nature reviews neuroscience

https://doi.org/10.1038/s41583-023-00740-7

Perspective

Check for updates

Reconstructing computational system dynamics from neural data with recurrent neural networks

Daniel Durstewitz [1,2,3] ✉, Georgia Koppe [1,4,5] & Max Ingo Thurm [1]

Abstract

Computational models in neuroscience usually take the form of systems of differential equations. The behaviour of such systems is the subject of dynamical systems theory. Dynamical systems theory provides a powerful mathematical toolbox for analysing neurobiological processes and has been a mainstay of computational neuroscience for decades. Recently, recurrent neural networks (RNNs) have become a popular machine learning tool for studying the non-linear dynamics of neural and behavioural processes by emulating an underlying system of differential equations. RNNs have been routinely trained on similar behavioural tasks to those used for animal subjects to generate hypotheses about the underlying computational mechanisms. By contrast, RNNs can also be trained on the measured physiological and behavioural data, thereby directly inheriting their temporal and geometrical properties. In this way they become a formal surrogate for the experimentally probed system that can be further analysed, perturbed and simulated. This powerful approach is called dynamical system reconstruction. In this Perspective, we focus on recent trends in artificial intelligence and machine learning in this exciting and rapidly expanding field, which may be less well known in neuroscience. We discuss formal prerequisites, different model architectures and training approaches for RNN-based dynamical system reconstructions, ways to evaluate and validate model performance, how to interpret trained models in a neuroscience context, and current challenges.

Sections

[1]Dept. of Theoretical Neuroscience, Central Institute of Mental Health, Medical Faculty Mannheim, Heidelberg University, Mannheim, Germany. [2]Interdisciplinary Center for Scientific Computing, Heidelberg University, Heidelberg, Germany. [3]Faculty of Physics and Astronomy, Heidelberg University, Heidelberg, Germany. [4]Dept. of Psychiatry and Psychotherapy, Central Institute of Mental Health, Medical Faculty Mannheim, Heidelberg University, Mannheim, Germany. [5]Hector Institute for Artificial Intelligence in Psychiatry, Central Institute of Mental Health, Medical Faculty Mannheim, Heidelberg University, Mannheim, Germany. ✉e-mail: daniel.durstewitz@zi-mannheim.de

Introduction

How are cognitive functions implemented in the brain? A long-standing tenet in theoretical neuroscience holds that computations in the nervous system can be described and understood in terms of the underlying non-linear system dynamics[1–18]. Viewing neural computations through the lens of dynamical systems theory (DST) is particularly powerful because on the one hand, many – if not most – physical and biological processes (such as weather dynamics or action potential propagation) are naturally formalized in terms of differential or difference equations (constituting dynamical systems (DSs)). On the other hand, DSs are computationally universal in the sense that they can mimic the operations of any computer algorithm (they are 'Turing complete')[19–21]. Hence, DST offers a mathematical language for understanding biochemical and physiological processes in the brain in their own right[14,22–27], as well as for understanding information processing and computation. It, thereby, provides a promising approach to tackle long-standing questions in neuroscience, connecting different levels of nervous system description by explaining the ways in which biochemical and biophysical mechanisms give rise to network dynamics and how network dynamics, in turn, implement computational and cognitive operations.

Although the value of DST for explaining physiological and computational processes in the nervous system has been appreciated for a long time, until the past 5–10 years, it was difficult to assess DS properties from neural time series recordings directly. However, the prospects for applying DST in neuroscience have changed dramatically with the advance of massively parallel neural recording techniques on the one hand[28–32] and powerful machine learning (ML) and artificial intelligence (AI) algorithms on the other[33–42]. These advances now enable researchers to infer DS models, the governing equations that underlie experimental data, directly from time series recordings of neural populations using an approach called DS reconstruction.

In this Perspective, we discuss DS reconstruction and its potential for revolutionizing neuroscience, and highlight recent developments and concepts in this emerging AI technology that are yet to be embraced by the neuroscience community. Recurrent neural networks (RNNs) as DS reconstruction tools will be our focus, but we will also survey other recent ML–AI approaches. First, we provide important background information about DST that is needed to understand the virtues of DS models as conceptual frameworks for understanding neural computation and review increasingly routine applications of DST and RNNs in neuroscience. Next, we describe some of the formal requirements that need to be met for a DS model to constitute an accurate representation of the underlying dynamics derived directly from neural time series data. We then explore RNN training algorithms for achieving these goals and discuss different network architectures available for DS reconstruction. We move on to the evaluation and validation of DS reconstructions to understand when, and under which conditions, a DS reconstruction is considered successful and permits the RNN to be used as a formal surrogate for the empirically observed system. We finish by describing the subsequent analysis and biological interpretation of a trained RNN that reconstructs a DS of interest and point to some open challenges.

Dynamical systems theory primer

State spaces

DST offers a general mathematical language that can be applied to any system that evolves across time and space and can be described by sets of differential (in continuous time) or recursive (in discrete time) equations, which provide the mathematical representation of the system under study. DST helps us to explain and understand some generic properties of natural systems, under which conditions these phenomena occur, and how they are modulated, created or destroyed (such as convergence to equilibrium states, switching among different stable states, chaotic behaviour or oscillations and synchrony)[43–45]. DST concepts often have a natural geometrical and topological representation that make them intuitively accessible[43,45,46]. At the heart of DST is the idea of a state or phase space, the space spanned by all the dynamical variables of the system (Fig. 1). For example, in a simple two-variable single-neuron model, a point in state space specifies the current voltage and the magnitude of a refractory (hyperpolarizing) variable (Fig. 1a). In a simple neural population model, each point in state space might correspond to precisely one pair of values for the instantaneous firing rates of an excitatory and an inhibitory neuron population (Fig. 1b).

Theoretically, a state space needs to be complete and unique to formally constitute a DS, in the sense that any point in state space contains all information there is about the current state of a system and its future evolution[44]. The set of governing differential or recursive equations that mathematically describes ('models') the system under study gives the precise rules according to which it evolves in state space. These precise dynamical rules of the time evolution of the system, constituting its flow, are geometrically given by its vector field (Fig. 1), which prescribes the direction in which the state of the system will move at any point in its state space. When started from any such initial condition, the system will move through state space in accordance with its vector field, giving rise to a specific trajectory or orbit (Fig. 1). Geometrically, each trajectory reflects the joint temporal evolution of the variables of the system (for example, spiking rates of different neurons). Formally, a trajectory corresponds to the unique time solution of the set of differential equations given a specific initial condition[44]. The beauty of a state space representation and its vector field is that it yields a compact and complete description of the behaviour of a DS. Furthermore, the topological and geometrical properties of state space will determine the computations performed by the system[10,13,17]. Attractors, limit cycles, chaos and bifurcations, the most important geometrical and topological concepts characterizing state spaces, are covered next.

Attractors

The state space of a DS is filled with geometrical objects that govern the fate of trajectories, and attractors are the most important class of such objects. For example, the vector field of the neural population model commands convergence of trajectories either to a specific point in the lower left (when the initial condition is to the left of the grey line) or to a specific point in the top right (when the initial condition is to the right of the grey line) (Fig. 1b). These points are called stable fixed points (or stable equilibrium points in continuous-time systems), and the neighbourhood from which there is convergence (for example, left from the grey line for the lower left equilibrium point) is their basin of attraction (see Supplementary Box 1 for a formal definition of attractors and basins of attraction). Such point attractors are the simplest of all attractor objects because the limit set to which the trajectories converge consists only of a single point. By contrast, unstable fixed points from which the state of the system diverges along one or more directions exist as well, as in the centre of the state space of the neural population model (called a saddle node in this example) (Fig. 1b).

A simple example of a point attractor in the nervous system is the resting potential of a single neuron recorded in isolation: any (sufficiently small) positive or negative deflection (perturbation) of

the membrane potential by a transient current injection will decay back (converge) to the stable equilibrium state of the membrane potential. Persistent neural activity recorded during simple working memory tasks is a more complex example that has been interpreted in terms of point attractors in neural firing rates[47–50] (Fig. 1b). Furthermore, for the same system, there may be multiple attractor states existing simultaneously, each with its own basin of attraction (such as for the neural population model; Fig. 1b). This phenomenon is called multistability and is of huge functional relevance in computational neuroscience. For instance, multiple attractors might encode different perceptual items to be held active in working memory or might correspond to different choice options in a decision-making task[7,51,52]. Rather than the limit set to which the trajectories converge consisting of a single point, stable equilibrium points can also form a continuous line, plane, torus or any other type of manifold, geometrical objects denoted a line, ring, plane, toroid or – more generally – manifold attractor[46,53–55] (Fig. 1c). For example, line attractors have been hypothesized to provide online representations of single continuously valued variables as in 'parametric' working memory[11], in maintaining arbitrary eye positions[55], in providing contextual information during decision-making[12] and in perceiving or producing temporal intervals[5,56], whereas ring, plane or toroid attractors support similar functions in spatial navigation (head direction cells[57] and grid cells[53], for instance).

Limit cycles and chaos

Attractors might not only be single points, as in stable equilibrium points, but also come in the form of closed orbits, called limit cycle attractors (Fig. 1a,d). Stable limit cycles correspond to a non-linear oscillation in a system: the state of a system continuously cycles along the closed orbit once on it and is attracted towards this orbit from some neighbourhood, its basin of attraction. Limit cycles, like equilibrium points, can also be unstable or half-stable, such that at least along one direction the state of the system would diverge from the cycle. Examples of limit cycle attractors in the nervous system abound: spiking patterns observed in single neurons constitute stable limit cycles[10], whether rather simple with a single period as in regular spiking neurons (Fig. 1d, right) or more complex, multi-period, as in bursting neurons (Fig. 1d, left). Many stereotypical motor or locomotion patterns, such as those produced by central pattern generators[58,59], might correspond to limit cycle attractors as well. Limit cycles have also been suggested to underlie the rotational dynamics observed in motor cortex of non-human primates[60–62].

However, stable activity patterns in the nervous system do not need to be regular at all, as in a limit cycle along which the state of the system periodically returns to all its previous positions; they could also be highly irregular, as in chaotic attractors (Fig. 1d, centre). Chaotic attractors are still attractors in the sense that they correspond to a bounded, finite region in state space surrounded by a basin from which nearby trajectories are attracted. But orbits on the chaotic attractor never precisely close up as in a limit cycle; they never precisely repeat, thereby offering rich temporal structure yet retaining some degree of predictability[24]. Evidence for chaotic attractors in neural activity has been obtained both at the single neuron[24] and the network level[63,64], but their potential computational role is less well understood than those of point or limit cycle attractors.

Bifurcations

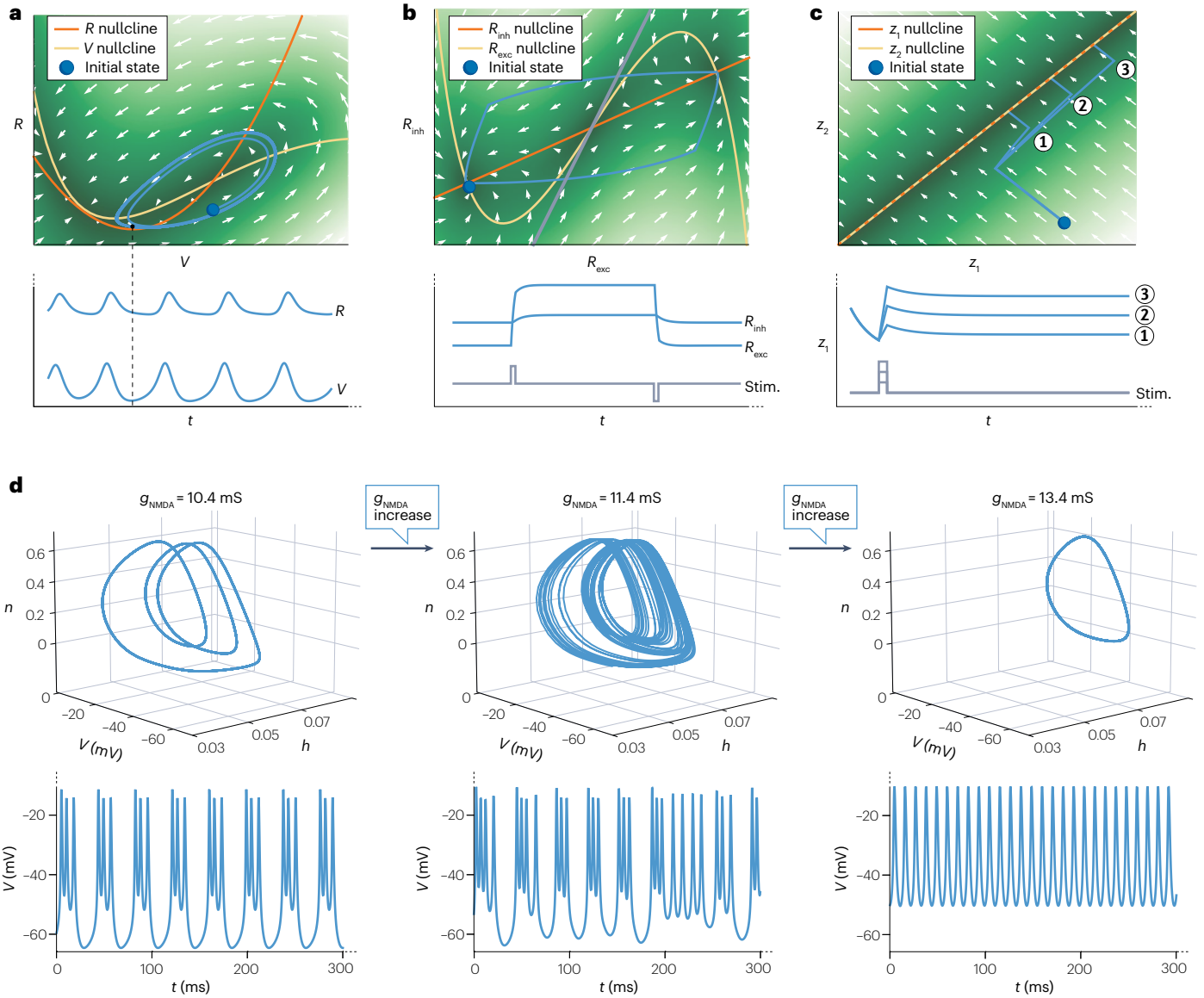
Another fundamental concept in DST that has huge functional implications for understanding physiology and computation is that of a bifurcation: bifurcations denote points (or curves) in parameter space at which qualitative (topological) changes in the system dynamics ensue as its parameters are smoothly varied (Fig. 1d). A parameter, in contrast to the dynamical variables of a system, is a comparatively stable characteristic of the system assumed to be constant from the perspective of the dynamical variables (such as maximum conductances or reversal potentials in Hodgkin–Huxley-type biophysical neuron models). At a bifurcation point, previously stable geometrical objects, such as point or chaotic attractors, may suddenly lose their stability, novel geometrical objects may come into existence or existing objects may vanish. A crucial point about many types of bifurcation is that they imply an abrupt change in system dynamics as a critical point is crossed. For example, a bifurcation occurs when a current injected into a single cell is gradually increased and the cell suddenly starts spiking[10,14,24]. Sudden transitions in neural population representations during rule-learning tasks[65,66] have also been interpreted as signatures of a bifurcation, possibly reflecting the animal switching to a different behavioural strategy.

