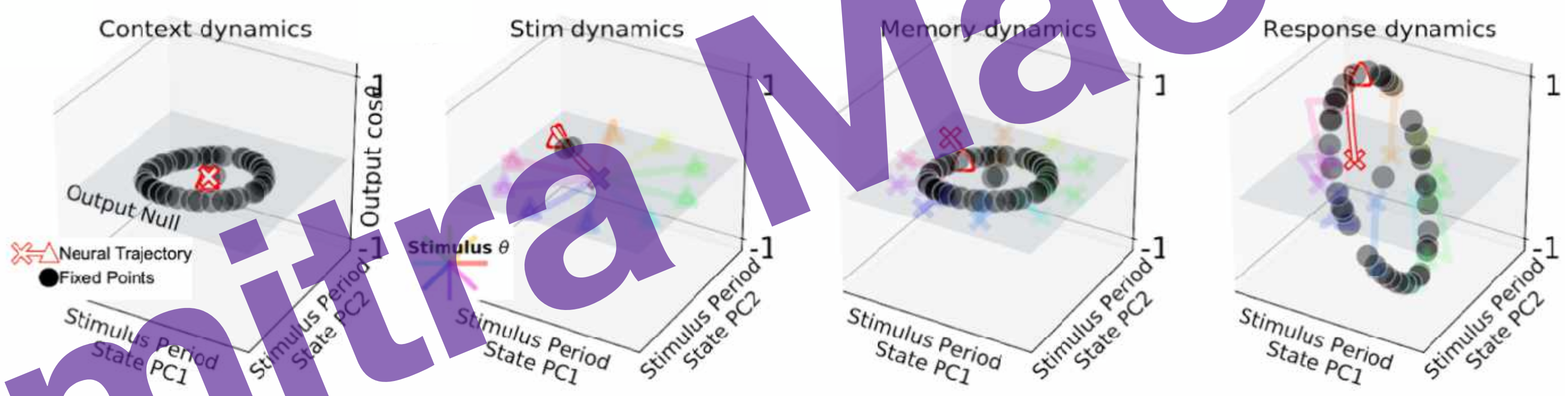
Influence of input structure and task similarity on continual learning

Dimitra Maoutsa - MCN summer course

How sequential task training affects the RNN dynamical landscape?

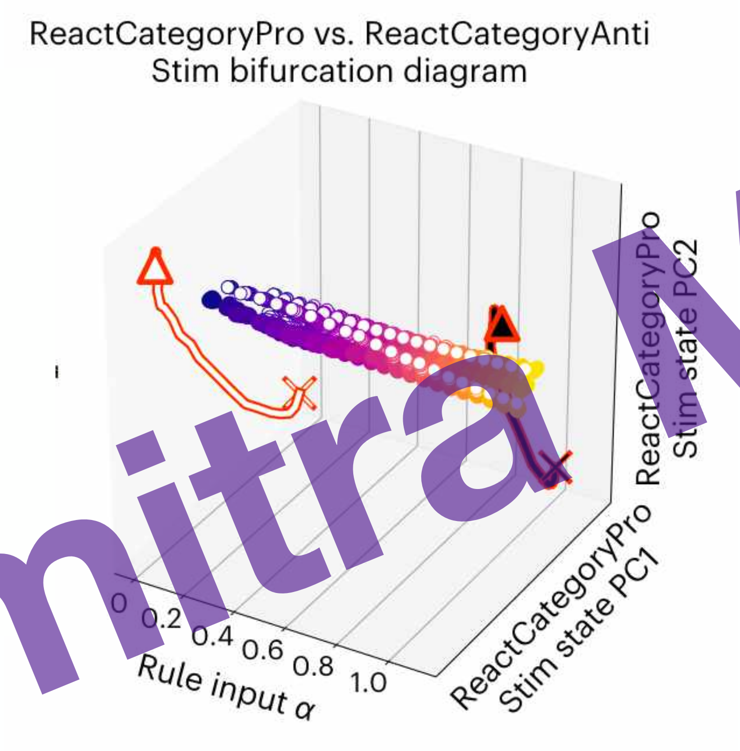
RNNs as tools to understand computations

Dynamics as computations



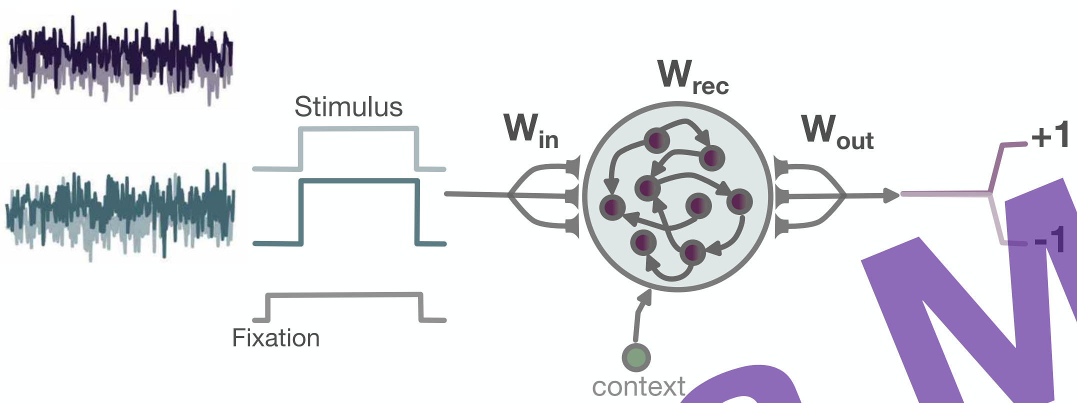
Driscoll et al. 2022

Individual brain regions are able to support computations required for multiple tasks
How?



networks trained on multiple tasks **simultaneously** tend to share same dynamical motifs if required computations are similar

Classical single-task RNN structure



Vanilla RNN

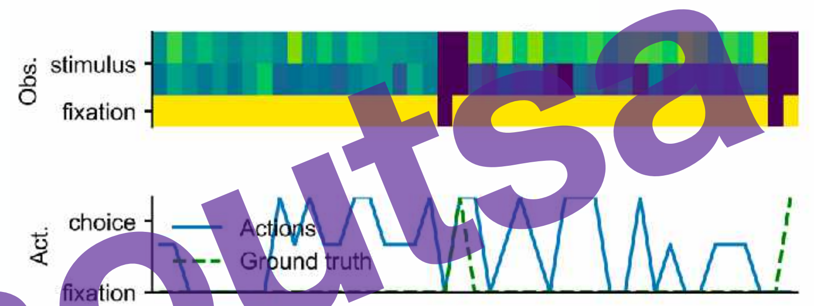
$$\frac{d\mathbf{r}_t}{dt} = -\mathbf{r}_t + \phi\left(\mathbf{W}_{\text{rec}}\mathbf{r}_t + \mathbf{W}_{\text{in}}\mathbf{I}_t^{\text{in}}\right)$$

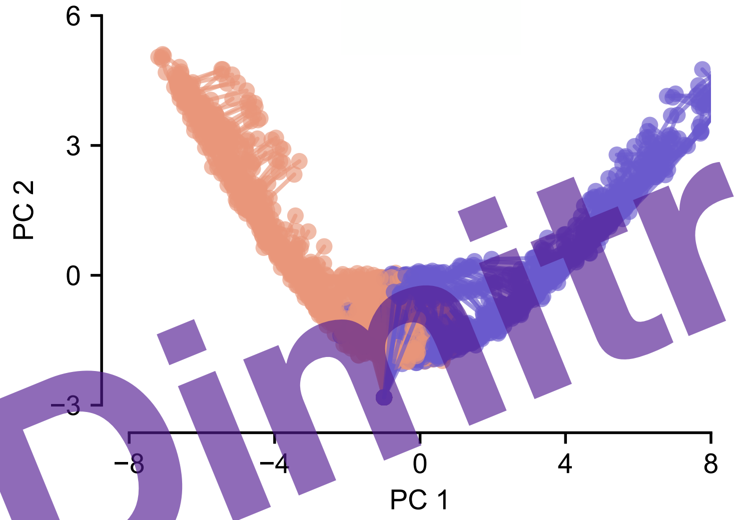
$$\mathbf{o}_t = \mathbf{W}_{out} \cdot \mathbf{r}_t + \mathbf{b}$$

