

RESEARCH ARTICLE | JUNE 16 2020

Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems

Zhixin Lu  ; Danielle S. Bassett 



Chaos 30, 063133 (2020)

<https://doi.org/10.1063/5.0004344>

 CHORUS



View
Online



Export
Citation

Articles You May Be Interested In

Fractional order-induced bifurcations in a delayed neural network with three neurons

Chaos (March 2023)

Training stiff neural ordinary differential equations with implicit single-step methods

Chaos (December 2024)

An implicit lipid model for efficient reaction-diffusion simulations of protein binding to surfaces of arbitrary topology

J. Chem. Phys. (September 2019)

Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems

Cite as: Chaos 30, 063133 (2020); doi: 10.1063/5.0004344

Submitted: 10 February 2020 · Accepted: 25 May 2020 ·

Published Online: 16 June 2020



View Online



Export Citation



CrossMark

Zhixin Lu and Danielle S. Bassett^{a)}

AFFILIATIONS

Department of Bioengineering, School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, Pennsylvania 19104, USA

^{a)}**Also at:** Department of Physics and Astronomy, College of Arts and Sciences, University of Pennsylvania, Philadelphia, PA, 19104; Department of Electrical and Systems Engineering, School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA, 19104; Department of Neurology, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, 19104; Department of Psychiatry, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, 19104; and Santa Fe Institute, Santa Fe, NM, 87501. **Author to whom correspondence should be addressed:** dsb@seas.upenn.edu

ABSTRACT

Regardless of the marked differences between biological and artificial neural systems, one fundamental similarity is that they are essentially dynamical systems that can learn to imitate other dynamical systems whose governing equations are unknown. The brain is able to learn the dynamic nature of the physical world via experience; analogously, artificial neural systems such as reservoir computing networks (RCNs) can learn the long-term behavior of complex dynamical systems from data. Recent work has shown that the mechanism of such learning in RCNs is *invertible generalized synchronization* (IGS). Yet, whether IGS is also the mechanism of learning in biological systems remains unclear. To shed light on this question, we draw inspiration from features of the human brain to propose a general and biologically feasible learning framework that utilizes IGS. To evaluate the framework's relevance, we construct several distinct neural network models as instantiations of the proposed framework. Regardless of their particularities, these neural network models can consistently learn to imitate other dynamical processes with a biologically feasible adaptation rule that modulates the strength of synapses. Further, we observe and theoretically explain the spontaneous emergence of four distinct phenomena reminiscent of cognitive functions: (i) learning multiple dynamics; (ii) switching among the imitations of multiple dynamical systems, either spontaneously or driven by external cues; (iii) filling-in missing variables from incomplete observations; and (iv) deciphering superimposed input from different dynamical systems. Collectively, our findings support the notion that biological neural networks can learn the dynamic nature of their environment through the mechanism of IGS.

© 2020 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0004344>

Recent studies have shed light on the mechanisms by which reservoir computers can predict the future of dynamical systems. Notably, the human brain, and possibly the neural systems of many other species, can also imitate complicated dynamical patterns after learning from exemplars. In this work, we take the perspective of dynamical systems and posit that the mechanism underlying learning in reservoir computers may also be exploited in biological neural systems. Our theory not only explains the emergence of learning in biological systems but also predicts several related functions, such as multi-tasking, task switching, and parsing mixed signals.

I. INTRODUCTION

The human brain is essentially a dynamical system that evolves non-autonomously according to physical principles. One of its particularly notable functions is the capacity to imitate some other dynamical process solely by being exposed to the output of that process without ever formally encoding the governing equations.^{1–6} To gain a physical intuition for this phenomenon, we can consider a few pertinent examples from the perspective of dynamical systems. Imitating a dynamical system with a fixed point or limit cycle attractor corresponds to memorizing and reciting an exemplary trajectory. Imitating a more complex dynamical system with, for example, a

quasi-periodic torus or chaotic attractor corresponds to inventing new trajectories that are distinct from the original exemplary trajectory while still obeying its underlying dynamical rules. Practically, this same phenomenon manifests in the human's ability to envision the orbits of physical objects based on previous observations of the physical world, the human's ability to speak grammatically correct sentences simply by listening to others, and the human's ability to hum a familiar or invented melody in their mind. While we may often take these capabilities for granted, their underlying mechanism remains far from understood.

Notably, the capacity for dynamical systems to imitate other dynamical systems generalizes beyond humans and, in fact, even beyond biological systems. Indeed, over the last two decades intricate machine learning models and algorithms have been engineered to learn complex dynamics from data.^{7–25} Such models and algorithms come in many flavors, with a particularly powerful type being reservoir computing (RC) networks, also known as echo-state networks or liquid state machines. RC networks demonstrate a marked capacity to learn and persistently imitate various dynamical systems. For example, RC networks can learn (i) the recorded periodic movement of a human running,²³ (ii) low-dimensional chaotic systems such as the Lorenz or Rössler attractors,^{22,24} and even (iii) the Kuramoto–Sivashinsky system which is governed by partial differential equations that exhibit spatiotemporal chaos.²⁵ It was recently shown in Ref. 26 that RC networks learn to imitate other dynamical systems by capitalizing on the power of invertible generalized synchronization (IGS). Briefly, as the RC is driven by an input signal $s(t)$ on an attractor \mathcal{A} , it encodes \mathcal{A} into its own phase space $\mathbf{x} \in \mathbb{R}^n$ as a homeomorphism $\mathcal{P} = \psi(\mathcal{A})$. Then, the output of the RC is trained through supervised learning algorithms such that it can reproduce the target dynamics by autonomously evolving on the attained \mathcal{P} when the driving trajectory $s(t)$ is later removed.

We posit that the invertible generalized synchronization previously derived from RC²⁶ is a general principle that allows not only artificial neural systems but also biological systems, such as the human brain, to persistently imitate and reproduce the behavior of a target dynamical system after implicitly assimilating only example trajectories. In this work, however, we do not seek to directly identify IGS in experimental systems with observable measurements due to current methodological obstacles in comprehensive sampling. Rather, we show that IGS arises naturally in a general learning framework that we construct based on previous observations of cognition and behavior in humans. In contrast to RC, the framework is further characterized by a biologically relevant adaptation rule in which synaptic weights are iteratively tuned. Ongoing empirical works^{27–32} suggest that music and language follow definitive rules of composition or grammar, respectively, in a manner that is conceptually akin to distinctive trajectories on a chaotic attractor. Thus, without great loss of generality, we confine our discussion to cases where the dynamics being learned is definite: the target dynamical system is autonomous and its trajectory $s(t) \in \mathbb{R}^n$ on an attractor \mathcal{A} . By exercising this mechanism in a range of *in silico* experiments in Secs. III–IV, we observe and explain the spontaneous emergence of several other phenomena reminiscent of human cognitive functions: (i) the ability to switch between imitating different dynamical patterns, either spontaneously³³ or when driven by an external stimulus,³⁴ (ii) the capacity to fill in missing information

from sequential input;^{35,36} and (iii) the ability to resolve distinct rules underlying mixed inputs generated from distinct structures.^{37–40} The proposed learning mechanism based on invertible generalized synchronization provides a concise theory that unifies diverse learning functions.

II. INFORMING THE LEARNING FRAMEWORK WITH UNDERLYING BIOLOGICAL FEATURES

A. Attractor encoding by invertible generalized synchronization

We begin by considering the reinstatement hypothesis, which posits that the content-specific activity in the human cortex at the time of encoding is reinstated as the encoded information is being retrieved.^{41,42} Recent experimental evidence supports this idea, suggesting that patterns of neural activity initially encode sensory input and then those same patterns of neural activity recur when the encoded information is retrieved.^{43–45} In building our mathematical framework, we parallel this reinstatement hypothesis by recruiting a central dynamical system, $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \cdot)$, where sensory information is initially encoded and from which sensory information can later be retrieved (Fig. 1).

We consider the exemplary sensory input trajectory $s(t) \in \mathbb{R}^n$ being generated by an autonomous input dynamical system,

$$\mathbf{s}(t+1) = \mathbf{g}(\mathbf{s}(t)). \quad (1)$$

For input dynamical systems that are continuous in time, $\mathbf{g}(\cdot)$ can be considered as the flow map which takes s forward in time by a time step τ . During the learning phase [Fig. 1(a)], the central system is

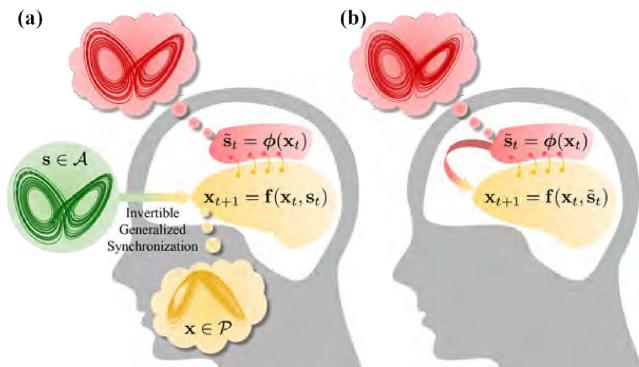


FIG. 1. Schematic depiction of our learning framework. (a) During the learning phase, the mind encodes the observed dynamics as it evolves on attractor \mathcal{A} . The mind is operationalized as a central dynamical system here shown in yellow, where sensory information is initially encoded and from which sensory information can later be retrieved. The exemplary sensory input trajectory being generated by an autonomous input dynamical system is shown in green. The internally generated output that imitates the external sensory input is shown in red. (b) During the testing phase, the mind reproduces the learned dynamics as it evolves on attractor \mathcal{P} . The internally generated signal is directed into the central system through a feedback loop (shown by the curved arrow) that is recruited during the testing phase.

driven by $\mathbf{s}(t)$, and evolves following

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{s}(t)), \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^N$, and $N \gg 1$.

In this setup, the autonomous input system [Eq. (1)] and the non-autonomous central system [Eq. (2)] are one-way coupled and collectively form a drive-response system. Through invertible²⁶ generalized synchronization,^{55–57} the central system becomes synchronized to the input system in a particular way. Specifically, starting from any initial state, the $\mathbf{x}(t)$, after a transient time, becomes uniquely determined by the state of the input system,

$$\mathbf{x}(t) = \psi(\mathbf{s}(t)), \quad (3)$$

and evolves onto $\mathcal{P} = \psi(\mathcal{A})$, where $\psi : \mathcal{A} \rightarrow \mathcal{P}$ is invertible (bijective between \mathcal{A} and \mathcal{P}). The attractor \mathcal{P} , an embedding of \mathcal{A} in \mathbb{R}^N , plays the role of the central system's internal representation of \mathcal{A} .

Invertible generalized synchronization allows the central dynamical system to embed a dynamical attractor \mathcal{A} into its own phase space. However, it is important to note that invertible generalized synchronization is not generically possible for any arbitrary pair of one-way coupled dynamical systems. In the context of the brain, this non-universality means that the human brain may not be able to learn any arbitrary dynamical system. To determine whether and when invertible generalized synchronization is possible between two coupled dynamical systems, we can consider the common criterion that the largest conditional Lyapunov exponent of the central system is negative (see Appendix A 1 for further details). As a consequence of the Whitney embedding theorem,⁵⁸ if generalized synchronization does occur, the $\psi(\cdot)$ in Eq. (3) is likely to be invertible as long as the dimension of the central system is large ($N \gg n$).²⁶

B. Internal imitation of the external signal

Complementing the central system that encodes the attractor \mathcal{A} , we consider an internally generated output that imitates the external sensory input \mathbf{s} . The internal imitator operationalizes the human capacity to enhance and consolidate learned information by interacting with working memory;^{59,60} colloquially, this phenomenon is sometimes called an *inner voice*.^{59–65} The external sensory signal \mathbf{s} that the central system encodes is internalized, and can be recreated even when there is no external signal.⁶⁵ We consider this process to be downstream of the central system, and formalize it as a decoding function,

$$\tilde{\mathbf{s}}(t) = \phi(\mathbf{x}(t)), \quad (4)$$

where $\tilde{\mathbf{s}} \in \mathbb{R}^n$ is the internally generated output. During the learning phase, the $\phi(\cdot)$ is adaptively modified towards the bijective function that maps \mathcal{P} back to \mathcal{A} , i.e.,

$$\phi^*(\mathbf{x}) \equiv \psi^{-1}(\mathbf{x}), \text{ for } \forall \mathbf{x} \in \mathcal{P}, \quad (5)$$

such that the internal imitator produces a trajectory that is concurrent with the external sensory input: $\tilde{\mathbf{s}}(t) = (\psi^{-1} \circ \psi)(\mathbf{s}(t)) = \mathbf{s}(t)$.

C. Retrieving and imitating the learned dynamics

Upon the completion of the learning phase, the whole system evolves autonomously due to the absence of the external input

signal \mathbf{s} . We refer to this period as a testing phase [Fig. 1(b)], during which the internal output $\tilde{\mathbf{s}}$ imitates the dynamical trajectory on the learned attractor \mathcal{A} , as the central system retrieves and reinstates its trajectories on \mathcal{P} . Here, we consider a specific form of information retrieval to constitute the testing phase. In the study of human memory, extra cortical regions are engaged when humans mentally rehearse and retrieve previously learned sensory information in the absence of exogenous input.^{43,66–69} Motivated by these observations, we let the internally generated $\tilde{\mathbf{s}}$ be directed into the central system through a feedback loop that is recruited only during the testing phase. As $\tilde{\mathbf{s}}$ replaces the absent external \mathbf{s} , the central system reinstates its behavior on \mathcal{P} . In this setup, the central system during the testing phase evolves autonomously without external input, following

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \tilde{\mathbf{s}}(t)) = \mathbf{f}(\mathbf{x}(t), \phi(\mathbf{x}(t))). \quad (6)$$

When the internal output function implemented in Eq. (6) is ideal, i.e., $\phi = \psi^{-1}$, any learning phase trajectory $\mathbf{x}(t)$ on \mathcal{P} [from the driven central system governed by Eqs. (1) and (2)], is also a solution of the autonomous central system equation [Eq. (6)]. Thus, we say that this N -dimensional autonomous central system incorporates the local dynamics of the n -dimensional dynamical system [Eq. (1)] on the attractor \mathcal{A} . Moreover, to guarantee that the internal output accurately and persistently imitates the learned dynamics on \mathcal{A} , the autonomous central system should evolve on \mathcal{P} in a manner that is transversely stable. To ensure that \mathcal{P} , an embedding of the low-dimensional dynamical attractor \mathcal{A} in the N -dimensional space, is indeed an attractor of the autonomous central system Eq. (6)], one can analyze its transverse stability by checking the sign of the corresponding transversal Lyapunov exponents. As previously shown,²⁶ the rate of divergence from the learned attractor during the autonomous mode can be predicted by the value of the largest transversal Lyapunov exponent (see Appendix A 2 for further details).

D. Instantiating the learning framework *in silico*

With this general dynamical framework, many complex learning functions naturally emerge, including learning multiple dynamics, switching between learned dynamics, filling-in missing variables, and separating superimposed signals. To study these functions, we instantiate the central system *in silico* as a high-dimensional dynamical system that responds consistently to external driving signals; this choice facilitates lossless encoding through invertible generalized synchronization in the learning phase. Here, we let the central system be a random RNN with sparse Erdős–Rényi topology akin to a typical reservoir computer.^{24,70,71} In Appendix C, we study other choices for the central system, including an RNN with a non-Erdős–Rényi topology and a dynamical network with random polynomial-type interactions.

