Recurrent neural networks in neuroscience

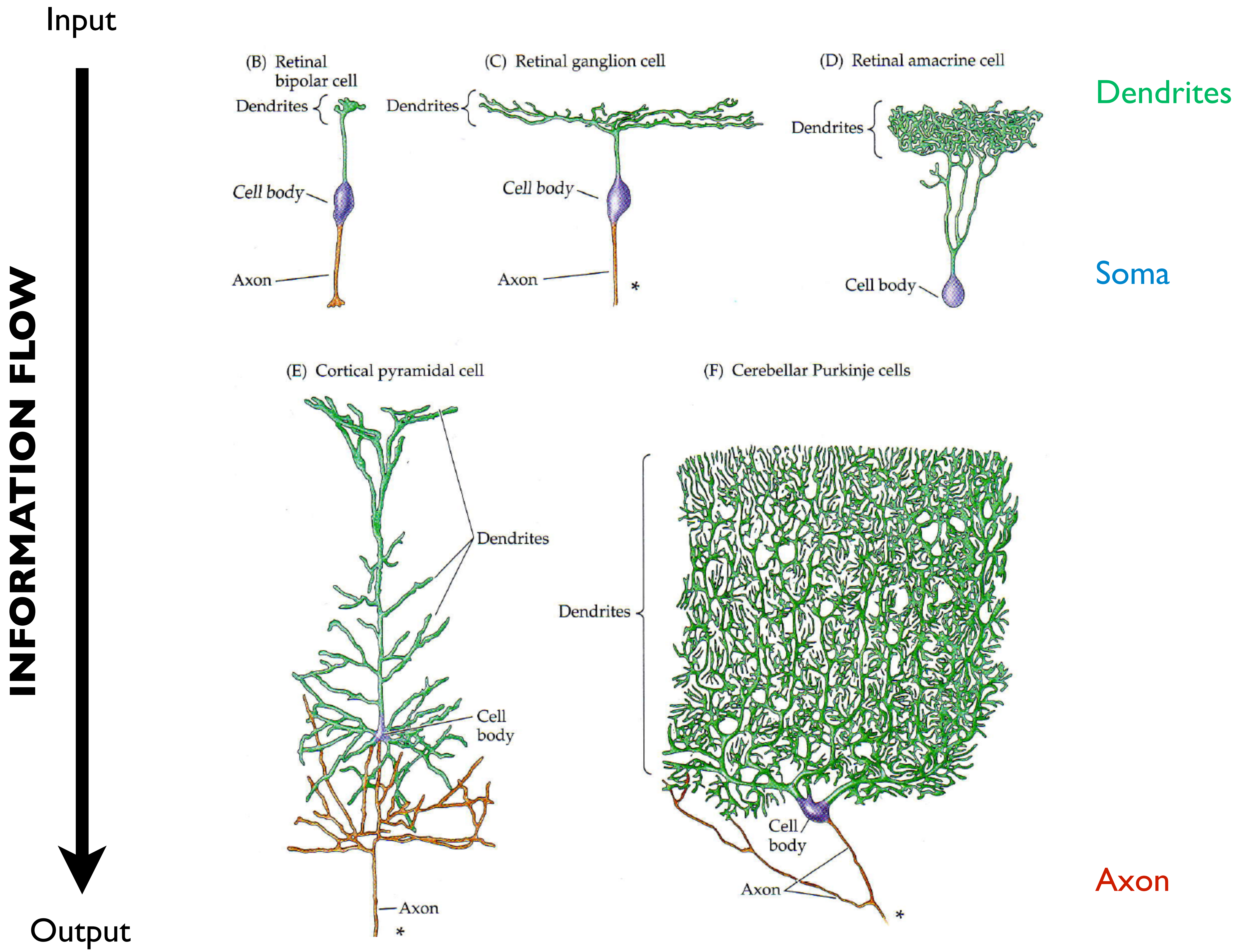
ENS course slides

Adrian Valente — 16/11/2022

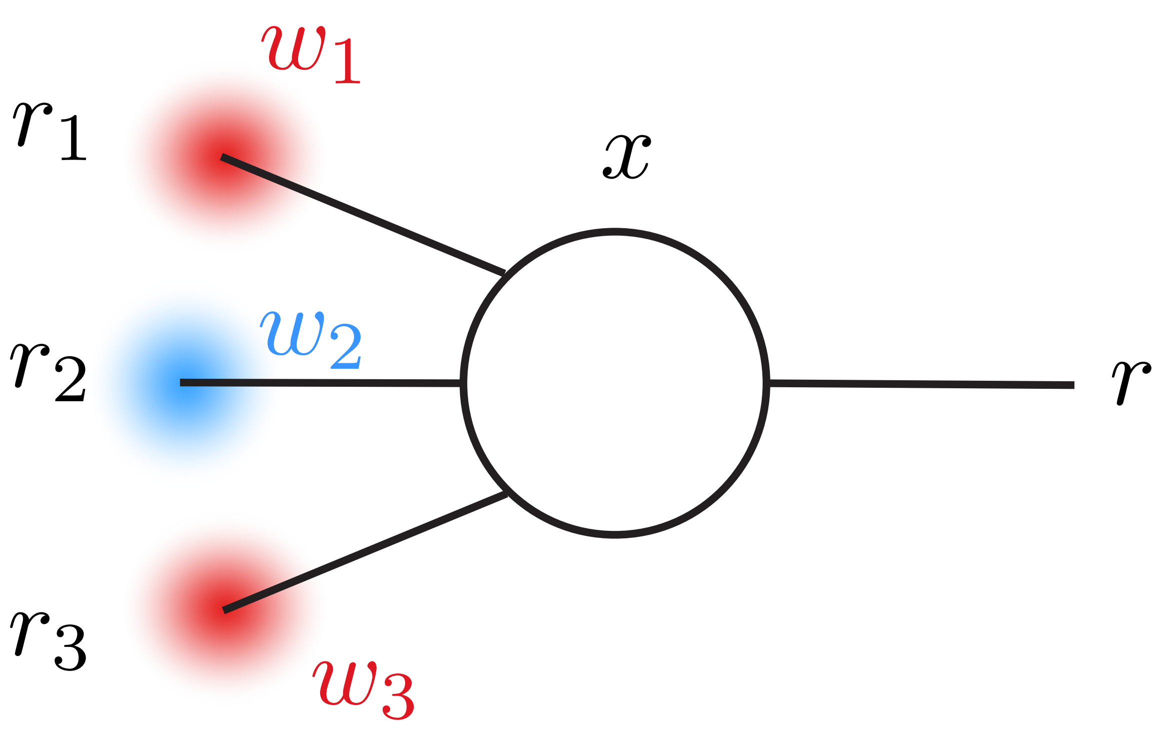
Chapter I

Networks and computations

Real neurons

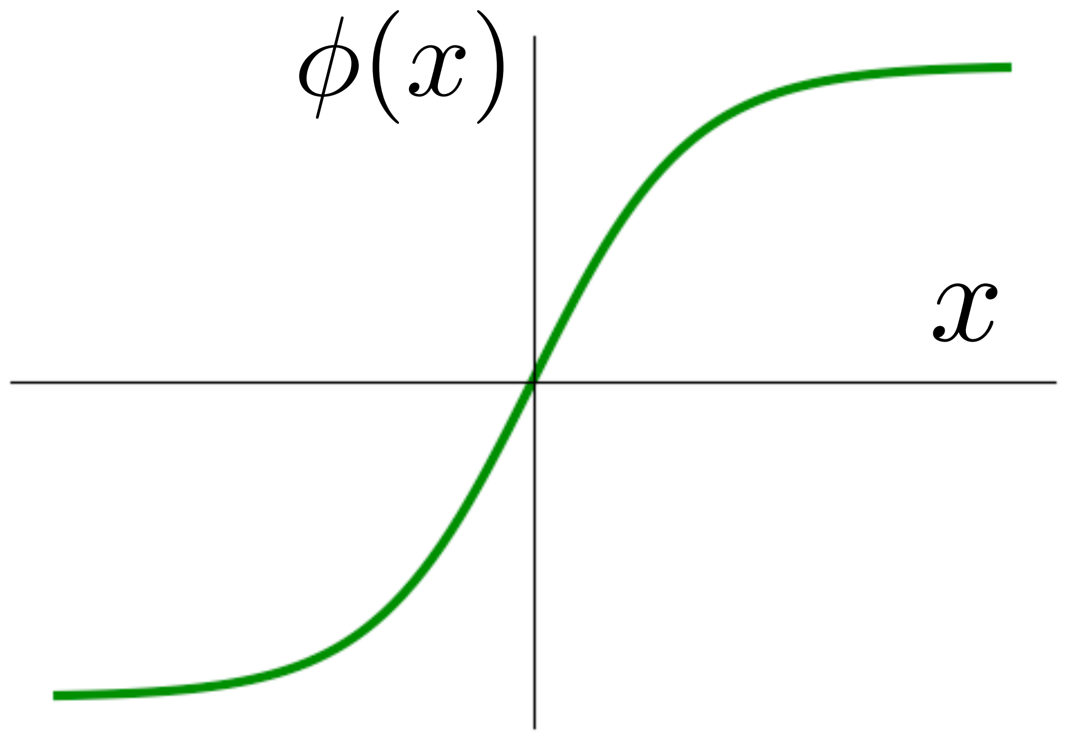


“Abstract neurons”



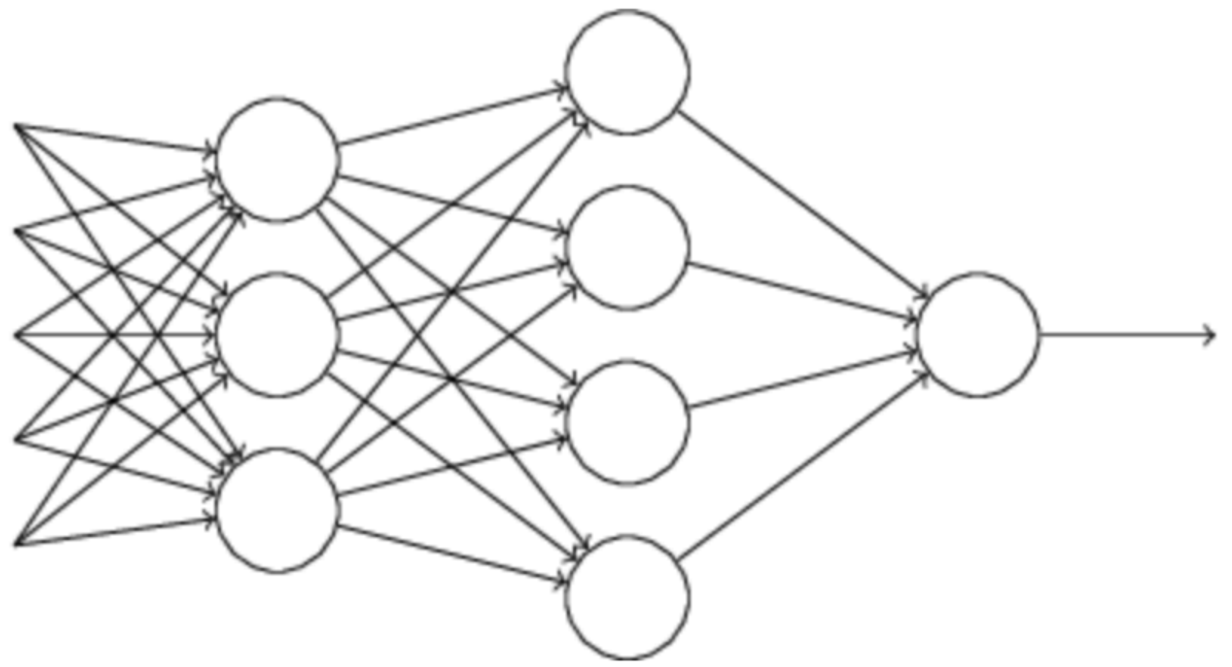
$$\tau \dot{x} = -x + \sum_i w_i r_i$$

$$r = \phi(x)$$



Feedforward architecture

inputs

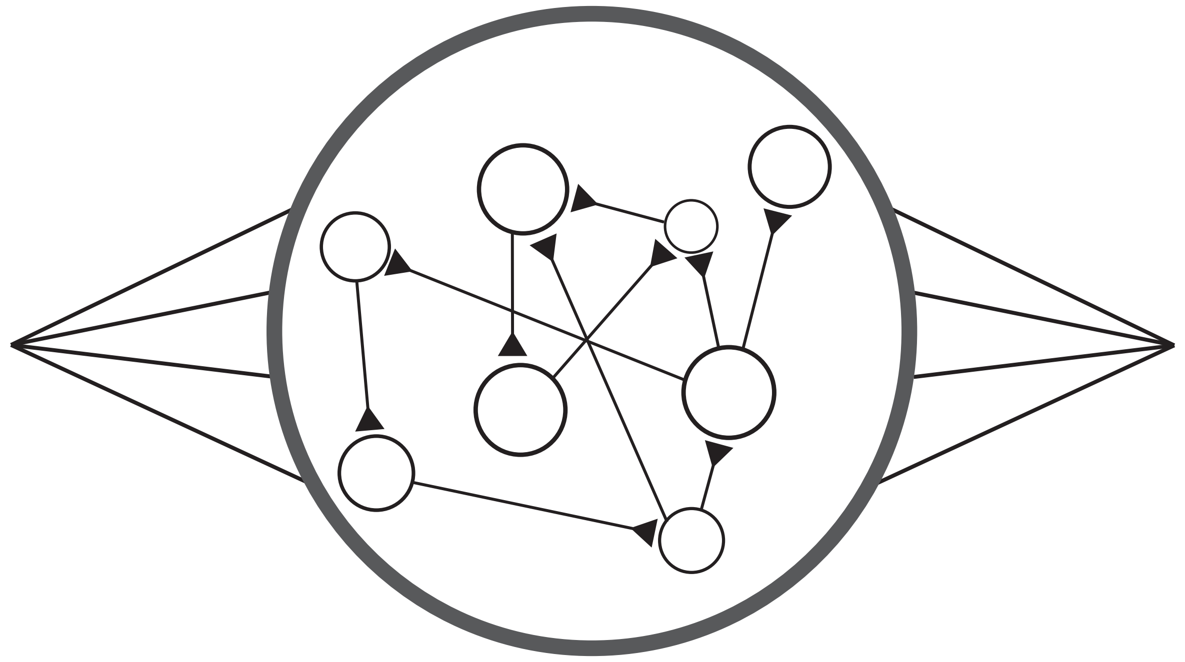


outputs

hidden layers

Recurrent architecture

inputs



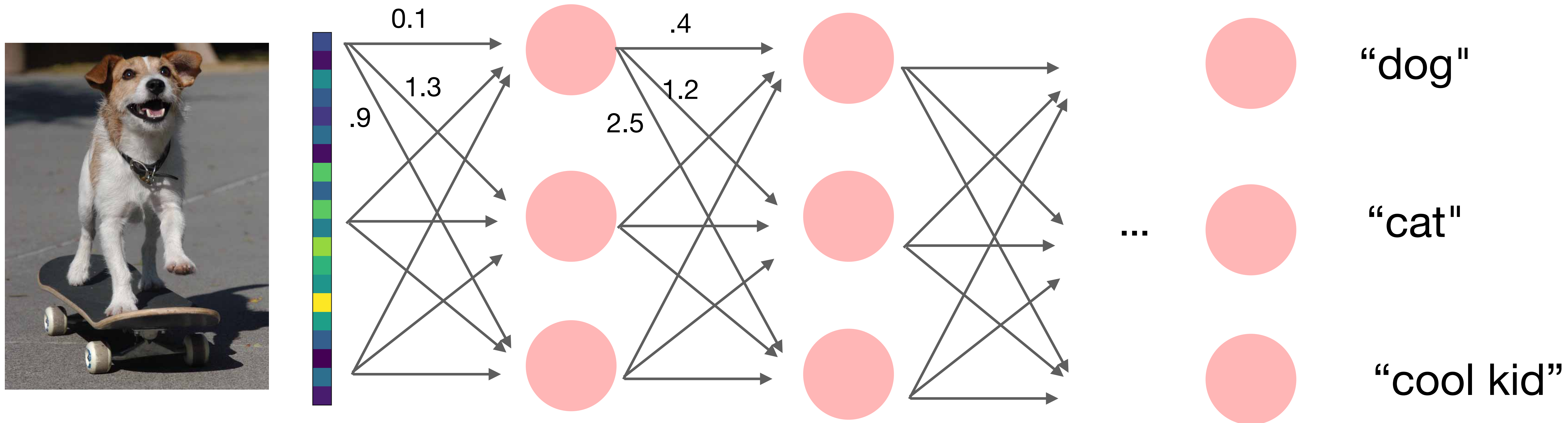
outputs

internal state

Learning tasks in feedforward networks

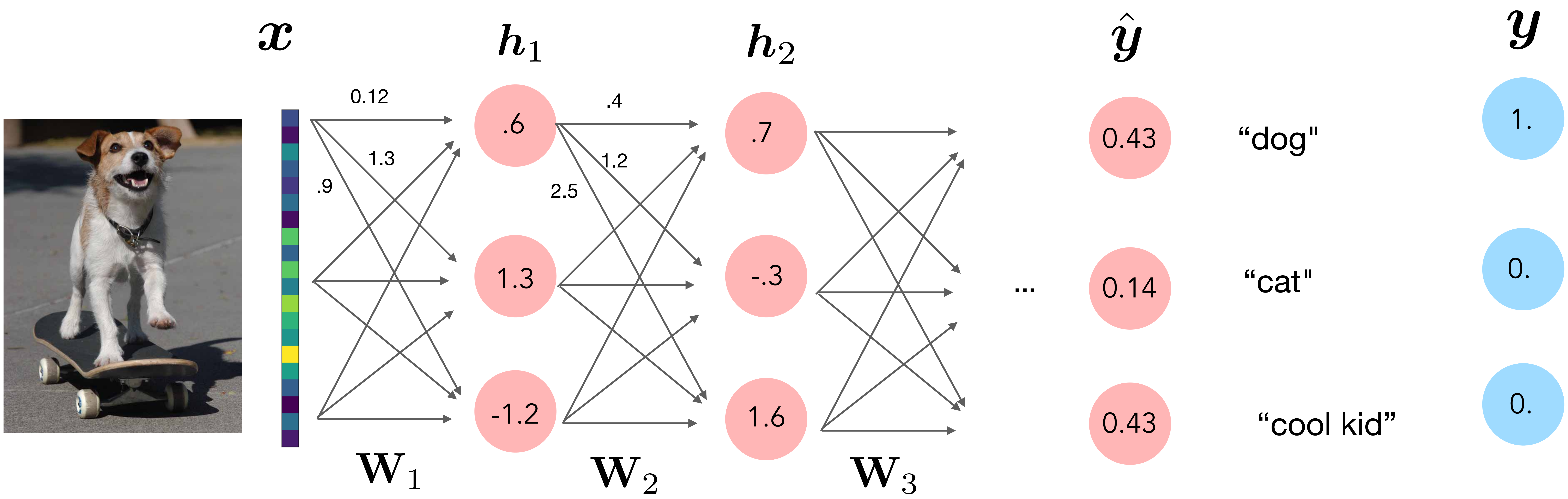
Learning representations by back-propagating errors (1986)

David E. Rumelhart*, Geoffrey E. Hinton†
& Ronald J. Williams*



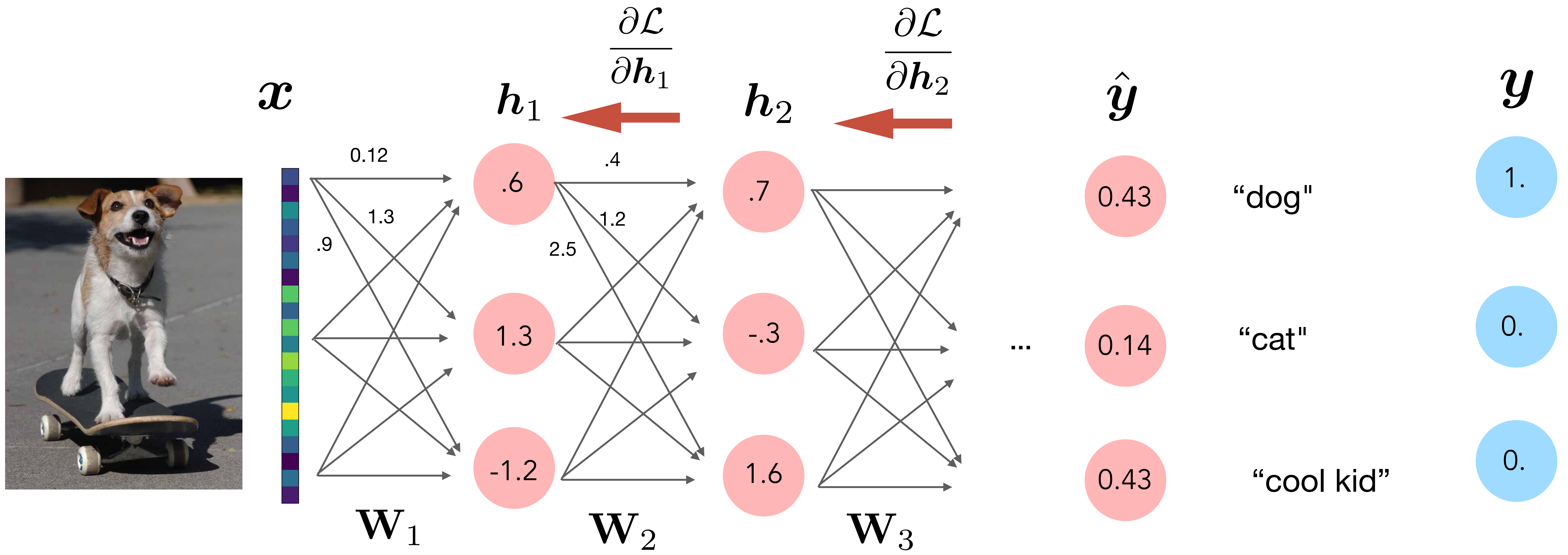
Goal: learn mapping vector -> vector

Learning tasks in feedforward networks



Error or "loss": $\mathcal{L} = (y - \hat{y})^2$

Backpropagation algorithm (feedforward nets)



Error or "loss": $\mathcal{L} = (y - \hat{y})^2$

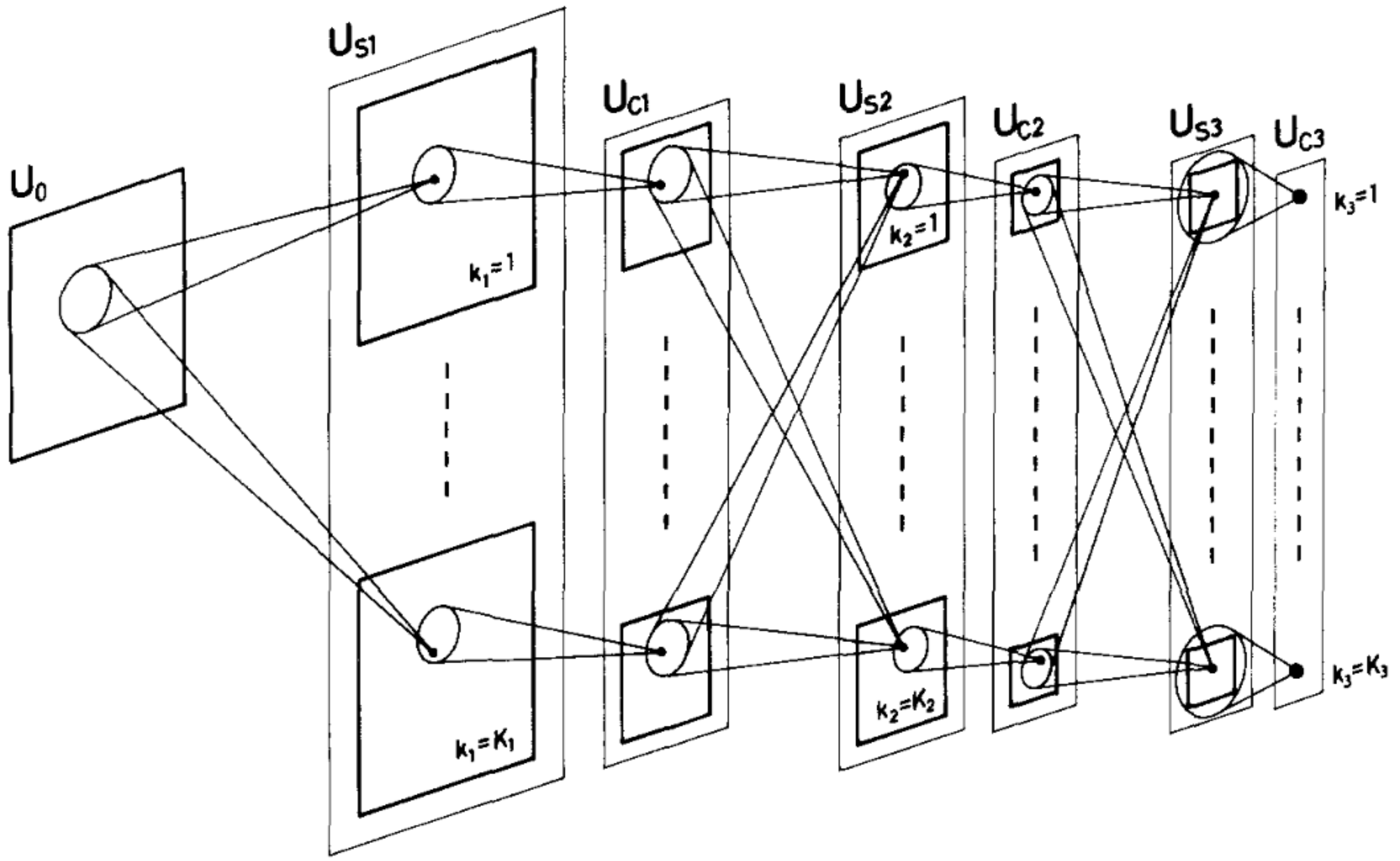
Gradient descent update: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$

Diverse architectures… (eg convolutional)

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

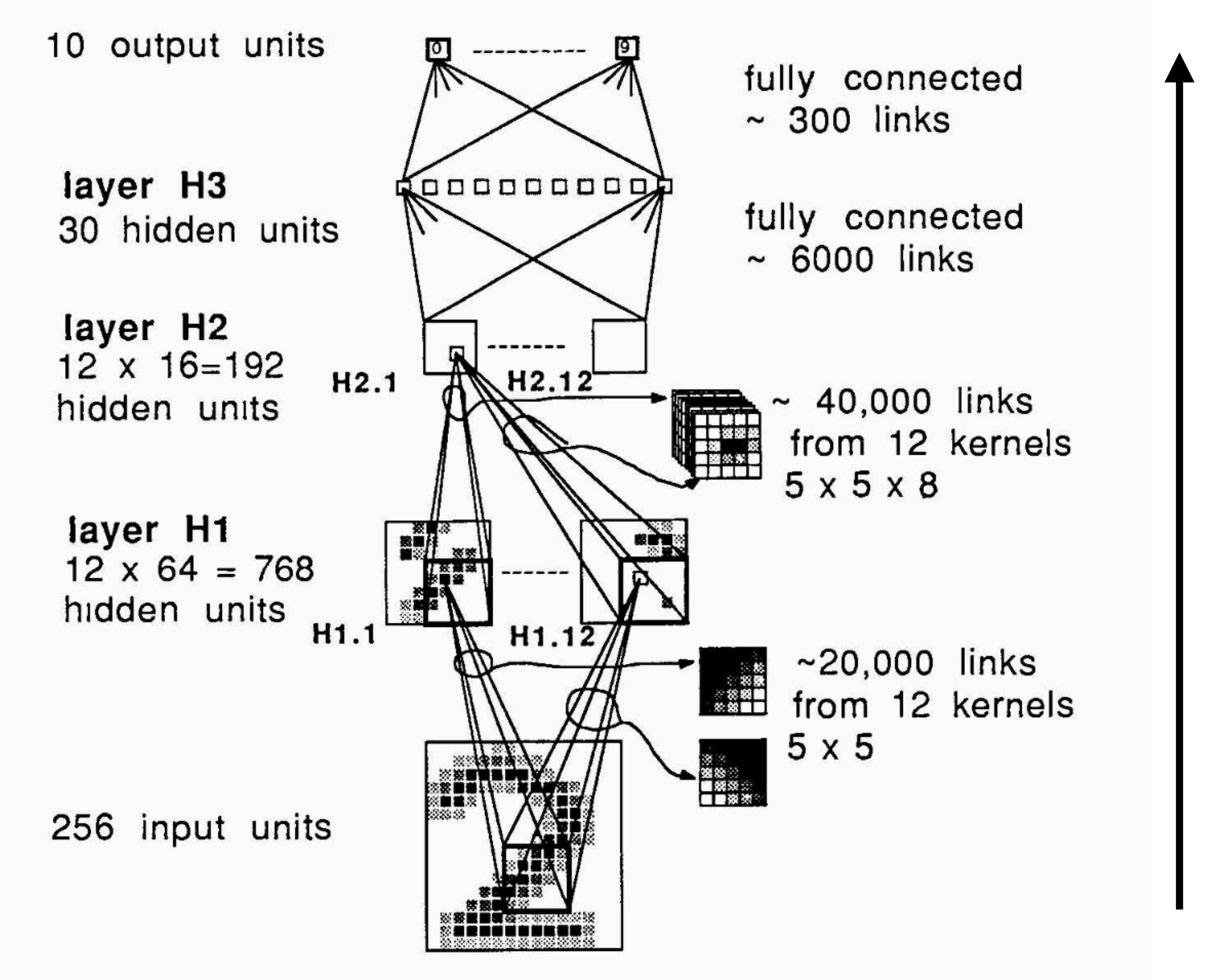
Kunihiko Fukushima
NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

(1980)



Backpropagation Applied to Handwritten Zip Code Recognition

LeCun et al., (1989)



