Processing information stored in working memory by modulating effective connectivity

Adrian Valente, Alexis Dubreuil, Srdjan Ostojic

LNC, Ecole Normale Supérieure, PSL Research University, Paris, France

Introduction

Comparison between a stimulus and an item stored in memory requires flexible neural responses. How can we implement them with recurrent connectivity?

We use low-rank RNNs [1] as a tool to "open the black box" of trained RNNs

Approach:

Train low-rank RNNs → Reverse-engineer trained RNNs → Construct a class of solutions

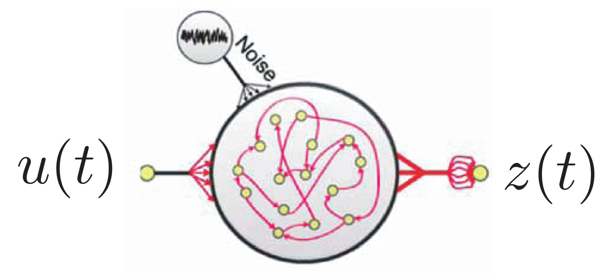
Methods

RNN equations:

$$\tau \frac{d\vec{x}}{dt} = -\vec{x} + W_{rec}\vec{\Phi}(\vec{x}) + u(t)\vec{W}_{in}$$

output: $z(t) = \vec{W}_{out}^T \vec{\Phi}(\vec{x}(t))$

with $\Phi_k(\vec{x}) = tanh(x_k)$



Low-rank connectivity [1] :

$$W_{rec} = \frac{1}{N}\sum_{k=1}^{K} \vec{m}_k \vec{n}_k^T$$

-> K-dimensional dynamics

$$[x_i] = \kappa_1 m_1^1 + \dots + \kappa_K m_1^K$$

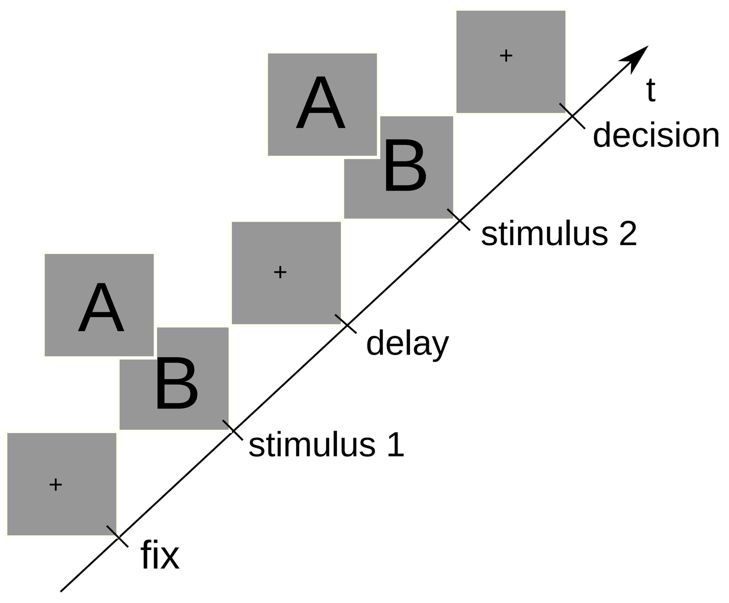
$$\dot{\kappa}_i = F(\kappa_1, \dots, \kappa_K)$$

Training:

