cliques are loosely connected or specialized on individual signal components (e.g.,[47]), our model strategically emphasizes full connectivity within cliques. This key difference, as our systematic analysis shows when combined with varying clique sizes significantly reduces the system's sensitivity to the choice of a specific global topology. Thus, our work offers a solution to one of the central problems identified in earlier studies: it combines the high performance provided by clique organization with increased generality and practical applicability, paving the way for the creation of more compact, effi-

cient, and robust physical implementations of reservoir computers.

In summary, we have introduced a novel reservoir computing model that enhances prediction accuracy by incorporating cliques with all-to-all connected nodes (CB-RC), namely, cliques with 3 and 4 elements. Reducing the number of required links and nodes accompanies this performance improvement. The clique topology's advantages are robust across diverse applications from stochastic neuron prediction to chaotic systems and real-world EEG analysis, demonstrating its general applicability. Through both theoretical analysis and extensive numerical validation, we show how this architecture achieves an optimal tradeoff between memory capacity, nonlinear processing, and hardware efficiency - making it both fundamentally interesting for statistical physics and immediately relevant for practical implementations. The potential applications encompass, for example, the development of dynamical models in brain science and the formulation of optimal control systems in engineering. Future work should deploy the proposed method to large-scale datasets and quantify performance and implementation complexity advantages in concrete applications, including low-power and real-time systems.

ACKNOWLEDGMENTS

This work was supported by Russian Science Foundation (Grant No. 23-71-30010). L.M. gratefully acknowledges the support of the "Hundred Talents" program of the University of Electronic Science and Technology of China, of the "Outstanding Young Talents Program (Overseas)" program of the National Natural Science Foundation of China, and of the talent programs of the Sichuan province and Chengdu municipality. S.B. acknowledges support from the project n.PGR01177 of the Italian Ministry of Foreign Affairs and International Cooperation.

DATA AVAILABILITY STATEMENT

The experimental data used in this study can be found in the repository: https://github.com/allivore/ho-interactions-rc/tree/main/Data

CODE AVAILABILITY STATEMENT

The code used in this study can be found in the repository: https://github.com/allivore/ho-interactions-rc

Appendix A: Appendix A: Methods

1. Numerical methods used

For the numerical solution of the stochastic FitzHugh-Nagumo equations (Eq. 7), we used the Euler-Maruyama method, with an integration time step $\Delta t = 0.1$. The differential equation systems of Lorenz (Eq. 9) and Rössler (Eq. 10) were instead solved numerically using a fourth-order Runge-Kutta method with fixed integration time steps $\Delta t = 0.01$ and $\Delta t = 0.02$, respectively.

2. EEG brain data acquisition

In order to reconstruct the time series of a hidden electrode in brain activity data, we used two-minute recordings of the background electrical activity of the brain in the resting state with open eyes recorded in healthy volunteers. Twenty-one healthy subjects (9 male and 12 female, aged between 18 and 26 years) participated in the experiments. All subjects provided written informed consent prior to participation. The experimental procedures were conducted in accordance with the principles of the Declaration of Helsinki and approved by the local Research Ethics Committee of Immanuel Kant Baltic Federal University (protocol No 32 from July 04, 2022). EEG signals were recorded using an actiCHamp system (Brain Products, Germany) for 63 channels according to the "10—10" scheme, with a sampling frequency of 1,000 Hz. Only minimal preprocessing was performed; namely, the signals were filtered in the range [1, 40] Hz using a band-pass finite impulse response (FIR) filter.

3. Finding cliques of a given size in a graph

To identify all cliques of a specified size within a graph, we employed the algorithm proposed by Zhang, et al.[28], which was adapted and implemented in the Python package NetworkX.

4. Estimation of the number of cliques required to achieve the desired forecasting accuracy

2,850 RC configurations with different hyperparameter values were used to compute the dependencies shown in Fig. 4. After the training and testing phases, each corresponding RC is characterized by its RMSE value. To facilitate a comparative analysis of the number of cliques comprising an RC across different clique's sizes and their corresponding RMSE values, the following procedure was applied: the RMSE range from 0 to 1 was divided into 20 equal intervals of length $\Delta\text{RMSE} = 0.05$. Then, for each interval, the number of cliques of each size that make up the reservoir whose RMSE falls within this range was calculated. Then, for each ith range of RMSE $(\Delta\text{RMSE}(i-0.5), \Delta\text{RMSE}(i+0.5))$, $i = 1, \ldots .20$, a set of reservoirs was obtained for which the number of cliques was calculated for each clique's size $(Q = 2, 3, 4)$. From this set, the mean and the standard deviation for each Q as a function of $i\Delta$RMSE were finally calculated.

