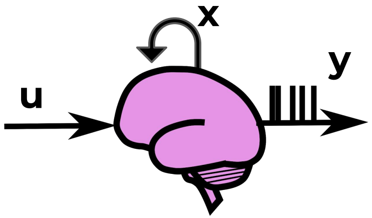
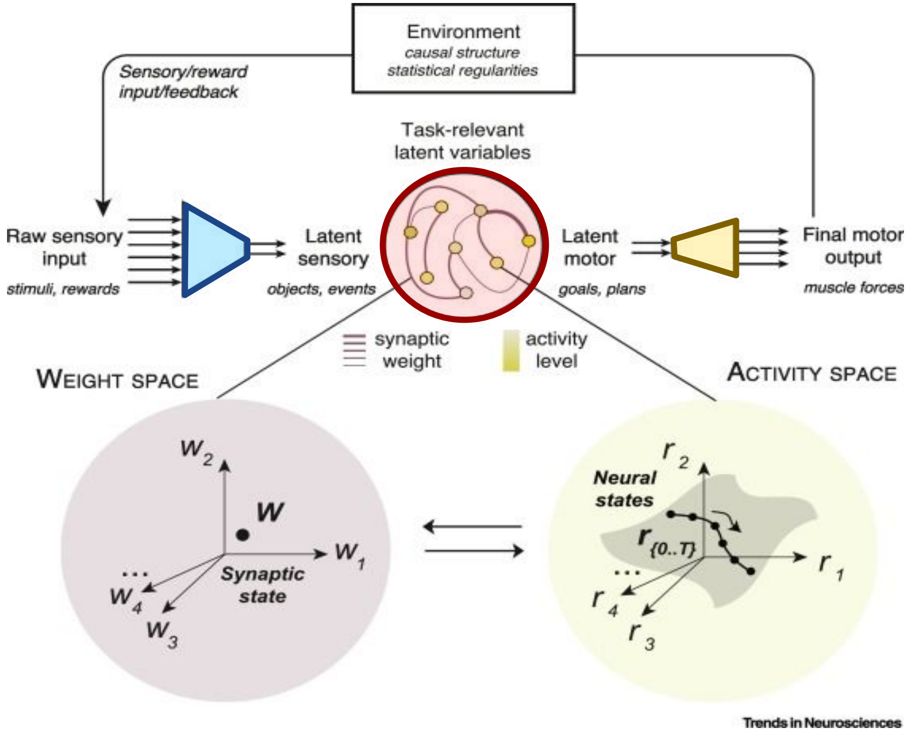
# Recurrent Neural Networks:

A model organism for studying computation over time in neuroscience



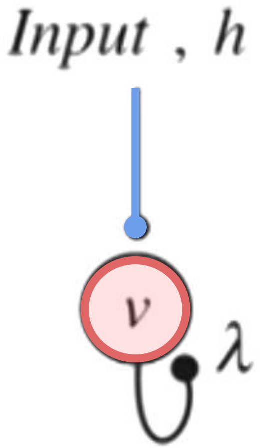




A Network Perspective on Sensorimotor Learning (2021) Sohn et al.

# **Simplest recurrent network:** a neuron with an self-synapse

$$\tau_n \frac{dv}{dt} = -v + h + \lambda v$$

$$\tau_n \frac{dv}{dt} = -(1-\lambda)v + h$$




- We examined three cases:

$\lambda < 1$ $\lambda = 1$ $\lambda > 1$

“Hopfield Networks - Intro to Neural Computation” Michale Fee

# RNNs as a description of firing rate dynamics in a network of neurons

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

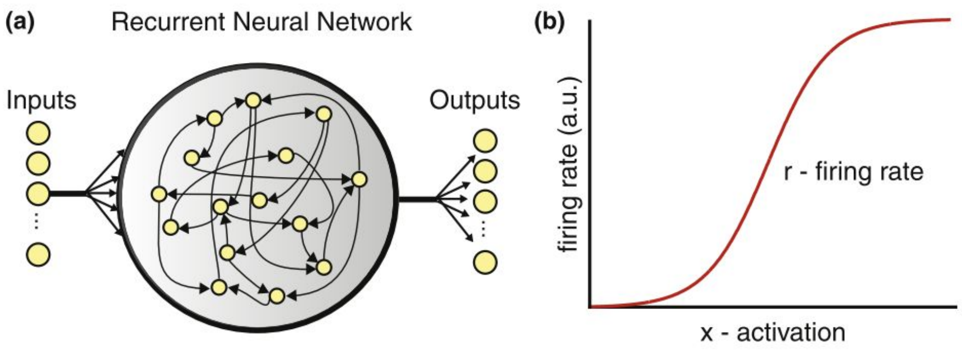
$$\tau\dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{J}\mathbf{r}(t) + \mathbf{B}\mathbf{u}(t) + b$$

*describes the network connections*

# RNNs as a description of firing rate dynamics in a network of neurons

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\tau\dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{J}\mathbf{r}(t) + \mathbf{B}\mathbf{u}(t) + b$$



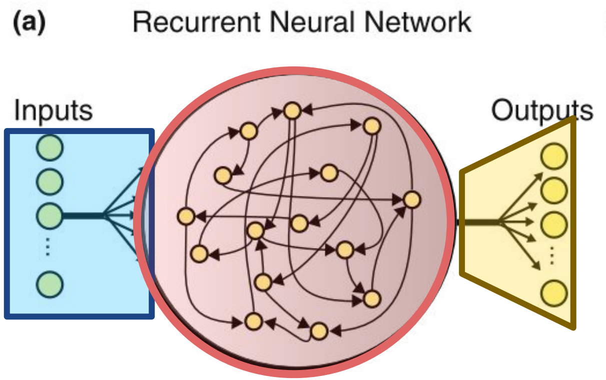
# RNNs as a description of firing rate dynamics in a network of neurons

$$\dot{\mathbf{x}} = \overbrace{\mathbf{A}\mathbf{x}} + \overbrace{\mathbf{B}\mathbf{u}}$$

$$\mathbf{y} = \overbrace{\mathbf{C}\mathbf{x}}$$

$$\tau\dot{\mathbf{x}}(t) = \overbrace{-\mathbf{x}(t) + \mathbf{J}\mathbf{r}(t)} + \overbrace{\mathbf{B}\mathbf{u}(t) + b}$$

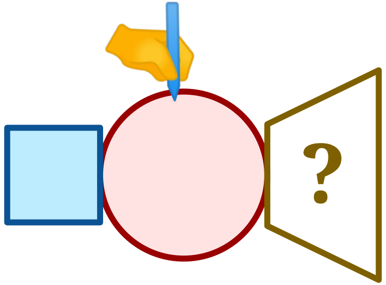
$$\mathbf{z}(t) = \overbrace{\mathbf{W}\mathbf{r}(t)}$$



# For much of the history of neuroscience..

Network models have been constructed **“by hand”** [Wang2018]

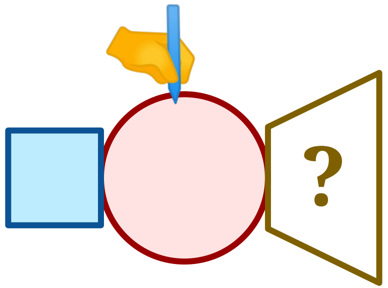
- Often with parameters drawn from distributions



# For much of the history of neuroscience..

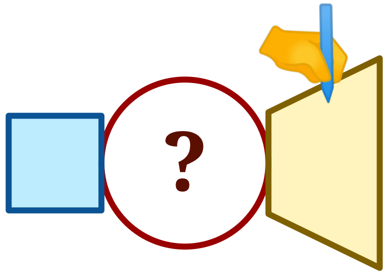
Network models have been constructed **“by hand”** [Wang2018]

Often with parameters drawn from distributions




**Key Twist:** Recent work has focused on optimizing RNNs (not for structural realism) but instead to **accomplish some task**

- e.g. integration of inputs
- Often some degree of “hand-tuning” & hyper parameter adjustment happens after



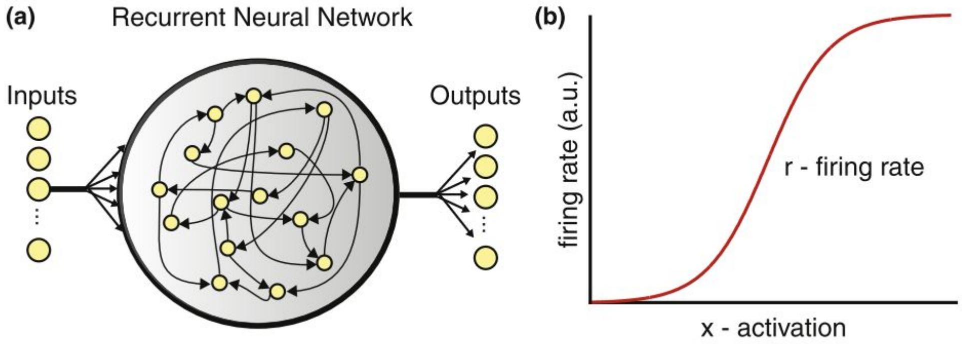
# Why Study RNNs?

Have brain-like features

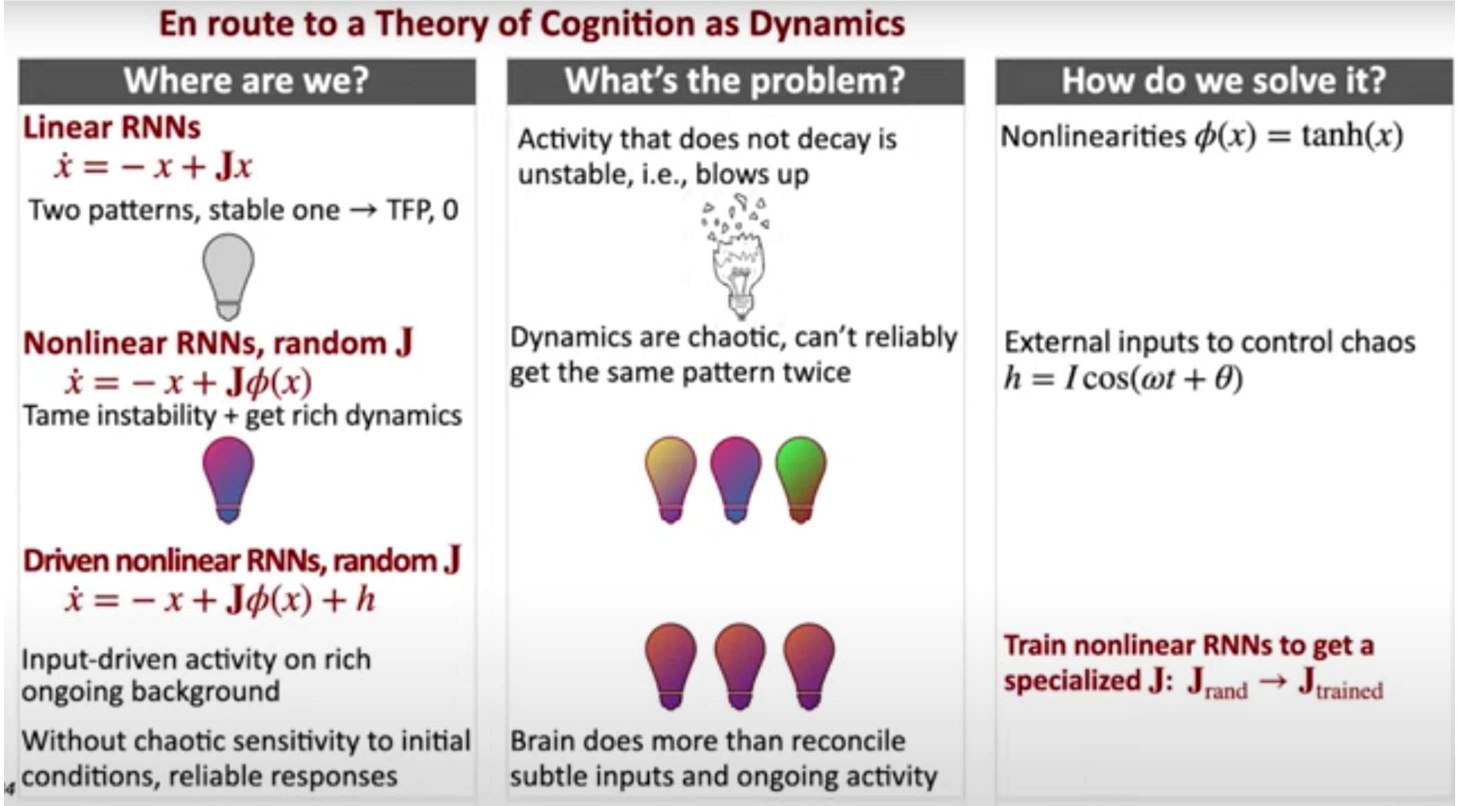
- Feedback
- Nonlinearity
- Distributed computing