State spaces, and some of their geometrical characteristics, are fairly straightforward to construct if the whole system of governing equations and, hence, the set of dynamical variables is explicitly given, as in the above examples, when a mathematical model of the whole process is available (Fig. 1). Empirically, when a DS is observed through an incomplete and noisy set of measurements, this is much less straightforward. This is exactly the topic of DS reconstruction that we will dive further into in the following sections. However, first we will give a short overview of how DST and RNNs have been used in neuroscience.

Dynamical systems theory and recurrent neural networks in neuroscience

DST has a long tradition in neuroscience as a theoretical framework for understanding neurophysiological function and computation. For instance, the pioneering work on the dynamical characterization of different spiking and bursting behaviours in single cells by Rinzel and Ermentrout[14] and Izhikevich[10] or the early use of DST for understanding large-scale network dynamics in populations of neurons and its links to computation[1,2,7,15,16,67–69]. However, this research often used 'hand-tuned' biophysical models of neural systems, and constructing such models is a laborious process. In addition, a biophysical description, as provided by many of these models, might not be required to understand the computational mechanisms of a system[17,46] (or might even hinder understanding by adding unnecessary complexity).

Zipser et al.[70,71] were among the first to realize the potential of RNNs for gaining insight into neural dynamics and computations. RNNs, originally introduced as formal abstractions of neuronal systems for modelling time-varying processes (lacking biophysical details)[72], consist of a set of 'neural' units that compute some non-linear (activation) function of a weighted sum of their inputs (or vice versa). RNNs differ from feedforward neural networks most used in ML, such as convolutional neural networks, by the presence of recurrent connections that allow for 'reverberation' of activity within the network. This feature also makes RNNs DSs themselves, which is important for mimicking other empirically observed DSs. One advantage of RNNs is that they can be trained using a training algorithm to perform a given task or to approximate a set of observed data[70,73,74]. For example, by training RNNs to perform a working memory task, Zipser et al.[70,71] discovered that the trained RNNs would form attractor states, with transients towards these states producing unit activation profiles resembling those found in primate electrophysiological recordings[48,49]. This type of approach of using task-trained RNNs and comparing their performance



to experimental data has experienced a recent renaissance, mainly triggered by the hugely influential work of Sussillo and collaborators on this methodological framework[12,75]. By training RNNs on similar cognitive or perceptual tasks to those used in animal experiments, hypotheses about the neural computations that underlie behavioural performance can be generated and compared[3,12,56,76–85]. This has led to many ingenious insights regarding the potential neural dynamics and computational mechanisms that facilitate multiple-task learning[85–88] and underlie cognitive flexibility and generalization[76,79,87], or how dynamical and computational mechanisms may relate to connectivity and population structure[89,90]. Although DST is often used to dissect the dynamical mechanisms of task-trained RNNs, in this approach the neural (or behavioural) data obtained from the animals themselves are not used as training data. RNNs trained on behavioural tasks from animal experiments are convenient and powerful tools for deriving computational theories, but — like biophysical models — still require post hoc matching with experimental data.

RNNs have also been trained directly on neurophysiological data[36,38–41,91–93], often within a statistical (maximum likelihood or Bayesian) framework[35,41,91,94–97]. Work by Yu et al.[91] studying neural trajectories in the macaque premotor cortex is one of the earliest examples. Much of this research using such data-inferred RNNs extended earlier work on linear or generalized linear latent state space models[98–103] — popular tools for inferring and visualizing smoothed neural trajectories in low-dimensional state spaces — by replacing linear with non-linear latent models (such as RNNs or similar formulations, for example, 'switching linear–dynamical systems'[99,104–106]). Pandarinath et al.[41,95] further developed this approach by embedding RNNs into a deep-learning (variational autoencoder[107]) framework. Besides inferring the most probable latent trajectories given the observed neural time series on a single-trial basis[38,41,92], this also enabled unobserved inputs to a target area of interest to be inferred[41]. Data-inferred RNNs have been used in many creative and versatile ways and could serve a variety of purposes. For instance, they could also be used to quantify

Fig. 1 | State spaces, vector fields and trajectories. a, Top part, the state space with vector field defined by the differential equations of a two-dimensional single-neuron model[17], which consists of a voltage (V) and a refractory (R) variable, exhibiting a limit cycle (regular spiking). The vector field (arrows; green shading indicates the magnitude of change of a state (darker denotes lower), equal to the length of the vector) gives rise to a trajectory converging to a limit cycle (sky blue line), which reflects the joint temporal evolution of the V and R variables started from an initial condition (dark blue dot). Nullclines (in red and yellow) are the sets of points in state space in which the time derivative (rate of change) for one of the specific dynamical variables becomes exactly zero ($dV/dt = 0$ and/or $dR/dt = 0$). The point at the intersection of the two nullclines is an equilibrium point, but it is unstable such that states in its neighbourhood diverge away from it and converge into the stable limit cycle. Bottom part, temporal evolution of the simulated activity of the neuron model as the system moves along the limit cycle in state space. A specific position in state space (black dot) corresponds to the reading of (V,R) at a specific time point (dashed black line). **b**, The state space with vector field of a Wilson–Cowan-type neural population model[207]. This model simulates a bistable 'working memory' system, with each point in state space corresponding to precisely one pair of values for the instantaneous firing rates of an excitatory (R_{exc}) and an inhibitory (R_{inh}) neuron population. From an initial condition (dark blue dot), an external stimulus switches the system back and forth (sky blue line) between its two stable point attractors (the equilibrium points given by the intersections of the nullclines in the lower left and upper right of the state space), with their basins of attraction (set of initial conditions from which there is convergence to one or the other point attractor) demarcated by the grey line. The equilibrium point at the intersection between the nullclines in the centre is unstable, as apparent from the vector field diverging from it along the horizontal direction. **c**, A two-dimensional linear neural ordinary differential equation system[143] forming a line attractor because of the precise overlap of the two nullclines of the system ($\dot{z}_1 = 0$ and $\dot{z}_2 = 0$), giving rise to a line of stable fixed points in state space. Starting from the same initial condition (dark blue dot), different stimulus strengths drive the system towards different states from which the trajectory will converge towards a unique point on the line, thereby encoding a graded memory of the driving stimulus. **d**, Bifurcations and chaos in a three-variable minimal biophysical NMDA-modulated bursting neuron model (with V a voltage and n, h channel gating variables) [246]: state spaces (top part) and time graphs of the voltage variable (bottom part) from the model with three increasing levels of NMDA input (g_{NMDA}), leading to transitions from bursting (left part) to chaotic (centre) and to regular spiking (right part). These transitions as the NMDA conductance parameter of the model is varied correspond to bifurcations associated with qualitative changes in the attractor of the system, such as the topological change from a chaotic attractor (centre) to a stable limit cycle (right). Similar transitions have been observed in rat prefrontal neurons in vitro driven by different levels of ambient NMDA[24]. Details on model equations and parameters used for generating the vector fields and simulated data visualized in each graph can be found in the Supplementary Methods.

(non-linear) interactions and information flow among brain areas or to analyse the relation between neural and behavioural variables[40].

Many of these data-inferred RNN models are considered generative models, in the sense that new data with similar statistical properties to those in the observed real data could be sampled from them, which may be sufficient for many applications. However, by itself, that does not imply that data-inferred RNNs are also generative in a DS sense, in that the trained RNN is an 'executable' model that, when simulated, will exhibit long-term behaviour or behaviour anywhere outside of the immediate training domain that resembles that of the underlying physiological system (Supplementary Fig. 1)[108,109]. Being generative in a DS sense is a stronger requirement, as any inferred model (RNN or otherwise) needs to converge into the same attractor states when run on its own and, at least locally, needs to feature the same vector field topology as the true system[34,92,110]. Whereas many RNNs in use in neuroscience might have this basic generative capability in a DS sense, often special training algorithms, optimization criteria, network architectures and — most importantly — validation tests are required to reliably ensure this stronger requirement is met. This is the subject of DS reconstruction (see also Supplementary Box 1), which builds on recent methodological advances in ML–AI. Next, we briefly review the classical DST-based approach to empirical data to highlight some of the mathematical considerations one needs to be aware of in DS reconstruction.

Reconstructing trajectories from time series data

When a mathematical model of a neural process is available, DST can be used to analyse its dynamical mechanisms and topological properties in detail[10,12,13,16,17,25]. But commonly, for complex systems such as the brain, a faithful mathematical model is not available to begin with. Thus, given only data to start with, one might ask: could state spaces and trajectories and their topological properties be inferred directly from experimental data instead? Could this be achieved even though typically only a (tiny) subset of all dynamically relevant variables or aggregated and indirect quantities such as extracellular potentials are observed?

The classical approach towards reconstructing trajectories directly from time series data[43,111] is based on the idea of a temporal delay embedding[112,113]. Assume scalar time series measurements x_t have been taken from a DS (extracellular potentials, for example). Typically, these measurements are some function $x_t = h(\boldsymbol{y}(t))$ of the underlying, unknown DS states $\boldsymbol{y}$ at time t, obtained through some recording device (h is also called the observation or measurement function). The unknown state vector $\boldsymbol{y}$, which gives rise to our observations, may refer to biophysical quantities, such as the membrane potentials of all neurons, or to more abstract quantities that exhaustively describe the underlying DS. From the measurements x_t, one then forms temporal delay vectors $\boldsymbol{x}_t = (x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-1)\tau})$ by concatenating the measured variables at different time lags, where m is the so-called embedding dimension and τ is a time-lag. The mechanism through which these delay embedding vectors are formed from time series is called a delay coordinate map.

A remarkable mathematical fact, enshrined in the delay embedding theorems[112,113], is that if m is large enough, the reconstructed trajectories in the space of the delay coordinate vectors $\boldsymbol{x}_t$ will represent the original trajectories in a 1:1 fashion in the sense that all their topological properties are preserved, yielding a reconstructed state space[43,112]. Topological here means that the constructed representation still allows for certain continuous deformations of the original state space. Think about it: even though the true DS and its dimensionality may not be known, one can, in principle, obtain a faithful representation of its trajectories, preserving the direction of flow from the true DS and potentially the attractor objects its trajectories may be converging to, from just scalar measurement probes into the system! To ensure this, the embedding dimension m needs to be large enough that trajectories in the reconstructed state space are fully disentangled (they do not intersect) and the resulting vector field is still smooth (it does not make sudden turns and jumps; a mapping with such properties is called a diffeomorphism[112]) (Supplementary Fig. 2).

There are important mathematical insights behind temporal delay embedding reconstructions: in any empirical situation, one can never

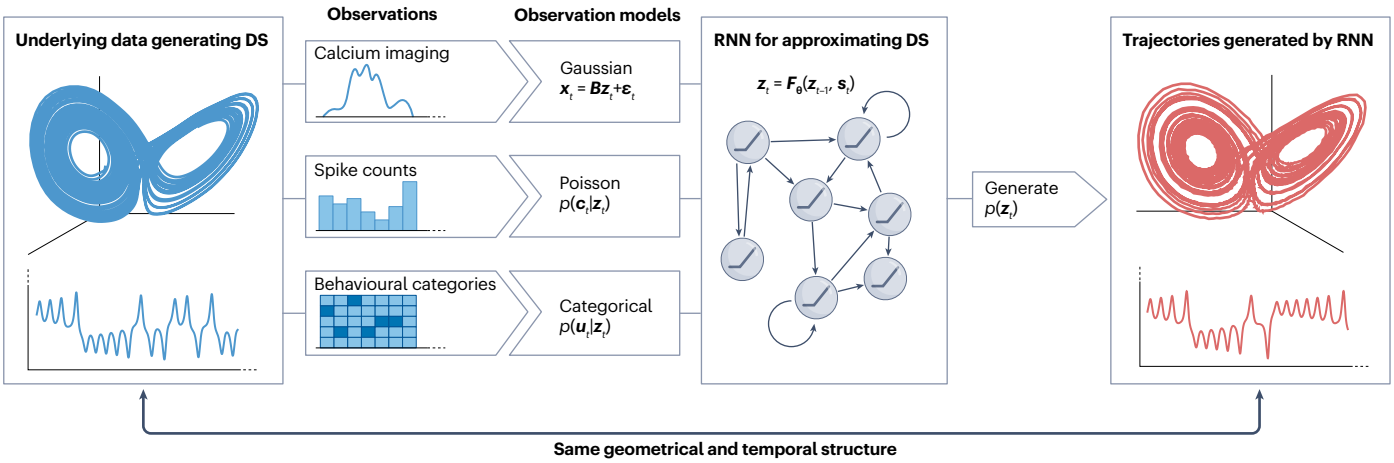


Fig. 2 | Dynamical system reconstruction via recurrent neural networks. In dynamical system (DS) reconstruction, recurrent neural networks (RNNs) (centre) are trained on time series observations from some (usually unknown) DS (left part). The unknown DS can be observed through multiple data channels with different statistical properties, which are connected to the latent RNN through different types of observation (decoder) models. These observation models capture the conditional distributions of the data ($\boldsymbol{x}_t$: Ca^{2+} imaging traces; $\boldsymbol{c}_t$: spike counts; $\boldsymbol{u}_t$: behavioural responses) given the latent states ($\boldsymbol{z}_t$) of the RNN. After successful training, the RNN is expected to generate trajectories and time series with the same geometrical and temporal structure (right part) as those produced by the underlying DS (left part) the RNN had been trained on. The centre box also gives a generic recursive equation for an RNN. This may be a 'classical' RNN with $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}, \boldsymbol{s}_t) = \varphi(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{h} + \boldsymbol{C}\boldsymbol{s}_t)$, where $\boldsymbol{\varphi}(\bullet)$ is usually a sigmoid transfer function, or it may be a piecewise-linear recurrent neural network[35,83,153] with $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}, \boldsymbol{s}_t) = \boldsymbol{A}\boldsymbol{z}_{t-1} + \boldsymbol{W}\varphi(\boldsymbol{z}_{t-1}) + \boldsymbol{h} + \boldsymbol{C}\boldsymbol{s}_t$, where $\boldsymbol{\varphi}(\bullet) = \max(\boldsymbol{0}, \bullet)$ is the so-called rectified linear unit activation function, or any other more complicated RNN architecture such as a long-short-term memory network[137]. Here, $\boldsymbol{A}$ is a trainable autoregressive matrix, matrix $\boldsymbol{W}$ contains the trainable connection weights among RNN units, $\boldsymbol{h}$ is an offset (bias) term and matrix $\boldsymbol{C}$ weighs external inputs $\boldsymbol{s}_t$ (such as sensory stimuli). Centre adapted from ref. 40. The reconstruction is illustrated here on the chaotic attractor of the famous Lorenz system[247], with details on model equations, parameters and RNN reconstruction techniques used provided in the Supplementary Methods.

be sure that all relevant dynamical variables were covered, even when recordings from hundreds of neurons are available. The delay embedding theorems give a guideline of how to augment the empirically assessed space to ensure that dynamical objects in the reconstructed state space topologically correspond to those in the original DS. Crucially, simple dimensionality reduction tools such as principal component analysis (PCA), Isomap[114] or Laplacian eigenmaps[115], as often used to represent 'neural state spaces', do not share these properties and theoretical guarantees. Dimensionality reduction tools may even destroy important dynamical features. For instance, PCA forms linear combinations of observations that may not distinguish between different states in the true state space of the DS or may break its vector field (PCA represents a linear projection operator into a lower-dimensional space that is not 1:1 and invertible, thus violating a diffeomorphism).