BPTT on the $\vec{m}_k$ and $\vec{n}_k$

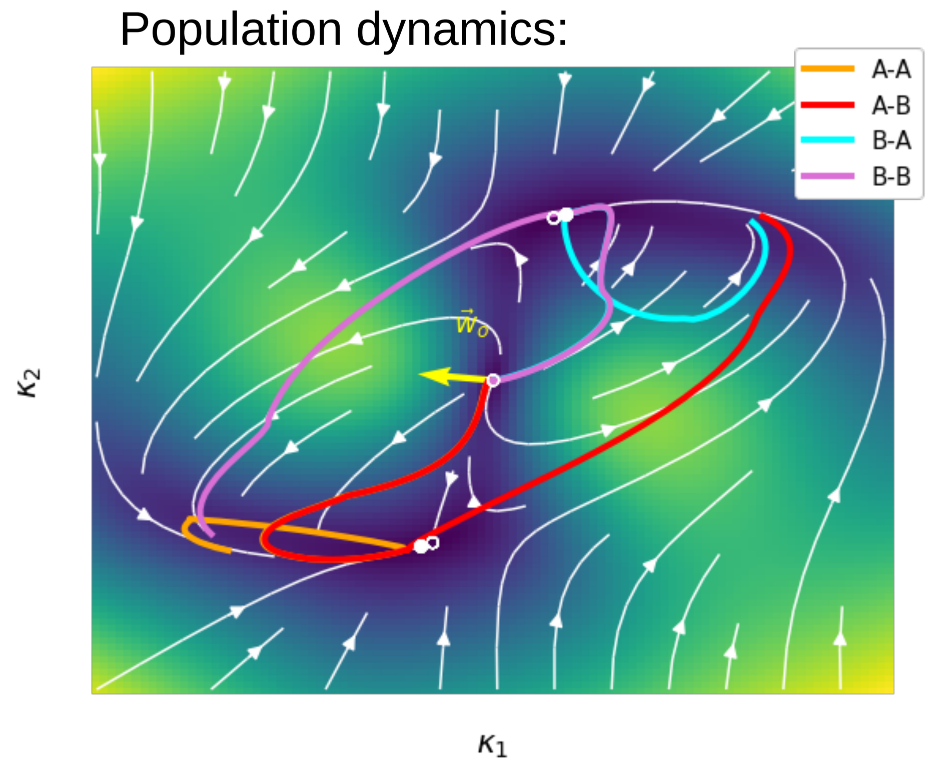
DMS implementation

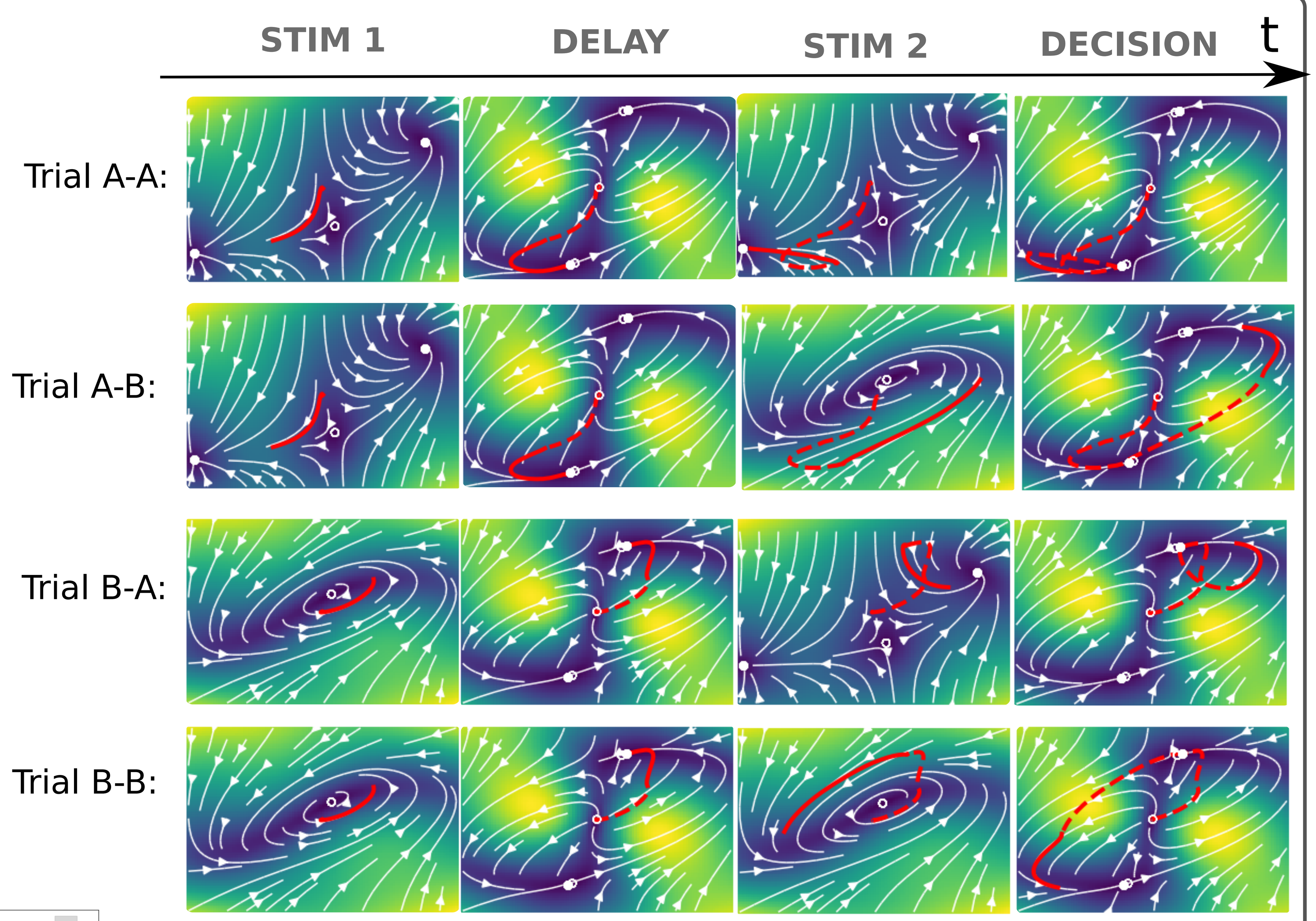
DMS Task



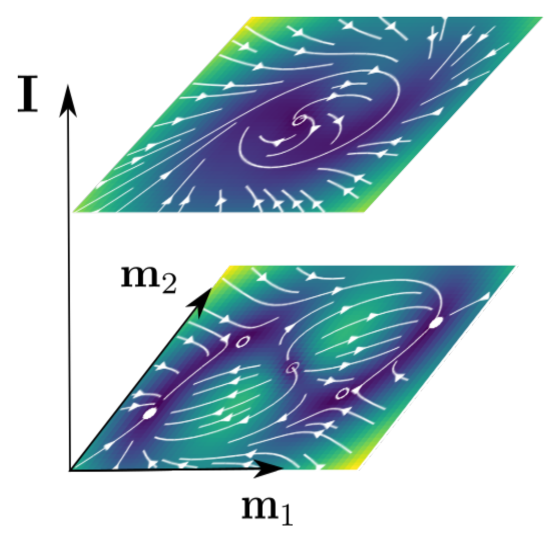