And crazy results since 2012

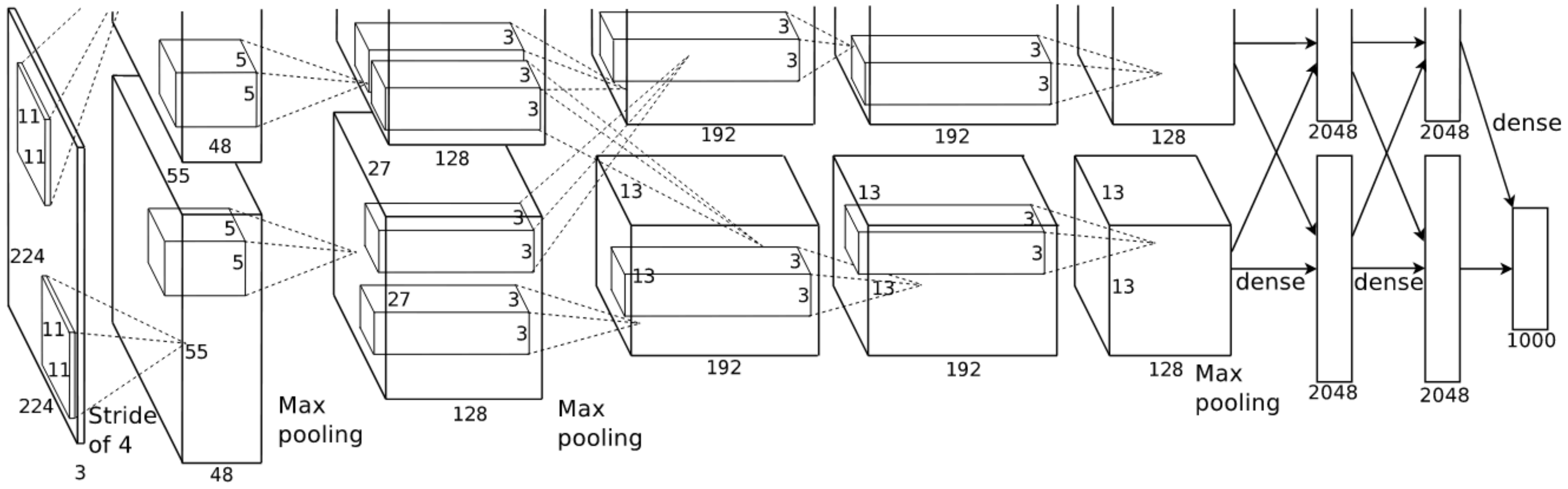
ImageNet Classification with Deep Convolutional Neural Networks

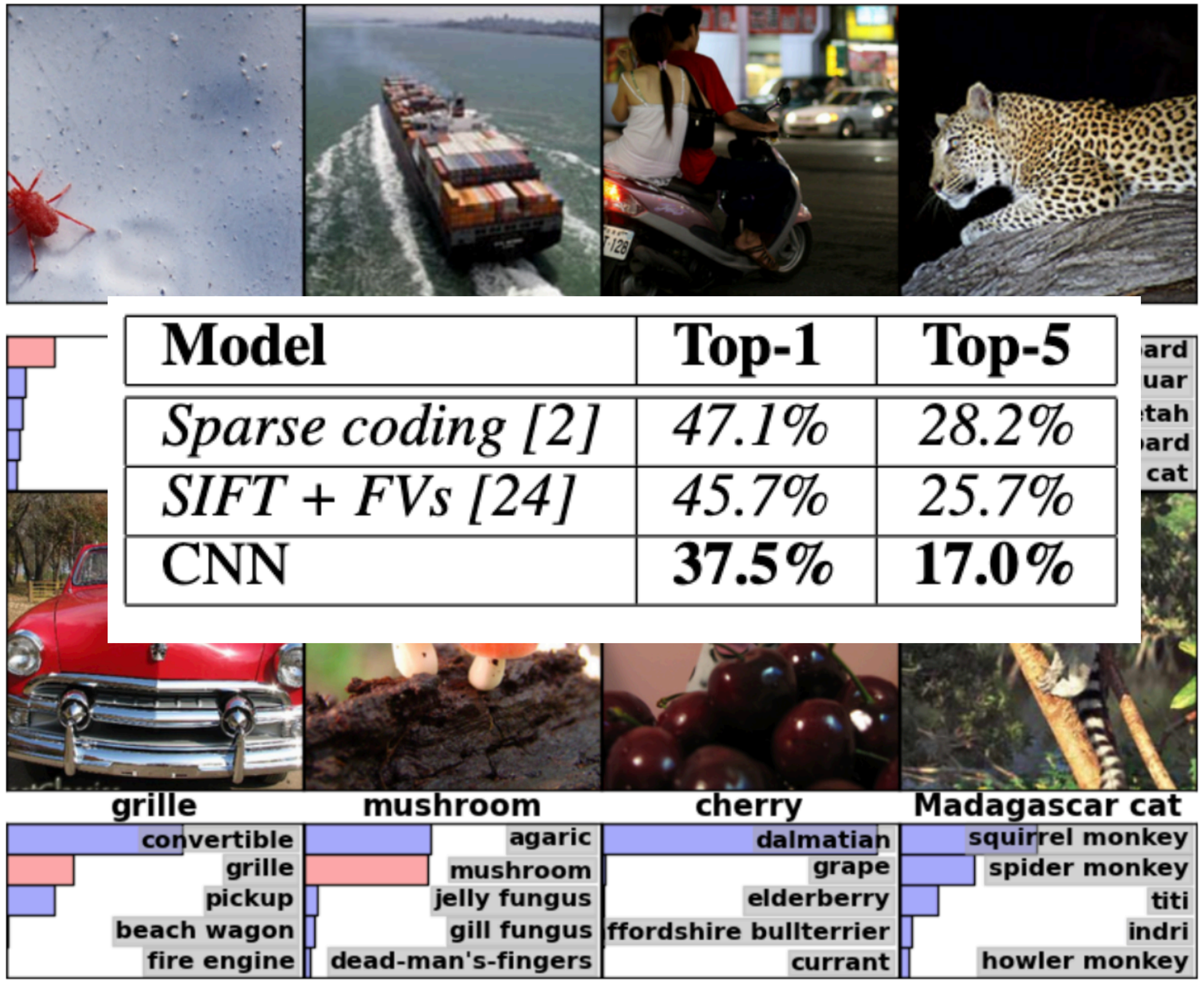
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

(2012)





Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

What kind of computations is this?

Data $\boldsymbol{x} \in \mathbb{R}^n$ $\xrightarrow{\text{network}}$ $\hat{\boldsymbol{z}} \approx \boldsymbol{z} \in \mathbb{R}^m$

A feedforward network learns a *function*.

Remaining problem: variable-length inputs

So far we consider inputs that can be mapped to vectors in $\mathbb{R}^n$

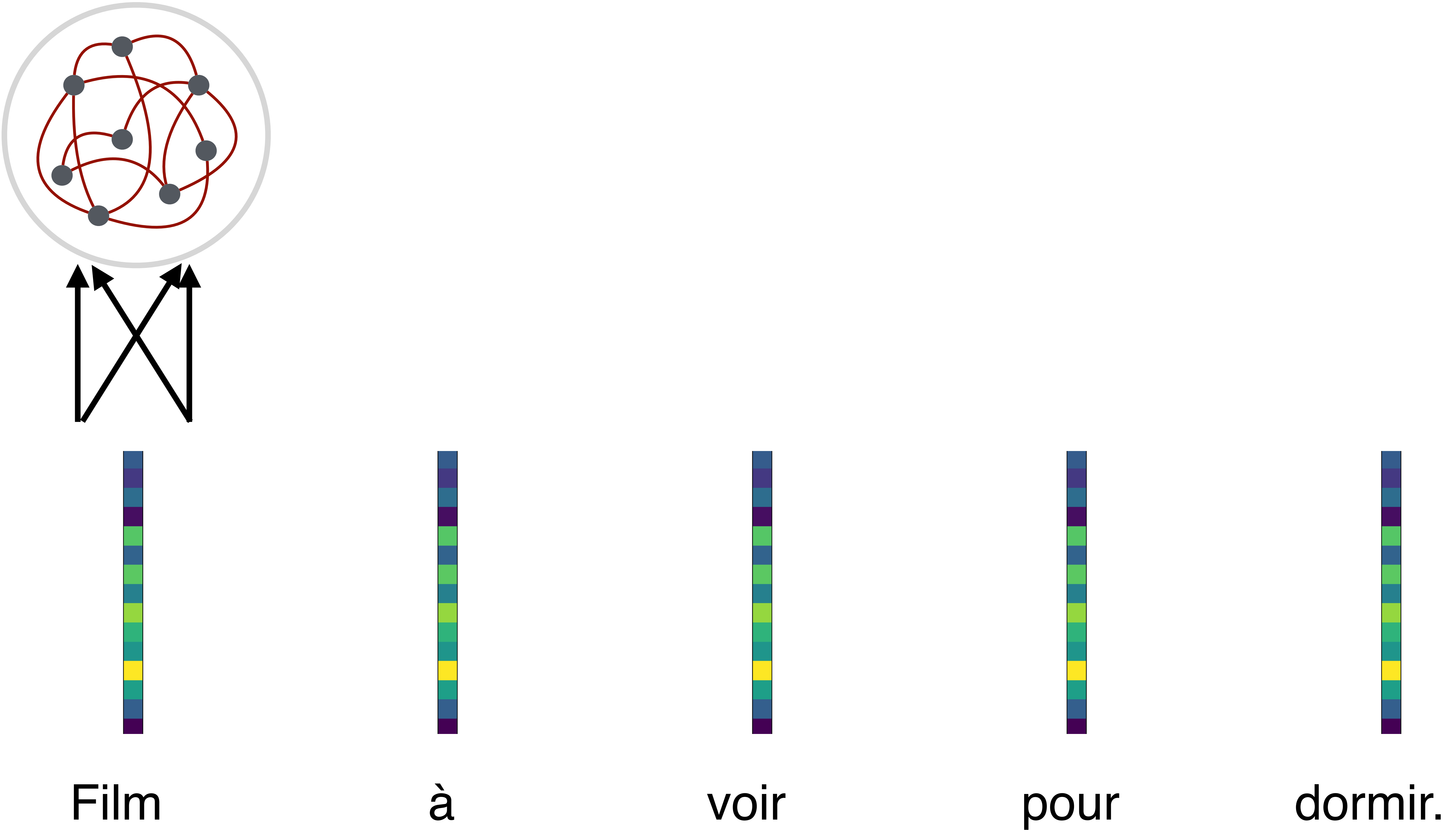
What about:

- text
- sound
- video
- time series…

Learning over sequences with RNNs

Objective: learn a mapping sequence -> vector

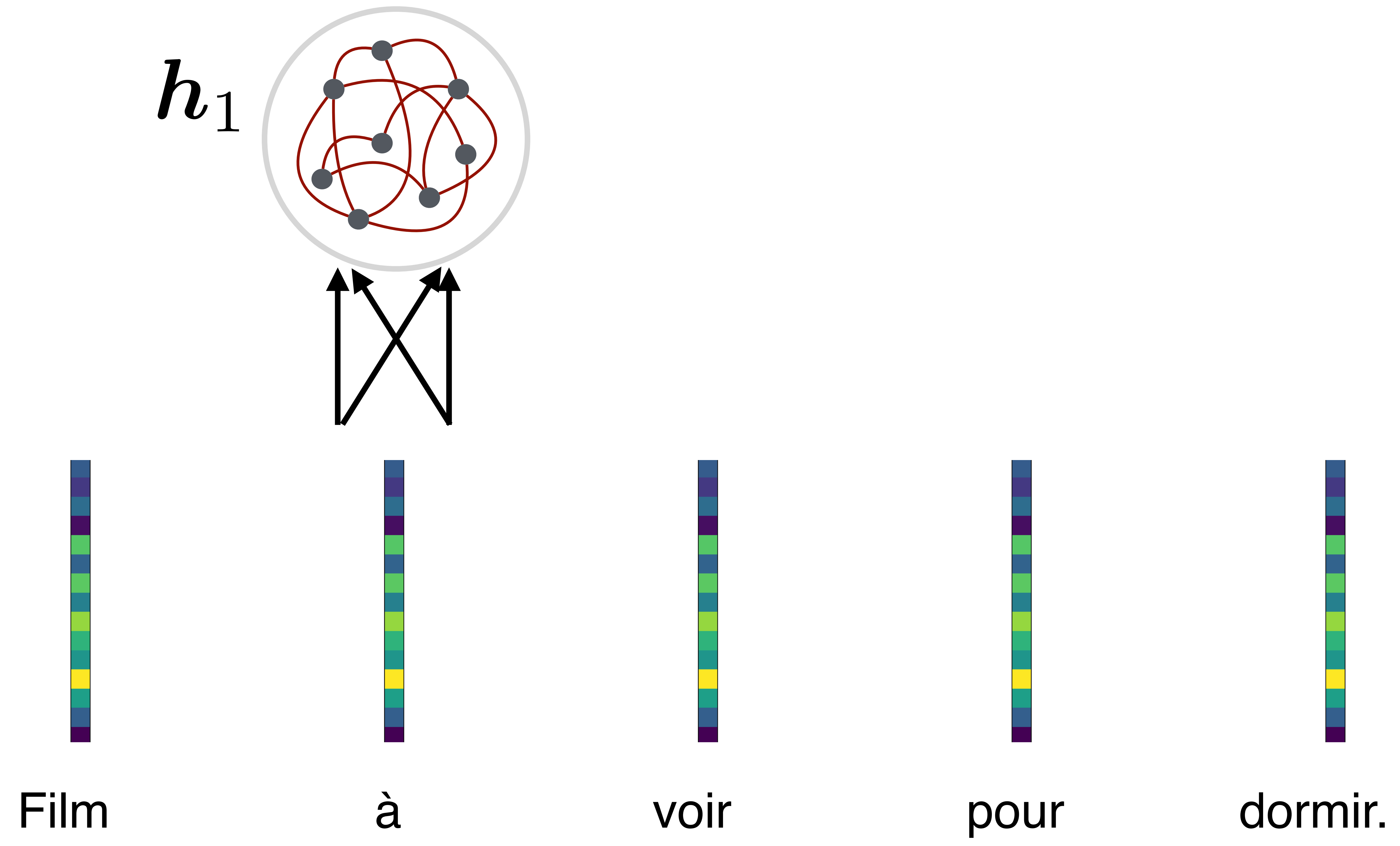
Example: sentiment analysis



Recurrent neural networks (RNNs)

Objective: learn a mapping sequence -> vector

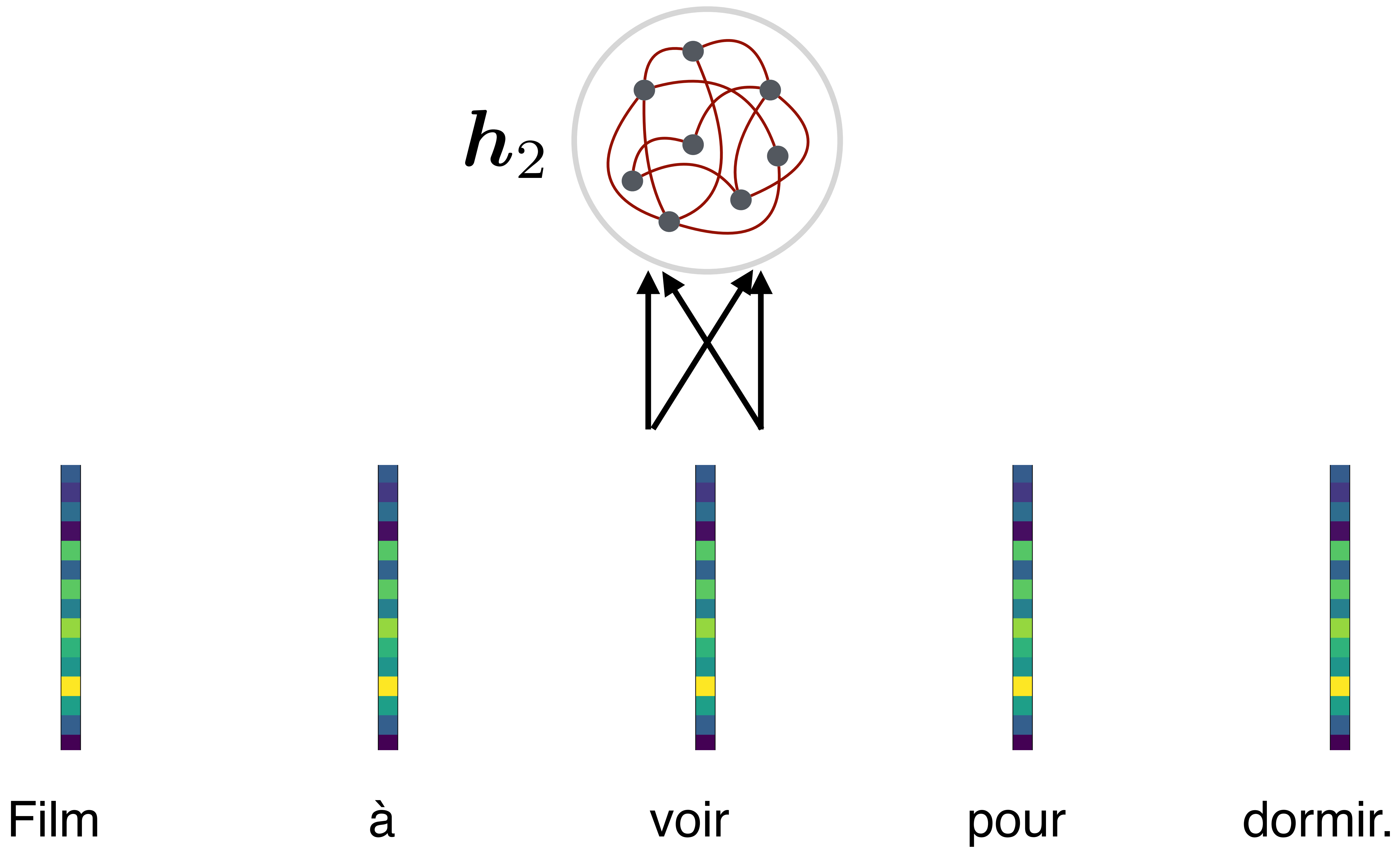
Example: sentiment analysis



Recurrent neural networks (RNNs)

Objective: learn a mapping sequence -> vector

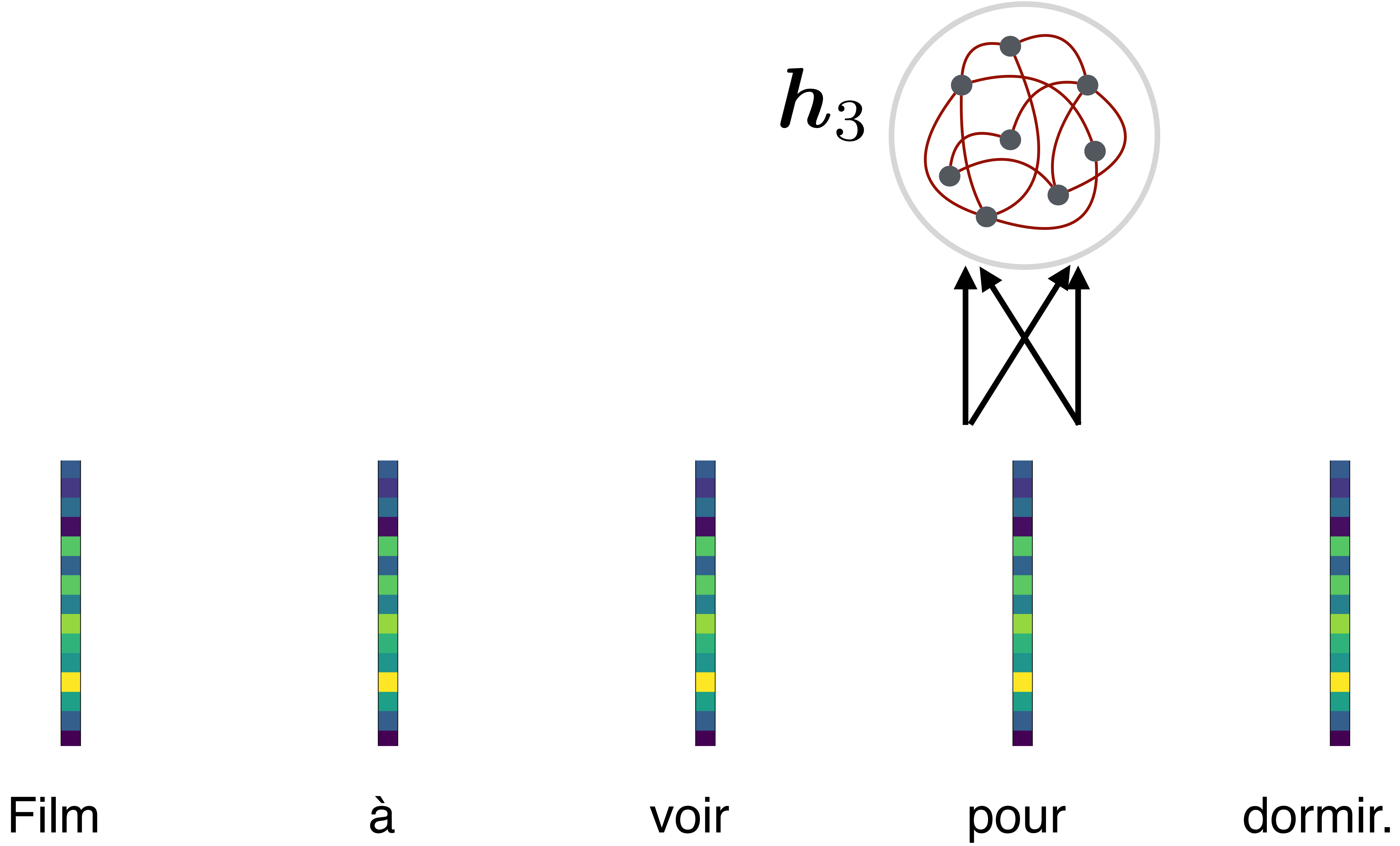
Example: sentiment analysis



Recurrent neural networks (RNNs)

Objective: learn a mapping sequence -> vector

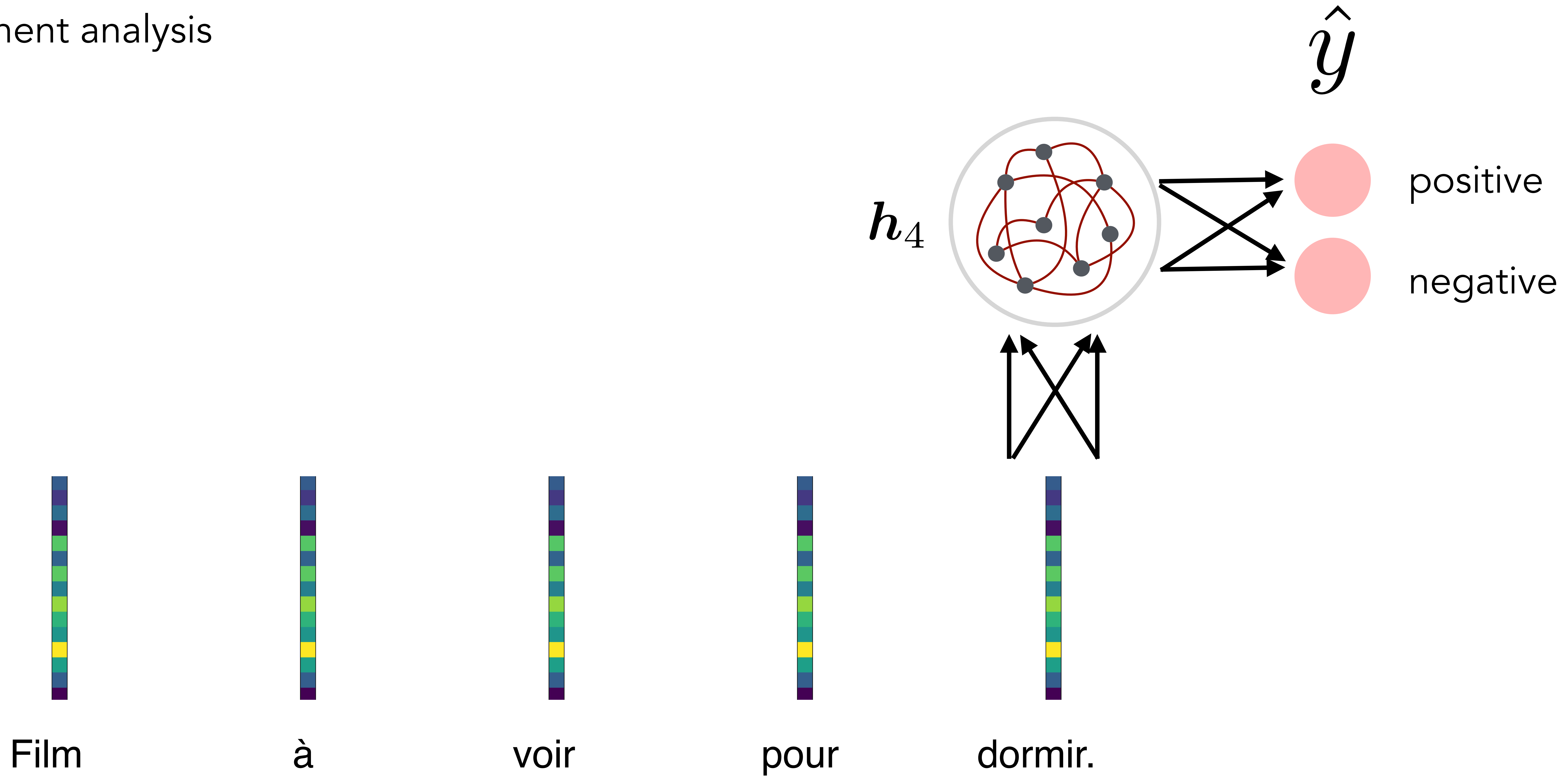
Example: sentiment analysis



Recurrent neural networks (RNNs)

Objective: learn a mapping sequence -> vector

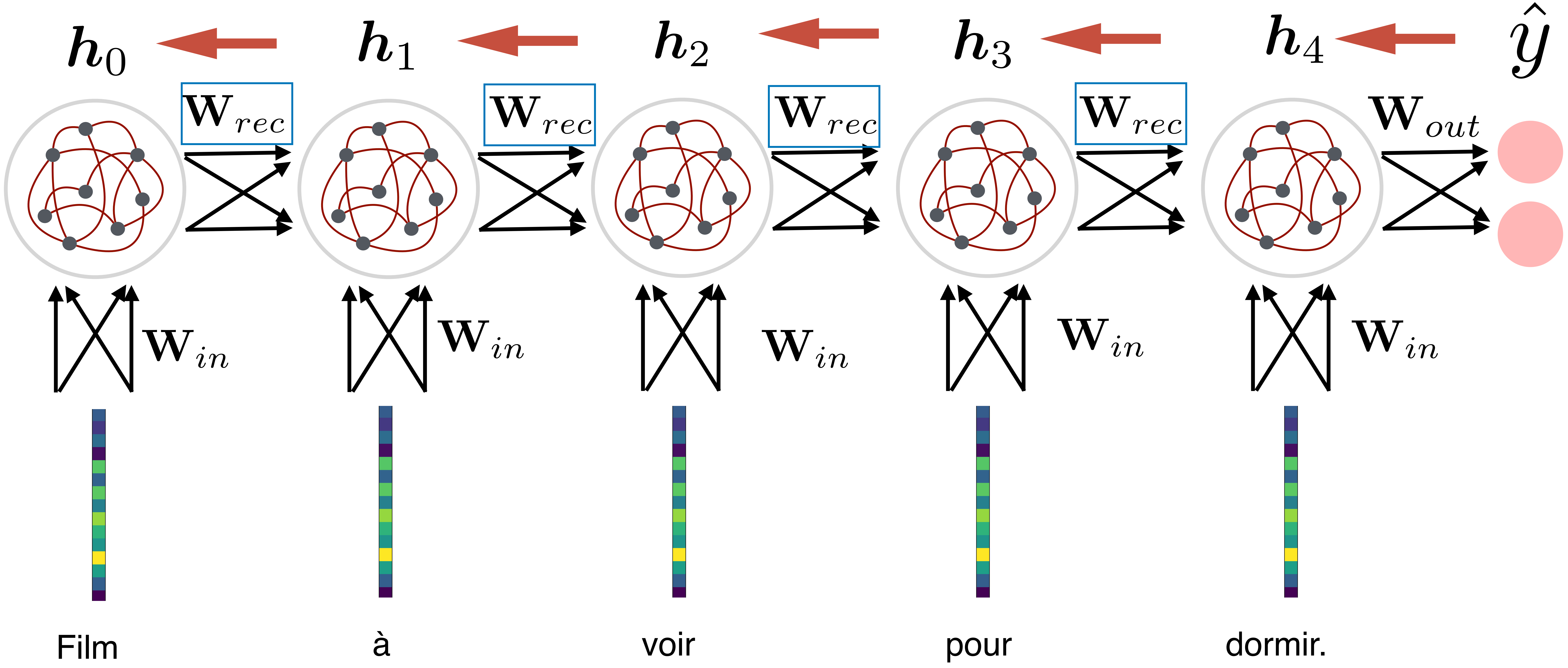
Example: sentiment analysis



Backpropagation through time

Williams & Zipser (1989), and others…

Unrolled computation graph



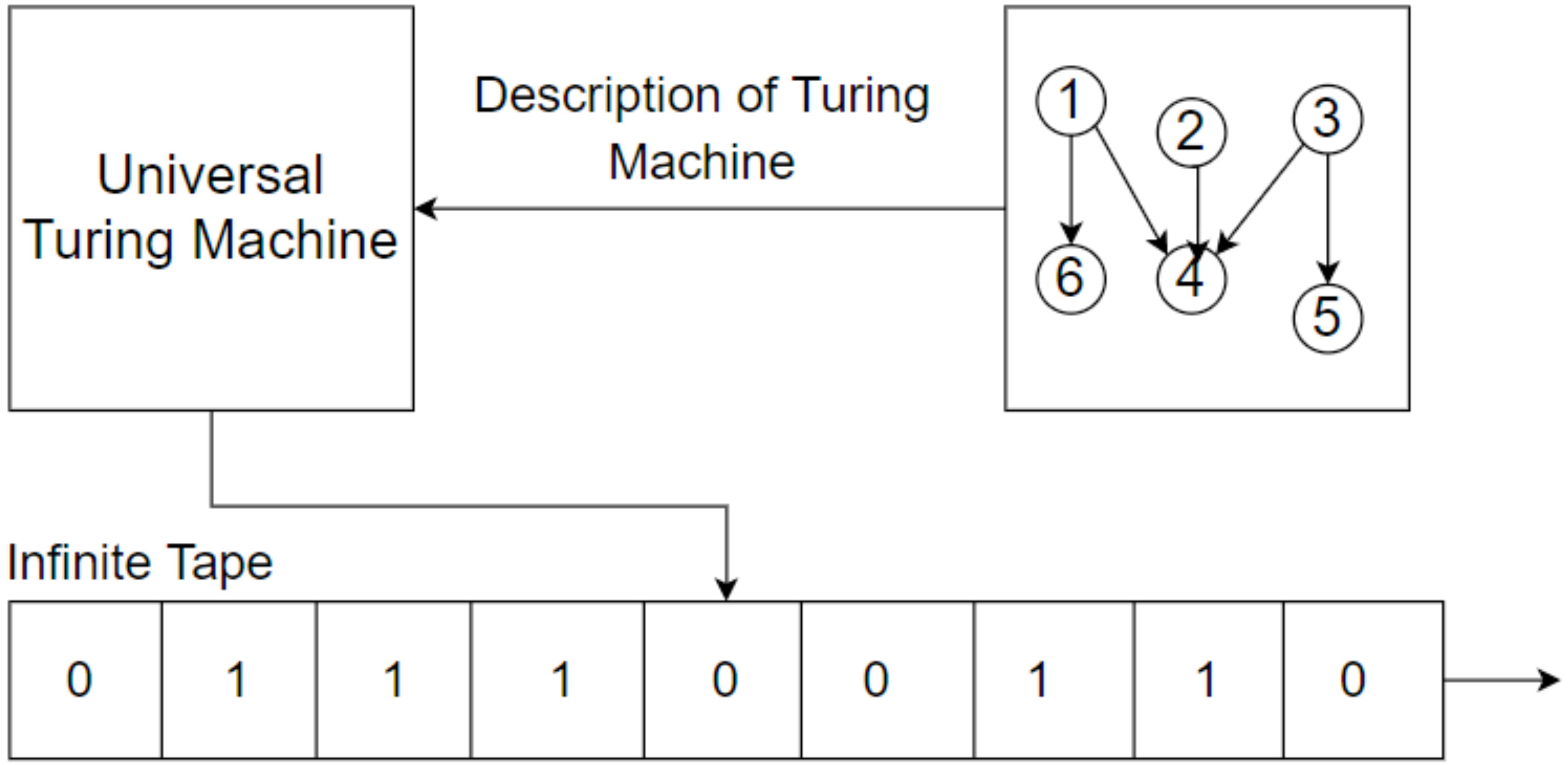
And now, what computations are these?