Delay coordinate maps only embed the particular trajectories and attractor objects traced out by the experimental measurements into a state space. Thus, temporal delay embedding reconstructions may yield only little insight into the behaviour or topology of the DS outside the immediate domain sampled by the experimental data, and they lack a mechanism to interact with the DS or to probe the rules governing its dynamics. To fully understand the computational processes carried out, access to a computational model of the DS is needed. But, as for delay coordinate maps, the computational model needs to adhere to certain topological requirements for a faithful representation of the underlying dynamics (see Supplementary Box 1). Only with such a model will it be possible to study the detailed topology and geometry of attractors and vector fields (Fig. 1).

This is the subject of DS reconstruction that is the process of inferring a mathematical model of the system dynamics directly from the experimental data (Fig. 2). In the next section, we discuss properties such a model should have, challenges in training models for DS reconstruction, and different algorithms and architectures to address these challenges.

Dynamical systems reconstruction

Universal approximation of dynamical systems

For centuries, humans used their imaginations and ingenuity to deduce the laws of nature from careful observation and experimental manipulation of the physical and biological world. But scientific model building is a laborious, long-winding and error-prone process. Can deep learning help to automatize this process? Can a mathematical model be inferred algorithmically, just from data, that behaves in every relevant aspect just like the system observed (Fig. 2)?

Deep-learning techniques being used to achieve this goal are commonly based on universal function approximators, which are essentially a set of equations that is powerful and expressive enough to approximate any other function to arbitrary precision. For example, sums of polynomial functions are known to have this property[116]. Neural networks (NNs) with at least one non-linear hidden layer also fall into this class of equations[117–119]. Thus, a library of basis functions[33,120] or a neural network[121–126] may be used to closely approximate the vector field or the trajectories of any given DS. The process by which this approximation is done is called a training algorithm in ML–AI, an iterative procedure for adjusting the parameters of the model system used for approximation such as to reduce a loss function – also known as a cost or objective function – which quantifies the deviation of any current model output from the one that optimally agrees with the data. It is important to emphasize that there are many popular ML models

that do not enjoy universal approximation properties. For instance, linear dynamical systems, such as linear state space models[100,101,127–129], are inherently incapable of producing most DS phenomena of interest, including limit cycles, multistability or chaos[44,45].

Here is another important point: the mathematical form of the equations used for approximation can be completely different from those that human observers deem to most naturally describe the true DS. An example is a 'piecewise-linear recurrent neural network' (PLRNN) (Fig. 2) that has been trained to approximate the dynamics of a biophysical spiking neuron model (Fig. 3a; see also the Video). Whereas the biophysical spiking neuron model consists of differential equations containing exponential and polynomial terms, the PLRNN itself has only piecewise-linear functions at its disposal. Thus, a detailed biological model specified by biophysical equations is not necessarily needed to perfectly mimic the dynamics of a biological neuron. This is a truly remarkable fact — in theory, a generic set of NN or basis functions can reproduce any unknown DS with its geometrical and temporal properties, which is the goal in DS reconstruction. Whether this theoretical ideal can be achieved in practice is a different question, and it depends to a considerable degree on the training algorithm and to a lesser degree on the network architecture used.

Training RNNs for DS reconstruction

In DS reconstruction, the aim is to generate models that, once trained on empirical data, will produce trajectories with topological (and ideally also geometrical) structure in state space and with long-term temporal signatures that correspond to those of the true DS. The most popular models for achieving this aim are RNNs[39,77,110,130–135] (Fig. 2). Various RNN architectures have been proposed over the past few decades, some formulated in discrete time (as time-recursive equations)[35,136,137] and some in continuous time (as systems of ordinary or partial differential equations)[83,130,138,139].

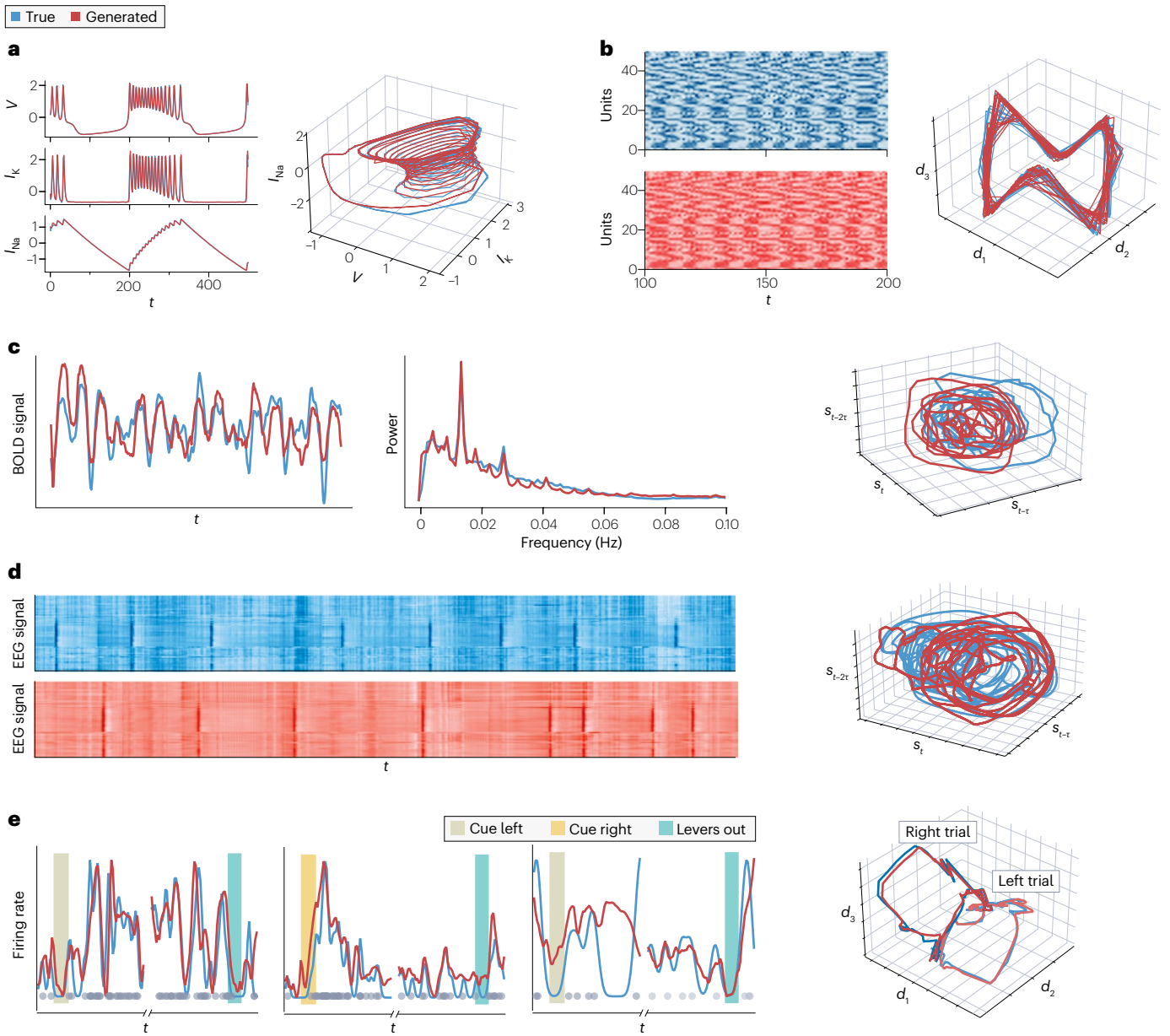
Many of these RNN architectures were motivated by the desire to solve certain practical issues in training, most prominently the 'exploding or vanishing gradient problem'[137,140,141]. Whereas in simple statistical models such as linear regression an optimal parameter solution can be calculated analytically in one step, this is no longer possible for most non-linear models such as RNNs. For non-linear network models, most commonly numerical gradient descent procedures are used, such as the famous back propagation through time (BPTT) algorithm[70,74,142]. These procedures seek an optimal solution by sliding down the gradients of the loss function, thus iteratively moving parameters towards a minimum of the loss at which the agreement of the model output with the observed data is best.

A severe problem that has limited the practical applicability of RNNs for quite some time is that these loss gradients tend to either quickly decay away or blow up during longer training sequences. This has made it difficult to train RNNs on time series that involved temporally widely separated events or processes that evolved very slowly (Fig. 1d; for example, the slow oscillation driving the bursting amidst fast spiking)[143]. Long-short-term memory (LSTM) networks were about the first architecture to address this issue by incorporating a kind of protected 'working memory' buffer within which loss gradients remain roughly constant[137] (Fig. 4a). Gated recurrent units (GRUs) simplified the LSTM structure and became a widely adopted alternative[136,144]. More recent architectures are based on coupled or independent oscillators that enable stable maintenance of information without loss divergence[145,146]. Another line of recent research aims to retain the structural simplicity of variants of classical RNNs (Fig. 2) but instead prevents unwieldly diverging loss gradients by placing specific constraints on the parameters[147–150] or by imposing 'soft constraints' through auxiliary terms in the loss function that gently push parameters into regimes that stabilize loss gradients while training[143,151,152]. ML researchers engineering such solutions usually had 'classical' ML applications such as prediction or sequence-to-sequence regression in mind, so these architectures have not necessarily made RNNs more suitable for DS reconstruction (sometimes less)[131]. For chaotic systems, exploding gradients cannot be avoided even in principle[131,153], as they are a consequence of the exponentially diverging trajectories in such systems[43]. However, chaotic dynamics are rather the rule than the exception in most complex biological or physical systems[131,154].

This leads to the following crucial insight for DS reconstruction: on the one hand, during training, the latent model (most often an RNN) needs some freedom to 'explore the future'; training on just one-step-ahead prediction errors, as implicitly done in more 'traditional' generative models, will usually fail[109,131,155–157]. On the other hand, the model cannot run unchecked for too long during training, as this leads to diverging trajectories and loss gradients. Thus, efficient training procedures for DS reconstruction build on control–theoretic methods such as modern variants of 'teacher forcing'[73,108,131,158–160], 'synchronization'[161–165] or 'multiple shooting'[166,167]. These methods are designed to keep or pull diverging trajectories back on track during training, by replacing the latent states of the model with data-inferred states at strategically chosen time points[131,166,167], optimally balancing these two[108], or forcing the model towards ('synchronizing' it with) the observed signal[161–164]. Often, keeping or pulling diverging trajectories back on track is done in ways that loosen the control through the teacher signal or observations as training progresses and the observed system is captured increasingly better, so-called annealing procedures[39,161]. Another recent strategy is to incorporate additional terms directly into the loss function to ensure that certain long-term (invariant) or geometrical properties are met (see also 'Evaluating dynamical system reconstructions')[109,157,168–170]. Besides such advances in training strategies, the development of open-source languages (such as Julia) and toolboxes (for example, DiffEqFlux[171], torchdiffeq[172] or PySINDy[173]) specifically geared towards scientific ML may have contributed to the recent surge of interest in DS reconstruction. Further impetus came from the design of network architectures construed specifically with DS in mind. The next section gives an overview of the major classes of model in use for DS reconstruction, mostly various forms of RNNs.

Categories of DS reconstruction models

By controlling gradient flows and other algorithmic tricks, RNNs such as LSTMs[134,166] (Fig. 4a) or PLRNNs[39,108,131,143] (Fig. 2) can be trained to reconstruct even complex chaotic, high-dimensional or only partially observed DSs. For example, after inferring only a suitable initial condition from the data, an RNN left to evolve freely (unconstrained by the data) according to its own governing equations can recapitulate the behaviour of the true DS (Fig. 3). To test the reconstruction capabilities of a specific ML model or training algorithm, an evaluation is usually first performed on a ground-truth DS for which the governing equations are precisely known (Fig. 3a,b). However, DS reconstruction models can also be successfully trained on experimental data such as fMRI (Fig. 3c), EEG signals (Fig. 3d) or multiple spike train data (Fig. 3e), although such experimental data may come with additional issues such as high levels of noise, non-stationarity, small sample size and potentially only a small fraction of all dynamically relevant variables being observed.



Reservoir computers and echo state machines[75,174–176] are another ingenious RNN design that is popular for DS reconstruction[132,177,178] and were originally introduced as a computationally highly efficient alternative to classical RNNs (Fig. 4b). They consist of a large pool or 'reservoir' of non-linear units with a fixed (non-trainable) network connectivity. Training proceeds solely by adapting a linear mapping from this reservoir to a layer of linear readout units that feeds back into the reservoir, entraining the network with the desired output (Fig. 4b). Because the trainable mapping is linear, learning for these systems is fast (one step) and immune to the exploding and vanishing gradient problem. However, because they rely on a fixed large reservoir, whether they really perform DS reconstruction or are just good at predicting a DS is less clear[110]. Reservoir computers and echo state machines are also rather complicated and high-dimensional and, thus, are difficult to analyse as a model of the underlying DS.

Most RNNs operate in discrete time steps, but underlying DSs are often assumed to evolve in continuous time and space. Thus, continuous-time RNNs approximate the vector field of an observed DS, estimated by taking the numerical differences along the observed time series, by a simple (one-layered) feedforward NN that is then reshaped into an RNN defined by differential equations[122,124,126]. Neural ordinary differential equations (neural ODEs[138,171,179]; also see[83,145,146,180,181] for related approaches) are essentially an extension of this idea that approximate the vector field using deeply layered feedforward NNs reshaped as RNNs (Fig. 4c). Neural ODEs are powerful tools for reconstructing observed DSs in low dimensions and naturally extend to spatially continuous systems (like dendrites), called neural partial differential equations or deep hidden physics models[130,179,181–184]. Because of their continuous time representation, neural ODEs can naturally cope with observations appearing at irregular intervals (like spike times)[38], as they do not depend on a

Fig. 3 | Dynamical system reconstruction of simulated and real physiological data by recurrent neural networks. For each dynamical system (DS) reconstruction, a recurrent neural network (RNN)[110], once trained on either simulated or physiological time series data, was freely forward-iterated in time from an initial condition inferred from the data, without any other further reference to the time series data used for training (see video for an animation). **a**, Ground truth (true) and RNN-generated trajectories as time graphs (left part) and in state space (right part) for a three-variable minimal biophysical NMDA-modulated bursting neuron model[246] in a bursting (non-chaotic) regime. **b**, Spatiotemporal patterns (left part) produced by the true (top part) and RNN-reconstructed (bottom part) systems for a neural population model[63] (a high-dimensional chaotic system). Owing to the chaotic nature of the true DS, the true and RNN-simulated patterns start to quickly diverge, yet — and importantly — retain the same temporal and geometrical structure throughout. Right part, three-dimensional state space projections, with coordinates d_1–d_3 obtained by Isomap, of the true and reconstructed trajectories. **c**, Overlaid true and RNN-generated blood oxygen level-dependent (BOLD) signal time series s_t (left part), power spectra (centre) and state space representations obtained through delay embedding of s_t (right part) from human functional MRI data[39]. Agreement in the true and RNN-generated power spectra may be quantified through the Hellinger distance ($D_H \approx 0.26$ in this example, where this value can vary between 0 and 1, with 0 indicating perfect overlap) and agreement in attractor geometries through a Kullback–Leibler divergence ($D_{stsp} \approx 0.4$ in this example, where values for bad reconstructions are usually above 3.3)[39]. **d**, DS reconstruction from human electroencephalogram (EEG) data[248], with true and RNN-generated EEG signal time series from 64 channels (left part) and state space representations obtained through delay embedding of one EEG time series s_t (right part). **e**, Multiple single-unit (MSU) data recorded from the rat anterior cingulate cortex during a delayed alternation task[249]. Left part, spike trains (grey dots) convolved with Gaussian kernels yielded an estimate of instantaneous firing rate so that true and RNN-generated firing rates could be compared for various single-unit examples, together with important task events (left and right lever presses that simultaneously serve as cues for the next trial in delayed alternation, and the response period during which both levers came out), during single trials (no averaging). There is a close agreement between RNN-generated and true MSU firing rate profiles across the different task periods for a wide variety of single-unit behaviours. Right part, a projection of true and RNN-generated MSU trajectories into a lower-dimensional representation of the state space obtained by Isomap (the true state space is much higher-dimensional and, hence, cannot be shown in full). Left versus right lever press trials are associated with distinct trajectories emerging from a common ground state. Details of the physiological time series data, model equations and parameters used for generating the simulated data and performing the RNN-based DS reconstructions (and creating each visualization) can be found in the Supplementary Methods.

discretization of time into equal bins[138]. Neural ODEs also easily allow for incorporating prior domain knowledge in the form of known differential equations, as in physics-informed neural networks[130,184]. However, as it currently stands, neural ODEs seem more tedious to train because they rely on numerical integration techniques for solving the differential equations and the loss gradients.