working memory + **XOR**

Population dynamics:

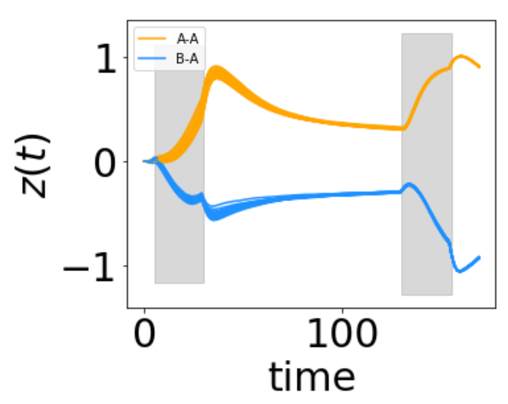




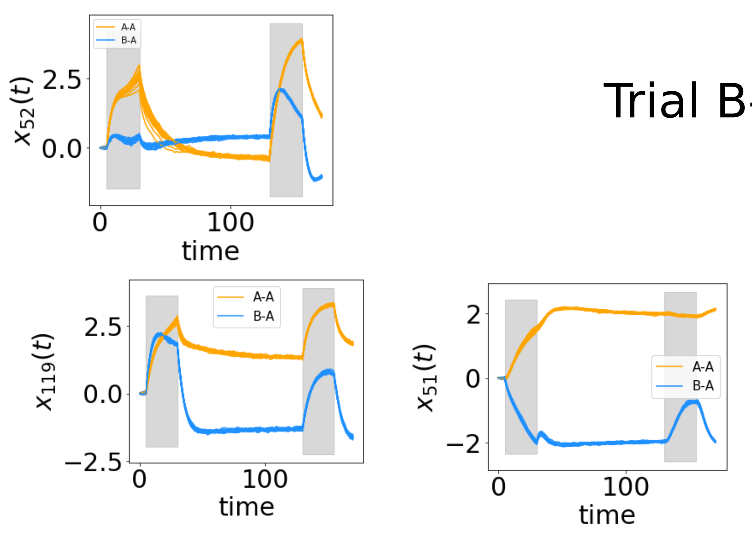
Inputs shift dynamics:



Readout:



Neuron traces:



-> 2 fixed points for memory, reused for decision

-> XOR implemented by an input-induced limit cycle

Mechanism: control of dynamical landscape via effective connectivity

2D equivalent system:

$$\dot{\kappa}_1 = -\kappa_1 + \tilde{\sigma}_{n_1 m_1}\kappa_1 + \tilde{\sigma}_{n_1 m_2}\kappa_2 + \tilde{\sigma}_{n_1 W_{in}} u(t)$$
$$\dot{\kappa}_2 = -\kappa_2 + \tilde{\sigma}_{n_2 m_1}\kappa_1 + \tilde{\sigma}_{n_2 m_2}\kappa_2 + \tilde{\sigma}_{n_2 W_{in}} u(t)$$

Functional overlaps with 2 populations:

$$\tilde{\sigma}_{ab} = \sigma_{ab}^{(1)} \langle \phi' \rangle_1 + \sigma_{ab}^{(2)} \langle \phi' \rangle_2$$

Jacobian of 2D system:

$$F'(\mathbf{I}) = \begin{pmatrix} \tilde{\sigma}_{11} - 1 & \tilde{\sigma}_{12} \\ \tilde{\sigma}_{21} & \tilde{\sigma}_{22} - 1 \end{pmatrix}$$

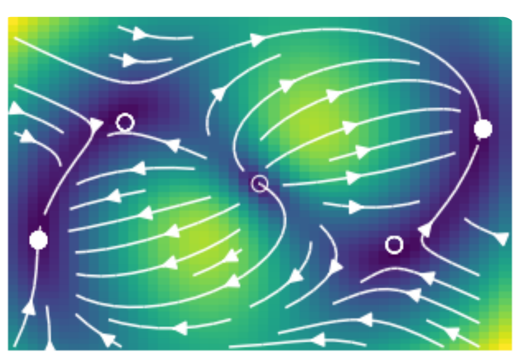
Reconstructed solution:

Modulation of $\tilde{\sigma}_{21}$

$\sigma_{21}^{(1)} = 1$ input A saturates pop 2

$\sigma_{21}^{(2)} = -1$ input B saturates pop 1

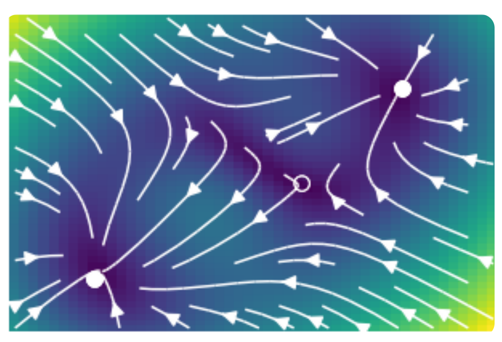
No input:



$$F'(0) = \begin{pmatrix} 3 & 1 \\ 0 & 2.5 \end{pmatrix}$$

$$\tilde{\sigma}_{21} = 1 \times 1 + (-1) \times 1$$

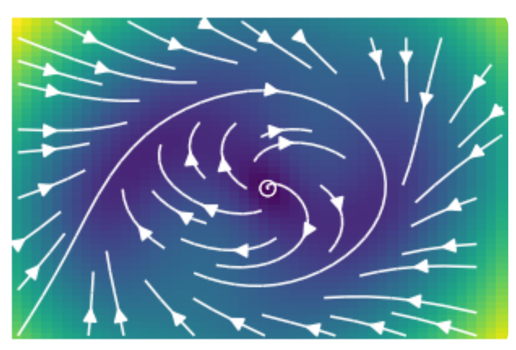
Input A:



$$F'(\mathbf{I_A}) = \begin{pmatrix} 1.1 & 0.3 \\ 0.3 & 0.8 \end{pmatrix}$$

$$\tilde{\sigma}_{21} = 1 \times 0.3 + (-1) \times 0$$

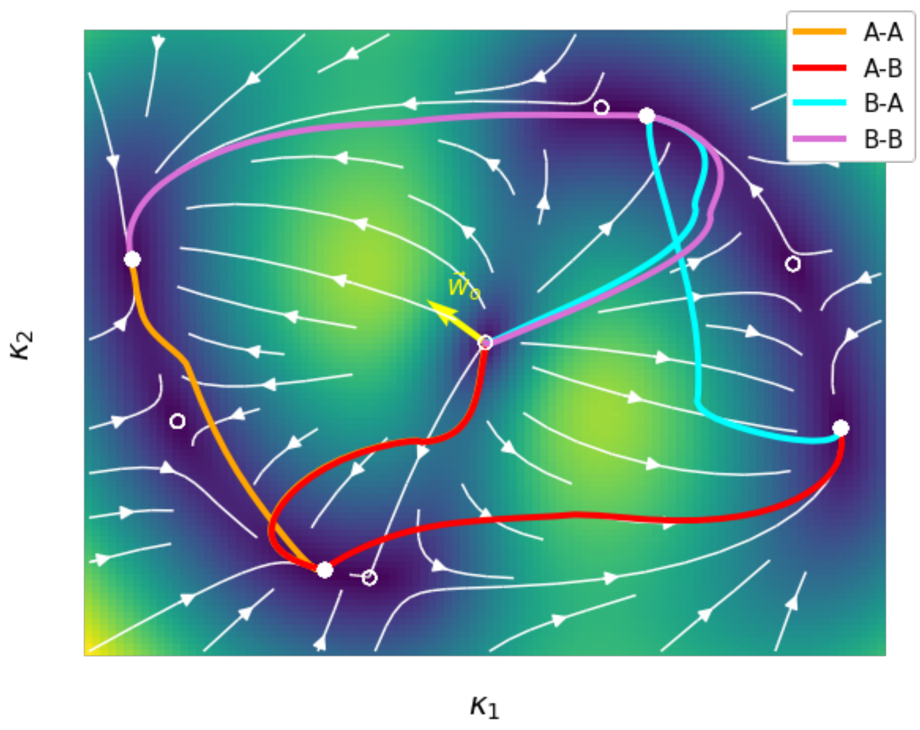
Input B:



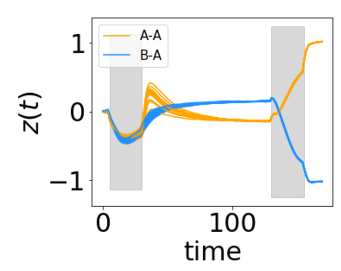
$$F'(\mathbf{I_B}) = \begin{pmatrix} 1.1 & 0.3 \\ -0.3 & 0.8 \end{pmatrix}$$

$$\tilde{\sigma}_{21} = 1 \times 0 + (-1) \times 0.3$$

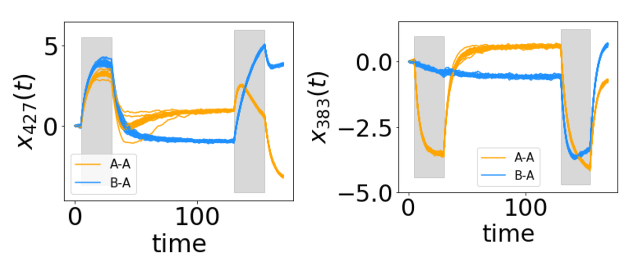
Training gives rise to different solutions



readout:



neuron traces:



Summary

- low-rank RNNs offer an interpretable and analytically tractable approximation to full-rank RNNs

- We have found a mechanism for flexibly transforming the dynamics of a neural circuit when an input is received

- This mechanism points to the complementary roles of rank and cell populations for implementing computations

Reference

[1] F. Mastrogiuseppe, S. Ostojic. Linking connectivity, dynamics and computations in low-rank recurrent neural networks. Neuron. 2018