Emerging computations in trained neural networks and real brains

Néstor Parga

Universidad Autónoma de Madrid

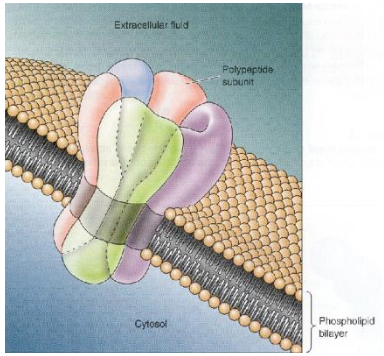
Synaptic plasticity allows cortical circuits to learn new tasks and to adapt to changing environments. How do cortical circuits use plasticity to acquire functions such as sensory coding, decision-making or working memory? Neurons are connected in complex ways, forming recurrent neural networks, and learning modifies the efficiency of the connections. Furthermore, neurons communicate emitting brief discrete electric signals or spikes. I my talk I will describe how to train recurrent neural networks of spiking units in task like those used to train animals in neuroscience laboratories and how computations emerge in the trained networks.

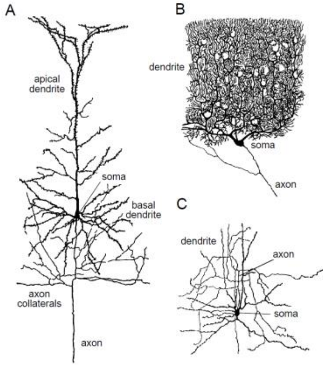
Surprisingly, artificial networks and real brains can use similar computational strategies.

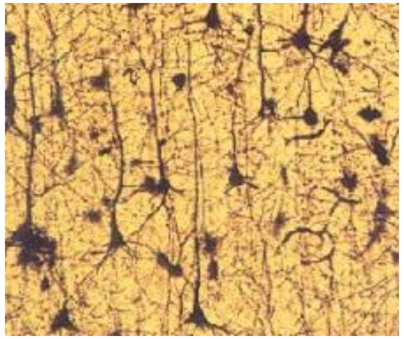
Work funded by:
PGC2018-101992-B-I00

Computational and Systems Neuroscience

The purpose of Systems Neuroscience is to explain behavior starting from neural networks and its constituents.

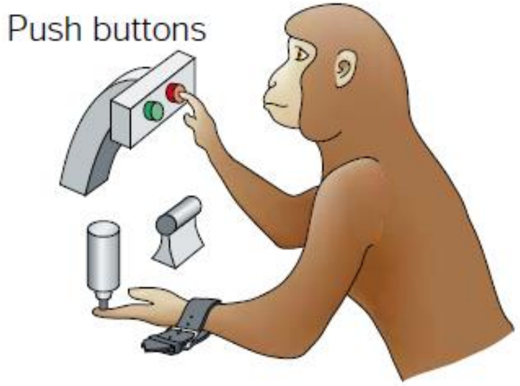




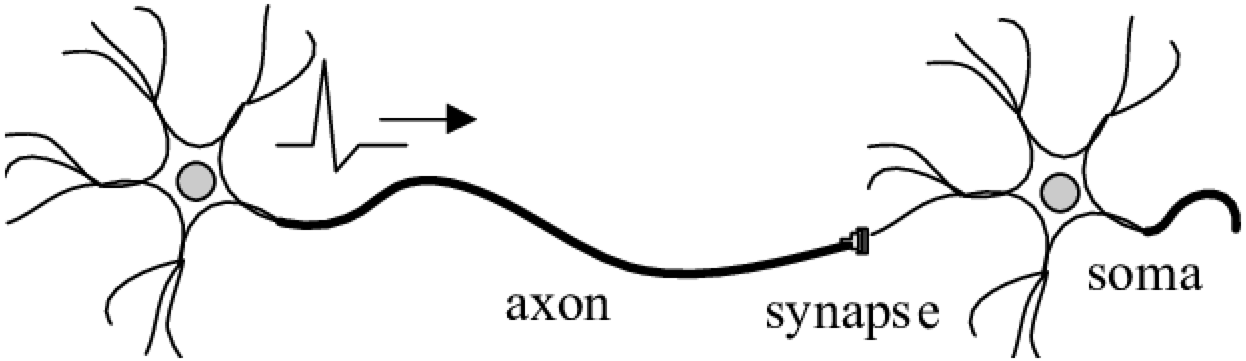


A monkey has to make a decision about the tactile stimuli applied to one of his fingertips.

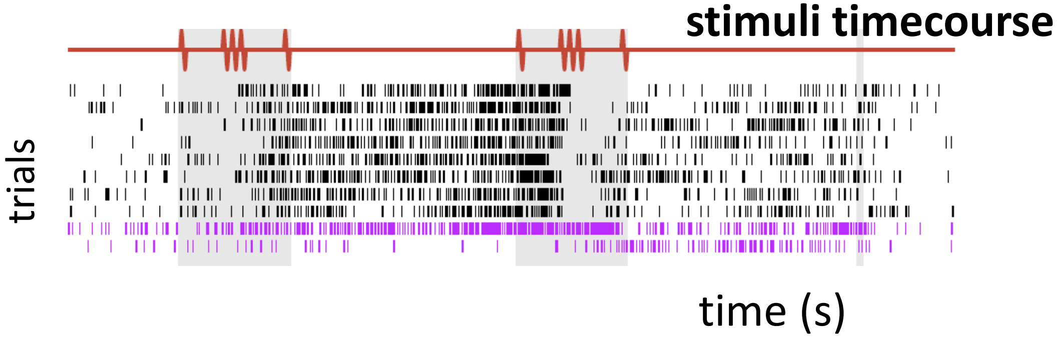
How are the stimuli processed by molecules, neurons and neural circuits in order to execute an action?



Action potential (spike)

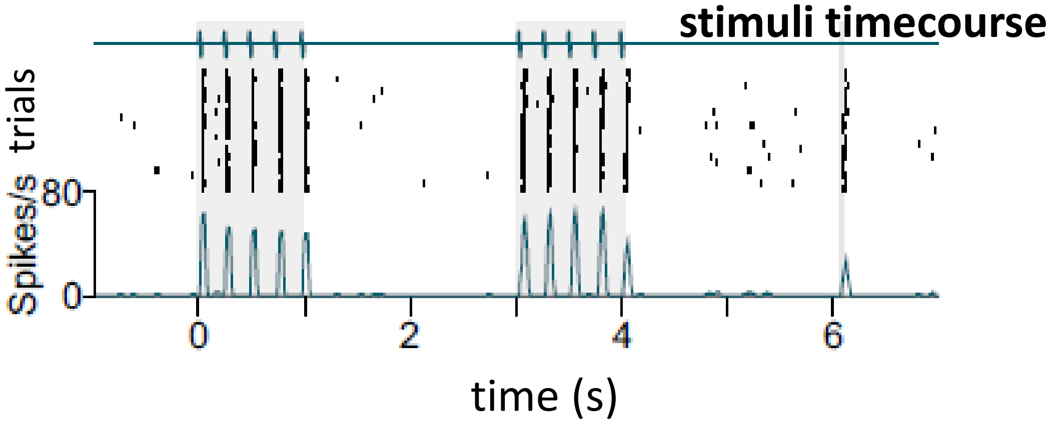


Rastergram
(frontal area)



Rastergram
(primary sensory area)

Firing Rate
(# of spikes/ time)



OUTLINE

A) Electrophysiological Experiments

B) Training Neural Networks: Artificial versus biologically-plausible features

C) Learning Algorithms: Reservoir learning, FORCE and full-FORCE

D) Detection Task: a RNN of Rate Neurons – State-space Analysis

E) Bayesian Computations in spiking RNNs

F) Beyond: Reinforcement Learning

Electrophysiological Experiments

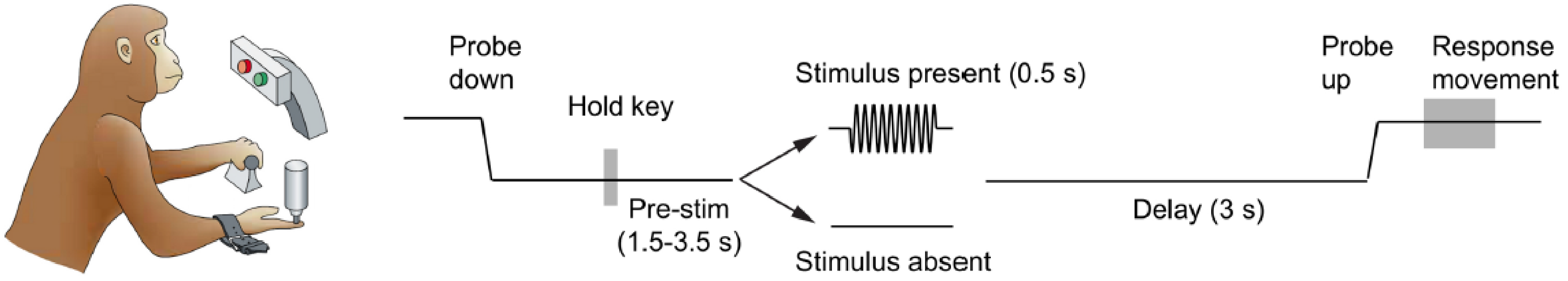
I will consider four experiments:

- A stimulus detection task
- A tactile frequency discrimination task
- A temporal interval discrimination task
- A time interval production task

Task: A mapping from stimuli to actions.

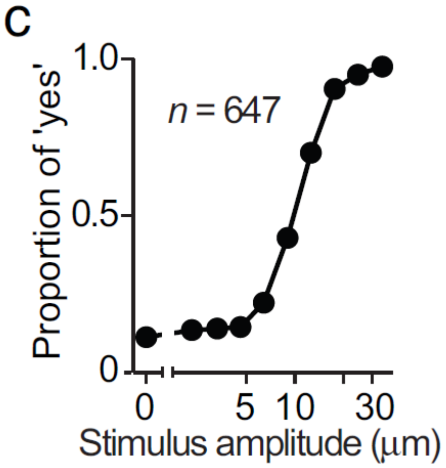
Correct animal's responses are rewarded.

Detection Task



- *The monkey has to detect a vibrating stimulus, which is present only in half of the trials.*
- *When it is applied, its amplitude is often rather weak (it takes 9 different values).*
- *The stimulation time is not fixed. There is a possible stimulation window of 2 seconds.*
- *The monkey reports his decision after a 3-second delay period*

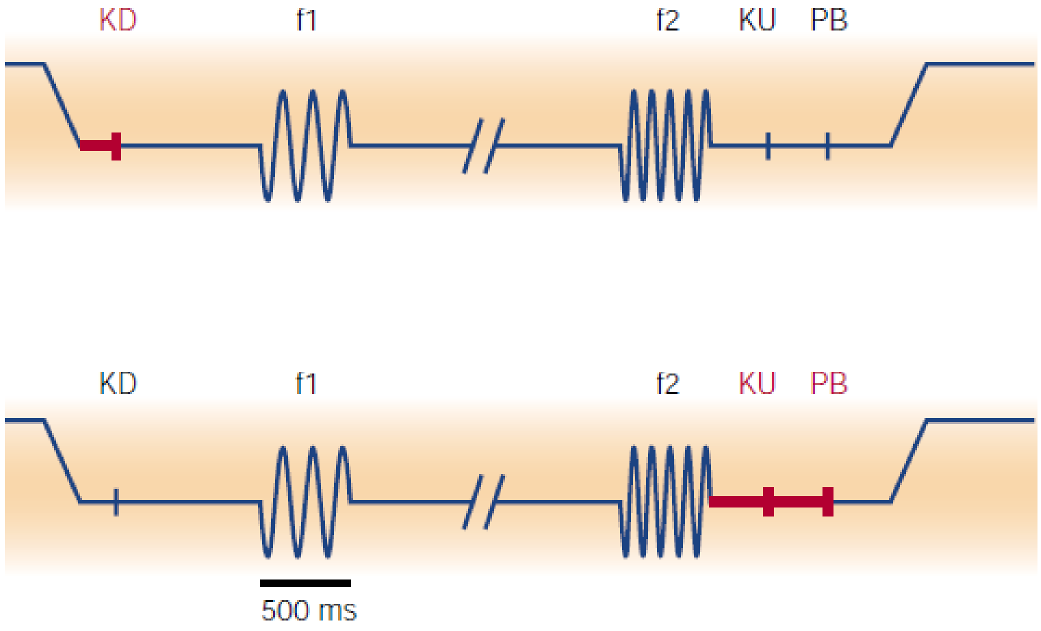
Response		Stimulus: Present	Stimulus: Absent
Response	Yes	Hit	FA
Response	No	Miss	CR

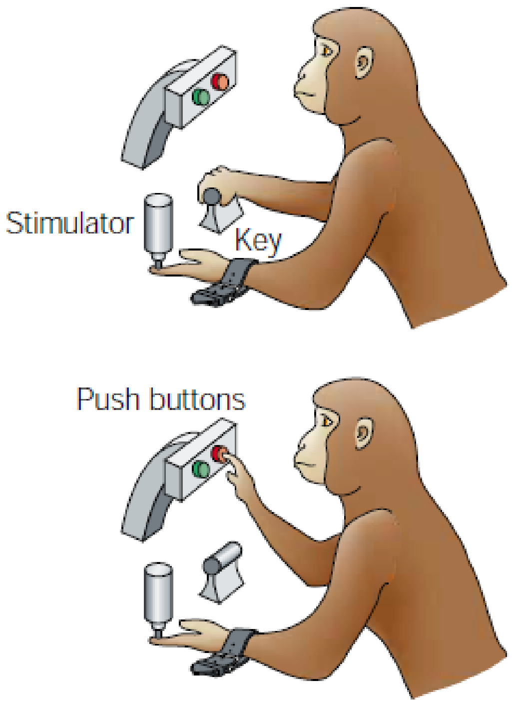


Performance

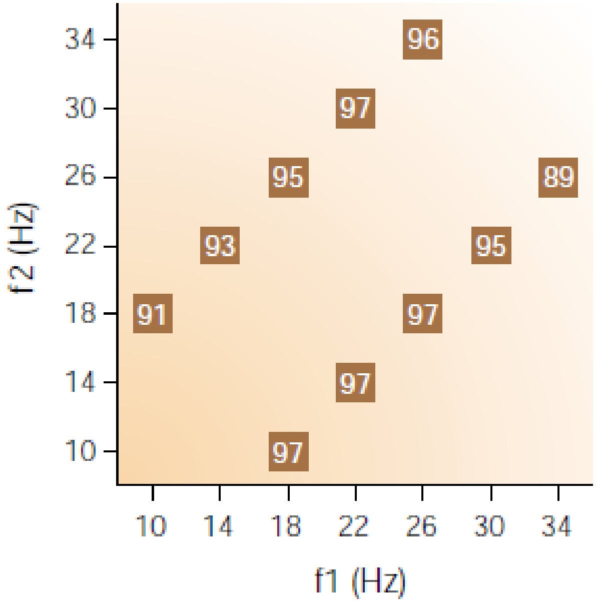
*de Lafuente & Romo, Nature Neuroscience **8**: 1698; 2005*