An elegant idea — different from RNNs — that uses a large library of basis functions to approximate an observed DS and, thereby, offers some level of symbolic interpretability (because the library consists of directly interpretable mathematical forms) is sparse identification of non-linear dynamical systems (SINDy)[33,34,185] (Fig. 4d; see refs. 120,186 for related ideas). SINDy rests on LASSO regression[187] (a form of linear regression imposing a sparsity penalty on parameters) for selecting from its large library only a small subset of functions, forcing all other regression coefficients to zero. SINDy is fast and highly accurate if the basis function library contains the right terms that most naturally describe the DS under study, for instance, if the DS equations consist of polynomial terms and the basis function library contains the right polynomial terms as well. But SINDy will often fail to converge to a solution if a suitable basis function library cannot be set up a priori, as in many empirical scenarios in which the precise DS equations are simply not known[110].

At this point, many readers may wonder about transformers[188–190], which underlie the recent success of large language models such as GPT-4[191]. However, transformers, in their original formulation[190], intentionally remove recurrence in connectivity and, thus, the temporal dimension. Therefore, unlike the other architectures above, transformers are not DS models themselves and introduce time only through the backdoor via explicitly time-dependent functions. Although their outputs may be connected back to the inputs, making them recursive, the strength of transformers is really in processing and predicting symbolic sequences, and their use for DS reconstruction has been limited so far[192,193].

Enhancing RNN capabilities with autoencoders. Often one is interested in finding the lowest-dimensional representation possible of the dynamics or suitable coordinate transformations that facilitate DS learning and interpretability. This can be achieved by embedding SINDy, or any RNN, into an autoencoder[194] architecture[34,195,196] (Fig. 4e). An autoencoder consists of a deeply layered encoder network that projects observed data into a usually much lower-dimensional latent space, configured to have certain desirable properties, from which the original data are to be recovered again through another deeply layered decoder network. By co-training such an autoencoder together with a DS reconstruction model using a combined loss function, a low-dimensional latent model most suitable for learning the underlying dynamics can be construed[34,195,196].

Probabilistic RNN formulations. Some of the currently most successful training methods for DS reconstruction (such as BPTT with variants of sparse teacher forcing)[108,131] implicitly assume the underlying latent model to be deterministic, but often it might seem more natural to assume that the underlying dynamical processes are stochastic, explicitly accounting for noise sources present in the brain, for instance[197]. Indeed, RNNs both in discrete time[35,36,38–41,92,99,105,155,198] and in continuous time[181,199–201] have been equipped with probability assumptions. Incorporating such stochasticity into latent models requires special training and inference methods that often rely on so-called state space frameworks and the expectation–maximization algorithm[35,39,40] or on the technique of variational inference[36,40,92,155] and variational autoencoders[41,107,202]. Such latent models provide a means for generating a whole probability distribution across the latent state space (and possibly across parameters)[36,38,40,41,92,110,198]. Probabilistic DS reconstruction models also account in a more natural way for a variety of statistically diverse, simultaneously recorded data modalities[40,155] (Fig. 2). For instance, neuroscience experiments might have Poisson-type spike count data from many neurons, on top of continuous (x, y) coordinates of a rodent in a maze and categorical behavioural choices. These could be integrated into the same latent DS model by connecting the RNN to different types of modality-specific decoder models that capture the unique statistical properties of the three unique data modalities observed (Fig. 2, left part). This establishes direct links between the different data modalities within the common latent space, making it

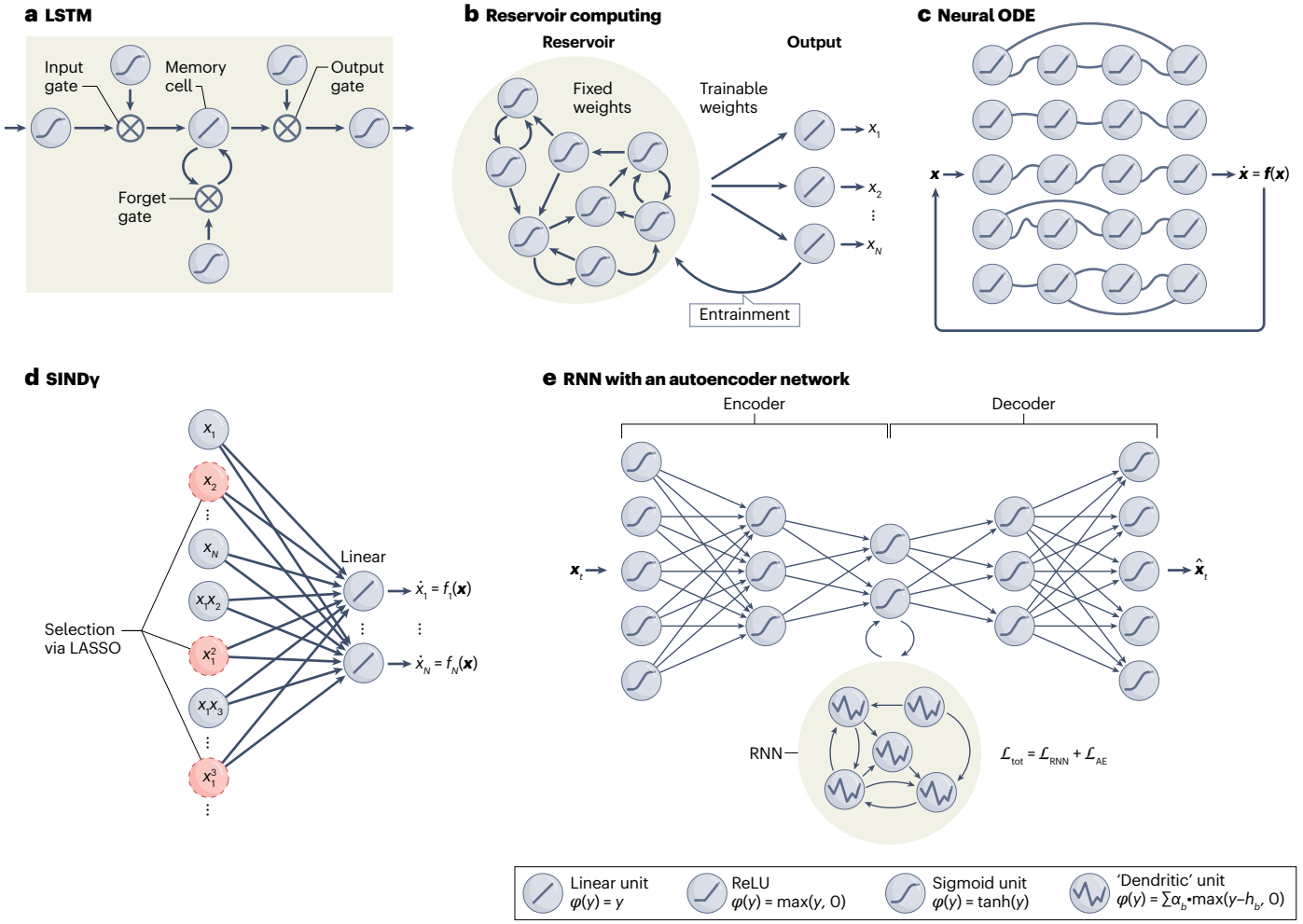


Fig. 4 | Architectures used for dynamical system reconstruction. **a**, Long-short-term memory (LSTM) cells[137] possess a gated memory buffer (memory cell in the centre) that protects contents from 'overwriting'. Stable maintenance, neither decaying nor exploding, is achieved by a linear activation function in the memory cell. Crossed circles indicate input, output and memory (multiplicative) updating gates that are controlled by trainable networks with non-linear (sigmoid-type) activation functions. **b**, Reservoir computers[175] consist of a large pool of non-linear units, but with fixed weight (non-trainable) local connectivity. Their activation states are 'read out' by a linear output layer, driven by the desired (observed) outputs during training. Only the weights that lead to these outputs are trainable. Crucially, the output layer feeds back into the reservoir, entraining the network with the desired outputs (the 'echo state property'). **c**, Neural ordinary differential equations (neural ODEs)[138], unlike many other recurrent neural networks (RNNs), are formulated in continuous time and possibly space (as differential equations). They consist of (possibly infinitely) deeply layered neural networks with skip connections (spanning more than one layer) that learn to approximate the vector field $\boldsymbol{f}(\boldsymbol{x})$ of an observed system (that is, the temporal derivatives $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$). **d**, Sparse identification of non-linear dynamical systems (SINDy)[33] attempts to approximate the vector field $\boldsymbol{f}(\boldsymbol{x})$ of an observed system by a very large library of basis functions that are linearly combined to yield the temporal derivatives as output. A selection procedure via least absolute shrinkage and selection operator (LASSO) regression removes all those terms from the library that are not required, producing a minimal representation. **e**, RNNs may be trained jointly, through a combined loss function, $\mathcal{L}_{tot}$, with an autoencoder network[194] that produces a lower-dimensional representation (encoder part) from which the original observations ($\boldsymbol{x}_t$) could be faithfully reconstructed ($\hat{\boldsymbol{x}}_t$) again (decoder part). The joint training leads the autoencoder to extract a lower-dimensional manifold from the data that optimally represents the observed DS[34]. AE, autoencoder; ReLU, rectified linear unit.

possible to reveal, for instance, the relation between neural trajectories, DS objects and behavioural choice processes[40,155].

Evaluating dynamical system reconstructions

How does one evaluate whether the DS reconstruction worked as supposed? In ML, RNNs are mostly used for ahead-predictions on the system under study, for example, predicting electricity consumption[203] or forecasting object trajectories[204]. Thus, mean squared prediction errors (MSPEs) are often used to evaluate the performance of an RNN training algorithm. However, MSPEs are not a sufficient (or even suitable) metric for the evaluation of DS reconstruction algorithms (defined here to include both the training algorithm and the chosen architecture). If the underlying DS is chaotic — owing to the exponential divergence of nearby trajectories — minuscule amounts of noise or differences in initial conditions will quickly lead to large MSPEs even if the observations were drawn from the exact same underlying DS with the same

Glossary

Activation function
The non-linear function in a neural network computed on the inputs to a unit (node) of the network.

Attractor
A subset of the state space of a DS towards which the DS evolves over time from the basin of attraction of the attractor; it can, for example, be a single point (point attractor), a closed orbit (limit cycle) or a complex, fractal geometrical structure (chaotic attractor).

Autoencoder
A type of (usually non-linear) neural network architecture used to learn a compressed (lower-dimensional) representation of the data in an unsupervised manner, consisting of an encoder that maps input data to the lower-dimensional latent representation and a decoder that reconstructs the input data from the encoded representation.

Back propagation through time
(BPTT). A gradient-based training algorithm for training RNNs; BPTT computes the gradients (partial derivatives) of the loss function between RNN-generated outputs and target values and propagates these backwards through time to update the RNN weights.

Basin of attraction
The set of initial conditions from which the trajectory of a DS will eventually converge into the attractor (in the limit $t \to \infty$).

Bifurcation
A sudden qualitative (topological) change in the state space and behaviour of a DS as one or more of its parameters cross a certain threshold, usually involving the creation or destruction of attractors.

Decoder
A component of a neural network model that maps the latent state of a model back into observation space (in other words, the space of the observed data).

Delay coordinate map
A map that embeds an observed time series into a space in which the resulting trajectory will be diffeomorphic to the true trajectory of the observed system.

Diffeomorphism
A bijective (1:1 and onto) function that maps one differentiable manifold onto another such that both the function and its inverse are continuously differentiable (implying a 1:1 relation also between gradients).

Dynamical system
A system that evolves in time (and possibly along other dimensions such as space) according to a set of rules or equations in a state space, which is the space spanned by all its dynamical variables.

Encoder
A component of a neural network model that maps input data (observations) into a latent space, in which an RNN may operate.

Equilibrium point (state)
Steady state of a DS described by differential equations, in which — when exactly placed at this point — the state of a DS would not change anymore (the same type of object is called a fixed point in a discrete-time DS).

Exploding or vanishing gradient problem
The problem that in RNNs or deep neural networks, the gradients of the loss function will eventually diverge ('explode') or vanish during the training process, if not controlled in some way.

Feedforward neural network
A neural network in which connections between nodes exclusively point in one direction, leading from input to final output.

Flow
A function that maps states of a DS to future or past states, given by the solution to the system of differential equations describing the DS.

Fractal dimensionality
The dimensionality of a geometrical object is commonly thought to be an integer number, but chaotic sets often have a self-similar geometrical structure that is more accurately captured by a non-integer (such as a transcendental real) number.

Gradient descent
A class of optimization techniques that aim to find a (local) minimum of a differentiable objective function (such as a loss function) by iteratively adjusting model parameters such that they are pushed into directions of descending slope (gradients).

Initial condition
The state in state space of a DS from which a trajectory originates (starts).

Invariant sets
Sets of states in state space of a DS, in which the state of the DS remains for all time under the action of the flow (the dynamical rules of the system).

Latent model
A statistical or ML model that contains unobserved (latent) variables that need to be inferred in order to account for the data observed.

Limit set
A set of states into which a DS converges as time goes to infinity.

Loss function
A function (also known as cost or objective function) that quantifies the mismatch between outputs predicted by a model and the target or desired outputs (it could be a negative likelihood, for instance).

Manifold
Any topological space that locally resembles Euclidean space (that is, for which there exists a continuous (bijective) function, with continuous inverse, that maps any neighbourhood of any point in that space to an open ball of Euclidean space).

Recurrent neural network
A type of neural network in which connections also recurrently couple different network units, in other words can run both forwards and backwards, unlike in feedforward neural networks.