Formally, we define the central system as

$$\mathbf{f}(\mathbf{x}, \mathbf{s}) = \tanh(\mathbf{A}\mathbf{x} + \mathbf{W}_{\text{in}}\mathbf{s} + \mathbf{c}), \quad (7)$$

where the vector hyperbolic tangent function applied on a vector $\mathbf{v} = [v_1, v_2, \dots]^T$ is defined as $\tanh(\mathbf{v}) = [\tanh v_1, \tanh(v_2), \dots]^T$. Here, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix of a sparse Erdős–Rényi network, $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N \times n}$ is the input weight matrix, and $\mathbf{c} \in \mathbb{R}^{N \times 1}$ is

a constant bias vector. While more complex approaches exist,⁷² we implement the internal output as a simple linear function

$$\phi(\mathbf{x}) = \mathbf{W}_{\text{out}} \mathbf{x}, \quad (8)$$

where \mathbf{W}_{out} is an n -by- N matrix. From a random initial \mathbf{W}_{out} , it is adapted during the learning phase according to a delta rule,

$$\mathbf{W}_{\text{out}}(t+1) = \mathbf{W}_{\text{out}}(t) + \alpha \Delta(t) \mathbf{x}^T(t), \quad (9)$$

such that the discrepancy between the input and the output, $\mathbf{s}(t) - \tilde{\mathbf{s}}(t) = \Delta(t) \in \mathbb{R}^{n \times 1}$, is gradually eliminated. This particular adaptive learning rule is inspired by the biological process of spike-time-dependent plasticity and is also used in other artificial learning systems.⁷³

III. MULTISTABILITY IN TASK SWITCHING BETWEEN MULTIPLE LEARNED RULES

Capitalizing on the power of invertible generalized synchronization, the embedding-based learning framework that we describe can be easily extended to learn multiple attractors, which are either chaotic or non-chaotic, denoted by $\mathcal{A}_k \subset \mathbb{R}^n$, where $k = 1, 2, \dots, K$. As a proof-of-principle experiment, we show in Fig. 2 a system that learns to imitate $K = 4$ attractors: one periodic, one quasi-periodic, and two chaotic (see Appendix B for further details). We refer to the imitation of each dynamical attractor as a task. The linear output weight matrix \mathbf{W}_{out} is adapted following Eq. (9) as the whole system repeatedly goes through the four learning phases (see Appendix D for further details regarding this multi-attractor learning schedule).

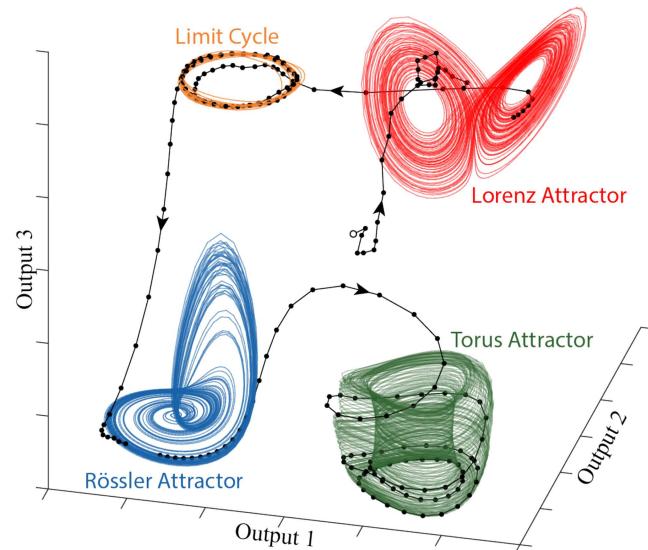


FIG. 2. Multistability in task switching between multiple learned rules. An example learning system with a trained \mathbf{s} that successfully imitates the dynamics of four different attractors. The black dotted trajectories are the system's output signals as it switches between different tasks. This switching is induced by external cues. The instantiated central system in this example is an RNN with $N = 2000$ nodes and an Erdős-Rényi topology.

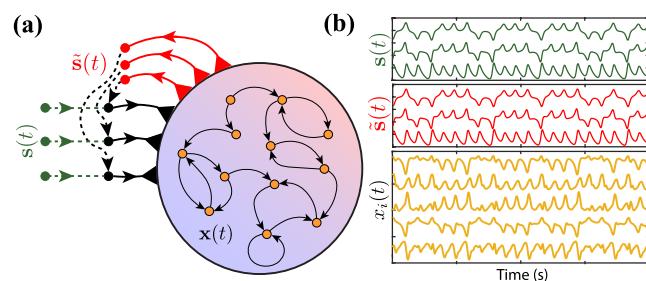


FIG. 3. Instantiating the learning framework *in silico*. (a) We let the central system be a random RNN with network size $N = 2000$ and sparse Erdős-Rényi topology akin to a typical reservoir computer. The output function (red) is adaptively updated following a simple delta rule that is inspired by the biological process of spike-timing-dependent plasticity. (b) In green, we show an exemplary external input trajectory $\mathbf{s}(t)$ generated by a Lorenz system. In red, we show the successfully trained concurrent internal output $\tilde{\mathbf{s}}(t)$ that imitates $\mathbf{s}(t)$. In yellow, we show the concurrent activity of five randomly chosen nodes in the central system $x_i(t)$.

While this learning framework can learn to imitate non-chaotic attractors, we are particularly interested in the learning of chaotic attractors, where the system does more than reciting a memorized exemplary trajectory. Thus, and also for simplicity and clarity of exposition, in the remainder of this paper, we only consider learning and imitating two chaotic attractors: the Lorenz and the Rössler attractors.

Formally, through invertible generalized synchronization in each of the K learning phases, the central system forms the corresponding internal representation by embedding $\mathcal{A}_k \subset \mathbb{R}^n$ into its own phase space \mathbb{R}^N . That is,

$$\mathcal{P}_k = \psi_k(\mathcal{A}_k), \quad (10)$$

where $\psi_k : \mathcal{A}_k \rightarrow \mathcal{P}_k$ is invertible. Given that $\mathcal{P}_k \cap \mathcal{P}_l = \emptyset$ and $\mathcal{A}_k \cap \mathcal{A}_l = \emptyset$ for any $k \neq l$, we combine the K generalized synchronization functions into a single invertible function $\psi : \bigcup_{k=1}^K \mathcal{A}_k \rightarrow \bigcup_{k=1}^K \mathcal{P}_k$. (See Appendix B 5 for a method to improve learning performance by avoiding overlap between \mathcal{P} s.) Then, the ideal output function is the inverse of ψ that maps each \mathcal{P}_k back onto \mathcal{A}_k , i.e., $\mathcal{A}_k = \phi^*(\mathcal{P}_k)$ for $k = 1, 2, \dots, K$. Given that $\phi = \phi^*$ in Eq. (6), if each representation \mathcal{P}_k is an attractor of the testing phase central system, the central system is able to imitate any of the K attractors. After implementing this setup *in silico*, we observe that the central system depicted in Fig. 3(a) can successfully learn to imitate both the Lorenz and the Rössler dynamics [Fig. 4(a)].

Once multiple attractors \mathcal{A}_k are learned, it is of interest to consider the phenomenon of task switching, which is a capacity displayed by many natural neural systems. When \mathcal{P}_k becomes an attractor of Eq. (6), we say that the attractor \mathcal{A}_k is stably learned, since the output trajectory $\tilde{\mathbf{s}}(t)$ cannot spontaneously depart from that attractor. In this case, we can consider explicitly triggering a switch from one task to another; that is, from the scenario in which the central system imitates one attractor to the scenario in which the central system imitates another attractor. Notice that the generalized synchronization guarantees that the central system state $\mathbf{x}(t)$ converges to $\psi(\mathbf{s}(t))$ after a short period of time in the testing phase.

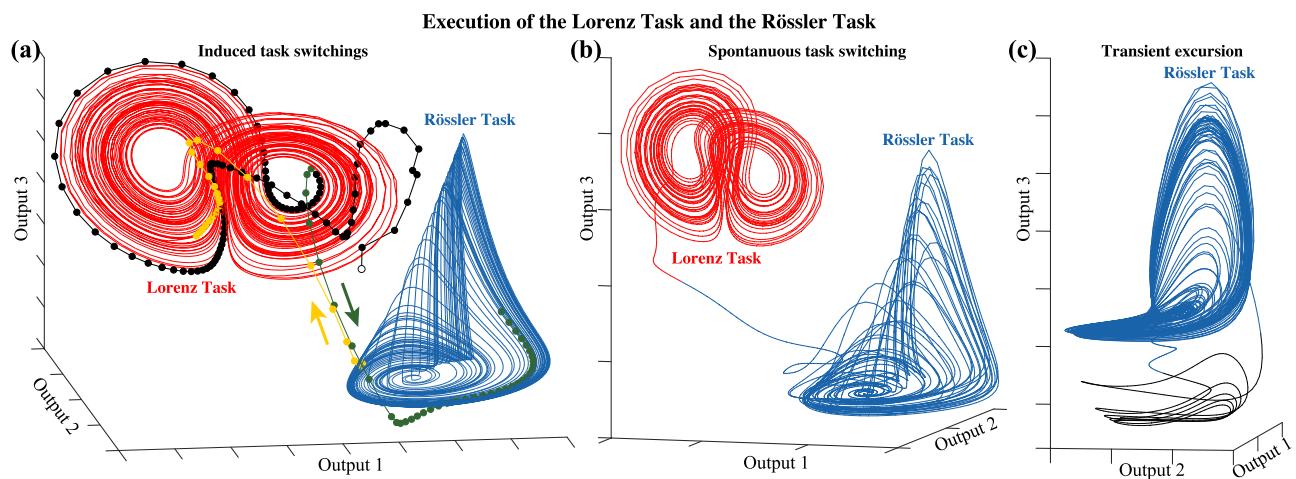


FIG. 4. Distinct types of task switching between multiple learned rules. (a) Output \tilde{s} generated by a learning system that successfully learns both the Lorenz and the Rössler attractors. Switching back and forth between two attractors can be induced by external cues. (b) Spontaneous task switching from one attractor that is not stably learned to another attractor that is stably learned. (c) A transient excursion observed when the Rössler attractor is not stably learned.

Thus, we can use a short external input cue, $s(t) \in \mathcal{A}_k$, and lead the central system x from an arbitrary state to the desired \mathcal{P}_k .

Instantiating these ideas *in silico*, we can begin the testing phase dynamics from a random initial state $x(t=0)$. In Fig. 4(a), the output $\tilde{s}(t)$ evolves following the black dotted line and converges onto the Lorenz attractor $\mathcal{A}_{\text{Lorenz}}$. Then, for a very short period of time, we provide an external input $s(t)$ from a Rössler system. We observed that this external input successfully induced a switch from $\mathcal{A}_{\text{Lorenz}}$ to $\mathcal{A}_{\text{Rössler}}$ (green dotted line). Thereafter, the central system evolves autonomously without any external input for a long period of time, generating the $\tilde{s}(t)$ that traces out the Rössler attractor. We then use a short Lorenz input $s(t)$ to drive the central system away from imitating the Rössler attractor and towards imitating the Lorenz attractor (yellow dotted line). Thereafter, the central system evolves autonomously with the testing phase dynamics, and generates a very long output trajectory that traces out the Lorenz attractor (red).

Next, it is interesting to consider the case in which not all attractors are stably learned. In this scenario, we would expect spontaneous task switching, independent of any external input. In Fig. 4(b), we consider a case in which the Lorenz system is stably learned but the Rössler system is not. That is, the autonomously evolving central system is transversely stable on $\mathcal{P}_{\text{Lorenz}}$ but transversely unstable on $\mathcal{P}_{\text{Rössler}}$. Here, we show that starting from an initial condition $x \in \mathcal{P}_{\text{Rössler}}$, the output of the testing phase \tilde{s} remains on the Rössler attractor (blue) for an extended period of time and then spontaneously departs from the Rössler attractor and converges onto the Lorenz attractor (red). Relatedly, it is interesting to consider the scenario in which an attractor is not stably learned. In Fig. 4(c), we observe that the output of the testing phase $\tilde{s}(t)$ exhibits a transient excursion as it unstably imitates the Rössler attractor, in contrast to the dynamics observed in spontaneous task switching.

IV. NETWORK MECHANICS UNDERLYING MULTIPLE LEARNED ATTRACTORS

Critically, because the learning of multiple attractors is achieved by embedding and incorporating multiple attractors into a single system [Eq. (6)], different imitation tasks can be executed simply by letting the central system state $x(t)$ revisit the corresponding \mathcal{P}_k . An important corollary of this fact is that no hyperparameters such as synaptic strengths are altered. We refer to the physical synapses as structural connections, and note that they are given by A . While we cannot explain the learning of multiple attractors with the pattern of structural connections, it is possible that there is explanatory content in the emergent pattern of functional connections, which are defined as statistical similarities in neuronal time series.

To investigate this possibility, we consider the structural connectivity given by the random adjacency matrix $A \in \mathbb{R}^{N \times N}$, and the functional connectivity given by the Pearson correlation matrix of the recorded activity $x(t) \in \mathbb{R}^{N \times 1}$ for t during the Lorenz (Rössler) task, $F_L \in \mathbb{R}^{N \times N}$ ($F_R \in \mathbb{R}^{N \times N}$). To summarize the emergent patterns of functional connectivity, we apply a commonly used community detection technique⁷⁴ known as modularity maximization⁷⁵ enacted with a particularly effective heuristic⁷⁶ to identify groups of neurons that show similar time series. We found strong but distinct community structure in F_L and F_R . In Fig. 5(e), we show F_L (red) and F_R (blue) with all nodes sorted by the community structure identified in F_L . Similarly, in Fig. 5(f), we show F_L (red) and F_R (blue) with all nodes sorted by the community structure identified in F_R . These observations indicate that while neurons remain identically structurally connected in both tasks, their emergent collective dynamics differ.