Data $(x_1, \dots, x_T)$ —network→ $(\hat{z}_1, \dots, \hat{z}_{\tilde{T}}) \approx (z_1, \dots, z_{T'})$

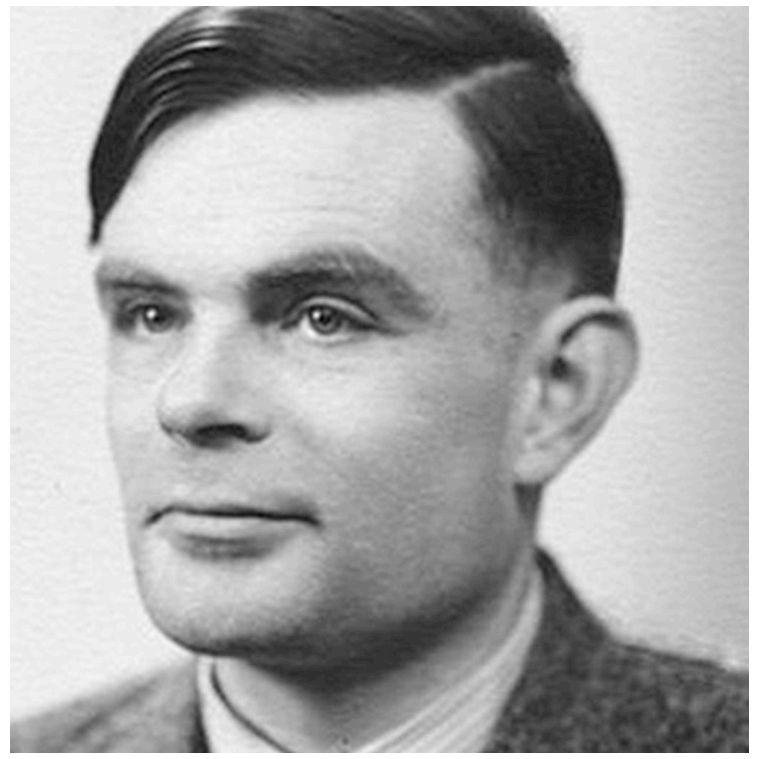
A recurrent network learns to produce *adaptive behavior*.

What is computation anyway?

Turing machine



Characterized by:
- an internal state at each time point
- descriptions of transitions between states
- a set of final states



Alan Turing

Alonzo Church

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

What is computation anyway?

A system is **Turing-complete** if it can imitate a Turing machine.

Church-Turing thesis: the set of computable problems is exactly the set of algorithms that can be implemented in a Turing machine.

$$\text{Computations} \iff \text{Algorithms} \iff \text{Turing machines}$$

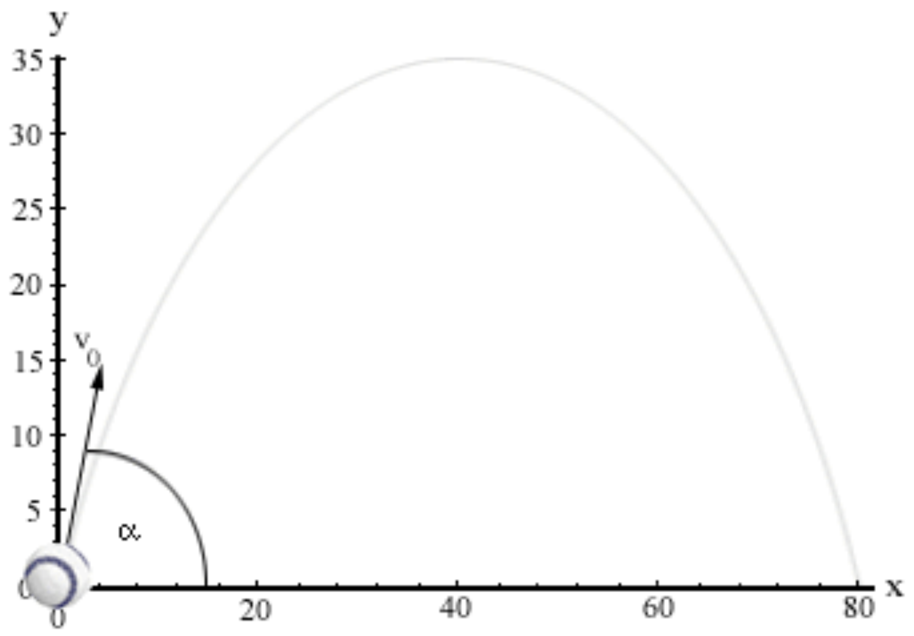
Notion of dynamical systems

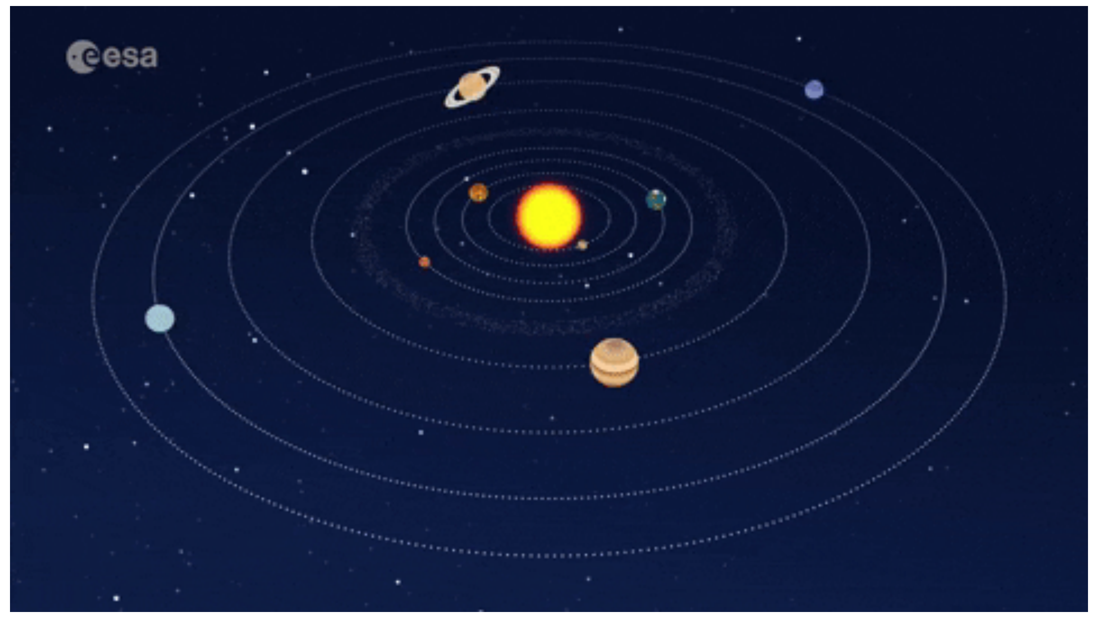
	Discrete	**Continuous**
Two formulations:	$x_{t+1} = f(x_t, u_t)$	$\frac{dx}{dt} = f(x(t), u(t))$

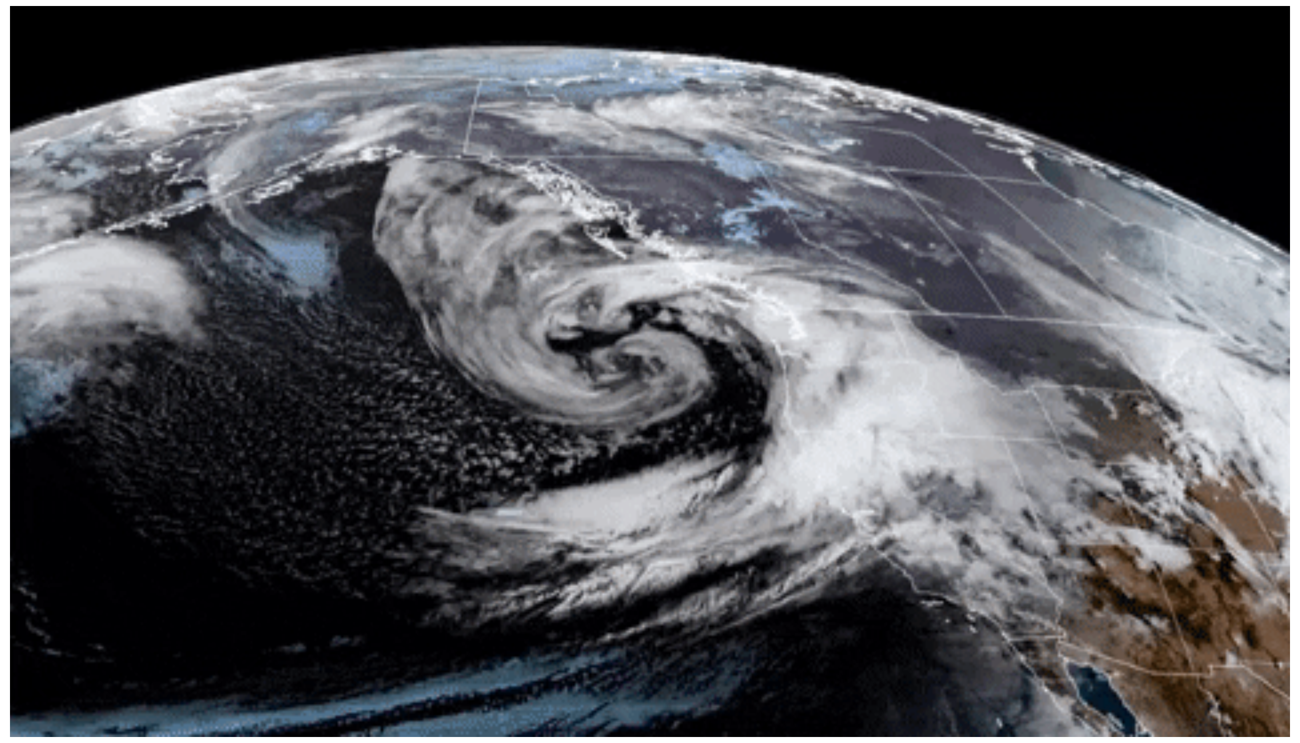
RNNs and Turing machines are dynamical systems! (Brains too?)

Other examples of dynamical systems:

- a ball in a gravitational field
- planets
- climate







Notion of dynamical systems

"Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective positions of the beings which compose it, if moreover this intelligence were vast enough to submit these data to analysis, it would embrace in the same formula both the movements of the largest bodies in the universe and those of the lightest atom; to it nothing would be uncertain, and the future as the past would be present to its eye."

Pierre-Simon de Laplace

Universal approximation theorem for RNNs

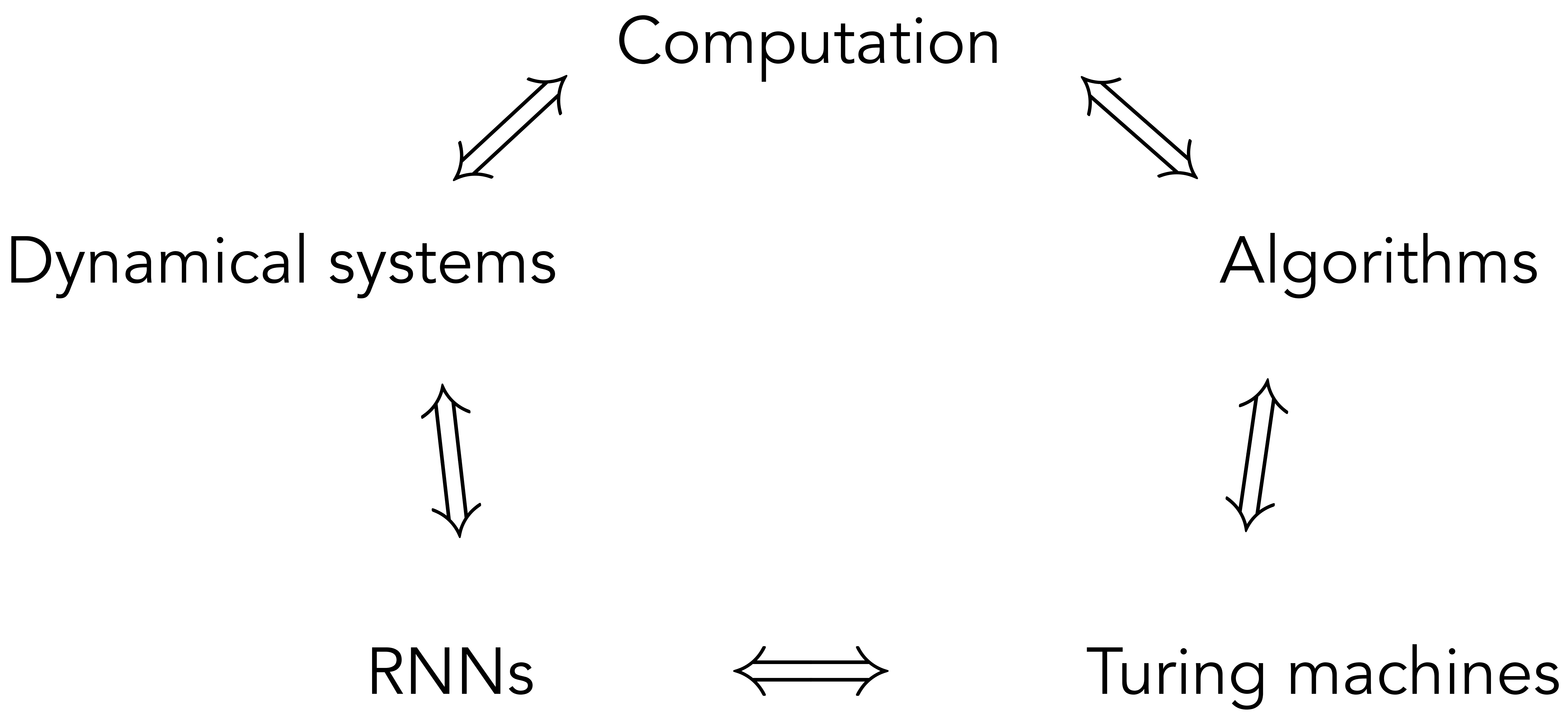
Provided sufficient hidden neurons and the correct weights, an RNN can approximate with arbitrary precision any dynamical system on $\mathbb{R}^n$ (of compact support). Equivalently, an RNN can approximate with arbitrary precision a finite automaton (ie. it is a universal computation machine).

Kenji Doya, ***Universality of fully-connected recurrent neural networks, 1993***

Abstract

It is shown from the universality of multi-layer neural networks that any discrete-time or continuous-time dynamical system can be approximated by discrete-time or continuous-time recurrent neural networks, respectively.

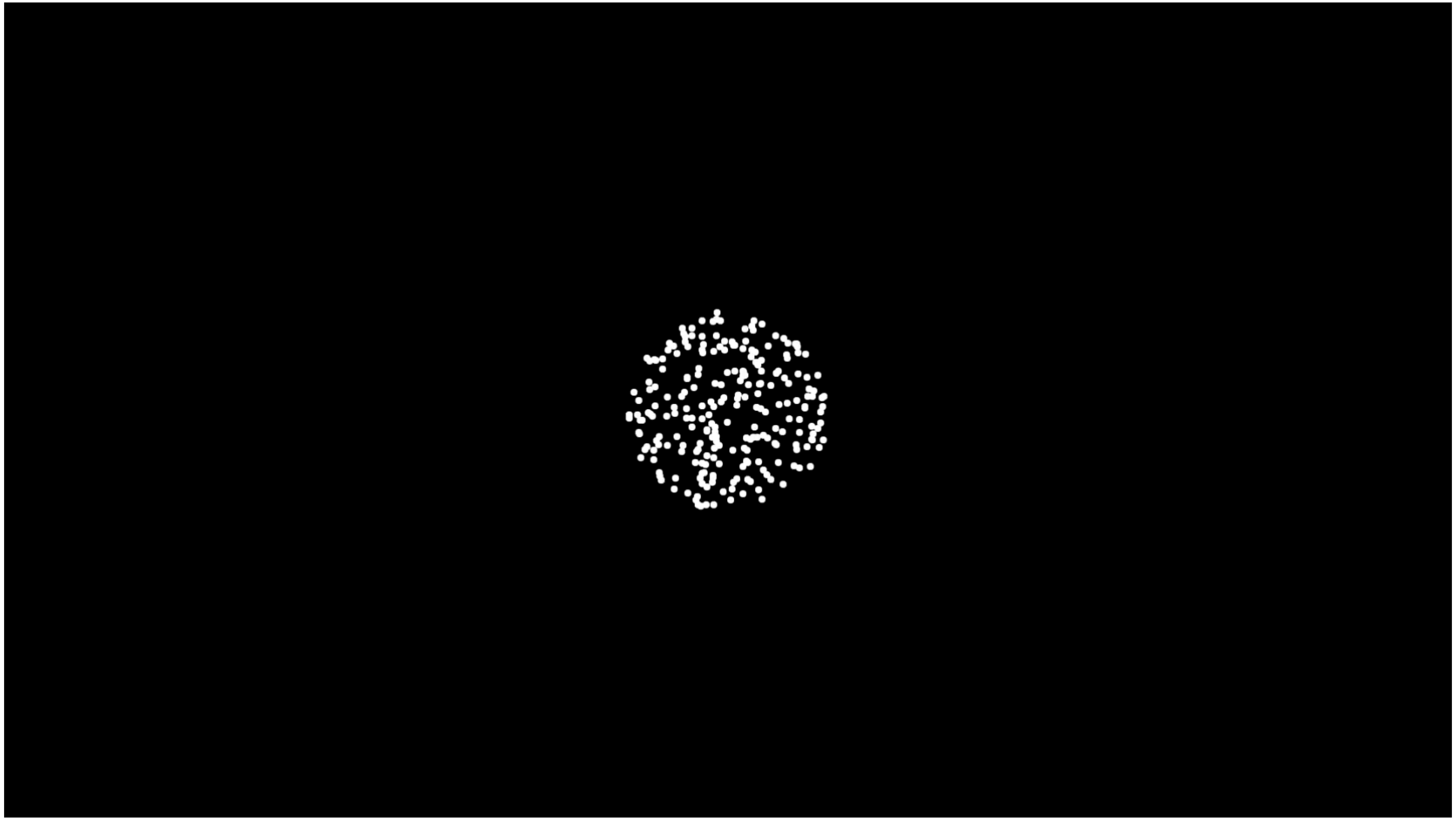
The 5 fingers of computation



Chapter II

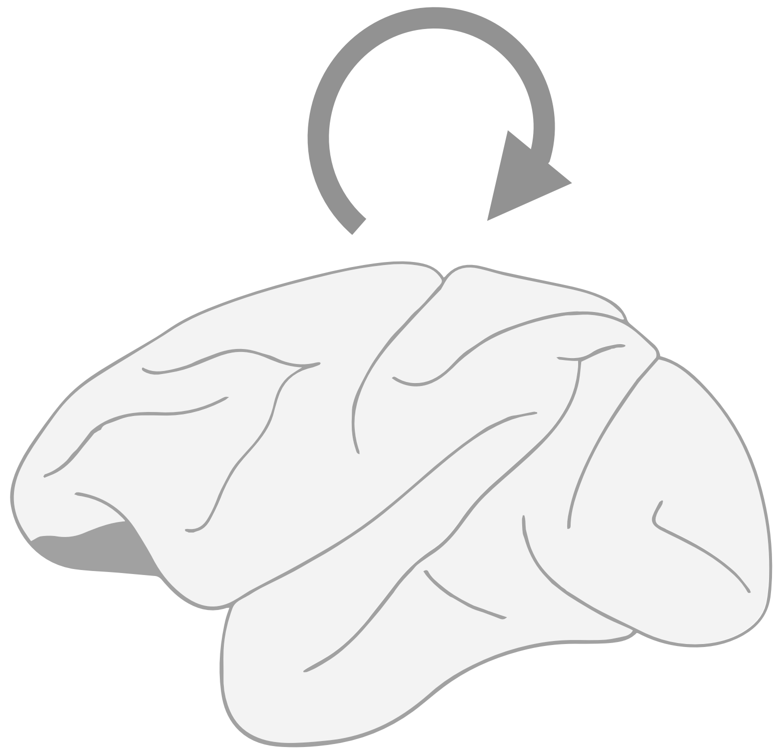
Back to neuroscience

Decision-making tasks



Reflected by computations in the brain

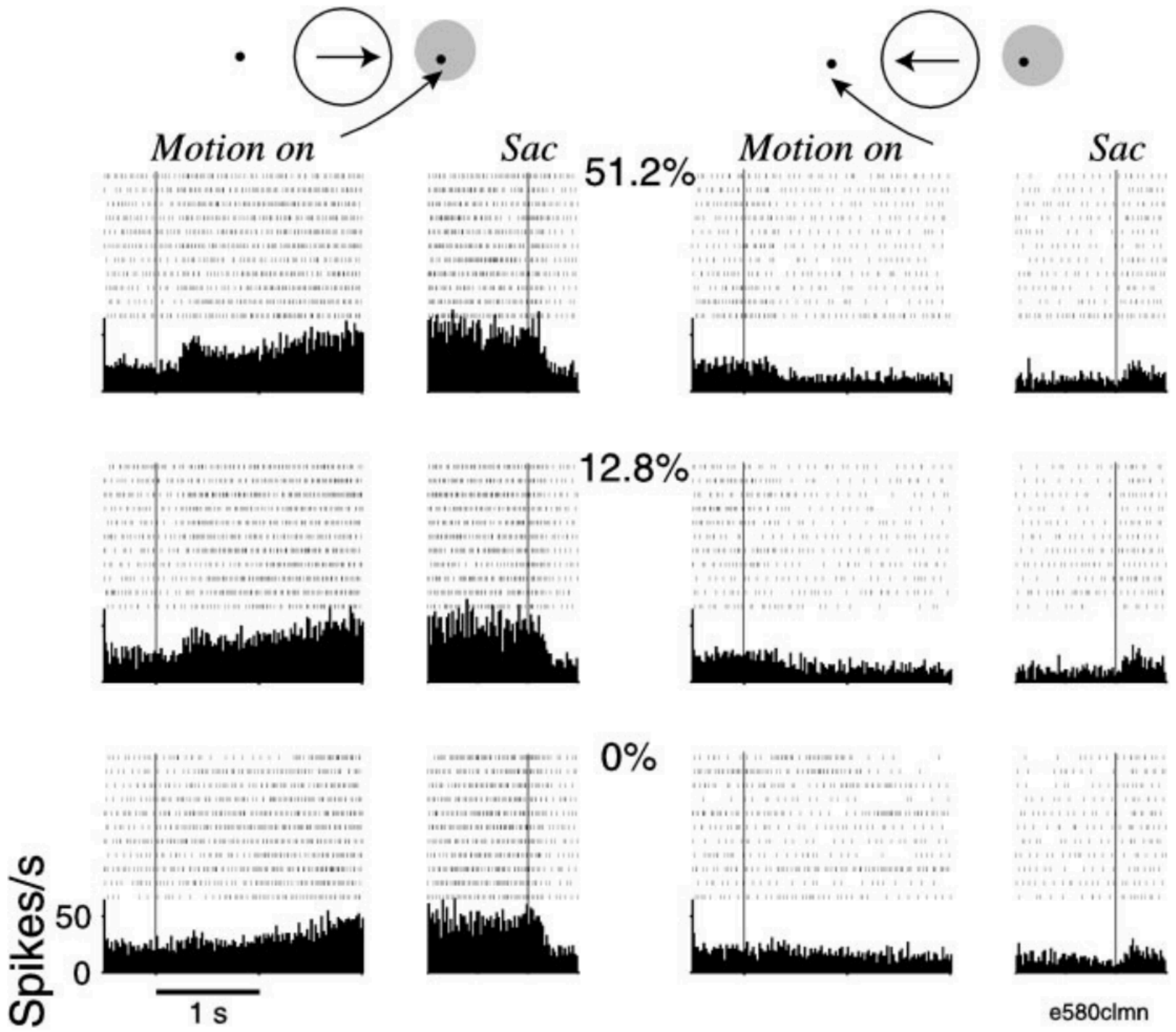
computing…



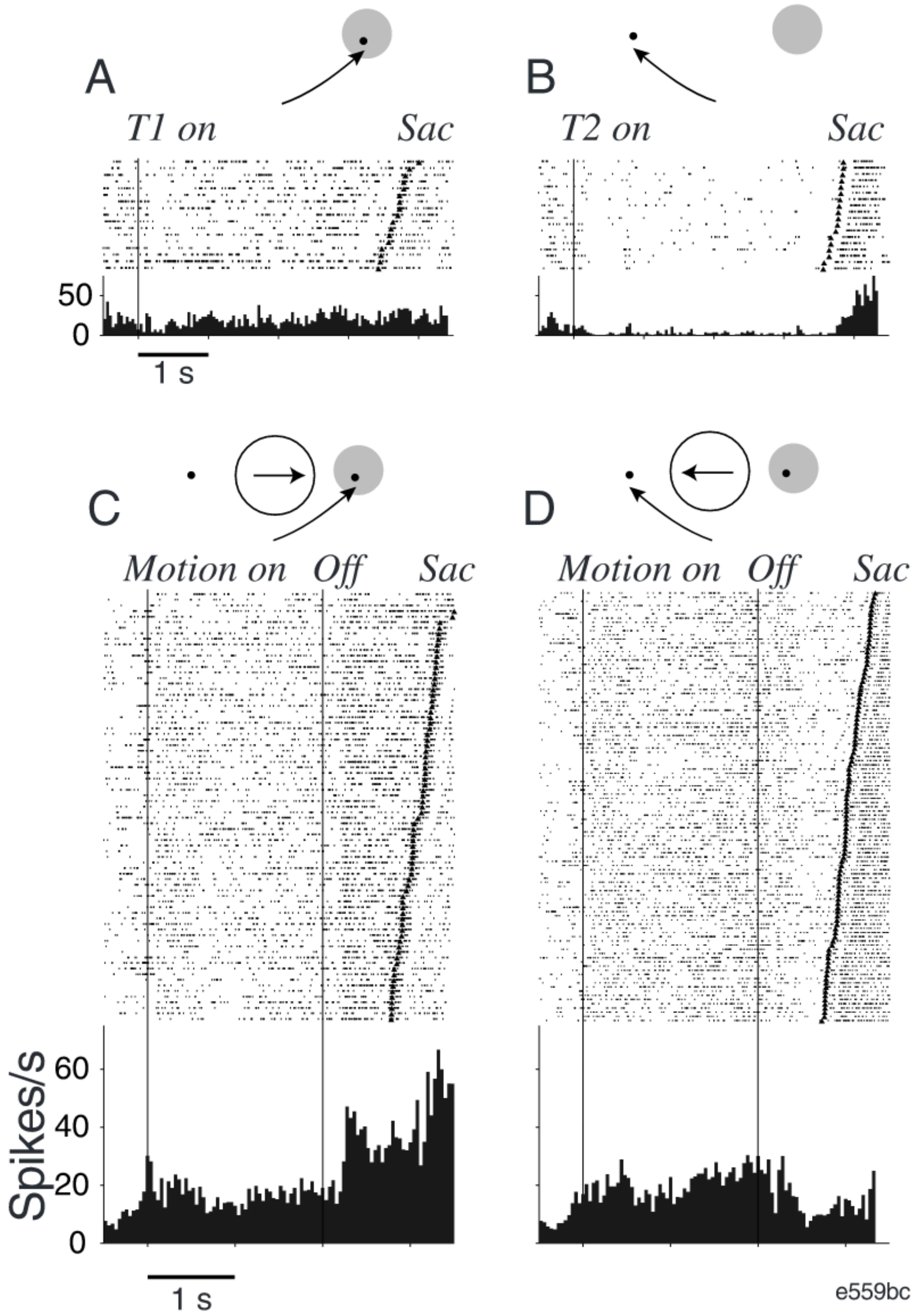
Sensory signals → Reactions/behavior

Neural correlates

Shadlen & Newsome, *Neural basis of a perceptual decision in the parietal cortex (are LIP) of the Rhesus monkey*, 2001