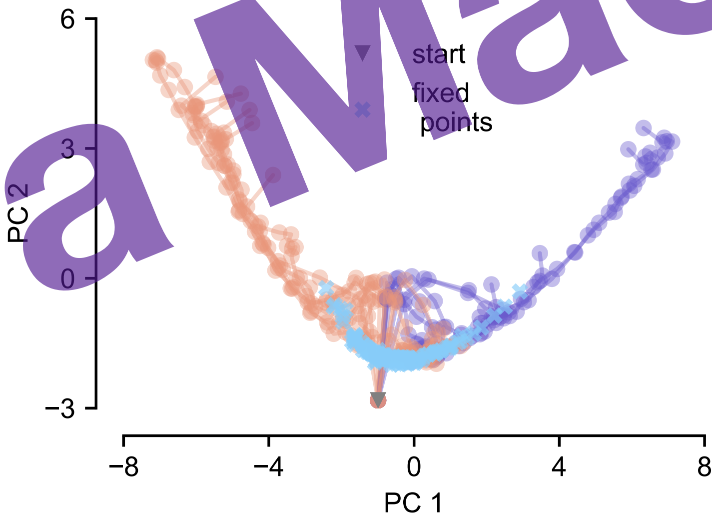
RNNs trained on single tasks

Perceptual decision making task

Single-task trained RNN







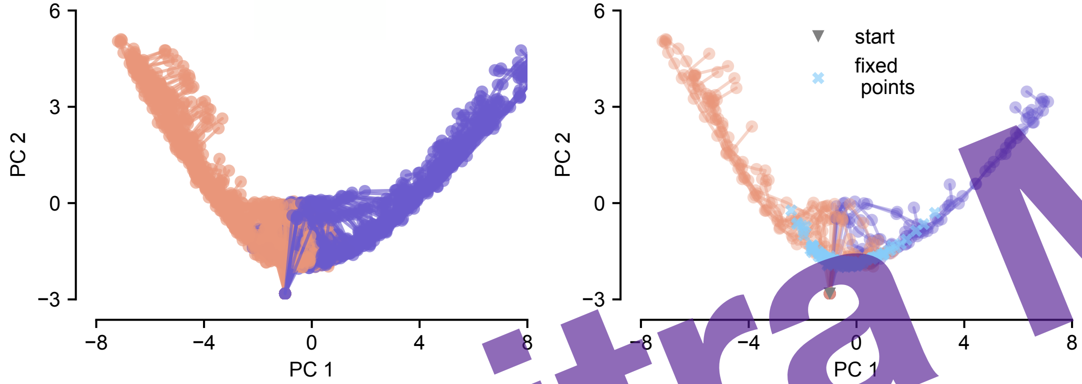
Define energy function

$$q(\mathbf{x}) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|^2$$

q is zero only at fixed points

Perceptual decision making task

Single-task trained RNN



Define energy function

$$q(\mathbf{x}) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|^2$$

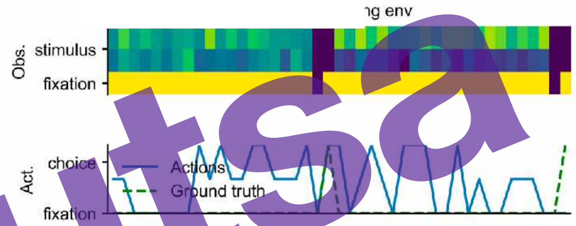
q is zero only at fixed points

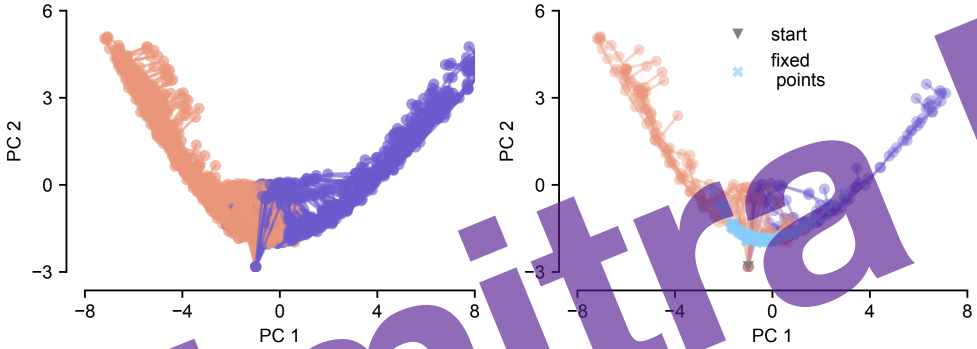
$$\frac{\partial q}{\partial x_i} = \sum_{k=1}^{N} \frac{\partial F_k}{\partial x_i}\dot{x}_k \quad = 0$$

$$\frac{\partial^2 q}{\partial x_i \partial x_j} = \sum_{k=1}^{N} \frac{\partial F_k}{\partial x_i}\frac{\partial F_k}{\partial x_j} + \sum_{k=1}^{N} \dot{x}_k \frac{\partial^2 F_k}{\partial x_i \partial x_j} \quad > 0$$

Perceptual decision making task

Single-task trained RNN





Define energy function

$$q(\mathbf{x}) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|^2$$

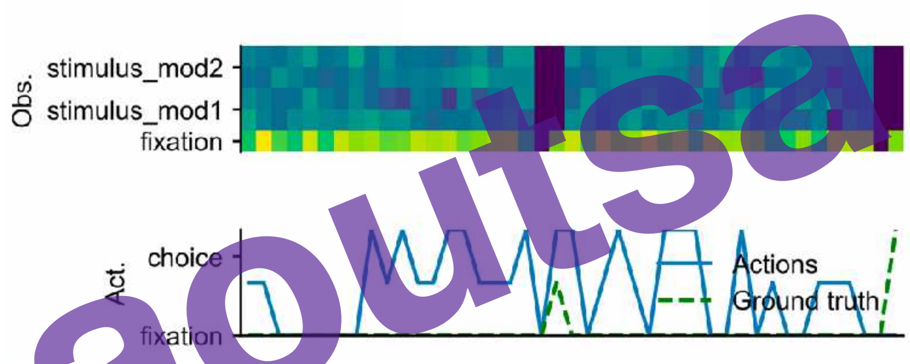
q is zero only at fixed points

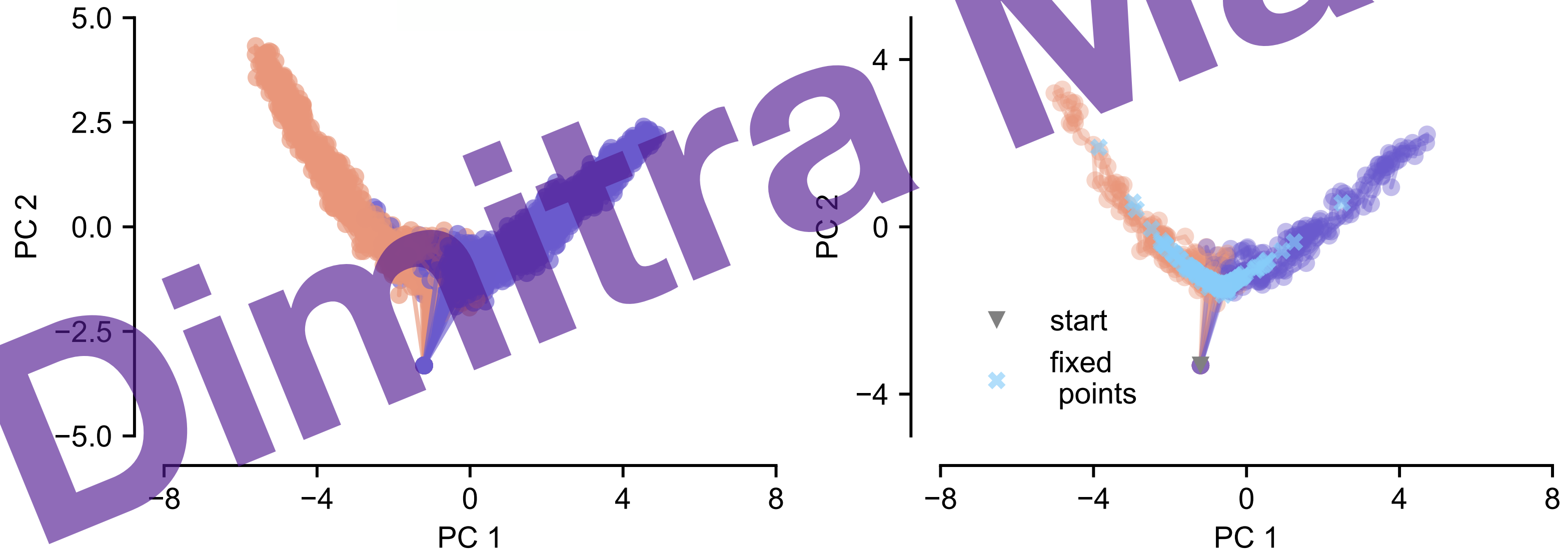
$$\frac{\partial q}{\partial x_i} = \sum_{k=1}^{N} \frac{\partial F_k}{\partial x_i}\dot{x}_k \quad = 0$$