Tactile Frequency Discrimination Task





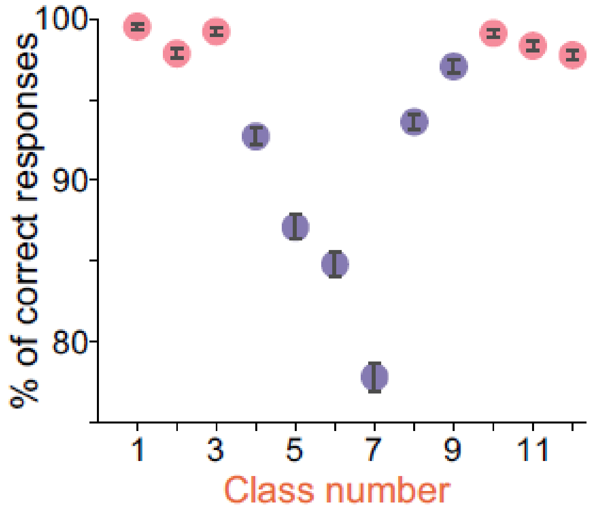
Stimulus Set and Performance



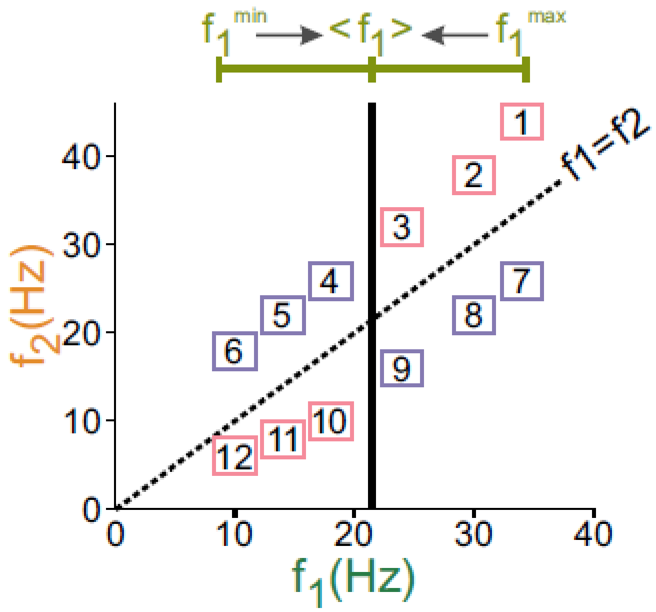
Perceptual Bias (Contraction Bias or Central Tendency)

Romo & Salinas, Nature Reviews Neuroscience 4, 203-218; 2003

Performance



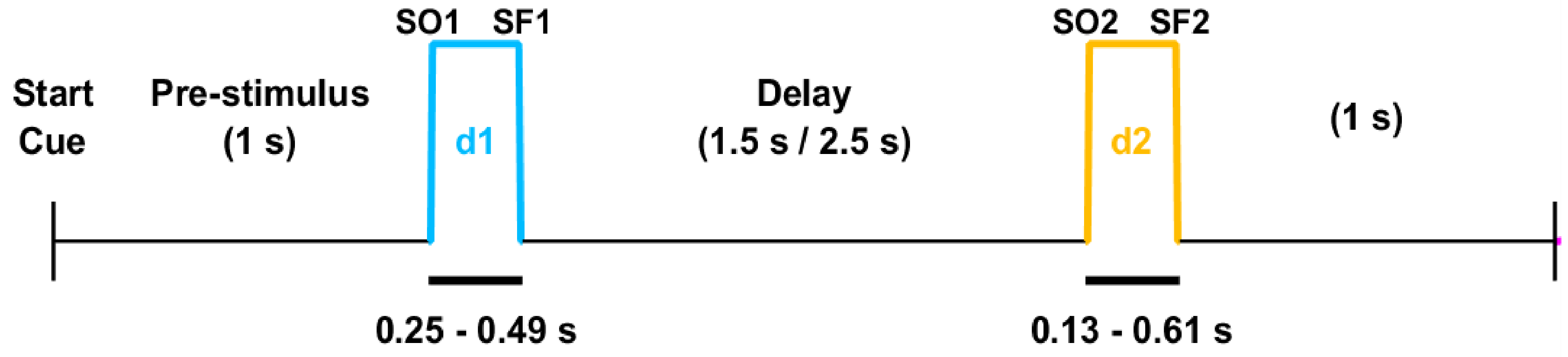
Perceptual Bias
(Contraction Bias or Central Tendency)

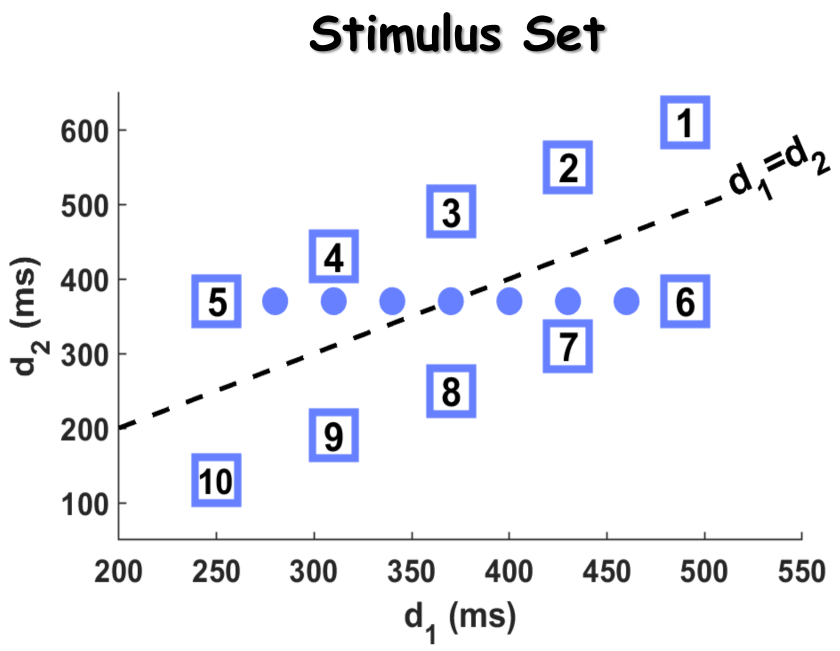


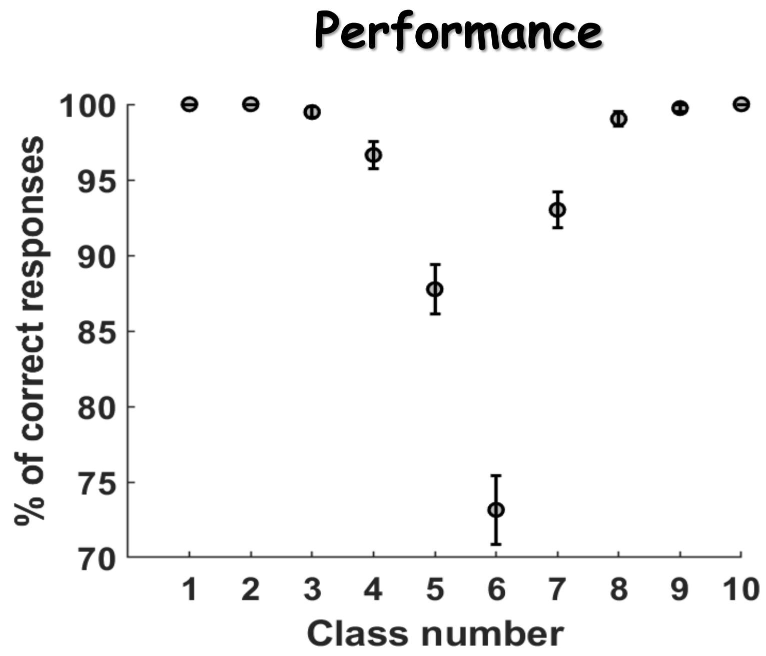
(anoher monkey, a different set)

Sarno, Beirán, Falcó-Roget, Diaz-deLeon, Rossi-Pool, Romo, & Parga , bioRxiv 2021

Time Interval Discrimination Task

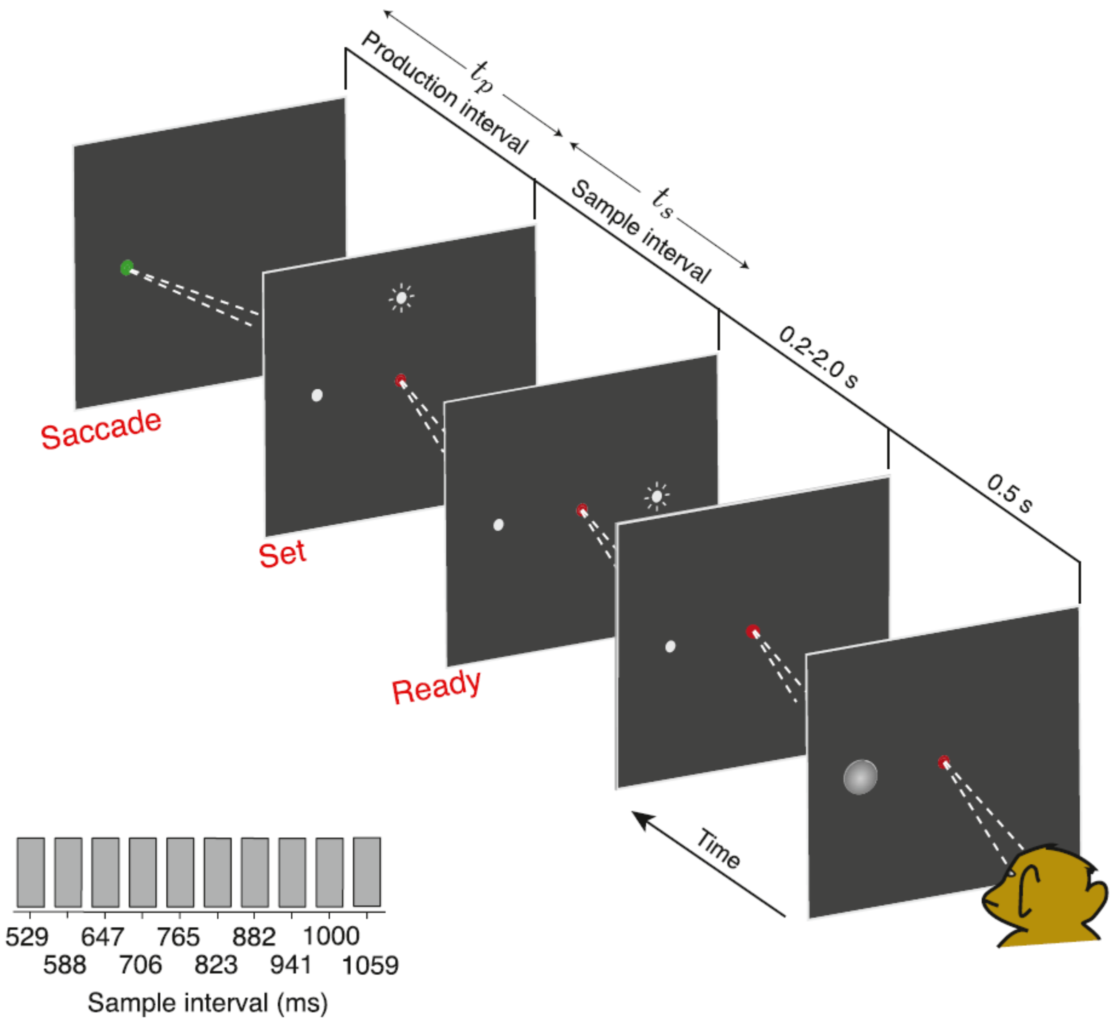






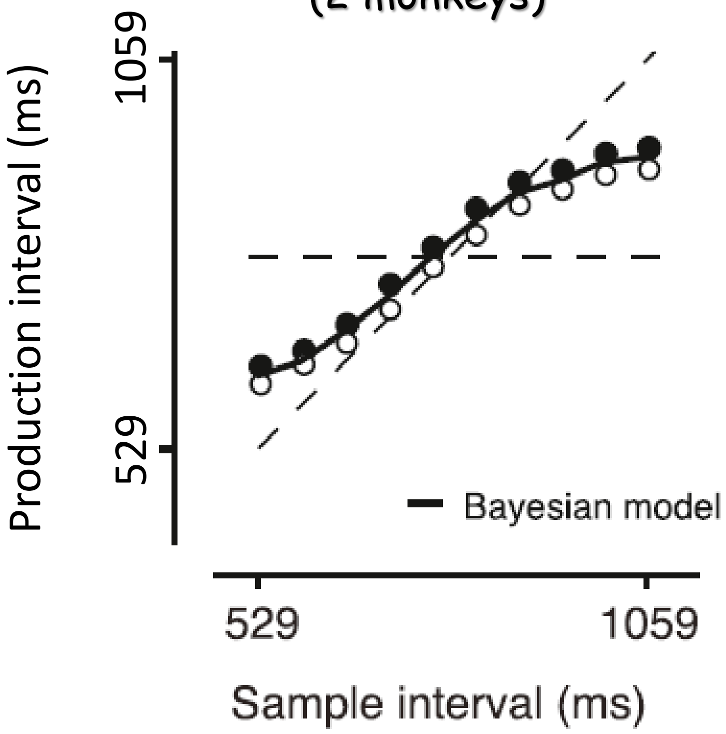
Serrano-Fernández, Beirán & Parga, 2015, in preparation

Time Interval Production Task



Stimulus Set

Performance
(2 monkeys)



Perceptual Bias
(Central tendency)

Jazayeri & Shadlen, 2015, Current Biology 25, 2599–2609

OUTLINE

A) Electrophysiological Experiments

B) Training Neural Networks: Artificial versus Biologically plausible features

C) Learning Algorithm: Reservoir learning, FORCE and full-FORCE

D) Detection Task: a RNN of Rate Neurons – State-space Analysis

E) Bayesian Computations in spiking RNNs

F) Beyond: Reinforcement Learning

Training of Neural Networks

some conditions for biological plausibility

We consider several features of model networks:

- Network Architecture,
- Model Neurons,
- Type of Learning,
- Learning Rules.

