ARTICLES
https://doi.org/10.1038/s42256-021-00321-2
nature machine intelligence

Check for updates

Teaching recurrent neural networks to infer global temporal structure from local examples

Jason Z. Kim[1], Zhixin Lu[1], Erfan Nozari[2,3], George J. Pappas[2] and Danielle S. Bassett[4,5] ✉

The ability to store and manipulate information is a hallmark of computational systems. Whereas computers are carefully engineered to represent and perform mathematical operations on structured data, neurobiological systems adapt to perform analogous functions without needing to be explicitly engineered. Recent efforts have made progress in modelling the representation and recall of information in neural systems. However, precisely how neural systems learn to modify these representations remains far from understood. Here, we demonstrate that a recurrent neural network (RNN) can learn to modify its representation of complex information using only examples, and we explain the associated learning mechanism with new theory. Specifically, we drive an RNN with examples of translated, linearly transformed or pre-bifurcated time series from a chaotic Lorenz system, alongside an additional control signal that changes value for each example. By training the network to replicate the Lorenz inputs, it learns to autonomously evolve about a Lorenz-shaped manifold. Additionally, it learns to continuously interpolate and extrapolate the translation, transformation and bifurcation of this representation far beyond the training data by changing the control signal. Furthermore, we demonstrate that RNNs can infer the bifurcation structure of normal forms and period doubling routes to chaos, and extrapolate non-dynamical, kinematic trajectories. Finally, we provide a mechanism for how these computations are learned, and replicate our main results using a Wilson–Cowan reservoir. Together, our results provide a simple but powerful mechanism by which an RNN can learn to manipulate internal representations of complex information, enabling the principled study and precise design of RNNs.

Computers analyse massive quantities of data with speed and precision. At both the hardware and software levels, this performance depends on fixed and precisely engineered protocols for representing and executing basic operations on binary data[1,2]. By contrast, neurobiological systems are characterized by flexibility and adaptability. At the biophysical level, neurons undergo dynamic changes in their composition and patterns of connectivity[3–6]. At the cognitive level, they abstract spatiotemporally complex sensory information to recognize objects, localize spatial position and even control new virtual limbs through experience[7–9]. Hence, neural systems appear to work on fundamentally different computing principles that are learned, rather than engineered.

To uncover these principles, artificial neural networks have been used to study the representation and manipulation of information. Although feed-forward networks can classify input data[10], biological organisms contain recurrent connections that are necessary to continuously sustain short-term memory of internal representations[11], allowing for more complex functions such as tracking time, distance and emotional context[12–16]. Furthermore, recurrent neural systems actually manipulate internal representations, as observed in meta-learning and adaptive networks[17–20], to simulate the outcome of dynamic processes such as kinematic motion and navigation[21–23], and to decide between different actions[24]. So, how do recurrent neural systems learn to represent and manipulate complex information?

One promising line of work involves representing static memories as patterns of neural activity, or 'attractors', to which a network evolves over time[25]. These attractors can exist in isolation (for example, an image of a face) or as a continuum (for example, smooth translations of a face) using Hopfield or continuous attractor neural networks (CANNs), respectively[26,27]. Other studies engineer neural connectivity as in the Neural Engineering Framework or the differentiable neural computer to encode, modify and decode internal representations or to solve complex puzzles[28–30]. For understanding neurobiological systems, these memory networks are limited by requiring specifically engineered patterns of connectivity, or cannot manipulate time-varying memories necessary to plan and produce speech and music[31–33]. Hence, we seek a single neural system that learns to both represent and manipulate temporally complex information by perceiving and replicating examples.

In this Article, we use the 'reservoir computing framework'[34] to obtain such a system (the reservoir), where the complex information is a chaotic attractor that is not static, but evolves in a deterministic yet unpredictable manner through time[35]. Previous work has demonstrated the ability of an RNN to represent and switch between isolated attractors by imitating examples[36,37], and to interpolate and extrapolate underlying generative dynamical processes in time-series prediction tasks using fixed-weight neural networks (FWNNs)[19,20,38,39]. In this Article, we demonstrate that reservoirs can further learn to interpolate and extrapolate translations, linear transformations and even bifurcations on their representations of chaotic attractor manifolds, simply by imitating examples. Furthermore, reservoirs can infer the bifurcation structure of dynamical normal forms and period doubling routes to chaos, as well as accurately extrapolate non-dynamical, kinematic trajectories. Finally, we put forth a mechanism of how these computations are learned, providing insights into the set of possible computations and offering principles by which to design networks.

[1]Department of Bioengineering, University of Pennsylvania, Philadelphia, PA, USA. [2]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. [3]Department of Mechanical Engineering, University of California, Riverside, CA, USA. [4]Departments of Bioengineering, Physics & Astronomy, Electrical & Systems Engineering, Neurology, and Psychiatry, University of Pennsylvania, Philadelphia, PA, USA. [5]Santa Fe Institute, Santa Fe, NM, USA. ✉e-mail: dsb@seas.upenn.edu

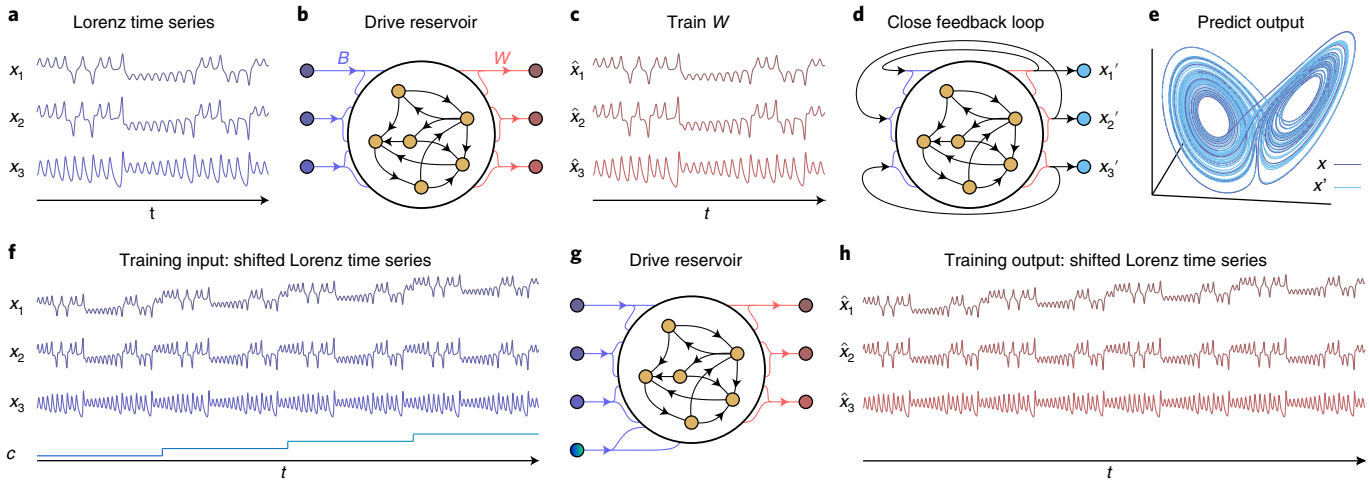


Fig. 1 | Representing chaotic attractors with reservoirs. a,**b**, Time series of a chaotic Lorenz attractor (**a**) that drives the RNN reservoir (**b**). **c**, Weighted sums of the reservoir states are trained to reproduce the original time series. **d**,**e**, By using these weighted sums of reservoir states to drive the reservoir instead of the inputs (**d**), the reservoir autonomously evolves along a trajectory that projects to a Lorenz-shaped chaotic manifold (**e**). **f**,**g**, Time series of shifted copies of the Lorenz input along x_1 (**f**) and control inputs that drive the reservoir (**g**). **h**, Weighted sums of the reservoir states are used to mimic the shifted Lorenz inputs.