Can perform complex tasks

Can be analyzed in detail

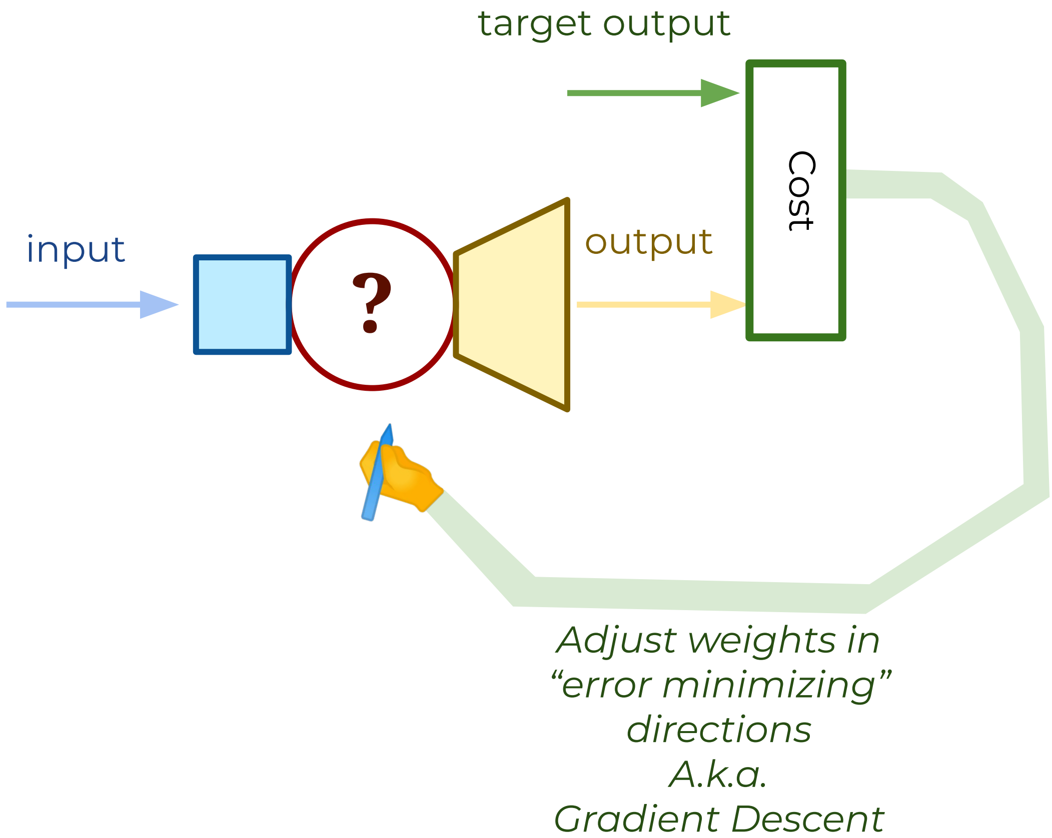


# Why Study (this specific form of) RNNs?

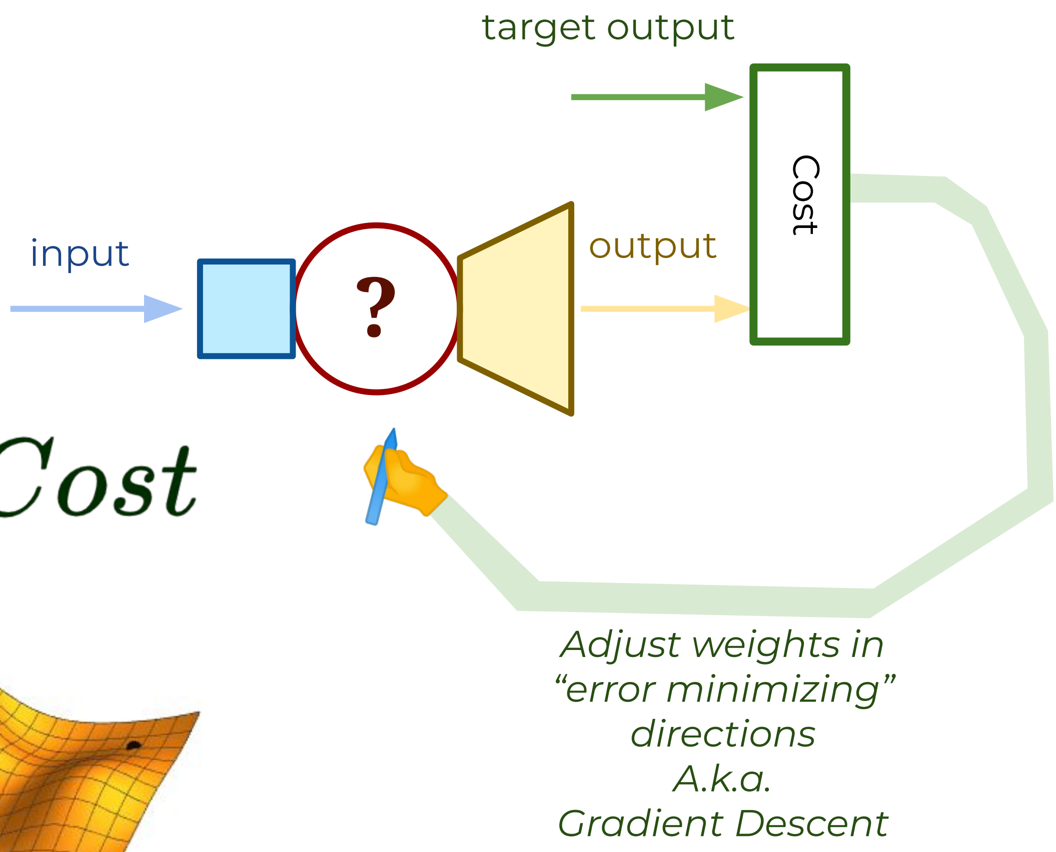



"COSYNE 2021 - Tutorial on RNN (Part 1)" by Kanaka Rajan

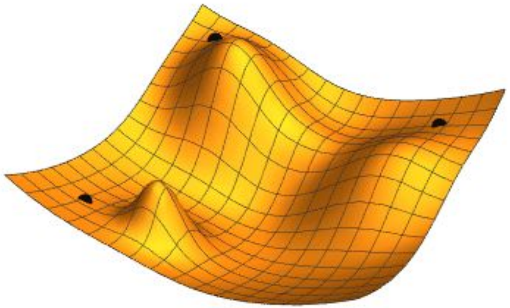
# Training an ANN



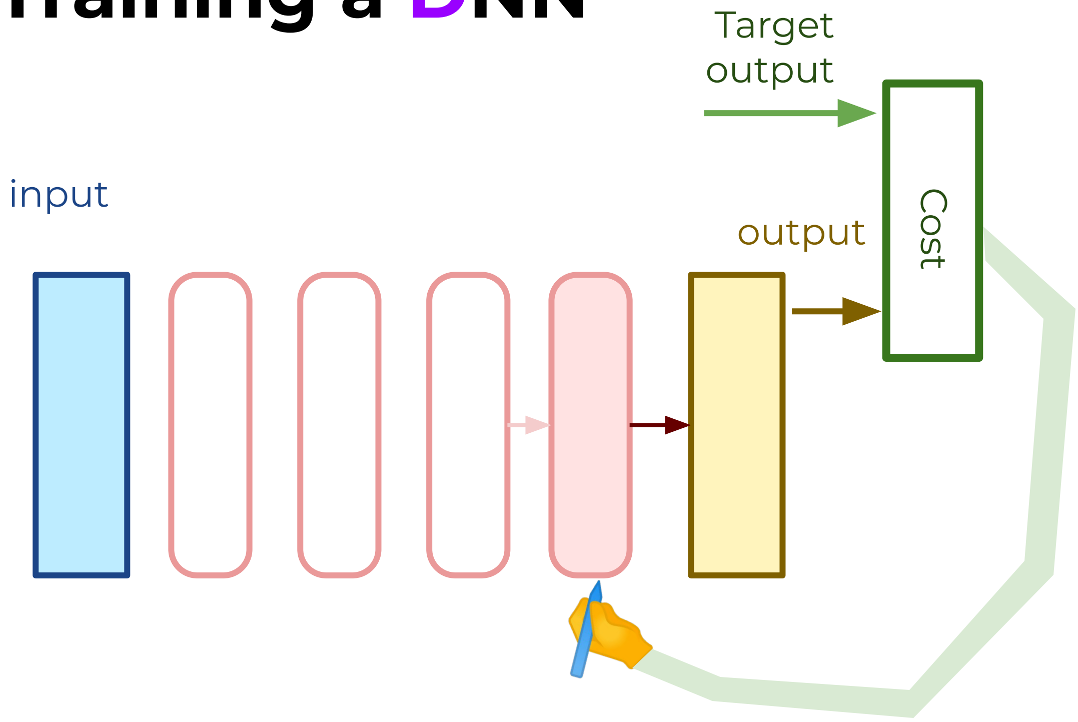
# Training an ANN




$$w \leftarrow w - \eta \nabla_w Cost$$



# Training a DNN

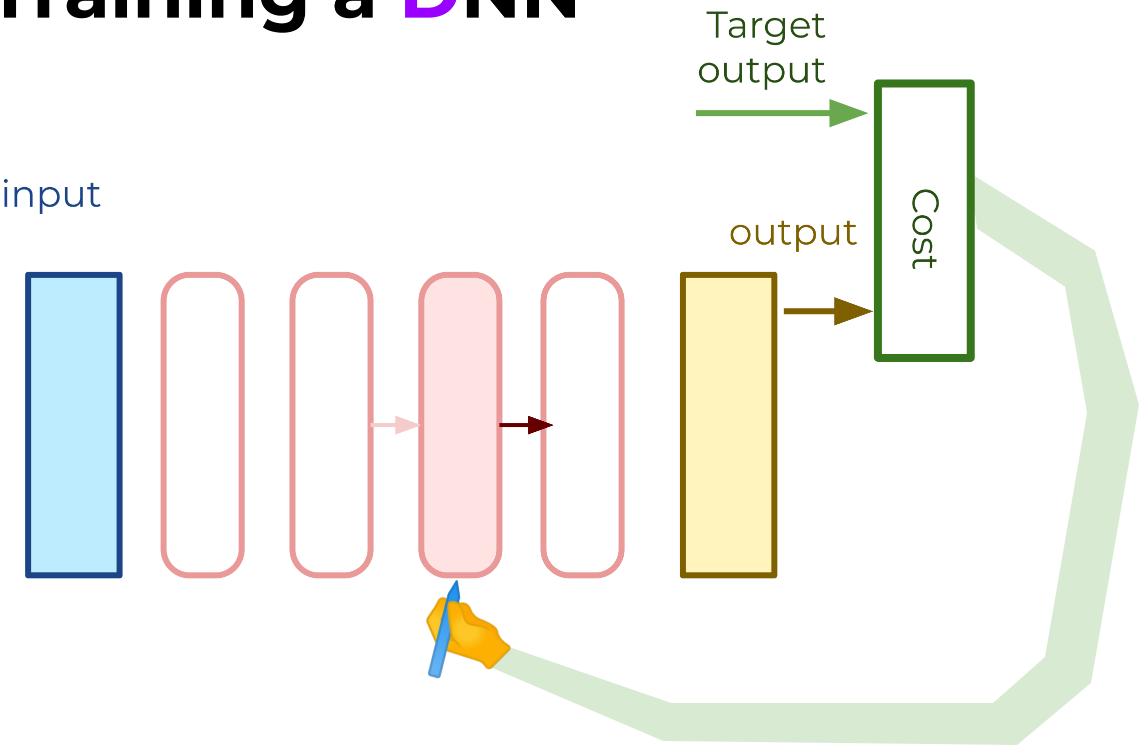



For each layer:

$$\mathbf{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

Backpropagation (through layers)

# Training a DNN

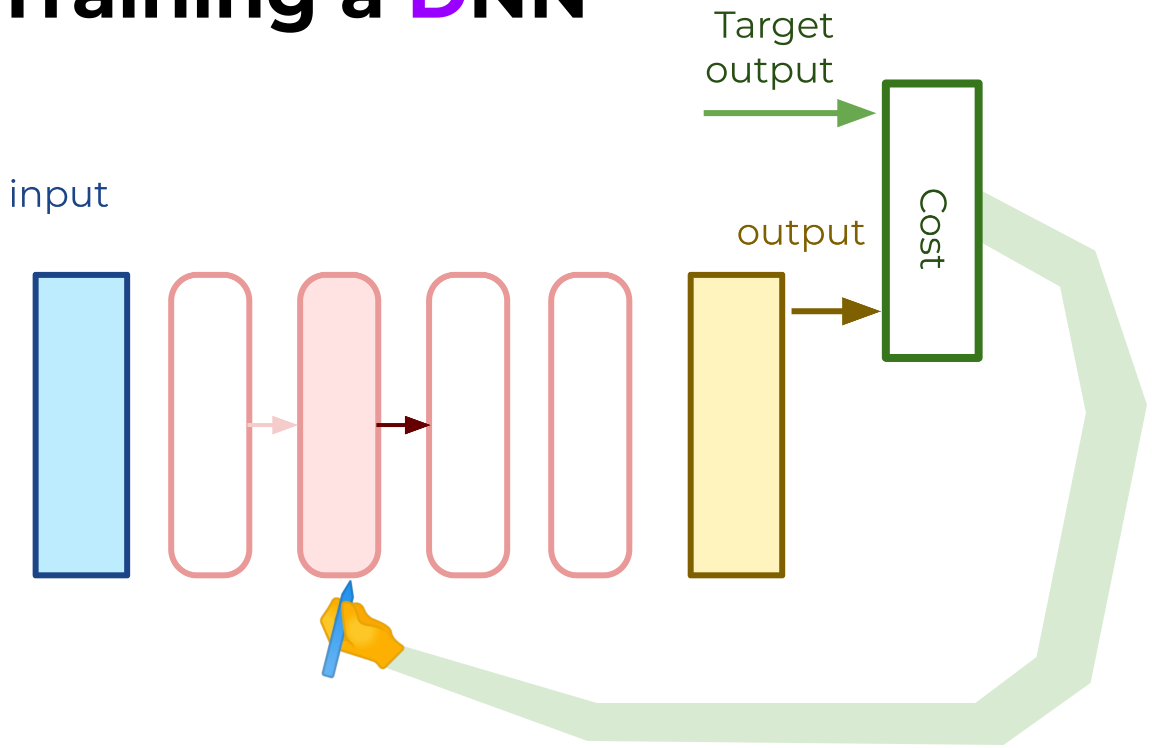



For each layer:

$$\mathbf{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

Backpropagation (through layers)

# Training a DNN

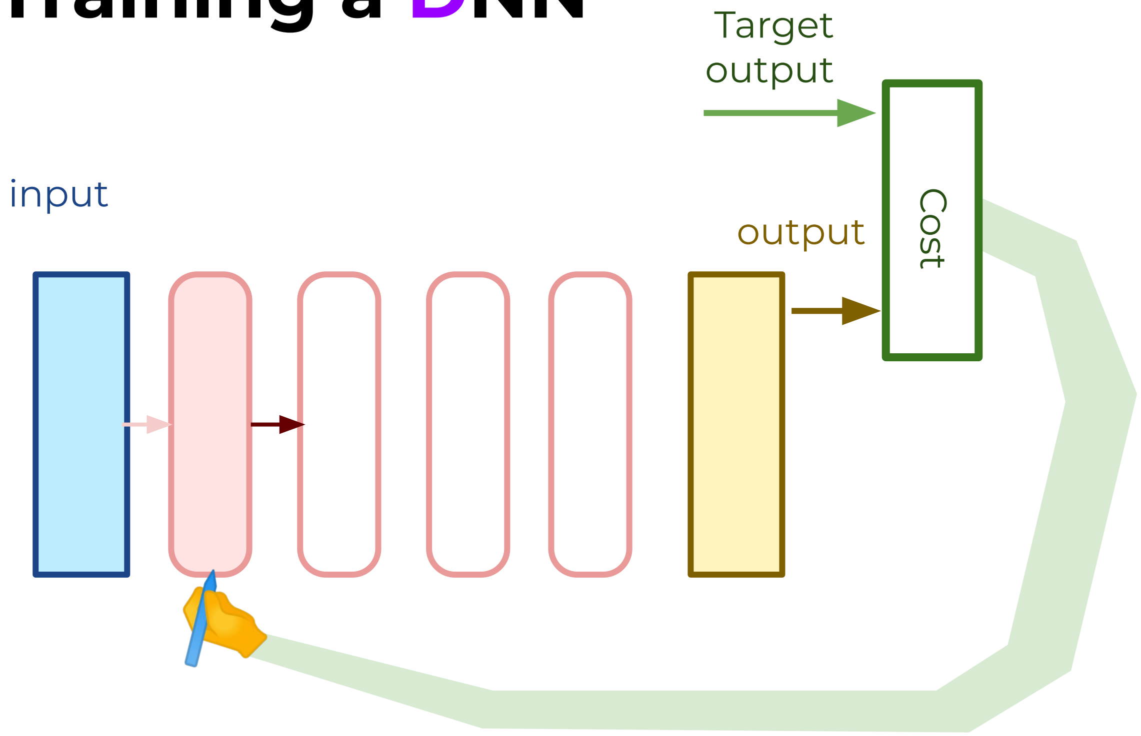



For each layer:

$$\mathbf{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

Backpropagation (through layers)

# Training a DNN




For each layer:

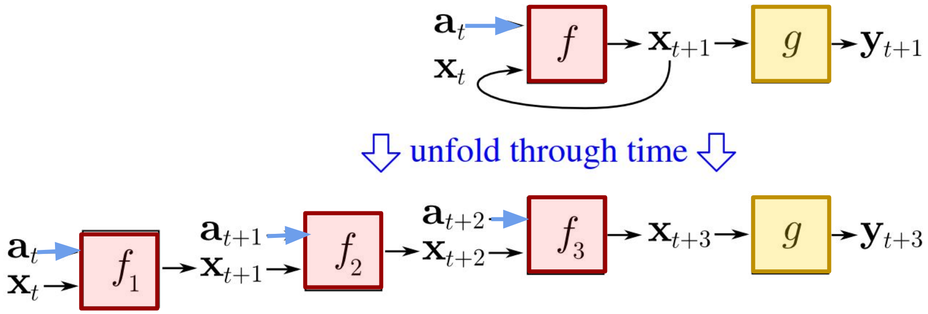
$$\mathbf{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

Backpropagation (through layers)

# Training an RNN

How to deal with each sample depending on the previous sample?

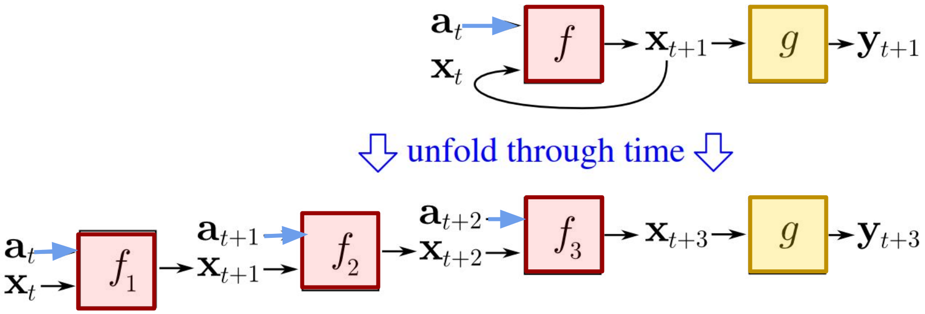
- "Unroll" network over time
- Then treat this like a DeepNN



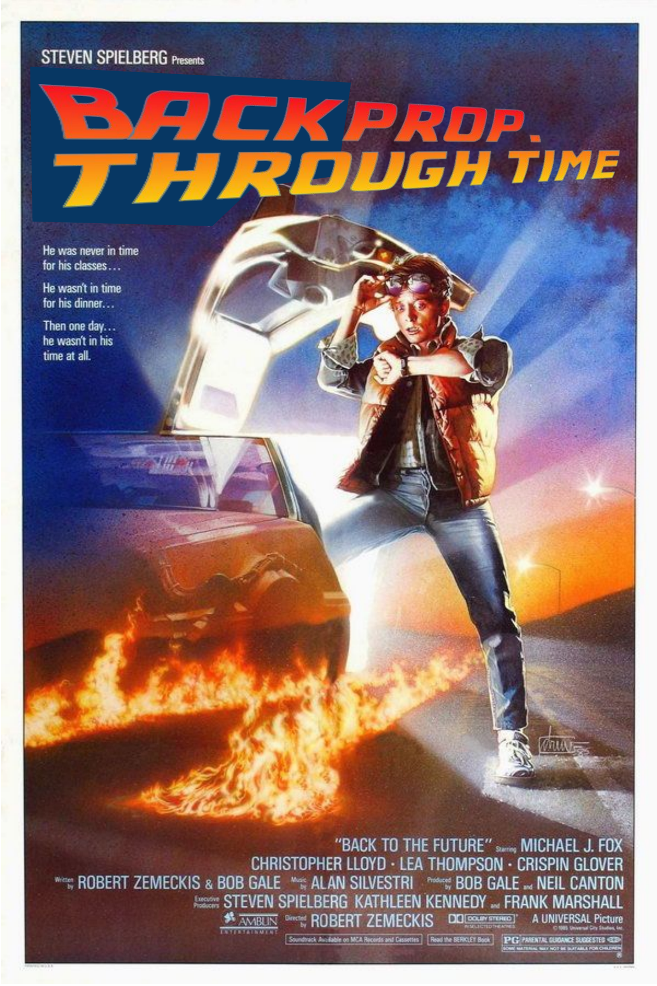

"Backpropagation through time" Wikipedia

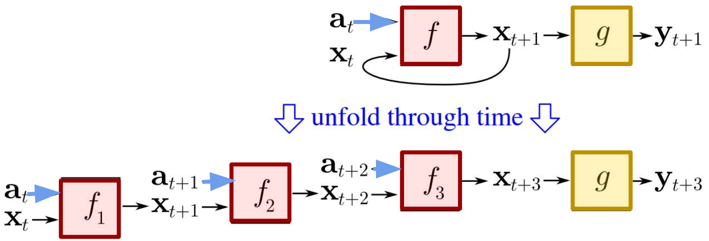
# Training an RNN

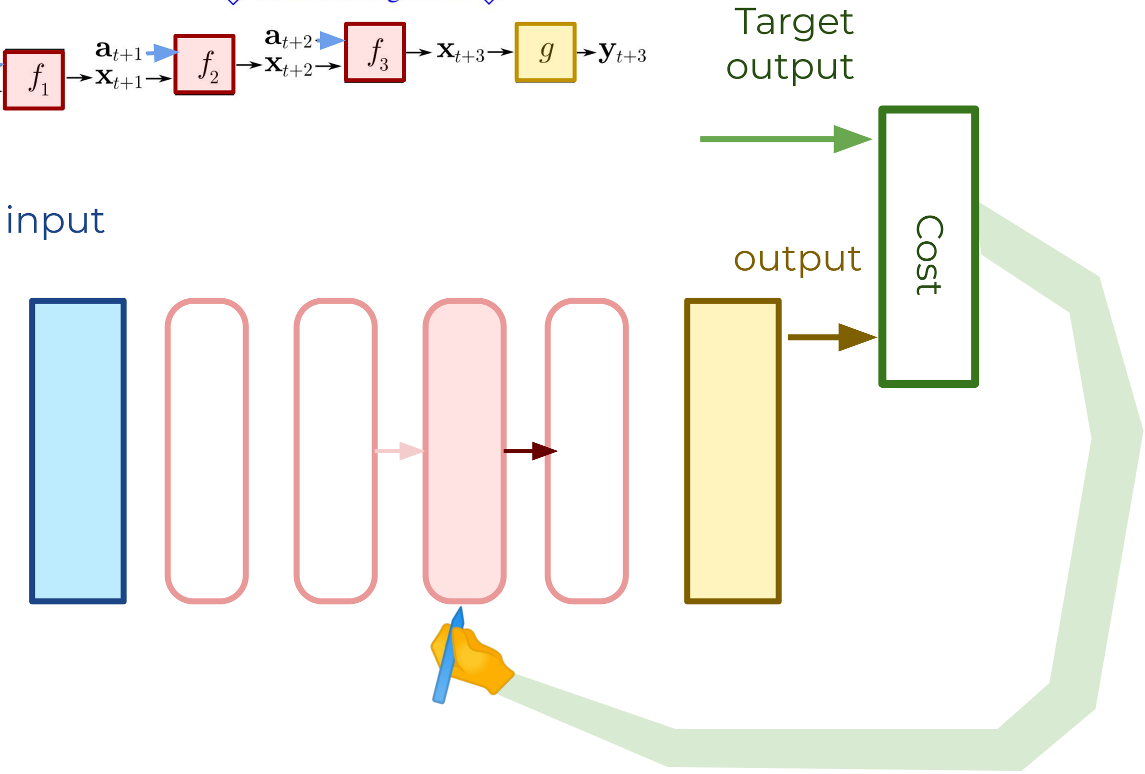
Backpropagation (**through time**)