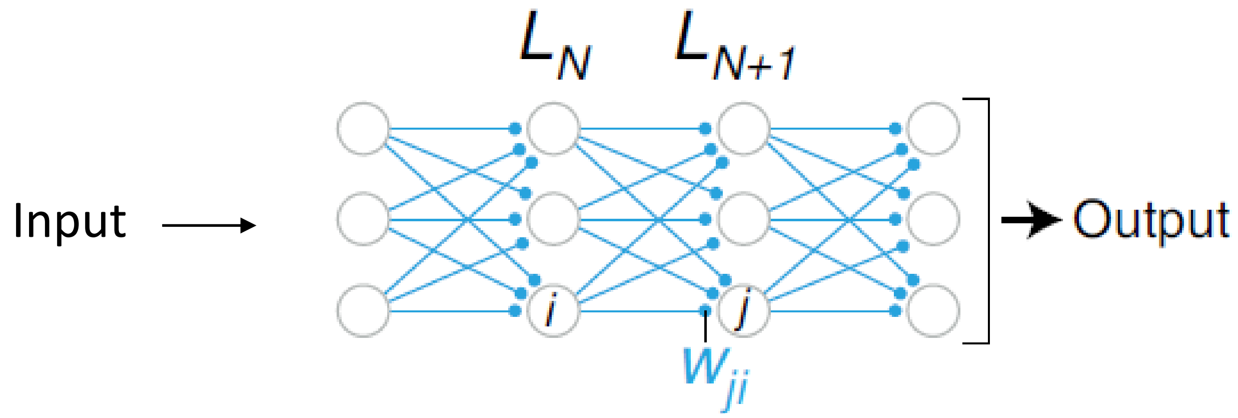
Biologically plausible trained networks:

- Network Architecture: Recurrent Neural Networks (RNNs),
- Model Neurons: Spiking Neurons,
- Type of Learning: Reinforcement Learning,
- Learning Rules: Hebbian learning rules, STDP, ecc.

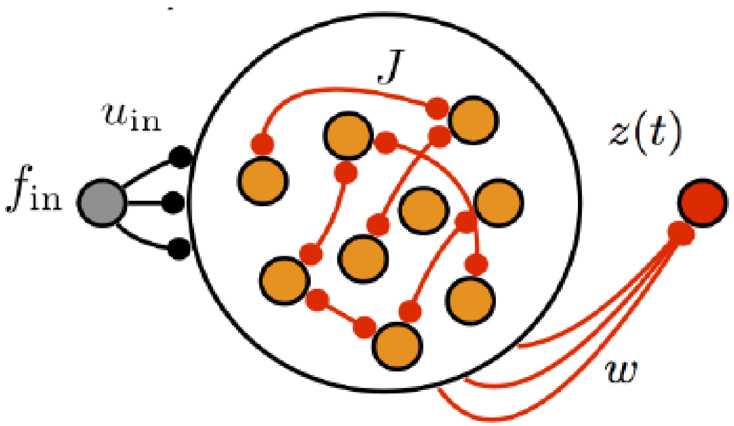
Ideally, the neural network and the training algorithm should be biologically realistic.

Network Architecture

Artificial arcuitectures:
Multilayer feedforward architecture (deep networks)



Biologically-plausible architectures:
***Recurrent architecture or Recurrent Neural Network* (RNN)**



Model Neurons

Artificial model neurons:

The activity of the neurons (units) are represented by real variables (continuous units)

Rate neurons

$$\tau \frac{d\mathbf{x}}{dt} = -\mathbf{x} + \mathbf{J}H(\mathbf{x}) + \mathbf{u}_{\mathrm{in}} f_{\mathrm{in}}(t)$$

$$H(\mathbf{x}) = \tanh(\mathbf{x}))$$

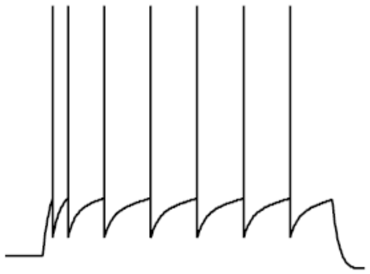
the "firing rates"

Biologically plausible models:

(spiking units)

Leaky integrate-and-fire neurons (LIF neurons)

$$\tau_{\mathrm{m}} \frac{dV}{dt} = E_{\mathrm{L}} - V - I_{\mathrm{syn}}(t)$$

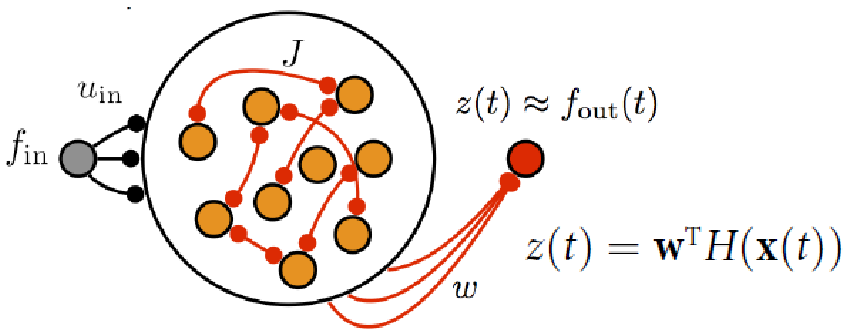


If V(t) > E_{th} then insert a spike and reset the membrane potential V(t) = E_L

Type of Learning

Artificial learning:
Supervised learning

Supervised learning: the correct output is known to the network. It is used as a **target** to determine the synaptic weights in such a way that a cost function is minimized



$$C_{\mathbf{w}} = \left\langle \left(z(t) - f_{\text{out}}(t)\right)^2 \right\rangle$$

average over time during a trial and training examples

Biologically-plausible learning:
Reinforcement learning

Uses a **reward** to indicate to the network if the action was correct or wrong

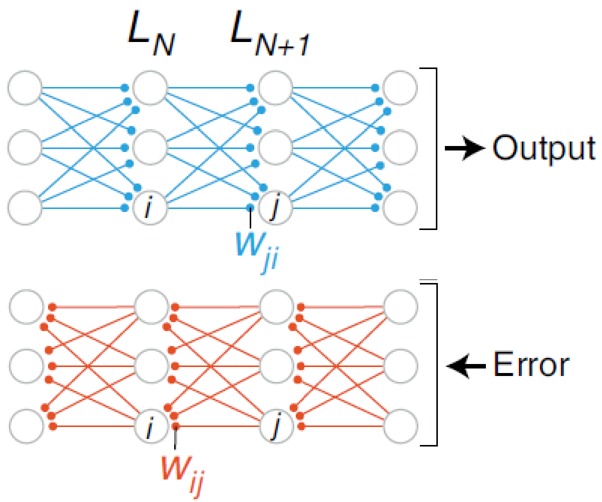
Learning rule

Artificial rules:

- ***Backpropagation*** (*Deep learning*)
- ***FORCE & full-FORCE***
- ***Ecc***

To assign the credit of errors to the weights the backprop algorithm computes the derivative of the cost function with respect to the weights.

This generates a non-local learning rule.



Biologically-plausible rules:

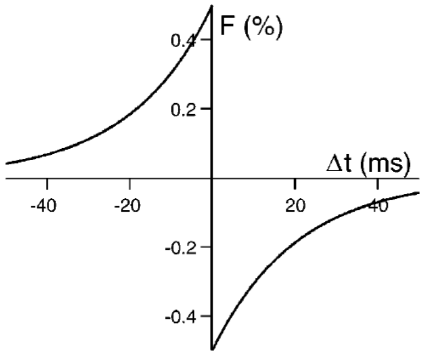
1. ***Hebbian rules***
2. ***Spike-timing-dependent plasticity (STDP)***
3. ***Dopamine-modulated synapses***
4. ***Ecc.***

1. Hebbian learning: "*neurons that fire together wire together*"

D Hebb , The organization of behavior, 1949

2. STDP: *amount of synaptic modification arising from a single pair of pre- and postsynaptic spikes separated by a time Δt. Time scale are about 20 ms.*

Bi, G.-q. & Poo, M.-m., J. Neurosci. 18, 10464–10472, 1998



$$F(\Delta t) = \begin{cases} A_+ \exp(\Delta t/\tau_+) & \text{if } \Delta t < 0 \\ -A_- \exp(-\Delta t/\tau_-) & \text{if } \Delta t \geq 0 \end{cases}$$

Song, Miller and Abbott, Nature Neuroscience 3, 919–926, 2000.

Fully biologically plausible trained networks are hard to obtain.

However, one can relax some constraints and investigate how the trained networks solve the tasks.

This can be used:

- to guide the analysis of experimental data. For instance, find out whether cortical networks use computational strategies similar to those observed in the computational models.

- help designing new experiments. Simulating the trained networks one can give information about the possible experimental results.

I will present results from a spiking RNN that solves a task for which experiments have not been done.

OUTLINE

A) Electrophysiological Experiments

B) Training Neural Networks: Artificial versus Biologically plausible features

C) Learning Algorithm: Reservoir learning, FORCE and full-FORCE

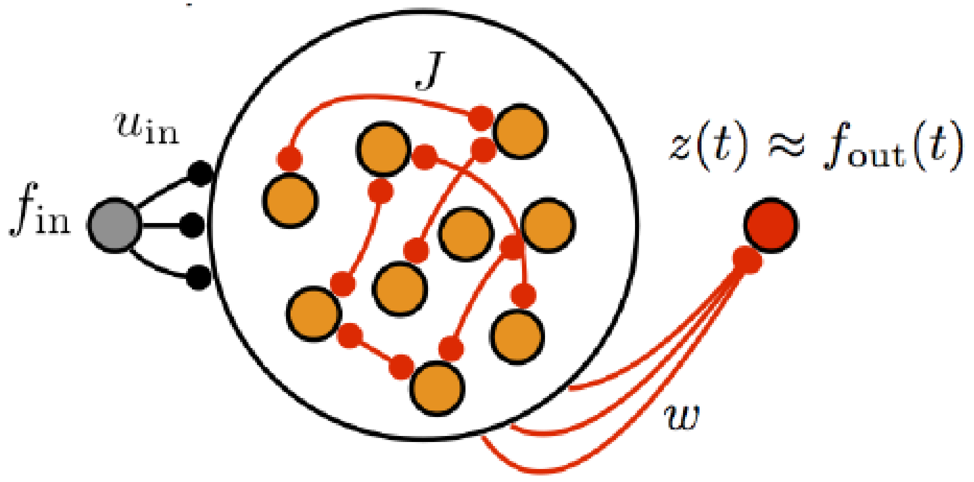
D) Detection Task: a RNN of Rate Neurons – State-space Analysis

E) Bayesian Computations in spiking RNNs

F) Beyond: Reinforcement Learning

Learning algorithms

Consider a RNN of rate neurons and supervised learning, with a target f_{out}



the task-performing network

$$\tau \frac{d\mathbf{x}}{dt} = -\mathbf{x} + \mathbf{J}H(\mathbf{x}) + \mathbf{u}_{in} f_{in}(t)$$

$$H(\mathbf{x}) = \tanh(\mathbf{x}))$$

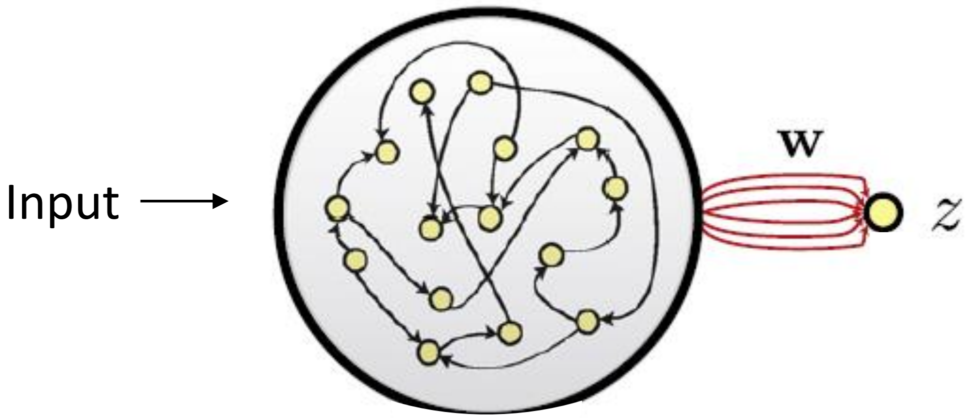
$$z(t) = \mathbf{w}^{\mathrm{T}} H(\mathbf{x}(t))$$

Plastic synapses: red lines

Cost function

$$C_w = \langle (z(t) - f_{out}(t))^2 \rangle$$

Reservoir Learning

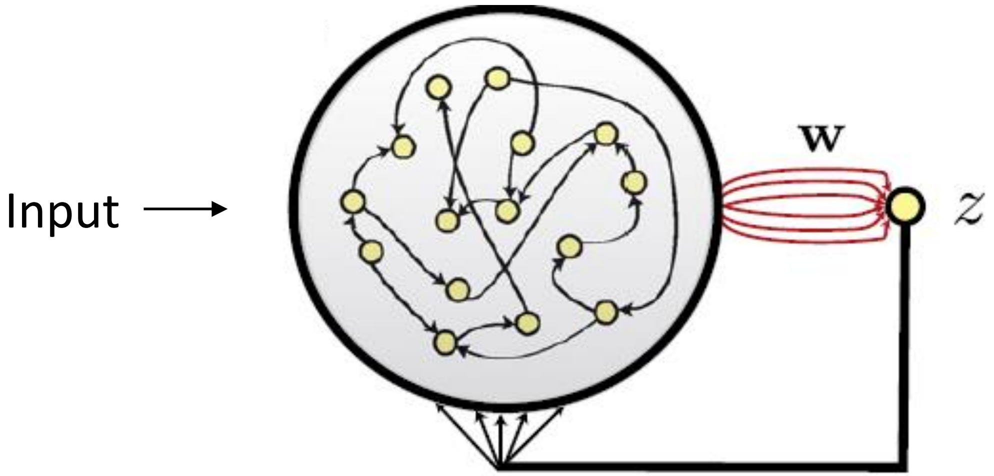


Reservoirs are RNNs with **fixed connections** that are randomly generated according to obtain rich spatial and temporal representations.

A linear output neuron reads the activity of the RNNs and the output weights w are learnt such that the network selects the correct actions given inputs.