Mathematical framework

Neural systems represent and manipulate periodic stimuli through example, such as baby songbirds modifying their song to imitate adult songbirds[32]. However, they also perform more advanced and original manipulations on aperiodic stimuli with higher-order structure, such as musicians improvising on jazz melodies[33]. To model such complex stimuli, we use chaotic attractors that evolve deterministically yet unpredictably along a global structure: a fractional-dimensional manifold. Specifically, we consider the Lorenz attractor defined as

$$\begin{aligned}\dot{x}_1 &= \sigma(x_2 - x_1)\\ \dot{x}_2 &= x_1(\rho - x_3) - x_2\\ \dot{x}_3 &= x_1 x_2 - \beta x_3\end{aligned} \tag{1}$$

and use the parameters $\sigma = 10, \beta = 8/3, \rho = 28$ from the original study[35].

Next, we model the neural system as an RNN driven by our inputs:

$$\frac{1}{\gamma}\dot{\mathbf{r}} = -\mathbf{r} + \mathbf{g}\left(A\mathbf{r} + B\mathbf{x} + \mathbf{d}\right) \tag{2}$$

where $\mathbf{r}$ is a real-valued vector of N reservoir neuron states, A is an $N \times N$ matrix of connections between neurons, B is an $N \times M$ matrix of connections from the M inputs to the neurons, $\mathbf{d}$ is an $N \times 1$ bias vector, $\mathbf{g} = [g_1; g_2; \ldots g_N]$ is a vector function comprising scalar activation functions g_i that are applied entry-wise to its input arguments (hence mapping vectors to vectors), and γ is a time constant. We use $g_1 = g_2 = \cdots = g_N = \tanh$ in the main text, and replicate many results using Wilson–Cowan oscillators in the Supplementary Information.

Several previous studies used echo state[36] and first-order reduced and controlled error (FORCE) learning[37], which allow reservoirs to predict a chaotic time series by modifying the inter-neuron connections. This modification can be accomplished by using the chaotic time series $\mathbf{x}(t)$ to drive the reservoir, thereby generating the reservoir time series $\mathbf{r}(t)$ (Fig. 1a,b). Here, we represent the time series vectors $\boldsymbol{x}(t)$ and $\boldsymbol{r}(t)$ (written using bold letters) as $M \times T$ and $N \times T$ matrices $x(t)$ and $r(t)$, respectively, from numerically evolving the differential equations over T time steps. By solving for a simple $M \times N$ readout matrix W that uses linear combinations of reservoir states to approximate the input by minimizing the matrix 2-norm (Supplementary Information)

$$W = \arg\min_W \| Wr(t) - x(t) \|_2 \tag{3}$$

the output $\hat{x}(t) = Wr(t)$ mimics the input $x(t)$ (Fig. 1c). Finally, we close the feedback loop by substituting the output as the input to create the autonomous reservoir (Fig. 1d)

$$\frac{1}{\gamma}\dot{\mathbf{r}}' = -\mathbf{r}' + \mathbf{g}\left((A + BW)\mathbf{r}' + \mathbf{d}\right) \tag{4}$$

whose evolution projects to a Lorenz-shaped attractor as $x'(t) = Wr'(t)$ (Fig. 1e). Hence, reservoirs sustain representations of complex temporal information by learning to autonomously evolve along a chaotic attractor from example inputs.

To study how reservoirs might perform computations by manipulating these representations, we use a modified framework[37] to include a vector, $\mathbf{c}$, of control parameters that map to the reservoir neurons through matrix C to yield

$$\frac{1}{\gamma}\dot{\mathbf{r}} = -\mathbf{r} + \mathbf{g}\left(A\mathbf{r} + B\mathbf{x} + C\mathbf{c} + \mathbf{d}\right) \tag{5}$$

We drive the reservoir with the original $x_0(t)$ and modified $x_c(t)$ versions of the input at varying values of c to generate the corresponding reservoir time series $r_0(t)$, $r_c(t)$. Then, by concatenating the time series along the time dimension for the inputs $x(t) = [x_0(t), x_1(t), \ldots]$ and the reservoir states $r(t) = [r_0(t), r_1(t), \ldots]$, we train a readout matrix according to equation (3) (Fig. 1f–h). In what follows, we demonstrate that the feedback reservoir not only autonomously evolves about the input trajectory as a stable memory, but also learns to interpolate and extrapolate the modification of this memory far outside of the training range.

Learning a translation operation by example

Reservoirs learn complex information through simple imitation—approximating the driving inputs using the reservoir states is enough to autonomously represent and evolve about a chaotic manifold. Here, we show that this simple scheme is also enough to learn to translate and transform the representation.