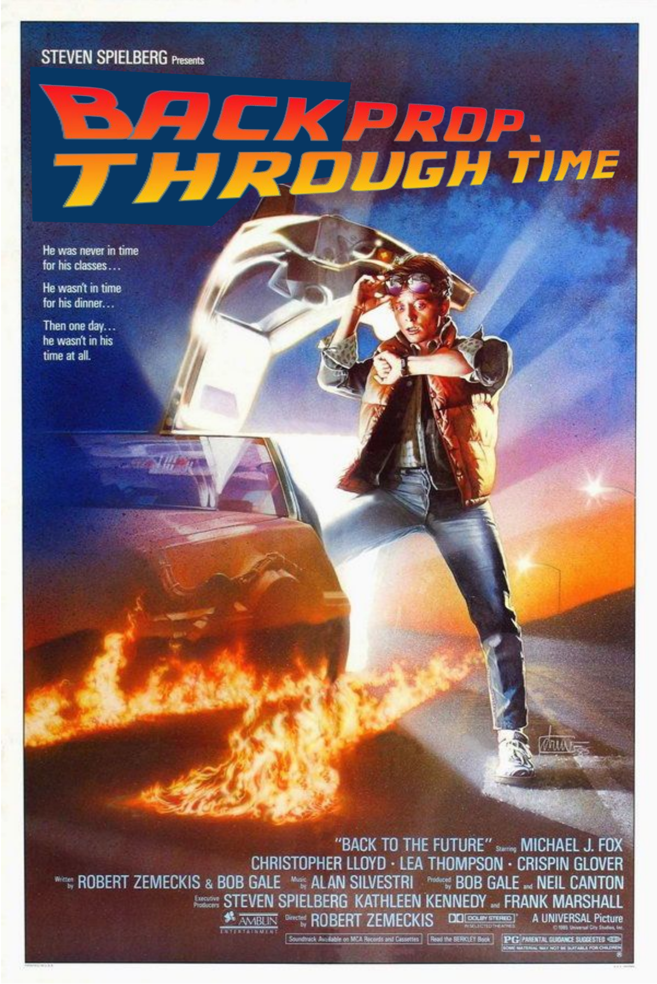



"[Backpropagation through time](https://en.wikipedia.org/wiki/Backpropagation_through_time)" Wikipedia



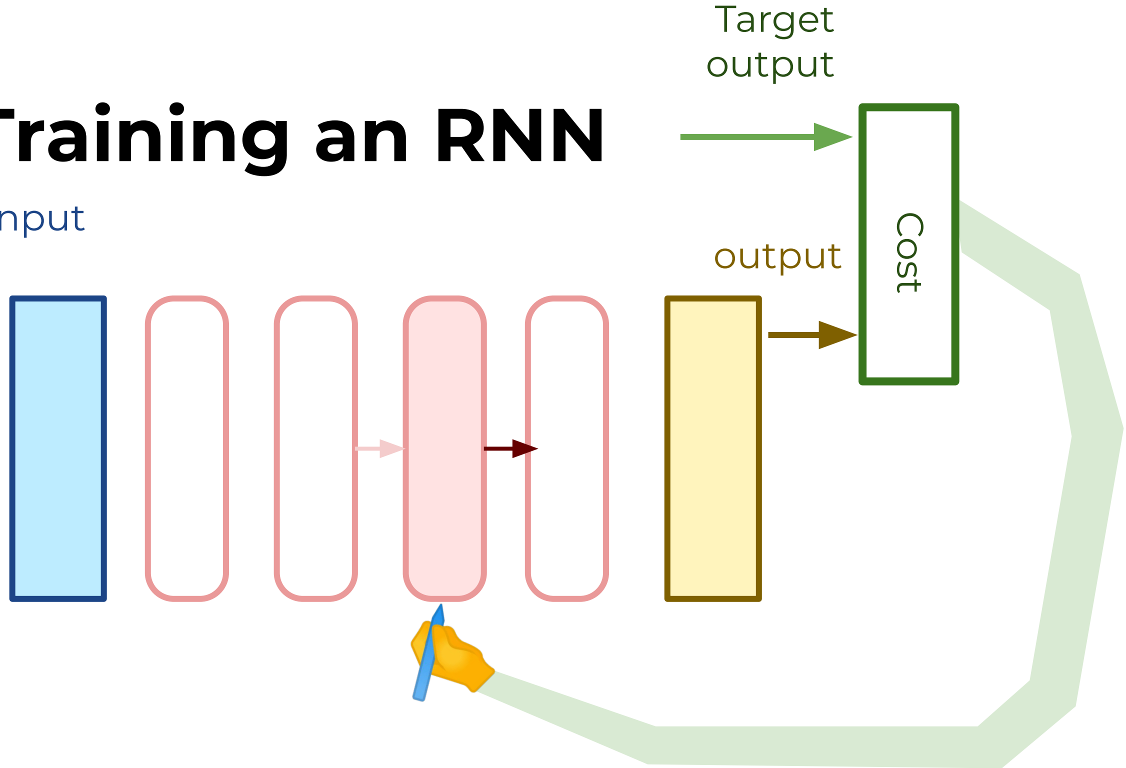


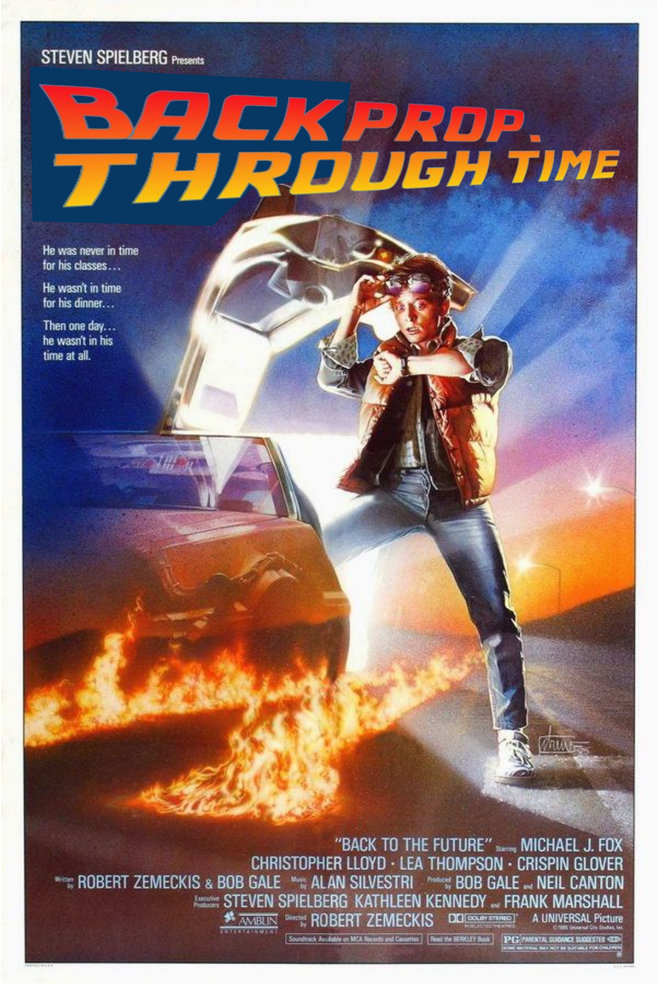




# Backpropagation (**through time**)

# Training an RNN

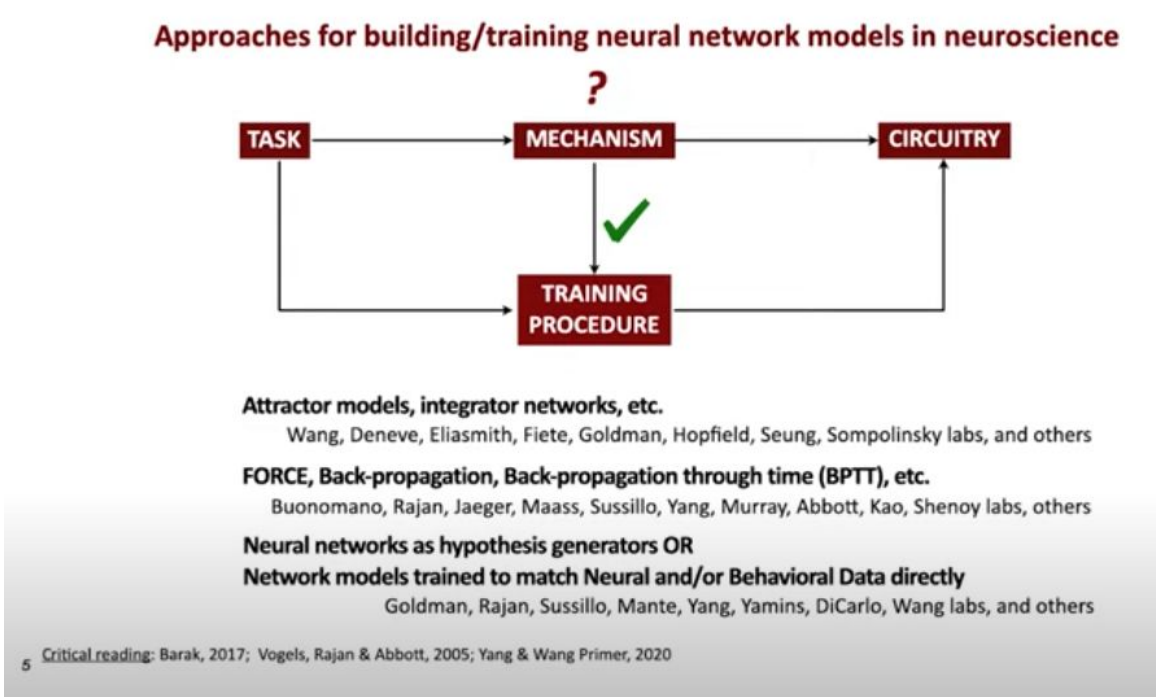



Backpropagation (**through time**)

See also:
"What is backpropagation really doing?" 3Blue1Brown
"Backpropagation through time: what it does and how to do it" (1990) Werbos

# **Training an RNN:** Further Reading



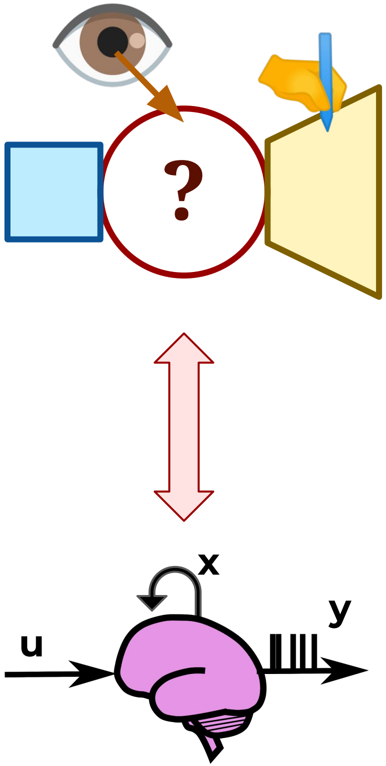

See also:

AutoLFADS -
"A large-scale neural network training framework for generalized estimation of single-trial population dynamics" (2021) Keshtkaran

# RNNs for hypothesis generation

By interrogating the structure of a task-optimized RNN  we can generate hypotheses :

- Suggest (novel? unexpected?) mechanisms for neural circuits accomplishing tasks
- Suggest new experiments & analysis
- Become a testing ground for analysis

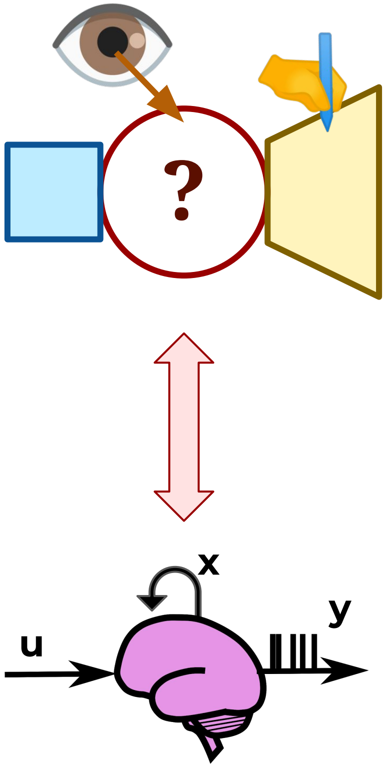



See
"[Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks.](#)" (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what's going on inside an RNN

# How to use RNNs to inpret how it accomplishes tasks

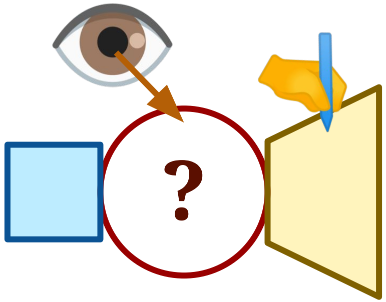
Interrogate & interpret through the lens of dynamical systems

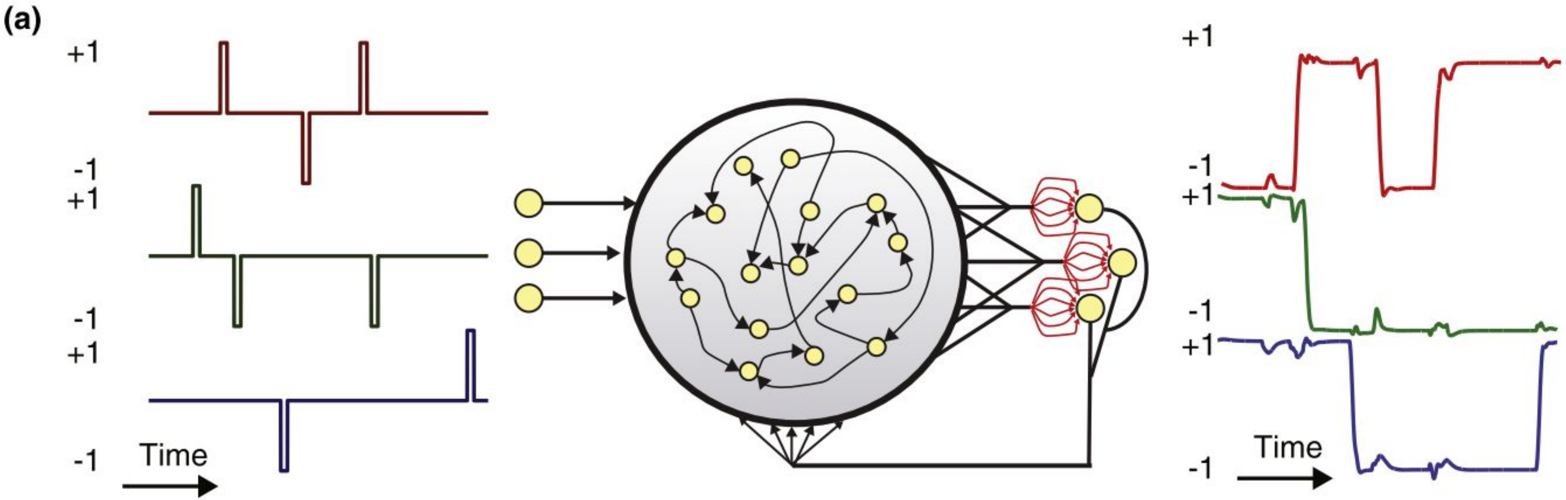
- Find fixed points
- Locally linearize
- Categorize behavior (by eigenvalues!)
- Fixed points form a “dynamical skeleton”




See
“Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks.” (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what’s going on inside an RNN

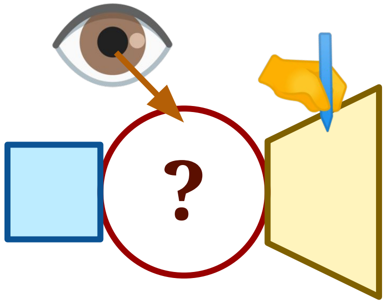
# Interrogate & interpret through the lens of dynamical systems

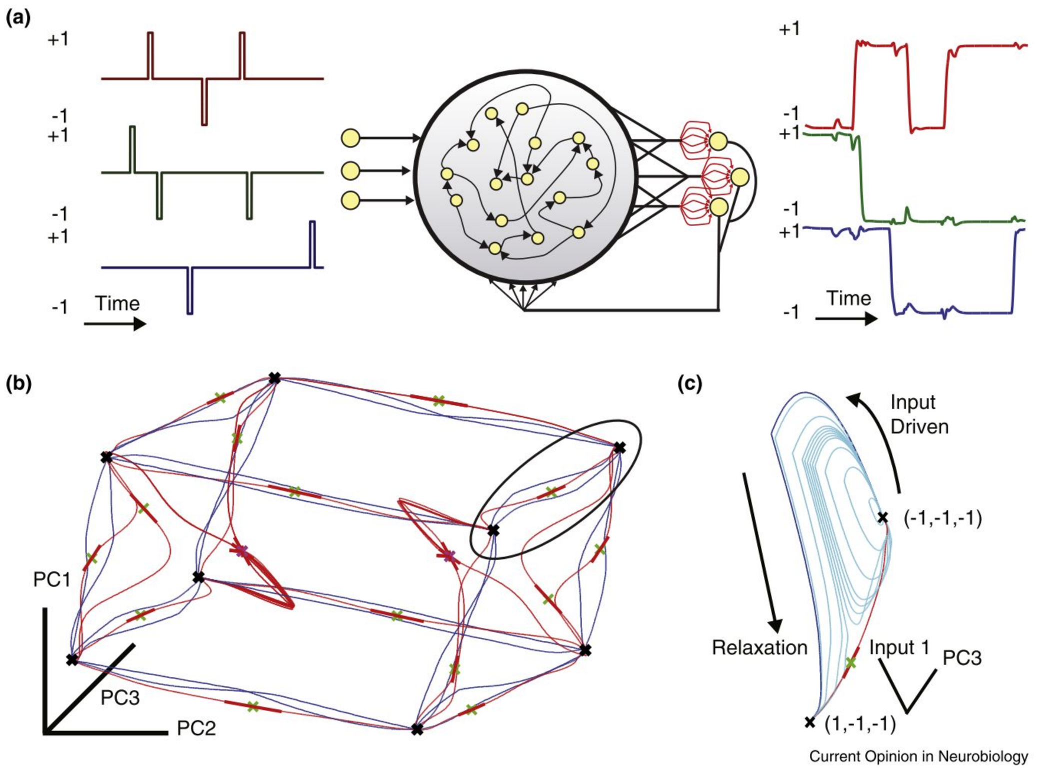



See
"Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks." (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what's going on inside an RNN

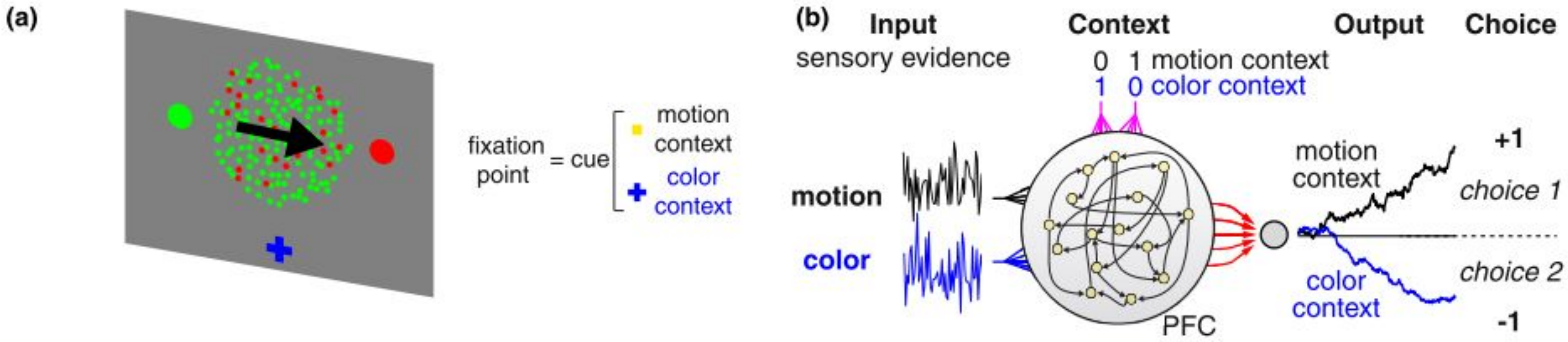
# Interrogate & interpret through the lens of dynamical systems