5. Entropy measures

To quantify the information relevance of different components in the CB-RC hidden layer, we propose an approach based on two entropy measures, redundant entropy (S_R) and synergistic entropy (S_S), which enable detecting information structures in complex networks[48,49] and in particular functional brain networks[39,50]. As noted in Ref. [39], redundant and synergistic information exchange between nodes in the subset $\{h_1, h_2, \ldots, h_Q\}$ of N reservoir nodes represent two distinct but related types of interaction. In this context, redundancy refers to information that is duplicated across many network nodes, such that the same information could be gathered by observing $h_1 \vee h_2 \vee \cdots \vee h_Q$. Synergy refers to information accessible only by considering the joint states of multiple nodes, rather than simpler combinations of sources. Synergistic information can only be gathered by observing $h_1 \wedge h_2 \wedge \cdots \wedge h_Q$.

In the case of a clique with size $Q = 4$, redundant and synergistic entropy measures can be computed as

$$\begin{aligned} S_R(h_1,h_2,h_3,h_4) = {} & H_\partial^{1-4}(\{1\}\{2\}\{3\}\{4\}) + \\ & + H_\partial^{1-4}(\{1\}\{2\}\{3\}) + H_\partial^{1-4}(\{1\}\{2\}\{4\}) + \\ & + H_\partial^{1-4}(\{1\}\{3\}\{4\}) + H_\partial^{1-4}(\{2\}\{3\}\{4\})) \end{aligned} \quad \text{(A1)}$$

and

$$\begin{aligned} S_S(h_1,h_2,h_3,h_4) = {} & H_\partial^{1-4}(\{1,2,3\}\{1,2,4\}\{1,3,4\}) + \\ & + H_\partial^{1-4}(\{1,2,3\}\{1,2,4\}\{1,3,4\}\{2,3,4\}) + \\ & + H_\partial^{1-4}(\{3,4\}\{1,2,4\}) + H_\partial^{1-4}(\{1,3\}\{1,4\}\{2,4\}) + \\ & + \cdots + H_\partial^{1-4}(\{1\}\{2\}\{3,4\}) + H_\partial^{1-4}(\{1\}\{3,4\}), \end{aligned} \quad \text{(A2)}$$

where H_∂^{1-4} is the partial entropy function for the tetrahedron $\mathbf{h} = \{h_1, h_2, h_3, h_4\}$ (see Refs. [38,39] for details). For example, $H_\partial^{1-4}(\{1\}\{2\}\{3\})$ refers to the information redundantly shared by h_1, h_2, and h_3, when these are considered as part of $\mathbf{h}$. When calculating S_S, 140 combinations of H_∂^{1-4} are considered, therefore, only some of them are given in Eq. (A2).

The partial entropy function determines the unique information about the whole system $\mathbf{h}$ contained in the considered elements (in our case, nodes) and not in the others. To obtain H_∂, one must compute the informative and misinformative pointwise shared information of an element, apply Möbius inversions to it, and compute the difference between the informative and misinformative ones (see Ref. 38 for details).

In the case of $Q = 4$, the measures in Eq. (A1) and Eq. (A2) are computed for each set of four nodes forming a clique, for two cases: in the absence of links $\left(S_R^0, S_S^0\right)$ (isolated nodes) and in the presence of links $\left(S_R^*, S_S^*\right)$ (clique). Then we consider a value equal to their ratio $\left(S_R^*/S_R^0, S_S^*/S_S^0\right)$, which allows us to estimate the change in the information characteristics of the considered nodes when forming a clique. When cliques are connected, the presence/absence of the entire connected component of the graph is analyzed. All entropy properties are averaged over all cliques of this component, with

$$S_{R,S} = \frac{1}{J} \sum_{j=1}^{J} S_{R,S}^j, \quad \text{(A3)}$$

where J is the number of cliques in the connected component.

Since these measures work well with discrete variables $h^H(t)$[38], we discretize the signals of each reservoir neuron, $h(t)$, by binarization, depending on the system being analyzed.

For the FitzHugh-Nagumo neuron, information is primarily carried by the spike events, so we first determine z-scores based on the output of each i-th neuron within the reservoir, determine the derivative of the evolution of the spike (increasing or decreasing), and set all values greater than the standard deviation $\sigma(h_i)$ for an increasing spike to 1, the rest to 0, and less than $-\sigma(h_i)$ equal to 1 for a decreasing spike, the rest equal to 0, as follows

$$h_i^H(t) = \begin{cases} 1, & \text{if } ((h_i(t) \geq \sigma(h_i) \wedge \\ & (|\max(h_i) - \overline{h}_i| \geq |\min(h_i) - \overline{h}_i|)) \vee \\ & ((h_i(t) \leq -\sigma(h_i)) \wedge \\ & (|\max(h_i) - \overline{h}_i| < |\min(h_i) - \overline{h}_i|)), \\ 0, & \text{if } ((h_i(t) < \sigma(h_i)) \wedge \\ & (|\max(h_i) - \overline{h}_i| \geq |\min(h_i) - \overline{h}_i|)) \vee \\ & ((h_i(t) > -\sigma(h_i)) \wedge \\ & (|\max(h_i) - \overline{h}_i| < |\min(h_i) - \overline{h}_i|)). \end{cases} \quad \text{(A4)}$$

Binarized time series h^H described by Eq. (A4) with 500,000 points are used to estimate the entropy characteristics.