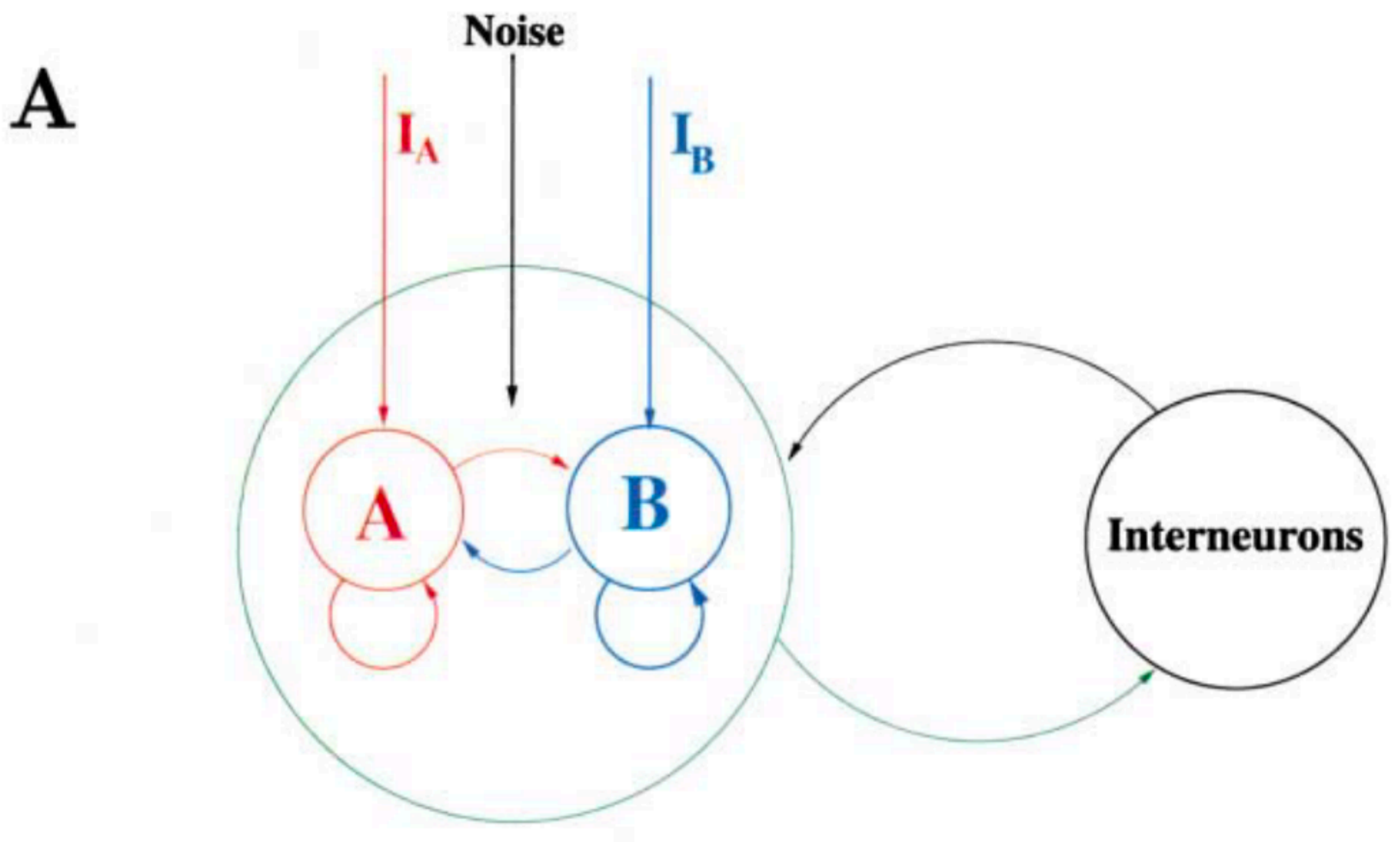
Neurons encoding evidence

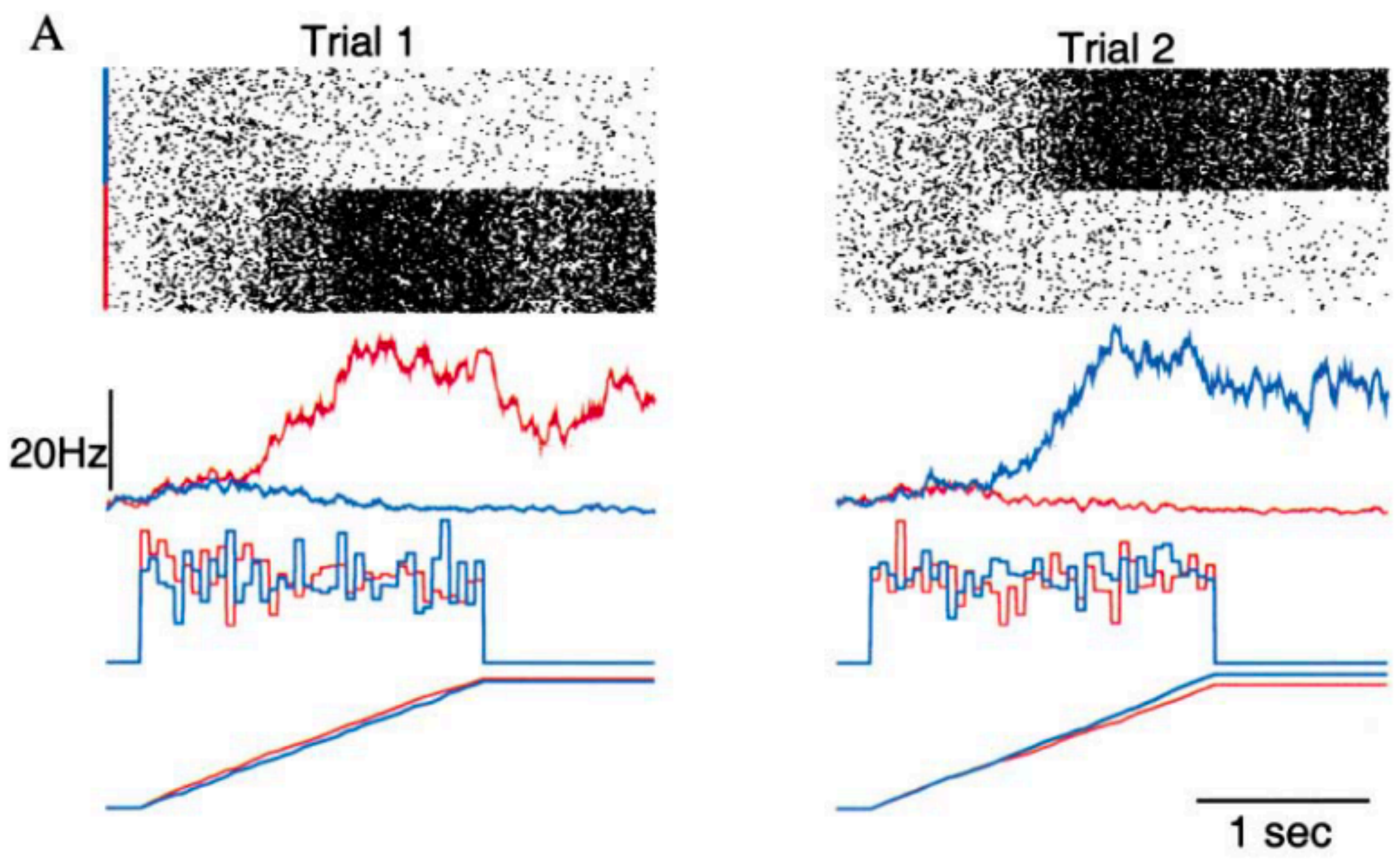


Neurons encoding decision

Phenomenological circuit models

XJ Wang, *Probabilistic decision making by slow reverberation in cortical circuits*, 2002



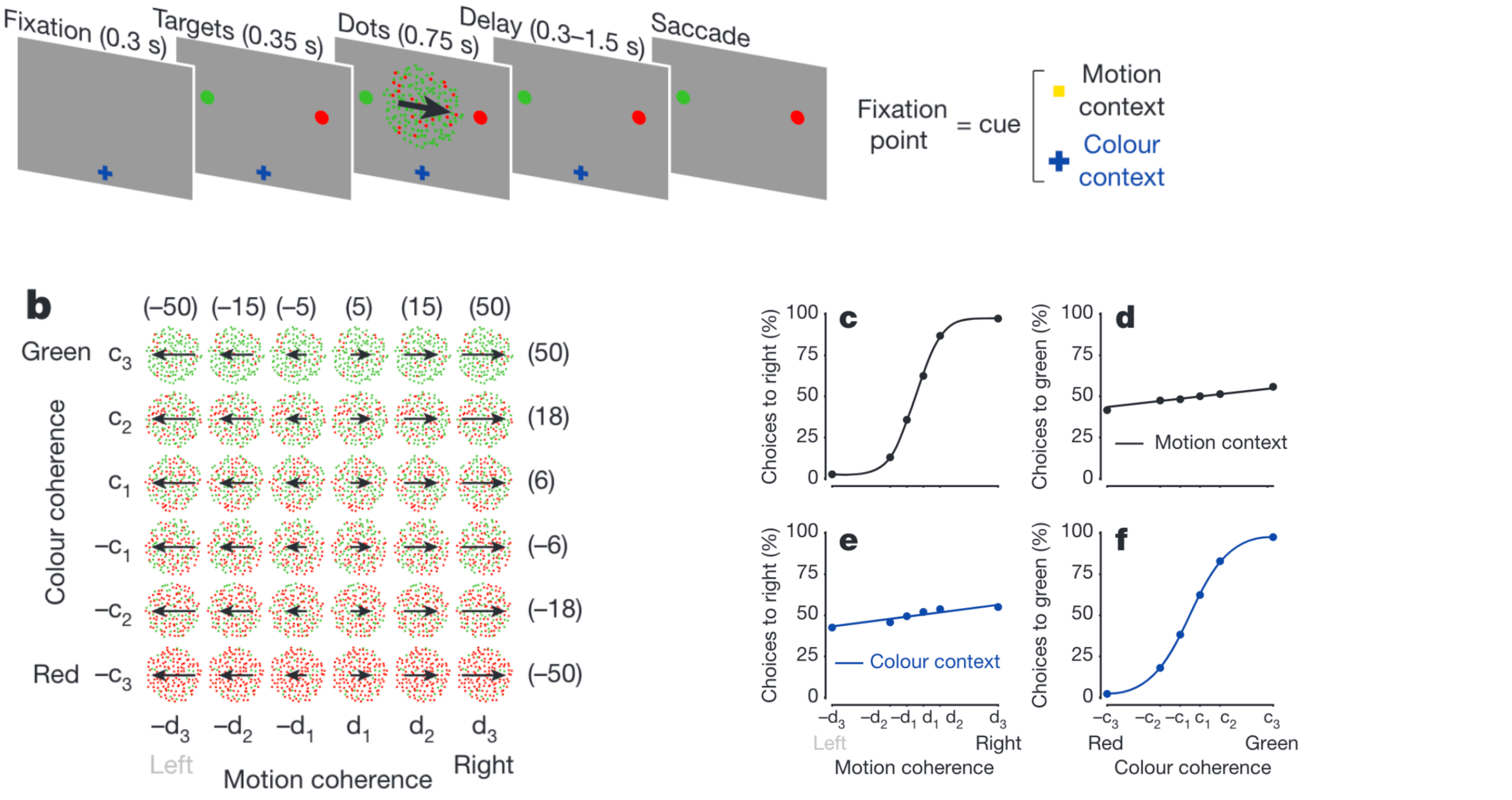


Handcrafted circuit

Parameters fitted to data

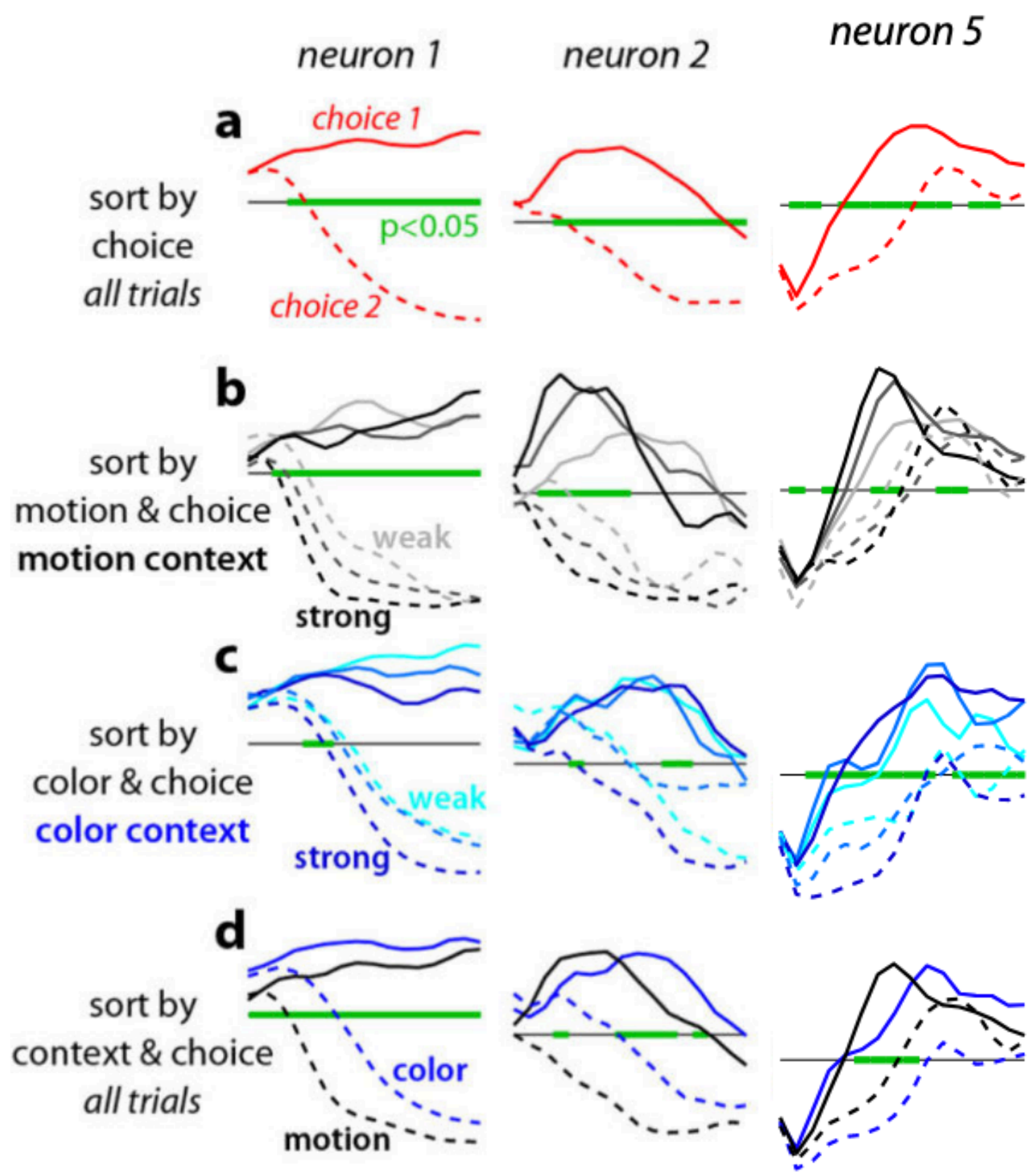
Context-dependent decision making

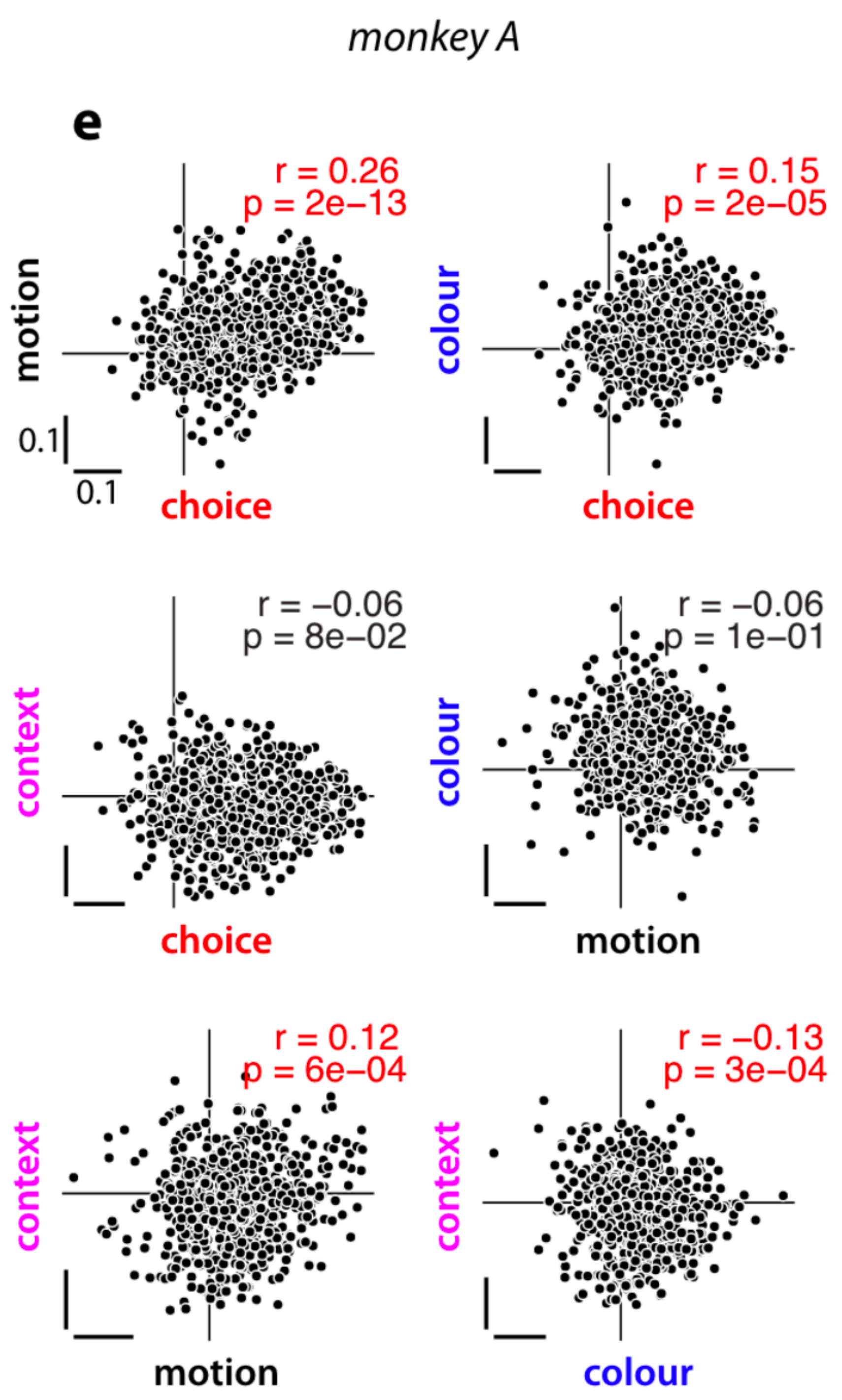
[Mante, Sussillo et al., *Context-dependent computations by recurrent dynamics in prefrontal cortex*, 2013]



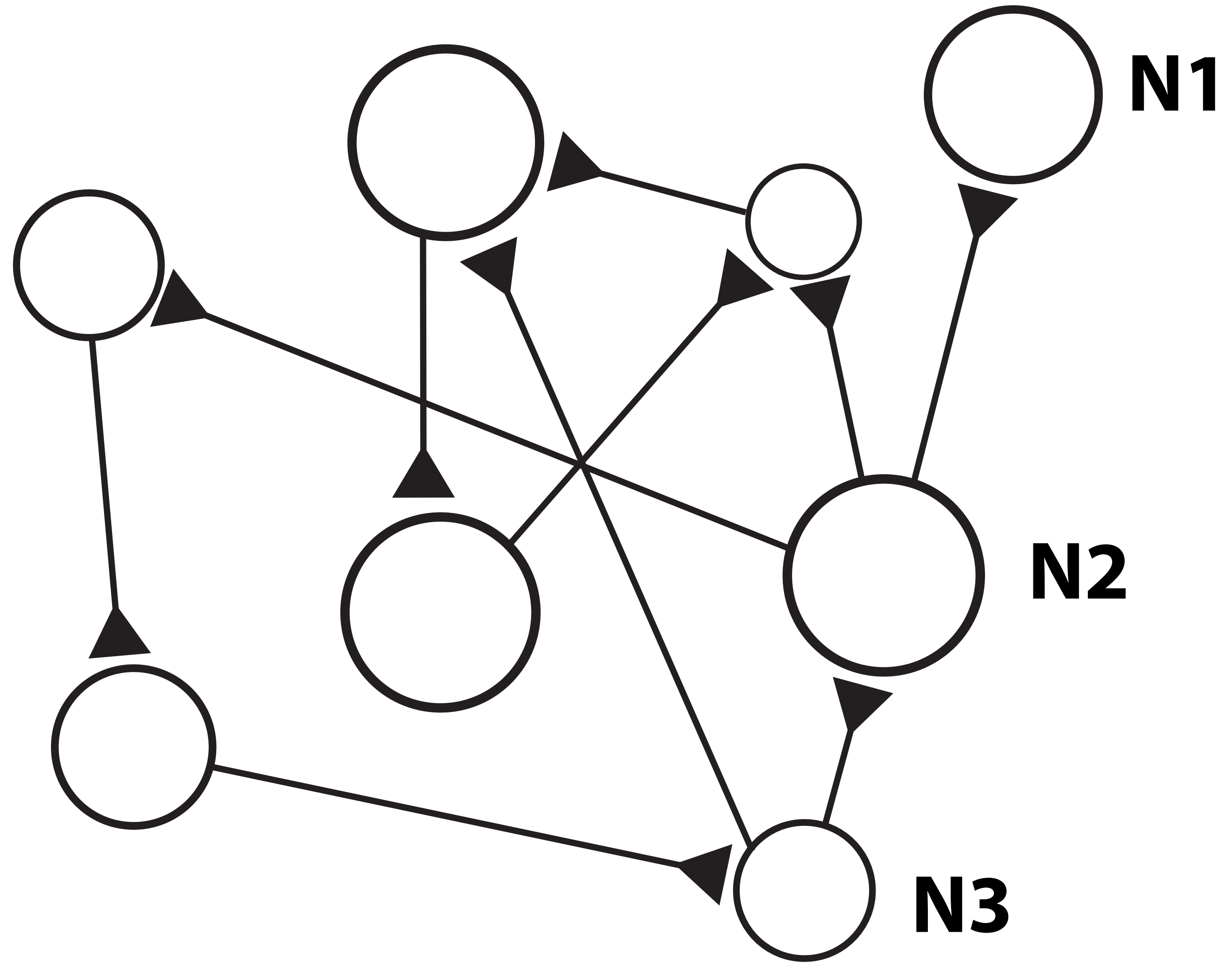
Mixed selective neurons

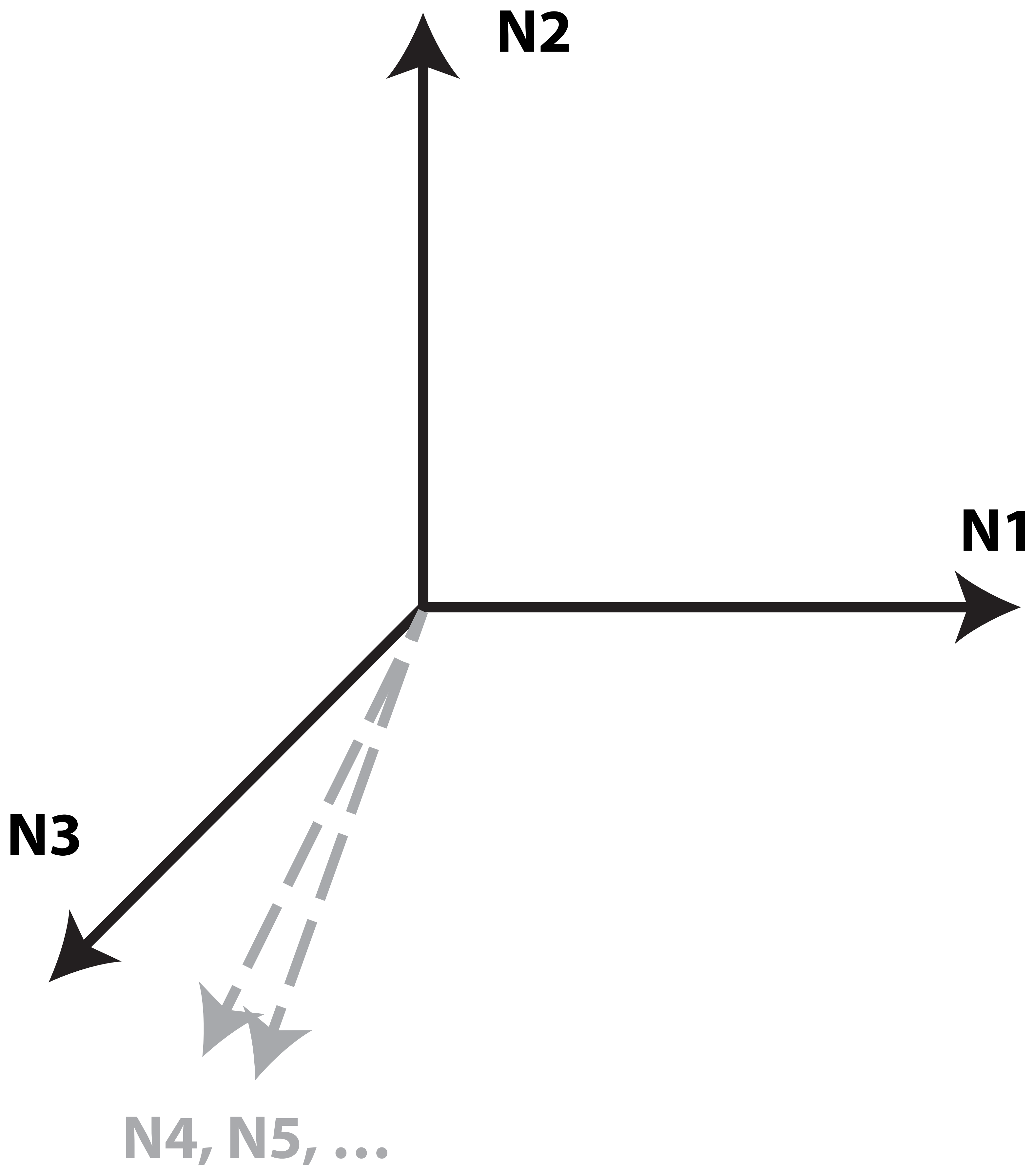
[Mante, Sussillo et al., *Context-dependent computations by recurrent dynamics in prefrontal cortex*, 2013]