State
A (vector) point in state space.

State (or phase) space
The space of all possible states a DS may be in, which is spanned by all dynamical variables of the DS.

Teacher forcing
A technique used in training algorithms for sequence generation and DS reconstruction tasks, in which during training (but not during model deployment), the latent states of an RNN are pushed to agree with the observations (in DS reconstruction models, specific, recently developed amendments of these techniques are used).

Temporal delay embedding
The vector space produced by the delay coordinate map.

Training algorithm
An algorithmic procedure by which the parameters of an ML model are obtained given a specified loss function and a set of training data as targets.

Glossary (continued)

Training data
The set of sampled data points used for training an ML model (part of the acquired empirical data are usually held back as validation and test sets and not used for training).

Trajectory or orbit
The sequence or continuous series of states a DS moves through, starting from some initial condition, as time progresses (for a continuous-time DS, it is formally the solution curve from a specific initial condition).

Turing complete
A system that can emulate the operations of any Turing machine, a general model of computation.

Variational autoencoder
A specific type of autoencoder in which the latent states are probabilistic (treated as random variables), such that the encoder and decoder operate on probability distributions rather than on single data points.

parameters[39,205]. Vice versa, a comparatively low MSPE might falsely indicate a good agreement between a true and a reconstructed DS, although the two systems could profoundly differ in their underlying dynamics (Supplementary Fig. 3).

Thus, in DS reconstruction, it is important to check for geometrical and other time-invariant properties of the DSs under study. For instance, the Kullback–Leibler divergence[39,110], Wasserstein distance[178] and Hellinger distance[206] have been used to assess the geometrical overlap in data point distributions across invariant sets in state space generated by the true and the reconstructed DS in the large time limit. The maximal Lyapunov exponent or the so-called correlation dimension (an empirical estimate of the fractal dimensionality of an attractor) is another example of such invariant dynamical and geometrical DS characteristics[111,157,196]. Agreement in the invariant temporal structure of true and reconstructed trajectories — that is, properties of the temporal behaviour that, in the limit, do not depend on when in time one takes measurements — may be assessed by autocovariance functions or overlap in power spectra[39,110,131,134,205]. Only if a reconstructed DS agrees well with the data on such invariant geometrical and temporal properties should it be further analysed and interpreted as a potential model of the underlying system dynamics.

Analysis and interpretation

A data-inferred RNN of a DS of interest offers two related levels of interpretability: first, inspection of the parameters of the model to infer physiological or anatomical properties of the underlying DS, such as connectivity between neurons or between brain areas (Fig. 5). Second, a formal surrogate of the data-generating dynamical process has been obtained that provides unprecedented access to the underlying computational mechanisms: DST tools can be used to unravel the inner workings of the model in detail. Whereas it is the latter level of interpretability in which this new ML–AI technology may become really transformative, the former level is important for specifically understanding how different neuronal circuit components and processes contribute to computation.

Numerous possibilities exist for endowing an RNN with direct physiological and anatomical interpretability during a DS reconstruction (Fig. 5). Commonly, a latent RNN is coupled to the actually measured neural or behavioural time series through an observation (or decoder) model (Fig. 2). If the decoder model used to link the latent RNN to the measured data takes the form of a generalized linear model, one can directly interpret the entries in the factor loading matrix $\boldsymbol{B}$ of the generalized linear model (the matrix of weights connecting observations to latent states) as in conventional[207] or Gaussian-process[129] factor analysis. The entries in $\boldsymbol{B}$ reflect the joint loading of different observables (such as recorded neurons and/or behavioural responses) on the same latent RNN states (factors). For instance, sets of units with large coefficients (strong loading) on the same latent states would be active within the same cell assembly (Fig. 5a). If measurements were taken from different data modalities such as neuronal and behavioural variables simultaneously[40,155], $\boldsymbol{B}$ would in addition identify relations between the different modalities.

An interesting option is to restrict the structure of $\boldsymbol{B}$: one could constrain $\boldsymbol{B}$ such that only subsets of the RNN latent states are allowed to map to specified subsets of observations, which could assign particular semantic roles, such as those of prefrontal neurons or pyramidal cells, to defined subsets of latent states. For example, given recordings from different brain areas or cortical layers, $\boldsymbol{B}$ could be constrained such that only a subset of the RNN latent states connect to one brain area and a different subset of RNN states connect to a second brain area (Fig. 5b). Thus, in that example, the entries in the connectivity matrix $\boldsymbol{W}$ of the RNN (the matrix of weights connecting different latent states) would automatically assume interpretations in terms of intra-areal and inter-areal connection strengths (Fig. 5b). As another example, assume that the recorded units could be sorted into different classes, such as pyramidal cells and interneurons. By assigning subsets of latent states to only one or the other class and enforcing their outgoing weights to be only positive or negative, $\boldsymbol{W}$ would yield information about excitatory and inhibitory microcircuits and connectivity motifs (Fig. 5c).

RNN reconstruction models become particularly strong when harnessed as conceptual frameworks for analysing the DS implementation of the neural computations behind animal cognition and behaviour. Different hypotheses about the neuro-computational realization of working memory have been advanced in the literature, for example, multistability[7,52] (Fig. 1b) or slow close-to-bifurcation dynamics in the absence of stable states[208,209]. Such hypotheses can now be directly tested by examining the vector fields and attractor objects of trained RNN-based reconstruction models (in ref. 208, this question is tackled with a related approach using RNNs). Furthermore, dynamical mechanisms not previously thought of might be discovered.

However, the success of analysing the DS implementation hinges on the dynamical accessibility and interpretability of the ML–AI model. If the mathematical set-up of the RNN used is itself rather complex, as in LSTMs or neural ODEs, reverting to approximate numerical methods is needed to find dynamical objects and structure of interest[199,210]. Thus, many ideas for rendering ML–AI models such as RNNs interpretable in a DS sense rest on some form of locally linear dynamics[36,39,40,99,105,110,143,156,211–213], as linear models are analytically tractable, well understood and easy to analyse. However, globally, a proper DS model still needs to be non-linear; otherwise, it could not produce phenomena such as limit cycles or chaos as described in the primer on DST earlier in this article. Interpretability is further enhanced by discovering low-dimensional DS representations, for example, through boosting the expressivity of single network units[108,110] or co-training with autoencoders to extract low-dimensional DS manifolds[34,214]. Equally important

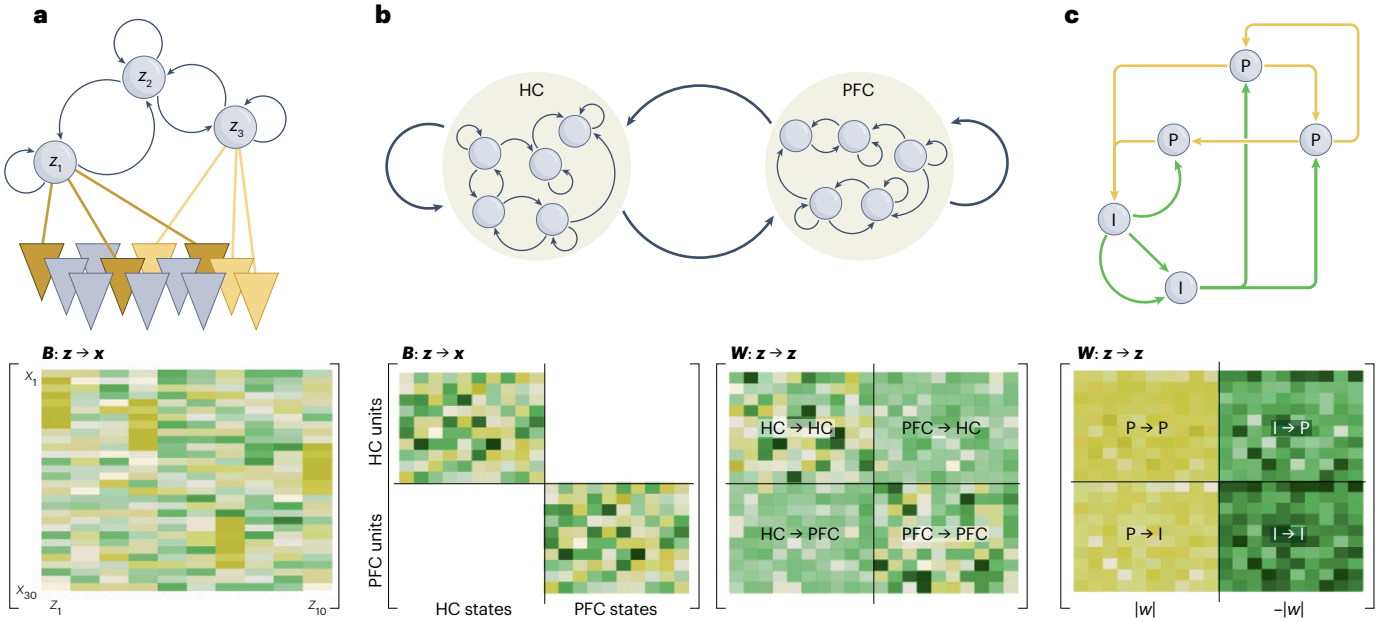


Fig. 5 | Interpreting the relationship of a data-inferred recurrent neural network to the biological substrate. The structure in the observation matrix $\boldsymbol{B}$, linking latent states to observations, and the connectivity matrix $\boldsymbol{W}$, specifying the connection weights among RNN latent states, can be interpreted or constrained such as to link dynamics to different biological underpinnings. **a**, Different recorded units x loading on the same latent state z may be interpreted as part of the same assembly. In this example, the sets of units appearing in darker yellow within one column of $\boldsymbol{B}$ would be active within the same cell assembly (with hue indicating association strength). **b**, By constraining the observation matrix $\boldsymbol{B}$ such that subsets of RNN latent states z can couple to recorded neurons x from one specific brain area only – hippocampus (HC) or prefrontal cortex (PFC) in this example – the connectivity matrix $\boldsymbol{W}$ acquires biological meaning in terms of within-area versus between-area connections. This can be used to examine inter-areal information transfer (HC to PFC or PFC to HC in this example) in a time-resolved manner. Yellow versus green colours indicate excitatory versus inhibitory weights, with hue indicating weight magnitude. **c**, Likewise, $\boldsymbol{B}$ may be structured such that different latent states z couple only to specific neuron types – for example, pyramidal neurons (P) versus interneurons (I) – and their weights can easily be constrained by taking the absolute value, $|w|$, to be only excitatory (positive) or inhibitory (negative). This could dissect the role of different neuron types in the dynamics. Colour coding has the same interpretation as in part **b**.

are clever analysis tools for dissecting and relating RNN structure and dynamics to computation and task performance, lines along which computational neuroscience has made tremendous progress in recent years[12,56,76,77,79,80,86–90,128,208,215]. Thus, in the field of DS reconstruction, a particular challenge is to design simple and mathematically tractable, yet expressive, model architectures.

Outlook and future challenges

The field of model-based DS reconstruction is still in its infancy. So far, little is known about the empirical and theoretical conditions under which training algorithms will yield topologically and geometrically faithful reconstructions of the underlying DSs. Until recently[40,110,131,178,195,216], many training algorithms for DS reconstruction have been tested mainly on rather small (less than four-dimensional) benchmark systems, with no or little process and observation noise, and assuming full access to all system variables, large sample sizes and stationary conditions. This is in stark contrast to the neuroscientific reality.

First, neural systems are extremely high-dimensional. Modern recording techniques now routinely provide hundreds (multiple single-unit recordings) to thousands (Ca^{2+} imaging or fMRI) of simultaneous time series observations[28,29,31,32]. But even this remains a minuscule fraction of all the dynamical variables in the biological substrate, for example, the billions of neurons in a rodent brain alone, not to mention all the cellular and molecular processes. How can one be sure that all dynamically relevant variables have been observed? Does the – already high-dimensional – observation space need augmenting even further, for instance through delay embedding? It has been speculated that behaviourally relevant neural dynamics may be confined to much lower-dimensional manifolds[53,82,198,217–221], so co-training DS models with autoencoders for extracting these manifolds could be of help[34,214]. However, such low-dimensional representations may not always be preserved across time, task contexts and brain areas. For instance, in prefrontal areas, neural representations appear to be rather fleeting, with changing intercellular alliances[222–224], which is potentially incompatible with a low-dimensional structure[225].

Somewhat relatedly, neural processes are inherently stochastic (for example, owing to synaptic failures[197]), and the observation process infuses additional noise (such as spike-sorting errors). Moreover, neural observation techniques typically represent only lump signals requiring post-processing (extracellular electrodes), often provide filtered versions of the variables of interest (Ca^{2+} imaging) or produce variables that may be highly non-Gaussian (membrane voltage) or even non-continuous (spike counts). How much detailed information about the underlying dynamics can, even in principle, be retrieved from such types of observations[40,155,226,227] is yet unclear. A related question is by how much different types of data preprocessing (for example, various filtering operations or kernel density smoothing of spike trains) diminish or enhance the ability to reconstruct the true underlying DS.

Another fundamental challenge to many DS reconstruction algorithms is that neuroscience data are often highly non-stationary, involving systematic trends and drifts[222,224,228,229], slow changes in bodily, motivational or emotional states[230] or learning phenomena[65,66,231]. Slow drifts in the parameters of a neural system tend to produce numerous types of complex bifurcations (also called 'tipping points' in this context). The dynamical regime within which the cortex operates is not constant, although there are various ways to deal with such non-stationarities (for example, by treating the model parameters themselves as dynamical variables that can fluctuate across time[232,233]). In DS reconstruction, this additional layer of complexity has only recently started to receive more attention[178,234–236]. It is related to the concept of 'out-of-distribution'[235,236] generalization in ML (in contrast to 'mere' out-of-sample prediction[237]): if the data-inferred DS model captures the true governing equations (is 'correct' in this sense), it should be able to generalize beyond the data domain (parameter regime or basin of attraction) seen in training (for example, predicting transitions into epileptic activity even if trained only on healthy tissue[238,239]). More work on topological theory for RNN-based DS reconstruction will be necessary to clarify under which conditions this is possible.

Non-stationarity aside, neural dynamics evolve within a hierarchy of different temporal and spatial scales[240–242]. Take fast spiking activity nested within slower oscillations (Fig. 1d) as a simple example. Which of those time scales are computationally relevant in a given context? DS reconstruction methods explicitly designed for mapping processes on multiple time scales could help[143,243], but often the choice of variables confers importance only on particular ones. For instance, biophysical and synaptic heterogeneity, among the many strong non-linearities present in the nervous system, easily give rise to highly chaotic activity[24,63,64,244], which at first glance is incompatible with simple point attractor accounts of working memory and decision-making (but see ref. 245). Point attractors are often a consequence of focusing on averaged quantities like spike rates in many RNN-based analyses, essentially averaging out those faster time scales at which chaos reigns, yet the latter may be relevant for computation as well.

More generally, DS accounts in neuroscience so far have mostly focused on very simple DS objects such as line attractors[12,55] or limit cycles[61]. More complex DST concepts will need to be engaged as different temporal and spatial scales are to be traversed and as neuroscience continues to delve into more complex behaviours and natural environments. RNNs and related models, if used for DS reconstruction, offer more than just a powerful ML–AI methodology: they conceptually integrate different scales and levels of description, from cell assemblies to behaviour, within formal theories of neural computation. Their intimate quantitative relationship with the data and the degree of analytical access they provide may one day be transformative for our understanding of brain function, perhaps comparable in impact to the advances in optogenetics.