Learning is simple but it has limited performance.

The FORCE algorithm



Performance can be improved by feeding the output back into the network (with randomly chosen weights **u)**

Jaeger & Haas, Science, 2004

$$\tau \frac{d\mathbf{x}}{dt} = -\mathbf{x} + \mathbf{J}H(\mathbf{x}) + \mathbf{u}_{\text{in}} f_{\text{in}}(t) + \mathbf{u} z(t) \qquad z(t) = \mathbf{w}^{\mathrm{T}} H(\mathbf{x}(t))$$

The feedback term can be considered as an additive term $\mathbf{uw}^T$ to the recurrent weights.
The plastic changes in the output weights induce changes in the recurrent weights.

FORCE learning: results from combining this network with a **recursive least-squares algorithm** for minimizing C_w

$$C_{\mathbf{w}} = \left\langle (z(t) - f_{\text{out}}(t))^2 \right\rangle$$

FORCE: First-Order Reduced and Controlled Error algorithm

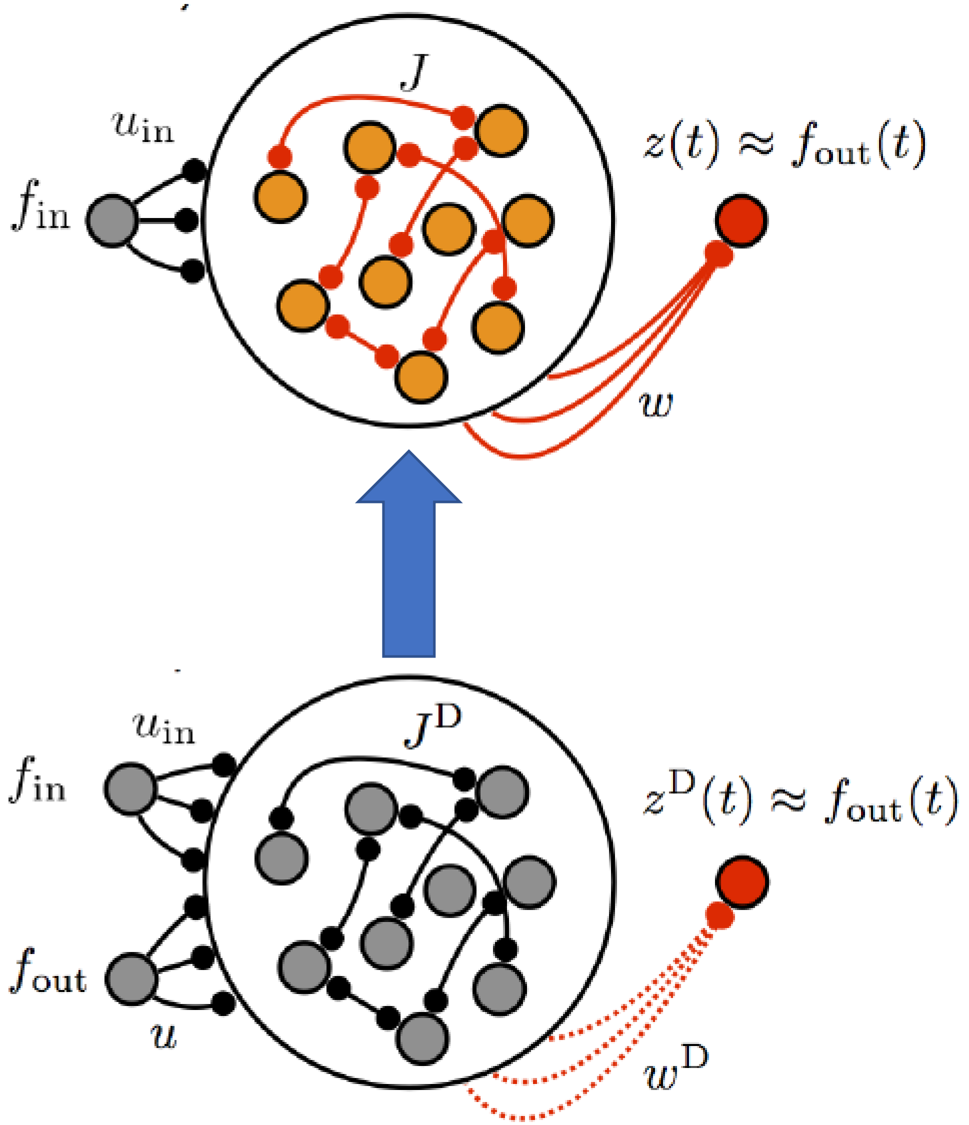
Sussillo & LF Abbott, Neuron, 2009

However, in the FORCE algorithm modifications of the recurrent weights are limited:

- it is low rank (rank one): $J \rightarrow J + \mathbf{uw}^T$
- it depends on the changes in the output weights.

The full-FORCE algorithm

the task-performing network



the target-generating network

The problem to use the recurrent connections as plastic weeights is that we do not know the targets for the hidden neurons.

Full-FORCE considers an auxiliary network that receives the target as an input and uses the currents as targets for the currents in the task-performing network

The **task-peforming network** can be either a network of rate neurons or a **spiking RNN**.

The **target-generating network** is a network of rate neurons.

DePasquale, LF Abbott et al., PLOS one, 2018

OUTLINE

A) Electrophysiological Experiments

B) Training Neural Networks: Artificial versus Biologically plausible features

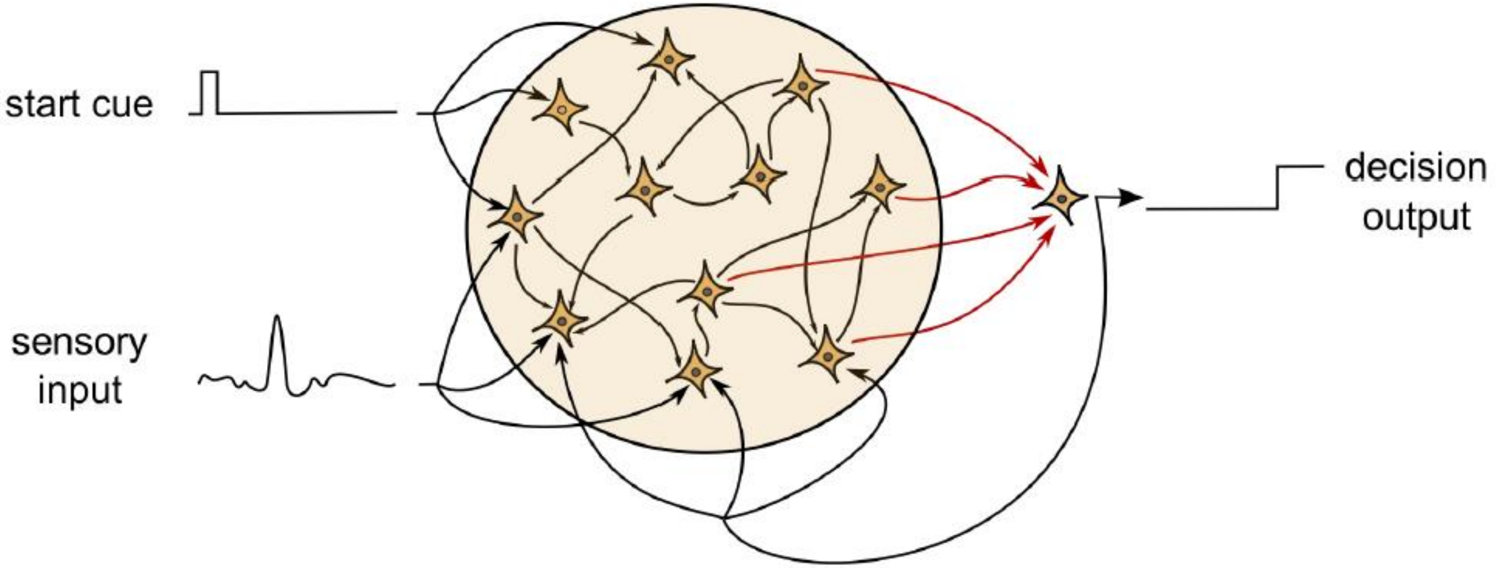
C) Learning Algorithm: Reservoir learning, FORCE and full-FORCE

D) Detection Task: a RNN of Rate Neurons – State-space Analysis

E) Bayesian Computations in spiking RNNs

F) Beyond: Reinforcement Learning

Detection: RNN model with rate neurons (FORCE)



N=500

$$\tau \frac{dx_i}{dt} = -x_i + g\sum_{j=1}^{N} J_{ij} r_j + w_i^{fb} z + w_i^{start} u_{start} + w_i^{stim} u_{stim}$$

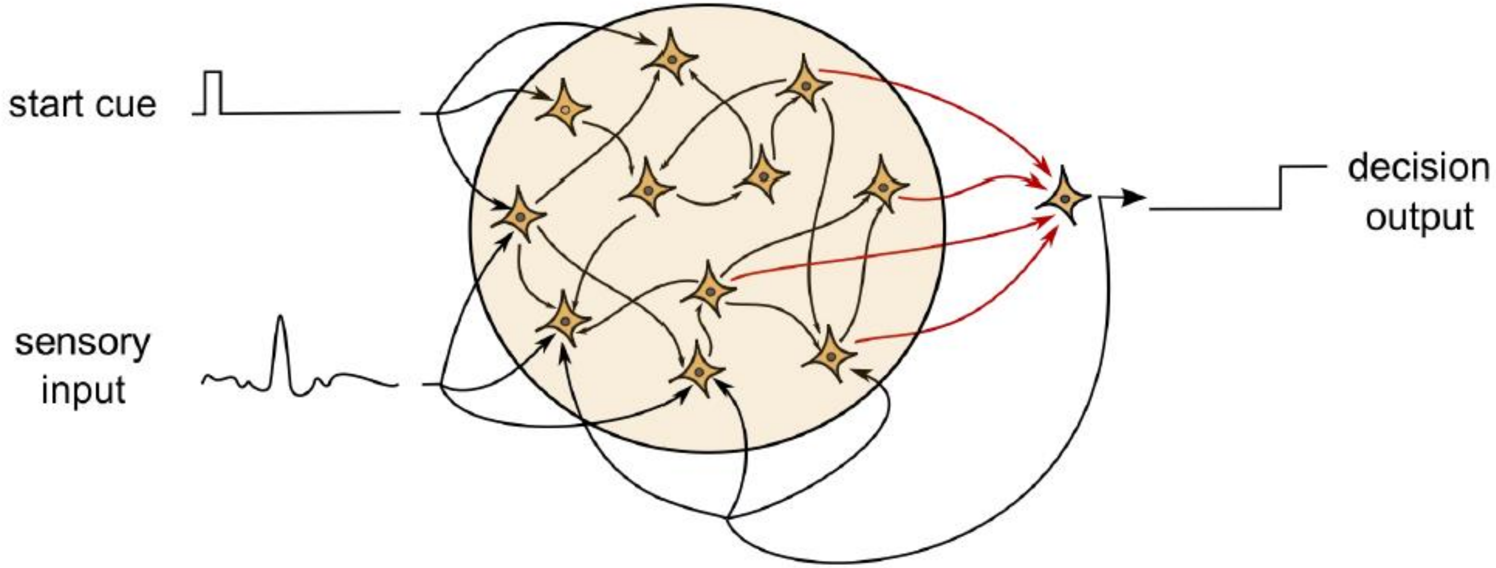
$r_i = \tanh(x_i)$ *are the "firing rates"*

$z = \sum_{i=1}^{N} w_i^{out} r_i$ *is the network's output, it is used as a feedback signal*

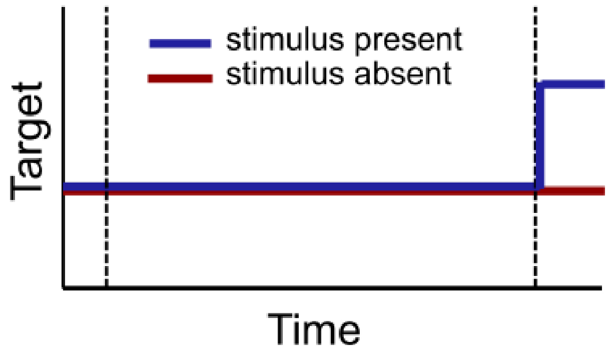
u_{start} *start cue (a 100ms pulse)*

u_{stim} *sensory input (a 300ms pulse, proportional to the stimulus amplitude, plus noise)*

Carnevale, de Lafuente, Romo, Barak & Parga (2015). Neuron, 86(4), 1067-1077.



target signal



No information about task timing is given!

The information given during training was restricted to the behavioral outcome on each trial.