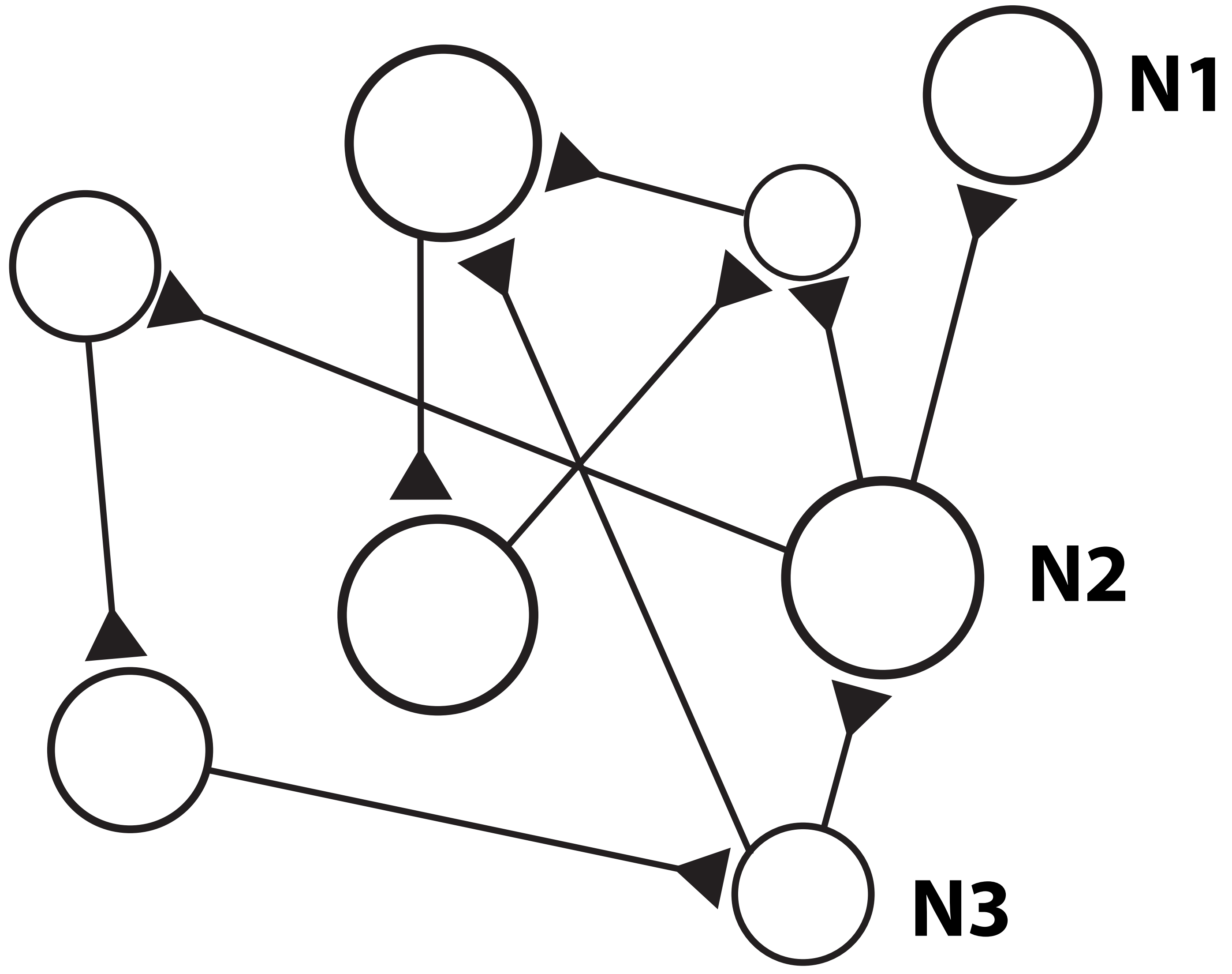


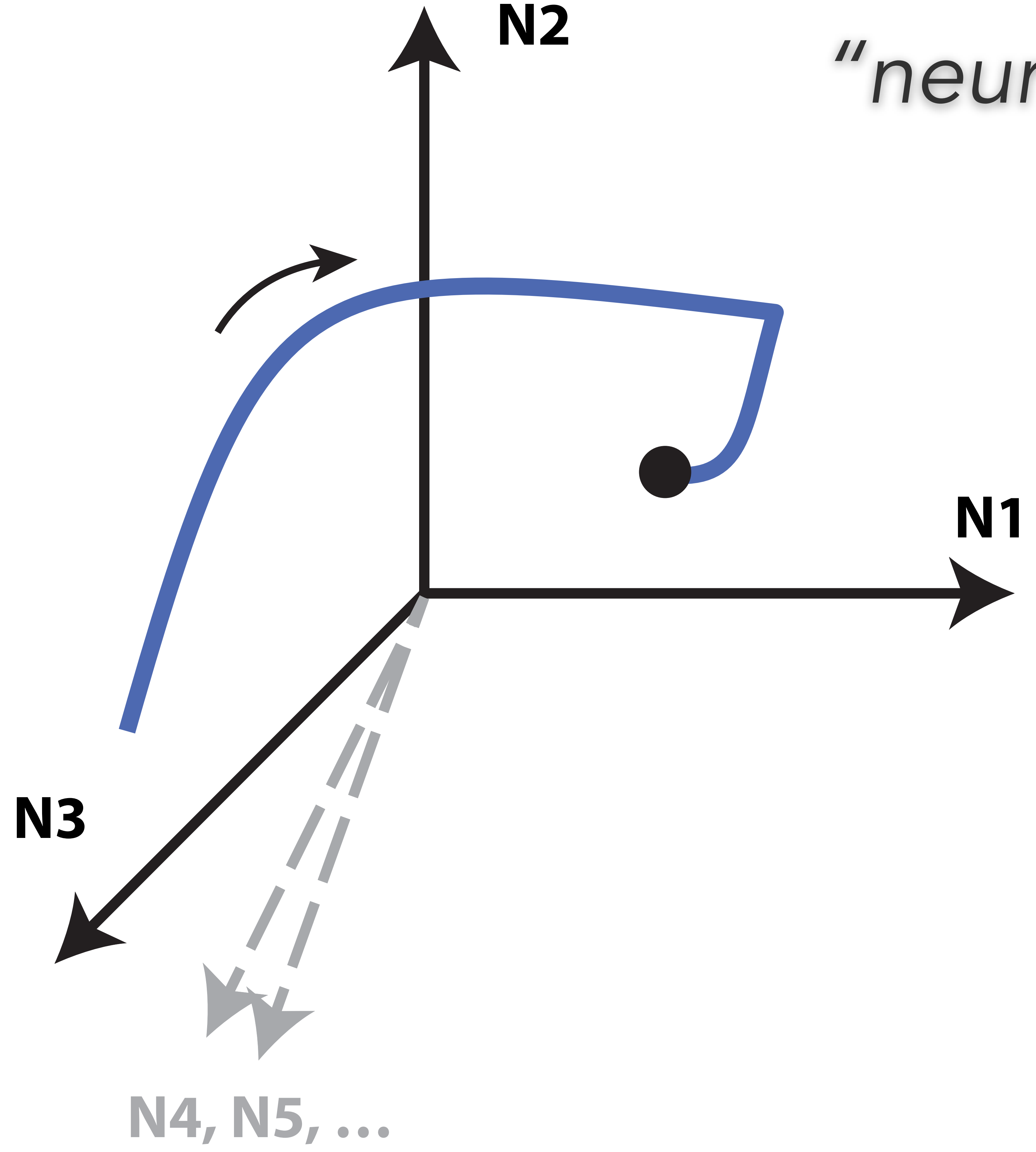
State-space analysis





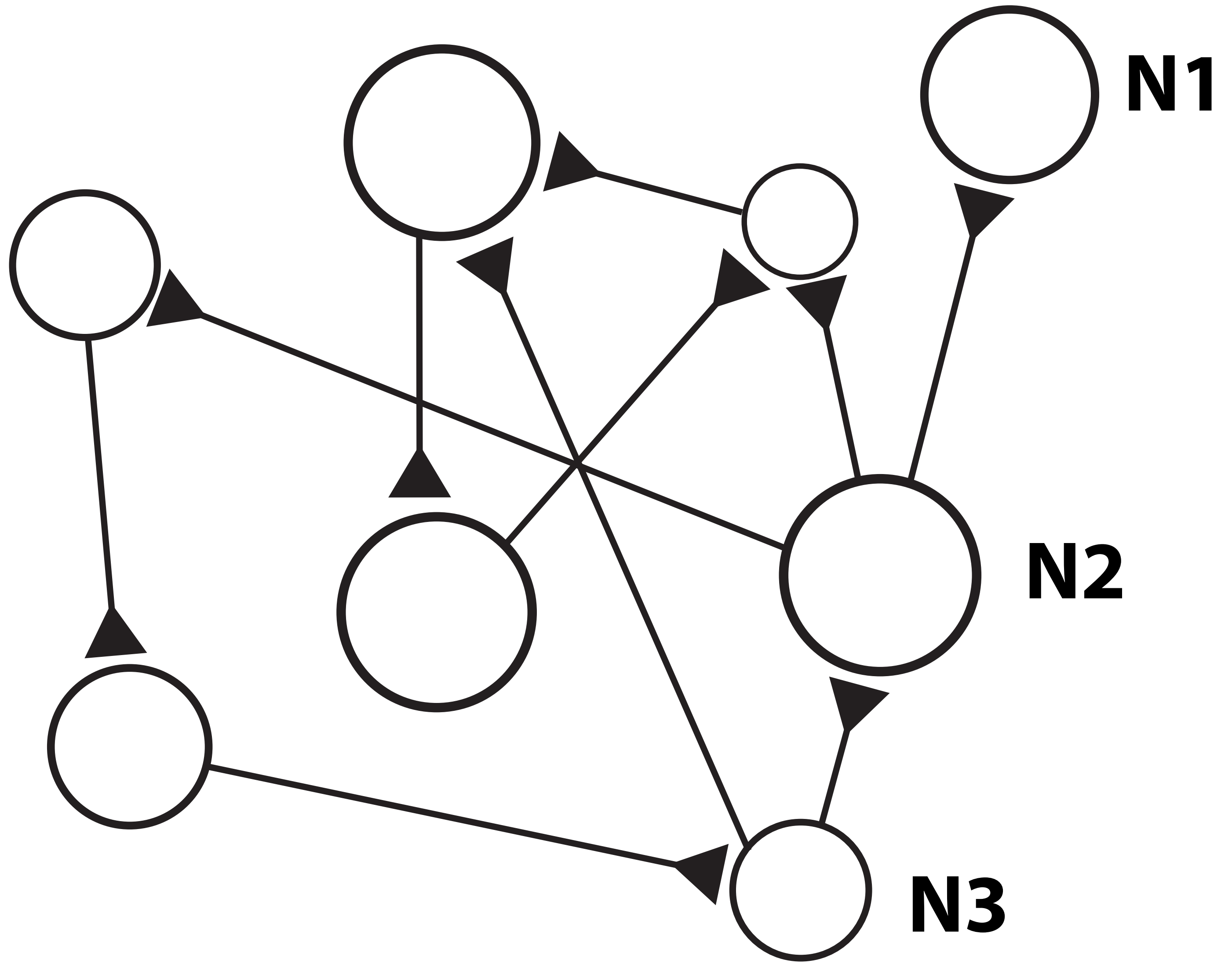
State-space analysis

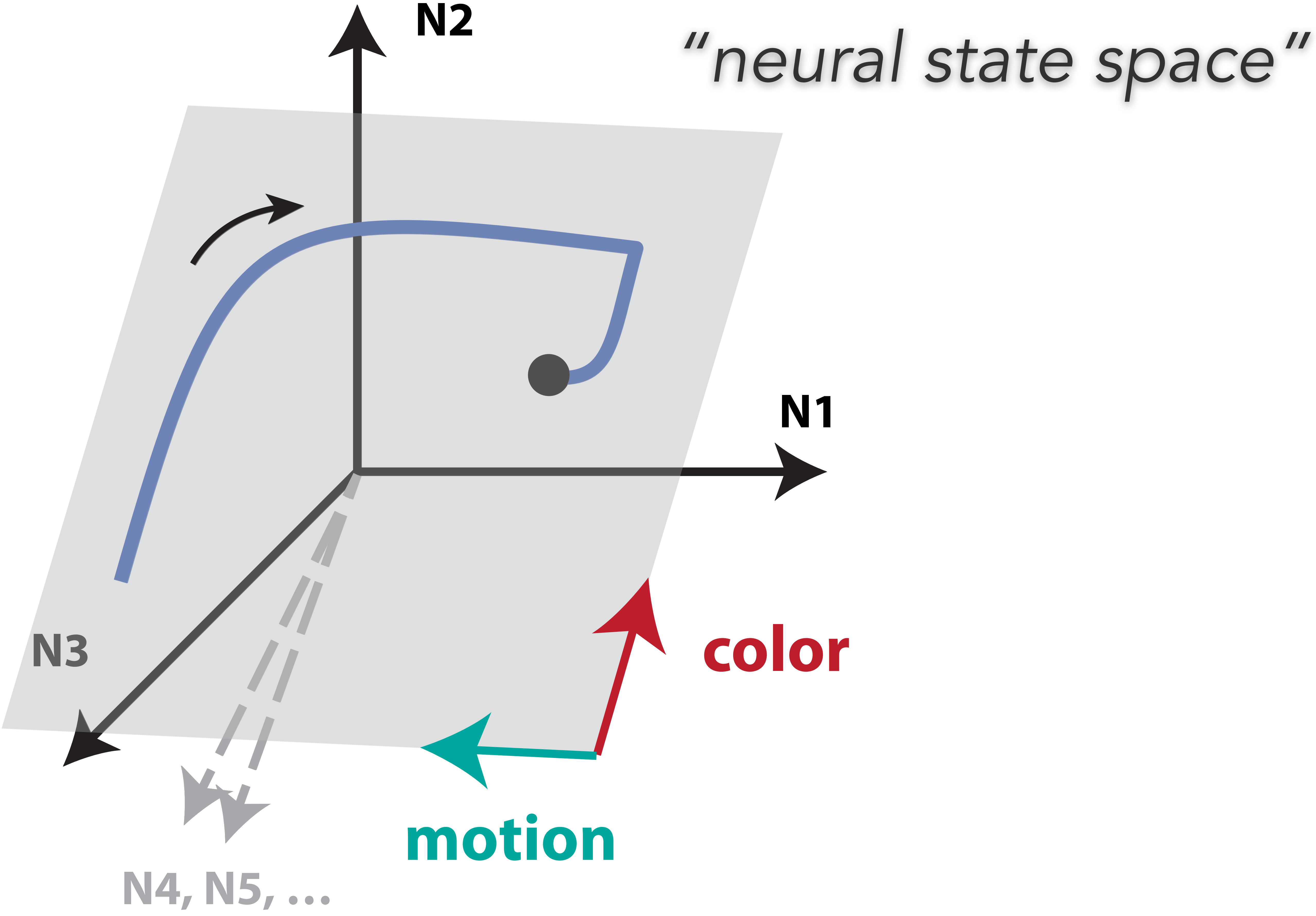




"neural state space"

State space analysis

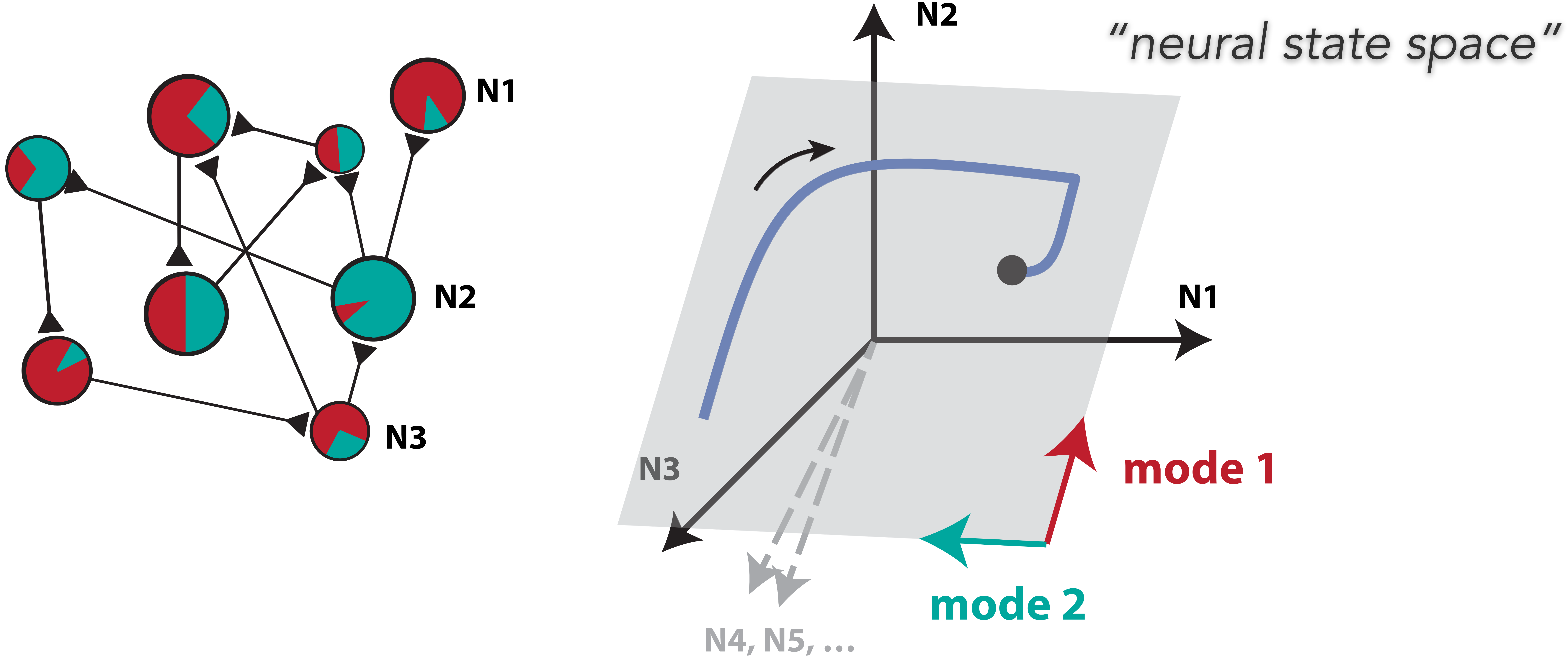




State space analysis

[Yuste, Nat Rev Neuro, 2015]

[Gallego et al., Neuron, 2017]

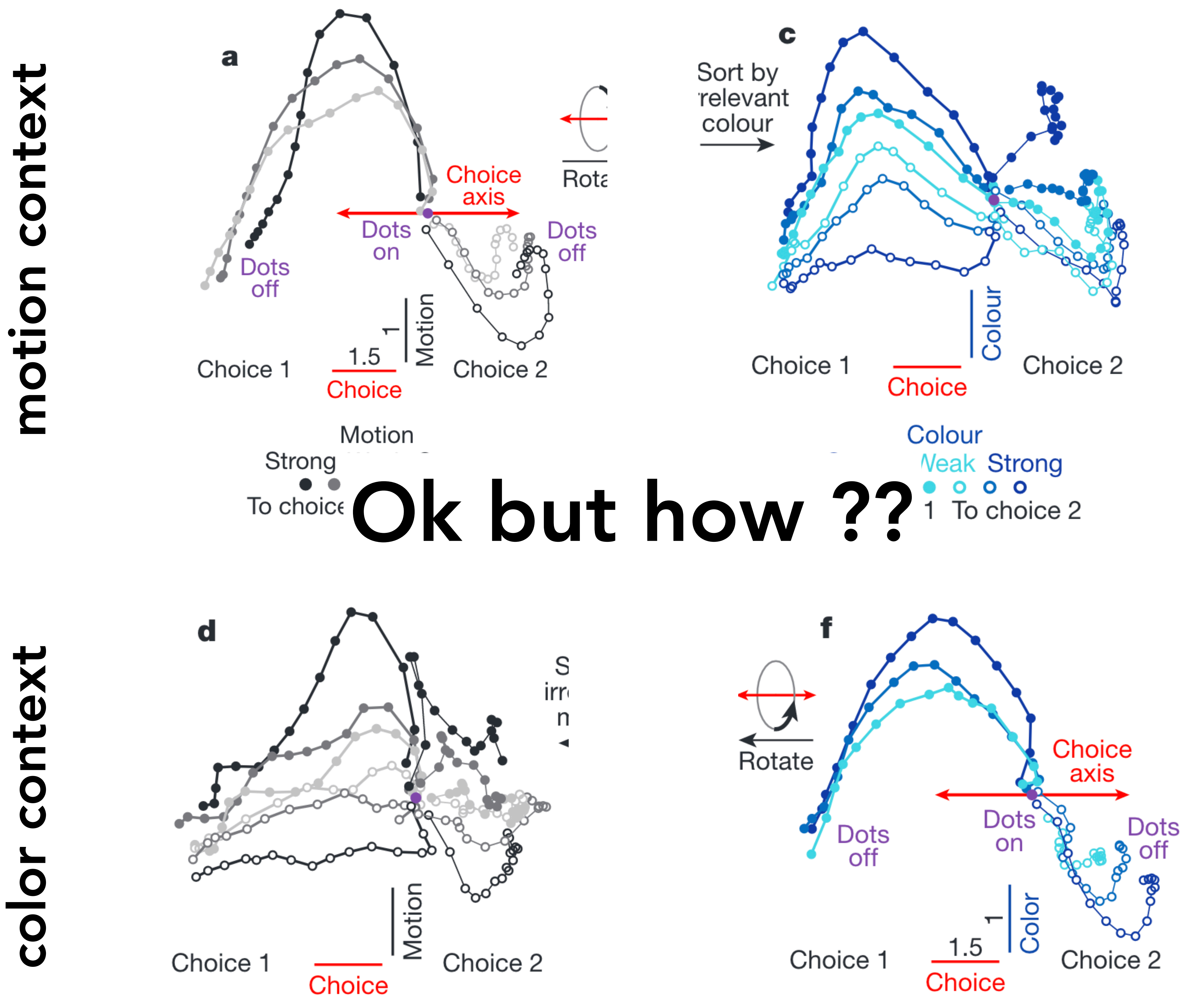


State space analysis: the Mante task

[Mante, Sussillo et al., *Context-dependent computations by recurrent dynamics in prefrontal cortex*, 2013]

Find 3 axes:

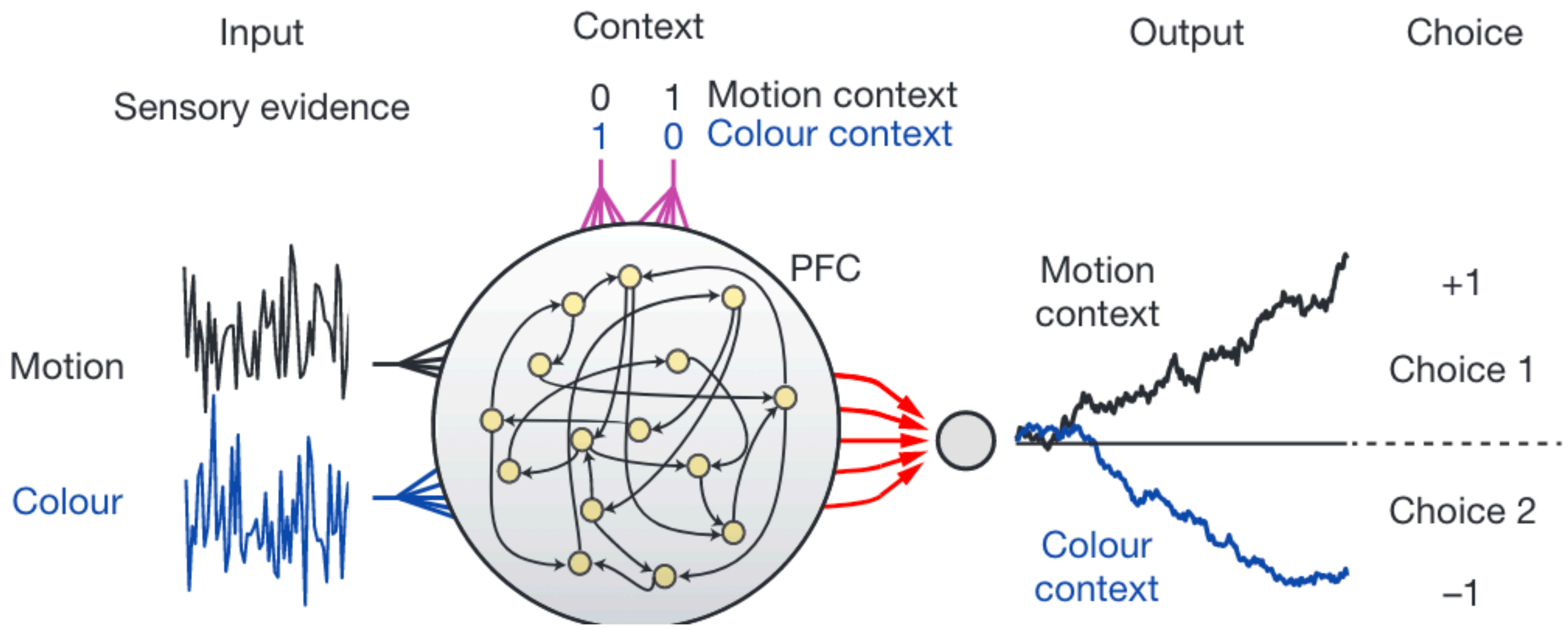
- color
- motion
- choice



Ok but how ??

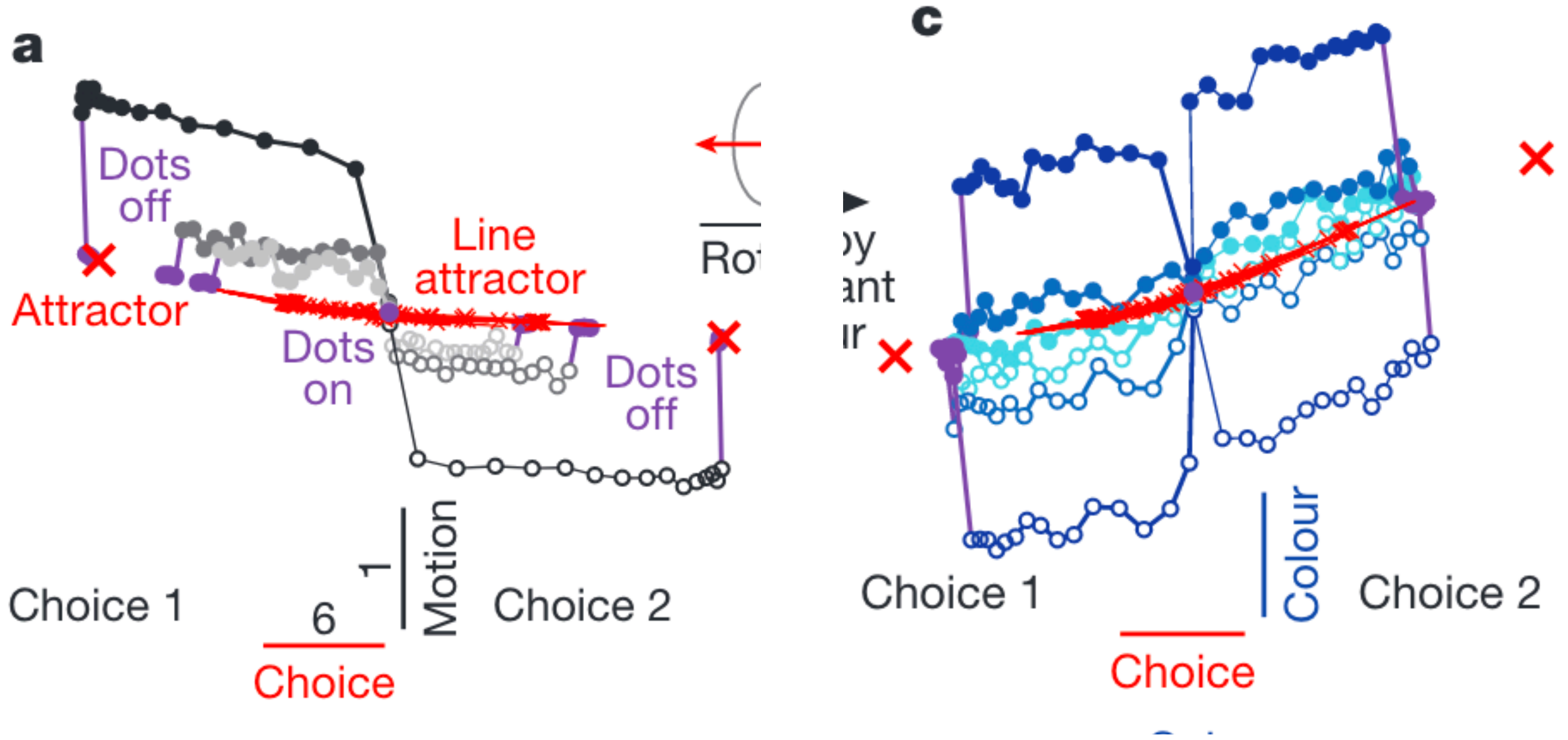
What would an RNN do ?

[Mante, Sussillo et al., *Context-dependent computations by recurrent dynamics in prefrontal cortex*, 2013]

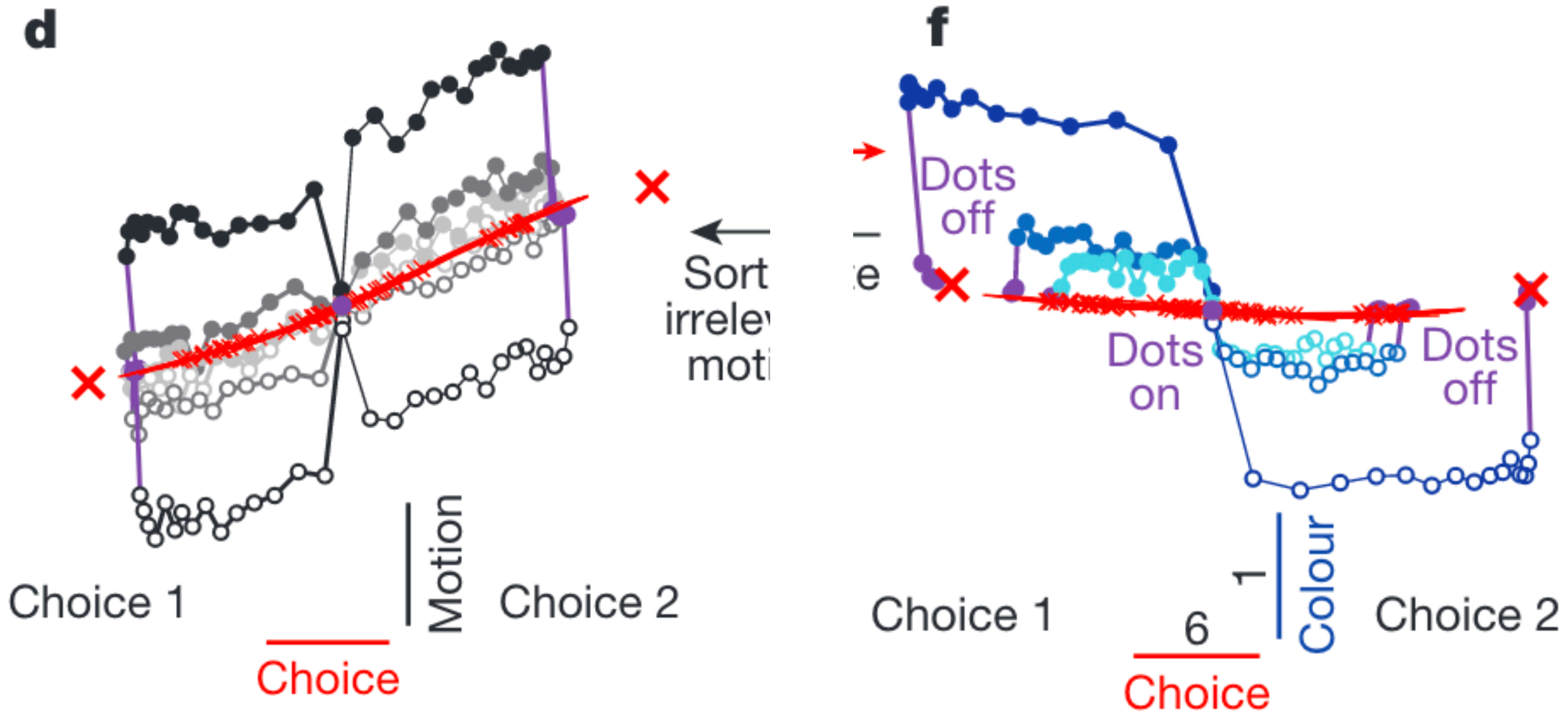


What would an RNN do ?