Data availability

All data used to create the RNN reconstructions in Fig. 3 are publicly available. See Supplementary Methods for details.

Code availability

All codes used to create the RNN reconstructions in Figs. 2 and 3 are publicly available. The code for the models used in Fig. 1b,d is publicly available. See Supplementary Methods for details.

Published online: 4 October 2023

References

1. Amit, D. J. & Brunel, N. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cereb. Cortex* **7**, 237–252 (1997).
2. Brunel, N. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J. Comput. Neurosci.* **8**, 183–208 (2000).
3. Carnevale, F., de Lafuente, V., Romo, R., Barak, O. & Parga, N. Dynamic control of response criterion in premotor cortex during perceptual detection under temporal uncertainty. *Neuron* **86**, 1067–1077 (2015).
4. Deco, G. & Rolls, E. T. in *Creating Brain-Like Intelligence* (eds Sendhoff, B. et al.) 31–50 (Springer, 2009).
5. Durstewitz, D. Self-organizing neural integrator predicts interval times through climbing activity. *J. Neurosci.* **23**, 5342–5353 (2003).
6. Durstewitz, D., Huys, Q. J. M. & Koppe, G. Psychiatric illnesses as disorders of network dynamics. *Biol. Psychiatry Cogn. Neurosci. Neuroimaging* **6**, 865–876 (2021).
7. Durstewitz, D., Seamans, J. K. & Sejnowski, T. J. Neurocomputational models of working memory. *Nat. Neurosci.* **3**, 1184–1191 (2000).
8. Goel, A. & Buonomano, D. V. Timing as an intrinsic property of neural networks: evidence from in vivo and in vitro experiments. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **369**, 20120460 (2014).
9. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
10. Izhikevich, E. M. *Dynamical Systems in Neuroscience* (MIT Press, 2007).
11. Machens, C. K., Romo, R. & Brody, C. D. Flexible control of mutual inhibition: a neural model of two-interval discrimination. *Science* **307**, 1121–1124 (2005).
12. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
A milestone in RNN-based analysis of neural data, in which task-trained RNNs were used to elucidate potential dynamical mechanisms of context-dependent decision-making, involving the context-dependent integration of evidence by approximate line attractors, similar to the patterns observed in the actual experimental data.
13. Miller, P. Dynamical systems, attractors, and neural circuits. *F1000Res.* **5**, F1000 (2016).
14. Rinzel, J. & Ermentrout, G. B. in *Methods of Neuronal Modeling: From Synapses to Networks* (eds Koch, C. & Segev, I.) 251–292 (MIT Press, 1998).
15. Wang, X.-J. Synaptic basis of cortical persistent activity: the importance of NMDA receptors to working memory. *J. Neurosci.* **19**, 9587–9603 (1999).
16. Wang, X.-J. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron* **36**, 955–968 (2002).
17. Wilson, H. R. *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience* (Oxford Univ. Press, 1999).
18. Wilson, H. R. & Cowan, J. D. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* **12**, 1–24 (1972).
19. Branicky, M. S. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theor. Comput. Sci.* **138**, 67–100 (1995).
20. Koiran, P., Cosnard, M. & Garzon, M. Computability with low-dimensional dynamical systems. *Theor. Comput. Sci.* **132**, 113–128 (1994).
21. Siegelmann, H. & Sontag, E. D. On the computational power of neural nets. *J. Comput. Syst. Sci.* **50**, 132–150 (1995).
22. Bhalla, U. S. & Iyengar, R. Emergent properties of networks of biological signaling pathways. *Science* **283**, 381–387 (1999).
23. Bhalla, U. S. & Iyengar, R. Robustness of the bistable behavior of a biological signaling feedback loop. *Chaos* **11**, 221–226 (2001).
24. Durstewitz, D. & Gabriel, T. Dynamical basis of irregular spiking in NMDA-driven prefrontal cortex neurons. *Cereb. Cortex* **17**, 894–908 (2007).
25. Durstewitz, D. & Seamans, J. K. The computational role of dopamine D1 receptors in working memory. *Neural Netw.* **15**, 561–572 (2002).
26. Mackey, M. C. & Glass, L. Oscillation and chaos in physiological control systems. *Science* **197**, 287–289 (1977).
27. Sherman, A. Dynamical systems theory in physiology. *J. Gen. Physiol.* **138**, 13–19 (2011).
28. Machado, T. A., Kauvar, I. V. & Deisseroth, K. Multiregion neuronal activity: the forest and the trees. *Nat. Rev. Neurosci.* **23**, 683–704 (2022).
29. Paulk, A. C. et al. Large-scale neural recordings with single neuron resolution using Neuropixels probes in human cortex. *Nat. Neurosci.* **25**, 252–263 (2022).
30. Steinmetz, N. A. et al. Neuropixels 2.0: a miniaturized high-density probe for stable, long-term brain recordings. *Science* **372**, eabf4588 (2021).
31. Urai, A. E., Doiron, B., Leifer, A. M. & Churchland, A. K. Large-scale neural recordings call for new insights to link brain and behavior. *Nat. Neurosci.* **25**, 11–19 (2022).
32. Vogt, N. Massively parallel intracellular recordings. *Nat. Methods* **16**, 1079–1079 (2019).
33. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937 (2016).
Introduces the sparse identification of non-linear dynamical systems (SINDy) framework for DS reconstruction that delivers an interpretable representation of the dynamics, based on a known function library, and can be trained in a very efficient way.
34. Champion, K., Lusch, B., Kutz, J. N. & Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci. USA* **116**, 22445–22451 (2019).
The first study to combine autoencoders with a DS reconstruction model (SINDy) in order to find suitable low-dimensional latent representations and coordinate transformations on which the dynamics can be efficiently learned.

35. Durstewitz, D. A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. *PLoS Comput. Biol.* **13**, e1005542 (2017).
36. Hernandez, D. et al. Nonlinear evolution via spatially-dependent linear dynamics for electrophysiology and calcium data. *Neurons Behav. Data Anal. Theory* **3**, 3 (2020).
37. Kass, R. E., Eden, U. T. & Brown, E. N. *Analysis of Neural Data* (Springer, 2014).
38. Kim, T. D., Luo, T. Z., Pillow, J. W. & Brody, C. D. Inferring latent dynamics underlying neural population activity via neural differential equations. In *Proc. 38th International Conference on Machine Learning* (eds Meila, M. & Tong, Z.) 5551–5561 (PMLR, 2021).
39. Koppe, G., Toutounji, H., Kirsch, P., Lis, S. & Durstewitz, D. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI. *PLoS Comput. Biol.* **15**, e1007263 (2019).
40. Kramer, D., Bommer, P. L., Tombolini, C., Koppe, G. & Durstewitz, D. Reconstructing nonlinear dynamical systems from multi-modal time series. In *Proc. 39th International Conference on Machine Learning* (eds Chaudhuri, K. et al.) 11613–11633 (PMLR, 2022).
Develops an architecture specifically for DS reconstruction that enables the exploitation of many statistically different data modalities simultaneously for reconstruction, such as neural recordings and behavioural responses.
41. Pandarinath, C. et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat. Methods* **15**, 805–815 (2018).
Takes previous statistical inference frameworks for RNNs from neural data one step further, situating them in a deep variational autoencoder structure that also allows for the inference of unobserved inputs to a given target area.
42. Paninski, L. & Cunningham, J. P. Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. *Curr. Opin. Neurobiol.* **50**, 232–241 (2018).
43. Alligood, K. T., Sauer, T. D. & Yorke, J. A. *Chaos: An Introduction to Dynamical Systems* (Springer, 1996).
44. Perko, L. *Differential Equations and Dynamical Systems* Vol. 7 (Springer, 2001).
45. Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* (CRC, 2018).
46. Vyas, S., Golub, M. D., Sussillo, D. & Shenoy, K. V. Computation through neural population dynamics. *Annu. Rev. Neurosci.* **43**, 249–275 (2020).
47. Funahashi, S., Bruce, C. J. & Goldman-Rakic, P. S. Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *J. Neurophysiol.* **61**, 331–349 (1989).
48. Fuster, J. Unit activity in prefrontal cortex during delayed-response performance: neuronal correlates of transient memory. *J. Neurophysiol.* **36**, 61–78 (1973).
49. Fuster, J. *The Prefrontal Cortex* 5th edn (Academic, 2015).
50. Miller, E. K., Erickson, C. A. & Desimone, R. Neural mechanisms of visual working memory in prefrontal cortex of the macaque. *J. Neurosci.* **16**, 5154 (1996).
51. Albantakis, L. & Deco, G. The encoding of alternatives in multiple-choice decision making. *Proc. Natl Acad. Sci. USA* **106**, 10308–10313 (2009).
52. Wang, X.-J. Decision making in recurrent neuronal circuits. *Neuron* **60**, 215–234 (2008).
53. Gardner, R. J. et al. Toroidal topology of population activity in grid cells. *Nature* **602**, 123–128 (2022).
54. Seung, H. S. How the brain keeps the eyes still. *Proc. Natl Acad. Sci. USA* **93**, 13339–13344 (1996).
55. Seung, H. S., Lee, D. D., Reis, B. Y. & Tank, D. W. Stability of the memory of eye position in a recurrent network of conductance-based model neurons. *Neuron* **26**, 259–271 (2000).
56. Wang, J., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible timing by temporal scaling of cortical responses. *Nat. Neurosci.* **21**, 102–110 (2018).
57. Zhang, K. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *J. Neurosci.* **16**, 2112–2126 (1996).
58. Marder, E. & Bucher, D. Central pattern generators and the control of rhythmic movements. *Curr. Biol.* **11**, R986–R996 (2001).
59. Marder, E., Goeritz, M. L. & Otopalik, A. G. Robust circuit rhythms in small circuits arise from variable circuit components and mechanisms. *Curr. Opin. Neurobiol.* **31**, 156–163 (2015).
60. Lindén, H., Petersen, P. C., Vestergaard, M. & Berg, R. W. Movement is governed by rotational neural dynamics in spinal motor networks. *Nature* **610**, 526–531 (2022).
61. Russo, A. A. et al. Motor cortex embeds muscle-like commands in an untangled population response. *Neuron* **97**, 953–966.e8 (2018).
62. Russo, A. A. et al. Neural trajectories in the supplementary motor area and motor cortex exhibit distinct geometries, compatible with different classes of computation. *Neuron* **107**, 745–758.e6 (2020).
63. Landau, I. D. & Sompolinsky, H. Coherent chaos in a recurrent neural network with structured connectivity. *PLoS Comput. Biol.* **14**, e1006309 (2018).
64. London, M., Roth, A., Beeren, L., Häusser, M. & Latham, P. E. Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature* **466**, 123–127 (2010).
65. Durstewitz, D., Vittoz, N. M., Floresco, S. B. & Seamans, J. K. Abrupt transitions between prefrontal neural ensemble states accompany behavioral transitions during rule learning. *Neuron* **66**, 438–448 (2010).
66. Karlsson, M. P., Tervo, D. G. R. & Karpova, A. Y. Network resets in medial prefrontal cortex mark the onset of behavioral uncertainty. *Science* **338**, 135–139 (2012).
67. Kopell, N., Ermentrout, G. B., Whittington, M. A. & Traub, R. D. Gamma rhythms and beta rhythms have different synchronization properties. *Proc. Natl Acad. Sci. USA* **97**, 1867–1872 (2000).
68. Roxin, A., Brunel, N. & Hansel, D. Rate models with delays and the dynamics of large networks of spiking neurons. *Prog. Theor. Phys. Supp.* **161**, 68–85 (2006).
69. Traub, R. D., Whittington, M. A., Stanford, I. M. & Jefferys, J. G. R. A mechanism for generation of long-range synchronous fast oscillations in the cortex. *Nature* **383**, 621–624 (1996).
70. Zipser, D., Kehoe, B., Littlewort, G. & Fuster, J. A spiking network model of short-term active memory. *J. Neurosci.* **13**, 3406 (1993).
71. Zipser, D. Recurrent network model of the neural mechanism of short-term active memory. *Neural Comput.* **3**, 179–193 (1991).
Early study that introduces the idea of gaining insight into neural dynamics and computation by training RNNs on similar tasks to those used in animal experiments and comparing RNN unit responses to those neurophysiologically observed.
72. Elman, J. L. Finding structure in time. *Cogn. Sci.* **14**, 179–211 (1990).
73. Pearlmutter, B. A. *Dynamic Recurrent Neural Networks* (Carnegie Mellon Univ., 1990).
74. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
75. Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**, 544–557 (2009).
Introduces a novel RNN training algorithm (FORCE) and developed the idea of shaping a repertoire of complex spontaneous chaotic dynamics into a variety of desired output patterns, such as human walking motions.
76. Beiran, M., Meirhaeghe, N., Sohn, H., Jazayeri, M. & Ostojic, S. Parametric control of flexible timing through low-dimensional neural manifolds. *Neuron* **111**, 739–753.e8 (2023).
77. Barbosa, J. et al. Flexible selection of task-relevant features through population gating. Preprint at *bioRxiv* https://doi.org/10.1101/2022.07.21.500962 (2022).
78. Chaisangmongkon, W., Swaminathan, S. K., Freedman, D. J. & Wang, X.-J. Computing by robust transience: how the fronto-parietal network performs sequential, category-based decisions. *Neuron* **93**, 1504–1517.e4 (2017).
79. Rajalingham, R., Piccato, A. & Jazayeri, M. Recurrent neural networks with explicit representation of dynamic latent variables can mimic behavioral patterns in a physical inference task. *Nat. Commun.* **13**, 5865 (2022).
Elegant work that illustrates how modifying the loss function of an RNN to accommodate specific assumptions about how animals or humans learn a task can substantially improve an RNN's fit with behavioural observations.
80. Remington, E. D., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible sensorimotor computations through rapid reconfiguration of cortical dynamics. *Neuron* **98**, 1005–1019.e5 (2018).
81. Roach, J. P., Churchland, A. K. & Engel, T. A. Choice selective inhibition drives stability and competition in decision circuits. *Nat. Commun.* **14**, 147 (2023).
82. Sohn, H., Narain, D., Meirhaeghe, N. & Jazayeri, M. Bayesian computation through cortical latent dynamics. *Neuron* **103**, 934–947.e5 (2019).
83. Song, H. F., Yang, G. R. & Wang, X.-J. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework. *PLoS Comput. Biol.* **12**, e1004792 (2016).
84. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* **18**, 1025–1033 (2015).
85. Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T. & Wang, X.-J. Task representations in neural networks trained to perform many cognitive tasks. *Nat. Neurosci.* **22**, 297–306 (2019).
86. Driscoll, L., Shenoy, K. & Sussillo, D. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. Preprint at *bioRxiv* https://doi.org/10.1101/2022.08.15.503870 (2022).
87. Goudar, V., Peysakhovich, B., Freedman, D. J., Buffalo, E. A. & Wang, X.-J. Schema formation in a neural population subspace underlies learning-to-learn in flexible sensorimotor problem-solving. *Nat. Neurosci.* **26**, 879–890 (2023).
88. Johnston, W. J. & Fusi, S. Abstract representations emerge naturally in neural networks trained to perform multiple tasks. *Nat. Commun.* **14**, 1040 (2023).
89. Dubreuil, A., Valente, A., Beiran, M., Mastrogiuseppe, F. & Ostojic, S. The role of population structure in computations through neural dynamics. *Nat. Neurosci.* **25**, 783–794 (2022).
A series of elegant methodological investigations showcasing how task-trained low-rank RNNs can be used and systematically dissected and analysed to reveal the computations implemented by the RNN dynamics and the underlying network structure.
90. Mastrogiuseppe, F. & Ostojic, S. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron* **99**, 609–623.e29 (2018).
91. Yu, B. M. et al. Extracting dynamical structure embedded in neural activity. In *Proc. 18th Advances in Neural Information Processing Systems* (eds. Weiss, Y., Schölkopf, B. & Platt, J.) 1545-1552 (MIT Press, Vancouver, 2005).
Early study that develops a statistical inference framework for probabilistic (data-inferred) RNNs in order to reveal smoothed latent trajectories underlying cortical multiple single-unit recordings.
92. Zhao, Y. & Park, I. M. Variational online learning of neural dynamics. *Front. Comput. Neurosci.* **14** (2020).
93. Rajan, K., Harvey, C. D. & Tank, D. W. Recurrent network models of sequence generation and memory. *Neuron* **90**, 128–142 (2016).
Trains RNNs using the FORCE algorithm directly on neurophysiological data to reveal dynamical mechanisms underlying sequence generation and working memory.