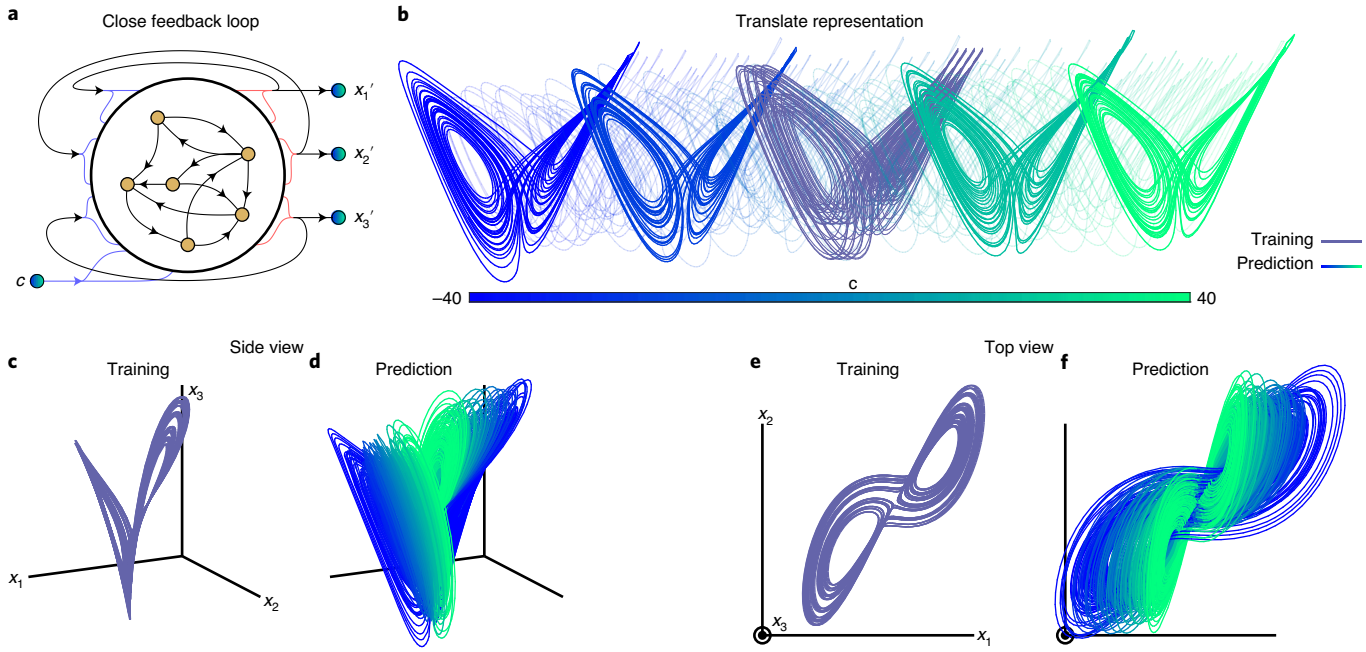


Fig. 2 | Learning and extrapolating translations and transformations by example. a, Schematic of a feedback reservoir where the outputs $W\mathbf{r}(t)$ replace the inputs $\mathbf{x}(t)$ to create a closed feedback loop, where c remains a changeable parameter. **b**, When the reservoir is trained on four Lorenz time series translated in the x_1 dimension (purple), the feedback reservoir evolves autonomously about a Lorenz-shaped manifold, and translates this representation along x_1 over the course of one simulation by smoothly and continuously changing c as a real number over a range much larger than the training range. **c**–**f**, When the reservoir is trained on four linearly transformed (squeeze in the x_1 direction, purple) time series (**c**,**e**), the feedback reservoir similarly interpolates and extrapolates the transformation of its representation (**d**,**f**).

We begin with a Lorenz time series $x_0(t)$, and create shifted copies

$$x_c(t) = x_0(t) + Pc \tag{6}$$

For the purposes of demonstration, we consider a translation in the x_1 direction such that $P=[1; 0; 0]$ is a column vector, and $c=0, 1, 2, 3$ is a scalar. We use these four time series to drive our reservoir according to equation (5), thereby generating four reservoir time series $r_c(t)$. We concatenate these series into $r(t)$ and $x(t)$, and compute output weights according to equation (3), such that our output $\hat{x}(t) = Wr(t)$ approximates our input $x(t)$ (Fig. 1f–h). Finally, we substitute the output as the input to yield the feedback system (Fig. 2a)

$$\frac{1}{\gamma}\dot{\mathbf{r}}' = -\mathbf{r}' + \mathbf{g}(R\mathbf{r}' + C\mathbf{c} + \mathbf{d}) \tag{7}$$

where $R=A+BW$. As we evolve this autonomous reservoir while varying c to extreme values $-40\leq c\leq 40$ both inside and outside of the training values, it has learned to evolve about a Lorenz-shaped manifold that is translated based on the value of c (Fig. 2b; Supplementary Information presents more examples and information about the Wilson–Cowan implementation).

We use the same scheme to teach reservoirs linear transformations. We begin with the Lorenz time series $x_0(t)$ and create linearly transformed copies of the time series such that

$$x_c(t) = (I + cP)x_0(t) \tag{8}$$

for $c=0, 1, 2, 3$, where P is a matrix encoding a transformation (Fig. 2c,e). Specifically, we perform a squeeze along x_1 by setting $[P]_{11}=-0.012$ and the remaining elements to 0. Exactly as before, we drive the reservoir according to equation (5), concatenate our input and reservoir time series into $x(t)$ and $r(t)$ to train the output weights W according to equation (3), and feed the outputs back as inputs to yield the feedback system equation (7). This reservoir autonomously evolves about a Lorenz-shaped manifold that stretches based on the parameter $-40\leq c\leq 40$ far outside of the parameters used in the training regime $c=0, 1, 2, 3$ (Fig. 2d,f; Supplementary Information provides examples and information about the Wilson–Cowan implementation). Hence, by training the network on translated and linearly transformed copies of the input, the reservoir has learned translation and linear transformation operations on the attractor.

Learning to infer bifurcation structure by example

Next, we demonstrate that a reservoir can infer, without actually ever having observed, a much more dramatic change: a bifurcation. The first bifurcation occurs in the typical parameter regime (equation (1) with $\rho\approx 28$, $\sigma=10$, $\beta=8/3$). Here, the Lorenz system contains two fixed points, $\mathbf{x}_\rho^{*a}$ and $\mathbf{x}_\rho^{*b}$, that undergo a subcritical Hopf bifurcation when $\rho=\rho^\star\approx 24.7$ (ref. [25]). When $\rho<\rho^\star$, these two fixed points are stable. When $\rho>\rho^\star$, the fixed points become unstable, yielding the characteristic wing-shaped flow.

To infer the bifurcation, we drive the reservoir with four training trajectories: $x_{23}^a(t)$ and $x_{23}^b(t)$ that evolve stably towards the fixed points for $\rho=23$ while $c=0$, and $x_{24}^a(t)$ and $x_{24}^b(t)$ that evolve stably towards the fixed points for $\rho=24$ while $c=1$ (Fig. 3a). Importantly, we stress that the inputs include the transient spiral trajectory towards the fixed point. By training the output weights and evolving the feedback reservoir while changing c from 0 to 5, the trajectory bifurcates into a Lorenz-shaped attractor (Fig. 3b). By subsequently changing c back to 0, the reservoir displays transient chaos, whereby it temporarily evolves about a Lorenz-shaped attractor, but eventually falls into a stable fixed point (Fig. 3c). Finally, by changing c back to 5, the fixed points again become unstable (Fig. 3d).

In addition to inferring the Lorenz system's bifurcation and chaotic manifold geometry, reservoirs can accurately infer more detailed