[Mante, Sussillo et al., *Context-dependent computations by recurrent dynamics in prefrontal cortex*, 2013]



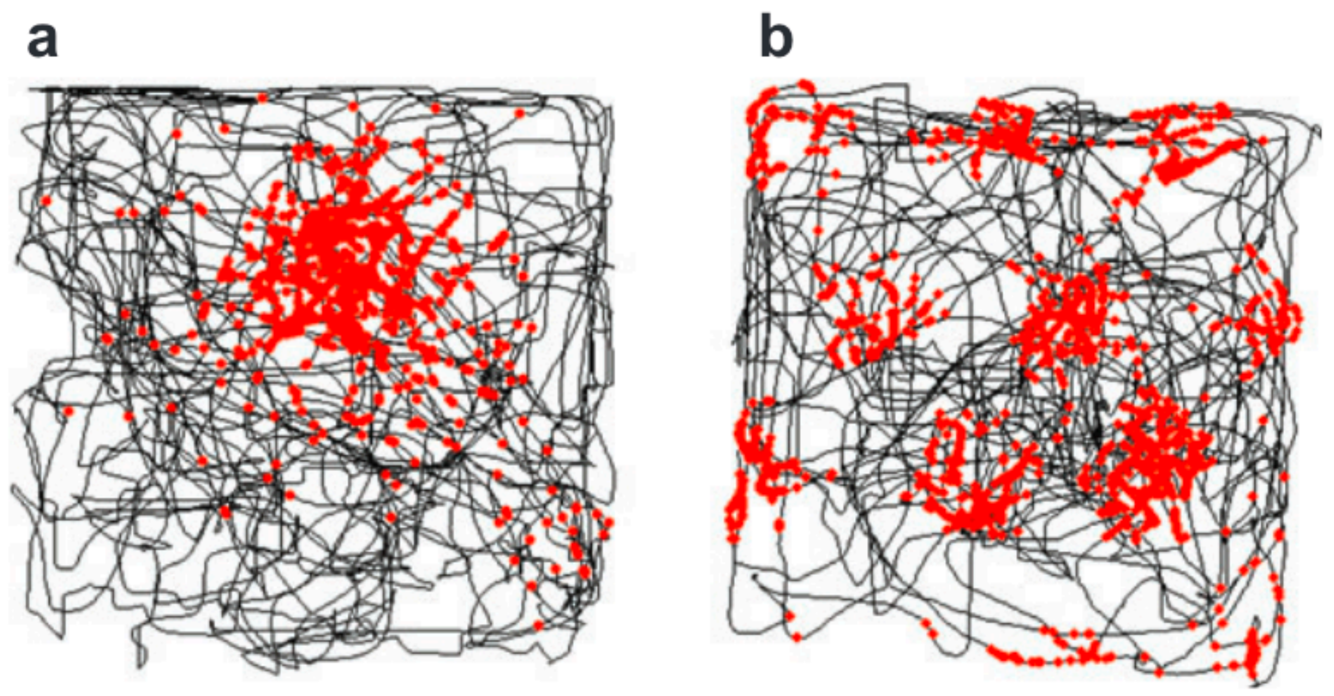
➡ **Very similar dynamics as the monkey brain**

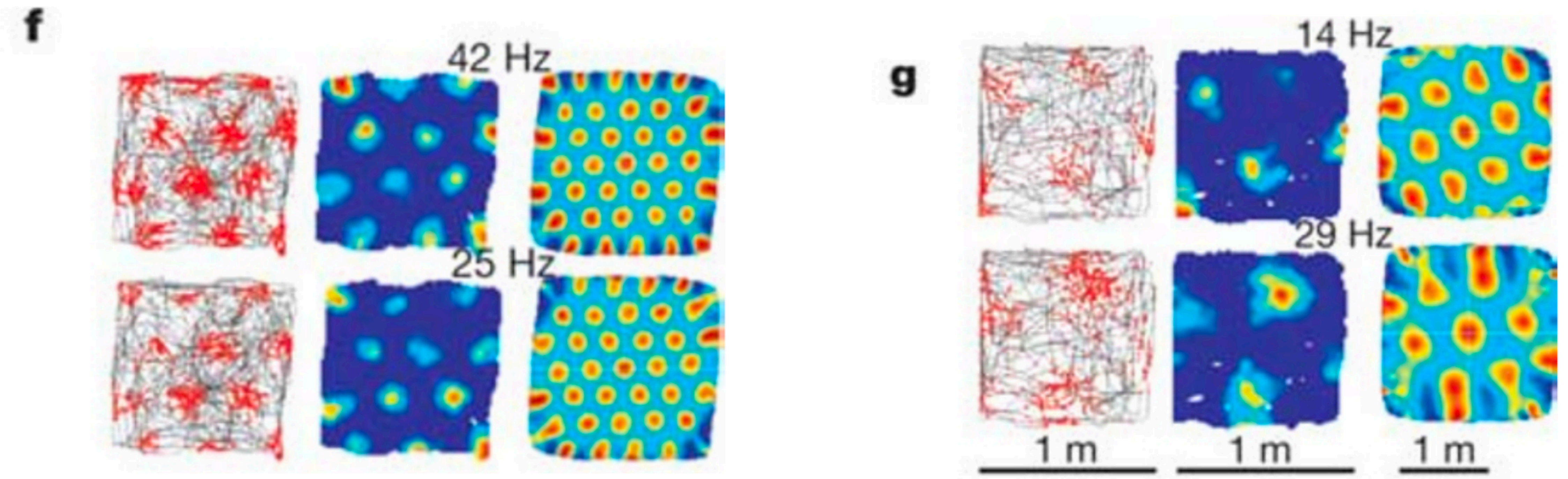


Example 2: position encoding and grid cells

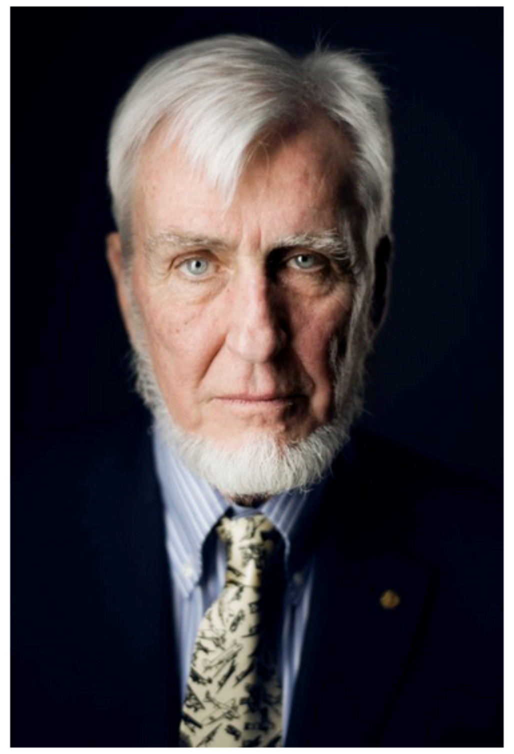
[O'Keefe & Dostrovsky, 1971] [Hafting et al., 2005], [Moser, Kropff, Moser, 2008]

place cell grid cell





The Nobel Prize in Physiology or Medicine 2014



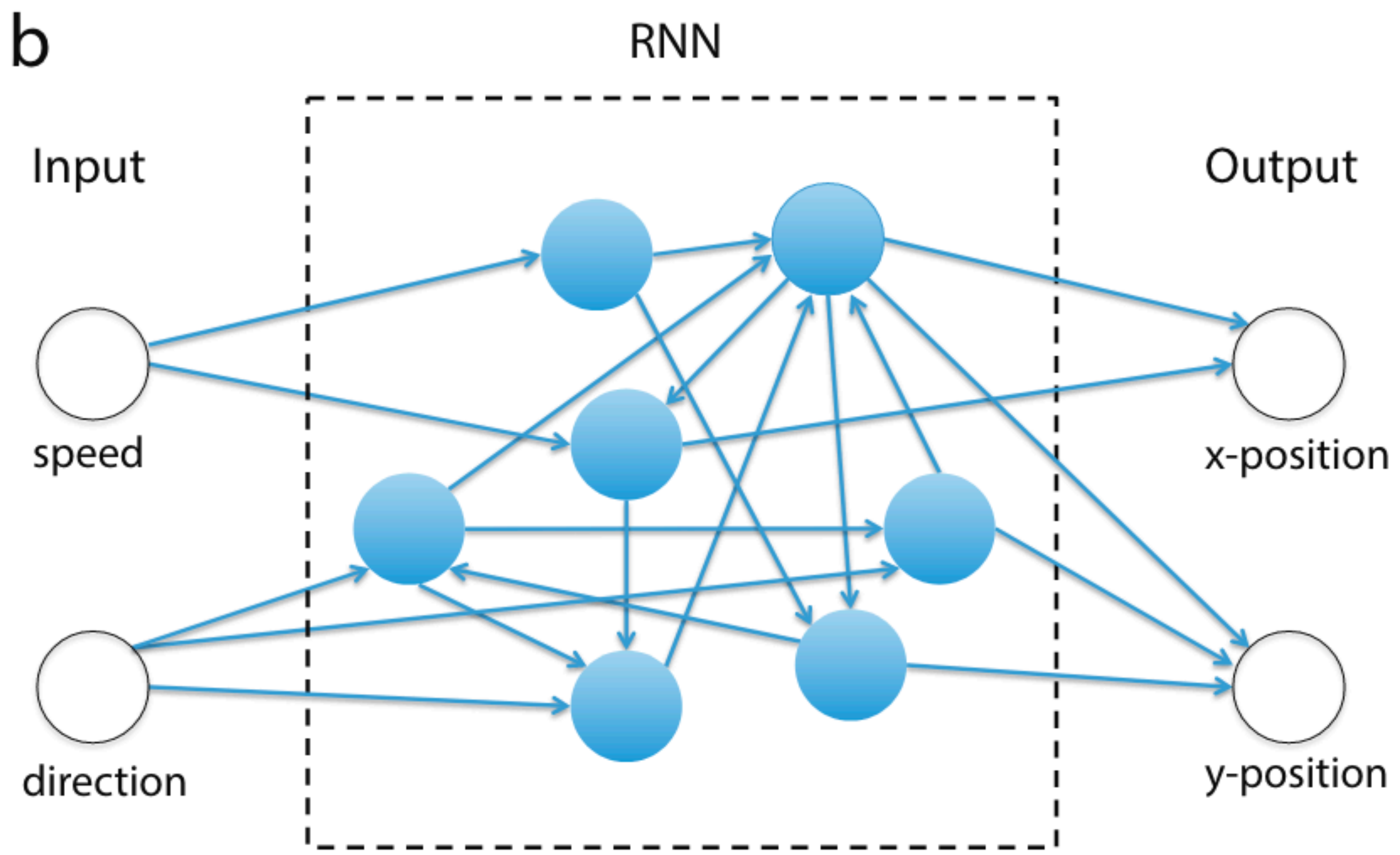
John O'Keefe
Prize share: 1/2

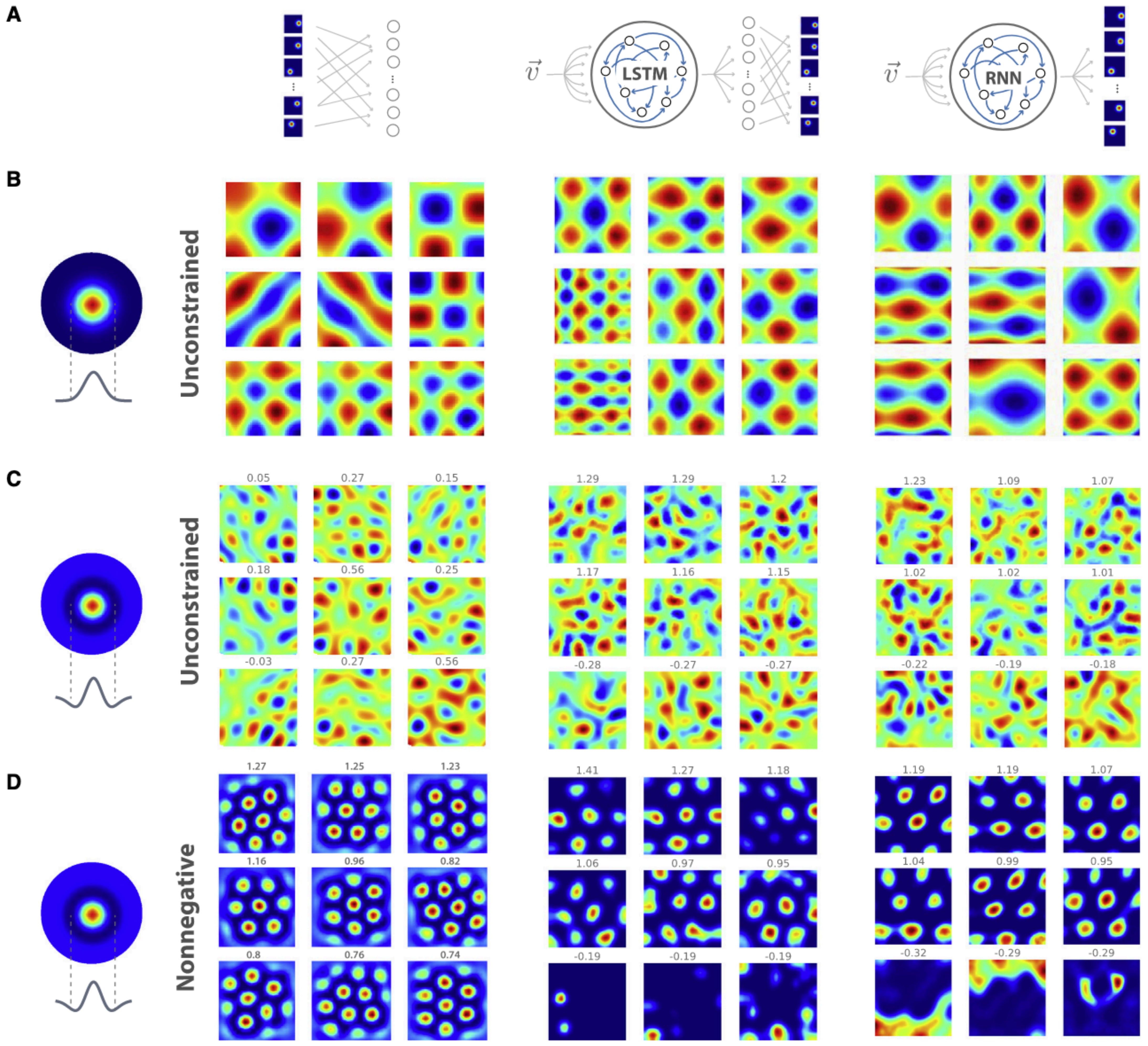
May-Britt Moser
Prize share: 1/4

Edvard I. Moser
Prize share: 1/4

Example 2: Retrieving grid cells in RNNs

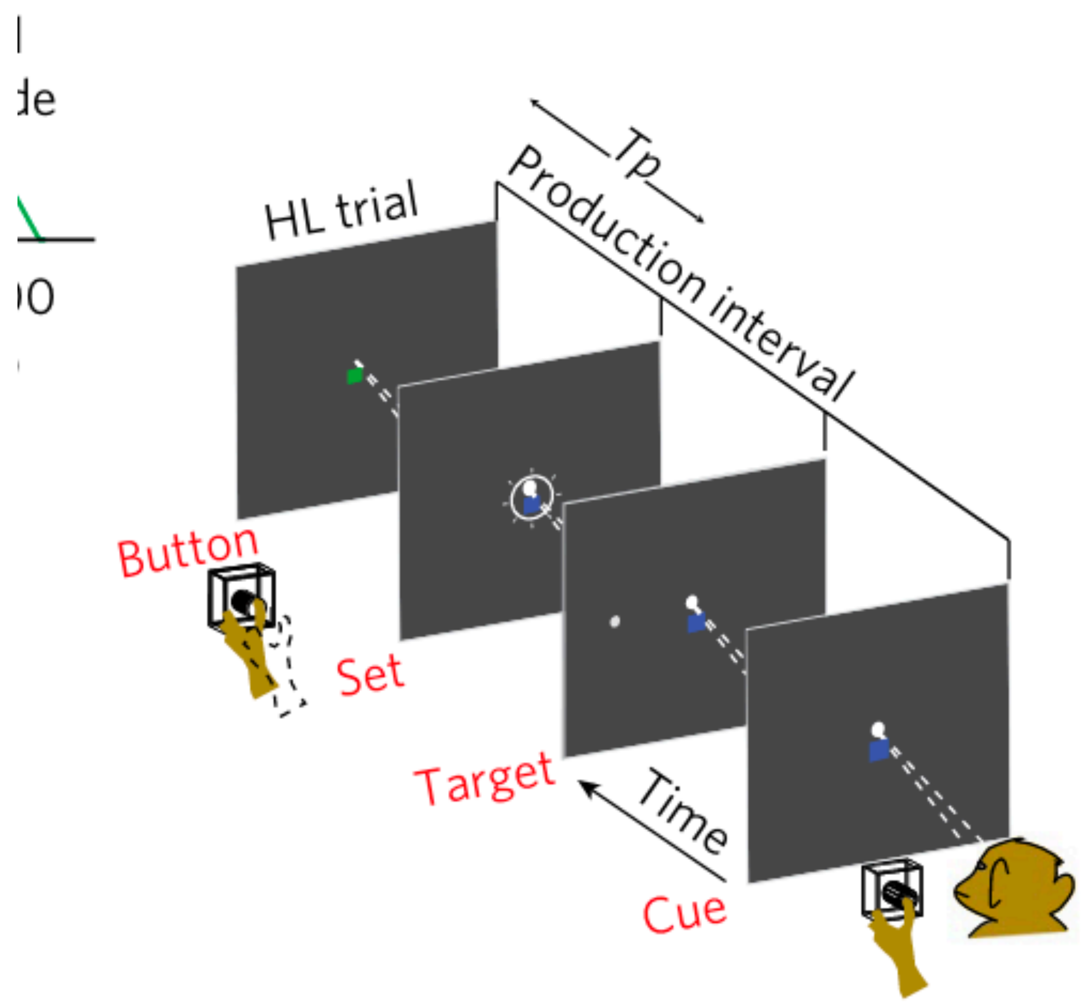
[Cueva & Wei, 2018] [Sorscher et al., 2022]



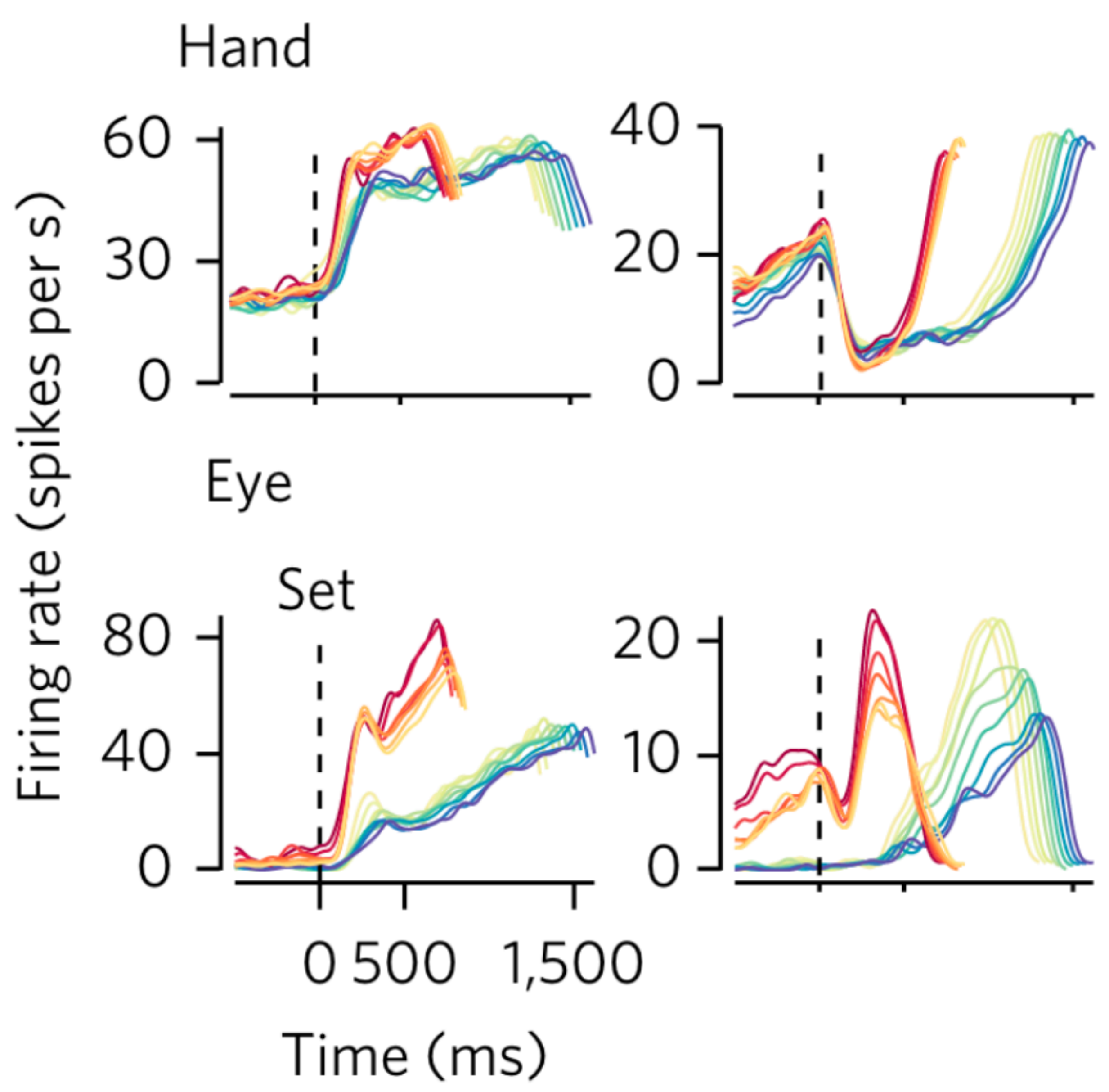


Example 3: timing tasks in RNNs

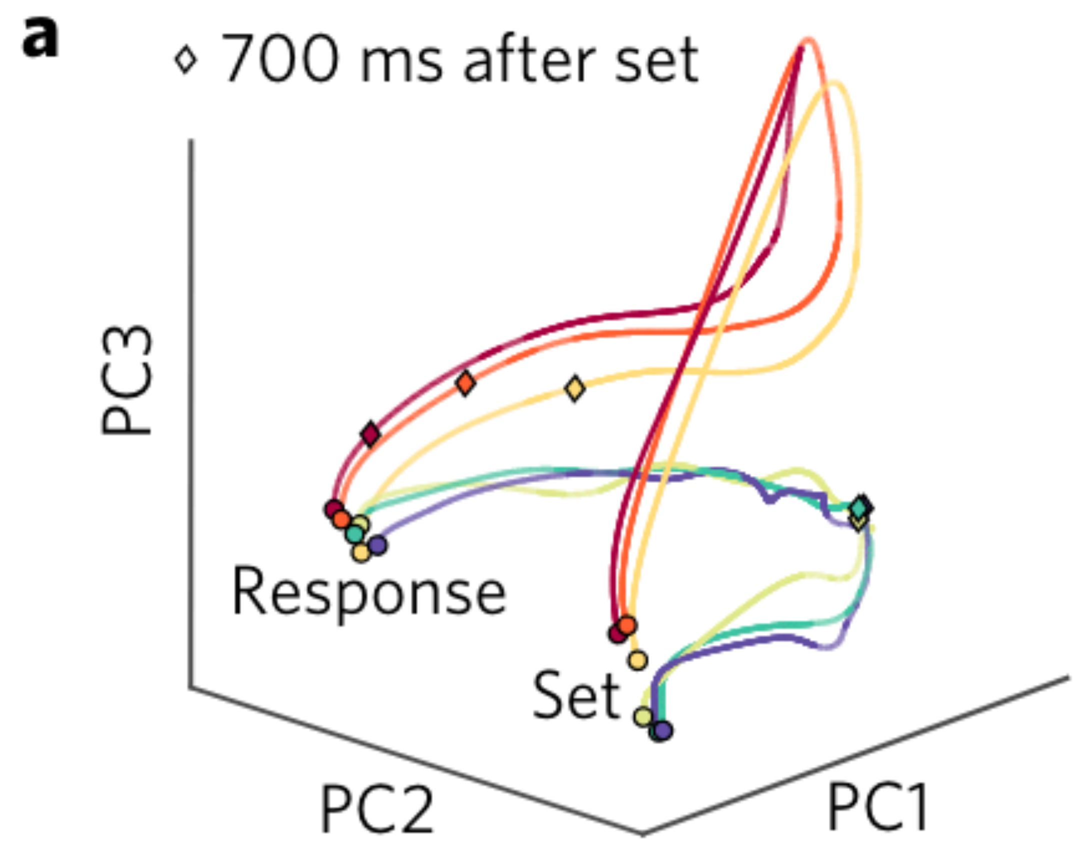
[J. Wang, D. Narain et al., *Flexible tiing by temporal scaling of cortical responses*, 2018]



Measure-Set-Go task

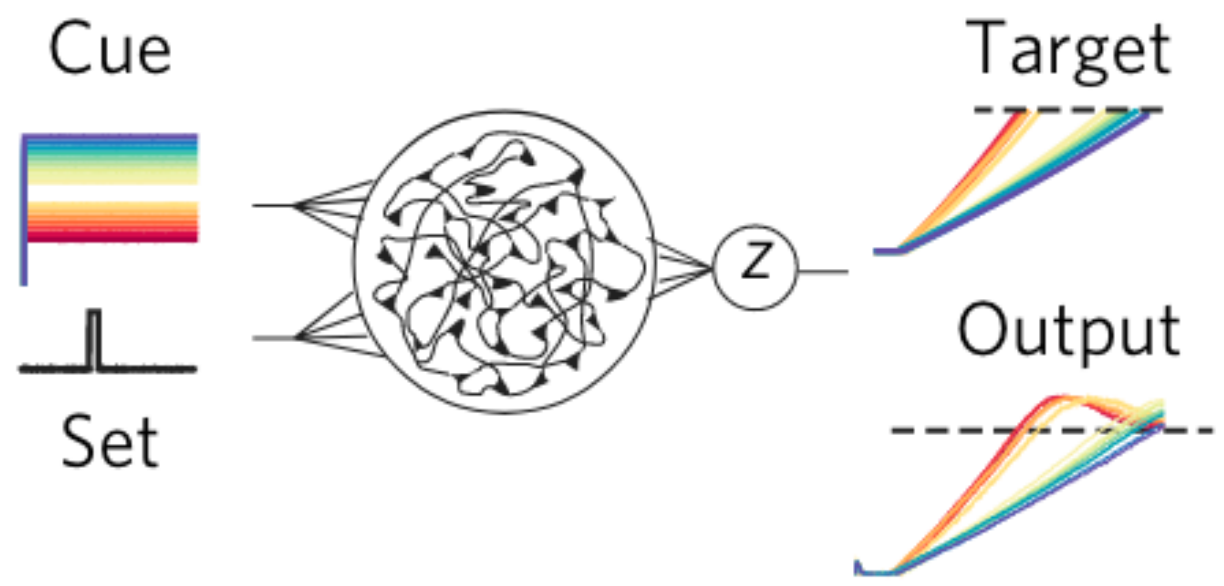


Single neuron responses

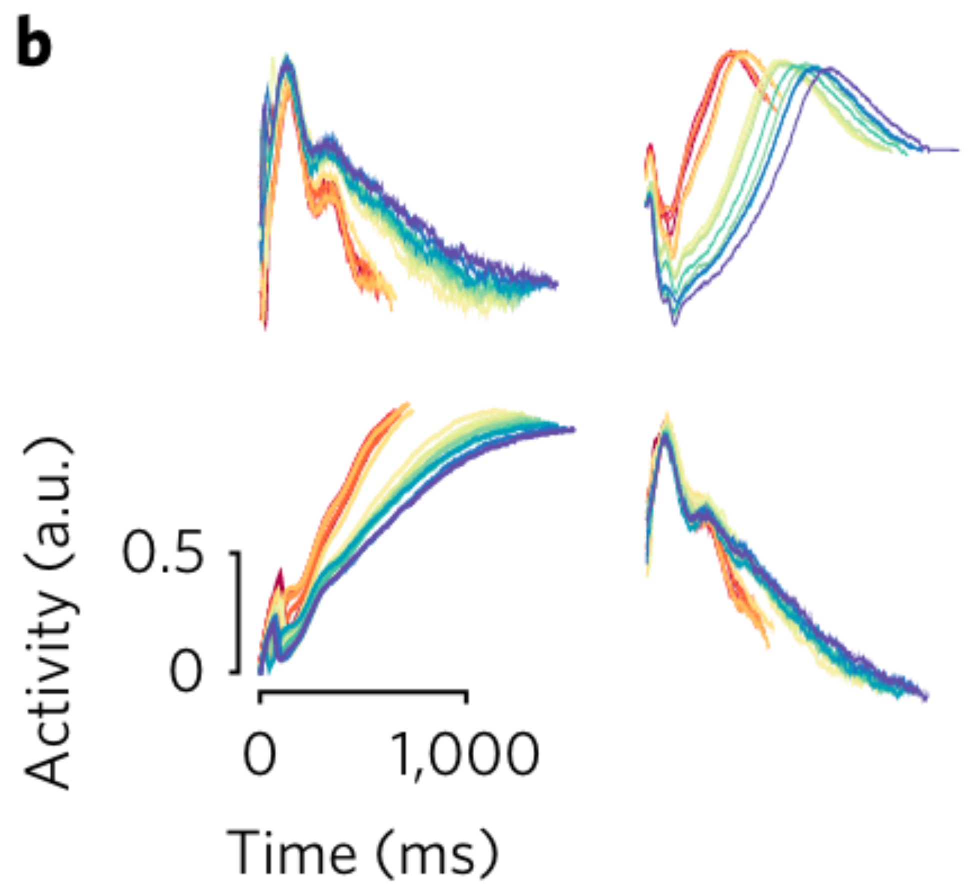


State-space dynamics

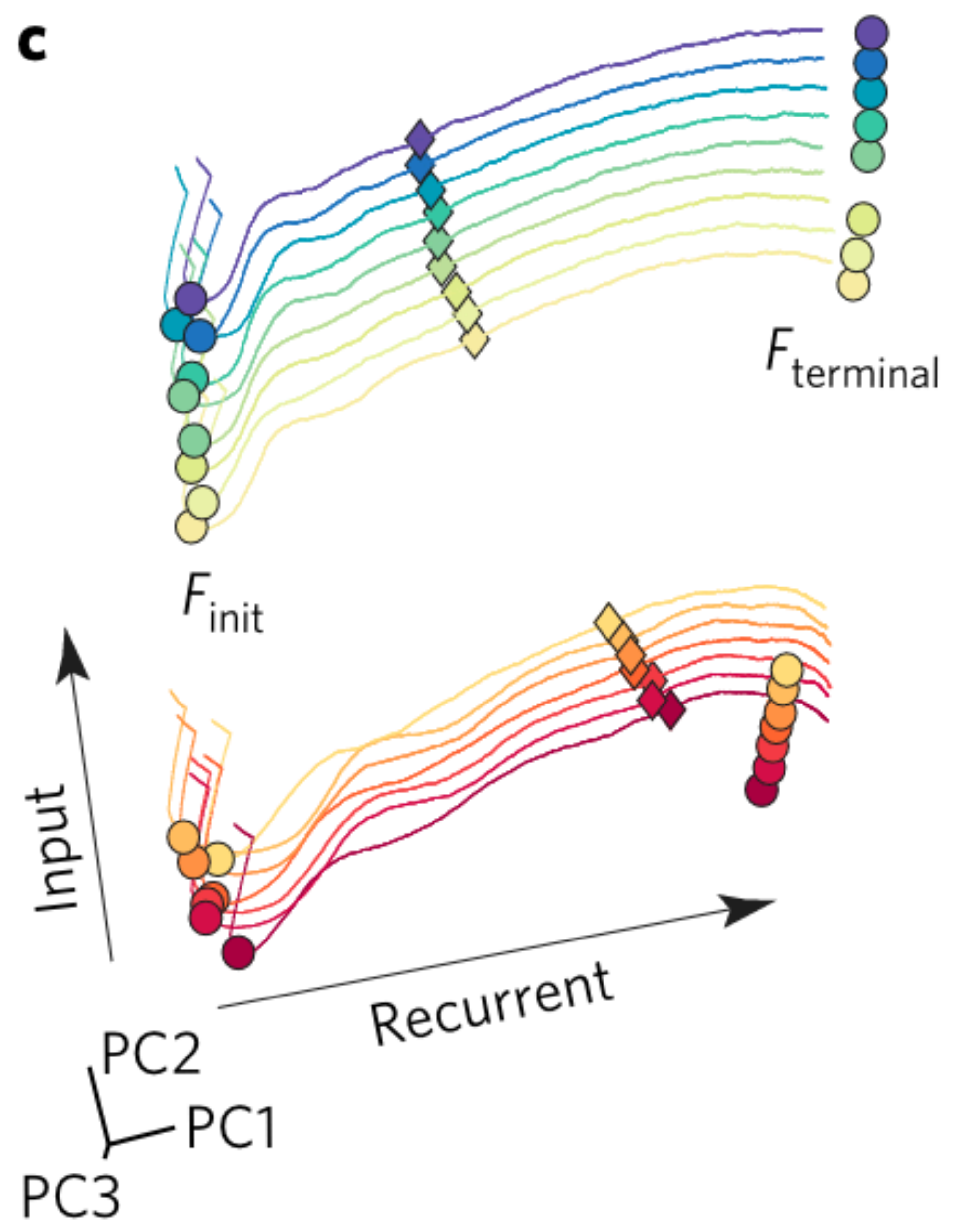
Example 3: timing tasks in RNNs



RNN version



Single units



State-space

Similar to brain data!!

Summary

- RNNs allow to build *normative* models of neural computations.
- They are a tool for *unsupervised hypothesis generation*.
- They turn out to reveal similarities with brain data.

Disadvantages:

- We replace a complicated object by another big, complicated *black box*.
- Lots of biological details are different.