$$\frac{\partial^2 q}{\partial x_i \partial x_j} = \sum_{k=1}^{N} \frac{\partial F_k}{\partial x_i}\frac{\partial F_k}{\partial x_j} + \sum_{k=1}^{N} \dot{x}_k \frac{\partial^2 F_k}{\partial x_i \partial x_j} \quad > 0$$

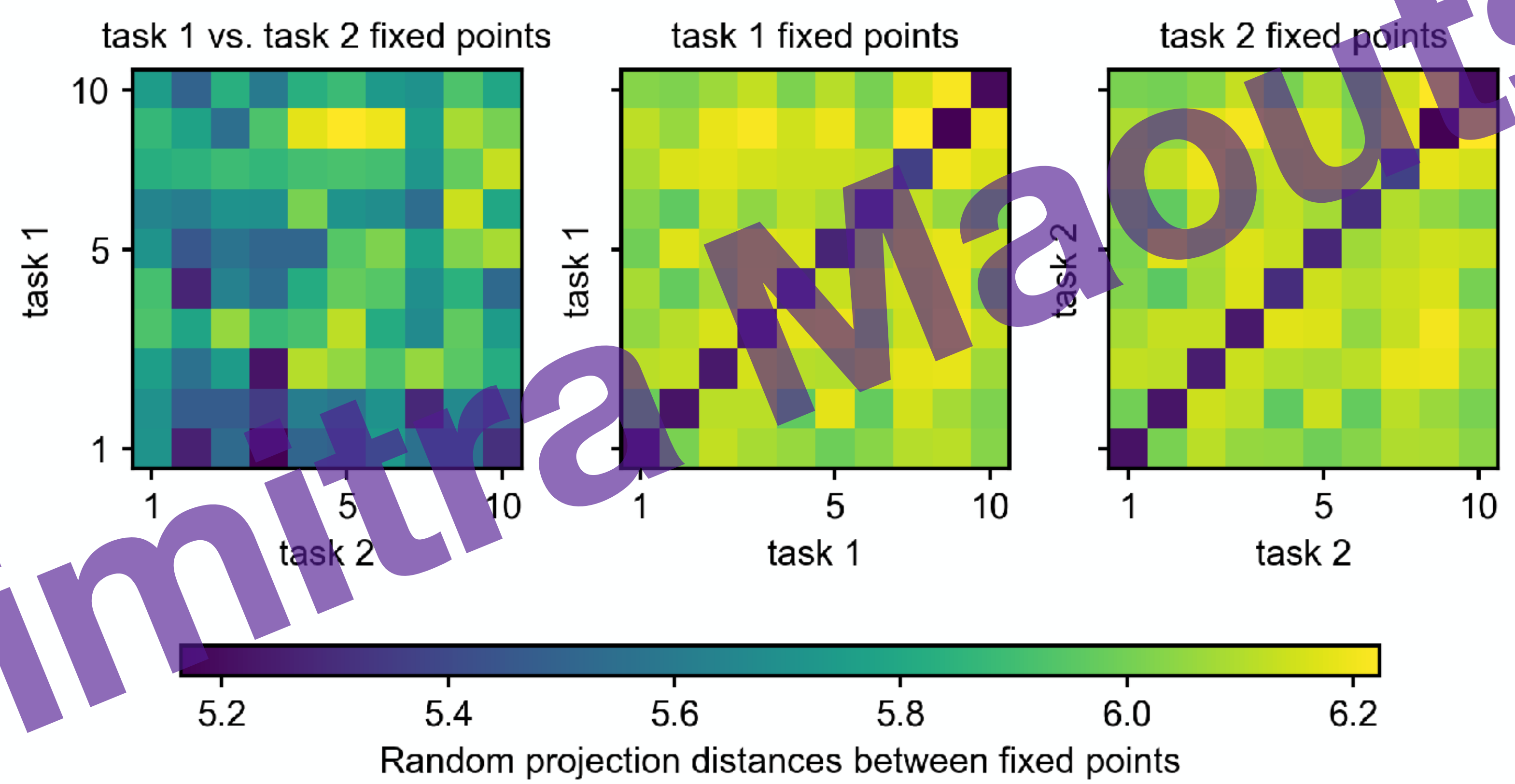
Multisensory integration task

Single-task trained RNN

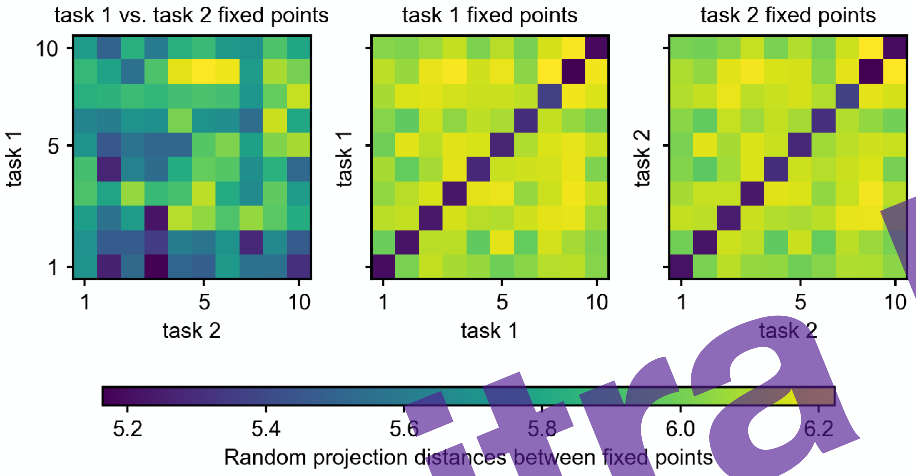


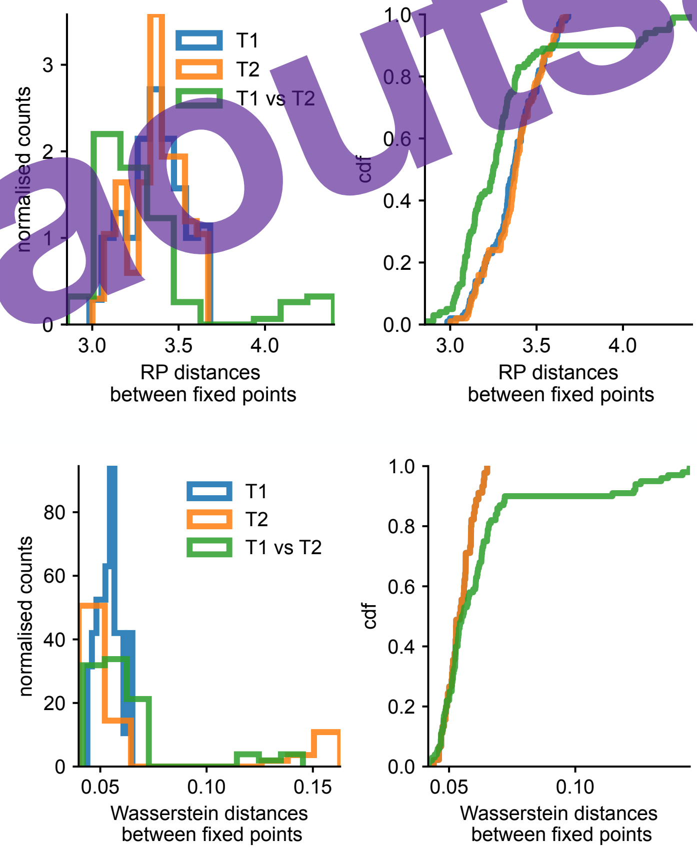


Distances between fixed points of the two tasks