To quantify these observations more fully, we constructed 100 randomly organized and trained neural networks, and for each we calculated the average functional connectivity among pairs of

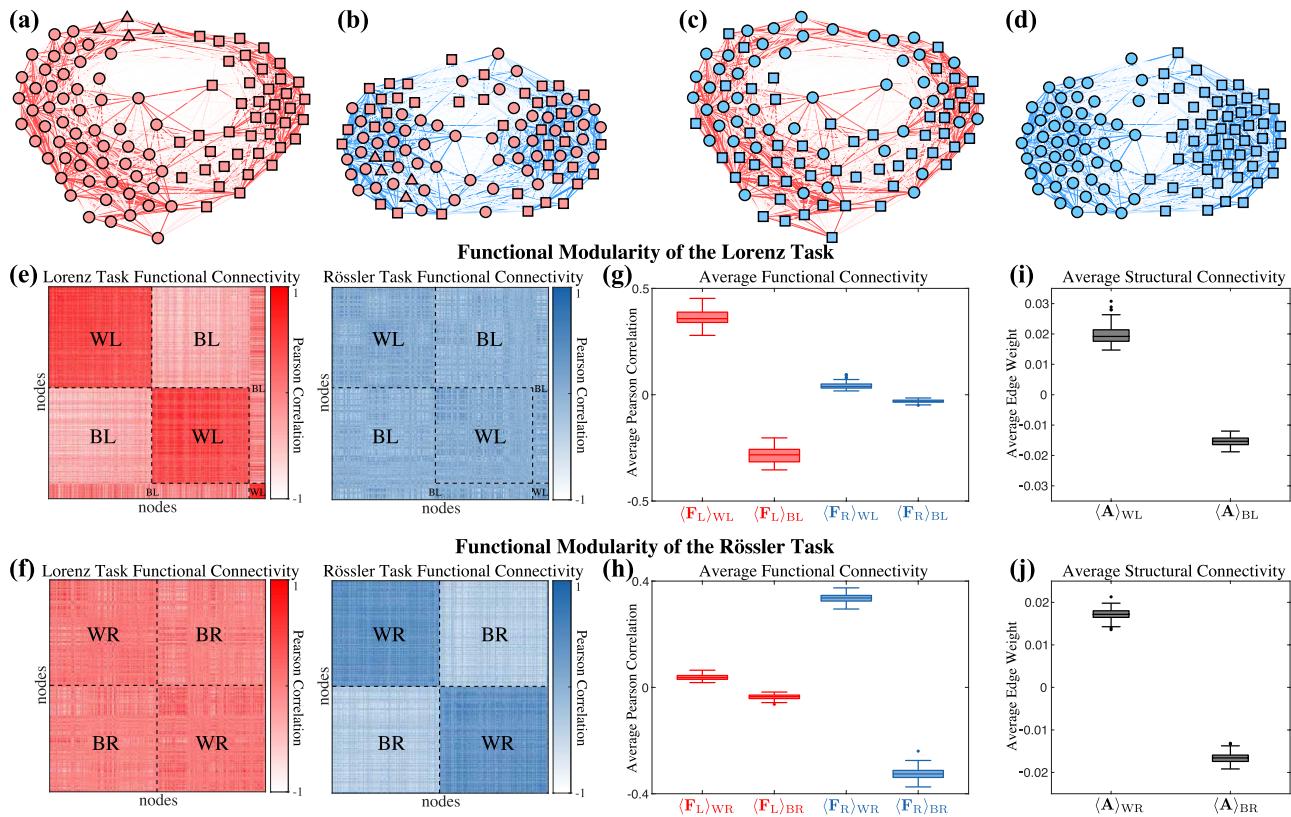


FIG. 5. The task-dependent functional connectome estimated from the statistical similarity in activity time series between neurons in the RNN. Panels (a)–(d) are visualizations of the two functional connectomes during the Lorenz-task [(a) and (c)] and the Rössler-task [(b) and (d)]. For clarity, we randomly select 100 neurons out of the full $N = 2000$, and we plot them using the force atlas layout in Gephi. These 100 neurons, in panels (a) and (b), are marked as either circles, squares, or triangles, corresponding to the three clusters detected from \mathbf{F}_L ; all markers are shown in red. In contrast, in panels (c) and (d), the 100 neurons are marked by circles and squares, corresponding to the 2 clusters detected from \mathbf{F}_R ; all markers are shown in blue. In panels (e) and (f), we show the full functional connectivity matrix using a data-driven community detection algorithm. The functional connectivity matrix is divided into regions that are within or between the detected clusters during the Lorenz- or the Rössler-task, and are denoted by WL, BL, WR, and BR, respectively. (g) and (h) The average functional connectivity within and between communities as estimated in an ensemble of 100 RNNs. (i) and (j) The average structural connectivity (synaptic weights) within and between communities as estimated in the same ensemble of 100 RNNs.

neurons that are within *vs* between communities identified from either \mathbf{F}_L or \mathbf{F}_R [labeled “WL,” “BL,” “WR,” and “BR,” respectively, in Figs. 5(e) and 5(f)]. In Figs. 5(g) and 5(h), we observe, as one may expect after application of the community detection algorithm, that the functional connectivity estimated from the Lorenz-task and averaged *within* the communities identified from the Lorenz-task data, $\langle \mathbf{F}_L \rangle_{WL}$, is significantly larger than the functional connectivity estimated from the Lorenz-task and averaged *between* the communities identified from the Lorenz-task data, $\langle \mathbf{F}_L \rangle_{BL}$. Similarly, for the Rössler-task, $\langle \mathbf{F}_R \rangle_{WL}$ is significantly larger than $\langle \mathbf{F}_R \rangle_{BR}$. In contrast, the fact that $\langle \mathbf{F}_L \rangle_{WR}$ is not significantly larger than $\langle \mathbf{F}_L \rangle_{BR}$, and similarly $\langle \mathbf{F}_R \rangle_{WL}$ is not significantly larger than $\langle \mathbf{F}_R \rangle_{BL}$, supports our qualitative observation that the community structure in the functional connectivity matrix of the Lorenz-task is distinct from the community structure in the functional connectivity matrix of the Rössler-task.

Lastly, we asked whether there existed any structural basis for the observed emergent functional communities. Importantly, random networks are far from homogeneous, and can display locally dense areas as well as locally sparse areas that occur simply by chance. It is therefore intuitively possible that the random network A contains degenerate weak community structure that supports the distinct community structure present in the functional connectivity when the system generates different patterns of emergent dynamics. To investigate this possibility, we calculated the average structural connectivity within the Lorenz communities, $\langle \mathbf{A} \rangle_{WL}$ within the Rössler communities, $\langle \mathbf{A} \rangle_{WR}$, between the Lorenz communities, $\langle \mathbf{A} \rangle_{BL}$, and between the Rössler communities, $\langle \mathbf{A} \rangle_{BR}$. We observed greater average structural connectivity within communities than between communities: that is, $\langle \mathbf{A} \rangle_{WL}$ and $\langle \mathbf{A} \rangle_{WR}$ were significantly larger than $\langle \mathbf{A} \rangle_{BL}$ and $\langle \mathbf{A} \rangle_{BR}$ [Figs. 5(g) and 5(h)]. This observation motivates two open questions: (i) whether some structural

networks more easily (or less easily) support diverse functional community structures, and thus the learning of multiple systems, and (ii) for a given structural network, can one predict the number of possible emergent community structures, and therefore the number of systems that can be learned. While both questions remain to be answered, in Appendix E we provide additional quantitative analysis to assess differences in the strength and topology of structural support for functional communities elicited by the two tasks.

V. INFER MISSING VARIABLES FROM LEARNED DYNAMICAL SYSTEMS

A recent work⁷⁰ has shown that a reservoir computer can be successfully trained to solve the missing-variable-inference problem for chaotic systems. Here, with our biological plausible learning framework, we consider whether the experience of learning the full dynamical system can, thereafter, immediately allow the neural network to fill in missing variables from a sparse measurement of the learned dynamical system. Given that dynamical attractors \mathcal{A}_k are successfully learned, we consider the case in which a new trajectory $s(t)$ on attractor \mathcal{A}_k is given but with some of the variables $[s]_i$ missing. The goal is to use the acquired knowledge of \mathcal{A}_k , together with the remaining variables $[s]_j$, to infer values of the missing variables $[s]_i$, where $i \neq j$. To perform this inference, we evolve the central system following the testing phase dynamics shown in Eq. (2), and we replace the missing variables $[s]_i$ in s by the corresponding output variables $[\tilde{s}]_i$ obtained from Eq. (4) [Figs. 6(a)–6(c)]. If the central system driven by available variables $[s]_j$ and its own output variables $[\tilde{s}]_i$ maintains generalized synchronization with the input dynamical system that generates s , then the inference is expected to be successful.

To instantiate this problem *in silico*, we train an RNN with $N = 2000$ neurons to learn both the Lorenz and the Rössler systems. Then, we test the inference of this RNN in three scenarios, where zero, one, and two variables $[s]_i$ from $s(t)$ are missing [Figs. 6(a)–6(c)]. The inferred trajectories are the central system outputs \tilde{s} for the Lorenz- and Rössler-tasks, respectively. In agreement with intuition, more missing variables leads to poorer inference quality, where the quality is quantified by the normalized mean squared error $\text{NMSE} = \text{mean}(\|\tilde{s}(t) - s(t)\|^2)/\text{var}(\|s(t)\|^2)$; note that the average occurs over a time window of 500 units and a time step 0.02 units. Since the inference of missing variables is by nature different from autonomous prediction of chaotic systems, the inferred values do not necessarily diverge from the truth even over a long period of time. This NMSE assessing the encoding error is also referred to as the “learning phase error” in Appendix A. To show that the inference quality is related to the quality of the generalized synchronization, we train an ensemble of 100 RNNs, and we calculate both their inference error as well as the largest conditional Lyapunov exponent LYP_{\max} . A more negative LYP_{\max} suggests a stronger generalized synchronization. In Fig. 6(d), we observe that, for the three scenarios, the inference error $\log_{10}(\text{NMSE})$ is indeed higher whenever the generalized synchronization is weaker; that is, when the largest conditional Lyapunov exponent of the central system LYP_{\max} is less negative.

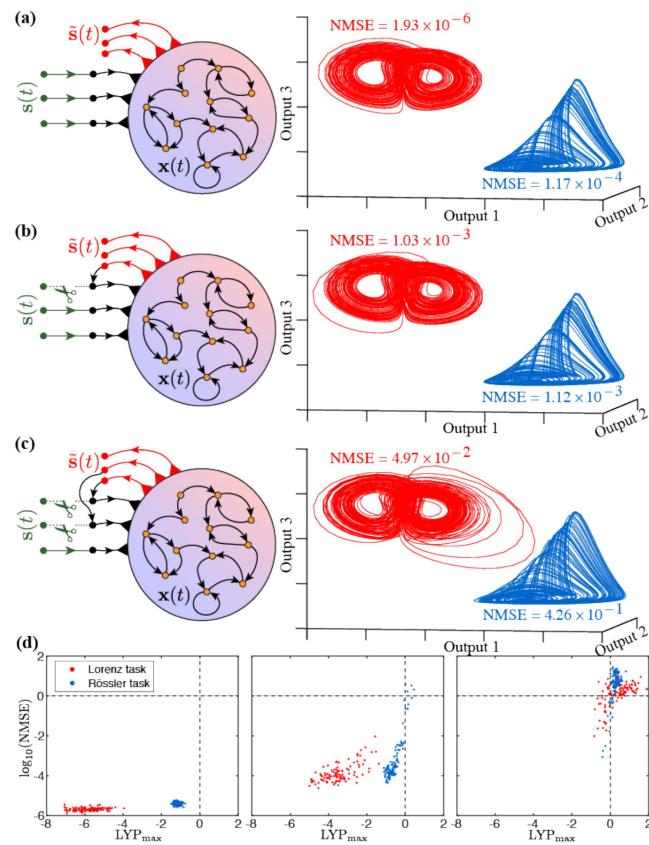


FIG. 6. Inferring the task for both the Lorenz and Rössler systems when full or partial input variables are provided. (a) The inferred output \tilde{s} from the central system as well as the normalized mean squared error for the Lorenz (red) and the Rössler (blue) tasks when three out of three input variables are provided. (b) The inferred output \tilde{s} from the central system as well as the normalized mean squared error for the Lorenz (red) and the Rössler (blue) tasks when two out of three input variables are provided. (c) The inferred output \tilde{s} from the central system as well as the normalized mean squared error for the Lorenz (red) and the Rössler (blue) tasks when one out of three input variables is provided. (d) The log of the normalized mean squared error in the inference task with respect to the largest conditional Lyapunov exponents, from 100 randomly constructed neural networks.

VI. SEPARATING SUPERIMPOSED INPUT FROM DIFFERENT SOURCES

Thus far, we have considered cases in which the external input comprises a trajectory generated by one chaotic system at a time. However, in real environments, complex adaptive systems—including the human brain—often process mixed sensory input given by a superposition of multiple input systems.⁷⁷ Recent studies have shown that such separation problems can be successfully solved by various systems, including a reservoir computer⁷⁷ and even a tank of water.⁷⁸ It is therefore natural to ask whether the encoding mechanism (invertible generalized synchronization) also allows our more biologically plausible learning system to execute such a function. To address this question, we employ a 2000-node

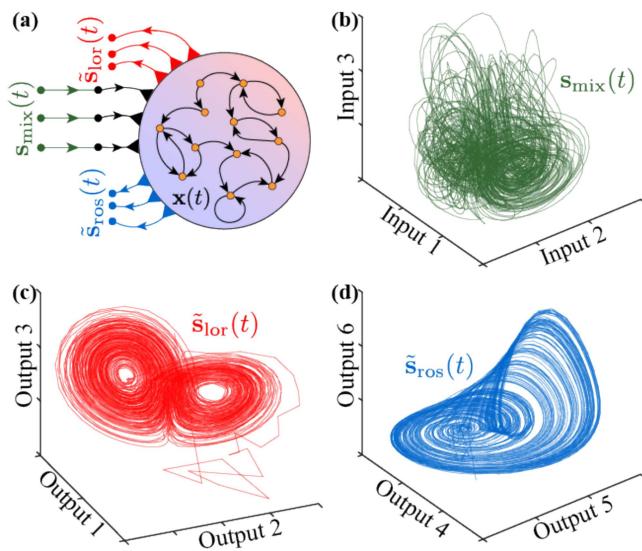


FIG. 7. Separating superimposed input from different dynamical system sources. (a) Schematic of a learning system being trained to separate superimposed input. The central system is driven by three-dimensional superimposed input $s_{\text{mix}}(t) = s_{\text{lor}}(t) + s_{\text{ros}}(t)$. During the learning phase, the output weight matrices for the 2 three-dimensional outputs \tilde{s}_{lor} and \tilde{s}_{ros} approximate the concurrent Lorenz trajectory s_{lor} and Rössler trajectory $s_{\text{ros}}(t)$. (b) The testing phase input trajectory to the central system, s_{mix} , which is a superposition of a new Lorenz trajectory and a new Rössler trajectory, both different from the exemplary trajectories used in the learning phase. (c) The \tilde{s}_{lor} deciphered by this model during the testing phase. After a transient excursion from the initial state, the deciphered output converges to a good approximation of the actual Lorenz trajectory. (d) The \tilde{s}_{ros} deciphered by this model during the testing phase. After a transient excursion from the initial state, the deciphered output converges to a good approximation of the actual Rössler trajectory.

RNN with Erdős–Rényi topology as the central system (Fig. 7), and we construct a three-dimensional mixed input signal $s_{\text{mix}}(t) = s_{\text{lor}}(t) + s_{\text{ros}}(t)$, where $s_{\text{lor}}(t)$ and $s_{\text{ros}}(t)$ are independent trajectories from the Lorenz attractor $\mathcal{A}_{\text{Lorenz}}$ and the Rössler attractor $\mathcal{A}_{\text{Rössler}}$, respectively. As the central system encodes the mixed input into its own state variable $x(t)$, we seek two output weight matrices, \mathbf{W}_{lor} and \mathbf{W}_{ros} , that can successfully reconstruct the segregated signals, i.e., $\mathbf{W}_{\text{lor}}x(t) \approx s_{\text{lor}}(t)$ and $\mathbf{W}_{\text{ros}}x(t) \approx s_{\text{ros}}(t)$. We find that such output matrices do exist, because the delta-rule adaptation of both \mathbf{W}_{lor} and \mathbf{W}_{ros} converges after many repetitions of the learning phase.