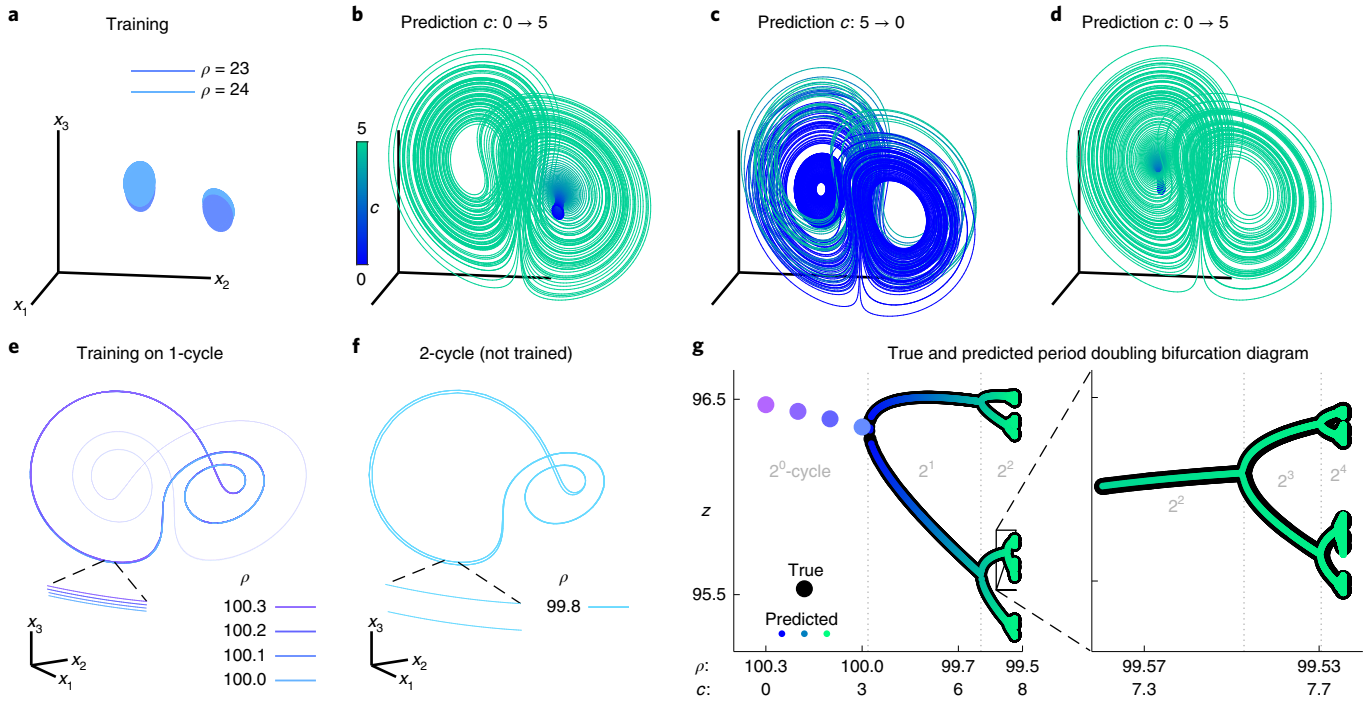


Fig. 3 | Inferring and extrapolating the bifurcation of the Lorenz. a, Two training trajectories for each of the stable Lorenz fixed points at the wings, for $\rho = 23$ with $c = 0$ (blue) and for $\rho = 24$ with $c = 1$ (light blue). **b**, The predicted trajectory of the feedback reservoir moves towards a stable fixed point for $c = 0$, and bifurcates into a Lorenz-shaped manifold as c is increased to 5. **c**, As c is decreased back to 0, the predicted trajectory eventually falls into a stable fixed point. **d**, As c is increased back to 5, the fixed point loses stability. **e**, Four training 1-cycle trajectories for each wing (left wing, dark; right wing, light) of the Lorenz at $\rho = 100.3$, 100.2, 100.1 and 100. **f**, Example of a Lorenz bifurcation into a 2-cycle at $\rho = 99.8$ that is not used in training. **g**, True (black) and predicted (blue to green) Poincaré sections of the true Lorenz and predicted reservoir time series along the plane $x = 0$ for $\dot{x} > 0$[54].

features of the Lorenz system's period doubling route to chaos in the parameter regime $\rho \approx 100$, $\sigma = 10$, $\beta = 8/3$. For $\rho \gtrapprox 99.98$, the Lorenz evolves about a stable 1-cycle (Fig. 3e). Between $99.95 \gtrapprox \rho \gtrapprox 99.63$, the Lorenz bifurcates into a stable 2-cycle (Fig. 3f). As we continue to decrease ρ, the number of cycles doubles until the flow becomes chaotic. We capture the period doubling phenomenon using a Poincarré section along the plane $x = 0$ for $\dot{x} > 0$ (see Supplementary Information for details).

In this demonstration, we drive the reservoir with eight 1-cycle trajectories comprising four values of ρ at both wings (Fig. 3e), and perform feedback. As we increase c, the reservoir undergoes a series of period doubling bifurcations that almost exactly trace the true Lorenz system as measured by their Poincaré sections (Fig. 3g). Hence, we teach reservoirs to quantitatively infer highly nonlinear, latent and unobserved bifurcations.

Mechanism of how operations are learned

Now that we have taught reservoirs to manipulate chaotic manifolds, we seek to understand the mechanism by explicitly relating the change in the control input, $\mathrm{d}c$, to the change in reservoir trajectory, $\mathrm{d}r'(t)$. We consider the time series $r'(t) = r'_{c=0}(t)$ generated by evolving the autonomous reservoir according to equation (7) at $c = 0$. Next, we take the total derivative of equation (7) evaluated at $r'(t)$ and $c = 0$ to yield

$$(I - KA)\mathrm{d}r' + \frac{1}{\gamma}\mathrm{d}\dot{r}' = K(BW\mathrm{d}r' + C\mathrm{d}c) \tag{9}$$

where $K = \mathrm{diag}(\mathrm{d}\mathbf{g}_{r'=r'(t),\, c=0})$, and aim to write $\mathrm{d}r'(t)$ as a function of $\mathrm{d}c$.

When learning translations, the output weights are trained such that $Wr_c(t) \approx x_c(t) = x(t) + Pc$. For sufficiently nearby training examples (small P, c), we also implicitly approximate the differential relation $W\mathrm{d}r(t) \approx P\mathrm{d}c$. Additionally, if the feedback reservoir stabilizes these examples, then $W\mathrm{d}r'(t) \approx P\mathrm{d}c$. Substituting this relation into equation (9) yields

$$(I - KA)\mathrm{d}r' + \frac{1}{\gamma}\mathrm{d}\dot{r}' \approx K(BP + C)\mathrm{d}c \tag{10}$$

If we fix $\mathrm{d}c$, we have $2N$ variables, $\mathrm{d}r'$ and $\mathrm{d}\dot{r}'$, but only N equations. By taking the time derivative of the differential relation, we generate other N variables and N equations. Continuing to take time derivatives yields the following system of equations

$$\begin{bmatrix} H_0 & H_{-1} & 0 & \cdots \\ H_1 & H_0 & H_{-1} & \cdots \\ H_2 & 2H_1 & H_0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathrm{d}r' \\ \mathrm{d}\dot{r}' \\ \mathrm{d}\ddot{r}' \\ \vdots \end{bmatrix} \approx \begin{bmatrix} K \\ \dot{K} \\ \ddot{K} \\ \vdots \end{bmatrix} (BP + C)\mathrm{d}c \tag{11}$$

where $H_{-1} = \frac{1}{\gamma}I$, $H_0 = I - KA$, and $H_i = -K^{(i)}A$ is the ith time derivative of KA. This matrix is a block-Hessenberg matrix, with an analytic solution[40] for the first term $\mathrm{d}r'$. We truncate this solution (Supplementary Information) to explicitly relate $\mathrm{d}r'$ to $\mathrm{d}c$ as follows:

$$\mathrm{d}r' \approx -\left[\gamma H_0^2 - H_1\right]^{-1}\left[-\gamma H_0 \;\; I\right]\begin{bmatrix} K \\ \dot{K} \end{bmatrix}(BP + C)\mathrm{d}c \tag{12}$$

As a demonstration, we pick a finite $\Delta c = 20$, and plot the original and predicted change in the reservoir states, and their outputs in spatial coordinates (Fig. 4b). Hence, using only the feedback dynamics equation (7) and sufficiently nearby training examples,