Distances between fixed points of the two tasks





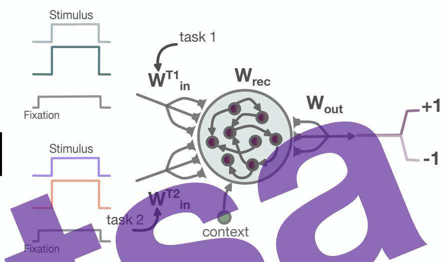
Procrustes distance
in the future

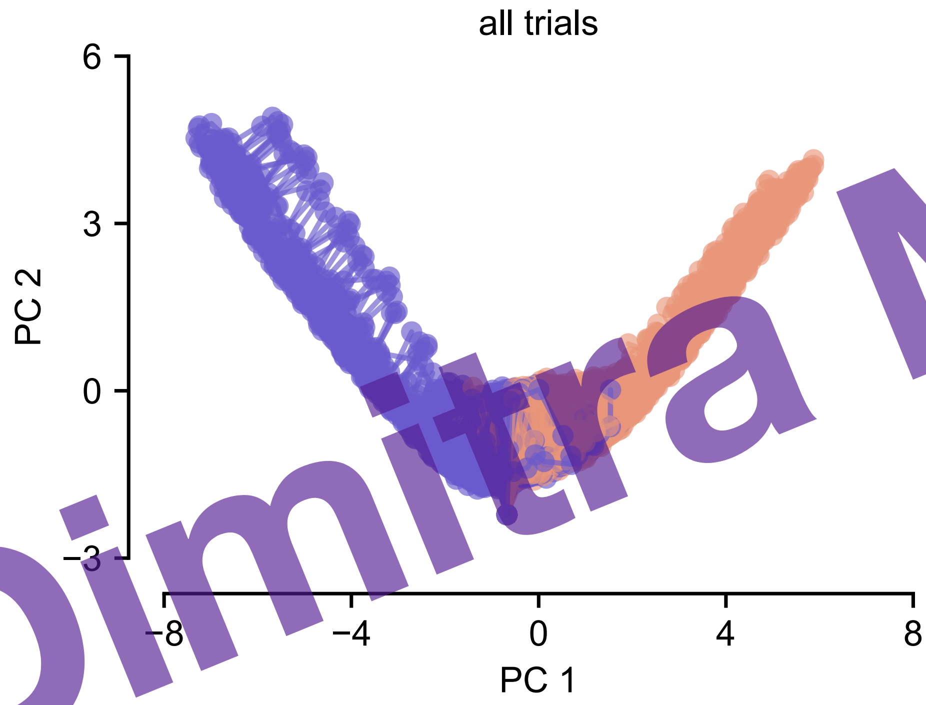
Wasserstein distance
yields more sensible
distance estimates

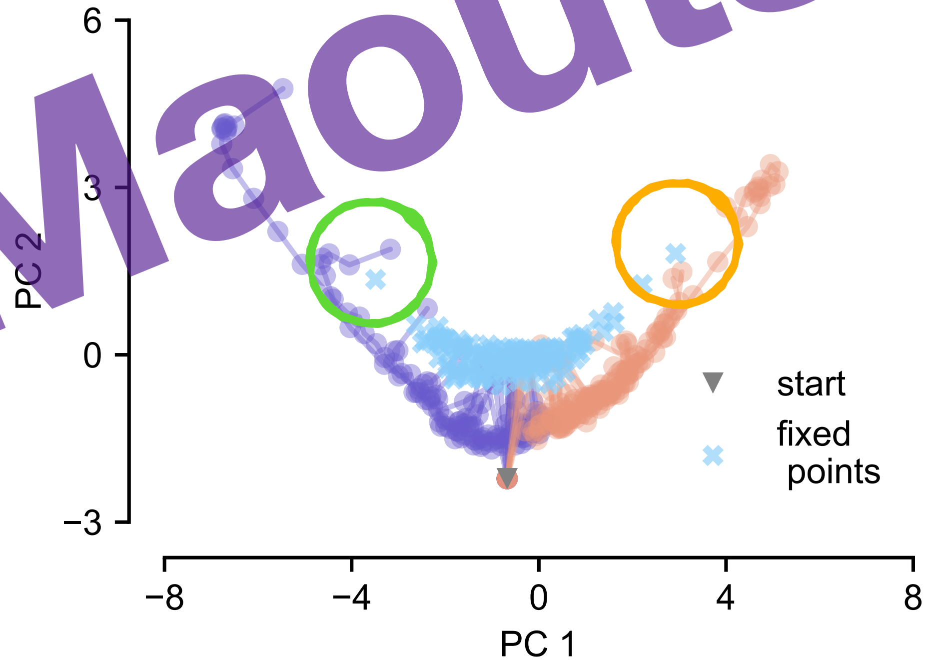
RNNs trained sequentially on two tasks

Multi-task trained RNN

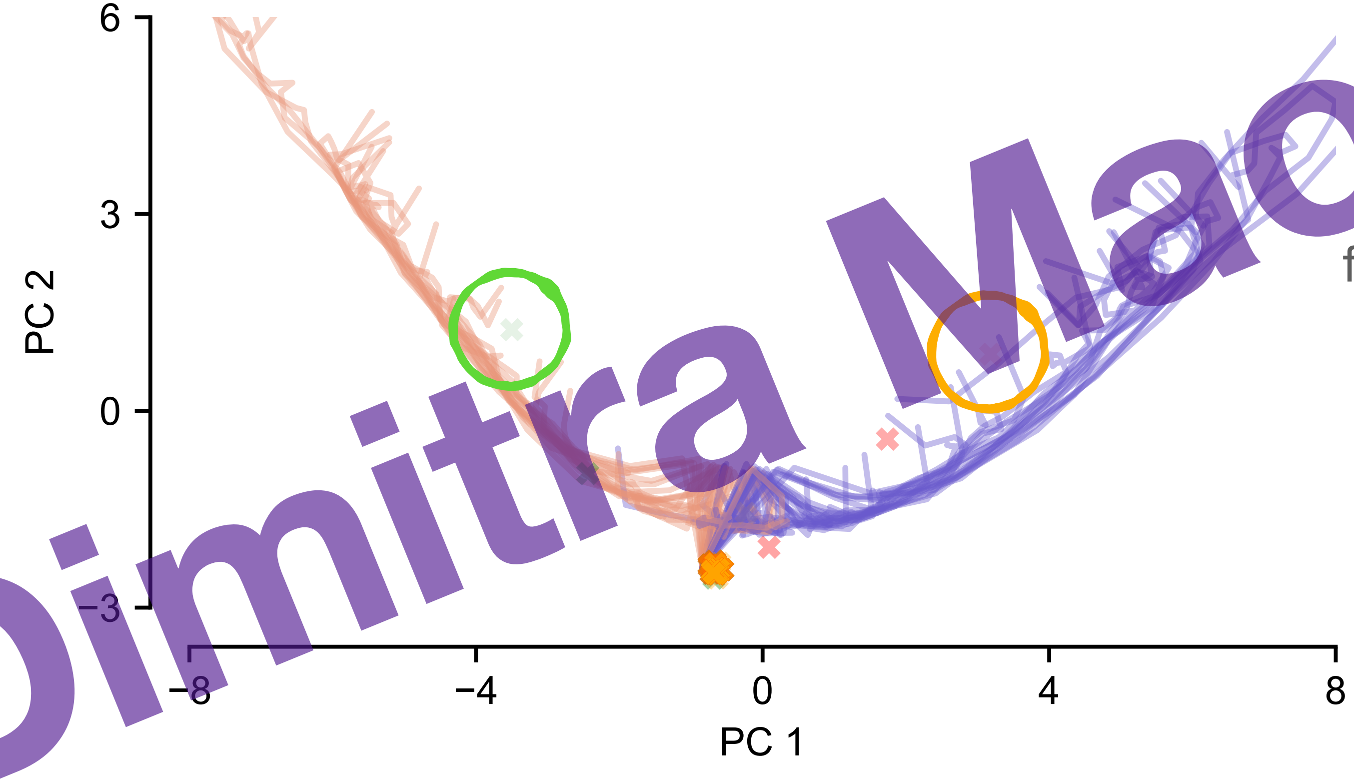
Representations and fixed points after training task 1