Next, we test the decoding power of the trained central system by constructing a new mixed input signal $s_{\text{mix}}(t)$ with a new set of initial conditions [Fig. 7(b)], which differs markedly from the mixed signal used in the learning phase. Starting from a random initial state $x(t)$, we drive the central system with this new mixed input $s_{\text{mix}}(t)$ and monitor the two three-dimensional outputs $\tilde{s}_{\text{lor}}(t) = \mathbf{W}_{\text{lor}}x(t)$ and $\tilde{s}_{\text{ros}}(t) = \mathbf{W}_{\text{ros}}x(t)$ [Figs. 7(c) and 7(d)]. We observe that the segregated signals are indeed reconstructed from the central system by the two output matrices. Taken together, these simulations demonstrate that segregated signals can be retrieved and

reconstructed from the state variable of the central system in our learning framework when the central system receives mixed signals.

In order to build a deeper intuition, it is helpful to consider the relevant underlying mechanism within the dynamical systems framework. Formally, we consider the input signal $s_{\text{mix}} = s_{\text{lor}} + s_{\text{ros}}$ a superposition of any Lorenz trajectory and any Rössler trajectory, and we formulate a 6-dimensional dynamical system that combines the Lorenz system and the Rössler system. The state variable of this combined system is the Cartesian product of the state variables of the Lorenz system and the Rössler system, which is

$$\mathbf{s}_{\text{com}}(t) = \begin{bmatrix} s_{\text{lor}}(t) \\ s_{\text{ros}}(t) \end{bmatrix} \in \mathbb{R}^6. \quad (11)$$

As $s_{\text{lor}}(t)$ and $s_{\text{ros}}(t)$ evolve on the Lorenz attractor $\mathcal{A}_{\text{Lorenz}}$ and the Rössler attractor $\mathcal{A}_{\text{Rössler}}$, respectively, the $\mathbf{s}_{\text{com}}(t)$ evolves onto an attractor that is the Cartesian product of the two attractors, $\mathcal{A}_{\text{com}} = \mathcal{A}_{\text{Lorenz}} \times \mathcal{A}_{\text{Rössler}} \subset \mathbb{R}^6$. Then, we consider the central system as a response system that is one-way coupled to the combined dynamical system through an incomplete measurement function $h(\cdot)$,

$$\mathbf{s}_{\text{mix}} = h(\mathbf{s}_{\text{com}}) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{\text{lor}} \\ s_{\text{ros}} \end{bmatrix}. \quad (12)$$

Although $h(\cdot)$ only provides a three-dimensional projection s_{mix} of the 6-dimensional full measurement of the combined system, \mathbf{s}_{com} , the invertible generalized synchronization between the central system and the combined system may still occur.²⁶ Once synchronized, the state of the central system $x(t) \rightarrow \psi(s_{\text{com}}(t))$, where $\psi(\cdot)$ is invertible on \mathcal{A}_{com} . The \mathcal{A}_{com} is then encoded into the central system in the form of an internal representation $\mathcal{P}_{\text{com}} = \psi(\mathcal{A}_{\text{com}})$. Thus, it becomes obvious that one can reconstruct both s_{lor} and s_{ros} by output weight matrices that approximate the $\psi^{-1}(\cdot)$. That is,

$$\begin{bmatrix} \mathbf{W}_{\text{lor}} \\ \mathbf{W}_{\text{ros}} \end{bmatrix} \mathbf{x} \approx \psi^{-1}(\mathbf{x}), \quad (13)$$

for $\mathbf{x} \in \mathcal{P}_{\text{com}}$. Thus we see that it is the nature of the information encoding that allows for the existence of output matrices that can reconstruct segregated signals from $\mathbf{x}(t)$. The mechanism of information encoding—invertible generalized synchronization—does not only encode the mixed input s_{mix} , but also the two segregated signals s_{lor} and s_{ros} into its state space, even though the segregated signals are not explicitly given to the central system.

VII. CONCLUSION

Complementing probabilistic models for learning hidden properties from experience,^{79–86} we propose a general dynamical systems model informed from biological features of the human brain. Surprisingly, this simple dynamical model produces successful learning of attractor dynamics from exemplary trajectories on these attractors. Our model also supports many other complex learning functions, such as task switching, filling-in missing variables, and separating mixed inputs. We believe that the model serves as a promising preliminary framework that can be extended to future studies to include additional biological details supporting better performance. But perhaps most importantly, the framework sheds light

on the possible dynamical origin of the emergence of many learning functions in biological systems.

ACKNOWLEDGMENTS

We thank Elisabeth A. Karuza, Christopher W. Lynn, Jason Z. Kim, Lia Papadopoulos, and Arian Ashourvan for helpful comments on earlier versions of this manuscript. Z.L. and D.S.B. acknowledge support from the John D. and Catherine T. MacArthur Foundation, the Alfred P. Sloan Foundation, the ISI Foundation, the Paul Allen Foundation, the Army Research Laboratory (No. W911NF-10-2-0022), the Army Research Office (Nos. Bassett-W911NF-14-1-0679, Grafton-W911NF-16-1-0474, DCIST-W911NF-17-2-0181), the Office of Naval Research, the National Institute of Mental Health (Nos. 2-R01-DC-009209-11, R01-MH112847, R01-MH107235, R21-M MH-106799), the National Institute of Child Health and Human Development (1R01HD086888-01), the National Institute of Neurological Disorders and Stroke (No. R01 NS099348), and the National Science Foundation (Nos. BCS-1441502, BCS-1430087, NSF PHY-1554488, and BCS-1631550). The content is solely the responsibility of the authors and does not necessarily represent the official views of any of the funding agencies.

APPENDIX A: LYAPUNOV EXPONENTS AND THE STABILITY OF ENCODING AND RETRIEVAL

1. Largest conditional Lyapunov exponent and generalized synchronization

One common criterion of generalized synchronization is the sign of the largest conditional Lyapunov exponent.⁵⁶ By conditional, we mean that—in this directionally coupled drive-response system [Eqs. (1) and (2)]—the stability of the response system [Eq. (2)] is measured under the condition that it is driven by an unperturbed input system [Eq. (1)]. Specifically, we drive the learning phase central system [Eq. (2)] with the same input trajectory $s(t)$ that is on \mathcal{A} , yet from two nearby random initial states: $x(0)$ and $x'(0) = x(0) + \delta_0$, where $\|\delta_0\|_2 \ll 1$. The largest conditional Lyapunov exponent λ_{\max} is then the exponential convergence or divergence rate of the distance $\|\delta_t\|_2 = \|x'(t) - x(t)\|_2$ between these two trajectories,

$$\lambda_{\max} = \lim_{T \rightarrow \infty} \frac{1}{T} \log \left(\frac{\|\delta_T\|_2}{\|\delta_0\|_2} \right). \quad (\text{A1})$$

Notice that δ_t evolves as

$$\delta_{t+1} = J(t)\delta_t, \quad (\text{A2})$$

where $J(t) \in \mathbb{R}^{N \times N}$ is the Jacobian matrix of the externally driven response system,

$$J(t) = \left. \frac{\partial f(x, s)}{\partial x} \right|_{x=\hat{x}(t), s=\hat{s}(t)}. \quad (\text{A3})$$

Here, $\hat{x}(t)$ is the trajectory of the central system [Eq. (2)], and $\hat{s}(t)$ the driving trajectory from the input system [Eq. (1)]. With Eqs. (A1)–(A3), we can numerically estimate the largest conditional Lyapunov exponent λ_{\max} . If $\lambda_{\max} < 0$, we say that the central system exhibits generalized synchronization to the input system, and the input information is encoded into the central system in the form of

$x = \psi(s)$. Conversely, when $\lambda_{\max} \geq 0$, the state of the central system cannot be uniquely determined by the input s .

To provide further intuition, we perform a series of numerical experiments. Using the learning schedule discussed in Appendix D, we train the output functions of 640 randomly constructed RNNs with Erdős–Rényi topology on both the Lorenz- and Rössler-tasks. Once the learning schedules are finished, regardless of the convergence of W_{out} , we calculate the learning phase error for each task; that is, we calculate the mean squared error between the actual sensory input $s(t)$ and the predicted sensory input $\tilde{s}(t)$. These encoding errors are plotted together with the largest conditional Lyapunov exponents in Fig. 8. The results are sorted by the largest conditional Lyapunov exponent in an increasing order. Consistent with prior theoretical work,⁵⁶ the learning phase errors are particularly large when the largest conditional Lyapunov exponents become positive.

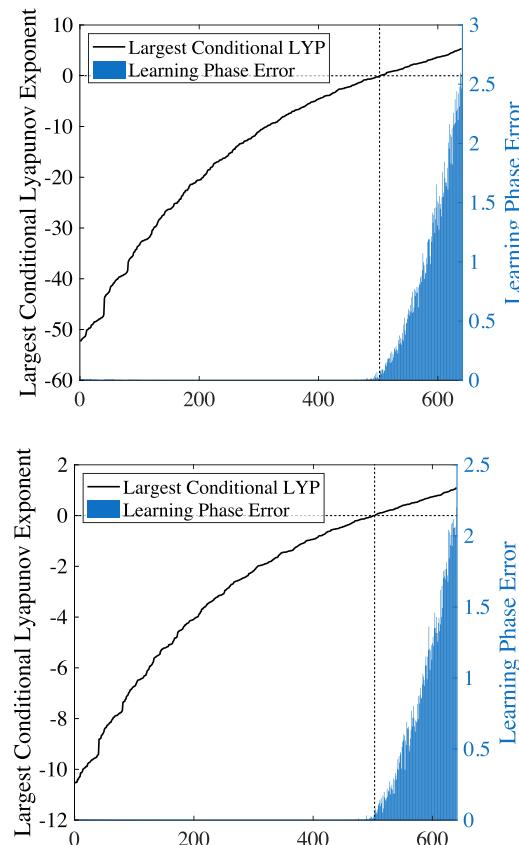


FIG. 8. Relation between the largest conditional Lyapunov exponent and the learning phase error. These two panels show (left y axis) the largest conditional Lyapunov exponent in the learning phase and (right y axis) the corresponding learning phase error, for the Lorenz-task (upper panel) and the Rössler-task (lower panel) for 640 trained RNNs with Erdős–Rényi topology. These 640 RNNs were constructed using ten random realizations for each of the 64 parameter pairs (ρ, ξ) spanning the two-dimensional space defined by the spectral radius $\rho = 0.5, 0.6, \dots, 2.0$ of the Erdős–Rényi graph and the separation parameter $\xi = 5, 10, 15, 20$ (see Appendix B 5 for definition) between the two attractors. Collectively, these results show that—for either task—the learning phase error is particularly great when the largest conditional Lyapunov exponent is positive.

2. Largest transversal Lyapunov exponent and the persistence of the imitation

To generate an output signal that reliably imitates the learned dynamics, the testing phase autonomous central system [Eq. (6)] should remain on \mathcal{P} . Although the generalized synchronization guarantees that the central system remains on \mathcal{P} during the learning phase, it is not necessarily the case when the central system becomes autonomous during the testing phase. Thus, it is worth analyzing the testing phase stability by calculating the Lyapunov exponents of this autonomous central system as it evolves on \mathcal{P} . Specifically, we calculate the Lyapunov exponents of the autonomous central system using the Jacobian of the testing phase central system,

$$J(t) = \frac{df(x, \phi(x))}{dx} \Big|_{x=\hat{x}(t)}, \quad (\text{A4})$$

evaluated along a trajectory $\hat{x}(t)$ from the learning phase central system [Eq. (2)], rather than the testing phase central system [Eq. (6)]. This choice is motivated by our goal to evaluate the stability of a trajectory that remains on \mathcal{P} even when the autonomous central system is transversely unstable and cannot be sustained on \mathcal{P} .

As the high-dimensional autonomous central system incorporates the local dynamics of the low-dimensional input system on \mathcal{A} , it inherits all the non-negative Lyapunov exponents from the input dynamical system.^{24,26} Thus, \mathcal{P} is an attractor of Eq. (6) if its $m+1$ -th Lyapunov exponent is negative, where m is the number of non-negative Lyapunov exponents from Eq. (1) on \mathcal{A} . For example, $m=2$ for both the Lorenz and the Rössler systems, as they both contain one positive and one zero Lyapunov exponent. For the 2-D torus attractor, $m=2$ because it contains two zero Lyapunov exponents and zero positive Lyapunov exponents. For the limit cycle attractor, $m=1$ because it has only one non-negative Lyapunov exponent that is zero. For a fixed point attractor, $m=0$ because it does not have any non-negative Lyapunov exponents. Thus, for either the Lorenz or the Rössler attractors, if the central system is transversally stable on the corresponding internal representation \mathcal{P} , the third largest Lyapunov exponent of the central system given by Eq. (6) should be negative.

3. Quantifying learning performance using the testing phase error

During the testing phase, the system autonomously generates new streams of signals using the implicitly acquired dynamical rules. Since the new streams of signals are chaotic and are not intended to be identical to any of the exemplary signals used in the learning phase, it is impossible to quantify the error in the same manner as we did in the learning phase. Thus, we propose a different quantitative performance measure that we call the “testing phase error,” which describes the difference between the implicit dynamical rules acquired and the dynamical rules underlying the system that generated the exemplary sensory inputs.

Along an extended trajectory $\tilde{s}(t)$ generated by the testing phase central system, we denote the system’s movement during one unit of time as $\delta\tilde{s}(t) = \tilde{s}(t+1) - \tilde{s}(t)$. Based on the actual dynamics of the input system [Eq. (1)], we also calculate the ideal movement vector $\delta s(t) = g(\tilde{s}(t)) - \tilde{s}(t)$ from each point $\tilde{s}(t)$. The testing phase error

is then defined as

$$\text{testing phase error} = \left(\frac{\|\delta\tilde{s}(t) - \delta s(t)\|_2}{\|\delta s(t)\|_2} \right)_t, \quad (\text{A5})$$

where $(\cdot)_t$ denote the time average along the trajectory. When the autonomously generated trajectory $\tilde{s}(t)$ evolves on \mathcal{A} with the correct underlying dynamics, the testing phase error is small (~ 0.1). When the generated trajectory escapes from \mathcal{A} , the testing phase error can be very large.