To binarize the data in the case of the Lorenz and the Rössler systems calculating the entropies, we first apply a z-score to the output of each i-th neuron within the clique, then set all positive values to 1 and the rest to 0:

$$h_i^H(t) = \begin{cases} 1, & \text{if } h_i(t) \geq 0, \\ 0, & \text{if } h_i(t) < 0. \end{cases} \quad \text{(A5)}$$

All partial information and entropy decompositions were performed using the SxPID package released by Ref. [38].

[1] R. C. Hilborn, *Chaos and nonlinear dynamics: an introduction for scientists and engineers* (Oxford university press, 2000).
[2] K. Nakajima and I. Fischer, *Reservoir computing* (Springer, 2021).
[3] M. Yan, C. Huang, P. Bienstman, P. Tino, W. Lin, and J. Sun, "Emerging opportunities and challenges for the future of reservoir computing," Nature Communications **15**, 2056 (2024).
[4] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data," Chaos: An Interdisciplinary Journal of Nonlinear Science **27** (2017).
[5] T. L. Carroll, "Using reservoir computers to distinguish chaotic signals," Physical Review E **98**, 052209 (2018).
[6] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," Physical review letters **120**, 024102 (2018).
[7] G. 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).
[8] A. E. Hramov, N. Kulagin, A. V. Andreev, and A. N. Pisarchik, "Forecasting coherence resonance in a stochastic fitzhugh–nagumo neuron model using reservoir computing," Chaos, Solitons & Fractals **178**, 114354 (2024).
[9] G. C. McDonald, "Ridge regression," Wiley Interdisciplinary Reviews: Computational Statistics **1**, 93–100 (2009).
[10] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," IEEE Transactions on Circuits and Systems **32**, 1150–1161 (1985).