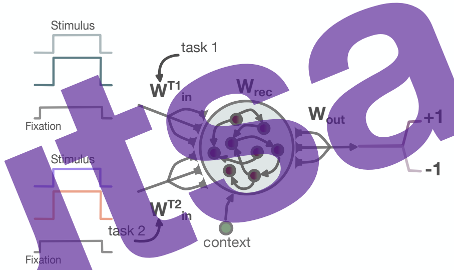






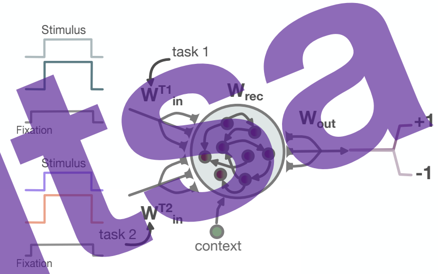
Evolution of fixed points of task 1 during training of task 2

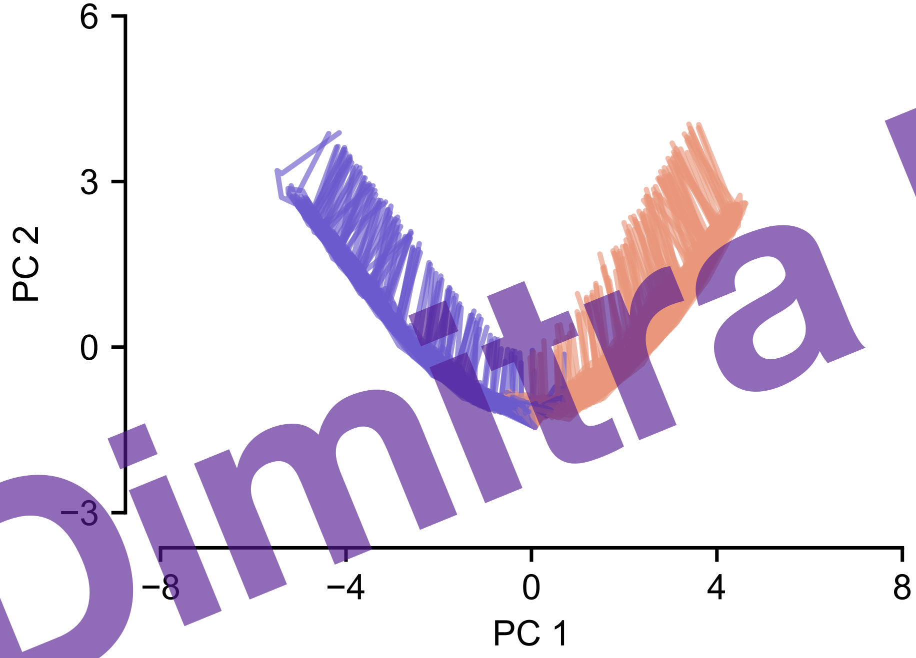


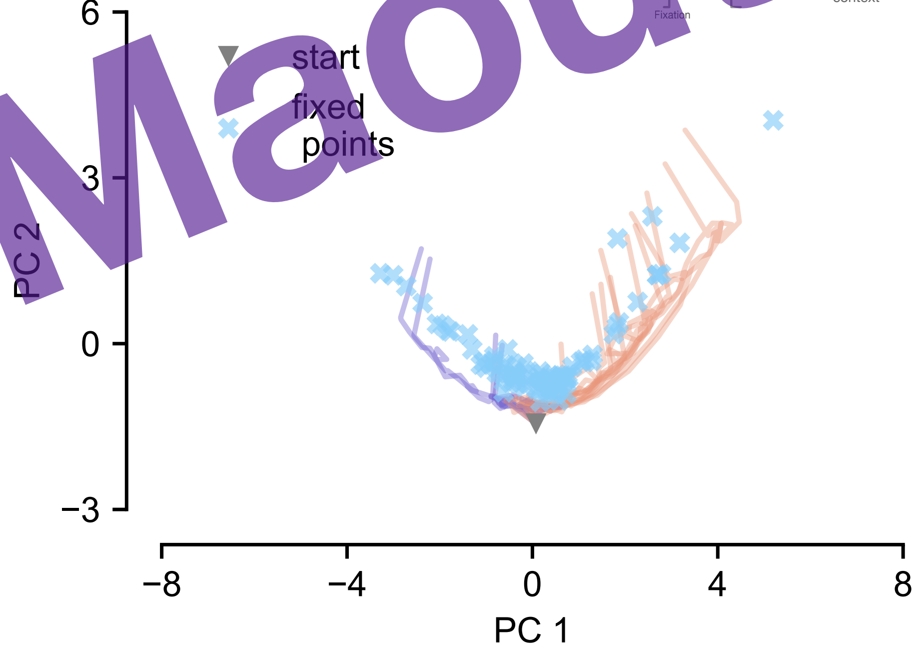


fast collapse of fixed points for task 1 during training of task 2

Low-dimensional representation of task 2