In the results shown in Fig. 9, the testing phase error is small when the central system is transversely stable on the internal representation \mathcal{P} . However, due to the bubbling phenomenon,⁸⁷ we also note that—while the largest transversal Lyapunov exponent

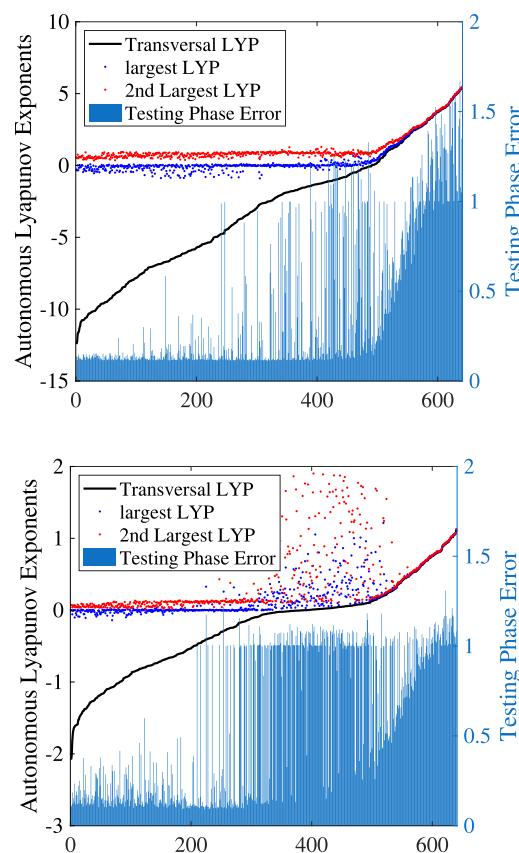


FIG. 9. Relation between the largest Lyapunov exponents of the testing phase autonomous system and the testing phase error. These two panels show (left y axis) the three largest Lyapunov exponents of the testing phase autonomous system as well as (right y axis) the testing phase error, on the Lorenz-task (upper panel) and the Rössler-task (lower panel) for 640 trained RNNs with Erdős-Rényi topology. These 640 RNNs are sorted from least to greatest third largest Lyapunov exponent. When the third largest Lyapunov exponent is non-negative, the transversal Lyapunov exponent is greater than zero, and thus the internal representations are not transversely stable, resulting in notably worse performance. For those instances that have relatively low error, the first two Lyapunov exponents are close to the positive and zero Lyapunov exponents from the Lorenz and Rössler systems.

describes the overall stability on \mathcal{P} —it is possible for the generated trajectory to escape from \mathcal{A} even when the transversal Lyapunov exponent is negative. From the results reported in Fig. 9, we observe that there are many instances where the testing phase error is close to 1. These instances may correspond to the cases where the generated trajectory escapes (via bubbling) from \mathcal{A} and lands on a fixed point, resulting in $\delta\tilde{s}(t) = 0$, and thereby leading to a testing phase error of approximately 1.

APPENDIX B: PREPARING INPUT TRAJECTORIES FROM CHAOTIC AND NON-CHAOTIC SYSTEMS

In the main text, we demonstrate the capacity to learn the dynamics of multiple attractors with a proof-of-principle study considering four attractors: one periodic, one quasi-periodic, and two chaotic. We also demonstrate the ability to successfully induce task switching among these four attractors. Outside of this specific study, the remainder of this paper predominantly focuses on the learning of the two widely studied three-dimensional chaotic dynamical systems: the Lorenz system and the Rössler system. In the following subsections, we describe the four dynamical systems in detail, and we also describe the construction of exemplary trajectories from each.

1. Input trajectory from the Lorenz system

The input trajectory $s_{\text{lor}}(t)$ from the Lorenz system is generated as follows. We integrate the differential equations of the Lorenz system,

$$\begin{aligned} \frac{d}{dt}X_{\text{lor}} &= 10Y_{\text{lor}} - 10X_{\text{lor}}, \\ \frac{d}{dt}Y_{\text{lor}} &= -X_{\text{lor}}Z_{\text{lor}} + 28X_{\text{lor}} - Y_{\text{lor}}, \\ \frac{d}{dt}Z_{\text{lor}} &= X_{\text{lor}}Y_{\text{lor}} - 8Z_{\text{lor}}/3, \end{aligned} \quad (\text{B1})$$

using a 4-th order Runge–Kutta integrator with time step $\delta t = 10^{-3}$ from a random initial state. Each of the three variables X_{lor} , Y_{lor} , and Z_{lor} of the trajectory is then normalized to have a mean of zero and a variance of one. This normalized trajectory is then saved as the Lorenz-task input trajectory $s_{\text{lor}}(t)$ with time resolution $\tau = 0.02$.

2. Input trajectory from the Rössler system

Similar to the Lorenz system, the input trajectory $s_{\text{ros}}(t)$ from the Rössler system is generated as follows. We integrate the differential equations of the Rössler system,

$$\begin{aligned} \frac{d}{dt}X_{\text{ros}} &= -5Y_{\text{ros}} - 5Z_{\text{ros}}, \\ \frac{d}{dt}Y_{\text{ros}} &= 5X_{\text{ros}} + 5Y_{\text{ros}}/2, \\ \frac{d}{dt}Z_{\text{ros}} &= 10 + 5Z_{\text{ros}}(X_{\text{ros}} - 4), \end{aligned} \quad (\text{B2})$$

using a 4-th order Runge–Kutta integrator with time step $\delta t = 10^{-3}$ from a random initial state. The coefficients of the Rössler system

in Eq. (B2) are chosen such that the system has a similar time scale to that of the Lorenz system. We then normalize each of the three variables, X_{ros} , Y_{ros} , and Z_{ros} , such that they all have a mean of zero and a variance of one. We then save the normalized trajectory as the Rössler-task input trajectory $s_{\text{ros}}(t)$ with time resolution $\tau = 0.02$.

3. Input trajectory from a system with a stable limit cycle

In addition to the two chaotic systems, we consider a dynamical system with less complex dynamics, where the attractor is a limit cycle. In this case, orbits in the phase space of such dynamical systems spiral into a closed periodic orbit. We engineer a three-dimensional dynamical system with such limit cycle attractor dynamics. We integrate the differential equations of the following system

$$\begin{aligned} \frac{d}{dt}X_{\text{lmc}} &= 10X_{\text{lmc}} * (2 - X_{\text{lmc}}^2 - Y_{\text{lmc}}^2) - 10Y_{\text{lmc}}, \\ \frac{d}{dt}Y_{\text{lmc}} &= 10Y_{\text{lmc}} * (2 - X_{\text{lmc}}^2 - Y_{\text{lmc}}^2) + 10X_{\text{lmc}}, \\ \frac{d}{dt}Z_{\text{lmc}} &= -10Z_{\text{lmc}}. \end{aligned} \quad (\text{B3})$$

using a 4-th order Runge–Kutta integrator with time step $\delta t = 10^{-3}$ from a random initial state. We thereby obtain an orbit that approaches a periodic orbit. We then normalize each of the three variables, X_{lmc} , Y_{lmc} , and Z_{lmc} , such that they all have a mean of zero and a variance of one, except for Z_{lmc} . We then save the normalized trajectory as the limit cycle-task input trajectory $s_{\text{lmc}}(t)$ with time resolution $\tau = 0.02$.

4. Input trajectory from a system with a quasi-periodic torus attractor

Another non-trivial attractor, which is neither chaotic nor periodic, is the quasi-periodic torus attractor. The orbit of such a system flows along the surface of a 2-dimensional torus. The motion is quasi-periodic because it is composed of oscillations of two incommensurate frequencies. The orbit on the torus attractor will densely cover the torus without collapsing into a periodic orbit. Following prior work,⁸⁸ we engineer a three-dimensional dynamical system with such quasi-periodic torus attractor dynamics. We integrate the differential equations of the following system:

$$\begin{aligned} \frac{d}{dt}X_{\text{trs}} &= 2\pi Y_{\text{trs}}, \\ \frac{d}{dt}Y_{\text{trs}} &= 2\pi Y_{\text{trs}}(X_{\text{trs}}^2 - 0.5X_{\text{trs}}^4) - 4\pi^2 X_{\text{trs}}, \\ \frac{d}{dt}Z_{\text{trs}} &= 0.9 - X_{\text{trs}}^2, \end{aligned} \quad (\text{B4})$$

using a 4-th order Runge–Kutta integrator with time step $\delta t = 10^{-3}$ from a random initial state. We thereby obtain an orbit on the quasi-periodic torus attractor. We then normalize each of the three variables, X_{trs} , Y_{trs} , and Z_{trs} , such that they all have a mean of zero and a variance of one. We then save the normalized trajectory as the Torus-task input trajectory $s_{\text{trs}}(t)$ with time resolution $\tau = 0.02$.

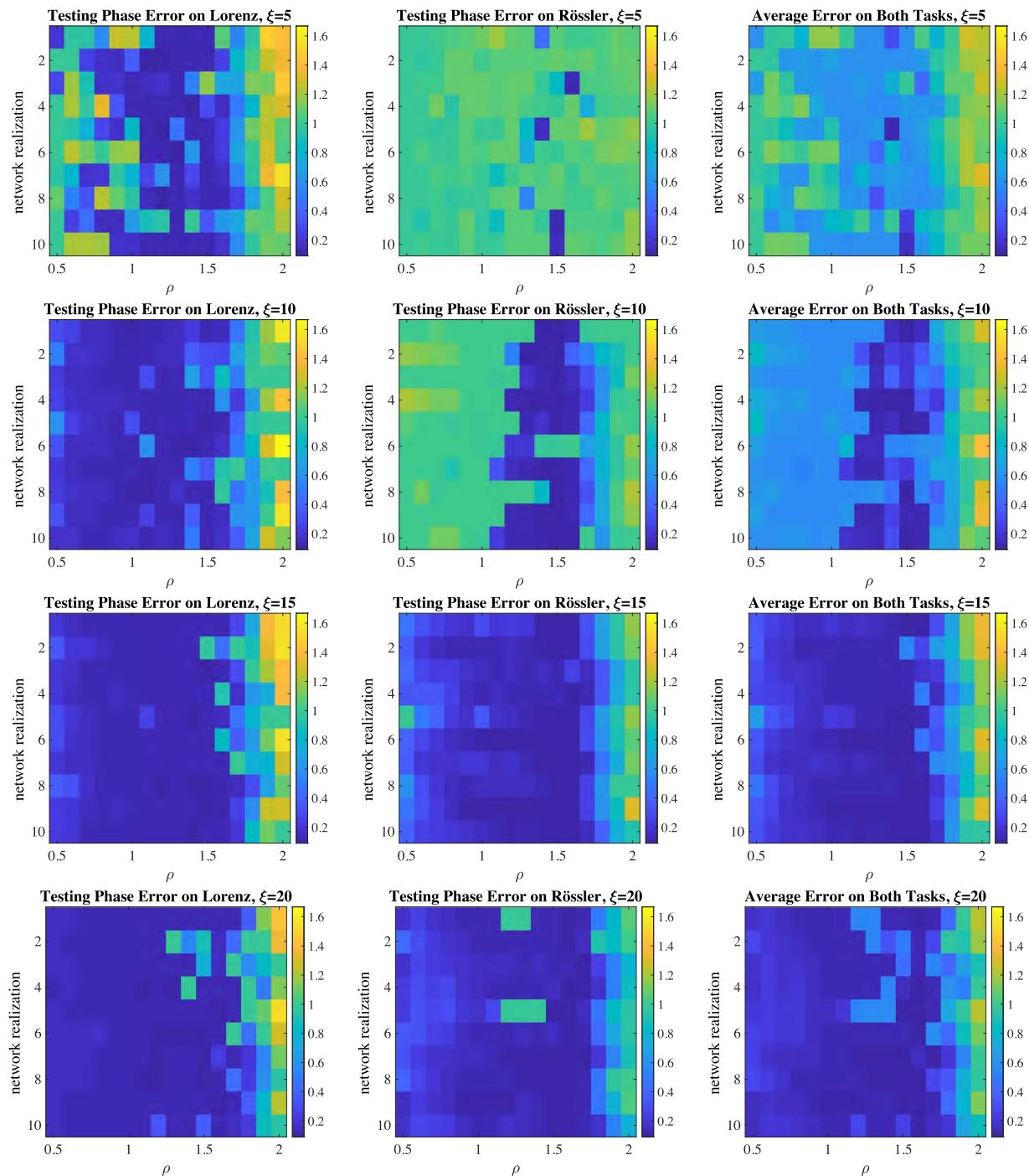


FIG. 10. Relation between the testing phase error and the separation between attractors, as well as the spectral radius of the structural graph on which the RNN is instantiated. In these heatmaps, color indicates the testing phase error on the Lorenz-task (left) and on the Rössler -task (center). We also show the mean testing phase error averaged across the two tasks (right). Each row of panels corresponds to a different value of the separation parameter $\xi = 5, 10, 15, 20$ (see Appendix B 5 for definition), with the top row of panels corresponding to $\xi = 5$ and the bottom row of panels corresponding to $\xi = 20$. All panels reflect results obtained from RNNs with Erds–Rényi topology, where the spectral radius ρ takes on values ranging from 0.5, 0.6, ..., 2.0 (x axis). For each (ρ, ξ) pair, we construct ten realizations of the Erds–Rényi topology (y axis). From these data, we can conclude that when ξ is larger, more RNNs learn both the Lorenz- and Rössler -tasks well.

5. Avoiding overlap between internal representations in multitask learning

In the multitask learning scenario, the central system [Eq. (2)] learns attractor dynamics by forming an internal representation during the learning phase. During the testing phase, a successful central system will evolve autonomously on the internal representations of the Lorenz task $\mathcal{P}_{\text{Lorenz}}$ - and of the Rössler -task $\mathcal{P}_{\text{Rössler}}$. However, if there exists any overlap between the two internal representations, at least one of the tasks will not be accurately learned. The reason for this inaccuracy is that the output mapping function $\phi : \mathcal{P} \rightarrow \mathcal{A}$, where $\mathcal{P} = \mathcal{P}_{\text{Lorenz}} \cup \mathcal{P}_{\text{Rössler}}$ and $\mathcal{A} = \mathcal{A}_{\text{Lorenz}} \cup \mathcal{A}_{\text{Rössler}}$, loses its one-to-one nature on the intersected region $\mathbf{x} \in \mathcal{P}_{\text{Lorenz}} \cap \mathcal{P}_{\text{Rössler}}$. As a consequence, when \mathbf{W}_{out} is sequentially updated following Eq. (9) on multiple attractors whose internal representations intersect each other, \mathbf{W}_{out} does not converge.

To explicitly prevent such overlap, we modify the input trajectories by increasing the mean of the Lorenz-task input $\mathbf{s}_{\text{lor}}(t)$ to $[\xi, \xi, \xi]$, and by decreasing the mean of the Rössler-task input $\mathbf{s}_{\text{ros}}(t)$ to $[-\xi, -\xi, -\xi]$, where $\xi \geq 0$ is the separation parameter. For the sake of visualization, we adopt $\xi = 10$ for the simulated examples shown in the main text. Results shown in Fig. 10 demonstrate that a larger value of ξ results in lower testing phase errors on both tasks.

APPENDIX C: INSTANTIATING THE LEARNING FRAMEWORK WITH DIFFERENT CENTRAL SYSTEMS

Our goal in building this general theoretical framework was to provide a simplistic and general explanation of learning dynamics observed in many neurobiological systems. To support its generalizability, we seek to demonstrate that this dynamical learning framework specified by Eqs. (1), (2), and (6) is robust to many different choices of the central system. To this end, we perform a range of numerical simulations to test several different central systems. In this section, we report on the nature of those experiments and also on the performance of these diverse central systems in learning both the Lorenz- and Rössler -tasks. Specifically, in Appendix C 1–3, we let the central system be an RNN with several different topologies, while in Appendix C 4, we let the central system be a random polynomial system with polynomial degree equal to 5, wrapped in the sigmoidal function $\tanh(\cdot)$ to prevent divergence.