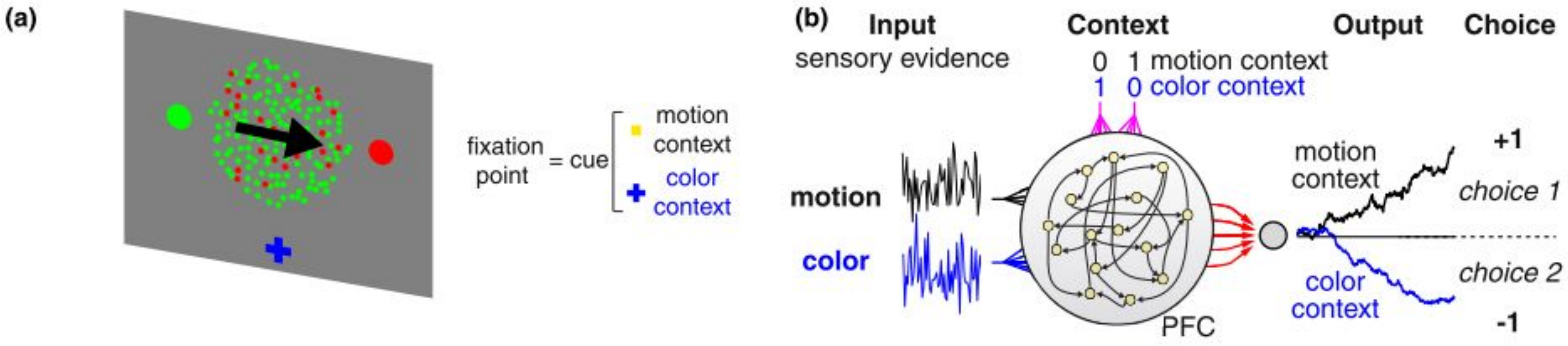

See
"[Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks.](#)" (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what's going on inside an RNN

# Revisiting context-dependent evidence accumulation




"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

# Revisiting context-dependent evidence accumulation

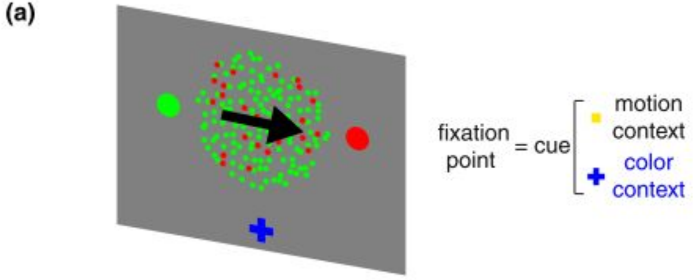


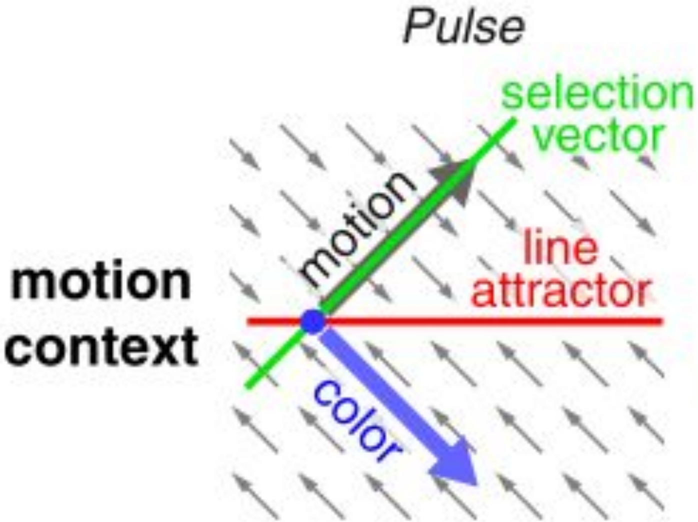

**Prevailing theory:** Top-down attention “gates” representation of features in cortex

***Let’s ask RNNs what they do!***

"[Context-dependent computation by recurrent dynamics in prefrontal cortex]" (2013) Mante, Sussilo et al.

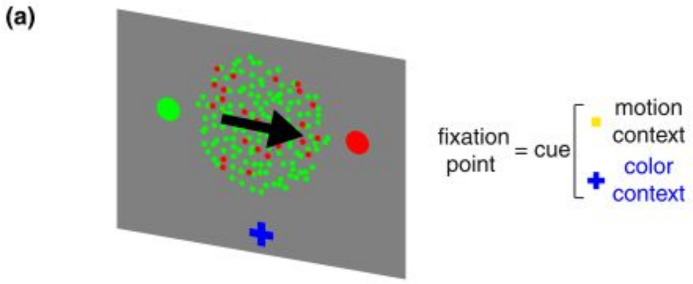
# RNN selects which features to accumulate

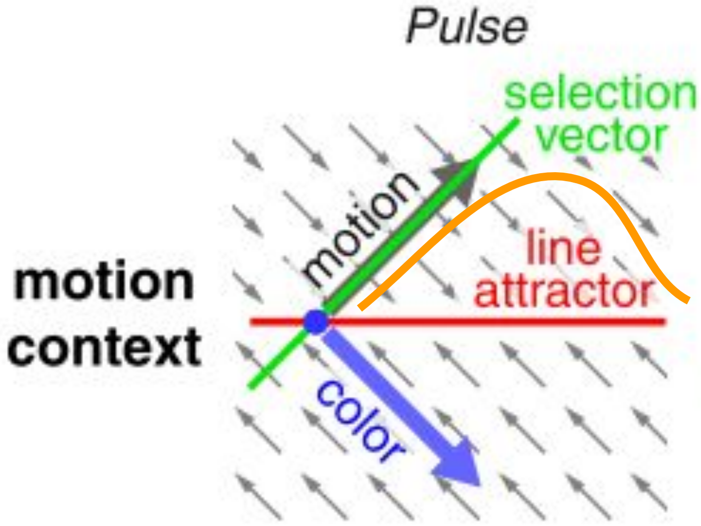



"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

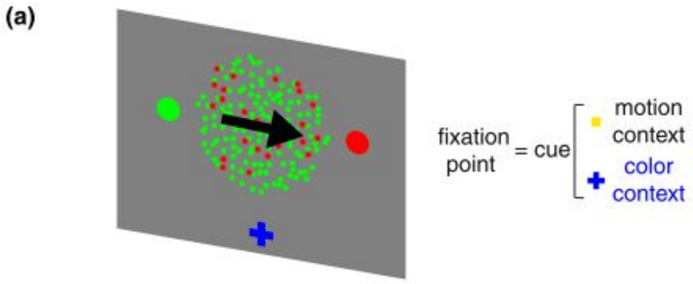
# RNN selects which features to accumulate

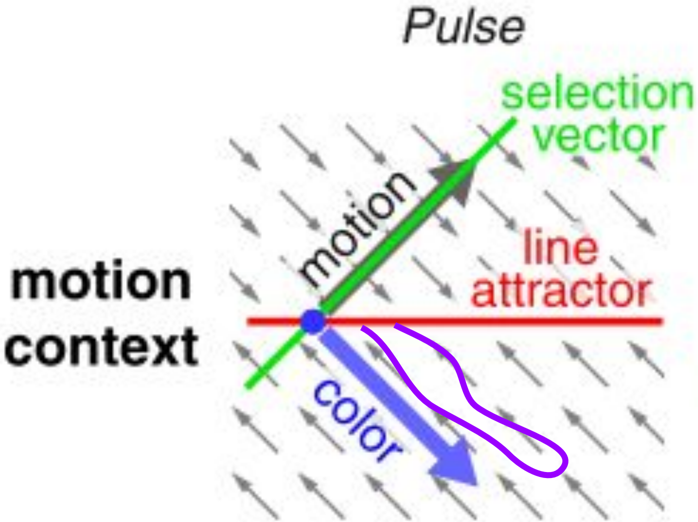



"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

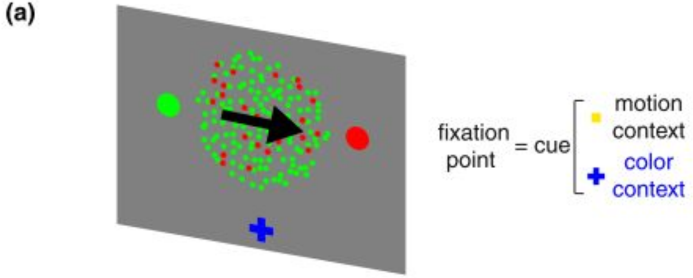
# RNN selects which features to accumulate

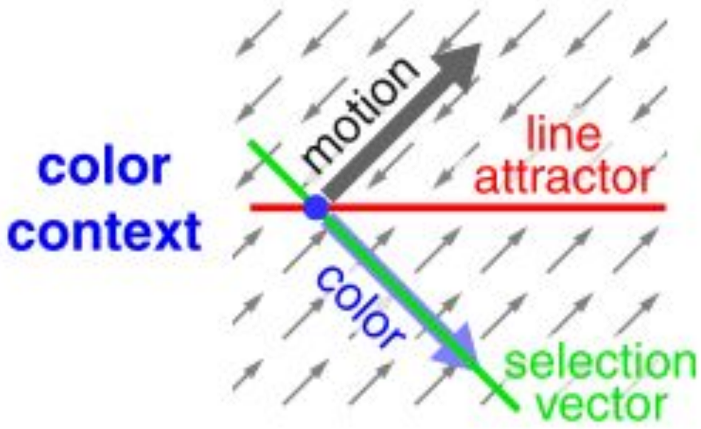



"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

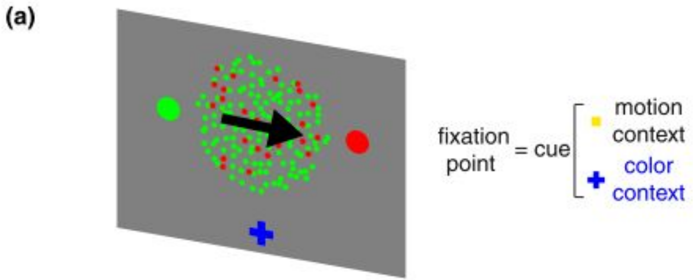
# RNN selects which features to accumulate

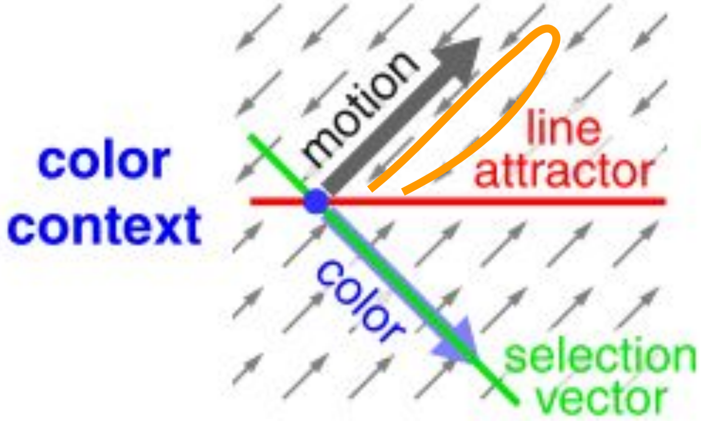



"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

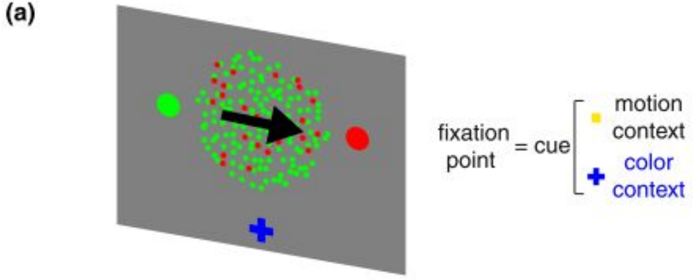
# RNN selects which features to accumulate

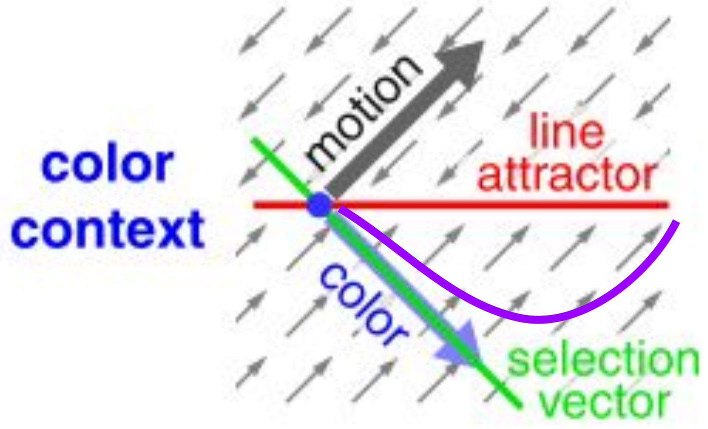



"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

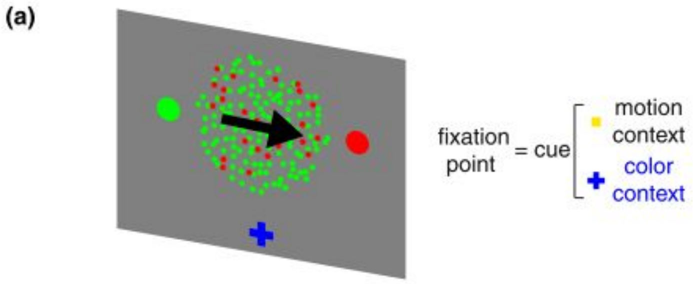
# RNN selects which features to accumulate

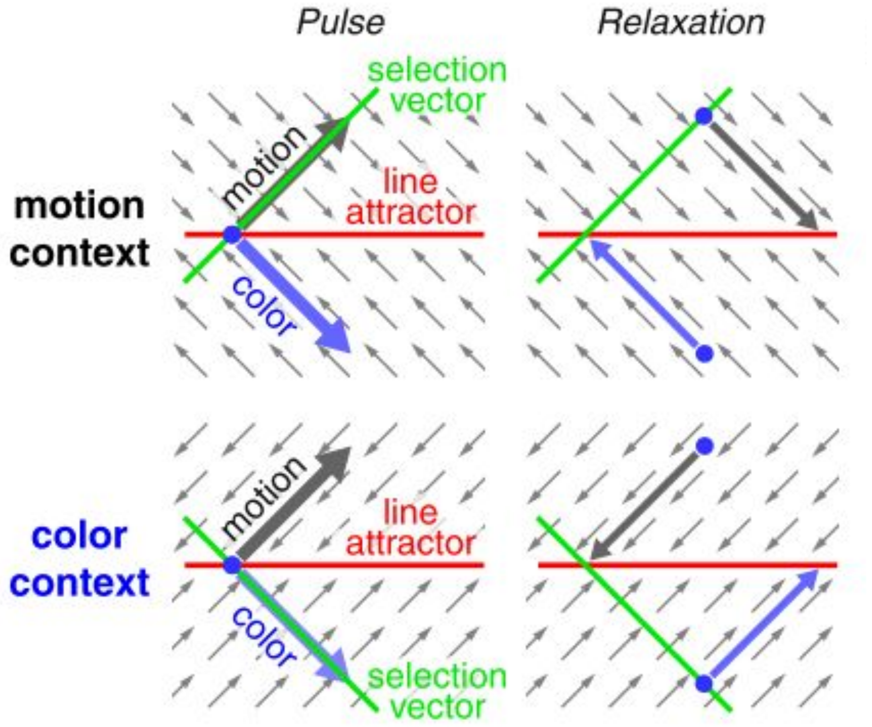



"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

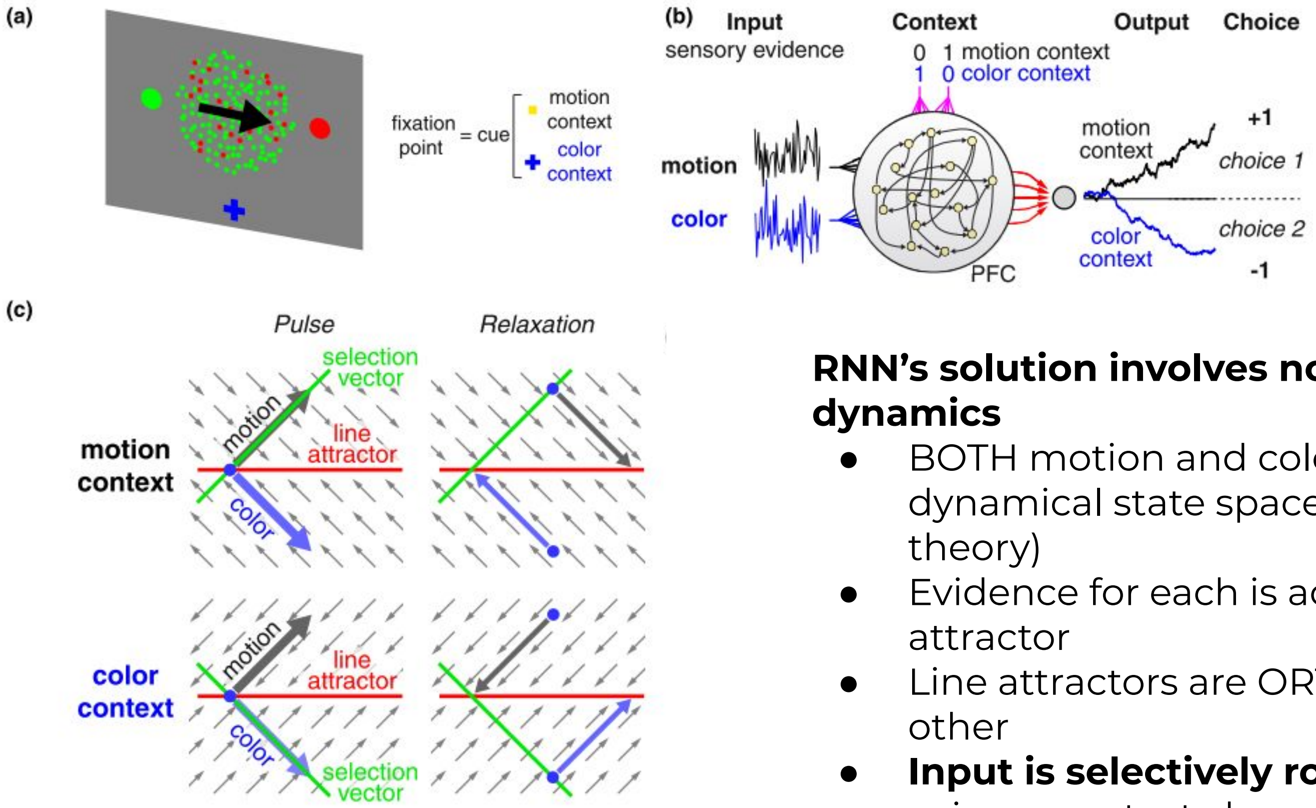
# RNN selects which features to accumulate




"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

# RNN selects which features to accumulate

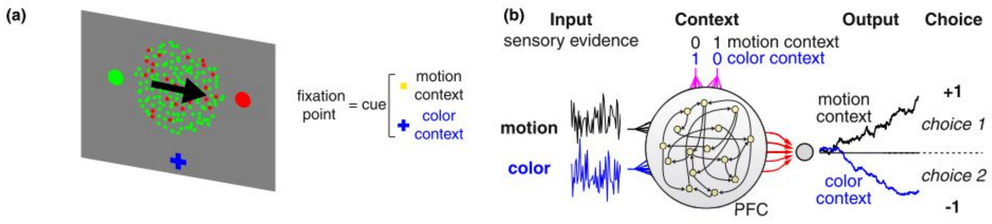



**RNN’s solution involves nonlinear “non-normal” dynamics**

- BOTH motion and color are represented in dynamical state space (contrary to gating theory)
- Evidence for each is accumulated with a line attractor
- Line attractors are ORTHOGONAL to each other
- **Input is selectively routed** to one or the other using a context-dependent selection vector

"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

# Revisiting context-dependent evidence accumulation




**RNN's solution involves nonlinear "non-normal" dynamics**

- BOTH motion and color are represented in dynamical state space (contrary to gating theory)
- Evidence for each is accumulated with a line attractor
- Line attractors are ORTHOGONAL to each other
- **Input is selectively routed** to one or the other using a context-dependent selection vector

**Non-normal network dynamics:**

"Structure and dynamical behavior of non-normal networks" (2018) Asllani et al.