94. Archer, E., Park, I. M., Buesing, L., Cunningham, J. & Paninski, L. Black box variational inference for state space models. In *International Conference on Learning Representations* (ICLR, San Juan, 2016).
95. Keshtkaran, M. R. et al. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nat. Methods* **19**, 1572–1577 (2022).
96. Whiteway, M. R. & Butts, D. A. Revealing unobserved factors underlying cortical activity with a rectified latent variable model applied to neural population recordings. *J. Neurophysiol.* **117**, 919–936 (2016).
97. Zhao, Y. & Park, I. M. Interpretable nonlinear dynamic modeling of neural trajectories. In *Proc. 29th Advances in Neural Information Processing Systems* (eds. Lee D. et al.) 3333–3341 (Curran Associates, Inc., 2016).
98. Buesing, L., Macke, J. H. & Sahani, M. Learning stable, regularised latent models of neural population dynamics. *Network* **23**, 24–47 (2012).
99. Linderman, S. et al. Bayesian learning and inference in recurrent switching linear dynamical systems. (eds Singh, A. & Zhu, J.) In *Proc. of the 20th International Conference on Artificial Intelligence and Statistics* 914–922 (PMLR, Ft. Lauderdale, 2017).
100. Macke, J. H., Buesing, L. & Sahani, M. in *Advanced State Space Methods for Neural and Clinical Data* 137–159 (Cambridge Univ. Press, 2015).
101. Paninski, L. et al. A new look at state-space models for neural data. *J. Comput. Neurosci.* **29**, 107–126 (2010).
102. Pillow, J. W., Ahmadian, Y. & Paninski, L. Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. *Neural Comput.* **23**, 1–45 (2011).
103. Smith, A. C. & Brown, E. N. Estimating a state-space model from point process observations. *Neural Comput.* **15**, 965–991 (2003).
104. Ghahramani, Z. & Hinton, G. E. Variational learning for switching state-space models. *Neural Comput.* **12**, 831–864 (2000).
105. Nassar, J., Linderman, S., Bugallo, M. & Park, I. M. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. In *International Conference on Learning Representations* (ICLR, New Orleans, 2019).
106. Nair, A. et al. An approximate line attractor in the hypothalamus encodes an aggressive state. *Cell* **186**, 178–193.e15 (2023).
107. Rezende, D. J., Mohamed, S. & Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. 31st International Conference on Machine Learning* (eds. Xing, E. P & Jebara. T) 1278–1286 (PMLR, 2014).
108. Hess, F., Monfared, Z., Brenner, M. & Durstewitz, D. Generalized teacher forcing for learning chaotic dynamics. In *Proc. 40th International Conference on Machine Learning* (eds Krause, A. et al.) 13017–13049 (PMLR, 2023).
Introduces a highly efficient algorithm based on the idea of generalized teacher forcing for training low-dimensional RNNs for DS reconstruction on complex chaotic real-world data, overcoming the exploding-gradient problem.
109. Arribas, D., Zhao, Y. & Park, I. M. Rescuing neural spike train models from bad MLE. In *Proc. 33rd Advances in Neural Information Processing Systems* (eds. Larochelle, H. et al.) 2293–2303 (Curran Associates, Inc., 2020).
110. Brenner, M. et al. Tractable dendritic RNNs for reconstructing nonlinear dynamical systems. In *Proc. 39th International Conference on Machine Learning* (eds. Chaudhuri, K. et al.) 2292–2320 (PMLR, 2022).
111. Kantz, H. & Schreiber, T. *Nonlinear Time Series Analysis* Vol. 7 (Cambridge Univ. Press, 2004).
112. Sauer, T., Yorke, J. A. & Casdagli, M. Embedology. *J. Stat. Phys.* **65**, 579–616 (1991).
A landmark paper generalizing and extending previous delay embedding theorems by Whitney and Takens to account for attractors with fractal geometry such as chaotic sets.
113. Takens, F. in *Dynamical Systems and Turbulence, Warwick 1980* Vol. 898 pp. 366–381 (Springer, 1981).
A landmark paper formally developing the idea that a topologically equivalent reconstruction (embedding) of the trajectories of a dynamical system (and possibly attractor) can be achieved through a delay coordinate map under specific conditions.
114. Tenenbaum, J. B., Silva, V. D. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323 (2000).
115. Belkin, M. & Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. (eds Dietterich, T., Becker, S. & Ghahramani, Z.) In *Proc. 14th Advances in Neural Information Processing Systems* 585–591 (Curran Associates, Inc., Vancouver, 2001).
116. Llavona, J. G. *Approximation of Continuously Differentiable Functions* (Elsevier, 1986).
117. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2**, 303–314 (1989).
118. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
119. Lu, Z., Pu, H., Wang, F., Hu, Z. & Wang, L. The expressive power of neural networks: a view from the width. In *Proc. 30th Advance on Neural Information Processing Systems* (eds. Guyon, I. et al.) 6231–6239 (Curran Associates, Inc., 2017).
120. Storace, M. & De Feo, O. PWL approximation of nonlinear dynamical systems, part I: structural stability. *J. Phys. Conf. Ser.* **22**, 208 (2005).
121. Chen, T. & Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Netw.* **6**, 911–917 (1995).
122. Funahashi, K. I. & Nakamura, Y. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Netw.* **6**, 801–806 (1993).
Early study proving that finite-time trajectories from DS can be universally approximated to arbitrary precision by RNNs, results that were later extended to infinite-time trajectories and DS more generally.
123. Hanson, J. & Raginsky, M. In *Learning for Dynamics and Control* (eds Bayen, A. M. et al.) 384–392 (PMLR, 2020).
124. Kimura, M. & Nakano, R. Learning dynamical systems by recurrent neural networks from orbits. *Neural Netw.* **11**, 1589–1599 (1998).
125. Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229 (2021).
126. Trischler, A. P. & D'Eleuterio, G. M. T. Synthesis of recurrent neural networks for dynamical system simulation. *Neural Netw.* **80**, 67–78 (2016).
127. Friston, K. J., Harrison, L. & Penny, W. Dynamic causal modelling. *Neuroimage* **19**, 1273–1302 (2003).
128. Sani, O. G., Abbaspourazad, H., Wong, Y. T., Pesaran, B. & Shanechi, M. M. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nat. Neurosci.* **24**, 140–149 (2021).
129. Yu, B. M. et al. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *J. Neurophysiol.* **102**, 614–635 (2009).
130. Haußmann, M., Gerwinn, S., Look, A., Rakitsch, B. & Kandemir, M. Learning partially known stochastic dynamics with empirical PAC Bayes. In *International Conference on Artificial Intelligence and Statistics* (eds. Banerjee, A. & Fukumizu, K.) 478–486 (PMLR, 2021).
131. Mikhaeil, J. M., Monfared, Z. & Durstewitz, D. On the difficulty of learning chaotic dynamics with RNNs. In *Proc. 35th Conference on Neural Information Processing Systems* (eds. Koyejo, S. et al.) (Curran Associates, Inc., 2022).
Establishes a formal connection between the dynamics of an empirically observed system and the RNN used for learning its dynamics, and the exploding and vanishing gradient problem.
132. Pathak, J., Hunt, B., Grivan, M., Lu, Z. & Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102 (2018).
133. Seleznev, A., Mukhin, D., Gavrilov, A., Loskutov, E. & Feigin, A. Bayesian framework for simulation of dynamical systems from multidimensional data using recurrent neural network. *Chaos* **29**, 123115 (2019).
134. Vlachas, P. R., Byeon, W., Wan, Z. Y., Sapsis, T. P. & Koumoutsakos, P. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc. A: Math. Phys. Eng. Sci.* https://doi.org/10.1098/rspa.2017.0844 (2018).
135. Vlachas, P. R. et al. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Netw.* **126**, 191–217 (2020).
136. Cho, K., van Merrienboer, B., Bahdanau, D. & Bengio, Y. On the properties of neural machine translation: encoder–decoder approaches. In *Proc. of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (Association for Computational Linguistics, 2014).
137. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
Introduces the LSTM gated memory architecture for dealing with the previously unresolved exploding-gradient and vanishing-gradient problem, one of the most widely applied RNNs that led to much renewed interest in up-to-that-point difficult-to-train RNNs.
138. Chen, R. T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. Neural ordinary differential equations. In *Proc. 31st Advances in Neural Information Processing Systems* (eds. Bengio, S. et al.) 6571–6583 (Curran Associates, Inc., 2018).
Introduces a novel class of continuous-time RNNs (neural ODEs) and efficient training algorithms for this class, which extend conventional deep NNs into possibly infinitely deep architectures.
139. Rusch, T. K., Mishra, S., Erichson, N. B. & Mahoney, M. W. Long expressive memory for sequence modeling. In *International Conference on Learning Representations* (ICLR, 2022).
140. Bengio, Y., Simard, P. & Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **5**, 157–166 (1994).
141. Hochreiter, S. *Untersuchungen zu Dynamischen Neuronalen Netzen* Diploma thesis, Technische Universität München (1991).
142. Werbos, P. J. Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.* **1**, 339–356 (1988).
143. Schmidt, D., Koppe, G., Monfared, Z., Beutelspacher, M. & Durstewitz, D. Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies. In *International Conference on Learning Representations* (ICLR, 2021).
144. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1412.3555 (2014).
145. Rusch, T. K. & Mishra, S. UnICORNN: a recurrent model for learning very long time dependencies. In *Proc. 38th International Conference on Machine Learning* (eds. Meila, M. & Tong, Z.) 9168–9178 (PMLR, 2021).
146. Rusch, T. K. & Mishra, S. Coupled oscillatory recurrent neural network (coRNN): an accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations* (ICLR, Vienna, 2021).