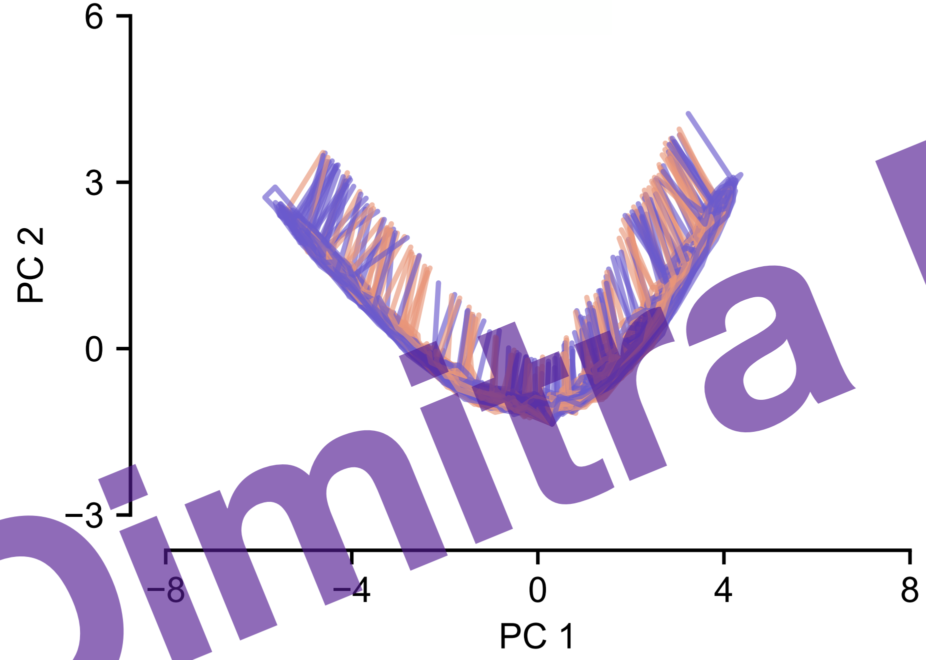


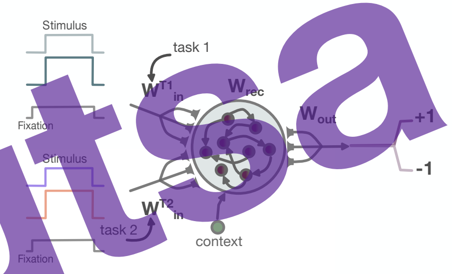


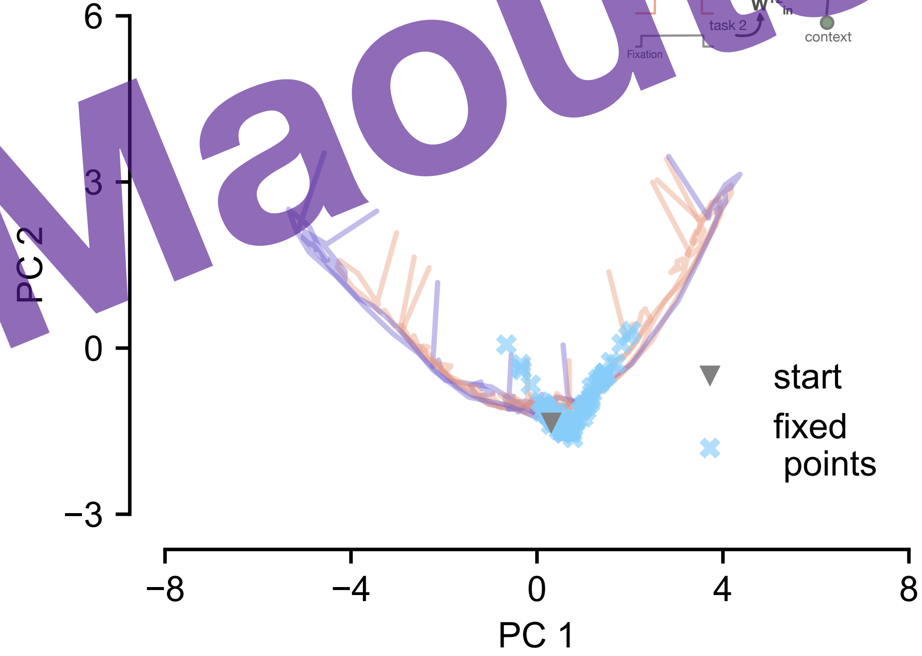


Performance of task 1 after task 2 drops

trajectories of task 1 get mixed for both stimulus conditions



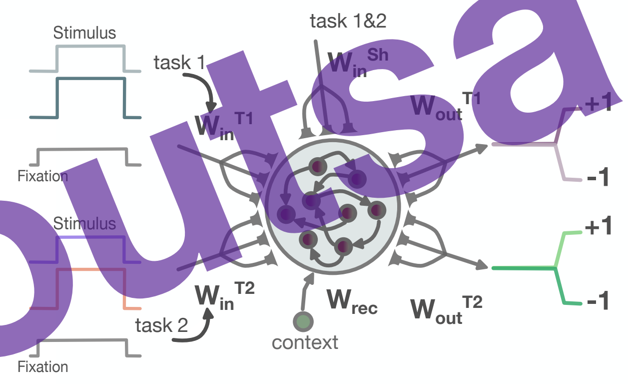


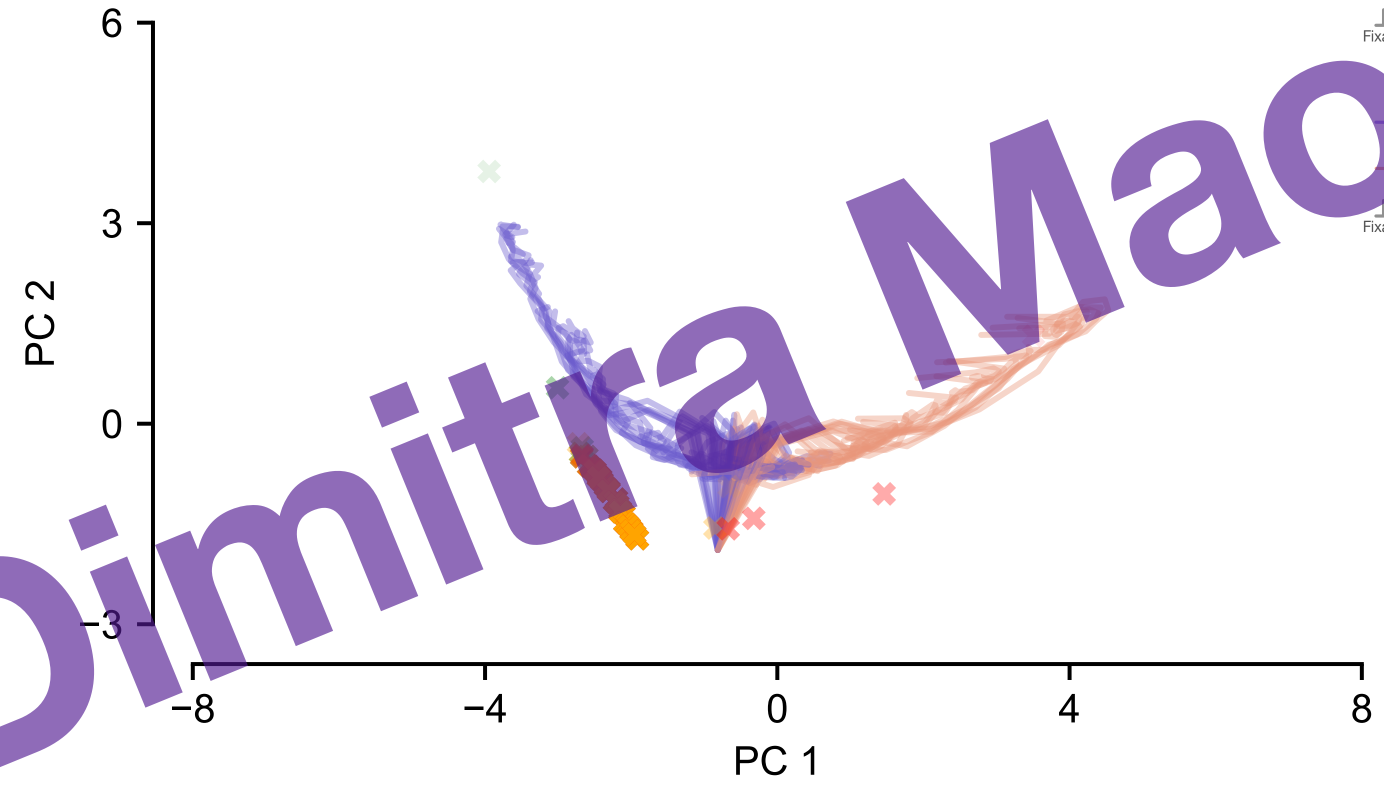