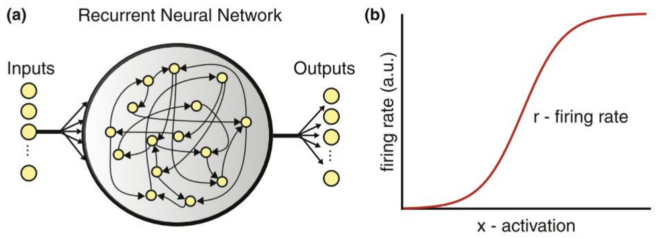
"Efficient communication over complex dynamical networks: The role of matrix non-normality" (2020) Baggio et al.

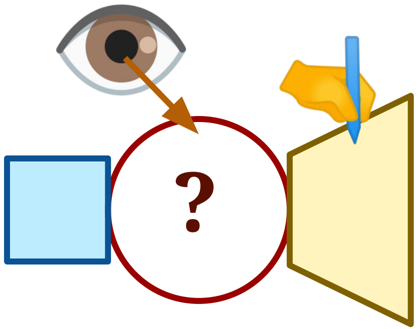
"Talk: Just-in-time computations by non-normal attractor dynamics in the prefrontal cortex" (2020) Stroud

"Balanced Amplification: A New Mechanism of Selective Amplification of Neural Activity Patterns" (2009) Murphy & Miller

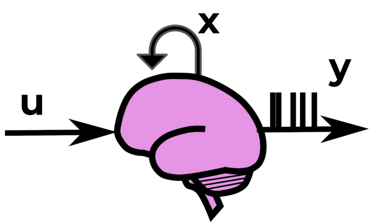
# Intermediate Summary

- RNNs combine dynamics with function-approximation capability of ANNs
- By training them to accomplish complex tasks we can **generate hypothesis** for how
- The structure of RNNs can be interpreted through the **lens of dynamical systems**



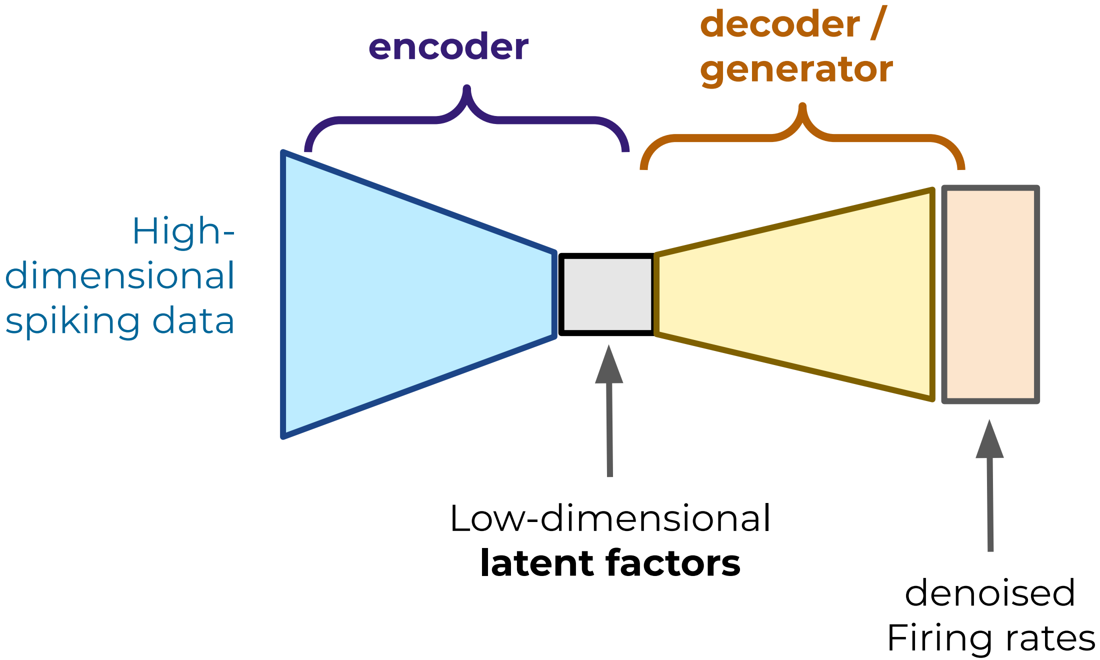




$$\tau\dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{J}\mathbf{r}(t) + \mathbf{B}\mathbf{u}(t) + b$$



# Latent Factor Analysis via Dynamical Systems (LFADS)

An autoencoder…



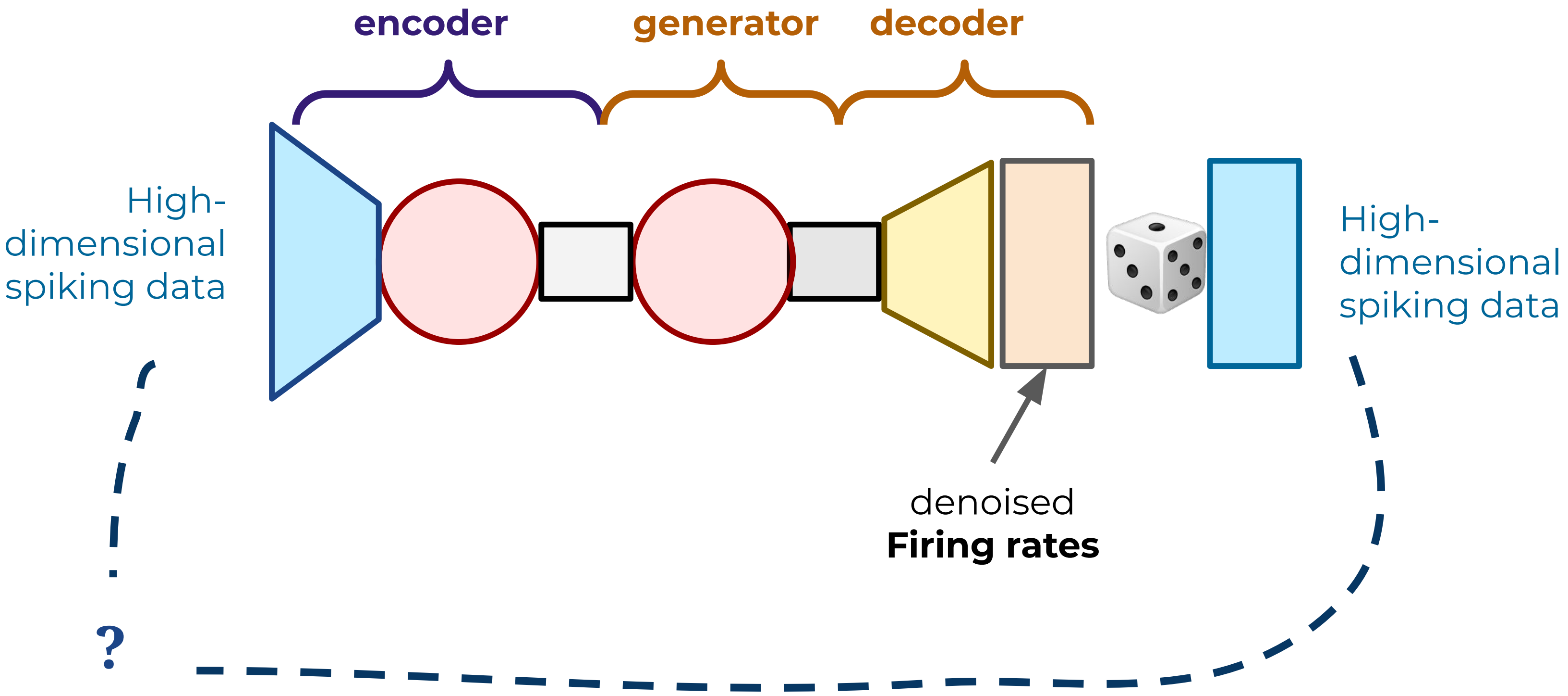

**Denoise (single trial) neural data**

By reconstructing it from low-dimensional factors

**Learn something about the* underlying mechanism along the way**

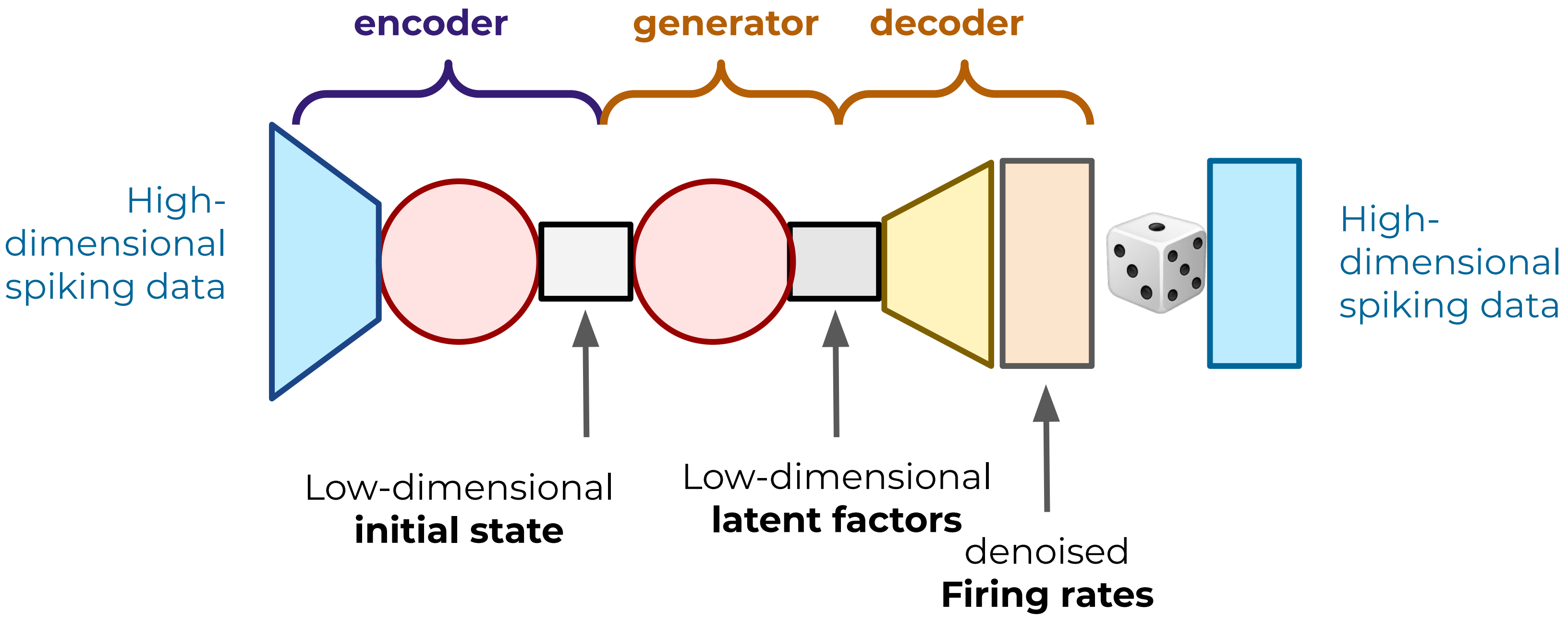
# Latent Factor Analysis via Dynamical Systems (LFADS)

An autoencoder + dynamics!



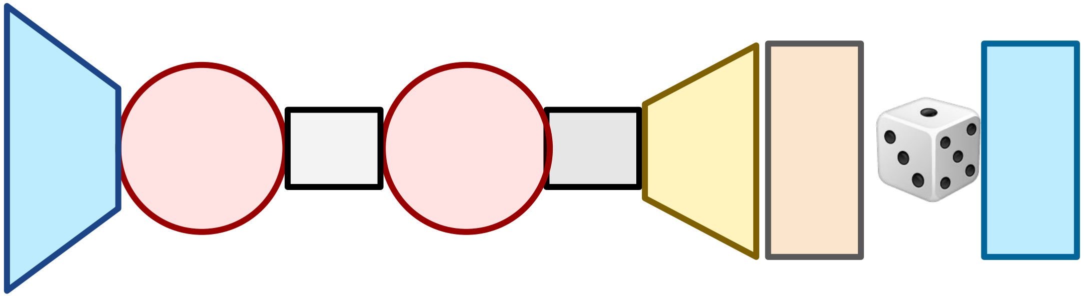
# Latent Factor Analysis via Dynamical Systems (LFADS)

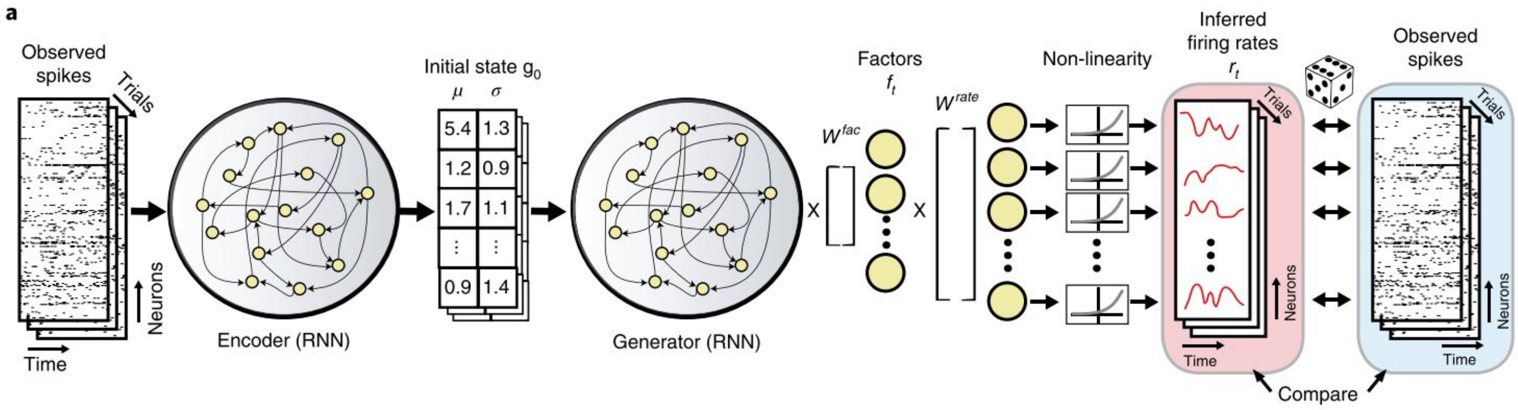
An autoencoder + dynamics!



# Latent Factor Analysis via Dynamical Systems (LFADS)

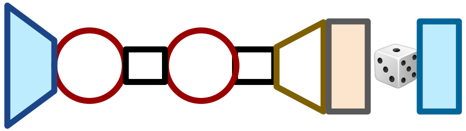
An autoencoder + dynamics!

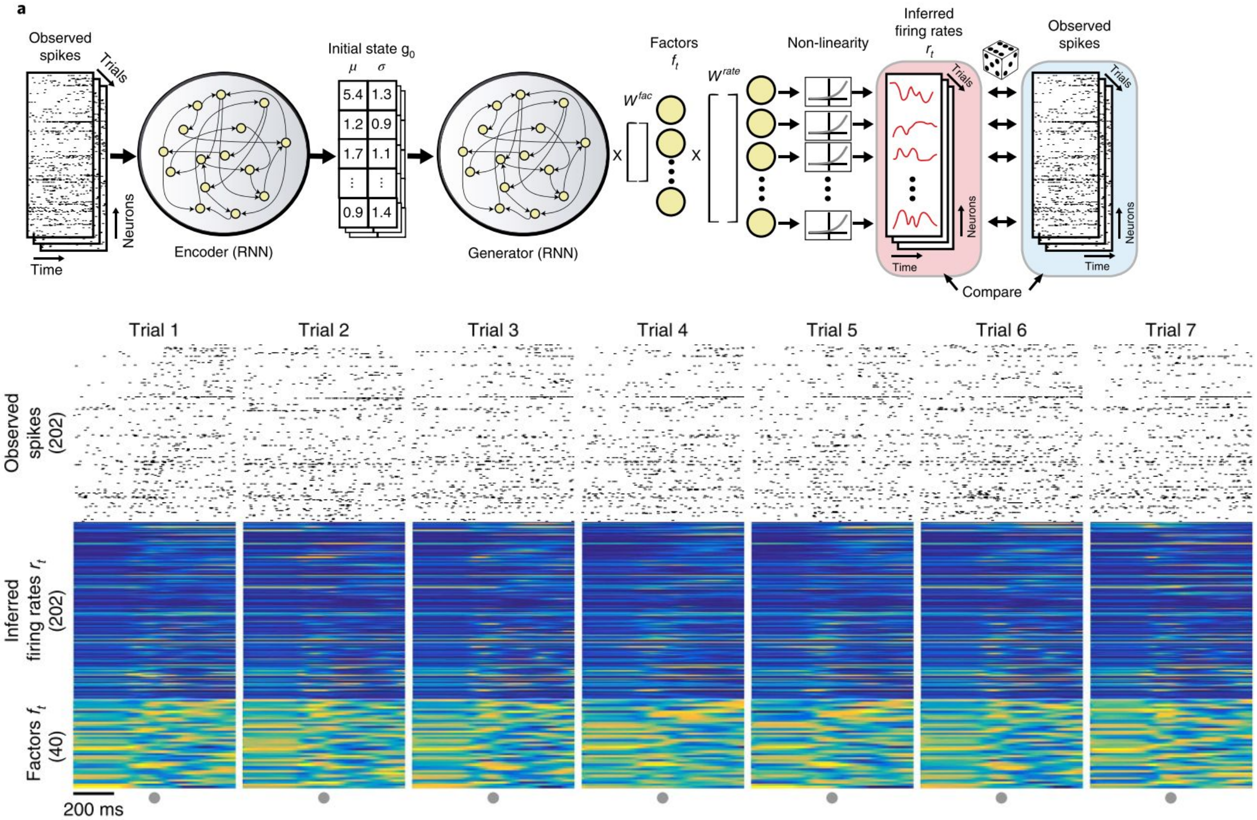





"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018) Pandarinath et al.