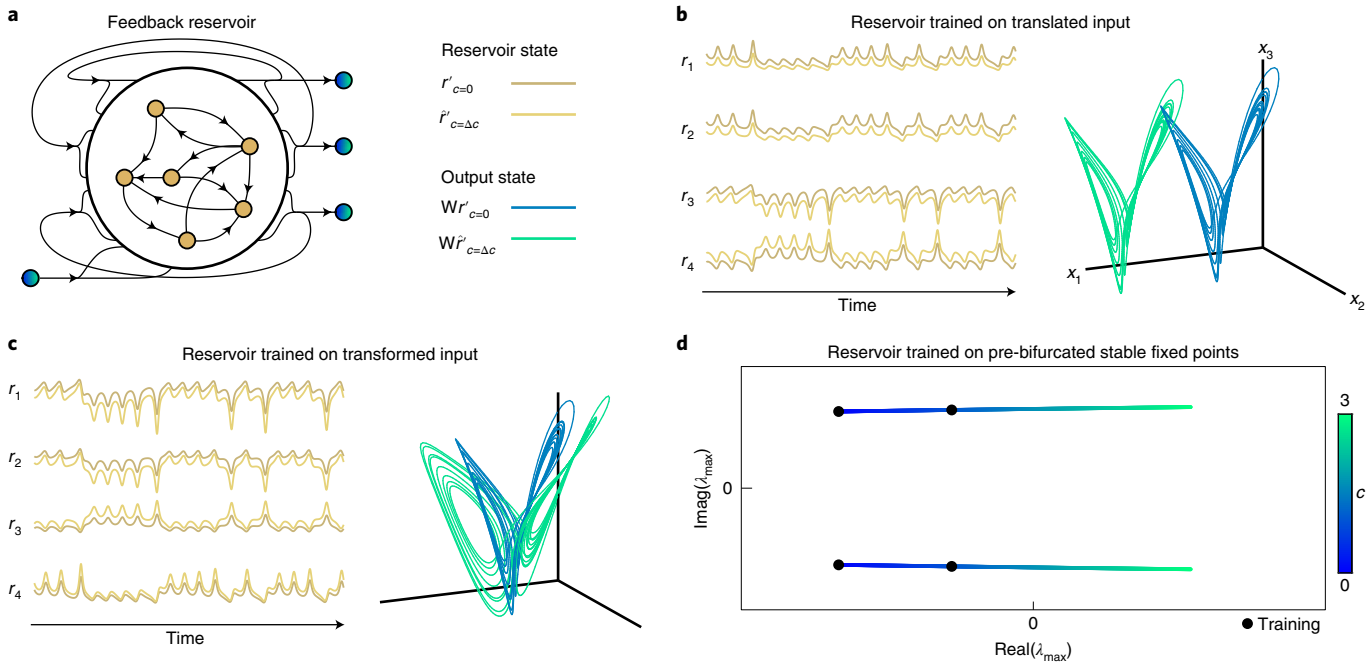


Fig. 4 | Changing the control parameter changes the reservoir dynamics to manipulate representations. a, Schematic of a reservoir with feedback connections after the output weights W have been trained. **b**, Reservoir time series generated by evolving the autonomous reservoir with the original Lorenz input with $c=0$ (dark gold). We also show the predicted time series from solving equation (12) after training on translated examples and computing $\Delta c\frac{d\mathbf{r}'}{dc}$, where $\Delta c=20$ (light gold). The output projections of the two time series are shown in blue and green, respectively. **c**, The original and predicted reservoir states and their output projections for $\Delta c=-40$ after training on transformed Lorenz inputs by solving equation (13). **d**, Plot of the real and imaginary components of the two most unstable eigenvalues of the autonomous reservoir trained on two stable Lorenz trajectories (Fig. 3a). The reservoir is linearized about its equilibrium point $\mathbf{r}_c^*$ as we change c^*.

changing c causes changes in the reservoir states from equation (12) that map to a translation.

The same approach can be used for linear transformations, where the output weights are trained such that $W\mathbf{r}_c(t)\approx x_c(t)=(I+cP)x(t)$. For sufficiently nearby training examples, we implicitly approximate the differential relation $W\mathrm{d}\mathbf{r}(t)\approx Px(t)\mathrm{d}c\approx PW\mathbf{r}(t)\mathrm{d}c$, which, if properly stabilized, yields $W\mathrm{d}\mathbf{r}'(t)\approx PW\mathbf{r}'(t)\mathrm{d}c$. Performing the same time derivatives and solution truncation as in the translation, we get the following relation between $\mathrm{d}c$ and $\mathrm{d}\mathbf{r}'$:

$$\mathrm{d}\mathbf{r}'\approx-\left[\gamma H_0^2-H_1\right]^{-1}\left[-\gamma H_0\ \ I\right]\begin{bmatrix}K(BPW\mathbf{r}'+C)\\ \dot{K}(BPW\mathbf{r}'+C)+KBPW\dot{\mathbf{r}}'\end{bmatrix}\mathrm{d}c \tag{13}$$

As another demonstration, we set $\Delta c=-40$, and plot the original and predicted change in the reservoir states, and their outputs (Fig. 4c).

Finally, to understand how the reservoir is able to infer a bifurcation, we demonstrate that it learns a smooth translation of eigenvalues. Specifically, at ρ^*, the fixed points at the wings of the Lorenz system undergo a Hopf bifurcation, whereby the real component of complex conjugate eigenvalues goes from negative to positive. To track the eigenvalues of the autonomous reservoir, we linearize equation (7) about the equilibrium point $\mathbf{r}_c^*, c^*$ (Supplementary Information). Then, using the output weights trained only on stable Lorenz trajectories (at $c=0$, $\rho=23$ and $c=1$, $\rho=24$; Fig. 3a,b), we track the autonomous reservoir's two most unstable eigenvalues (largest real component) at the fixed point as we vary the control parameter from $c=0$ to $c=3$. We find that these eigenvalues are complex conjugates whose real components go from negative to positive (Fig. 4d). Hence, we demonstrate that not only can reservoirs learn smooth translations and transformations by mapping $\mathrm{d}c$ to $\mathrm{d}\mathbf{r}'$, but they can also perform bifurcations by learning smooth changes in their eigenvalues.

Bifurcation normal forms and non-dynamical time series

To demonstrate the generalizability of these principles, we teach reservoirs to infer the bifurcation diagrams of other dynamical normal forms, and to extrapolate non-dynamical time-series data. We begin with a saddle-node bifurcation with dynamical normal form

$$\dot{x}=a+x^2 \tag{14}$$

where a is the bifurcation parameter. When $a<0$, the system has a stable fixed point at $-\sqrt{-a}$ and an unstable fixed point at $\sqrt{-a}$. When $a>0$, the system has no fixed points. We first generate two time series at each of four values of $a<0$: one above the stable fixed point and one below (Fig. 5a). We then drive the reservoir with these inputs while varying c, train the output weights, and track the feedback reservoir's fixed points and largest real eigenvalue component as we change the control parameter c (Supplementary Information). We observe that the feedback reservoir's fixed points almost exactly trace both the position and stability of the real saddle-node normal form as we vary a far outside of the training regime (Fig. 5b).

Next, we consider the supercritical pitchfork bifurcation with normal form

$$\dot{x}=ax-x^3 \tag{15}$$

When $a<0$, equation (15) has one stable fixed point at $x=0$. When $a>0$, equation (15) has two stable fixed points at $\pm\sqrt{a}$, and one unstable fixed point at $x=0$. We again generate two time series at each of four values of $a<0$: one where $x(t)>0$, and another where $x(t)<0$ (Fig. 5c). We then drive the reservoir while varying c, train