1. RNNs with Erdős-Rényi topology

In the numerical simulations whose results we have reported thus far, we model the central system as an RNN of $N = 2000$ neurons with Erdős-Rényi topology. The $\tanh(\cdot)$ in Eq. (7) operates on a vector and returns a vector with the same shape that satisfies $[\tanh(\mathbf{x})]_i = \tanh([\mathbf{x}]_i)$. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{2000 \times 2000}$ is the weighted adjacency matrix of the RNN. The input weight matrix $\mathbf{W}_{\text{in}} \in \mathbb{R}^{2000 \times 3}$ propagates a three-dimensional input \mathbf{s} (either externally or internally generated) to the 2000 neurons. The vector $\mathbf{c} \in \mathbb{R}^{2000 \times 1}$ is a random vector whose elements are drawn uniformly from $[-1, 1]$. We choose a simple topology of the recurrent neural network where the adjacency matrix \mathbf{A} is a sparse Erdős-Rényi random matrix. The sparseness of the Erdős-Rényi -type random matrix \mathbf{A} (the fraction of non-zero elements) is set to be 0.02. All non-zero elements in \mathbf{A} are drawn uniformly from $[-\sigma, \sigma]$. The

value of σ is determined by having the spectral radius of \mathbf{A} (the magnitude of the largest eigenvalue of \mathbf{A}) be equal to $\rho = 1.4$. We construct the input weight matrix \mathbf{W}_{in} in such a way as to ensure that each neuron receives one and only one input variable from the three-dimensional input. The input connection strength (each nonzero element in \mathbf{W}_{in}) is drawn uniformly at random from the interval $[-0.05, 0.05]$.

2. RNNs with a balanced Watts-Strogatz topology

In addition to modeling the central system as an RNN with Erdős-Rényi topology, we sought to determine whether similar learning success could be attained by a central system modeled as an RNN with non-random topology. Specifically, we considered a commonly studied small-world topology as instantiated in so-called Watts-Strogatz networks³⁹ that are balanced with inhibitory synapses. The setup of the central system is identical to that of the Erdős-Rényi examples [Eq. (7)] as previously discussed, except for a different adjacency matrix \mathbf{A} . We construct the Watts-Strogatz adjacency matrix following the protocol described in Ref. 90, with network size $N = 2000$ and degree $\kappa = 40$. When the rewiring probability equals 0, the network is a regular ring lattice. Before the random rewiring process, the adjacency matrix is symmetric and satisfies

$$\mathbf{A}_{ij>i} = \begin{cases} \frac{1}{d_{ij}} & \text{if } d_{ij} \leq \kappa/2, \\ 0 & \text{otherwise,} \end{cases} \quad (\text{C1})$$

where $d_{ij} = j - i \bmod N$. Then, with the given rewiring probability, each existing unidirectional connection between node i and node j is replaced with another unidirectional connection between node i and a randomly chosen node $j' \neq i$, to which node i was not already connected. To balance the network with inhibitory synapses, we independently assign a sign to each element in the adjacency matrix \mathbf{A} uniformly at random and with a chance of one half. Then, we achieve a desired spectral radius ρ (the largest absolute value of its eigenvalues) by multiplying the matrix \mathbf{A} by a scalar. As shown in Fig. 11, this system can consistently learn both the Lorenz- and Rössler -tasks, and its performance rises as the rewiring probability increases and the spectral radius decreases. Thus, the greatest performance is reached for the most randomly wired balanced Watts-Strogatz networks. However, we note that the networks obtained when the rewiring probability is equal to 1 do not display the same topology as Erdős-Rényi networks, because synapse strengths are not uniformly distributed over the range $[-\sigma, \sigma]$.

3. RNNs with the pseudo-Watts-Strogatz random networks

Continuing in our study of the robustness of our learning framework to the topology of the network, we further consider a slightly modified Watt-Strogatz model. Specifically, we modify the protocol used in the previous example by first constructing a

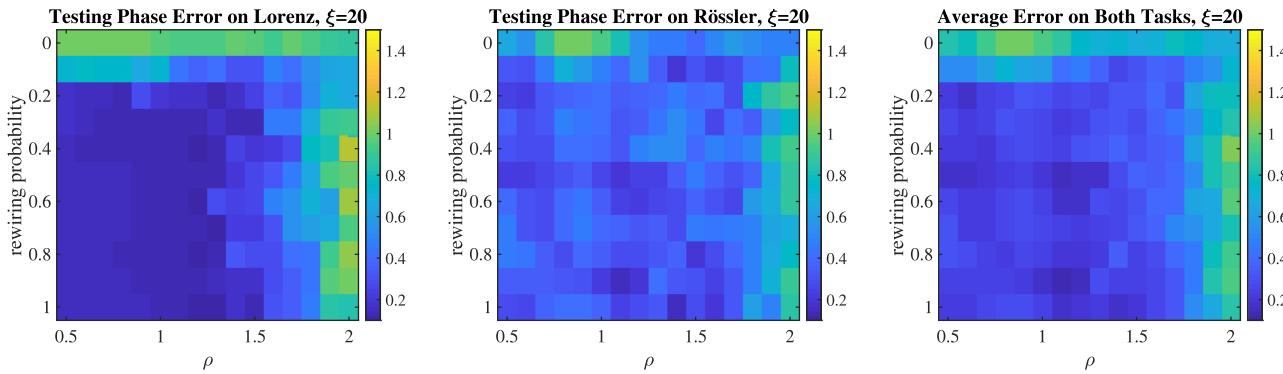


FIG. 11. Relation between the testing phase error and the spectral radius of a balanced Watts–Strogatz graph on whose structure the RNN is instantiated. In these heatmaps, color indicates the testing phase error on the Lorenz-task (left) and on the Rössler -task (center). We also show the mean testing phase error averaged across the two tasks (right). This set of panels corresponds to a separation parameter of $\xi = 20$. All panels reflect results obtained from RNNs with a balanced and randomly rewired Watts–Strogatz topology, where the spectral radius ρ takes on values ranging from 0.5, 0.6, ..., 2.0 (x axis). For each (ρ, ξ) pair, we construct ten realizations of the balanced Watts–Strogatz topology for a rewiring probability between 0 and 1 (y axis), and we show the mean testing phase error across these ten realizations.

ring lattice,

$$A_{ij>i} = \begin{cases} \frac{2d_{ij}}{k} & \text{if } d_{ij} \leq \kappa/2, \\ 0 & \text{otherwise,} \end{cases} \quad (\text{C2})$$

that is antisymmetric (i.e., $A = -A^T$). Then, given the rewiring probability, each directional connection from node i to node j is replaced by a connection to a randomly chosen node $j' \neq i$, to which node i was not already connected. Then, we multiply the matrix A by a scalar to obtain the desired spectral radius ρ . With this pseudo-Watt–Strogatz model, the connection strengths are approximately uniformly distributed over $[-\sigma, \sigma]$. Thus, using this model we can test how the learning performance varies as the network changes from a ring lattice structure to an Erdős–Rényi network. When the rewiring probability is 0, each node is connected to only $\kappa = 40$

neighboring nodes on the ring lattice, with $\kappa/2 = 20$ on each side. When the rewiring probability is 1, all connections are randomly rewired, and thus the topology approximates that of an Erdős–Rényi network. The performance of this pseudo-Watt–Strogatz network is shown in Fig. 12. Again, we find that this system can consistently learn both the Lorenz- and Rössler -tasks, and its performance rises as the rewiring probability increases and the spectral radius decreases.

4. Random polynomial network model

To further demonstrate the generalizability of this learning framework [Eqs. (1), (2), and (6)], we consider another high-dimensional dynamical system that differs markedly from an RNN. Specifically, we construct the following high-dimensional random polynomial-type central system with polynomial degree 5. In this

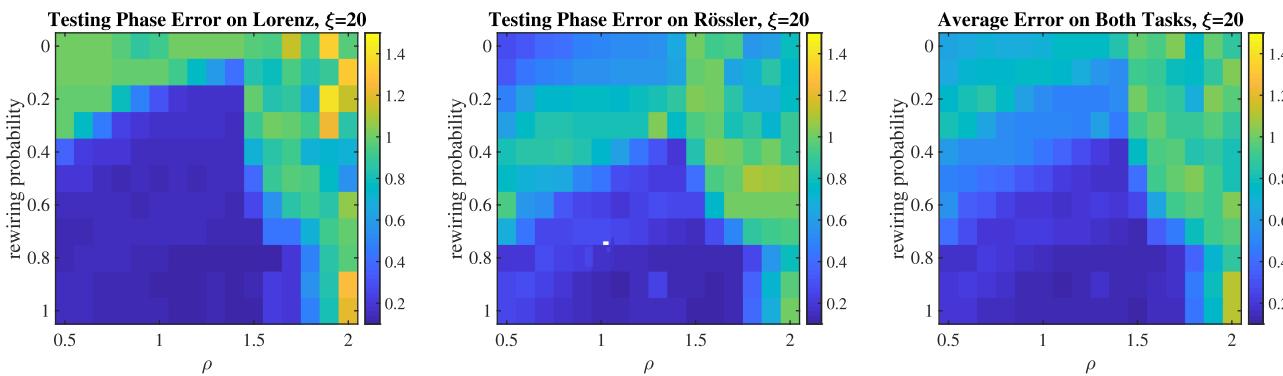


FIG. 12. Relation between the testing phase error and the spectral radius of a pseudo-Watts–Strogatz graph on whose structure the RNN is instantiated. In these heatmaps, color indicates the testing phase error on the Lorenz-task (left) and on the Rössler -task (center). We also show the mean testing phase error averaged across the two tasks (right). This set of panels corresponds to a separation parameter of $\xi = 20$. All panels reflect results obtained from RNNs with a pseudo- and randomly rewired Watts–Strogatz topology, where the spectral radius ρ takes on values ranging from 0.5, 0.6, ..., 2.0 (x axis). For each (ρ, ξ) pair, we construct ten realizations of the balanced Watts–Strogatz topology for a rewiring probability between 0 and 1 (y axis), and we show the mean testing phase error across these ten realizations.

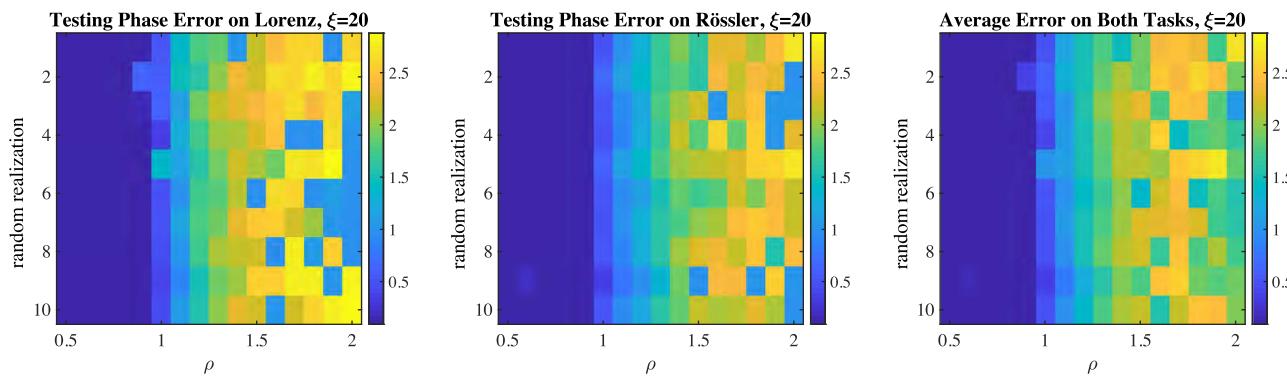


FIG. 13. Relation between the testing phase error and the spectral radius of the structural graph on which the polynomial system is instantiated. In these heatmaps, color indicates the testing phase error on the Lorenz-task (left) and on the Rössler -task (center). We also show the mean testing phase error averaged across the two tasks (right). This set of panels corresponds to a separation parameter of $\xi = 20$. Panels reflect results obtained from 160 randomly constructed polynomial systems, where the sparse random coefficient matrices $\mathbf{A}_{1,2,\dots,5}$ have been scaled to obtain a spectral radius of $\rho = 0.5, 0.6, \dots, 2.0$ (x axis). For each of the 16 values of ρ , we construct ten realizations of the topology (y axis).

model, the state of the central system $\mathbf{x} \in \mathbb{R}^{N=2000}$, driven by a three-dimensional input signal $\mathbf{s} \in \mathbb{R}^{n=3}$ (either externally or internally generated), evolves following

$$x_i(t+1) = \tanh \left(\sum_{j=1}^{2000} \sum_{k=1}^5 [\mathbf{A}_k]_{ij} \cdot x_j^k(t) + \sum_{j=1}^3 [\mathbf{W}_{\text{in}}]_{ij} \cdot s_j(t) + [\mathbf{c}]_i \right), \quad (\text{C3})$$

where the input weight matrix \mathbf{W}_{in} and the bias term \mathbf{c} are defined similarly to those in the previous models. To prevent divergence of the central system, we wrap the random polynomial with the sigmoidal function $\tanh(\cdot)$. The coefficients of these polynomials are denoted by five random matrices $\mathbf{A}_k \in \mathbb{R}^{N \times N}$, each with a fraction of non-zero elements set to 0.02. The non-zero elements are randomly and independently drawn from the uniform distribution $[-\sigma, \sigma]$, where $\sigma > 0$ is chosen such that the spectral radius of each \mathbf{A}_k equals ρ . In Fig. 13, we show the performance of the polynomial-type central system. We observe that these systems can robustly learn both tasks with very small testing phase error when the spectral radius ρ is less than one.

APPENDIX D: LEARNING BY ADAPTING THE OUTPUT WEIGHT MATRICES

Throughout our numerical simulations, all exemplary Lorenz and Rössler input trajectories have length 1000 time units, spanned by 5×10^4 data points obtained with a sampling rate of $\tau = 0.02$. During each learning phase, the output matrix \mathbf{W}_{out} adapts following Eq. (9) where $\alpha > 0$ is the learning rate.

Notice that by starting the central system from an arbitrary initial state $\mathbf{x}(0)$, the generalized synchronization occurs after a short transient period. As a result, we consider that the input signal is not being encoded into the state of the central system during this transient period. We therefore freeze the $\mathbf{W}_{\text{out}}(t)$ for the first 5×10^3 time points (chosen to be longer than the transient period) and only

allow $\mathbf{W}_{\text{out}}(t)$ to adapt for the remaining 4.5×10^4 time points during the learning phase. We repeat this process 1000 times to allow $\mathbf{W}_{\text{out}}(t)$ to converge. The learning rate is empirically set at $\alpha = 0.001$ during the first 300 rounds of learning. Then, for the purpose of fine tuning, α is set to 2×10^{-4} for the following 150 rounds of learning. Then, for the remaining 550 rounds of learning, α is set to 4×10^{-5} .