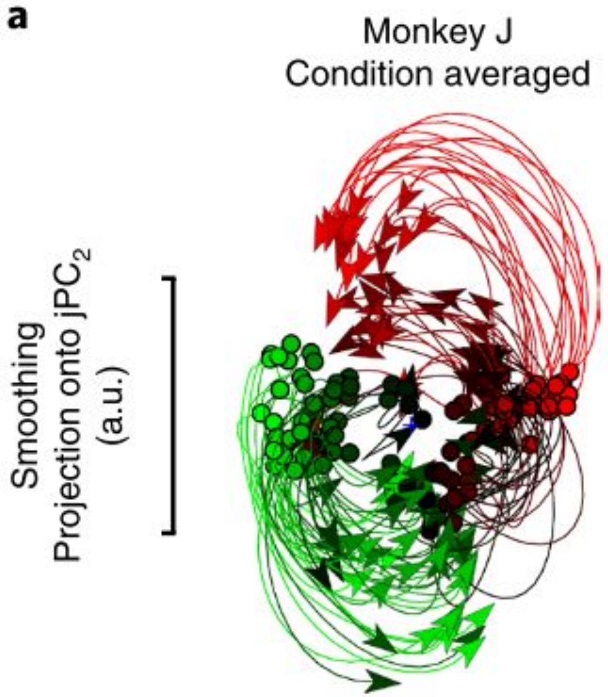
# LFADS Generates predictions of single-trial firing rates & spiking





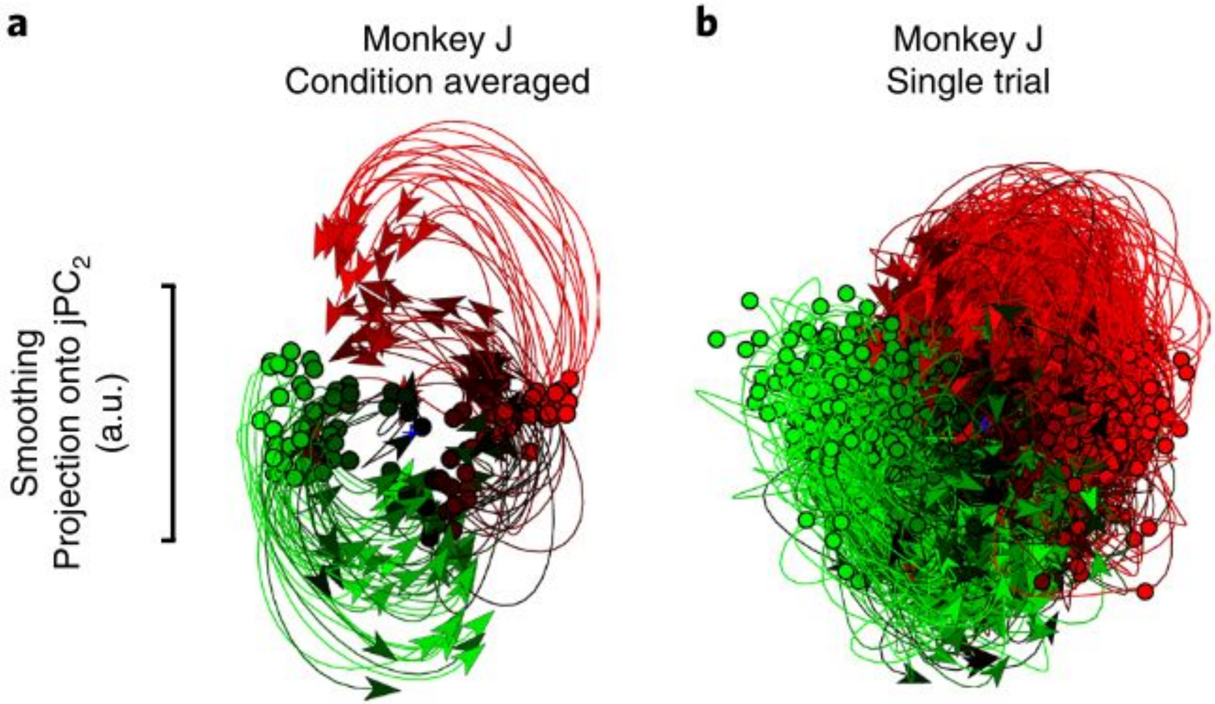

"[Inferring single-trial neural population dynamics using sequential auto-encoders]" (2018) Pandarinath et al.

# Spike-smoothing is insufficient to uncover rotation trajectories on single trials



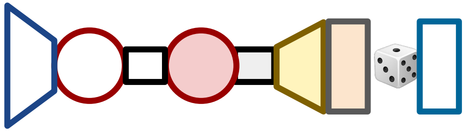

“Inferring single-trial neural population dynamics using sequential auto-encoders” (2018) Pandarinath et al.

# Spike-smoothing is insufficient to uncover rotation trajectories on single trials

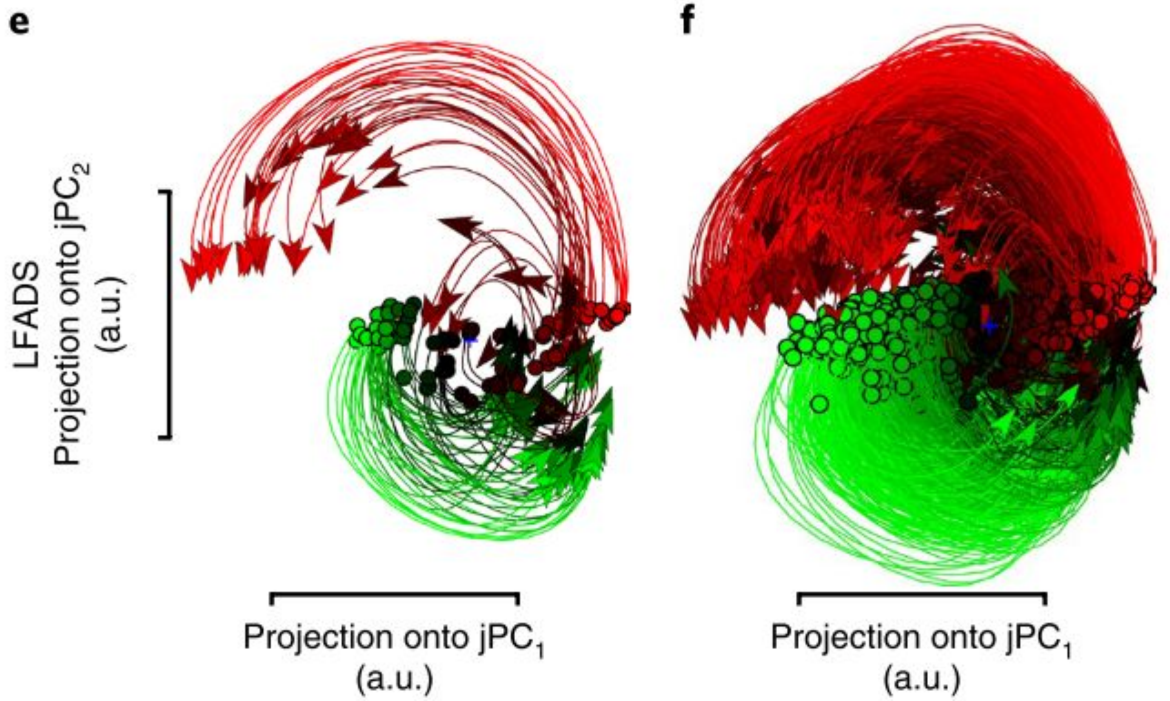



"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018) Pandarinath et al.

# But LFADS uses optimized latent structure to reveal **rotations on single-trials**



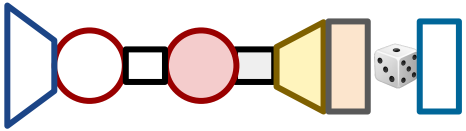
**condition-averaged** **single-trial**



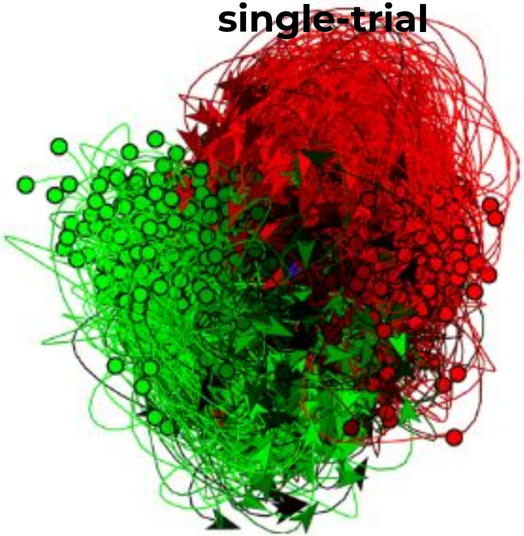

*This is a critical tool for understanding computation through dynamics!*

"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018) Pandarinath et al.

# But LFADS uses optimized latent structure to reveal **rotations on single-trials**

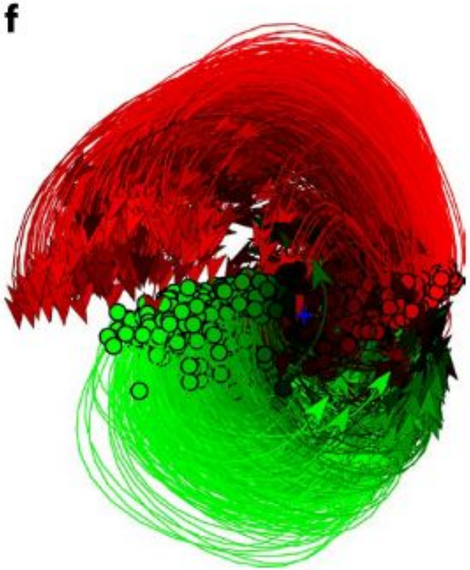


**Smoothing single-trial**



***This is a critical tool for understanding computation through dynamics!***

**LFADS single-trial**




"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018) Pandarinath et al.

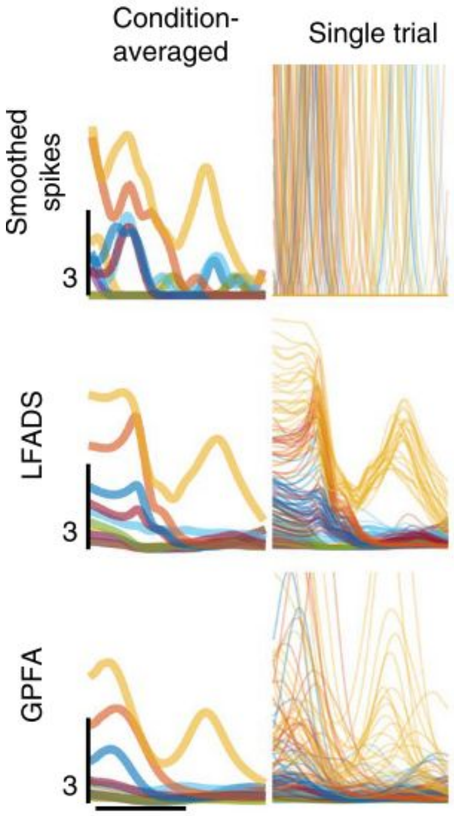
# But LFADS uses optimized latent structure to reveal **single-trial trajectories**

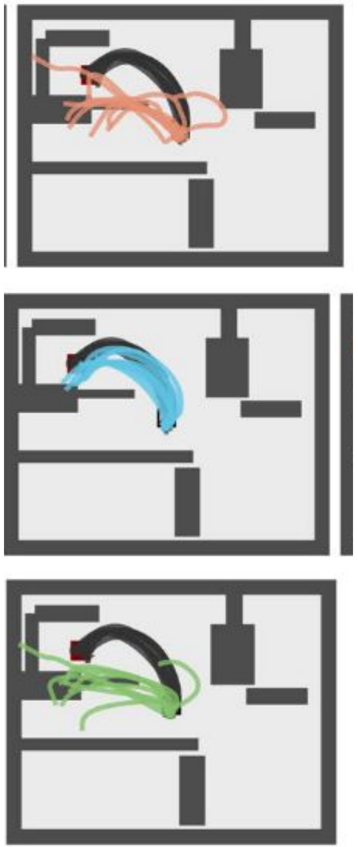
***This is a critical tool for understanding computation through dynamics!***

“Inferring single-trial neural population dynamics using sequential auto-encoders” (2018) Pandarinath et al.

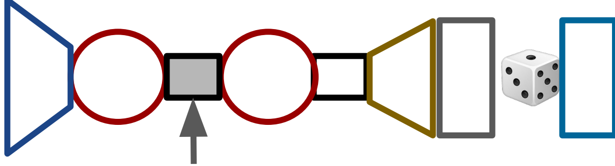
***Firing rates***

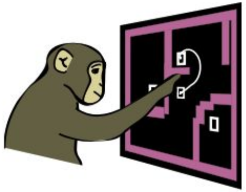
***Reach kinematics***

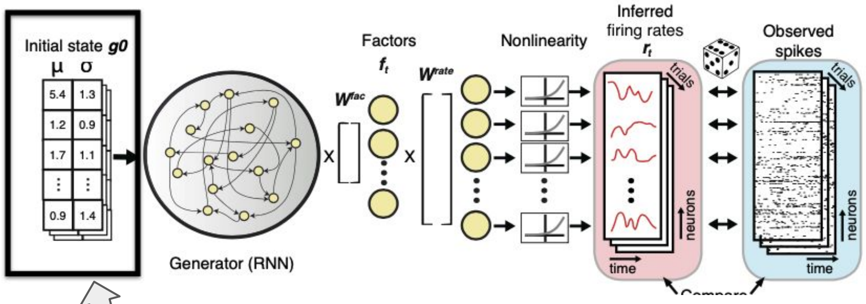


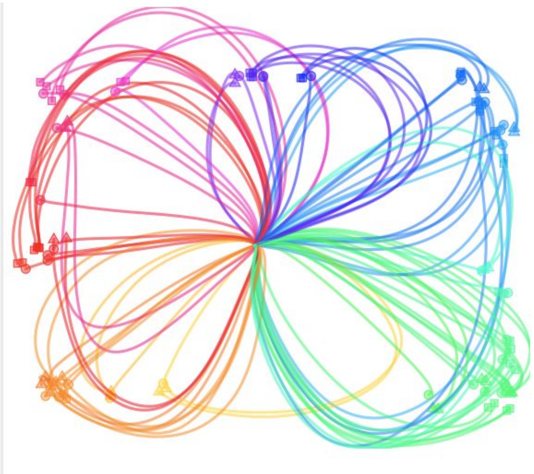



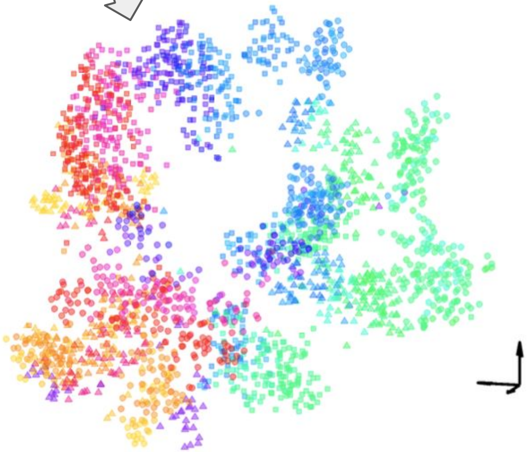
# Inferred initial state corresponds with task / behavioral variables







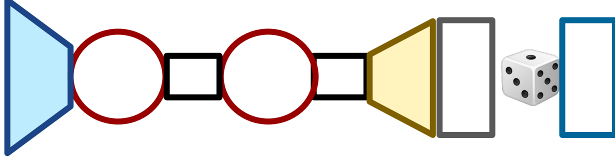


*tSNE of inferred initial conditions*

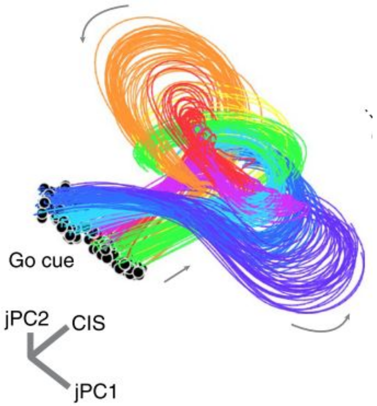
*Color = target location*

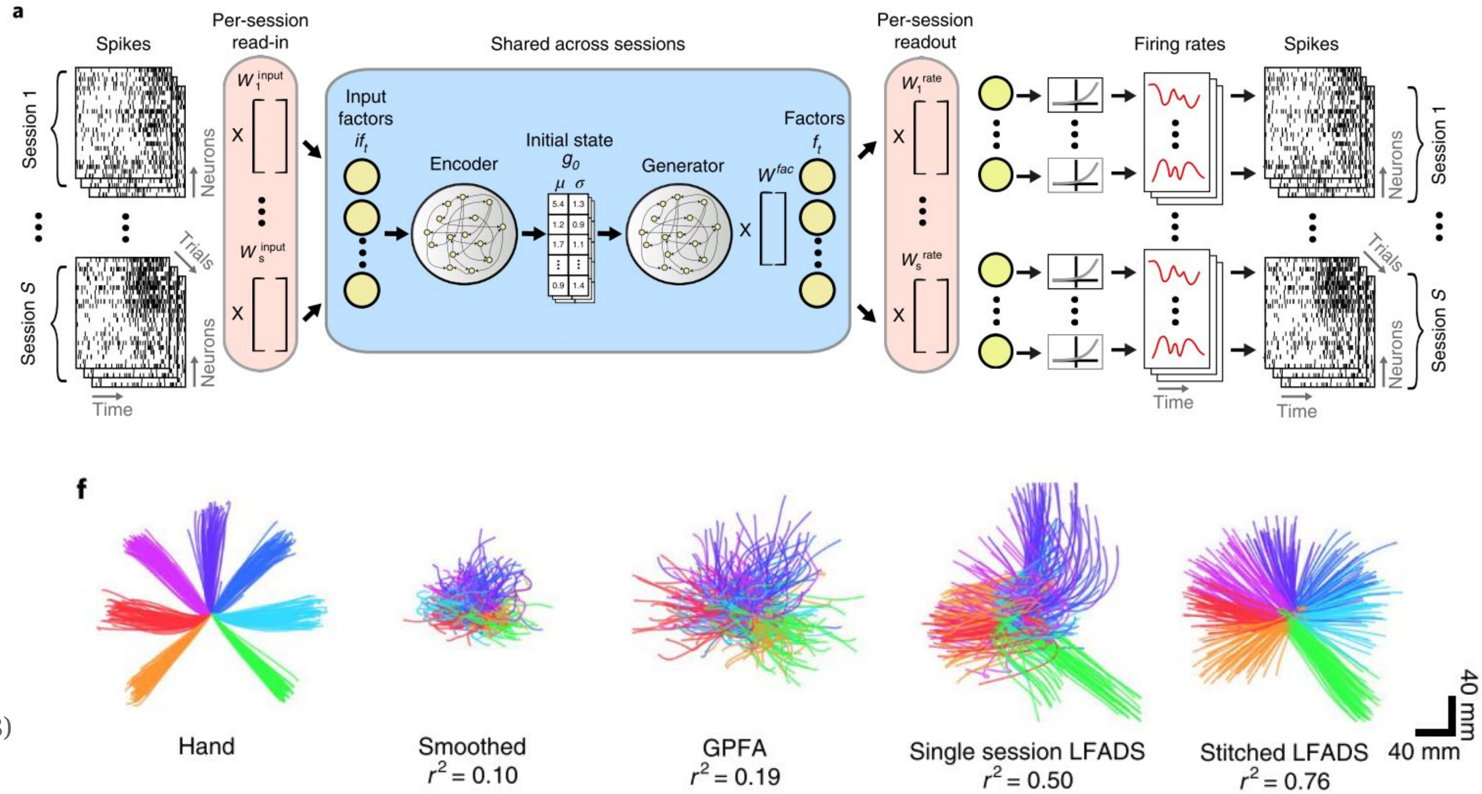
"[Inferring single-trial neural population dynamics using sequential auto-encoders]" (2018) Pandarinath et al.