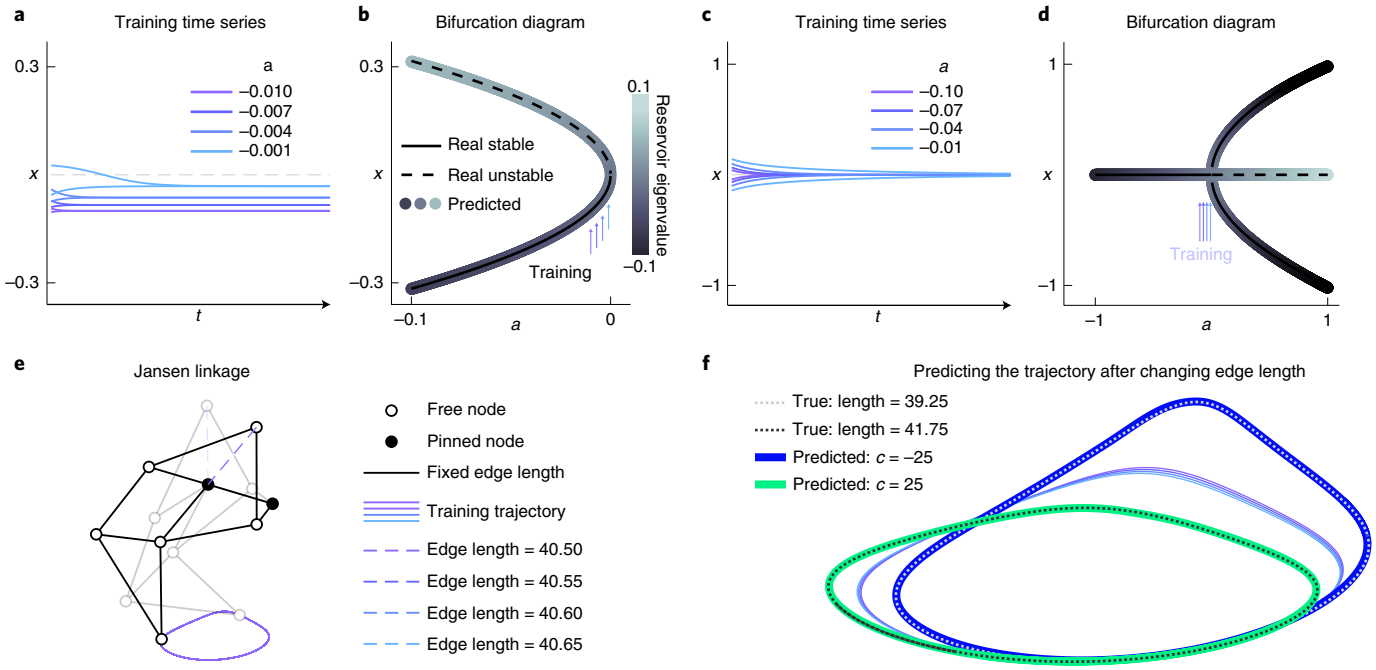


Fig. 5 | Inferring bifurcation normal forms and extrapolating kinematic trajectories. a, Training time series of the saddle-node normal form, where $c=0$ for $a=-0.01$, $c=1$ for $a=-0.007$, $c=2$ for $a=-0.004$, and $c=3$ for $a=-0.001$. **b**, Bifurcation diagram of both the saddle-node normal form as we vary a, and the projection of reservoir fixed points $W\mathbf{r}_c^*$ coloured by the largest real eigenvalue of the reservoir Jacobian at $\mathbf{r}_c^*$ as we vary c. **c**,**d**, Training time series of the supercritical pitchfork normal form, where $c=0$ for $a=-0.1$, $c=1$ for $a=-0.07$, $c=2$ for $a=-0.04$, and $c=3$ for $a=-0.01$ (**c**), along with the corresponding real and predicted bifurcation diagram (**d**). **e**, Jansen linkage, which is pinned at the black nodes, leaving one degree of freedom whereby the bottom joint traces a cyclic trajectory. **f**, Training (light blue to purple) and testing (dotted lines) trajectories of the true Jansen linkage by varying the dashed link, and the predicted reservoir trajectories (blue, green).

the output weights, and again observe that the feedback reservoir's fixed points almost exactly trace both the position and stability of the pitchfork normal form's fixed points (equation (15) and Fig. 5d). We emphasize that in both normal forms, the reservoir was able to infer the location and stability of all fixed points after only observing a few trajectories to one of the stable fixed points.

Finally, we test whether the reservoir's ability to extrapolate generalizes to non-dynamical time series. Specifically, we consider the kinematic trajectory of a modified version of the Jansen linkage[41], whose bottom joint traces out a cyclical walking path with one degree of freedom (Fig. 5e). As the lengths of the links change, the geometry of the traced trajectory also changes in a complex manner. Additionally, the evolution of the original and changed trajectories cannot be written as a dynamical system of the form $\dot{x}=f(x)$. We first generate four training trajectories by changing the length of the dashed link by $\Delta l=0.05$ while varying c by $\Delta c=1$ (Fig. 5e,f), and perform training and feedback. Then, we use the feedback reservoir to generate predicted trajectories at $c=-25$ and $c=25$, and observe that they very closely follow the true trajectories when the link length changes by $-25\Delta l$ and $25\Delta l$ (Fig. 5f).

Simultaneous learning of multiple operations

To close, we demonstrate that reservoirs can easily learn multiple computations by changing multiple control inputs. We train a translation in the x_1 direction with control parameter c_1, and a translation in the x_3 direction with control parameter c_2. As before, we begin with a Lorenz time series $x_{0,0}(t)$ generated from equation (1), and created shifted copies

$$x_{c_1,c_2}(t)=x_{0,0}(t)+c_1a_1+c_2a_2 \tag{16}$$

where $a_1=[1;\ 0;\ 0]$ corresponds to an x_1 shift, and $a_2=[0;\ 0;\ 1]$ corresponds to an x_3 shift. We generate 10 shifted inputs, with one unshifted attractor ($c_1=0$, $c_2=0$), three shifts in the x_1 direction ($c_1=1, 2, 3$, $c_2=0$), three shifts in the x_3 direction ($c_1=0$, $c_2=1, 2, 3$) and three shifts in both directions ($c_1=1, 2, 3$, $c_2=1, 2, 3$). We use these shifted copies along with their corresponding control inputs to drive our reservoir and produce 10 reservoir time series $r_{c_1,c_2}(t)$. Then, we concatenate these 10 time series into $x(t)$ and $r(t)$ to train output weights W according to equation (3) and perform the feedback according to equation (7), where $\mathbf{c}=[c_1;\ c_2]$ is a vector. By changing parameters c_1 and c_2, the reservoir evolves about a Lorenz-shaped manifold that is shifted in the x_1 and x_3 directions (Fig. 6).

Discussion

In this Article, we teach an RNN to interpolate, extrapolate and infer global bifurcation structures and manipulations after observing only a few local exemplars. Our approach contributes to earlier work on artificial neural networks in three important ways[27,36,37,39,42]. First, we provide a means by which a neural system performs accurate and extreme interpolations and extrapolations of modifications to its own representation far outside of its training regime. Second, we provide an analytic mechanism by which this meta-learning occurs, thereby providing design principles for effective teaching (for example, small c, closely spaced exemplars). Finally, we demonstrate that neural systems can infer global and highly nonlinear bifurcation structure using only local, pre-bifurcated example trajectories in many systems. Importantly, we use a randomly generated network that does not need to be artificially engineered to preserve invariance or manipulate information[27].