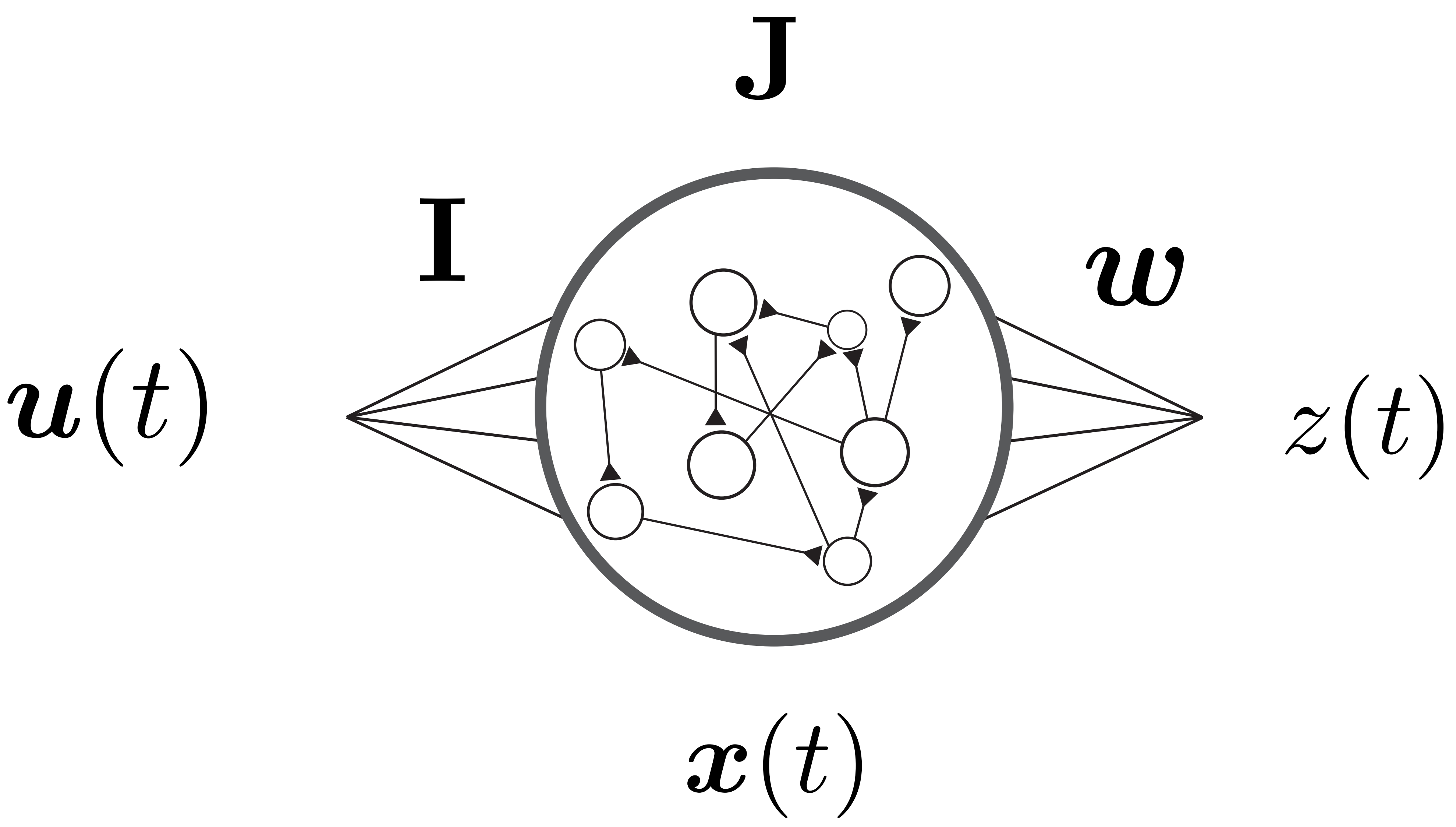
Advantage:

- We have access to everything in this new black box (connectivity, perturbations, unlimited data…)

Chapter II

Mathematics of RNNs

Recurrent neural network - formal description



internal state $\boldsymbol{x}(t) = (x_1(t), \ldots, x_N(t))$

inputs $\boldsymbol{u}(t) = (u_1(t), \ldots, u_P(t))$

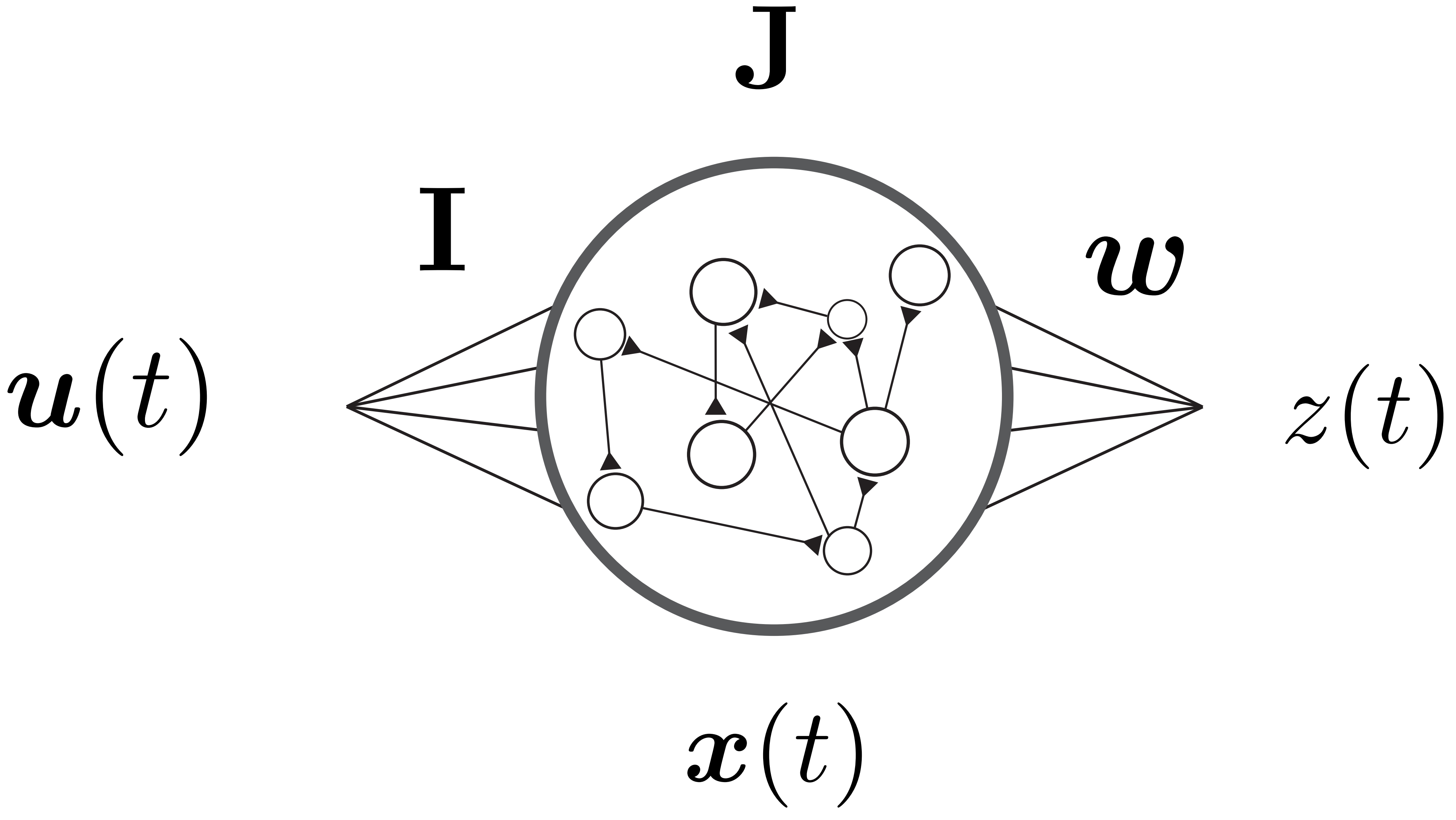
output $z(t) \in \mathbb{R}$

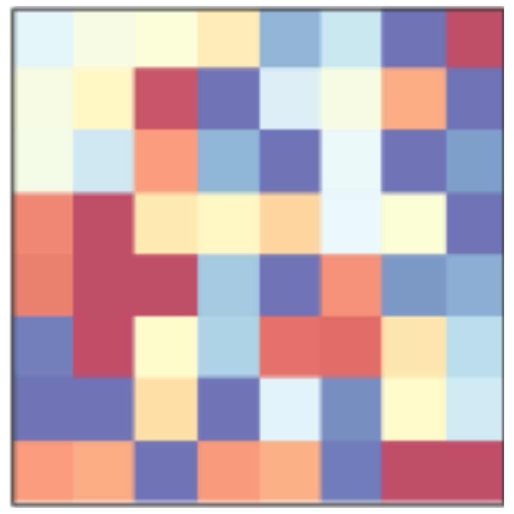
recurrent connectivity $\mathbf{J} = (J_{ij})$ (shape NxN)

input connectivity $\mathbf{I}$ (shape NxP)

output connectivity $\boldsymbol{w}$ (shape 1xN)

Recurrent neural network - formal description



$\boldsymbol{J} =$ 

Discrete-time equations

$$\boldsymbol{x}(t+1) = \mathbf{J}\phi(\boldsymbol{x}(t)) + \mathbf{I}\boldsymbol{u}(t)$$

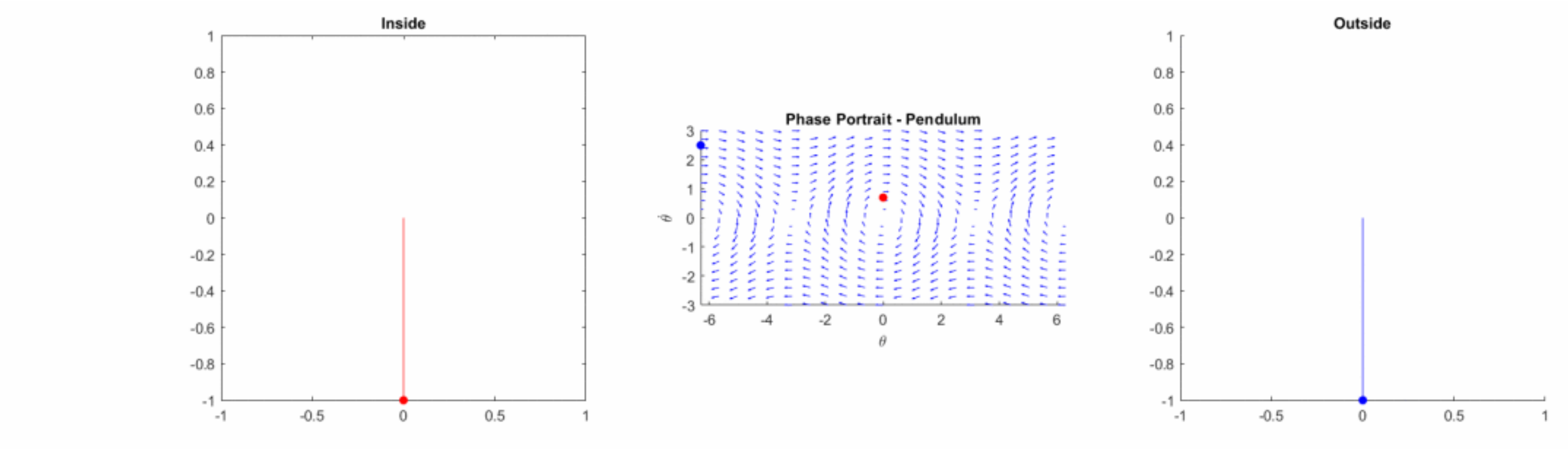
$$x_i(t+1) = \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) + \sum_{p=1}^{P} W_{ip}^{in} u_p(t)$$

Continuous-time equations

$$\tau \frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x}(t) + \mathbf{J}\phi(\boldsymbol{x}(t)) + \mathbf{I}\boldsymbol{u}(t)$$

Back to dynamical systems - flow fields

(aka phase portraits)

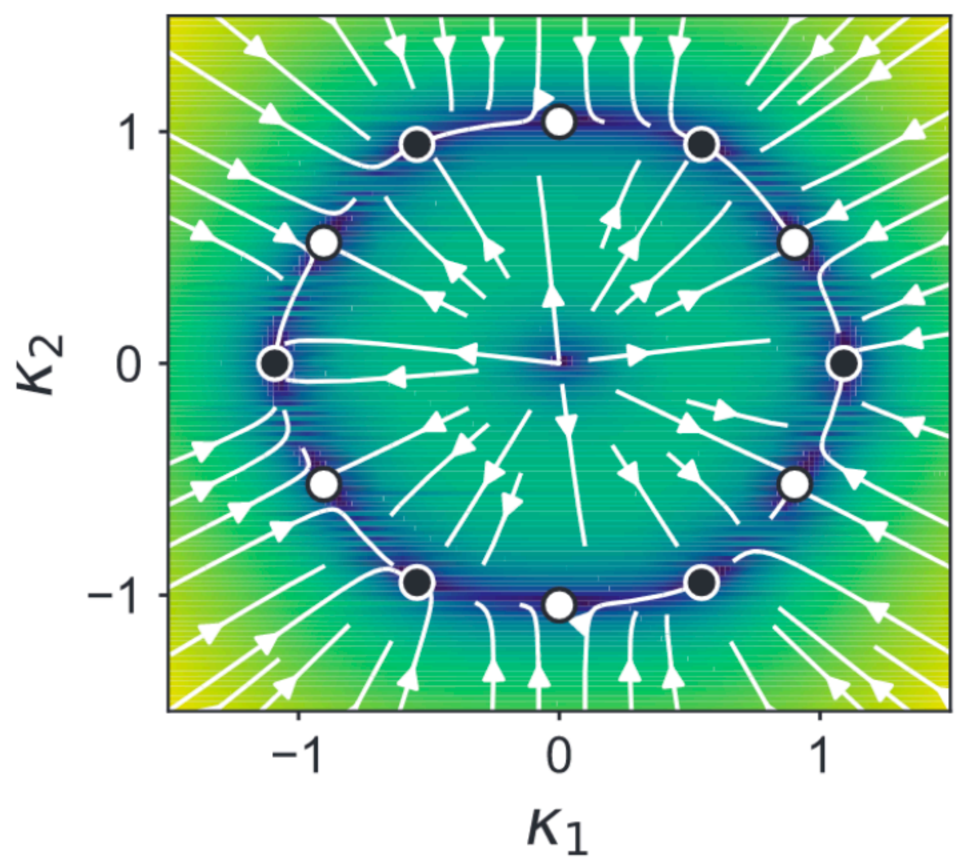


Long-term behavior of dynamical systems

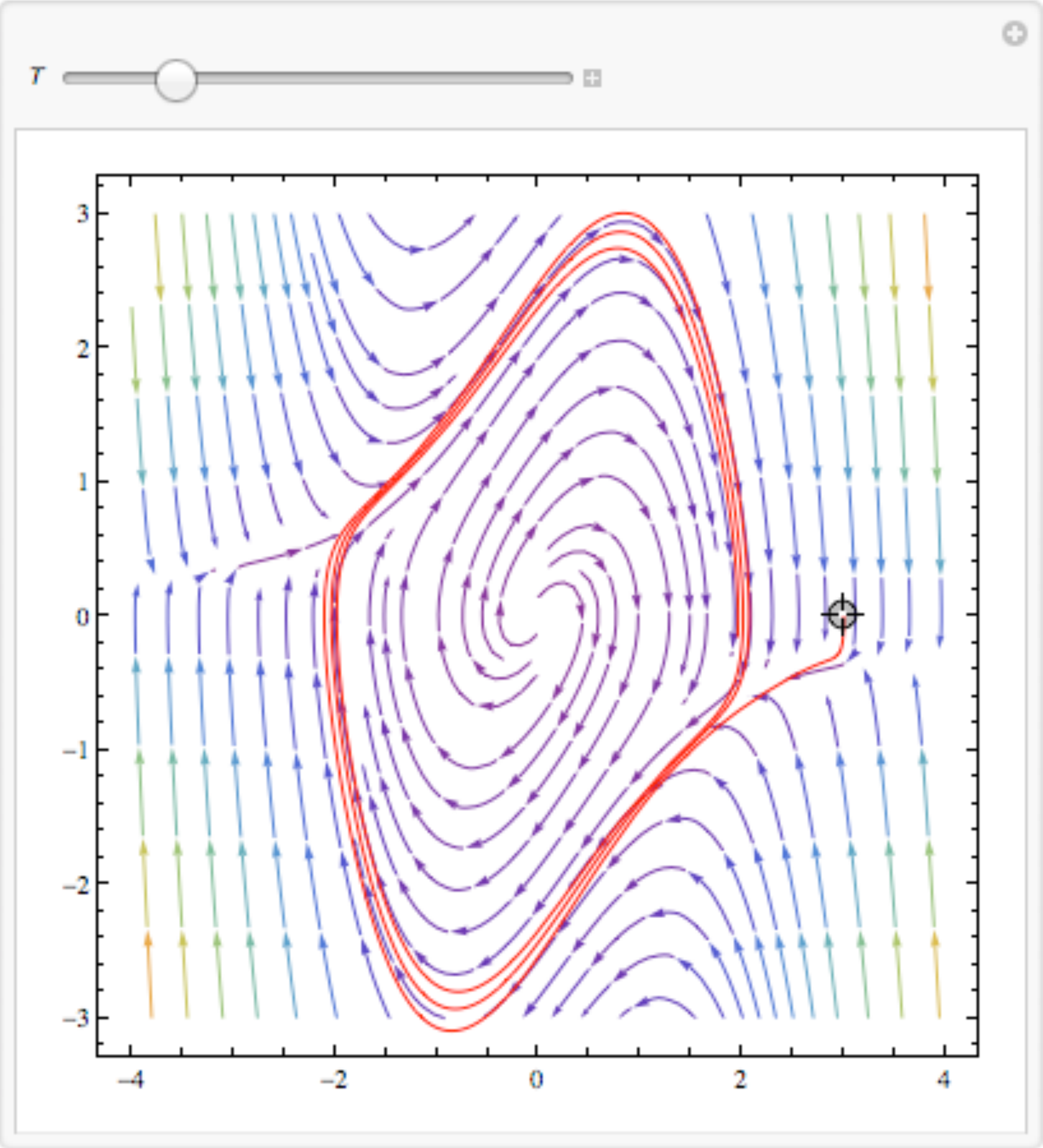
[Steven Strogatz, *Nonlinear dynamics and chaos*]

[Vyas et al., *Computation through neural population dynamics*, 2020]

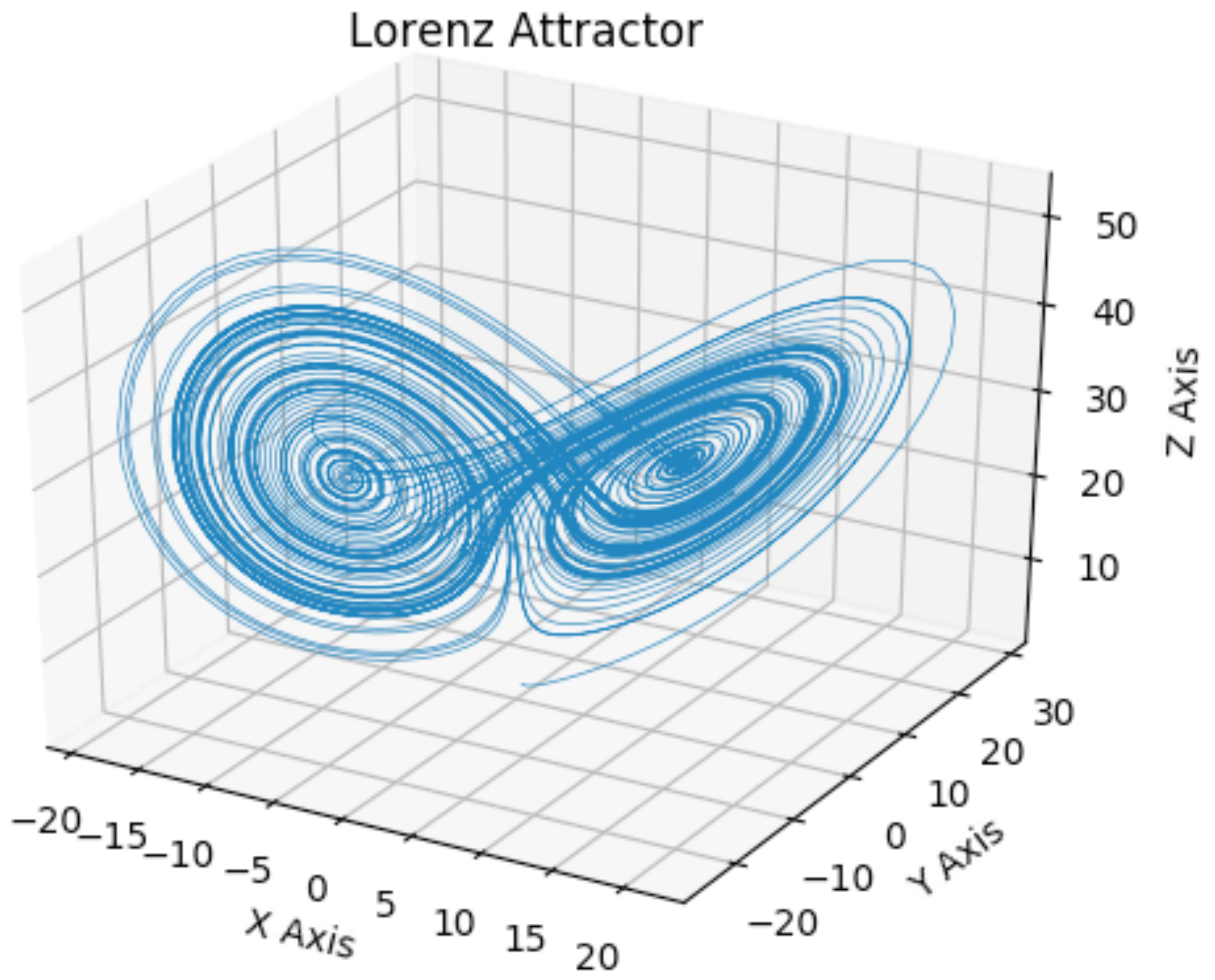
Fixed points



Periodic behavior



Chaotic behavior



> 3 dimensions

Example 1 - Hopfield networks

Neural networks and physical systems with emergent collective computational abilities

(associative memory/parallel processing/categorization/content-addressable memory/fail-soft devices)

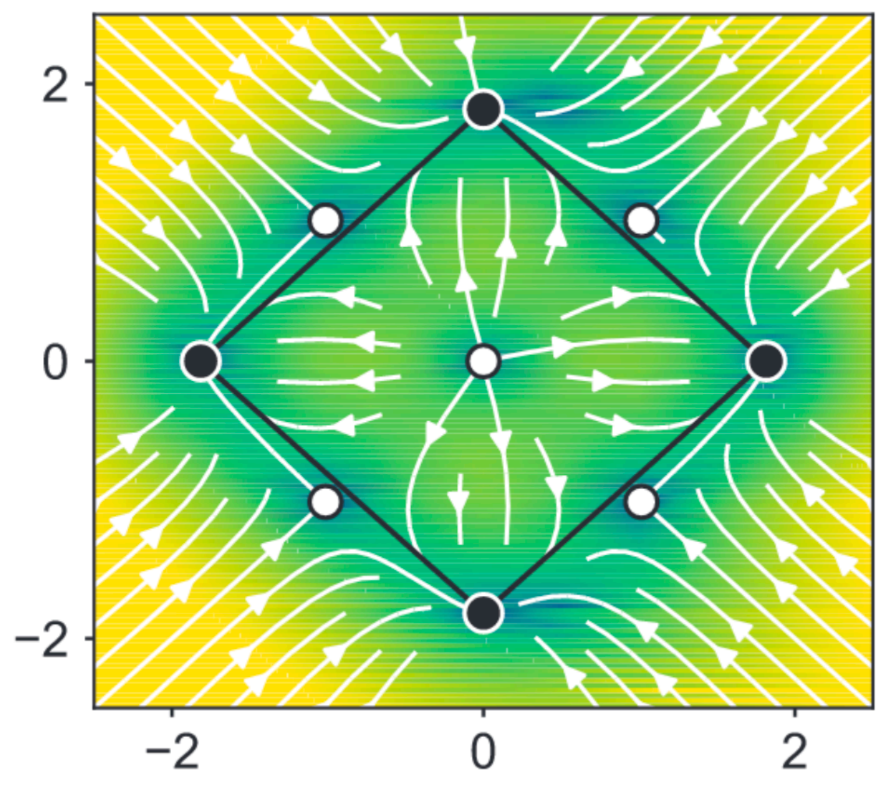
J. J. HOPFIELD

Division of Chemistry and Biology, California Institute of Technology, Pasadena, California 91125; and Bell Laboratories, Murray Hill, New Jersey 07974

Contributed by John J. Hopfield, January 15, 1982

To memorize a vector (v_i), use the connectivity matrix defined by:

$$J_{ij} = v_i v_j \quad \text{ie.} \quad \mathbf{J} = \boldsymbol{v}\boldsymbol{v}^T$$



Example 2: random RNNs

VOLUME 61, NUMBER 3 PHYSICAL REVIEW LETTERS 18 JULY 1988

Chaos in Random Neural Networks

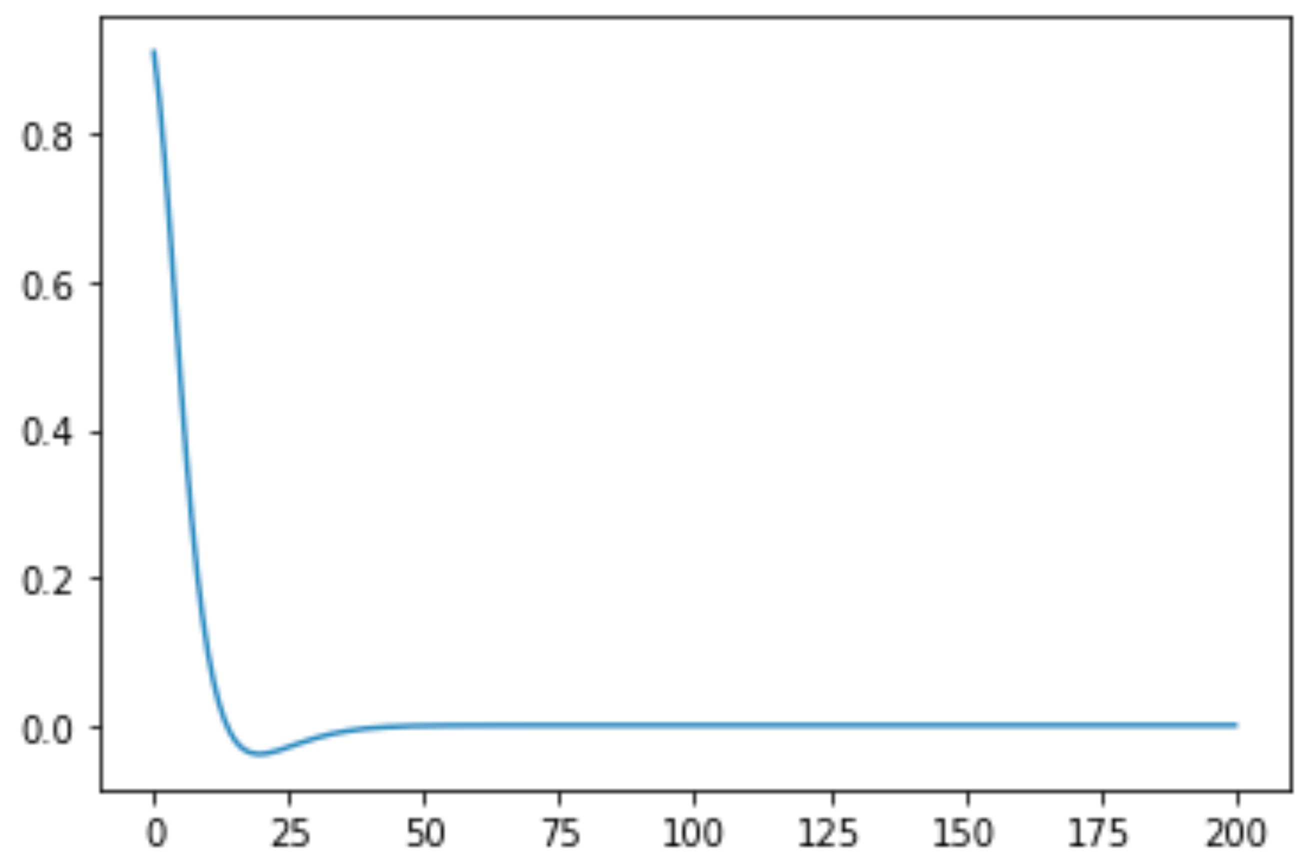
H. Sompolinsky[a] and A. Crisanti
AT&T Bell Laboratories, Murray Hill, New Jersey 07974, and
Racah Institute of Physics, The Hebrew University, 91904 Jerusalem, Israel[b]

and

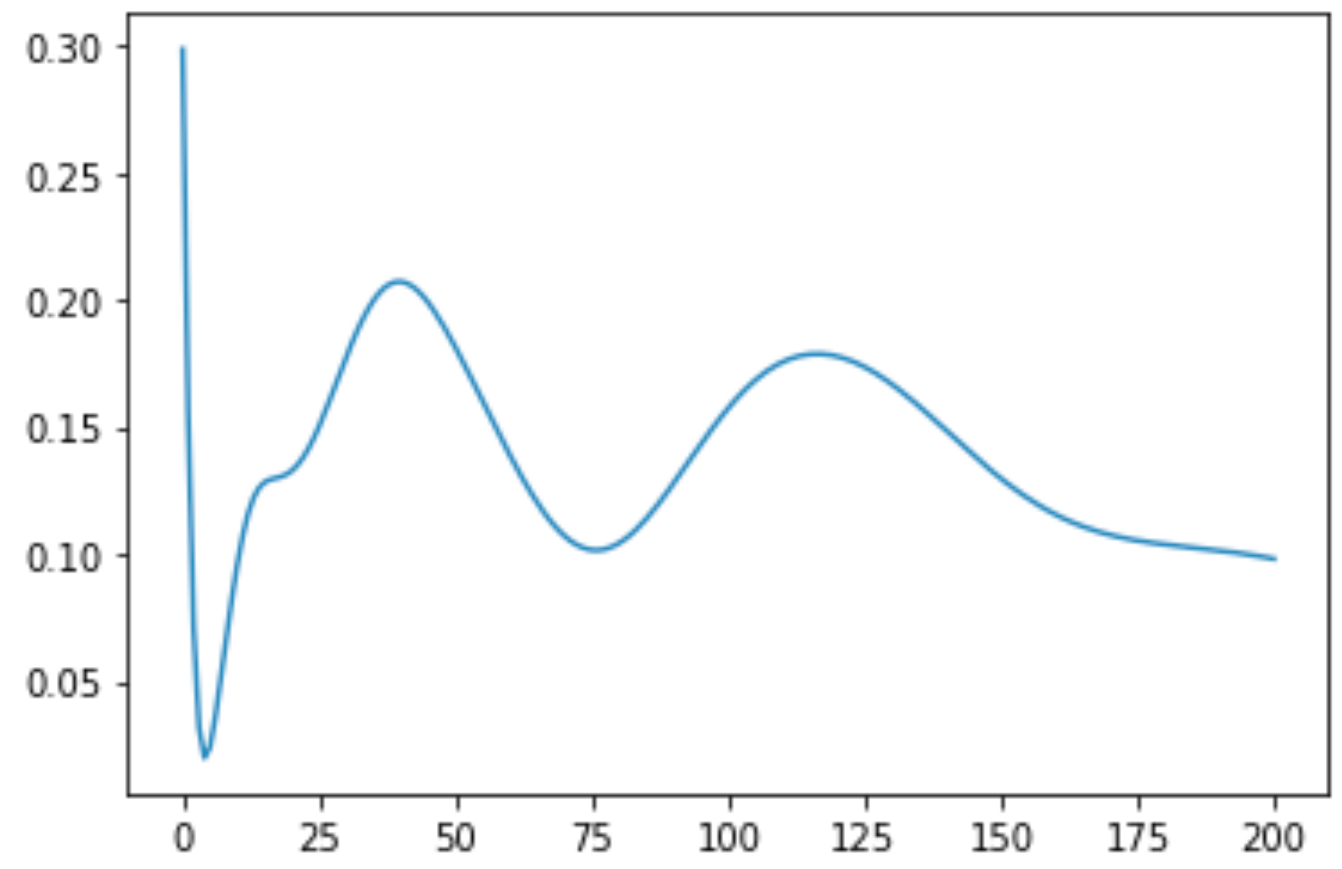
H. J. Sommers[a]
Fachbereich Physik, Universität-Gesamthochschule Essen, D-4300 Essen, Federal Republic of Germany
(Received 30 March 1988)