Supervised learning rule

*Affects only the **output** couplings:*

$$W^{out} = W^{out} - e(t)Pr(t)$$

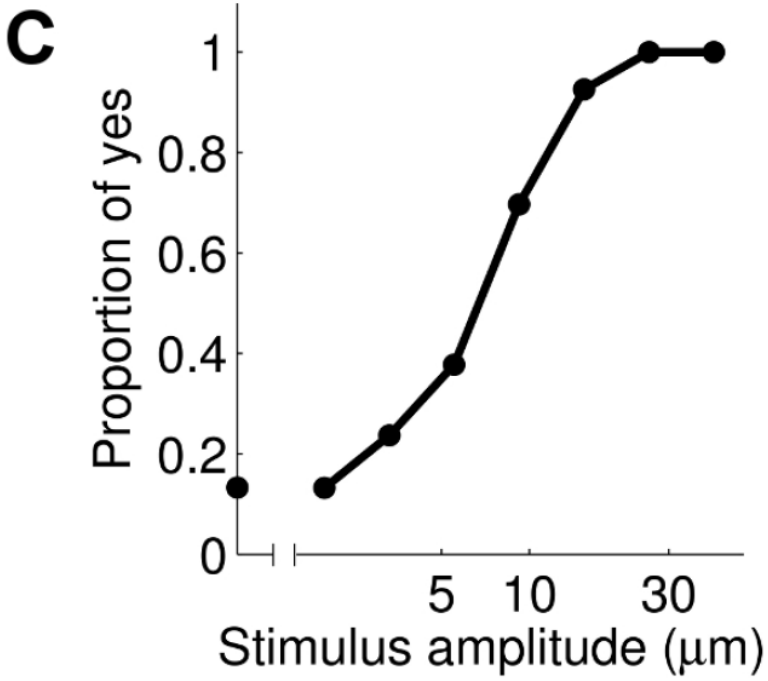
$$e(t) = z(t) - target(t)$$

*Because of the feedback of the output unit z, this rule effectively changes the **recurrent** couplings to:*

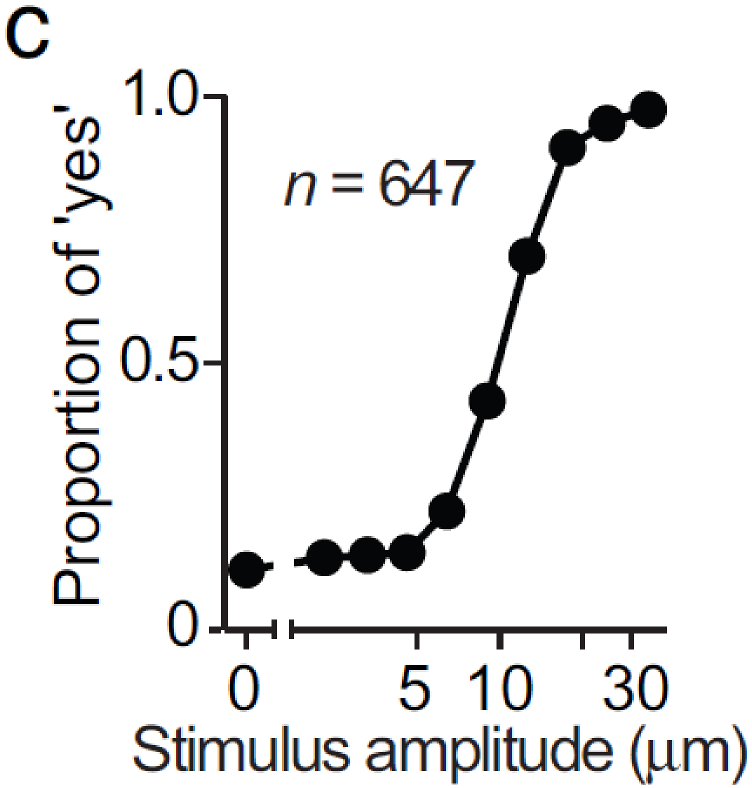
$$\mathbf{J}_{\text{eff}} = \left(g\mathbf{J} + \mathbf{w}^{fb}\mathbf{w}^{out'}\right)$$

The network learns to solve the task

The noise amplitude was calibrated to approximately reproduce the experimental psychometric function.

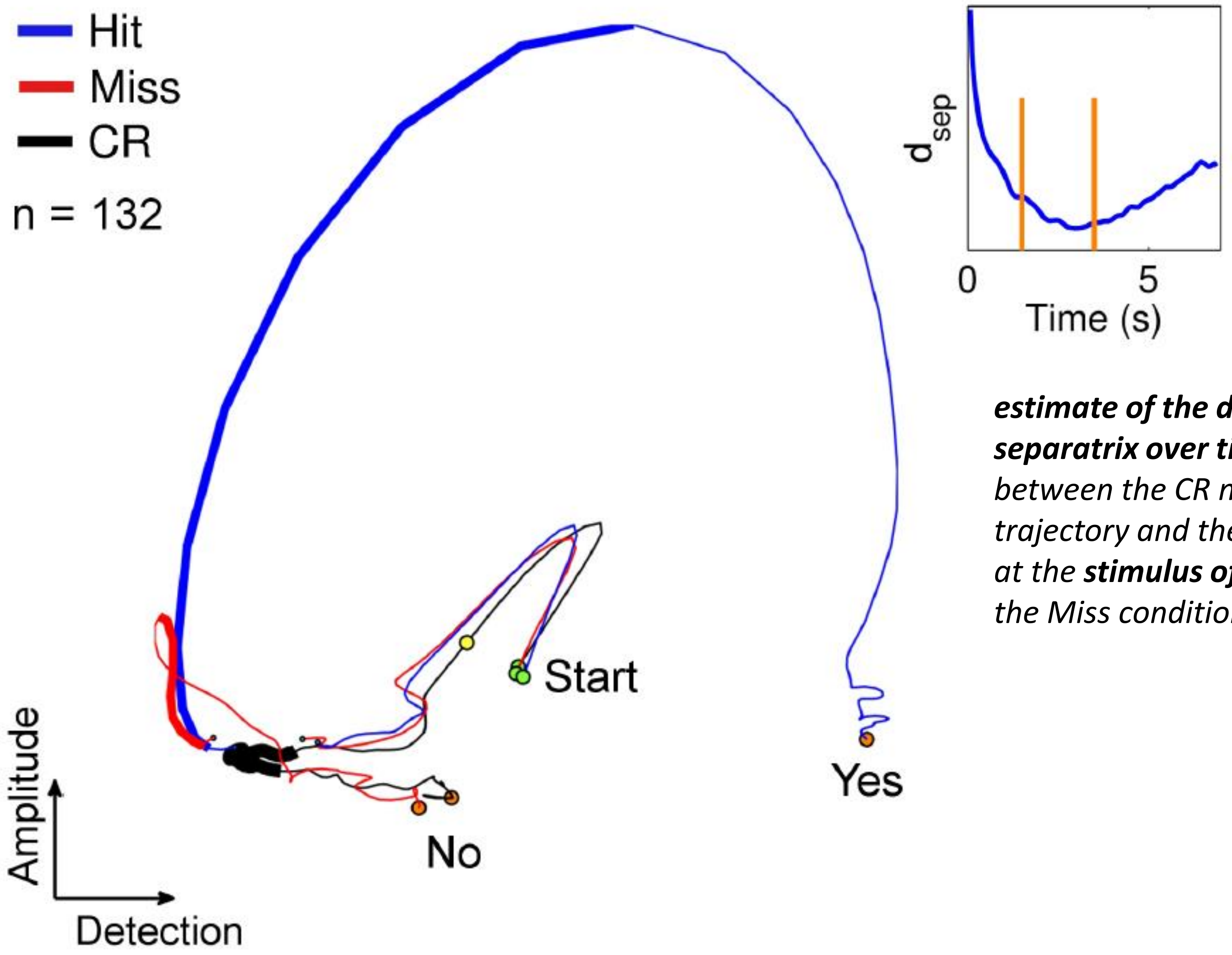


MODEL



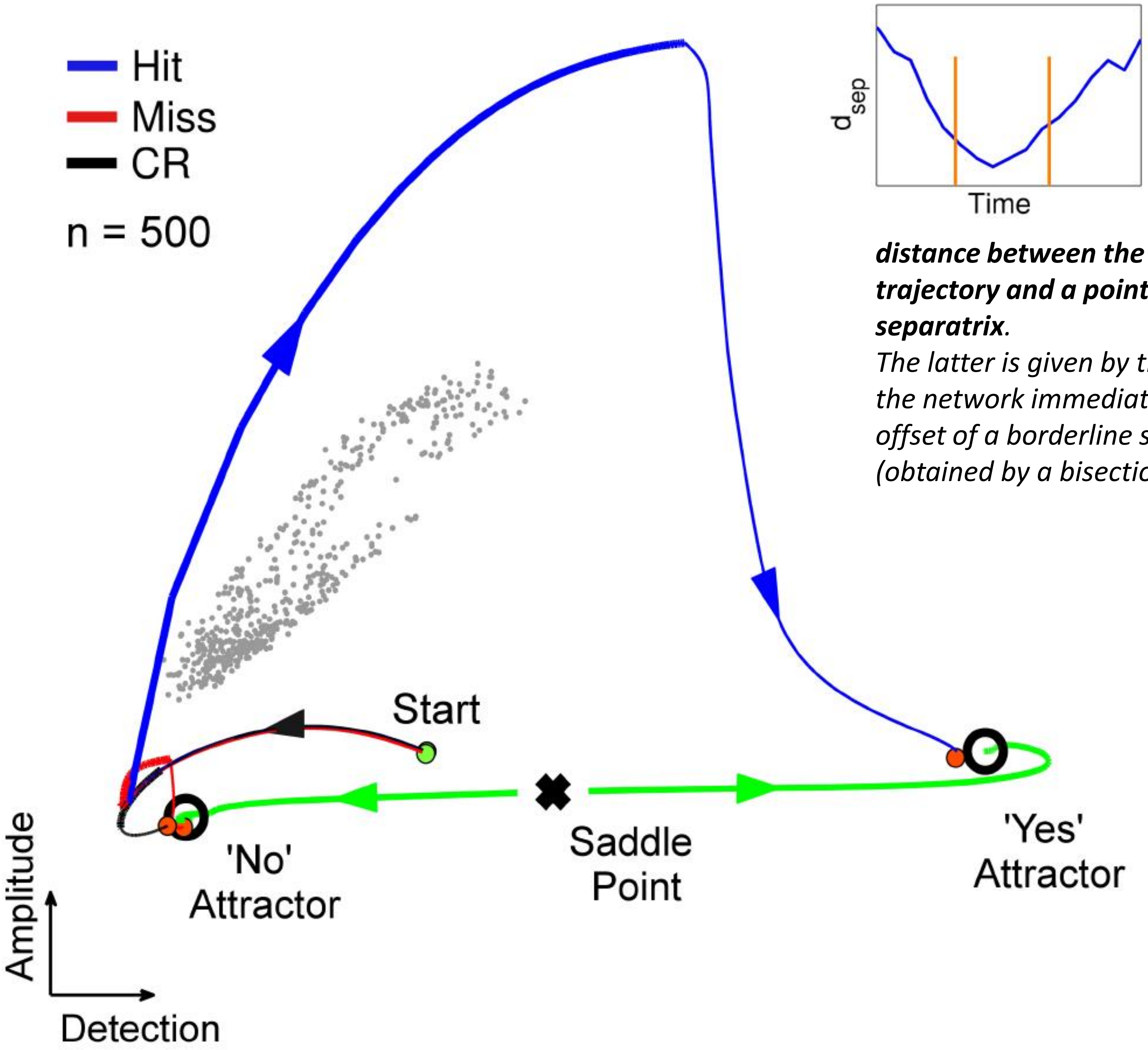
EXPERIMENT

Neural dynamics of perceptual detection



estimate of the distance to the separatrix over time: *distance between the CR neural trajectory and the neural state at the* ***stimulus offset*** *time in the Miss condition*

The dynamic mechanism supporting the modulation of the RC



distance between the CR neural trajectory and a point in the separatrix*.*

The latter is given by the state of the network immediately after the offset of a borderline stimulus (obtained by a bisection method)

OUTLINE

A) Electrophysiological Experiments

B) Training Neural Networks: Artificial versus Biologically plausible features

C) Learning Algorithm: Reservoir learning, FORCE and full-FORCE

D) Detection Task: a RNN of Rate Neurons – State-space Analysis

E) Bayesian Computations in spiking RNNs

F) Beyond: Reinforcement Learning

Bayesian Computations in spiking RNNs (full-FORCE)

Now I will train spiking RNNs with the full-FORCE algorithm for the following tasks

- temporal Interval discrimination
- tactile frequency discrimination
- time Interval production

and show that in all of them **the first stimulus is represented in terms of a Bayesian estimator**.

The contraction bias results from Bayesian computations.

Work in collaboration with:

Luis Serrano-Fernández (UAM, Madrid)

Manuel Beirán (ENS, Paris)

Pablo Crespo Darriba (URJC, Madrid)

Martín Zamarbide (UAM, Madrid)

$$P(s,r) = P(r \mid s)\, P(s)$$
$$= P(s \mid r)\, P(r)$$

$\Longrightarrow$

$$P(s \mid r) \;\alpha\; P(r \mid s)\, P(s)$$

Bayes relationship

s: **stimulus (state of the world)**
r: **firing rate (noisy internal representation of the stimulus)**

$P(s)$ prior probability

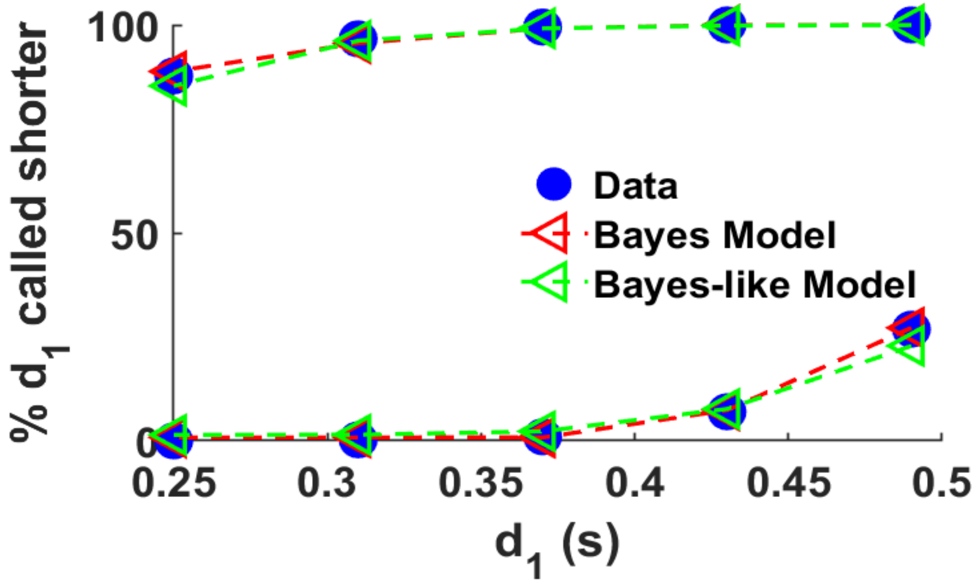
$P(r|s)$ likelihood (noise model)

$P(s|r)$ posterior probability (belief about the state of the world)

In the **Bayesian models of the tasks,** after the stimulus is presented we can obtain a Bayesian estimator of d1, f1 or ts.