RNNs trained sequentially on two tasks with mixed selective inputs pathways

Mixed selective pathway slows down FP forgetting

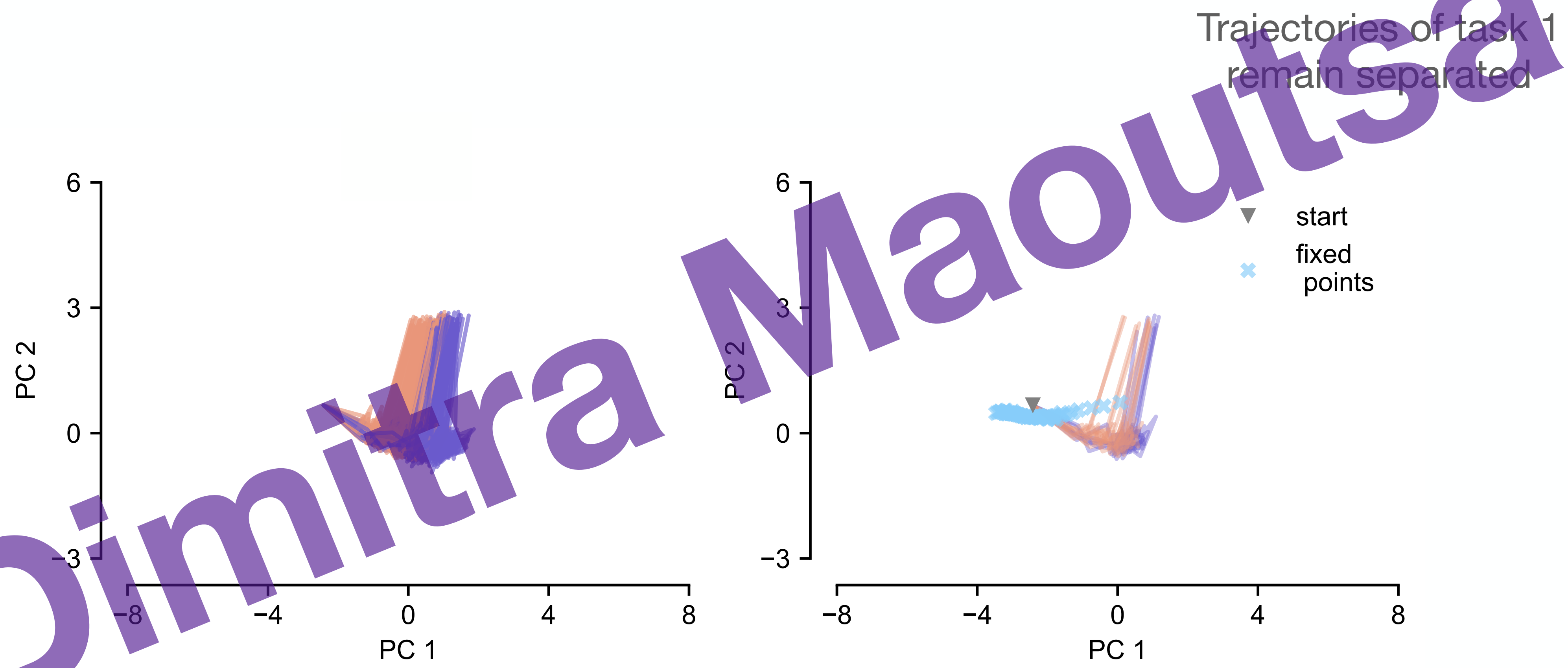
α=0.5



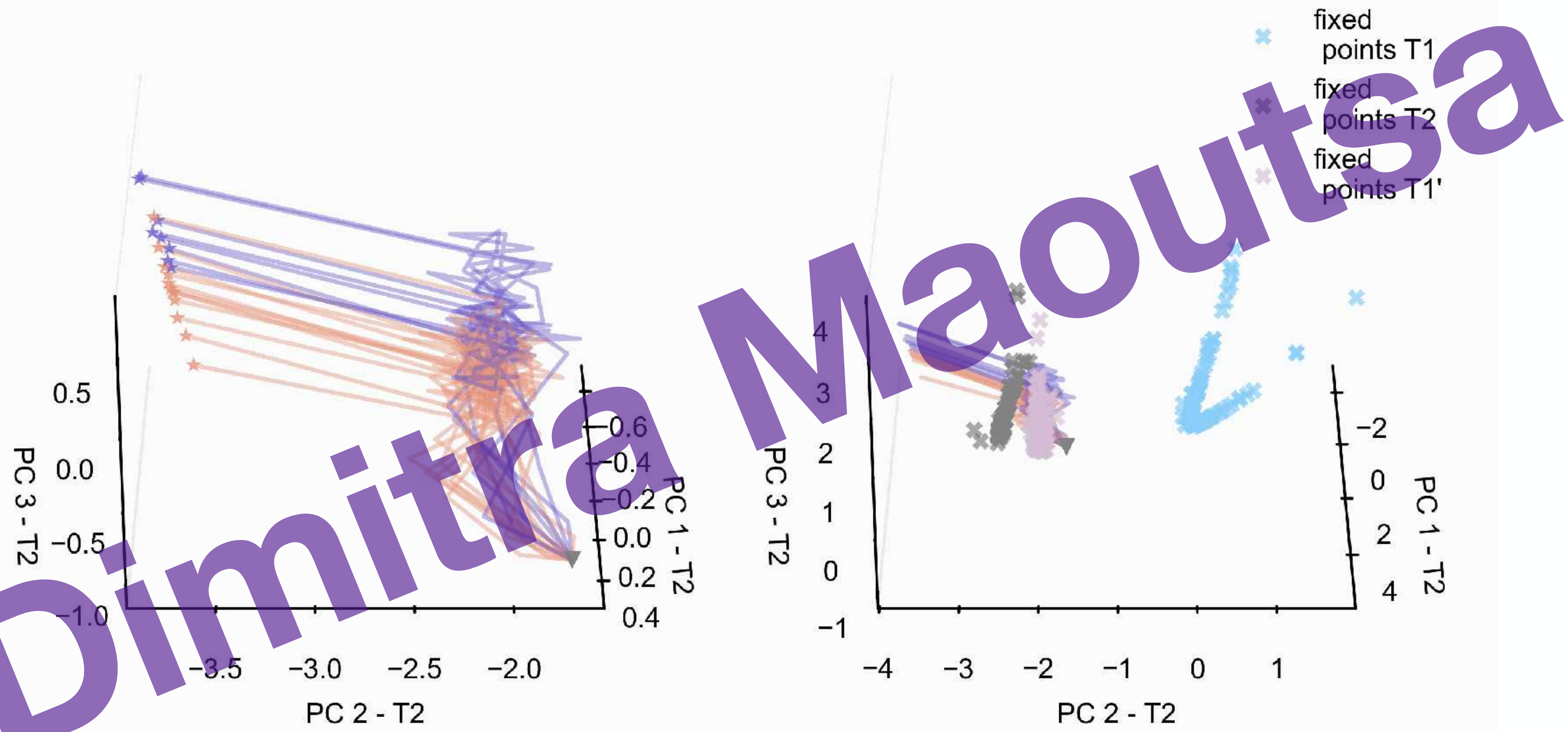


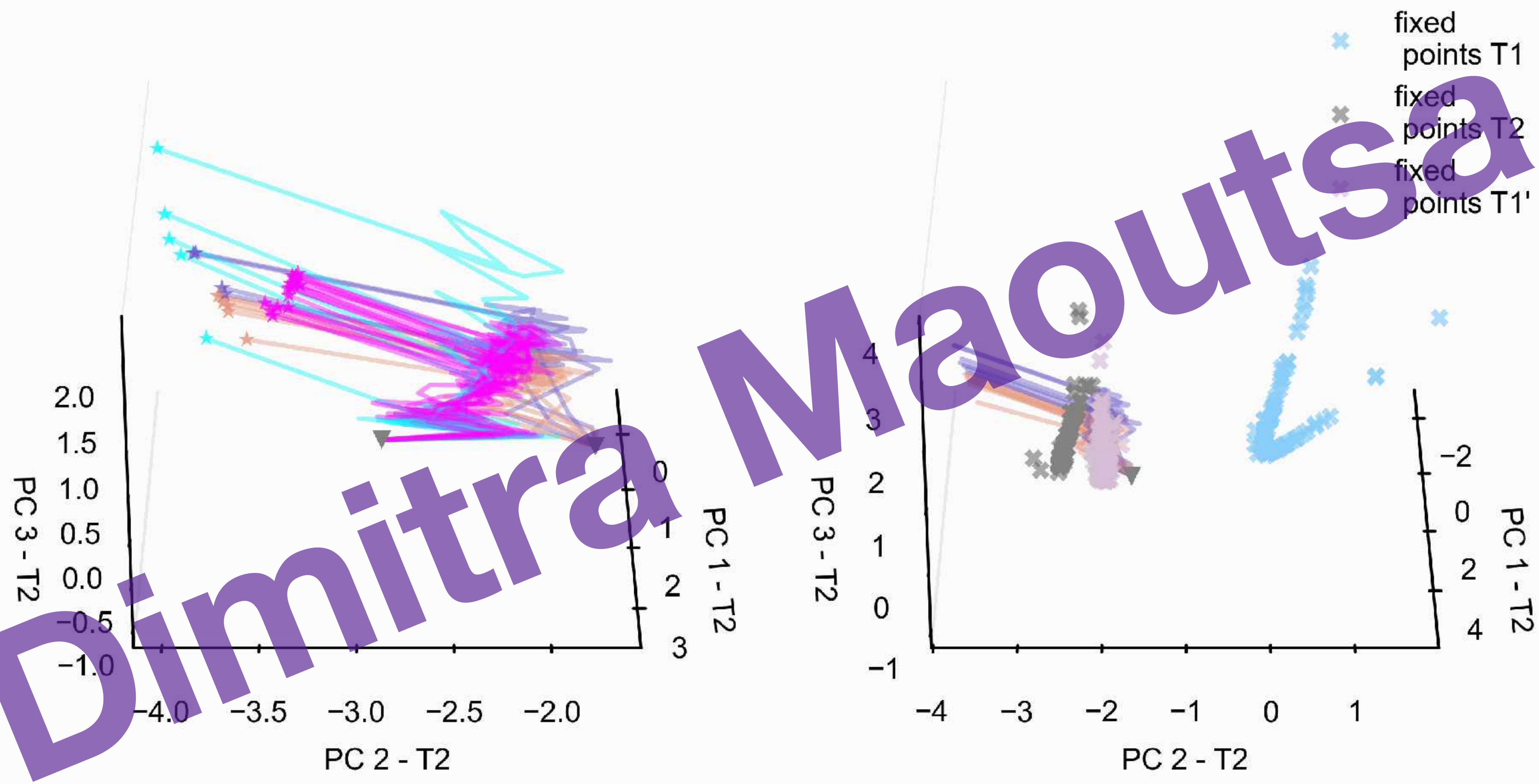
fixed points for task 1 during training of task 2