# Bonus:



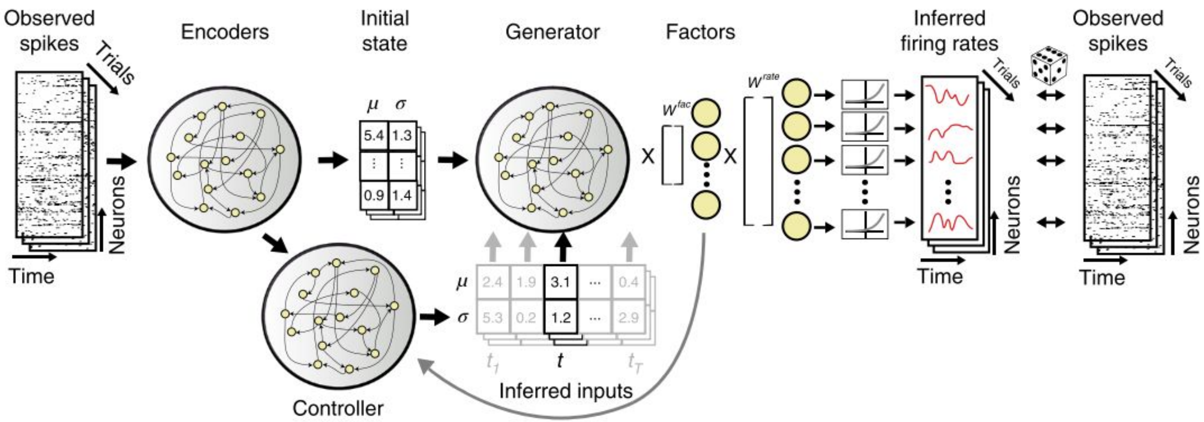
LFADS allows “stitching” together of neural data across sessions

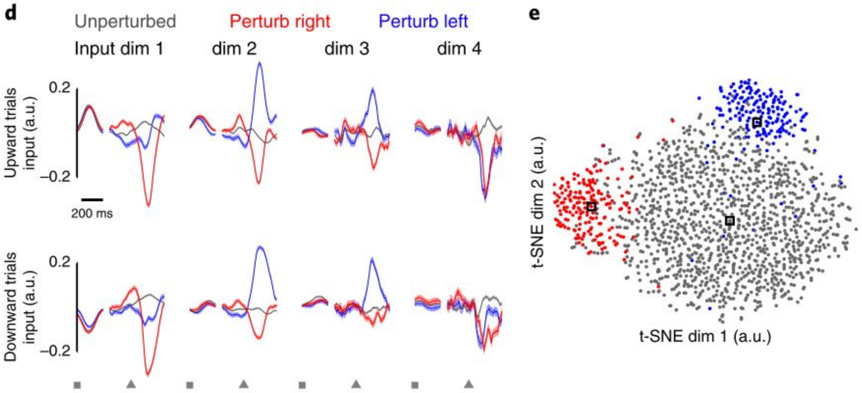


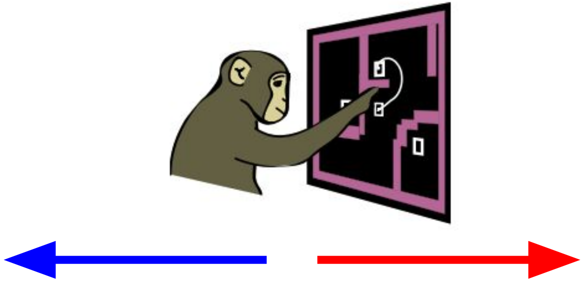




“[Inferring single-trial neural population dynamics using sequential auto-encoders](#)” (2018) Pandarinath et al.

# **Bonus:** LFADS infers latent perturbations



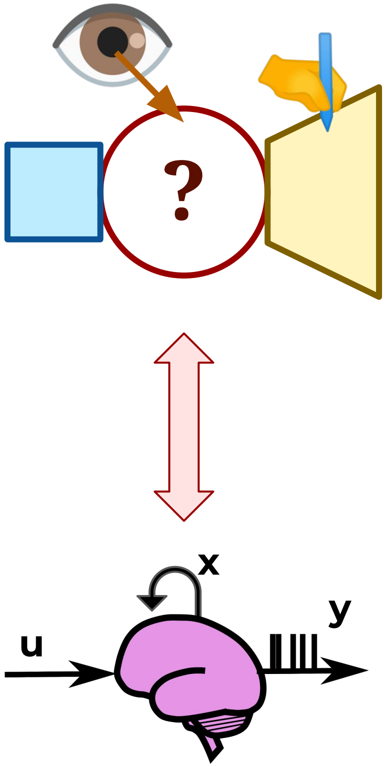



"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018) Pandarinath et al.

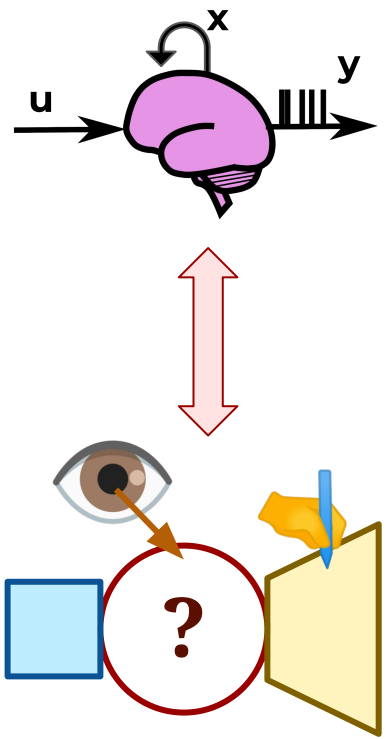
# Takeaway Points

- RNNs combine **dynamics with function-approximation** capability of ANNs
- By training them to accomplish complex tasks we can **generate hypothesis** for how
- The structure of RNNs can be interpreted through the **lens of dynamical systems**
- **If outputs look similar,** some evidence for the RNN’s hypothesis
    - Regardless, gives you a space to **perturb and evaluate alternate hypotheses**
- **Estimation of single-trial dynamics unlocks unprecedented access to test hypotheses**



# Recurrent Neural Networks:

A model organism
for studying
computation
over time
in neuroscience

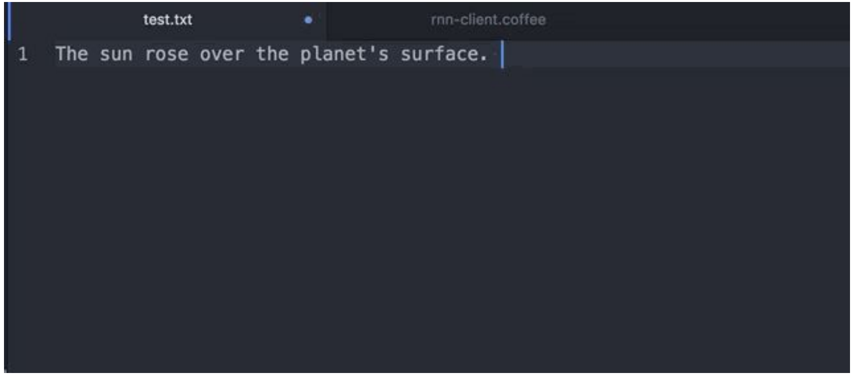


# Additional Resources

# Diverse application of RNNs for ML

Text generation (recipes, dungeons & dragons spells, “the onion” headlines)

Story “autocompletion” [demo]

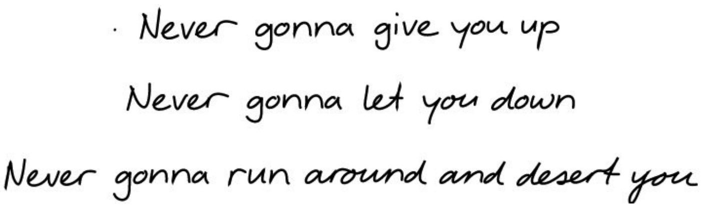


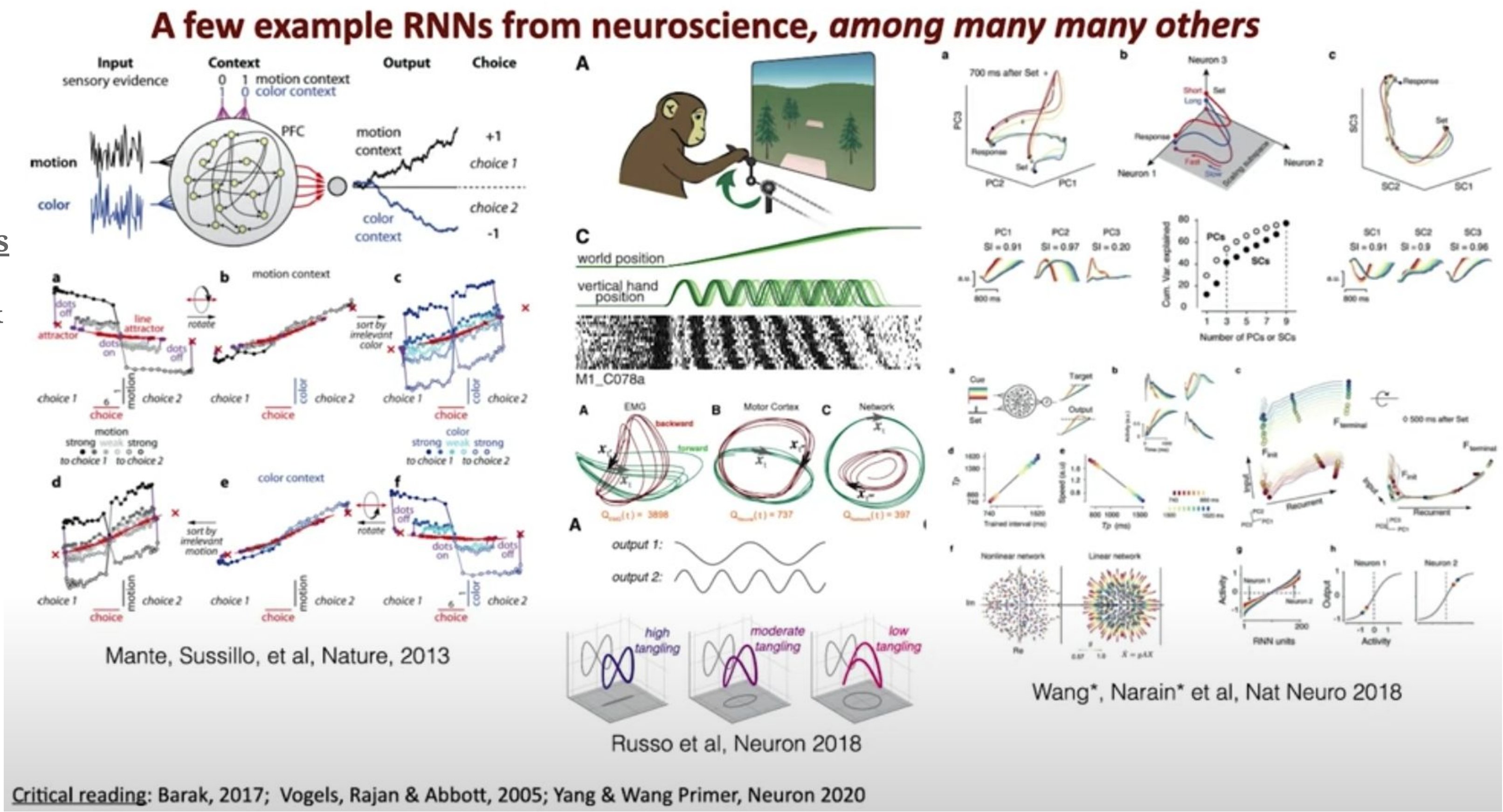

Feeling writer's block? Hit **<Tab>** and an artificial neural network running in your **browser** will finish your sentence as if it were written by **Shakespeare**, the **US Supreme Court**, or **Tupac Shakur**.

Expressive piano performance [video]

Doodle alongside Sketch-RNN

handwriting synthesis






"Artificial Neural Networks for Neuroscientists: A Primer" (2020) Yang & Wang

"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

"Motor cortex embeds muscle-like commands in an untangled population response" (2018) Russo et al.

"Flexible timing by temporal scaling of cortical responses" (2017) Wang et al.

# Applications of RNNs to understand the brain

"Task representations in neural networks trained to perform many cognitive tasks" (2019) Yang et al.

"Flexible Sensorimotor Computations through Rapid Configuration of Cortical Dynamics" (2018) Remington et al.

"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussilo et al.

> *"The authors provide a detailed example of how the mechanisms of a dynamical computation can be discovered in an RNN, providing a powerful hypothesis for the underlying mechanism in a complex cortical circuit."*

"A neural network that finds a naturalistic solution for the production of muscle activity" (2015) Sussillo et al.

"Hierarchical recurrent state space models reveal discrete and continuous dynamics of neural activity in C. elegans" (2019) Linderman et al.

"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018) Pandarinath et al.

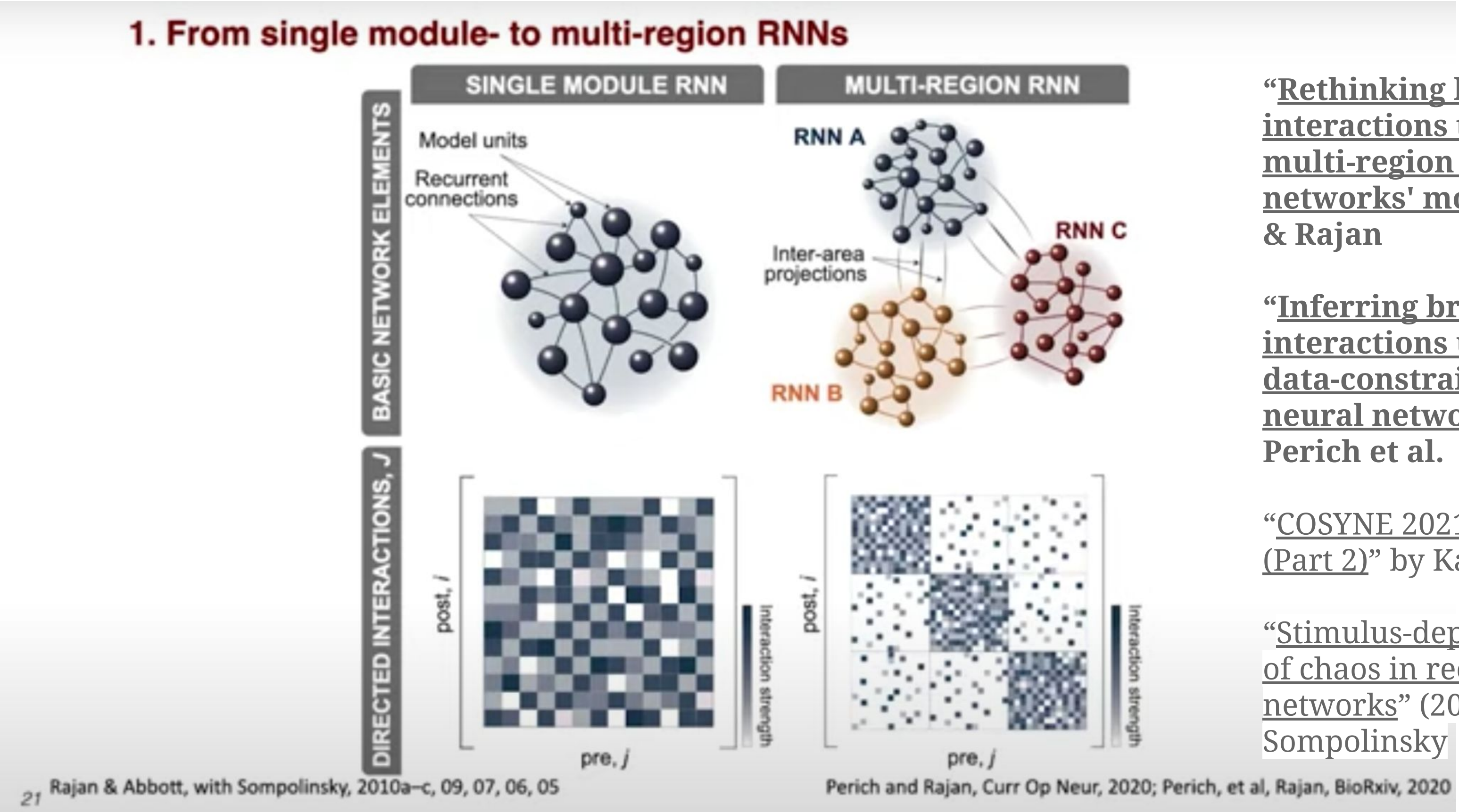
"Emergence of grid-like representations by training recurrent neural networks to perform spatial localization" (2018) Cueva & Wei

"Task-Driven Convolutional Recurrent Models of the Visual System" (2018) Nayebi

"Predictive coding arises in RNNs due to energy constraints" (2021) Ali et al.

"Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity" (2021) Saxena et al.

"Motor cortex embeds muscle-like commands in an untangled population response" (2018) Russo et al.