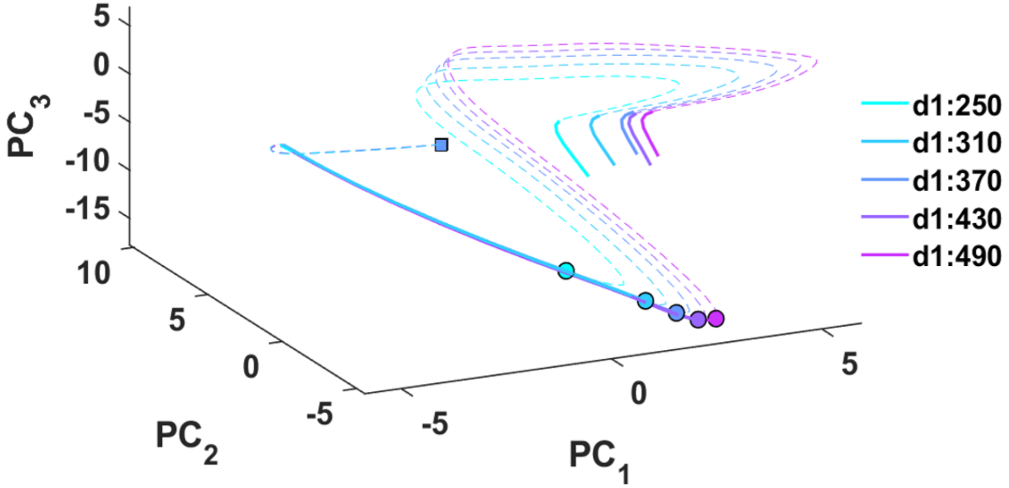
Time Interval Discrimination Task: Solution with spiking RNNs

Performance



BAYESIAN MODEL FIT

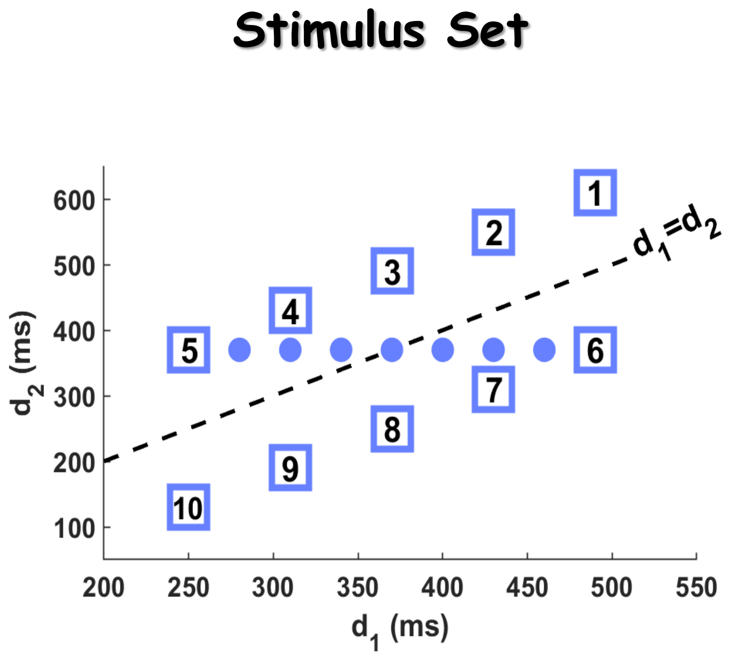
Population activity (state space)

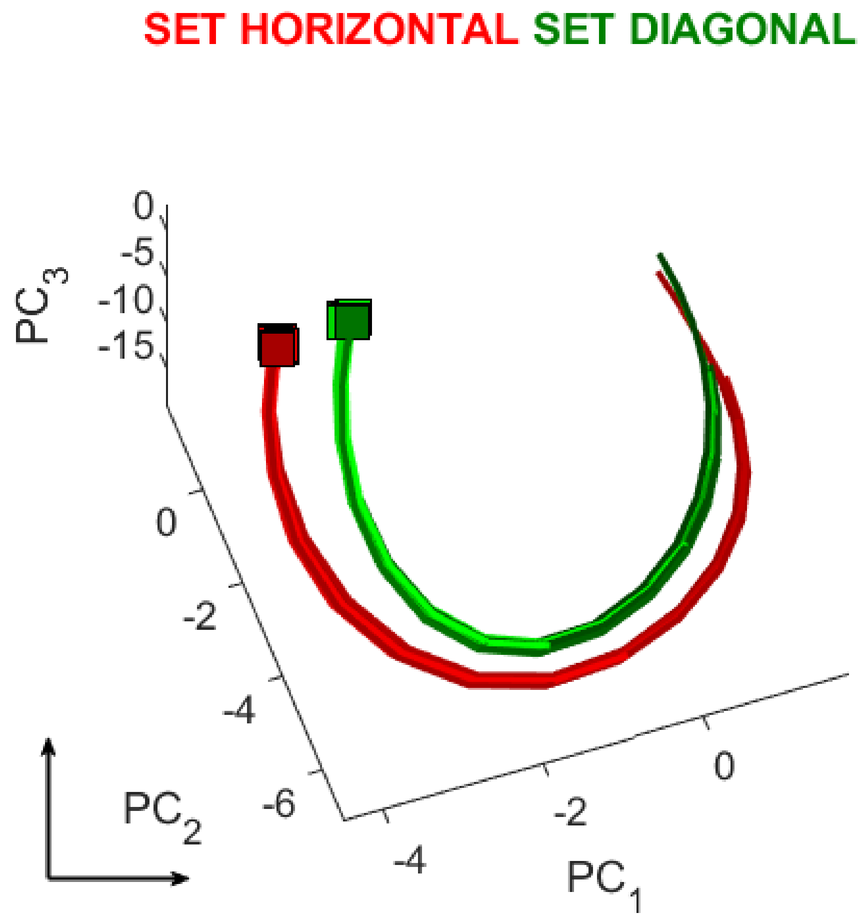


Dimensionality reduction

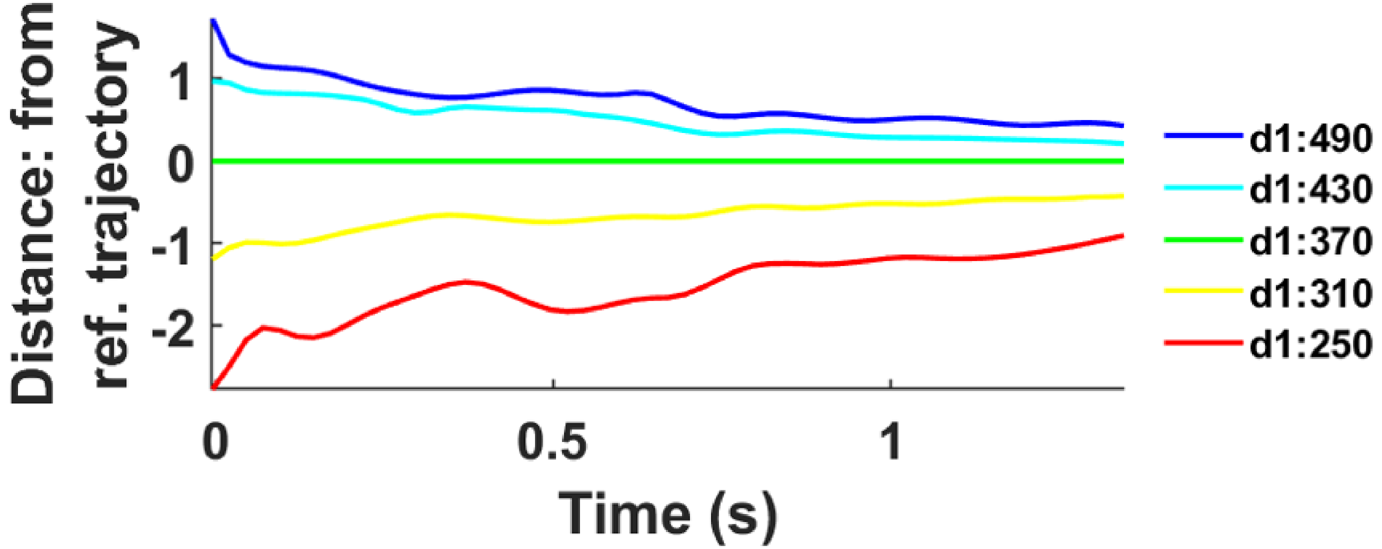
Trajectories describe **orbits** even if there not cyclic components in the task

Prior-dependent trajectories





Relative distances between state space trajectories (delay period)



It is reasonable to asume that the relative distances depend on the time Interval d1, AS PERCEIVED BY THE NETWORK.

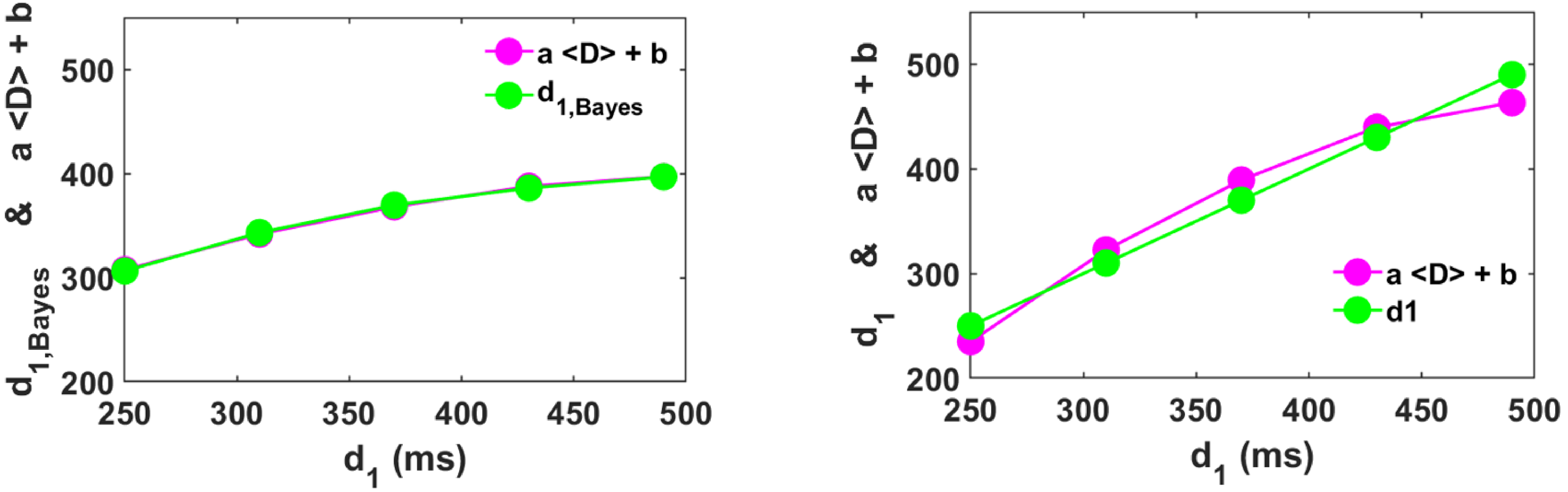
Is the decoded d1 the true value of d1 or is it closer to the Bayesian estimator?

$$H_1 : d_1 = \alpha \langle D_i \rangle + \xi$$

$$H_2 \;\; d_{1,e} = \alpha \langle D_i \rangle + \xi$$

Comparing Behavior and Neural Population Activity

Two hypotheses relating behavior (bias) with state space structure were tested: (i) mean distances code d_1 or (ii) code a combination of current d_1 and prior knowledge of d_1 (a Bayesian estimator)