For cases where two or more attractors are learned, we let the system repetitively learn all tasks in an alternate manner for 1000 rounds. We notice that the trajectory of the central system $\mathbf{x}(t)$ is not affected by the adaptation of the output matrix \mathbf{W}_{out} , because it is driven by the external signal. Thus, in our simulations, it is not necessary for us to repetitively simulate the central system during repeated learning phases. Rather, for each task, we record, after the transient period, one long epoch of $\mathbf{x}(t)$, when the central system is driven by the particular external signal. Then, to learn multiple tasks, the \mathbf{W}_{out} is adapted in an alternating fashion based on the epochs of $\mathbf{x}(t)$ and the corresponding exemplary signals $\mathbf{s}(t)$. In practice, we alternate the adaptation of \mathbf{W}_{out} on each task using each of the long epochs $\mathbf{x}(t)$ for 200 rounds. Then, we cut the long epochs of $\mathbf{x}(t)$ for all tasks into many shorter epochs and interweave them into a single epoch of $\mathbf{x}(t)$. For the remaining 800 rounds of learning, the \mathbf{W}_{out} is adapted as it goes through this interwoven epoch 800 times. This single epoch interwoven from $\mathbf{x}(t)$ during different tasks provides a more frequent switching between the learning of different tasks. Although this interweaving procedure provides more frequent switching between the learning of different tasks, it does not need to discard a transient $\mathbf{x}(t)$ after switchings, since the transient \mathbf{x} has been discarded at the very beginning of the recording.

At this point in the exposition, we wish to emphasize that the above learning scheme is a somewhat biologically plausible scheme that we chose from a set of many other possible schemes. The convergence of this particular adaptive learning scheme depends on many heuristic factors, such as the value of the learning rate α . However, it is of particular interest to address the general convergence for different adaptive learning schemes. Based on our previous discussion of the information encoding mechanism, we are guaranteed

that the internal representations \mathcal{P} s are solely determined by the generalized synchronization function ($\mathcal{P} = \Psi(\mathcal{A})$), and are thus not affected by updating \mathbf{W}_{out} . Therefore, regardless of the particular learning scheme that one adopts, the learning of \mathbf{W}_{out} only converges if there exists an output weight matrix such that $\mathbf{s} = \mathbf{W}_{\text{out}}\mathbf{x}$ where $\mathbf{x} \in \mathcal{P}$ and $\mathbf{s} \in \mathcal{A}$. Consequently, from our theory we obtain two necessary but not sufficient conditions for learning convergence. One necessary condition is an invertible generalized synchronization between the drive system and the central system, which occurs when the largest conditional Lyapunov exponent is negative. The other necessary condition specific to learning multiple attractors is that the internal representations of different attractors cannot overlap.

APPENDIX E: DIFFERENCE IN STRUCTURAL CONNECTIVITY MATRIX WITHIN DIFFERENT FUNCTIONAL CONNECTIVITY CLUSTERS

In this section, we examine the intra-community structural connectivity for functional communities elicited by either the Rössler or Lorentz tasks. First, we visualize the structural connectivity matrices, ordered either by the functional communities elicited by the Lorentz task (left panel in Fig. 14) or by the functional communities elicited by the Rössler task (right panel in Fig. 14). Because of the marked sparsity of the structural matrix, it is difficult to make any explicit inferences from this visualization.

We therefore turn to statistical analyses. Specifically, we begin by evaluating the strength of structural connectivity within the functional communities elicited by the Rössler task, and we compare it to the strength of structural connections within the functional communities elicited by the Lorentz task [Figs. 15(a)–15(b)]. We consider the positive and negative structural connections separately. We find that the strength of structural connectivity is consistently greater within communities elicited by the Rössler task than within communities elicited by the Lorentz task (two-sample t -tests for positive

connections in community one $t = -11.57$, $p = 5.53 \times 10^{-24}$ and in community two $t = -10.05$, $p = 1.88 \times 10^{-19}$, and for negative connections in community one $t = -12.02$, $p = 2.37 \times 10^{-25}$ and in community two $t = -10.58$, $p = 5.22 \times 10^{-21}$). We also note visually that the variance of structural connection strength in the functional communities elicited by the Rössler task is smaller than the variance of structural connection strength in the functional communities elicited by the Lorentz task.

To further assess potential structural differences in the functional communities elicited by each task, we calculate two different graph statistics: the clustering coefficient and the global efficiency [Figs. 15(c) and 15(f)]. We begin with a signed clustering coefficient metric, and find no significant differences between (i) the clustering coefficient of the structural subnetwork composed of nodes assigned to the first functional community elicited by the Rössler task and (ii) the clustering coefficient of the structural subnetwork composed of nodes assigned to the first functional community elicited by the Lorentz task (two-sample t -test $t = -0.39$, $p = 0.70$). We observe similar non-significant results for the second functional community ($t = 0.04$, $p = 0.96$). Finally, we consider the global efficiency, and we assess this metric in the structural subnetwork of positive connections and negative connections, separately, as well as in the full network constructed by taking the absolute value of edge strengths in \mathbf{A} . We find consistently that global efficiency of the structural network is higher in functional communities elicited by the Rössler task than in functional communities elicited by the Lorentz task. Specifically, when considering the subnetwork of positive structural connections a two-sample t -test gives $t = -9.86$, $p = 6.66 \times 10^{-19}$ for the first community and $t = -9.83$, $p = 8.3 \times 10^{-19}$ for the second community. Similarly, when considering the subnetwork of negative structural connections, a two-sample t -test gives $t = -6.31$, $p = 1.75 \times 10^{-9}$ for the first community and $t = -9.57$, $p = 4.71 \times 10^{-18}$ for the second community. Finally, when considering the full network constructed by taking the absolute value of edge strengths in \mathbf{A} , a two-sample t -test gives $t = -8.25$, $p = 2.14$.

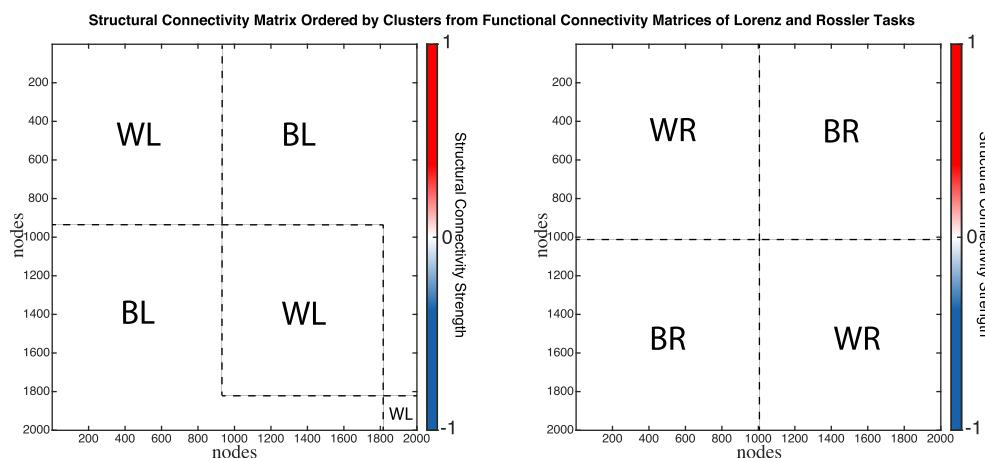


FIG. 14. Structural connectivity matrix ordered based on the community structure from functional connectivity matrices. (Left) The structural connectivity matrix \mathbf{A} ordered based on the functional community structure elicited by the Lorenz task. (Right) The structural connectivity matrix \mathbf{A} ordered based on the functional community structure elicited by the Rössler task. We conclude that the marked sparsity of the structural matrix hampers any explicit inferences from this virtually blank visualization.

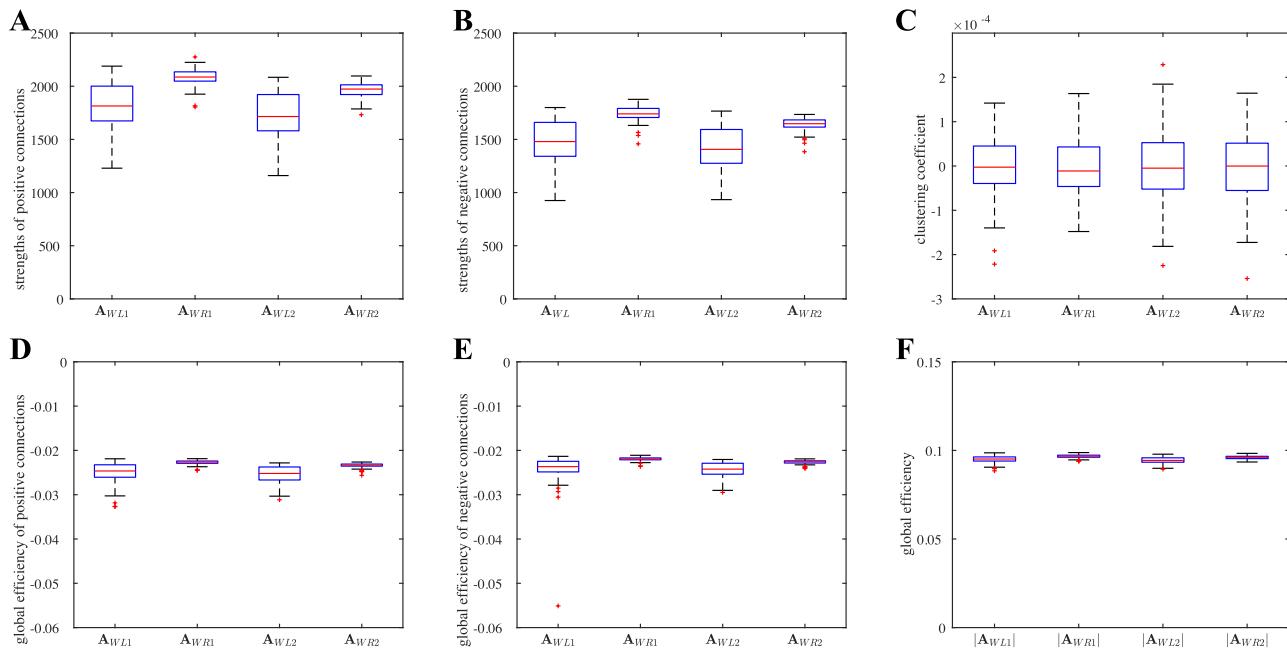


FIG. 15. Topological measures of subnetworks in the structural matrix that comprise the clusters detected from the functional matrices. (a) Strength of positive connections in the structural subnetwork of functional community one and two elicited by the Rössler and Lorentz tasks. (b) Strength of negative connections in the structural subnetwork of functional community one and two elicited by the Rössler and Lorentz tasks. (c) Signed clustering coefficient in the structural subnetwork of functional community one and two elicited by the Rössler and Lorentz tasks. (d) Global efficiency of positive connections in the structural subnetwork of functional community one and two elicited by the Rössler and Lorentz tasks. (e) Global efficiency of negative connections in the structural subnetwork of functional community one and two elicited by the Rössler and Lorentz tasks. (f) Global efficiency of the magnitude of connections in the structural subnetwork of functional community one and two elicited by the Rössler and Lorentz tasks.

$\times 10^{-14}$ for the first community and $t = -7.70$, $p = 6.31 \times 10^{-13}$ for the second community.

Collectively, these results demonstrate that the functional communities elicited by the two tasks (Rössler and Lorentz) differ

in their structural support. This fact is perhaps intuitive given that the functional communities are driven by the input dynamics, which differ appreciably across the two attractors. Yet, further work is needed to understand whether there might be a way of predicting

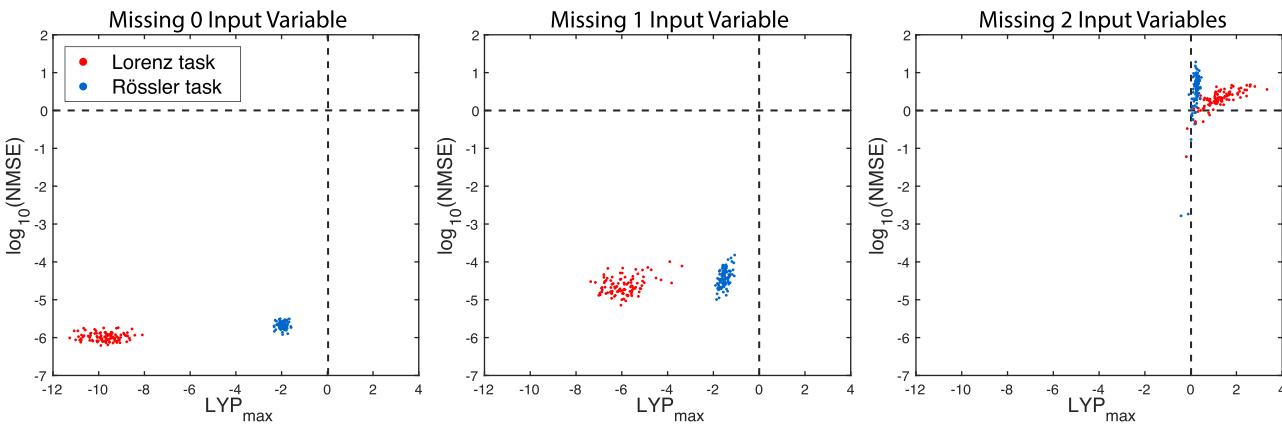


FIG. 16. Supplementary results related to the inference of missing input variables. In contrast to our simulations in the main text where we let each neuron connect to one out of three input channels, here instead we let each neuron connect to all three input channels. Specifically, each of the elements in the input weight matrix $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N \times 3}$ is randomly and uniformly drawn from $[-0.05, 0.05]$. Here, we show the performance of this new system on the task of inferring missing input variables, which we note is comparable to that observed for the original system [see Fig. 6(d) in the main text].

exactly what sort of structural topology is expected in the functional communities elicited by distinct attractors. Finally, we note that these data support the notion that attractor dynamics are not learned by distinct subnetworks of neurons, but instead by all neurons in specific regions in the phase space.