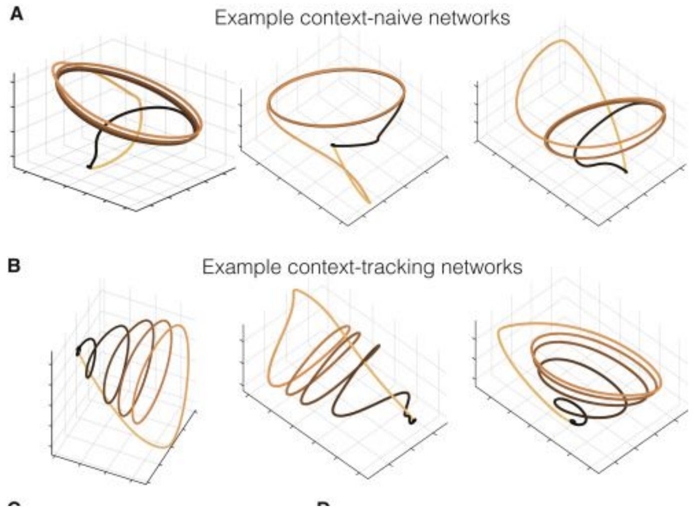

**“Rethinking brain-wide interactions through multi-region 'network of networks' models” (2020) Perich & Rajan**

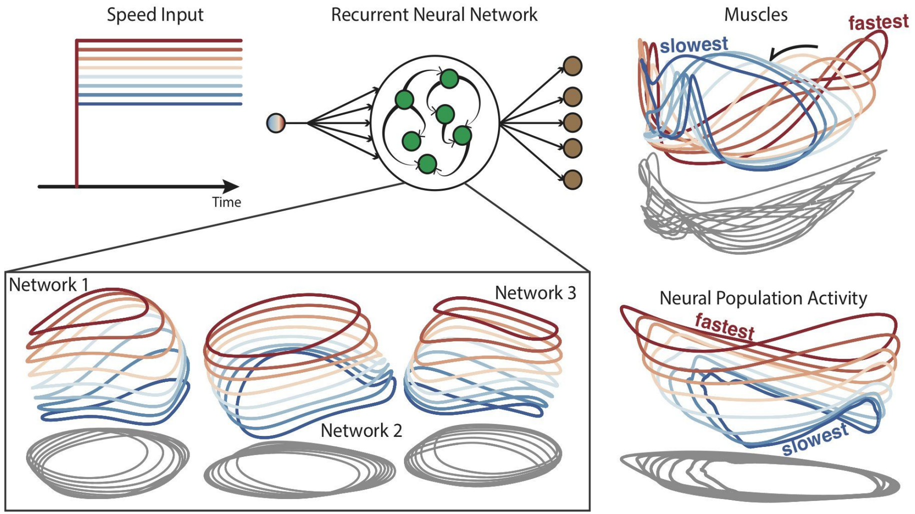
**“Inferring brain-wide interactions using data-constrained recurrent neural network models” (2020) Perich et al.**

“COSYNE 2021 - Tutorial on RNN (Part 2)” by Kanaka Rajan

“Stimulus-dependent suppression of chaos in recurrent neural networks” (2010) Rajan, Abbott, Sompolinsky

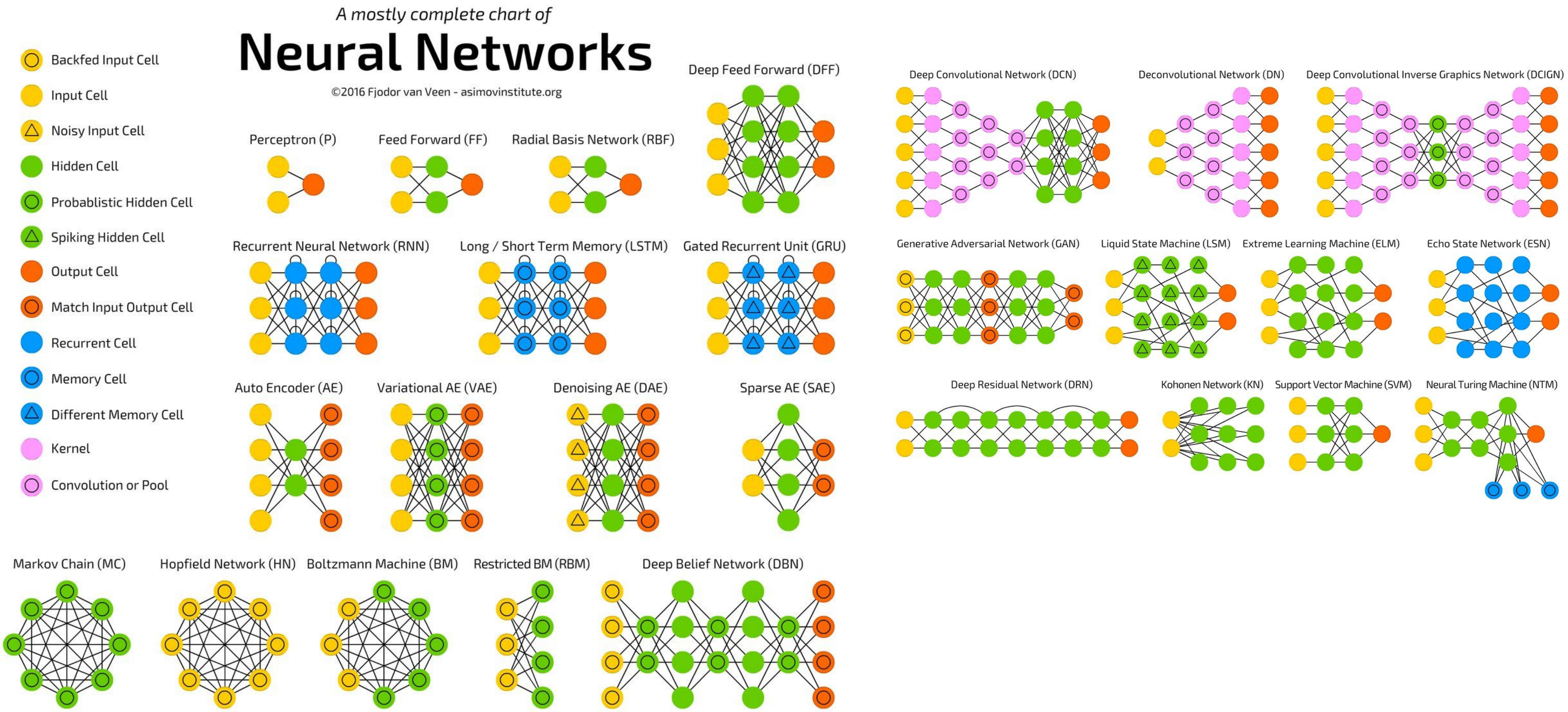
# RNNs for generating & testing hypotheses




"Neural trajectories in the supplementary motor area and primary motor cortex exhibit distinct geometries, compatible with different classes of computation" (2019) Russo et al.

"Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity" (2021) Saxena et al.




“The Neural Network Zoo” Fjodor Van Veen

# High Level Reviews of RNNs in Neuro

"Deep Reinforcement Learning and Its Neuroscientific Implications" (2020) Botvinick et al.

"A deep learning framework for neuroscience" (2019) Richards et al.

"Opportunities and obstacles for deep learning in biology and medicine" (2017) Ching et al.

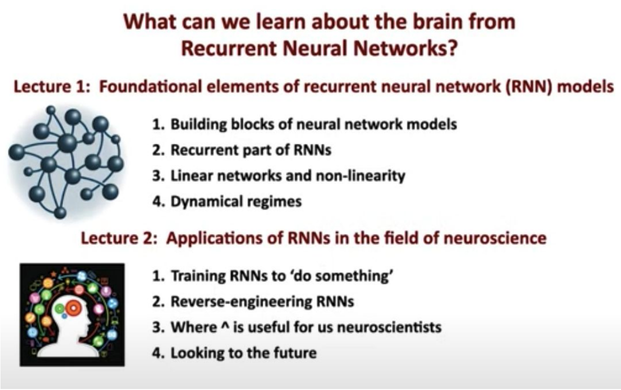
"Recurrent neural networks as versatile tools of neuroscience research" (2017) Barak

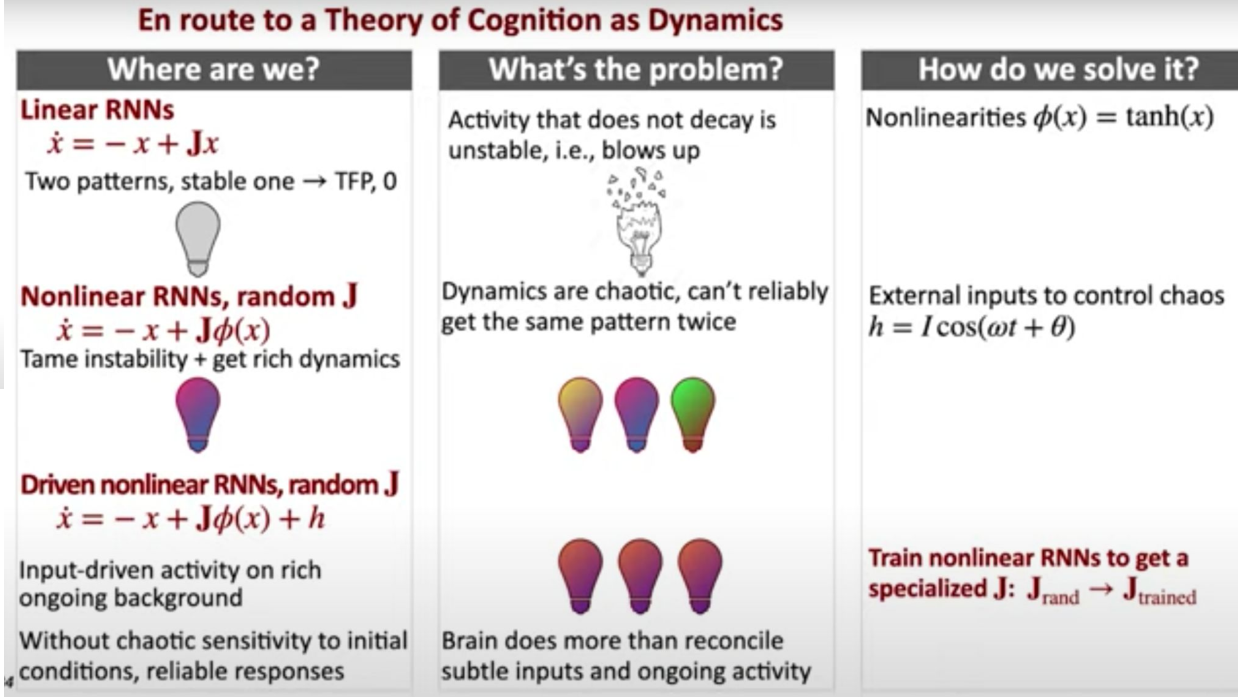
"Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing" (2015) Kriegeskorte

"Artificial Neural Networks for Neuroscientists: A Primer" (2020) Yang & Wang

# Interpretation of RNNs in Neuro

“COSYNE 2021 - Tutorial on RNN (Part 1)” by Kanaka Rajan



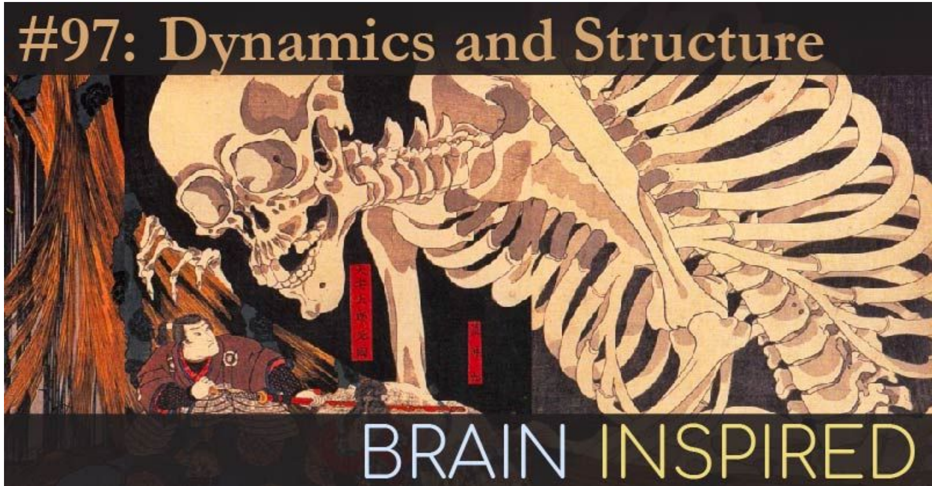
# Interpretation of RNNs in Neuro

"Learning interpretable continuous-time models of latent stochastic dynamical systems" (2019) Duncker et al.

"Interrogating theoretical models of neural computation with deep inference" (2019) Bittner

"Opening the Black Box - using Dynamics to understand RNNs (to understand the brain)" (2013) Sussillo, Barak

“Dynamics & Structure” Sussillo & Barak
@ Brain Inspired Podcast



# Tutorials / How-To for an RNN

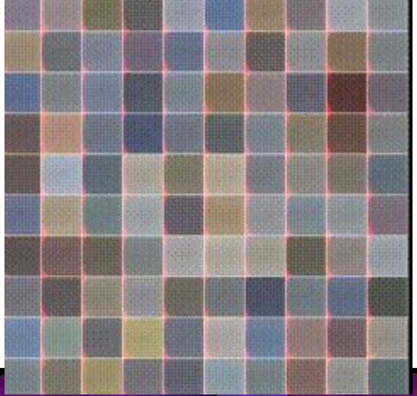
**“ [The Unreasonable Effectiveness of Recurrent Neural Networks]” - Andrej Karpathy**
- Easy to read
- Lots of fun examples
- Has code for RNNs for text generation

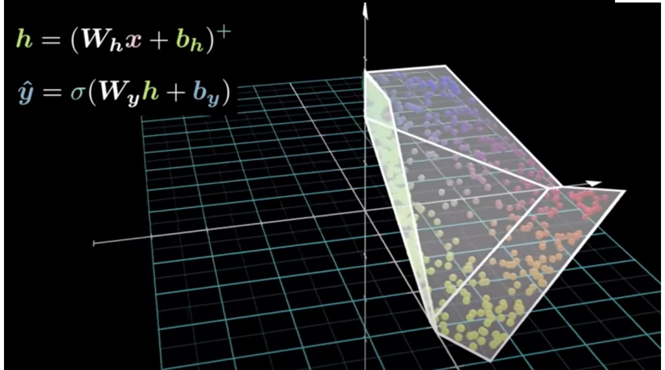
“Practicum: RNN and LSTM architectures” - Alfredo Canziani & Yann LeCunn
- Focus on visual explanation
- Has associated code notebooks

“The Neural Network, A Visual Introduction” vcubingx

“Deep Learning” 3Blue1Brown

“What is backpropagation really doing?” 3Blue1Brown

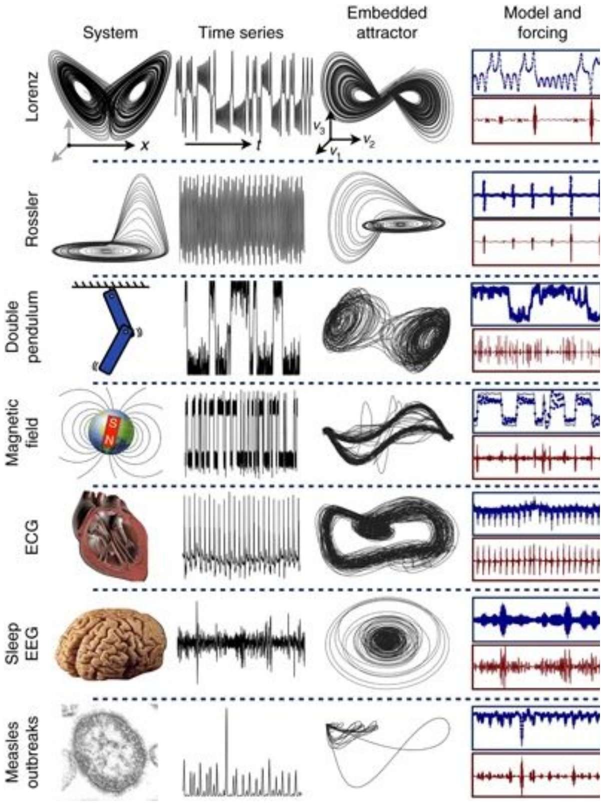


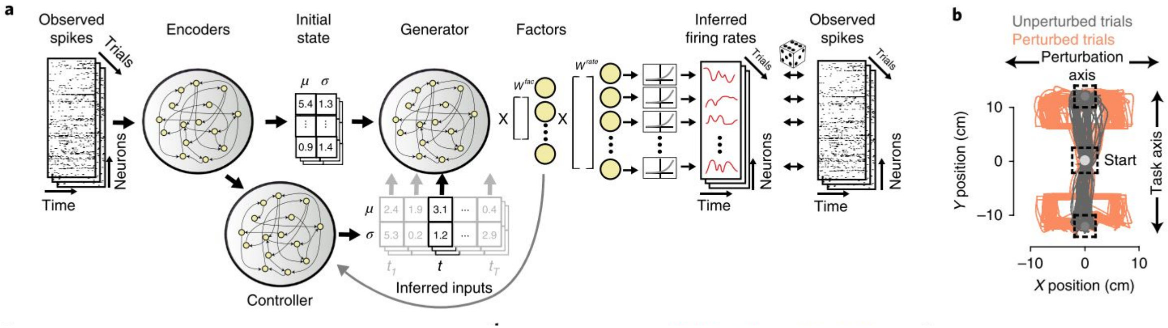



# Input-driven dynamics in RNNs

"Chaos as an intermittently forced linear system" (2017) Brunton et al.

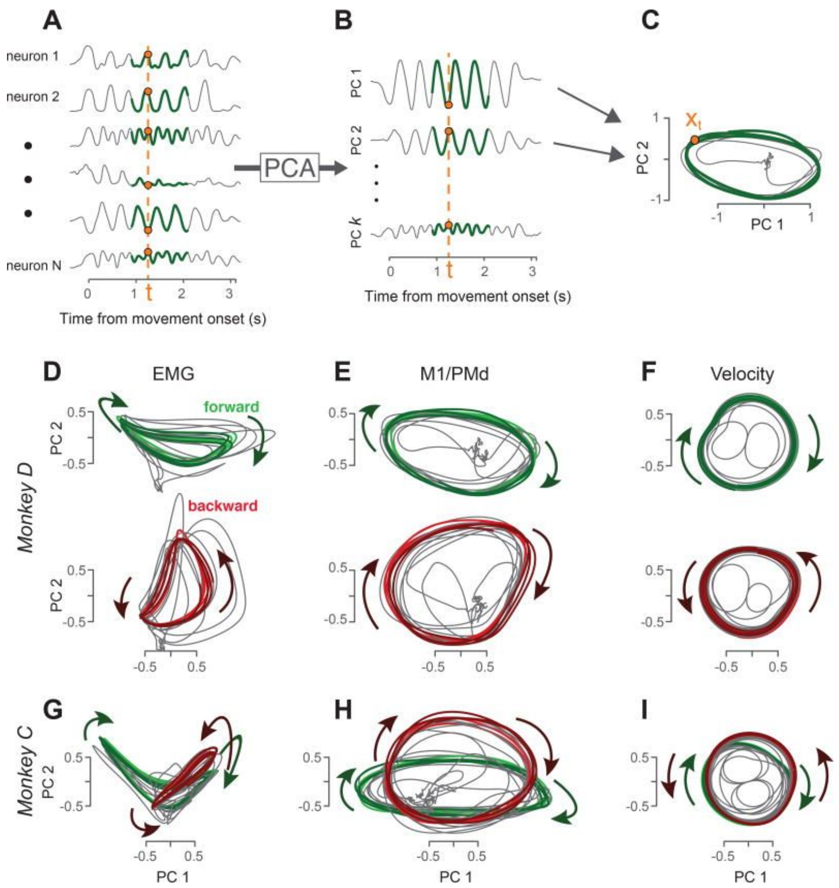
"Echo State Property Linked to an Input: Exploring a Fundamental Characteristic of Recurrent Neural Networks" (2013) Manjunath & Jaeger




**LFADS uncovers the presence, identity, and timing of unexpected perturbations in the cursor jump task**

# Tangling Metric




$$Q(t) = \max_{t'} \frac{\|\dot{x}_t - \dot{x}_{t'}\|^2}{\|x_t - x_{t'}\|^2 + \varepsilon}$$

"where $\boldsymbol{x}_t$ is the neural state at time *t: i.e.*, a vector containing the neural responses at that time, $\dot{x}_t$ is the temporal derivative of the neural state, $\|\cdot\|$ is the Euclidean norm, and ε is a small constant that prevents division by zero (*methods*). $Q(t)$ becomes high if there exists a state at a different time, $t'$, that is similar but associated with a dissimilar derivative. For example, tangling is high if two trajectories pass through similar points but in different directions"

"Motor cortex embeds muscle-like commands in an untangled population response" (2018) Russo et al.