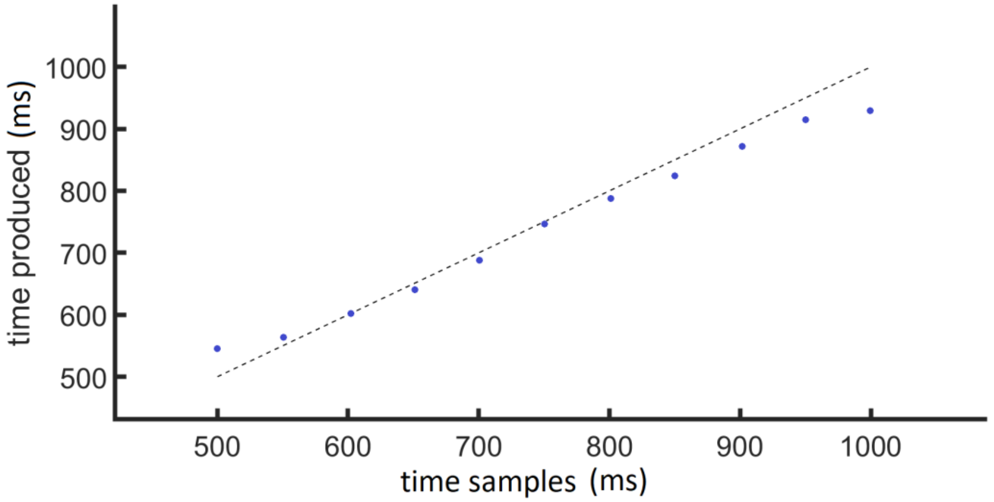


True d_1: RMSE = 17.68

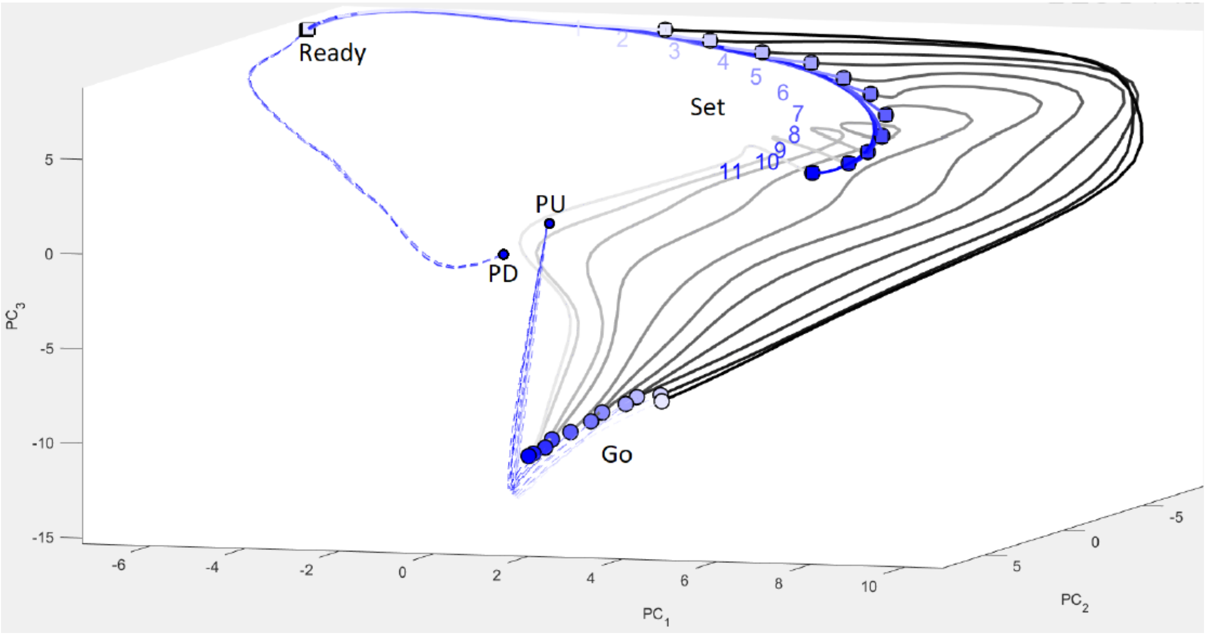
Bayesian estimator: RMSE = 1.38

Time Interval Production Task: Solution with spiking RNNs

Performance

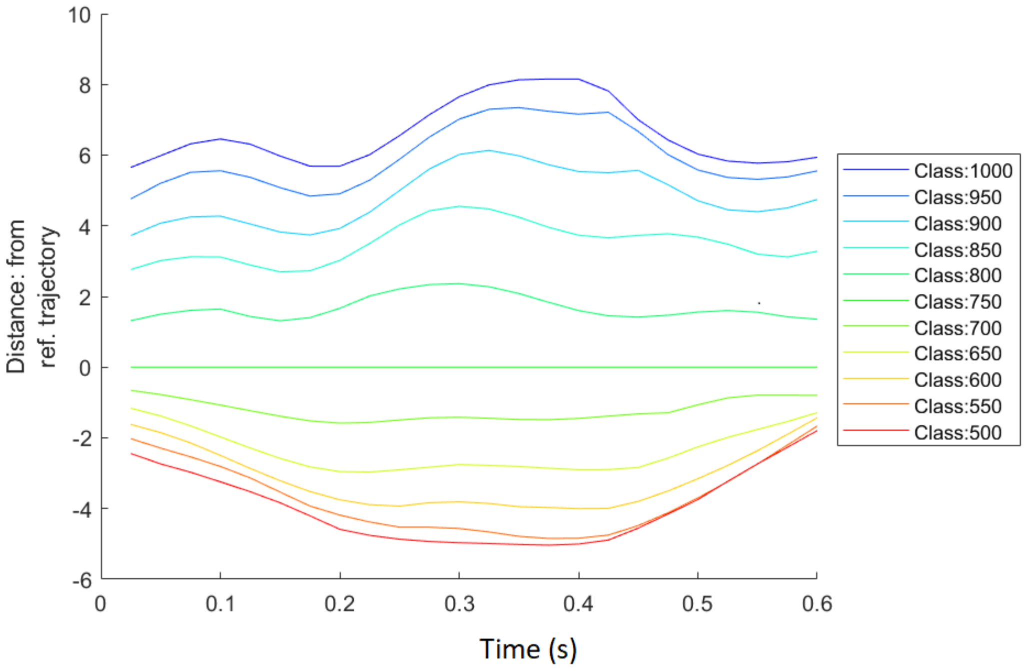


Population activity (state space)



The stimulus set contains time intervals from 500 ms to 1000 ms, with a difference of 50 ms between consecutive simples (11 classes)

Relative distances between state space trajectories

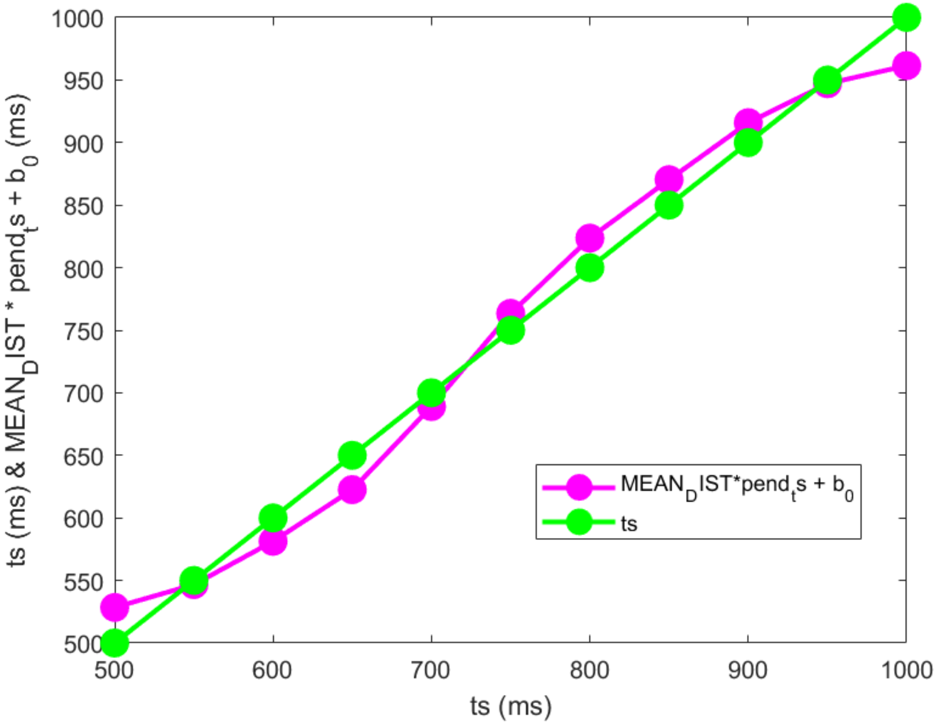


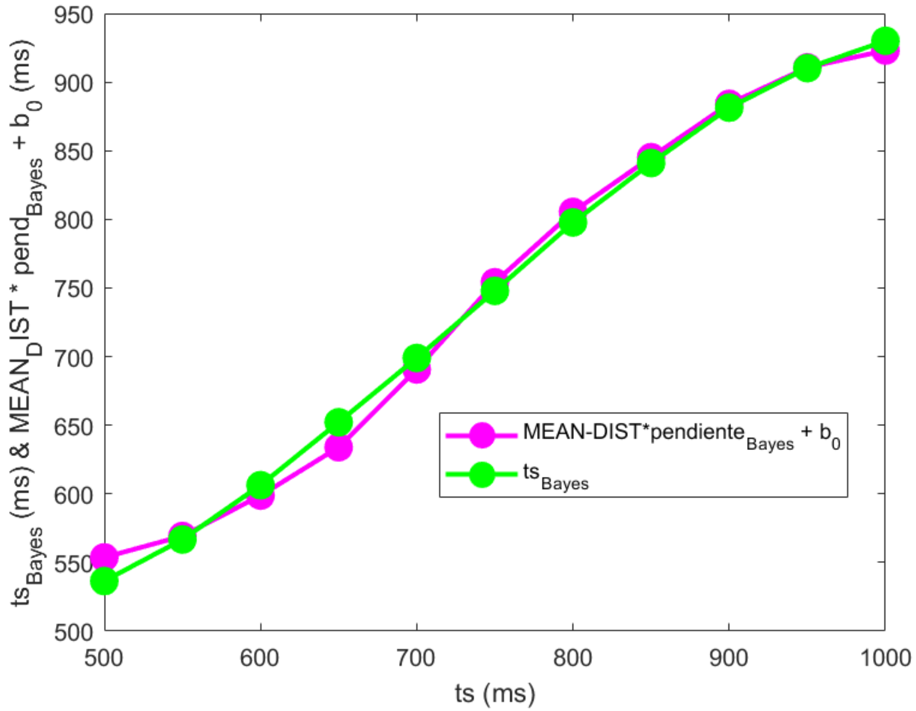
$$H_1 : ts = \alpha \langle D_i \rangle + \xi$$

$$H_2 : te = \alpha \langle D_i \rangle + \xi$$

Is the decoded ts the true value of ts or is it closer to the Bayesian estimator?

Comparing Behavior and Neural Population Activity

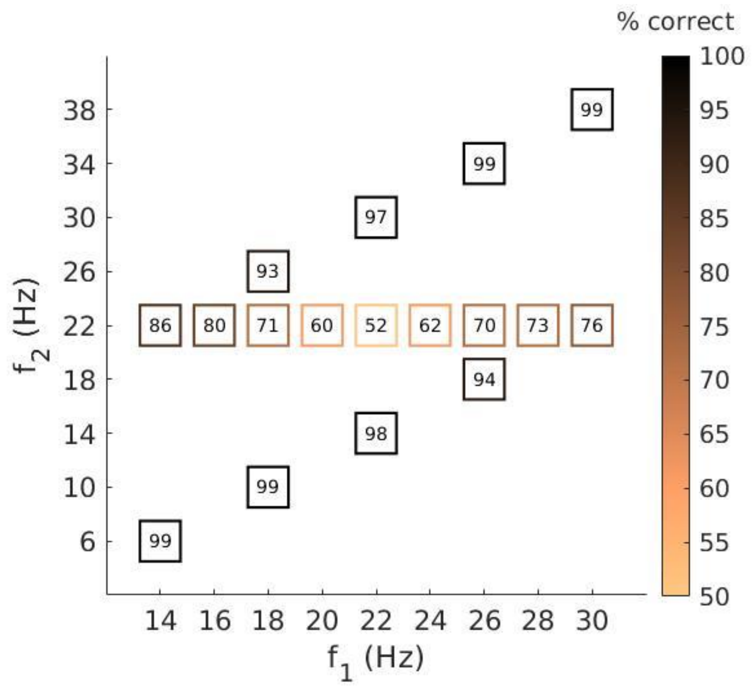


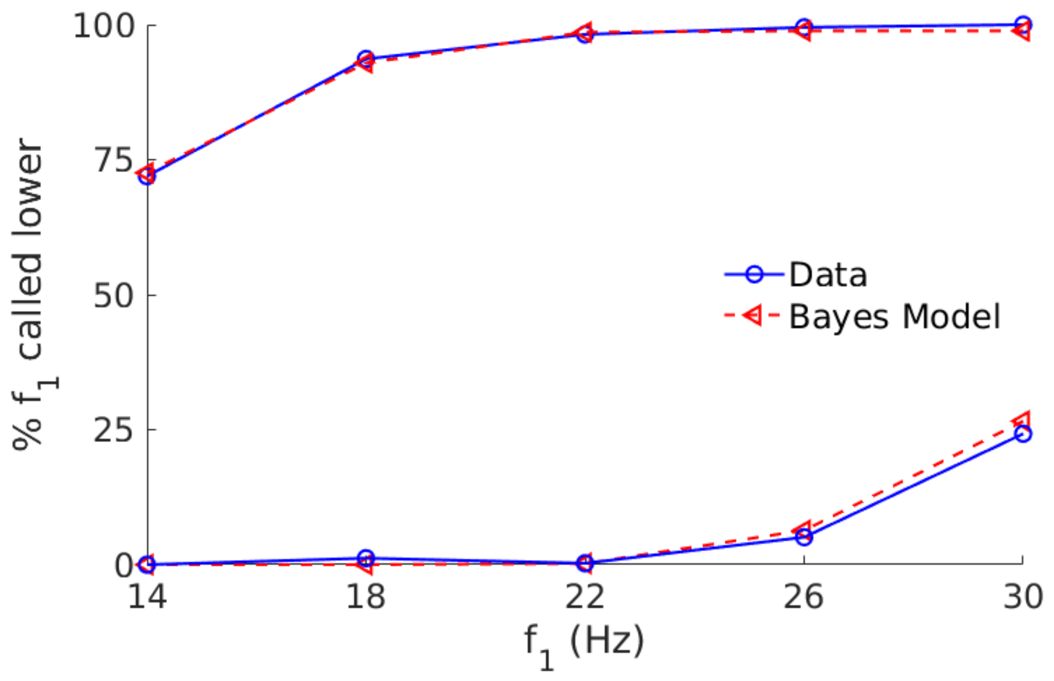


A statistical test compring the distributions of the RMSEs of the two hypotheses favoured the Bayesian estimator ($p > 0.0001$)