[11]L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," Neural Networks **108**, 495–508 (2018).
[12]L. Gonon and J.-P. Ortega, "Reservoir computing universality with stochastic inputs," IEEE transactions on neural networks and learning systems **31**, 100–112 (2019).
[13]E. Bollt, "On explaining the surprising success of reservoir computing forecaster of chaos? the universal machine learning dynamical system with contrast to var and dmd," Chaos: An Interdisciplinary Journal of Nonlinear Science **31** (2021).
[14]M. Dale, S. O'Keefe, A. Sebald, S. Stepney, and M. A. Trefzer, "Reservoir computing quality: connectivity and topology," Natural Computing **20**, 205–216 (2021).
[15]Y. Kawai, J. Park, and M. Asada, "A small-world topology enhances the echo state property and signal propagation in reservoir computing," Neural Networks **112**, 15–23 (2019).
[16]S. K. Rathor, M. Ziegler, and J. Schumacher, "Asymmetrically connected reservoir networks learn better," Physical Review E **111**, 015307 (2025).
[17]D. J. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, "Next generation reservoir computing," Nature communications **12**, 1–8 (2021).
[18]Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," Chaos: An Interdisciplinary Journal of Nonlinear Science **28** (2018).
[19]J. Platt, H. Abarbanel, S. Penny, A. Wong, and R. Clark, "Robust forecasting through generalized synchronization in reservoir computing," in *AGU Fall Meeting Abstracts*, Vol. 2021 (2021) pp. NG25B–0522.
[20]L. A. Thiede and U. Parlitz, "Gradient based hyperparameter optimization in echo state networks," Neural Networks **115**, 23–29 (2019).
[21]A. Griffith, A. Pomerance, and D. J. Gauthier, "Forecasting chaotic systems with very low connectivity reservoir computers," Chaos: An Interdisciplinary Journal of Nonlinear Science **29** (2019).
[22]L. Livi, F. M. Bianchi, and C. Alippi, "Determination of the edge of criticality in echo state networks through fisher information maximization," IEEE transactions on neural networks and learning systems **29**, 706–717 (2017).
[23]P. Antonik, N. Marsal, D. Brunner, and D. Rontani, "Bayesian optimisation of large-scale photonic reservoir computers," Cognitive Computation , 1–9 (2021).
[24]D. D. Yaremkevich, A. V. Scherbakov, L. De Clerk, S. M. Kukhtaruk, A. Nadzeyka, R. Campion, A. W. Rushforth, S. Savel'ev, A. G. Balanov, and M. Bayer, "On-chip phonon-magnon reservoir for neuromorphic computing," Nature Communications **14**, 8296 (2023).
[25]A. Patania, F. Vaccarino, and G. Petri, "Topological analysis of data," EPJ Data Science **6**, 1–6 (2017).
[26]J. 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: An Interdisciplinary Journal of Nonlinear Science **28** (2018).
[27]Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," Chaos: An Interdisciplinary Journal of Nonlinear Science **27** (2017).
[28]Y. Zhang, F. N. Abu-Khzam, N. E. Baldwin, E. J. Chesler, M. A. Langston, and N. F. Samatova, "Genome-scale computational approaches to memory-intensive applications in systems biology," in *SC'05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing* (IEEE, 2005) pp. 12–12.
[29]L. Qi, H. Chen, and Y. Chen, *Tensor eigenvalues and their applications*, Vol. 39 (Springer, 2018).
[30]L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," Nature communications **2**, 468 (2011).
[31]L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," Physical Review X **7**, 011015 (2017).
[32]M. Roy, S. Mandal, C. Hens, A. Prasad, N. Kuznetsov, and M. Dev Shrimali, "Model-free prediction of multistability using echo state network," Chaos: An Interdisciplinary Journal of Nonlinear Science **32** (2022).
[33]E. M. Izhikevich, *Dynamical systems in neuroscience* (MIT press, 2007).
[34]S. Jarvis, S. Rotter, and U. Egert, "Extending stability through hierarchical clusters in echo state networks," Frontiers in neuroinformatics **4**, 11 (2010).
[35]K.-i. Kitayama, "Guiding principle of reservoir computing based on "small-world" network," Scientific reports **12**, 16697 (2022).
[36]J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," Scientific reports **2**, 514 (2012).
[37]N. Rodriguez, E. Izquierdo, and Y.-Y. Ahn, "Optimal modularity and memory capacity of neural reservoirs," Network Neuroscience **3**, 551–566 (2019).
[38]A. Makkeh, A. J. Gutknecht, and M. Wibral, "Introducing a differentiable measure of pointwise shared information," Physical Review E **103**, 032149 (2021).
[39]T. F. Varley, M. Pope, M. Grazia, Joshua, and O. Sporns, "Partial entropy decomposition reveals higher-order information structures in human brain activity," Proceedings of the National Academy of Sciences **120**, e2300888120 (2023).
[40]E. N. Lorenz, "Deterministic nonperiodic flow," Journal of atmospheric sciences **20**, 130–141 (1963).
[41]O. E. Rössler, "An equation for continuous chaos," Physics Letters A **57**, 397–398 (1976).
[42]L.-W. Kong, G. A. Brewer, and Y.-C. Lai, "Reservoir-computing based associative memory and itinerancy for complex dynamical attractors," Nature Communications **15**, 4840 (2024).
[43]J. Moon, W. Ma, J. H. Shin, F. Cai, C. Du, S. H. Lee, and W. D. Lu, "Temporal data classification and forecasting using a memristor-based reservoir computing system," Nature Electronics **2**, 480–487 (2019).
[44]X. Li, Q. Zhu, C. Zhao, X. Duan, B. Zhao, X. Zhang, H. Ma, J. Sun, and W. Lin, "Higher-order granger reservoir computing: simultaneously achieving scalable complex structures inference and accurate dynamics prediction," Nature Communications **15**, 2506 (2024).
[45]S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," IEEE Transactions on circuits and systems **32**, 1150–1161 (1885).
[46]Z. Deng and Y. Zhang, "Collective behavior of a small-world recurrent neural system with scale-free distribution," IEEE Transactions on neural networks **18**, 1364–1375 (2007).
[47]Y. Xue, L. Yang, and S. Haykin, "Decoupled echo state networks with lateral inhibition," Neural Networks **20**, 365–376 (2007).
[48]R. A. Ince, "The partial entropy decomposition: Decomposing multivariate entropy and mutual information via pointwise common surprisal," arXiv preprint arXiv:1702.01591 (2017).
[49]A. J. Gutknecht, M. Wibral, and A. Makkeh, "Bits and pieces: Understanding information decomposition from part-whole relationships and formal logic," Proceedings of the Royal Society A **477**, 20210110 (2021).
[50]T. F. Varley, M. Pope, J. Faskowitz, and O. Sporns, "Multivariate information theory uncovers synergistic subsystems of the human cerebral cortex," Communications biology **6**, 451 (2023).