One of the main limitations of this work is the lack of a clear mechanism of how the network connectivity ultimately stabilizes the chaotic manifold. Much progress has been made in tackling this limitation, both by exercising theoretical concepts of generalized

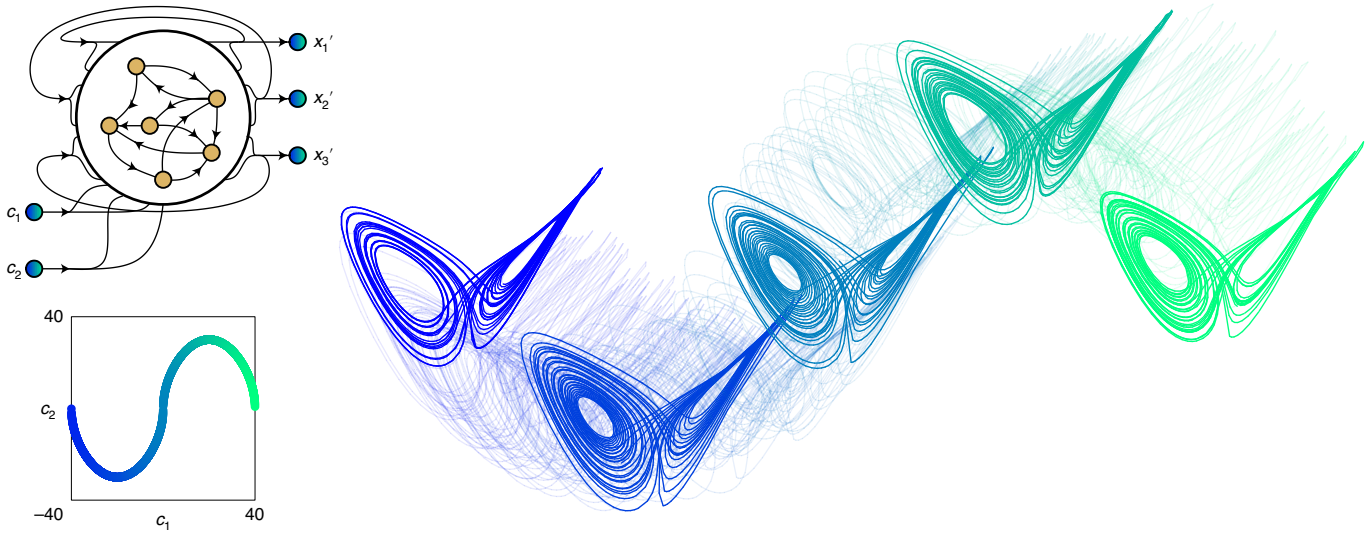


Fig. 6 | Flight of the Lorenz. A reservoir trained on translated inputs along the x_1 and x_3 directions evolves autonomously along a Lorenz-shaped chaotic manifold. We can change the x_1 and x_3 position of its representation by changing control parameters c_1 and c_2, respectively.

synchronization[43] and by developing tools for controlling chaos[44]. However, there is insufficient knowledge to guarantee that a set of training and reservoir parameters will always successfully teach the desired computation. Similarly, we are unable to specify exactly how far to space the training examples for the feedback reservoir to successfully learn the linear relationships between the differential of the reservoir states and the control parameter.

A particularly promising area for future work is related to the generalizability of the learning mechanism across activation functions, as demonstrated by the replication of our results in the more nonlinear Wilson–Cowan network model[45]. Understanding the role of various types and degrees of nonlinearity in learning computations may provide insights for the intervention and design of biological and artificial neural systems, respectively. Additionally, these results provide a basis for exploring more complex computations, such as inferring bifurcations in experimental data, and testing the reservoir's 'imagination' in reconstructing more complex chaotic manifolds using incomplete data. Finally, and perhaps most astonishing, is the reservoir's ability to accurately reconstruct the global nonlinear geometry of both the bifurcated Lorenz manifold and various bifurcation diagrams, as well as the complex changes in kinematic trajectories after only observing a narrow range of pre-bifurcated or kinematic examples. The accurate reconstruction implies that the reservoir is actually inferring higher-order nonlinear structure. This work therefore provides a starting point for exploring exactly how higher-order structure is learned by neural systems.

We would like to add a citation diversity statement following a recent proposal[46]. Recent work in several fields of science has identified a bias in citation practices such that papers from women and other minority scholars are under-cited relative to the number of such papers in the field[47–51]. Here we sought to proactively consider choosing references that reflect the diversity of the field in thought, form of contribution, gender, race, ethnicity and other factors. First, we obtained the predicted gender of the first and last author of each reference by using databases that store the probability of a first name being carried by a woman[51,52]. By this measure (and excluding self-citations to the first and last authors of our current paper), our references contain 9.48% woman(first)/woman(last), 18.78% man/woman, 11.66% woman/man and 60.08% man/man. This method is limited in that (1) names, pronouns and social media profiles used to construct the databases may not, in every case, be indicative of gender identity and (2) it cannot account for intersex, non-binary or transgender people. We look forward to future work that could help us to better understand how to support equitable practices in science.

Data availability

All data generated or analysed during this study can be found in the associated code.

Code availability

The code used to generate and analyse the data, and to produce the figures in the main text and the Supplementary Information, is available on CodeOcean at https://codeocean.com/capsule/2107188/tree/v1 (ref. [53]).

Received: 11 May 2020; Accepted: 16 February 2021;
Published online: 19 April 2021

References

1. von Neumann, J. First draft of a report on the EDVAC. *IEEE Ann. History Comput.* **15**, 27–75 (1993).
2. Alglave, J. et al. The semantics of power and ARM multiprocessor machine code. In *Proc. 4th Workshop on Declarative Aspects of Multicore Programming—DAMP '09*, 13 (ACM Press, 2008).
3. Zhang, Z., Jiao, Y.-Y. & Sun, Q.-Q. Developmental maturation of excitation and inhibition balance in principal neurons across four layers of somatosensory cortex. *Neuroscience* **174**, 10–25 (2011).
4. Faulkner, R. L. et al. Development of hippocampal mossy fiber synaptic outputs by new neurons in the adult brain. *Proc. Natl Acad. Sci. USA* **105**, 14157–14162 (2008).
5. Dunn, F. A. & Wong, R. O. L. Diverse strategies engaged in establishing stereotypic wiring patterns among neurons sharing a common input at the visual system's first synapse. *J. Neurosci.* **32**, 10306–10317 (2012).
6. Craik, F. I. & Bialystok, E. Cognition through the lifespan: mechanisms of change. *Trends Cogn. Sci.* **10**, 131–138 (2006).
7. Tacchetti, A., Isik, L. & Poggio, T. A. Invariant recognition shapes neural representations of visual input. *Annu. Rev. Vis. Sci.* **4**, 403–422 (2018).
8. Moser, E. I., Kropff, E. & Moser, M.-B. Place cells, grid cells and the brain's spatial representation system. *Annu. Rev. Neurosci.* **31**, 69–89 (2008).
9. Ifft, P. J., Shokur, S., Li, Z., Lebedev, M. A. & Nicolelis, M. A. L. A brain–machine interface enables bimanual arm movements in monkeys. *Sci. Transl. Med.* **5**, 210ra154 (2013).
10. Sainath, T. N. et al. Deep convolutional neural networks for large-scale speech tasks. *Neural Netw.* **64**, 39–48 (2015).
11. Jarrell, T. A. et al. The connectome of a decision-making neural network. *Science* **337**, 437–444 (2012).