Tactile Frequency Discrimination Task: Solution with spiking RNNs

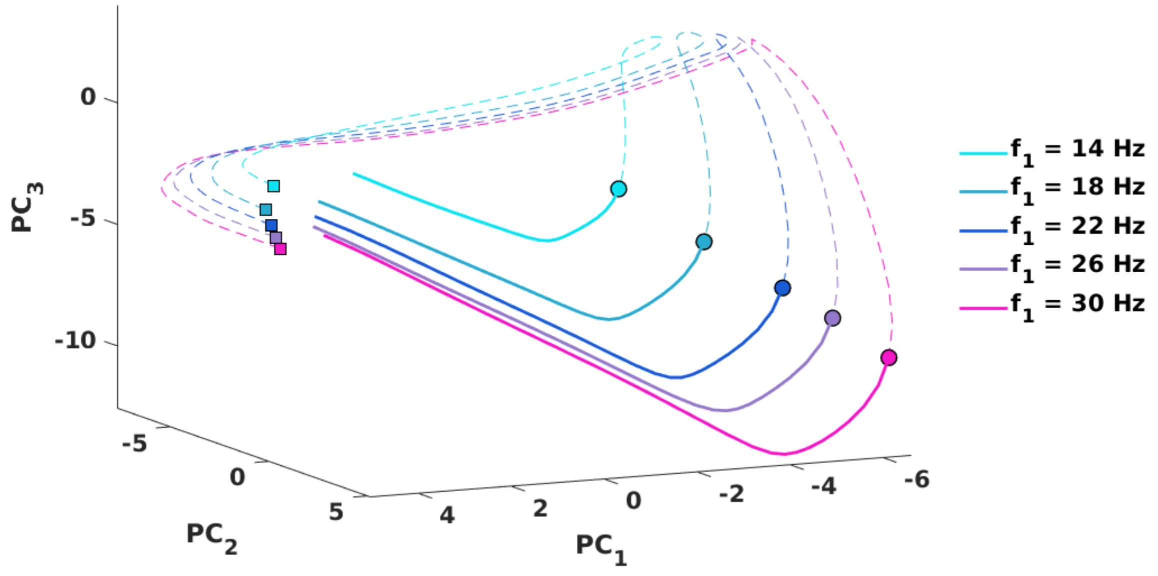
Performance



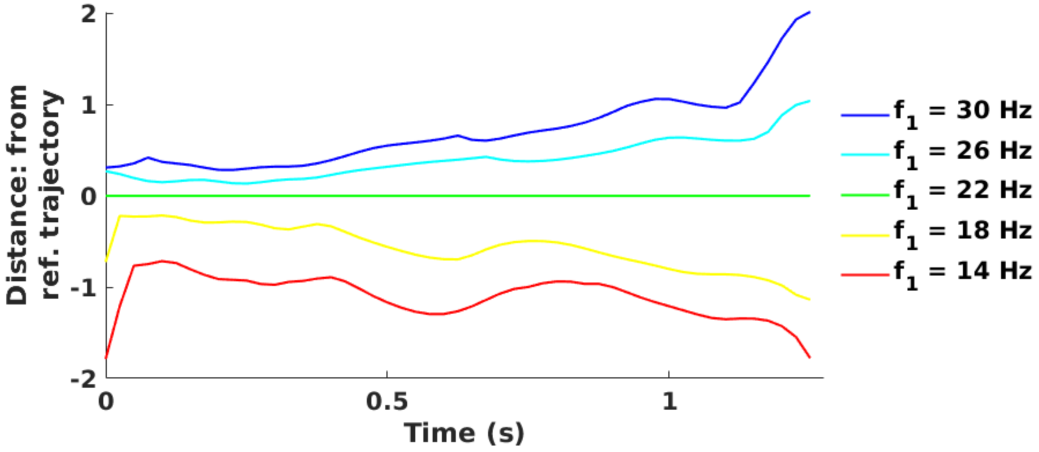


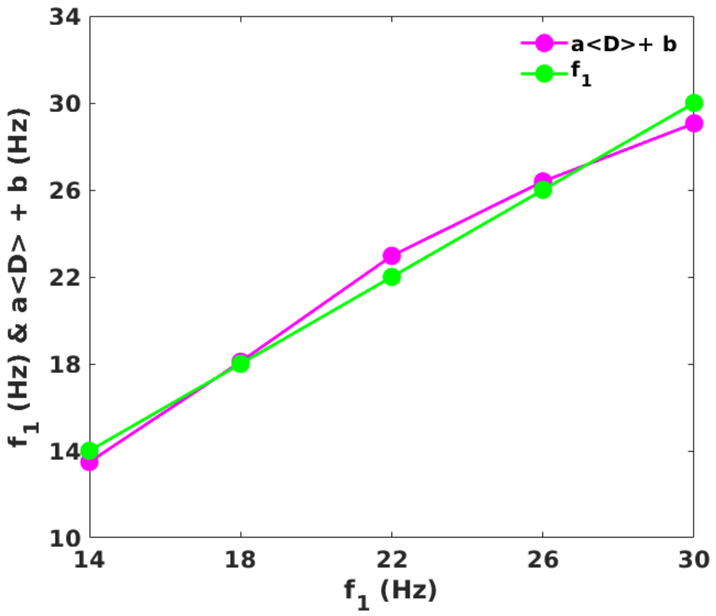
BAYESIAN MODEL FIT

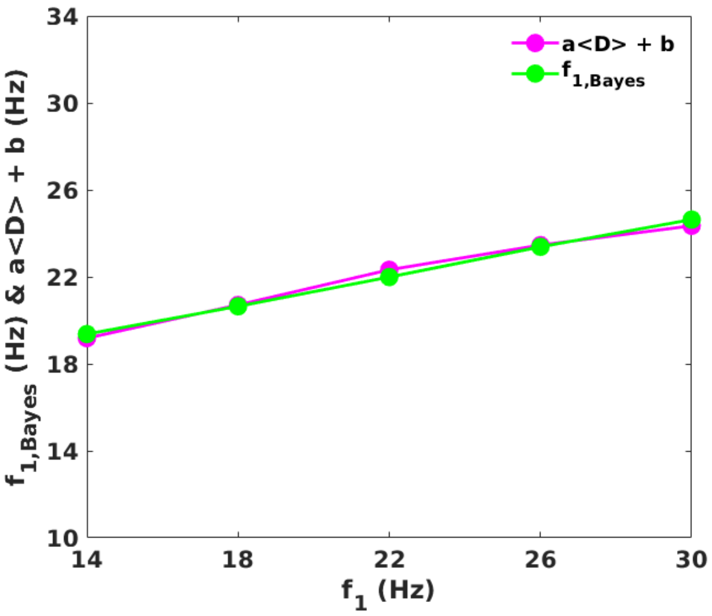
Population activity
(state space)



Relative distances between state space trajectories





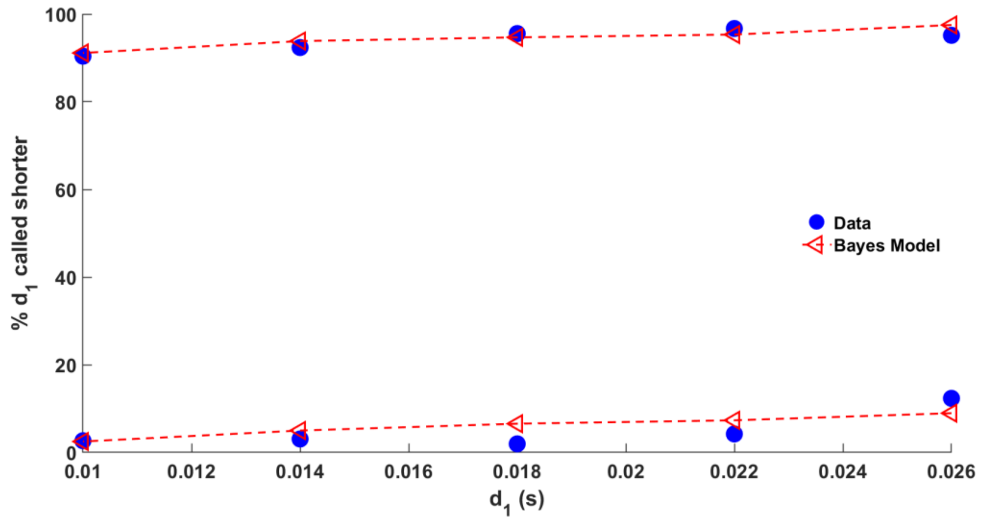


True f1: RMSE = 0.67

Bayesian estimator: RMSE = 0.22

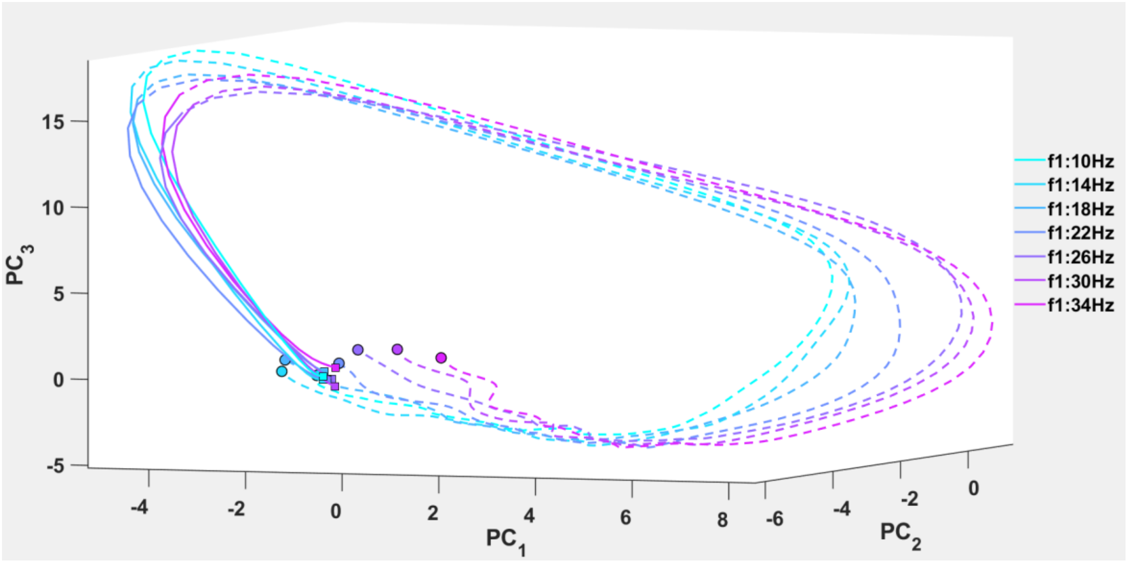
Tactile frequency discrimination task: Experiment (data analysis)

Performance



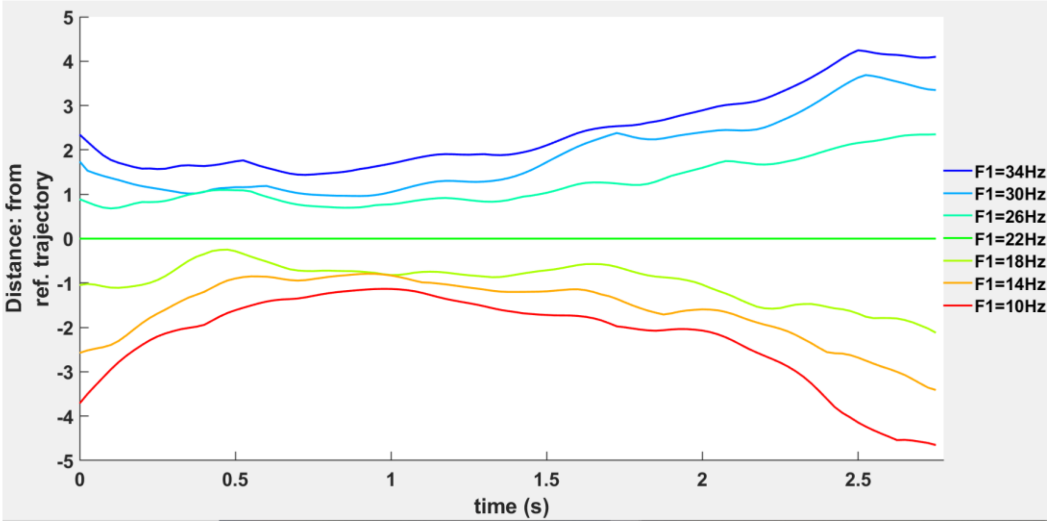
BAYESIAN MODEL FIT

Population activity (state space)



Serrano-Fernández, Romo, Parga & lab members UNAM (in preparation)

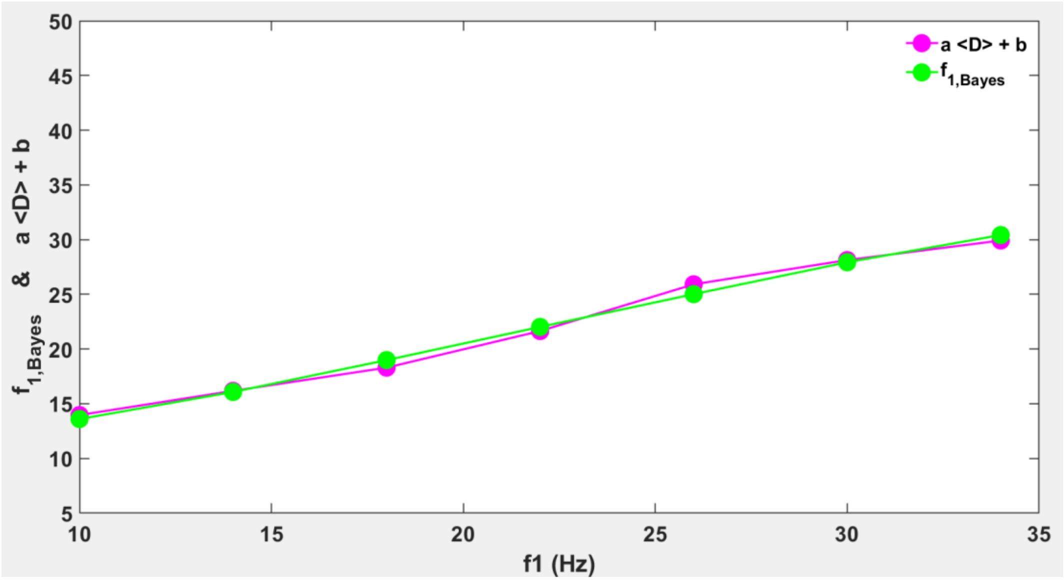
Comparing Behavior and Neural Population Activity



True f_1 : RMSE = 0.89

Bayesian estimator: RMSE = 0.51

(one monkey)



OUTLINE

A) Electrophysiological Experiments

B) Training Neural Networks: Artificial versus Biologically plausible features

C) Learning Algorithm: Reservoir learning, FORCE and full-FORCE

D) Detection Task: a RNN of Rate Neurons – State-space Analysis

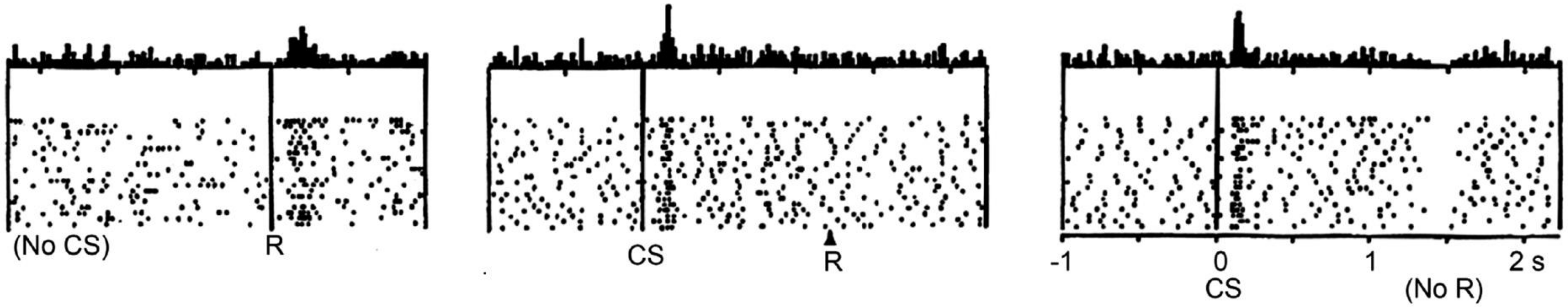
E) Bayesian Computations in spiking RNNs

F) Beyond: Reinforcement Learning

REINFORMENT LEARNING

Dopamine codes reward prediction errors

Dopamine codes reward prediction errors (RPEs)



Before training:
no predition, reward

After training:
prediction, reward

Reward omission:
prediction, no reward

Schultz et al., Science 1997

Then, dopamine provides a learning signal, the RPE

A few recent papers:

Song, H. F., Yang, G. R., & Wang, X. J. (2017). Reward-based training of recurrent neural networks for cognitive and value-based tasks. *Elife*, *6*, e21492.

Payeur, A., Guerguiev, J., Zenke, F., Richards, B. A., & Naud, R. (July, 2021).
Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits.
Nature neuroscience, 1-10.

Bono, J., Zannone, S., Pedrosa, V., & Clopath, C. (2021).
Learning predictive cognitive maps with spiking neurons during behaviour and replays.
bioRxiv.

Collaborators

Luis Serrano-Fernández (UAM, Madrid)

Manuel Beirán (ENS, Paris)

Pablo Crespo Darriba (URJC, Madrid)

Martín Zamarbide (UAM, Madrid)

Stefania Sarno (CENTAURI, Marseille) France

Joan Falcó-Roget (UAM, Madrid)

R. Romo (UNAM, Mexico)

Román Rossi-Pool (UNAM, Mexico)