$$J_{ij} \sim \mathcal{N}(0, g^2/N) \quad \text{i.i.d.}$$

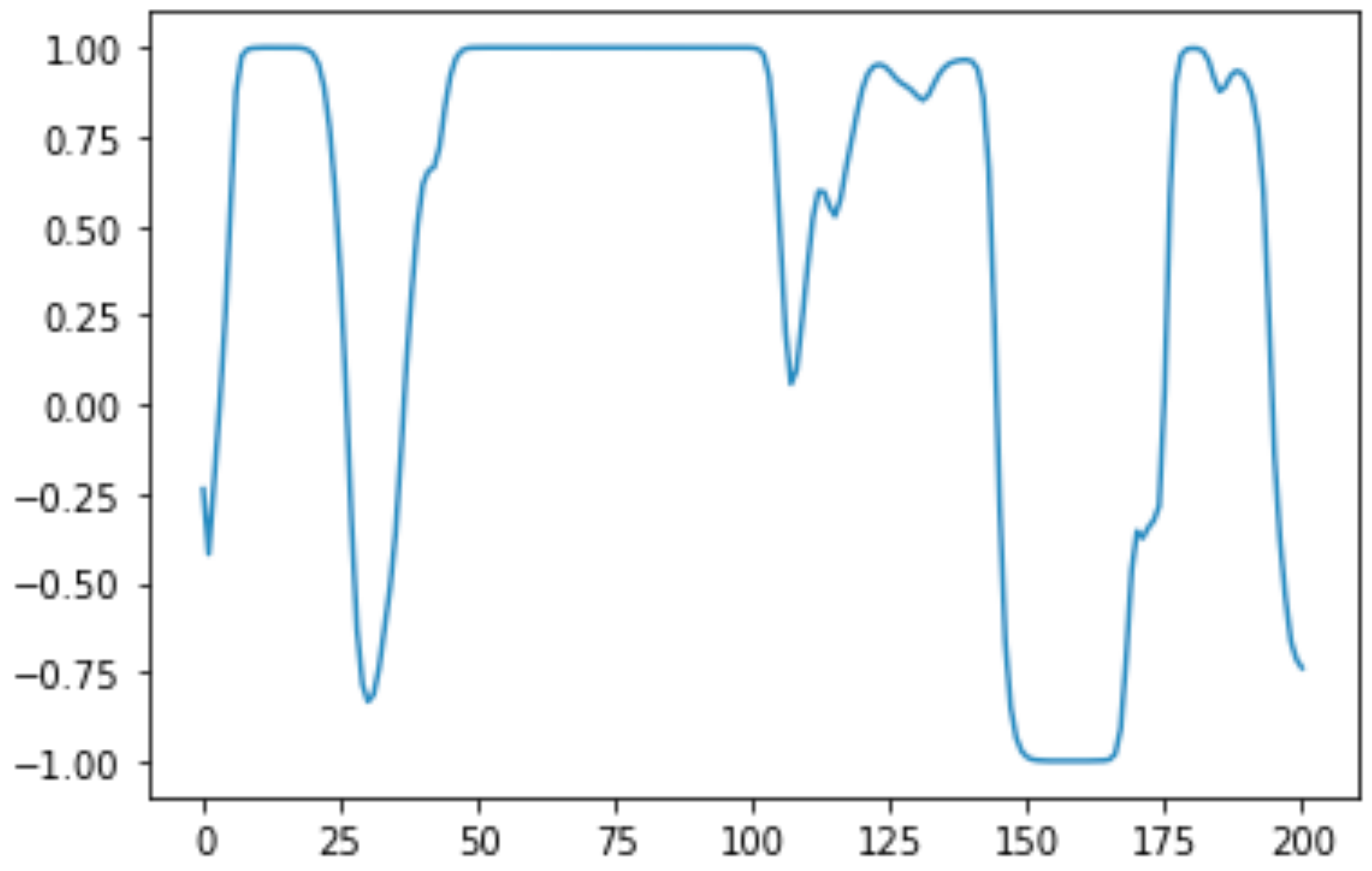
g=0.5



g=1.1



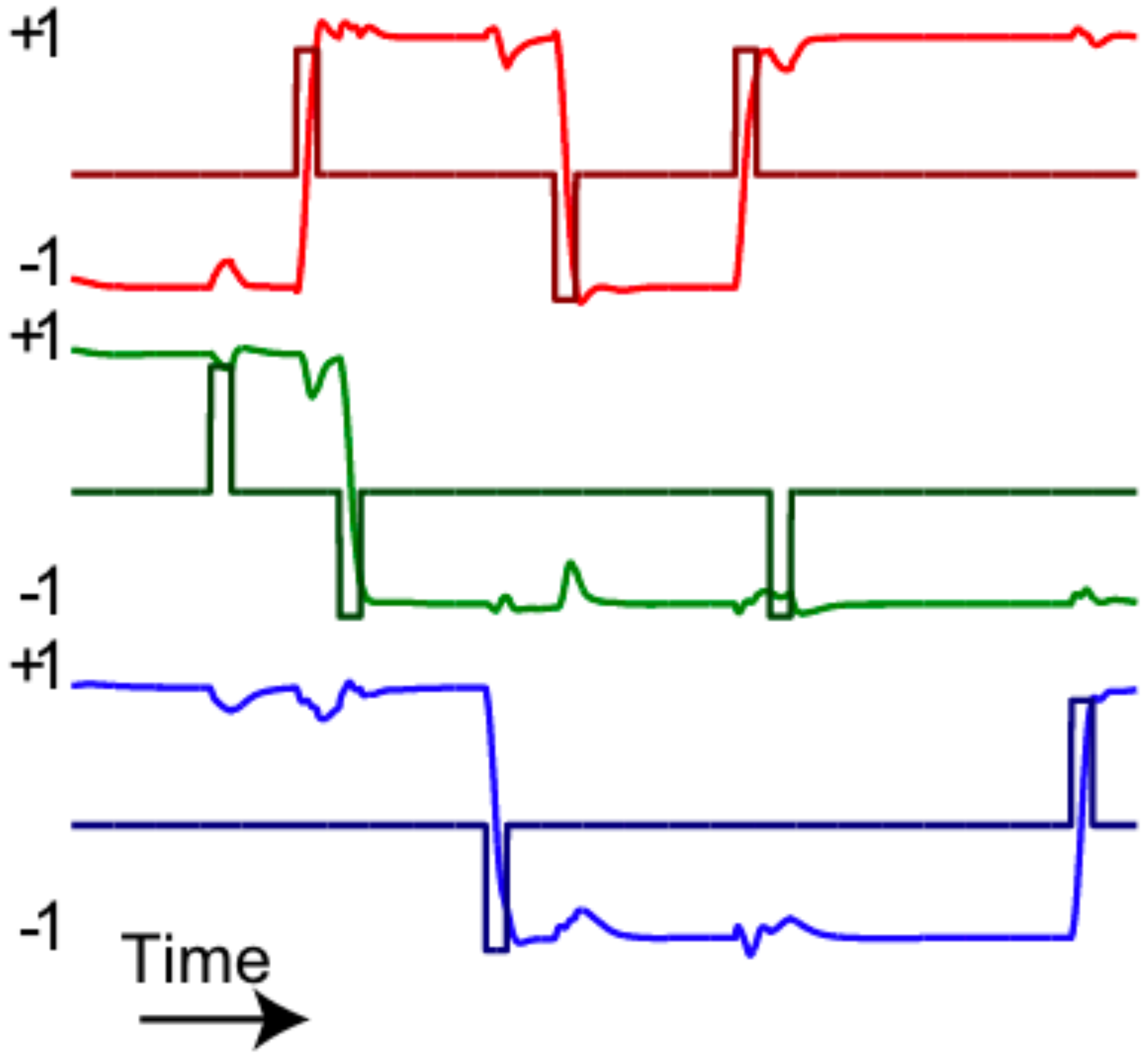
g=5

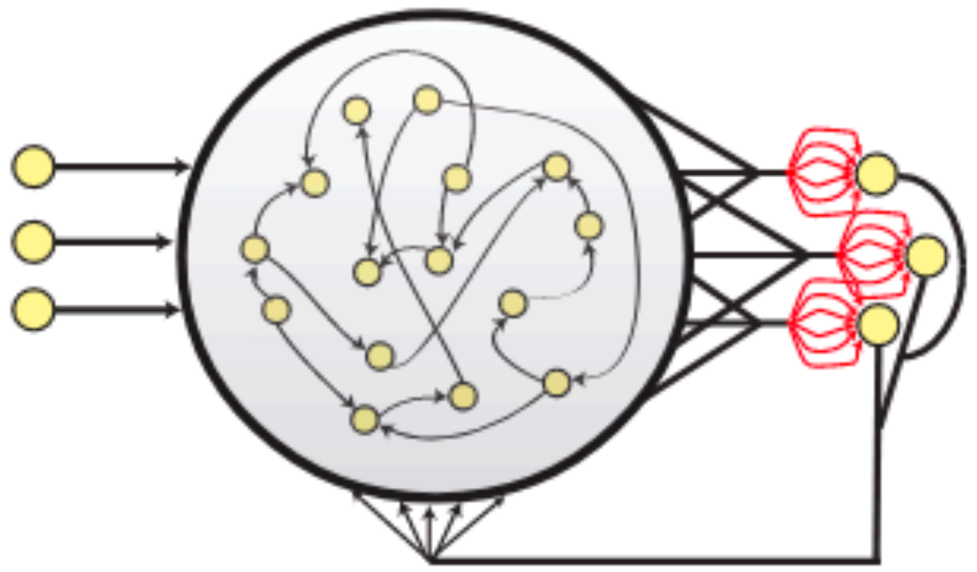


Reverse-engineering RNNs

[Barak & Sussillo, *Opening the black-box: low-dimensional dynamics in high-dimensional RNNs*, 2013]

Flip-flop task:





Reverse-engineering RNNs

[Barak & Sussillo, *Opening the black-box: low-dimensional dynamics in high-dimensional RNNs*, 2013]

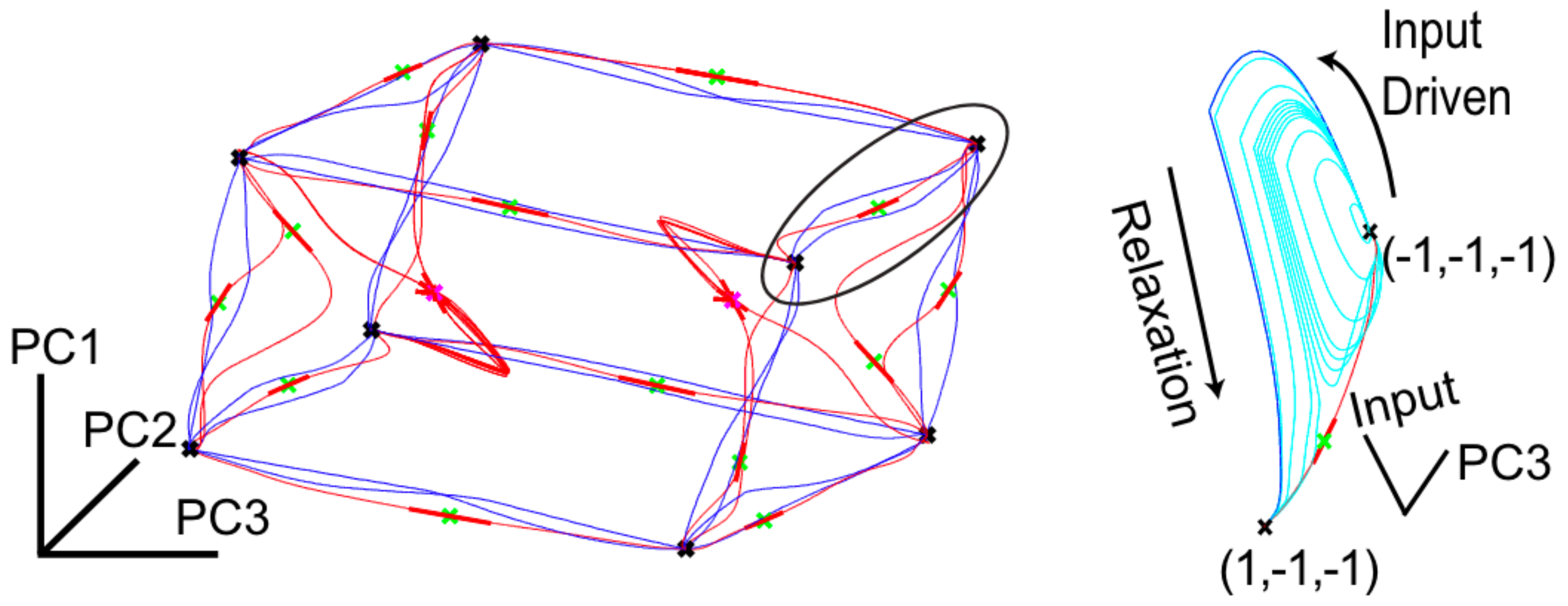
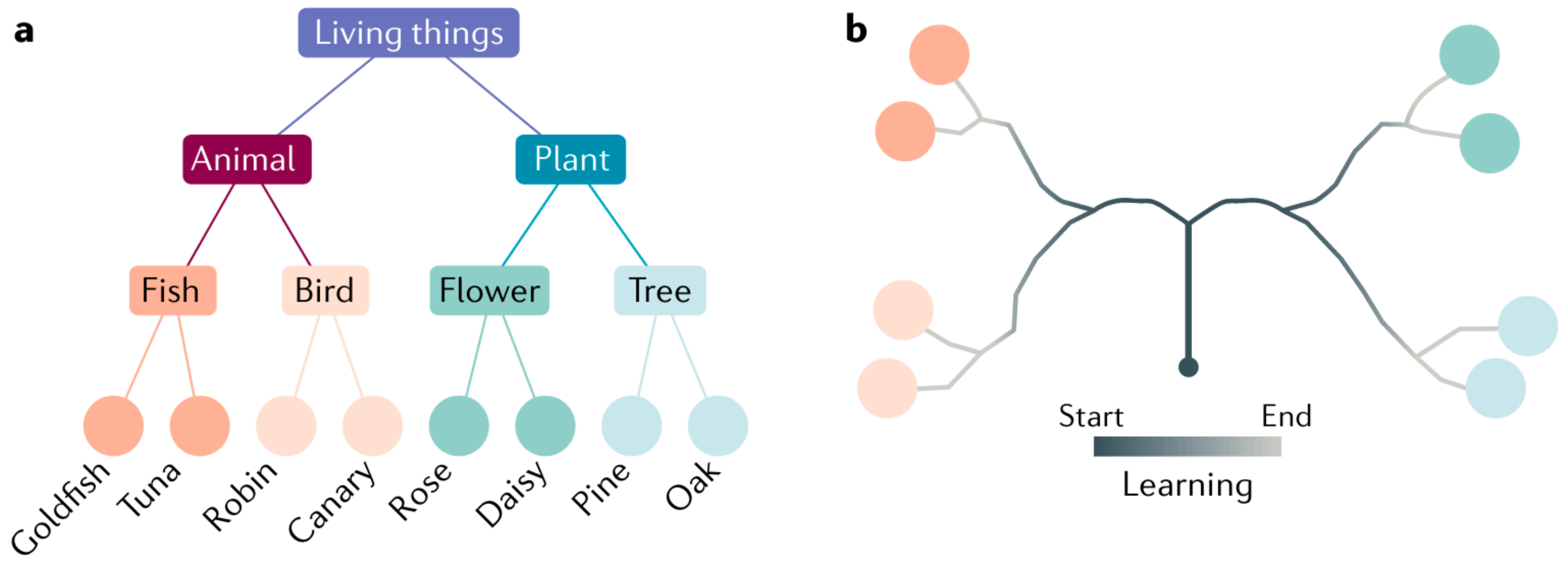


Figure 3: **Low-dimensional phase space representation of 3-bit flip-flop task.**

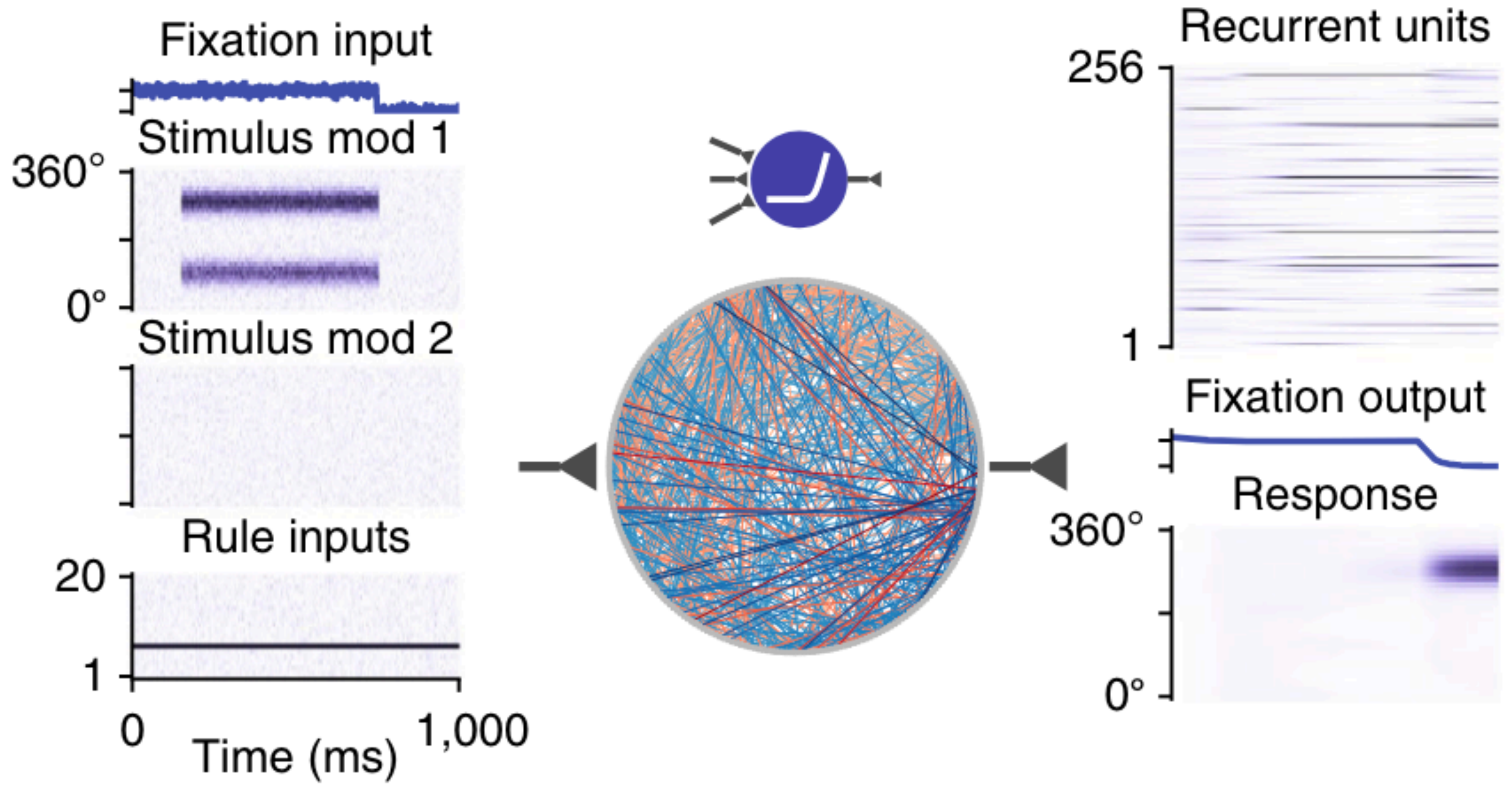
Ongoing challenges

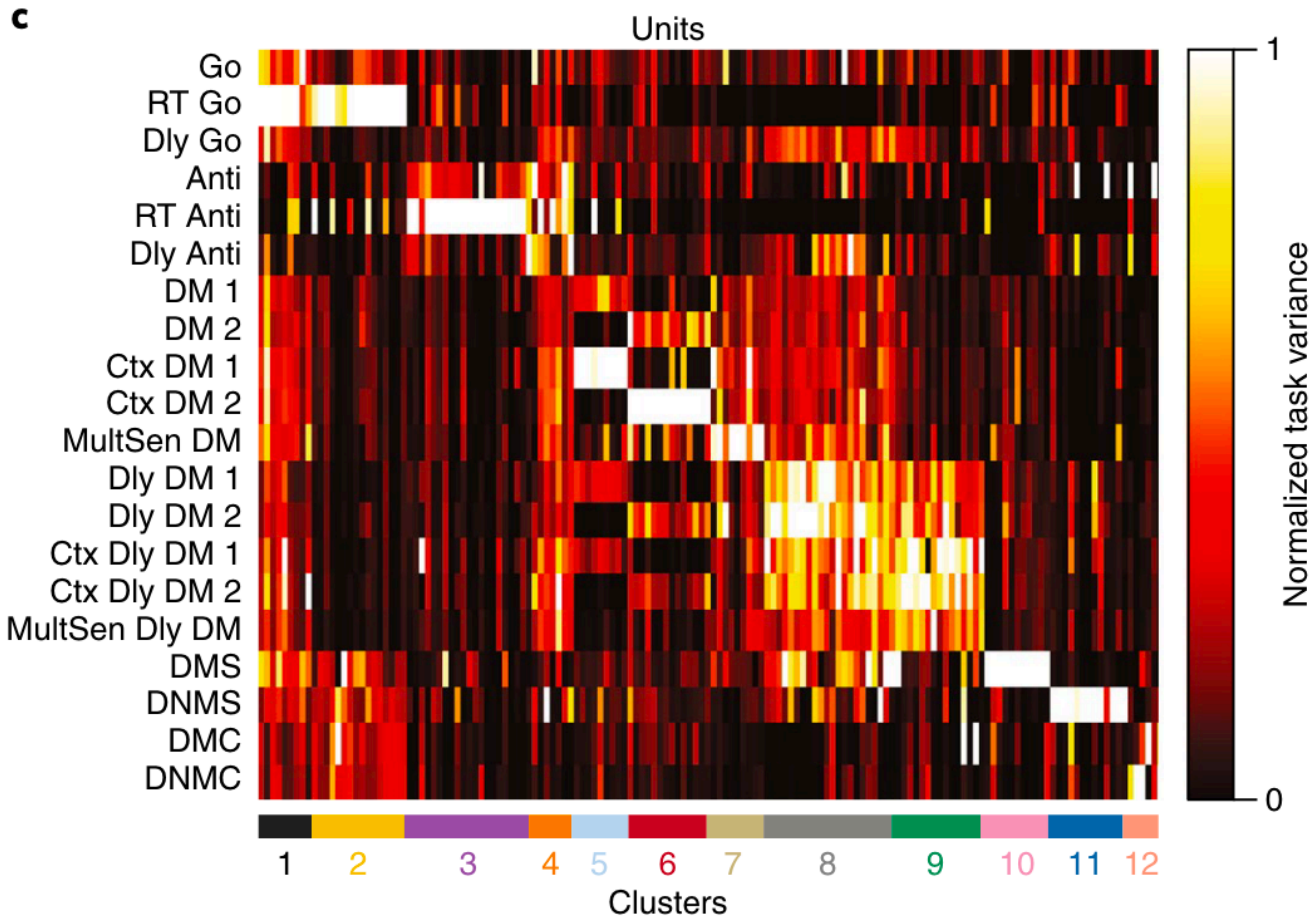
[Saxe et al., *If deep learning is the answer, what is the question?*, 2021]



Recent advances

[Robert Yang et al., *Task representations in neural networks trained to perform many cognitive tasks*, 2019]

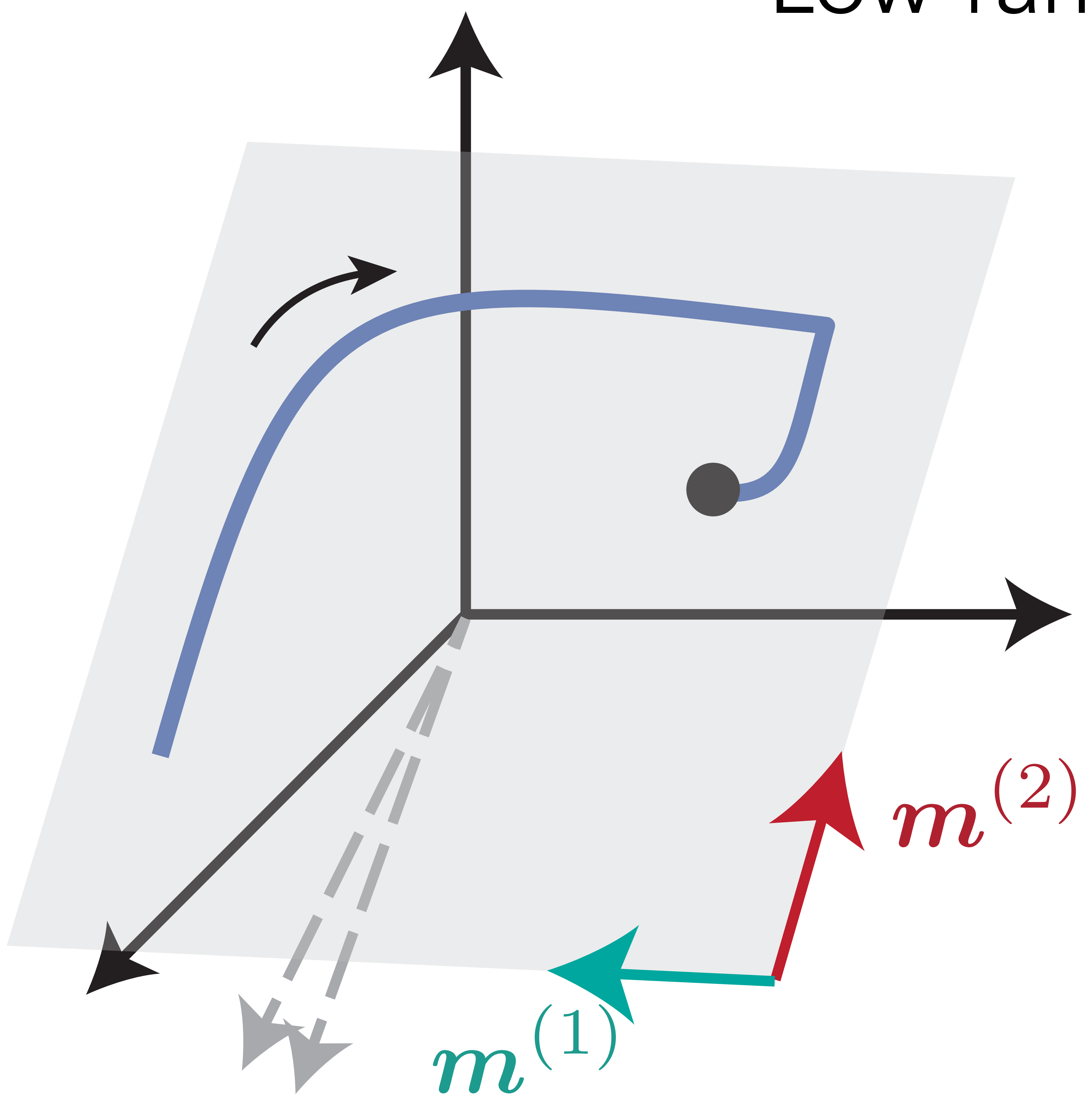


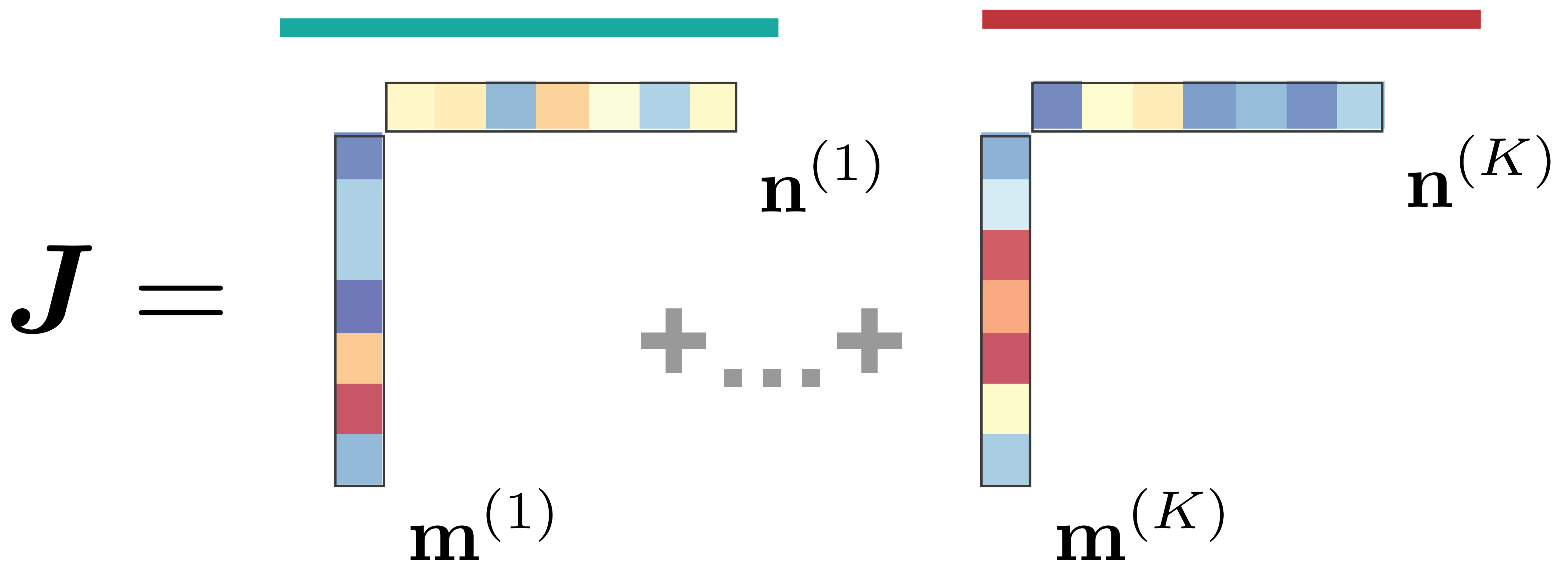


Recent advances

[Mastrogiuseppe & Ostojic 2018, Beiran et al. 2021, Dubreuil, Valente et al 2022]

Low-rank factorizations of connectivity





low-rank decomposition

Recent advances

[Mastrogiuseppe & Ostojic 2018, Beiran et al. 2021, Dubreuil, Valente et al 2022]

Low-rank factorizations of connectivity

$$\tau \dot{\boldsymbol{x}} = -\boldsymbol{x} + \boldsymbol{m}\boldsymbol{n}^T \phi(\boldsymbol{x}) + \boldsymbol{I}u(t)$$