APPENDIX F: MISSING VARIABLE INFERENCE WITH NEURONS RECEIVING MULTIPLE INPUT SIGNALS

In the demonstration given in the main text, the missing variable inference task is accomplished by the central system where each of the N neurons is connected to one of the three input channels. One may suspect that the success of the training is due to the fact that, by our construction, each neuron is only driven by one of the three variables from the external signal during the training phase, and thus, it is fairly simple to infer the three variables based on the responding dynamics of these driven neurons. To better understand this behavior, it is useful to consider whether the system could still perform the inference task if all N neurons in the central system are connected to all three input channels. In this section, we perform additional numerical simulations to directly address this question. Specifically, in contrast to our simulations in the main text, each of the elements in the input weight matrix $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N \times 3}$ is now randomly and uniformly drawn from $[-0.05, 0.05]$. In Fig. 16, we show that this new system's performance on the missing-variable-inference problem is similar to the performance that we observed for the original system where every neuron receives signals from only one input channel [compare to Fig. 6(d) in the main text]. We conclude that the robustness that we observe in Fig. 6 cannot be attributed solely to the fact that each neuron only receives one of the three input variables.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹E. A. Karuza, S. L. Thompson-Schill, and D. S. Bassett, “Local patterns to global architectures: Influences of network topology on human learning,” *Trends Cogn. Sci.* **20**, 629–640 (2016).
- ²A. S. Reber, *Implicit Learning and Tacit Knowledge: An Essay on the Cognitive Unconscious* (Oxford University Press, 1993).
- ³A. Cleeremans, A. Destrebecqz, and M. Boyer, “Implicit learning: News from the front,” *Trends Cogn. Sci.* **2**, 406–416 (1998).
- ⁴G. F. Marcus, S. Vijayan, S. Bandi Rao, and P. M. Vishton, “Rule learning by seven-month-old infants,” *Science* **283**, 77–80 (1999).
- ⁵R. A. Murphy, E. Mondragon, and V. A. Murphy, “Rule learning by rats,” *Science* **319**, 1849–1851 (2008).
- ⁶C. A. Seger, “Implicit learning,” *Psychol. Bull.* **115**, 163–196 (1994).
- ⁷I. Nemenman, “Information theory and learning: A physical approach,” preprint [arXiv:physics/0009032](https://arxiv.org/abs/physics/0009032) (2000).
- ⁸J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, “On dynamic mode decomposition: Theory and applications,” preprint [arXiv:1312.0041](https://arxiv.org/abs/1312.0041) (2013).
- ⁹J. Anderson, I. Kevrekidis, and R. Rico-Martinez, “A comparison of recurrent training algorithms for time series analysis and system identification,” *Comput. Chem. Eng.* **20**, S751–S756 (1996).
- ¹⁰D. L. Ly and H. Lipson, “Learning symbolic representations of hybrid dynamical systems,” *J. Mach. Learn. Res.* **13**, 3585–3618 (2012).
- ¹¹A. Hefny, C. Downey, and G. J. Gordon, “Supervised learning for dynamical system learning,” in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2015), pp. 1963–1971.
- ¹²M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Multistep neural networks for data-driven discovery of nonlinear dynamical systems,” preprint [arXiv:1801.01236](https://arxiv.org/abs/1801.01236) (2018).
- ¹³M. Raissi, “Deep hidden physics models: Deep learning of nonlinear partial differential equations,” *J. Mach. Learn. Res.* **19**, 932–955 (2018).
- ¹⁴G. T. M. Altmann, D. Dienes, and G. F. Marcus, “Rule learning by seven-month-old infants and neural networks,” *Science* **284**, 875 (1999).
- ¹⁵A. C. Schapiro, N. B. Turk-Browne, M. M. Botvinick, and K. A. Norman, “Complementary learning systems within the hippocampus: A neural network modelling approach to reconciling episodic memory with statistical learning,” *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* **372**, 20160049 (2017).
- ¹⁶K. Rajan, C. D. Harvey, and D. W. Tank, “Recurrent network models of sequence generation and memory,” *Neuron* **90**, 128–142 (2016).
- ¹⁷A. Alemi, C. Machens, S. Denève, and J.-J. Slotine, “Learning arbitrary dynamics in efficient, balanced spiking networks using local plasticity rules,” preprint [arXiv:1705.08026](https://arxiv.org/abs/1705.08026) (2017).
- ¹⁸A. Gilra and W. Gerstner, “Predicting non-linear dynamics: a stable local learning scheme for recurrent spiking neural networks,” preprint [arXiv:1702.06463](https://arxiv.org/abs/1702.06463) (2017).
- ¹⁹S. Denève, A. Alemi, and R. Bourdoukan, “The brain as an efficient and robust adaptive learner,” *Neuron* **94**, 969–977 (2017).
- ²⁰L. Abbott, B. DePasquale, and R.-M. Memmesheimer, “Building functional networks of spiking model neurons,” *Nat. Neurosci.* **19**, 350 (2016).
- ²¹W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural Comput.* **14**, 2531–2560 (2002).
- ²²H. Jaeger and H. Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *Science* **304**, 78–80 (2004).
- ²³D. Sussillo and L. F. Abbott, “Generating coherent patterns of activity from chaotic neural networks,” *Neuron* **63**, 544–557 (2009).
- ²⁴J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, “Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data,” *Chaos* **27**, 121102 (2017).
- ²⁵J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, “Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach,” *Phys. Rev. Lett.* **120**, 024102 (2018).
- ²⁶Z. Lu, B. R. Hunt, and E. Ott, “Attractor reconstruction by machine learning,” *Chaos* **28**, 061104 (2018).
- ²⁷G. Mayer-Kress, R. Bargar, and I. Choi, *Musical Structures in Data from Chaotic Attractors* (University of Illinois at Urbana-Champaign, 1992).
- ²⁸J. L. Elman, “Language as a dynamical system,” *Mind as Motion: Explorations in the Dynamics of Cognition* (MIT Press, 1995), pp. 195–225.
- ²⁹R. M. Winters, “Musical mapping of chaotic attractors,” Physics Department, The College of Wooster, Wooster, Ohio, 2009, see http://physics.wooster.edu/JrIS/Files/Winters_Web_article.pdf.
- ³⁰R. Bidlack, “Chaotic systems as simple (but complex) compositional algorithms,” *Comput. Music J.* **16**, 33–47 (1992).
- ³¹P. L. Castilho, “Chaotic systems as compositional algorithms mus-15” (2015).
- ³²J. P. Mackenzie, “Chaotic predictive modelling of sound,” in *ICMC 1995* (1995), see <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.1967&rep=rep1&type=pdf>.
- ³³Y. Kessler, Y. Shencar, and N. Meiran, “Choosing to switch: Spontaneous task switching despite associated behavioral costs,” *Acta Psychol.* **131**, 120–128 (2009).
- ³⁴I. Koch and A. Allport, “Cue-based preparation and stimulus-based priming of tasks in task switching,” *Mem. Cognit.* **34**, 433–444 (2006).
- ³⁵F. C. Constantino and J. Z. Simon, “Dynamic cortical representations of perceptual filling-in for missing acoustic rhythm,” *Sci. Rep.* **7**, 17536 (2017).
- ³⁶H. Komatsu, “The neural mechanisms of perceptual filling-in,” *Nat. Rev. Neurosci.* **7**, 220 (2006).
- ³⁷N. Ding and J. Z. Simon, “Emergence of neural encoding of auditory objects while listening to competing speakers,” *Proc. Natl. Acad. Sci. U.S.A.* **109**, 11854–11859 (2012).

- ³⁸J. Xiang, J. Simon, and M. Elhilali, "Competing streams at the cocktail party: Exploring the mechanisms of attention and temporal integration," *J. Neurosci.* **30**, 12084–12093 (2010).
- ³⁹H. Luo and D. Poeppel, "Phase patterns of neuronal responses reliably discriminate speech in human auditory cortex," *Neuron* **54**, 1001–1010 (2007).
- ⁴⁰N. Mesgarani and E. F. Chang, "Selective cortical representation of attended speaker in multi-talker speech perception," *Nature* **485**, 233 (2012).
- ⁴¹W. James, *The Principles of Psychology* (Henry Holt and Company, New York, 1890).
- ⁴²E. Tulving and D. M. Thomson, "Encoding specificity and retrieval processes in episodic memory," *Psychol. Rev.* **80**, 352 (1973).
- ⁴³M. E. Wheeler, S. E. Petersen, and R. L. Buckner, "Memory's echo: Vivid remembering reactivates sensory-specific cortex," *Proc. Natl. Acad. Sci. U.S.A.* **97**, 11125–11129 (2000).
- ⁴⁴J. E. Kragel, Y. Ezzyat, M. R. Sperling, R. Gorniak, G. A. Worrell, B. M. Berry, C. Inman, J.-J. Lin, K. A. Davis, S. R. Das *et al.*, "Similar patterns of neural activity predict memory function during encoding and retrieval," *NeuroImage* **155**, 60–71 (2017).
- ⁴⁵L. Nyberg, R. Habib, A. R. McIntosh, and E. Tulving, "Reactivation of encoding-related brain activity during memory retrieval," *Proc. Natl. Acad. Sci. U.S.A.* **97**, 11120–11124 (2000).
- ⁴⁶E. Düzel, R. Habib, M. Rotte, S. Guderian, E. Tulving, and H.-J. Heinze, "Human hippocampal and parahippocampal activity during visual associative recognition memory for spatial and nonspatial stimulus configurations," *J. Neurosci.* **23**, 9439–9444 (2003).
- ⁴⁷P. Khader, M. Burke, S. Bien, C. Ranganath, and F. Rösler, "Content-specific activation during associative long-term memory retrieval," *NeuroImage* **27**, 805–816 (2005).
- ⁴⁸C. Ranganath, A. Heller, M. X. Cohen, C. J. Brozinsky, and J. Rissman, "Functional connectivity with the hippocampus during successful memory formation," *Hippocampus* **15**, 997–1005 (2005).
- ⁴⁹C. C. Woodruff, J. D. Johnson, M. R. Uncapher, and M. D. Rugg, "Content-specificity of the neural correlates of recollection," *Neuropsychologia* **43**, 1022–1032 (2005).
- ⁵⁰S. D. Slotnick and D. L. Schacter, "The nature of memory related activity in early visual areas," *Neuropsychologia* **44**, 2874–2886 (2006).
- ⁵¹J. D. Johnson and M. D. Rugg, "Recollection and the reinstatement of encoding-related cortical activity," *Cereb. Cortex* **17**, 2507–2515 (2007).
- ⁵²R. A. Diana, A. P. Yonelinas, and C. Ranganath, "Parahippocampal cortex activation during context reinstatement predicts item recollection," *J. Exp. Psychol. Gen.* **142**, 1287 (2013).
- ⁵³M. D. Rugg and K. L. Vilberg, "Brain networks underlying episodic memory retrieval," *Curr. Opin. Neurobiol.* **23**, 255–260 (2013).
- ⁵⁴S. E. Bosch, J. F. Jehee, G. Fernández, and C. F. Doeller, "Reinstatement of associative memories in early visual cortex is signaled by the hippocampus," *J. Neurosci.* **34**, 7493–7500 (2014).
- ⁵⁵V. Afraimovich, N. Verichev, and M. I. Rabinovich, "Stochastic synchronization of oscillation in dissipative systems," *Radiophys. Q. Electron.* **29**, 795–803 (1986).
- ⁵⁶L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.* **64**, 821–824 (1990).
- ⁵⁷N. F. Rulkov, M. M. Sushchik, L. S. Tsimring, and H. D. Abarbanel, "Generalized synchronization of chaos in directionally coupled chaotic systems," *Phys. Rev. E* **51**, 980 (1995).
- ⁵⁸H. Whitney, "Differentiable manifolds," *Ann. Math.* **37**, 645–680 (1936).
- ⁵⁹C. L. Marvel and J. E. Desmond, "From storage to manipulation: How the neural correlates of verbal working memory reflect varying demands on inner speech," *Brain Lang.* **120**, 42–51 (2012).
- ⁶⁰M. Perrone-Bertolotti, L. Rapin, J.-P. Lachaux, M. Baciu, and H. Loevenbruck, "What is that little voice inside my head? Inner speech phenomenology, its role in cognitive performance, and its relation to self-monitoring," *Behav. Brain Res.* **261**, 220–239 (2014).
- ⁶¹A. D. Baddeley, G. J. Hitch, and G. A. Bower, "Recent advances in learning and motivation," *Work. Memory* **8**, 647–667 (1974).
- ⁶²A. Morin, B. Utzl, and B. Hamper, "Self-reported frequency, content, and functions of inner speech," *Proc. Soc. Behav. Sci.* **30**, 1714–1718 (2011).
- ⁶³D. M. Williams, D. M. Bowler, and C. Jarrold, "Inner speech is used to mediate short-term memory, but not planning, among intellectually high-functioning adults with autism spectrum disorder," *Dev. Psychopathol.* **24**, 225–239 (2012).
- ⁶⁴B. Alderson-Day and C. Fernyhough, "Inner speech: Development, cognitive functions, phenomenology, and neurobiology," *Psychol. Bull.* **141**, 931 (2015).
- ⁶⁵M. Scott, "Corollary discharge provides the sensory content of inner speech," *Psychol. Sci.* **24**, 1824–1830 (2013).
- ⁶⁶D. R. Addis, A. T. Wong, and D. L. Schacter, "Remembering the past and imagining the future: Common and distinct neural substrates during event construction and elaboration," *Neuropsychologia* **45**, 1363–1377 (2007).
- ⁶⁷S. M. Kosslyn, G. Ganis, and W. L. Thompson, "Neural foundations of imagery," *Nat. Rev. Neurosci.* **2**, 635 (2001).
- ⁶⁸A. R. Halpern and R. J. Zatorre, "When that tune runs through your head: A pet investigation of auditory imagery for familiar melodies," *Cereb. Cortex* **9**, 697–704 (1999).
- ⁶⁹R. L. Buckner and D. C. Carroll, "Self-projection and the brain," *Trends Cogn. Sci.* **11**, 49–57 (2007).
- ⁷⁰Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos* **27**, 041102 (2017).
- ⁷¹J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Phys. Rev. Lett.* **120**, 024102 (2018).
- ⁷²G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Control Signals Syst.* **2**, 183–192 (1989).
- ⁷³Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, "Towards biologically plausible deep learning," preprint arXiv:1502.04156 (2015).
- ⁷⁴M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks," *Not. Am. Math. Soc.* **56**, 1082–1097 (2009).
- ⁷⁵M. E. J. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. U.S.A.* **103**, 8577–8582 (2006).
- ⁷⁶V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.* **2008**, P10008 (2008).
- ⁷⁷S. Krishnagopal, M. Girvan, E. Ott, and B. R. Hunt, "Separation of chaotic signals by reservoir computing," *Chaos* **30**, 023123 (2020).
- ⁷⁸Z. Lu, J. Z. Kim, and D. S. Bassett, "Supervised chaotic source separation by a tank of water," *Chaos* **30**, 021101 (2020).
- ⁷⁹B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science* **350**, 1332–1338 (2015).
- ⁸⁰P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding," *Proc. Natl. Acad. Sci. U.S.A.* **110**, 18327–18332 (2013).
- ⁸¹J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman, "How to grow a mind: Statistics, structure, and abstraction," *Science* **331**, 1279–1285 (2011).
- ⁸²N. Chater, M. Oaksford, U. Hahn, and E. Heit, "Bayesian models of cognition," *Wiley Interdisciplin. Rev. Cognit. Sci.* **1**, 811–823 (2010).
- ⁸³T. L. Griffiths, C. Kemp, and J. B. Tenenbaum, "Bayesian models of cognition," in *The Cambridge Handbook of Computational Psychology* (2008).
- ⁸⁴C. Kemp and J. B. Tenenbaum, "The discovery of structural form," *Proc. Natl. Acad. Sci. U.S.A.* **105**, 10687–10692 (2008).
- ⁸⁵N. Chater, J. B. Tenenbaum, and A. Yuille, "Probabilistic models of cognition: Conceptual foundations," *Trends Cogn. Sci.* **10**, 287–291 (2006).
- ⁸⁶J. B. Tenenbaum, T. L. Griffiths, and C. Kemp, "Theory-based Bayesian models of inductive learning and reasoning," *Trends Cogn. Sci.* **10**, 309–318 (2006).
- ⁸⁷S. C. Venkataramani, B. R. Hunt, E. Ott, D. J. Gauthier, and J. C. Biengfang, "Transitions to bubbling of chaotic systems," *Phys. Rev. Lett.* **77**, 5361 (1996).
- ⁸⁸A. Kuznetsov, S. Kuznetsov, and N. Stankevich, "A simple autonomous quasiperiodic self-oscillator," *Commun. Nonlinear Sci. Numer. Simul.* **15**, 1676–1681 (2010).
- ⁸⁹D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature* **393**, 440 (1998).
- ⁹⁰S. F. Muldoon, E. W. Bridgeford, and D. S. Bassett, "Small-world propensity and weighted brain networks," *Sci. Rep.* **6**, 22057 (2016).