12. Lee, J. & Tashev, I. High-level feature representation using recurrent neural network for speech emotion recognition. In *Proc. Annual Conference of the International Speech Communication Association, INTERSPEECH*, Vol. 2015, January 1537–1540 (ISCA, 2015).
13. Wang, J., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible timing by temporal scaling of cortical responses. *Nat. Neurosci.* **21**, 102–110 (2018).
14. Weber, M., Maia, P. D. & Kutz, J. N. Estimating memory deterioration rates following neurodegeneration and traumatic brain injuries in a Hopfield network model. *Front. Neurosci* **11**, 623 (2017).
15. Burak, Y. & Fiete, I. R. Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput. Biol.* **5**, e1000291 (2009).
16. Yoon, K. et al. Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nat. Neurosci.* **16**, 1077–1084 (2013).
17. Kumar, S., Dasgupta, I., Cohen, J. D., Daw, N. D. & Griffiths, T. L. Meta-learning of compositional task distributions in humans and machines. Preprint at https://arxiv.org/pdf/2010.02317.pdf (2020).
18. Schweighofer, N. & Doya, K. Meta-learning in reinforcement learning. *Neural Netw.* **16**, 5–9 (2003).
19. Santiago, R. A. Context discerning multifunction networks: reformulating fixed weight neural networks. In *2004 IEEE International Joint Conference on Neural Networks* Vol. 1, 189–194 (IEEE, 2004).
20. Feldkamp, L. A., Puskorius, G. & Moore, P. Adaptive behavior from fixed weight networks. *Inform. Sci.* **98**, 217–235 (1997).
21. Hegarty, M. Mechanical reasoning by mental simulation. *Trends Cogn. Sci.* **8**, 280–285 (2004).
22. Kubricht, J. R., Holyoak, K. J. & Lu, H. Intuitive physics: current research and controversies. *Trends Cogn. Sci.* **21**, 749–759 (2017).
23. Pfeiffer, B. E. & Foster, D. J. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature* **497**, 74–79 (2013).
24. Gold, J. I. & Shadlen, M. N. The neural basis of decision making. *Annu. Rev. Neurosci.* **30**, 535–574 (2007).
25. Strogatz, S. H. *Nonlinear Dynamics and Chaos* 1st edn (Perseus Books, 1994).
26. Yang, J., Wang, L., Wang, Y. & Guo, T. A novel memristive Hopfield neural network with application in associative memory. *Neurocomputing* **227**, 142–148 (2017).
27. Wu, S., Wong, K. Y. M., Fung, C. C. A., Mi, Y. & Zhang, W. Continuous attractor neural networks: candidate of a canonical model for neural information representation. *F1000Res.* **5**, 156 (2016).
28. Eliasmith, C. & Anderson, C. H. *Neural Engineering*: *Computation*, *Representation and Dynamics in Neurobiological Systems* (MIT Press, 2003).
29. Bekolay, T. et al. Nengo: a Python tool for building large-scale functional brain models. *Front. Neuroinf.* **7**, 48 (2014).
30. Graves, A. et al. Hybrid computing using a neural network with dynamic external memory. *Nature* **538**, 471–476 (2016).
31. Carroll, J. M. Letter knowledge precipitates phoneme segmentation, but not phoneme invariance. *J. Res. Reading* **27**, 212–225 (2004).
32. Fee, M. S. & Scharff, C. The songbird as a model for the generation and learning of complex sequential behaviors. *ILAR J.* **51**, 362–377 (2010).
33. Donnay, G. F., Rankin, S. K., Lopez-Gonzalez, M., Jiradejvong, P. & Limb, C. J. Neural substrates of interactive musical improvisation: an FMRI study of trading fours in jazz. *PLoS ONE* **9**, e88665 (2014).
34. Qiao, J., Li, F., Han, H. & Li, W. Growing echo-state network with multiple subreservoirs. *IEEE Trans. Neural Netw. Learn. Syst.* **28**, 391–404 (2017).
35. Lorenz, E. N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963).
36. Jaeger, H. The echo state approach to analysing and training recurrent neural networks—with an erratum note. *GMD Rep.* **1**, 1–47 (2010).
37. Sussillo, D. & Abbott, L. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**, 544–557 (2009).
38. Tyukin, I. Y., Prokhorov, D. & Van Leeuwen, C. Adaptive classification of temporal signals in fixed-weight recurrent neural networks: an existence proof. *Neural Comput.* **20**, 2564–2596 (2008).
39. Klos, C., Kossio, Y. F. K., Goedeke, S., Gilra, A. & Memmesheimer, R.-M. Dynamical learning of dynamics. *Phys. Rev. Lett.* **125**, 088103 (2020).
40. Słowik, R. Inverses and determinants of Toeplitz–Hessenberg matrices. *Taiwan. J. Math.* **22**, 901–908 (2018).
41. Nansai, S., Elara, M. R. & Iwase, M. Dynamic analysis and modeling of Jansen mechanism. *Proc. Eng.* **64**, 1562–1571 (2013).
42. Seung, H. S. Learning continuous attractors in recurrent networks. In *Advances in Neural Information Processing Systems* 654–660 (MIT Press, 1998).
43. Rulkov, N. F., Sushchik, M. M., Tsimring, L. S. & Abarbanel, H. D. I. Generalized synchronization of chaos in directionally coupled chaotic systems. *Phys. Rev. E* **51**, 980–994 (1995).
44. Ott, E., Grebogi, C. & Yorke, J. A. Controlling chaos. *Phys. Rev. Lett.* **64**, 1196–1199 (1990).
45. Wilson, H. R. & Cowan, J. D. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* **12**, 1–24 (1972).
46. Zurn, P., Bassett, D. S. & Rust, N. C. The citation diversity statement: a practice of transparency, a way of life. *Trends Cogn. Sci.* **24**, 669–672 (2020).
47. Dworkin, J. D. et al. The extent and drivers of gender imbalance in neuroscience reference lists. *Nat. Neurosci.* **23**, 918–926 (2020).
48. Maliniak, D., Powers, R. & Walter, B. F. The gender citation gap in international relations. *Int. Organ.* **67**, 889–922 (2013).
49. Caplar, N., Tacchella, S. & Birrer, S. Quantitative evaluation of gender bias in astronomical publications from citation counts. *Nat. Astron.* **1**, 0141 (2017).
50. Mitchell, S. M., Lange, S. & Brus, H. Gendered citation patterns in international relations journals. *Int. Studies Perspect.* **14**, 485–492 (2013).
51. Dion, M. L., Sumner, J. L. & Mitchell, S. M. Gendered citation patterns across political science and social science methodology fields. *Political Anal.* **26**, 312–327 (2018).
52. Zhou, D. et al. Gender diversity statement and code notebook v1.0 (2020); https://doi.org/10.5281/zenodo.3672110
53. Kim, J. Z., Lu, Z., Nozari, E., Pappas, G. J. & Bassett, D. S. Teaching recurrent neural networks to infer global temporal structure from local examples (2020); https://codeocean.com/capsule/2107188/tree/v1
54. Langer, G. & Parlitz, U. Modeling parameter dependence from time series. *Phys. Rev. E* **70**, 056217 (2004).

Acknowledgements

We are thankful for the insightful feedback and comments from H. Ju and K. Wiley. We 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 (grant no. W911NF-10-2-0022), the Army Research Office (grants nos. Bassett-W911NF-14-1-0679, Grafton-W911NF-16-1-0474 and DCIST-W911NF-17-2-0181), the Office of Naval Research (ONR), the National Institute of Mental Health (grants nos. 2-R01-DC-009209-11, R01-MH112847, R01-MH107235 and R21-M MH-106799), the National Institute of Child Health and Human Development (grant no. 1R01HD086888-01), the National Institute of Neurological Disorders and Stroke (grant no. R01 NS099348) and the National Science Foundation (NSF; grants nos. DGE-1321851, 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.

Author contributions

J.Z.K. conceived the initial idea and developed the analyses in conversation with Z.L., E.N., G.J.P. and D.S.B. J.Z.K. and D.S.B. prepared the manuscript with feedback from Z.L., E.N. and G.J.P. All authors contributed to discussions and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-021-00321-2.

Correspondence and requests for materials should be addressed to D.S.B.

Peer review information *Nature Machine Intelligence* thanks Wilten Nicola and the other, anonymous, reviewer(s), for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2021