147. Arjovsky, M., Shah, A. & Bengio, Y. Unitary evolution recurrent neural networks. In *Proc. 33rd International Conference on Machine Learning* (eds Balcan M. F. & Weinberger K. Q.) 1120–1128 (PMLR, 2016).
148. Chang, B., Chen, M., Haber, E. & Chi, E. H. AntisymmetricRNN: a dynamical system view on recurrent neural networks. In *International Conference on Learning Representations* (ICLR, New Orleans, 2019)
149. Erichson, N. B., Azencot, O., Queiruga, A., Hodgkinson, L. & Mahoney, M. W. Lipschitz recurrent neural networks. In *International Conference on Learning Representations* (ICLR, Vienna, 2021).
150. Helfrich, K., Willmott, D. & Ye, Q. Orthogonal recurrent neural networks with scaled Cayley transform. In *Proc. 35th International Conference on Machine Learning* (eds. Dy, J. & Krause, A.) 1969–1978 (PMLR, 2018).
151. Kag, A., Zhang, Z. & Saligrama, V. RNNs incrementally evolving on an equilibrium manifold: a panacea for vanishing and exploding gradients? In *International Conference on Learning Representations* (ICLR, 2020).
152. Kolter, J. Z. & Manek, G. Learning stable deep dynamics models. In *Proc. 32nd Advances in Neural Information Processing Systems* (eds. Wallach, H. et al.) 11128–11136 (Curran Associates, Inc., 2019).
153. Engelken, R., Wolf, F. & Abbott, L. F. Lyapunov spectra of chaotic recurrent neural networks. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2006.02427 (2020).
154. Degn, H., Holden, A. V. & Olsen, L. F. *Chaos in Biological Systems* Vol. 138 (Springer, 2013).
155. Brenner, M., Koppe, G. & Durstewitz, D. Multimodal teacher forcing for reconstructing nonlinear dynamical systems. In *The 37th AAAI Conference on Artificial Intelligence* (AAAI, Washington, 2023).
156. Lusch, B., Kutz, J. N. & Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**, 4950 (2018).
157. Platt, J. A., Penny, S. G., Smith, T. A., Chen, T.-C. & Abarbanel, H. D. I. Constraining chaos: enforcing dynamical invariants in the training of recurrent neural networks. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2304.12865 (2023).
Considers the inclusion of invariant DS characteristics like Lyapunov exponents directly into the loss function of the training method to improve DS reconstruction and long-term behaviour.
158. Doya, K. Bifurcations in the learning of recurrent neural networks. In *Proc. IEEE International Symposium on Circuits and Systems* 2777–2780 (1992).
159. Vlachas, P. R. & Koumoutsakos, P. Learning from predictions: fusing training and autoregressive inference for long-term spatiotemporal forecasts. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2302.11101 (2023).
160. Williams, R. J. & Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.* **1**, 270–280 (1989).
161. Abarbanel, H. *Predicting the Future: Completing Models of Observed Complex Systems* (Springer, 2013).
162. Abarbanel, H. D. I., Creveling, D. R., Farsian, R. & Kostuk, M. Dynamical state and parameter estimation. *SIAM J. Appl. Dyn. Syst.* **8**, 1341–1381 (2009).
163. Abarbanel, H. D. I., Creveling, D. R. & Jeanne, J. M. Estimation of parameters in nonlinear systems using balanced synchronization. *Phys. Rev.* **77**, 016208 (2008).
164. Platt, J. A., Wong, A., Clark, R., Penny, S. G. & Abarbanel, H. D. I. Robust forecasting using predictive generalized synchronization in reservoir computing. *Chaos* **31**, 123118 (2021).
165. Verzelli, P., Alippi, C. & Livi, L. Learn to synchronize, synchronize to learn. *Chaos* **31**, 083119 (2021).
166. Singh, S. K. et al. PI-LSTM: physics-infused long short-term memory network. In *IEEE International Conference on Machine Learning and Applications* 34–41 (IEEE, 2019).
167. Voss, H. U., Timmer, J. & Kurths, J. Nonlinear dynamical system identification from uncertain and indirect measurements. *Int. J. Bifurcat. Chaos* **14**, 1905–1933 (2004).
One of the earlier studies reviewing ideas, multiple shooting, on how to improve model-based DS reconstruction in the face of complex (possibly fractal) loss function landscapes.
168. Botvinick-Greenhouse, J., Martin, R. & Yang, Y. Learning dynamics on invariant measures using PDE-constrained optimization. *Chaos* **33**, 063152 (2023).
169. Jiang, R., Lu, P. Y., Orlova, E. & Willett, R. Training neural operators to preserve invariant measures of chaotic attractors. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2306.01187 (2023).
170. Chen, J. & Wu, K. Deep-OSG: a deep learning approach for approximating a family of operators in semigroup to model unknown autonomous systems. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2302.03358 (2023).
171. Rackauckas, C. et al. Universal differential equations for scientific machine learning. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2001.04385 (2020).
172. Chen, R. T. Q., Amos, B. & Nickel, M. Learning neural event functions for ordinary differential equations. In *International Conference on Learning Representations* (ICLR, 2021).
173. Kaptanoglu, A. A. et al. PySINDy: a comprehensive python package for robust sparse system identification. *J. Open Source Softw.* **7**, 3994 (2022).
174. Bertschinger, N. & Natschläger, T. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Comput.* **16**, 1413–1436 (2004).
175. Jaeger, H. & Haas, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
A landmark paper that introduces echo state networks (or reservoir computers), one of the most successful and still widely used architectures and training methods for learning DS and predicting their temporal evolution.
176. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
177. Jüngling, T. et al. Reconstruction of complex dynamical systems from time series using reservoir computing. In *IEEE International Symposium on Circuits and Systems* 1–5 (IEEE, 2019)
178. Patel, D. & Ott, E. Using machine learning to anticipate tipping points and extrapolate to post-tipping dynamics of non-stationary dynamical systems. *Chaos* **33**, 023143 (2023).
179. Raissi, M. Deep hidden physics models: deep learning of nonlinear partial differential equations. *J. Mach. Learn. Res.* **19**, 1–24 (2018).
Introduces a new approach to DS reconstruction, partly similar in spirit to neural ODEs, which combines approximation of the vector field and that of the solution operator through deep neural networks, and at the same time makes it possible to incorporate physical domain knowledge.
180. Abarbanel, H. D. I., Rozdeba, P. J. & Shirman, S. Machine learning: deepest learning as statistical data assimilation problems. *Neural Comput.* **30**, 2025–2055 (2018).
181. Salvi, C., Lemercier, M. & Gerasimovics, A. Neural stochastic PDEs: resolution-invariant learning of continuous spatiotemporal dynamics. In *Proc. 35th Advances in Neural Information Processing Systems* (eds Koyejo, S. et al.) (Curran Associates, Inc., 2022).
182. Gelbrecht, M., Boers, N. & Kurths, J. Neural partial differential equations for chaotic systems. *New J. Phys.* **23**, 043005 (2021).
183. Li, Z. et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations* (ICLR, 2021).
Elegant and powerful solution for deep learning of DS described by (theoretically infinite dimensional) systems of partial different equations (PDEs), based on the idea of approximating the dynamics in function space by Fourier neural operators.
184. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
185. Rudy, S. H., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614 (2017).
186. De Feo, O. & Storace, M. PWL approximation of nonlinear dynamical systems, part II: identification issues. *J. Phys. Conf. Ser.* **22**, 002 (2005).
187. Tibshirani, R. Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. B Stat. Methodol.* **58**, 267–288 (1996).
188. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1409.0473 (2016).
189. Sukhbaatar, S., Szlam, A., Weston, J. & Fergus, R. End-to-end memory networks. In *Proc. 28th Advances in Neural Information Processing Systems* (eds. Cortes, C. et al.) 2440–2448 (Curran Associates, Inc., 2015).
190. Vaswani, A. et al. Attention is all you need. In *Proc. 30th Advances in Neural Information Processing Systems* (eds Guyon, I. et al.) 5998–6008 (Curran Associates, Inc., 2017).
191. OpenAi. GPT-4 technical report. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2303.08774 (2023).
192. Geneva, N. & Zabaras, N. Transformers for modeling physical systems. *Neural Netw.* **146**, 272–289 (2022).
193. Shalova, A. & Oseledets, I. Tensorized transformer for dynamical systems modeling. In *International Conference on Learning Representations* (ICLR, 2021).
194. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).
195. Bakarji, J., Champion, K., Kutz, J. N. & Brunton, S. L. Discovering governing equations from partial measurements with deep delay autoencoders. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2201.05136 (2022).
196. Gilpin, W. Deep reconstruction of strange attractors from time series. In *Proc. 33rd Advance on Neural Information Processing Systems* (eds Larochelle, H. et al.) 204–216 (Curran Associates, Inc., 2020).
197. Allen, C. & Stevens, C. F. An evaluation of causes for unreliability of synaptic transmission. *Proc. Natl Acad. Sci. USA* **91**, 10380–10383 (1994).
198. Zhao, Y. & Park, I. M. Variational latent Gaussian process for recovering single-trial dynamics from population spike trains. *Neural Comput.* **29**, 1293–1316 (2017).
199. Duncker, L., Bohner, G., Boussard, J. & Sahani, M. Learning interpretable continuous-time models of latent stochastic dynamical systems. In *Proc. 36th International Conference on Machine Learning* (eds. Chaudhuri, K. & Salakhutdinov, R.) 1726–1734 (PMLR, Los Angeles, 2019).
200. Look, A., Qiu, C., Rudolph, M. R., Peters, J. & Kandemir, M. Deterministic inference of neural stochastic differential equations. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2006.08973 (2020).
201. Xu, W., Chen, R. T. Q., Li, X. & Duvenaud, D. Infinitely deep Bayesian neural networks with stochastic differential equations. In *Proc. 25th International Conference on Artificial Intelligence and Statistics* (eds. Camps-Valls, G., Ruiz, F. J. R. & Valera I.) 721–738 (PMLR, 2022).
202. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations* (ICLR, 2013).
203. Rahman, A., Srikumar, V. & Smith, A. D. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Appl. Energy* **212**, 372–385 (2018).
204. Kim, B. et al. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *International Conference on Intelligent Transportation Systems* 399–404 (IEEE, 2017).

205. Wood, S. N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* **466**, 1102–1104 (2010).
Important paper from the statistical community that points out that conventional likelihood functions are not suitable for learning parameters of a chaotic dynamical system, and instead suggests a surrogate likelihood based on (time-invariant in the limit) summary statistics like autocovariance functions.
206. Das, S., Giannakis, D. & Székely, E. An information-geometric approach to feature extraction and moment reconstruction in dynamical systems. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2004.02172 (2020).
207. Durstewitz, D. *Advanced Data Analysis in Neuroscience: Integrating Statistical and Computational Models* (Springer, 2017).
208. Galgali, A. R., Sahani, M. & Mante, V. Residual dynamics resolves recurrent contributions to neural computation. *Nat. Neurosci.* **26**, 326–338 (2023).
209. Nakahara, H. & Doya, K. Near-saddle-node bifurcation behavior as dynamics in working memory for goal-directed behavior. *Neural Comput.* **10**, 113–132 (1998).
210. Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* **25**, 626–649 (2013).
211. Brunton, S. L., Budišić, M., Kaiser, E. & Kutz, J. N. Modern Koopman Theory for Dynamical Systems. *SIAM Rev.* **64**, 229–340 (2022).
212. Smith, J., Linderman, S. & Sussillo, D. Reverse engineering recurrent neural networks with Jacobian switching linear dynamical systems. In *Proc. 34th Advances in Neural Information Processing Systems* (eds. Ranzato, M. et al.) 16700–16713 (Curran Associates, Inc., 2021).
213. Smith, J. T., Warrington, A. & Linderman, S. W. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations* (ICLR, 2023).
214. Floryan, D. & Graham, M. D. Data-driven discovery of intrinsic dynamics. *Nat. Mach. Intell.* **4**, 1113–1120 (2022).
215. Turner, E., Dabholkar, K. V. & Barak, O. Charting and navigating the space of solutions for recurrent neural networks. In *Proc. 34th Advances in Neural Information Processing Systems* (eds. Ranzato, M. et al.) 25320–25333 (Curran Associates, Inc., 2021).
Introduces a set of ideas and tools of how dynamics and computations in RNNs trained on neuroscience tasks could be algorithmically interpreted.
216. Reinbold, P. A. K., Kageorge, L. M., Schatz, M. F. & Grigoriev, R. O. Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nat. Commun.* **12**, 3219 (2021).
217. Altan, E., Solla, S. A., Miller, L. E. & Perreault, E. J. Estimating the dimensionality of the manifold underlying multi-electrode neural recordings. *PLoS Comput. Biol.* **17**, e1008591 (2021).
218. Duncker, L. & Sahani, M. Dynamics on the manifold: identifying computational dynamical activity from neural population recordings. *Curr. Opin. Neurobiol.* **70**, 163–170 (2021).
219. Gallego, J. A., Perich, M. G., Miller, L. E. & Solla, S. A. Neural manifolds for the control of movement. *Neuron* **94**, 978–984 (2017).
220. Jazayeri, M. & Ostojic, S. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Curr. Opin. Neurobiol.* **70**, 113–120 (2021).
221. Melbaum, S. et al. Conserved structures of neural activity in sensorimotor cortex of freely moving rats allow cross-subject decoding. *Nat. Commun.* **13**, 7420 (2022).
222. Hyman, J. M., Ma, L., Balaguer-Ballester, E., Durstewitz, D. & Seamans, J. K. Contextual encoding by ensembles of medial prefrontal cortex neurons. *Proc. Natl Acad. Sci. USA* **109**, 5086–5091 (2012).
223. Kossio, Y. F. K., Goedeke, S., Klos, C. & Memmesheimer, R.-M. Drifting assemblies for persistent memory: neuron transitions and unsupervised compensation. *Proc. Natl Acad. Sci. USA* **118**, e2023832118 (2021).
224. Sadeh, S. & Clopath, C. Contribution of behavioural variability to representational drift. *eLife* **11**, e77907 (2022).
225. Feulner, B. & Clopath, C. Neural manifold under plasticity in a goal driven learning behaviour. *PLoS Comput. Biol.* **17**, e1008621 (2021).
226. Sauer, T. Reconstruction of dynamical systems from interspike intervals. *Phys. Rev. Lett.* **72**, 3811–3814 (1994).
227. Sauer, T. Interspike interval embedding of chaotic signals. *Chaos* **5**, 127–132 (1995).
228. Clopath, C., Bonhoeffer, T., Hübener, M. & Rose, T. Variance and invariance of neuronal long-term representations. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **372**, 20160161 (2017).
229. Ecker, A. S. et al. Decorrelated neuronal firing in cortical microcircuits. *Science* **327**, 584–587 (2010).
230. Mai, B., Sommer, S. & Hauber, W. Motivational states influence effort-based decision making in rats: the role of dopamine in the nucleus accumbens. *Cogn. Affect. Behav. Neurosci.* **12**, 74–84 (2012).
231. Russo, E. et al. Coordinated prefrontal state transition leads extinction of reward-seeking behaviors. *J. Neurosci.* **41**, 2406–2419 (2021).
232. Shimazaki, H., Amari, S.-i, Brown, E. N. & Grün, S. State-space analysis of time-varying higher-order spike correlation for multiple neural spike train data. *PLoS Comput. Biol.* **8**, e1002385 (2012).
233. Park, M., Bohner, G. & Macke, J. H. Unlocking neural population non-stationarities using hierarchical dynamics models. In *Proc. 28th Advances in Neural Information Processing Systems* (eds Cortes, C. et al.) 145–153 (Curran Associates, Inc., 2015).
234. 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. *Nat. Mach. Intell.* **3**, 316–323 (2021).
235. Kirchmeyer, M. et al. Generalizing to new physical systems via context-informed dynamics model. In *Proc. 39th International Conference on Machine Learning* (eds. Chaudhuri, K. et al.) 11283–11301 (PMLR, 2022).
236. Krueger, D. et al. Out-of-distribution generalization via risk extrapolation (REx). In *Proc. 38th International Conference on Machine Learning* (eds. Meila, M. & Tong, Z.) 5815–5826 (PMLR, 2021).
237. Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* 2nd edn (Springer, 2009).
238. Jirsa, V. K., Stacey, W. C., Quilichini, P. P., Ivanov, A. I. & Bernard, C. On the nature of seizure dynamics. *Brain* **137**, 2210–2230 (2014).
239. Naze, S., Bernard, C. & Jirsa, V. Computational modeling of seizure dynamics using coupled neuronal networks: factors shaping epileptiform activity. *PLoS Comput. Biol.* **11**, e1004209 (2015).
240. Fusi, S., Asaad, W. F., Miller, E. K. & Wang, X.-J. A neural circuit model of flexible sensorimotor mapping: learning and forgetting on multiple timescales. *Neuron* **54**, 319–333 (2007).
241. Russo, E. & Durstewitz, D. Cell assemblies at multiple time scales with arbitrary lag constellations. *eLife* **6**, e19428 (2017).
242. Spitmaan, M., Seo, H., Lee, D. & Soltani, A. Multiple timescales of neural dynamics and integration of task-relevant signals across cortex. *Proc. Natl Acad. Sci. USA* **117**, 22522–22531 (2020).
243. Tanaka, G., Matsumori, T., Yoshida, H. & Aihara, K. Reservoir computing with diverse timescales for prediction of multiscale dynamics. *Phys. Rev. Res.* **4**, L032014 (2022).
244. van Vreeswijk, C. & Sompolinsky, H. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* **274**, 1724–1726 (1996).
245. Pereira-Obilinovic, U., Aljadeff, J. & Brunel, N. Forgetting leads to chaos in attractor networks. *Phys. Rev. X* **13**, 011009 (2023).
246. Durstewitz, D. Implications of synaptic biophysics for recurrent network dynamics and active memory. *Neural Netw.* **22**, 1189–1200 (2009).
247. Lorenz, E. N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963).
248. Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N. & Wolpaw, J. R. BCI2000: a general-purpose brain–computer interface (BCI) system. *IEEE Trans. Biomed. Eng.* **51**, 1034–1043 (2004).
249. Hyman, J. M., Whitman, J., Emberly, E., Woodward, T. S. & Seamans, J. K. Action and outcome activity state patterns in the anterior cingulate cortex. *Cereb. Cortex* **23**, 1257–1268 (2013).

Acknowledgements

D.D. discloses support for this work from the German Research Foundation (DFG) through individual grants (Du 354/10–1; Du 354/15–1), within research cluster FOR-5159 ("Resolving prefrontal flexibility"; Du 354/14–1) and through Germany's Excellence Strategy EXC 2181/1–390900948 (STRUCTURES). The authors thank A. Draguhn, C. Lapish, J. Mikhaeil, K. Mitchell, A. Meyer-Lindenberg, Z. Monfared and R. Traub for providing detailed feedback and suggestions on this article, L. Judith for providing the EEG reconstructions in Fig. 3d, J. Hyman for providing the multiple single-unit data used in Fig. 3e, F. Hess for generating the DS reconstruction used in Supplementary Fig. 4 and M. Brenner for providing the code for the RNN animation.

Author contributions

All authors reviewed and/or edited the manuscript before submission and researched data for the article. D.D. wrote the article and contributed substantially to the discussion of the content.

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/s41583-023-00740-7.

Peer review information *Nature Reviews Neuroscience* thanks Demba Ba and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.