Representations of task 1 after training task 2



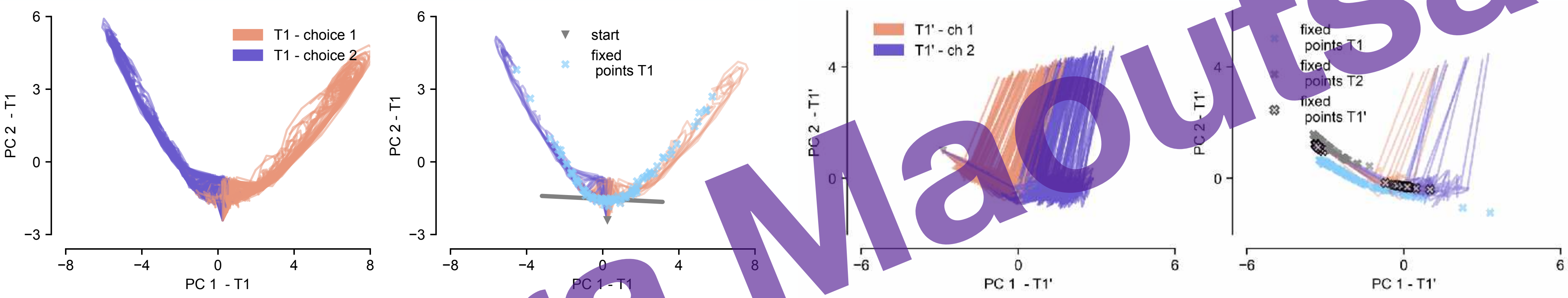
α = 0.5


PC 3 - T2
0.5
0.0
−0.5
−1.0
−3.5
−3.0
−2.5
−2.0
PC 2 - T2
−0.6
−0.4
−0.2
0.0
0.2
0.4
PC 1 - T2
fixed points T1
fixed points T2
fixed points T1'
PC 3 - T2
4
3
2
1
0
−1
−4
−3
−2
−1
0
1
PC 2 - T2
−2
0
2
4
PC 1 - T2

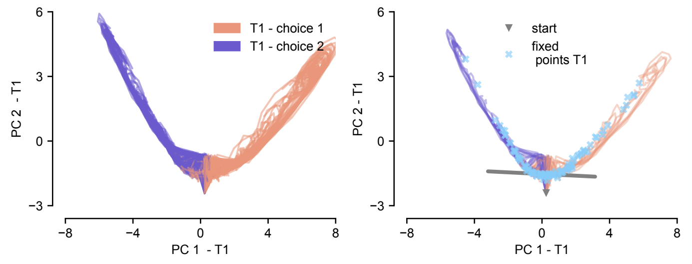


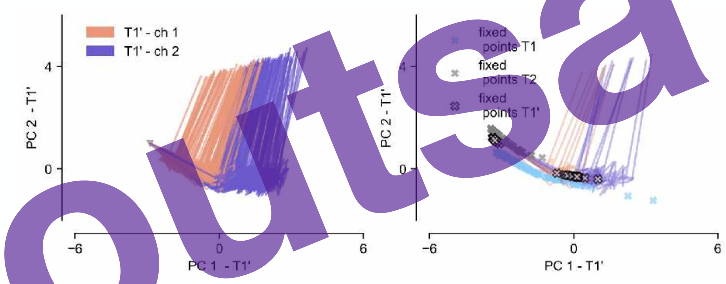
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
PC 3 - T2
−4.0
−3.5
−3.0
−2.5
−2.0
PC 2 - T2
0
1
2
3
PC 1 - T2
fixed points T1
fixed points T2
fixed points T1'
4
3
2
1
0
−1
PC 3 - T2
−4
−3
−2
−1
0
1
PC 2 - T2
−2
0
2
4
PC 1 - T2

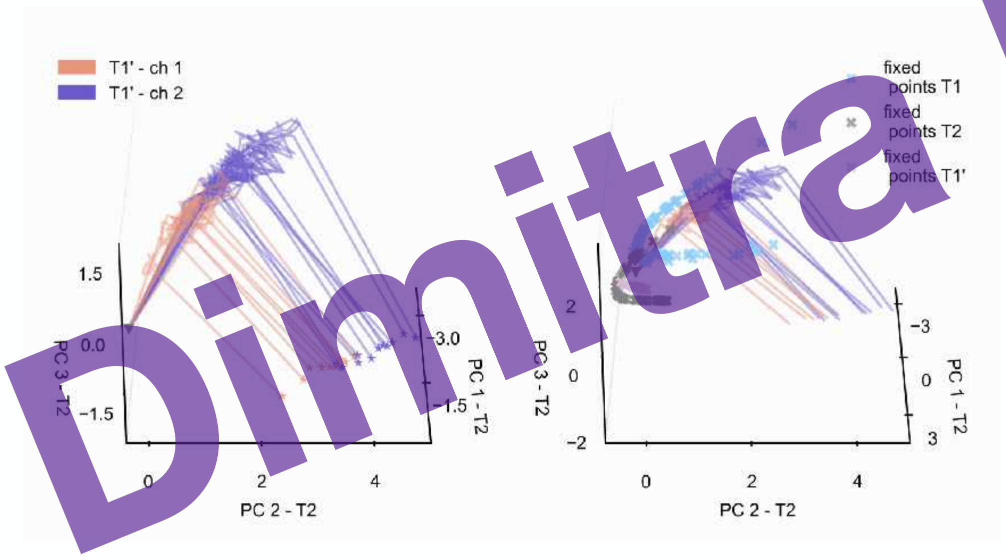
Representation of task 1 for α=0

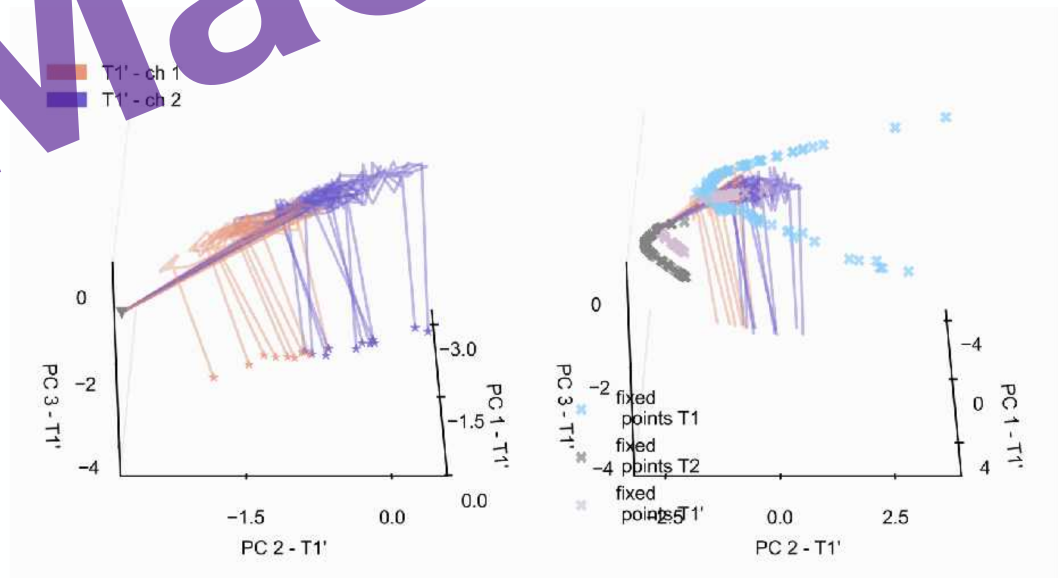


Representation of task 1 for a=0



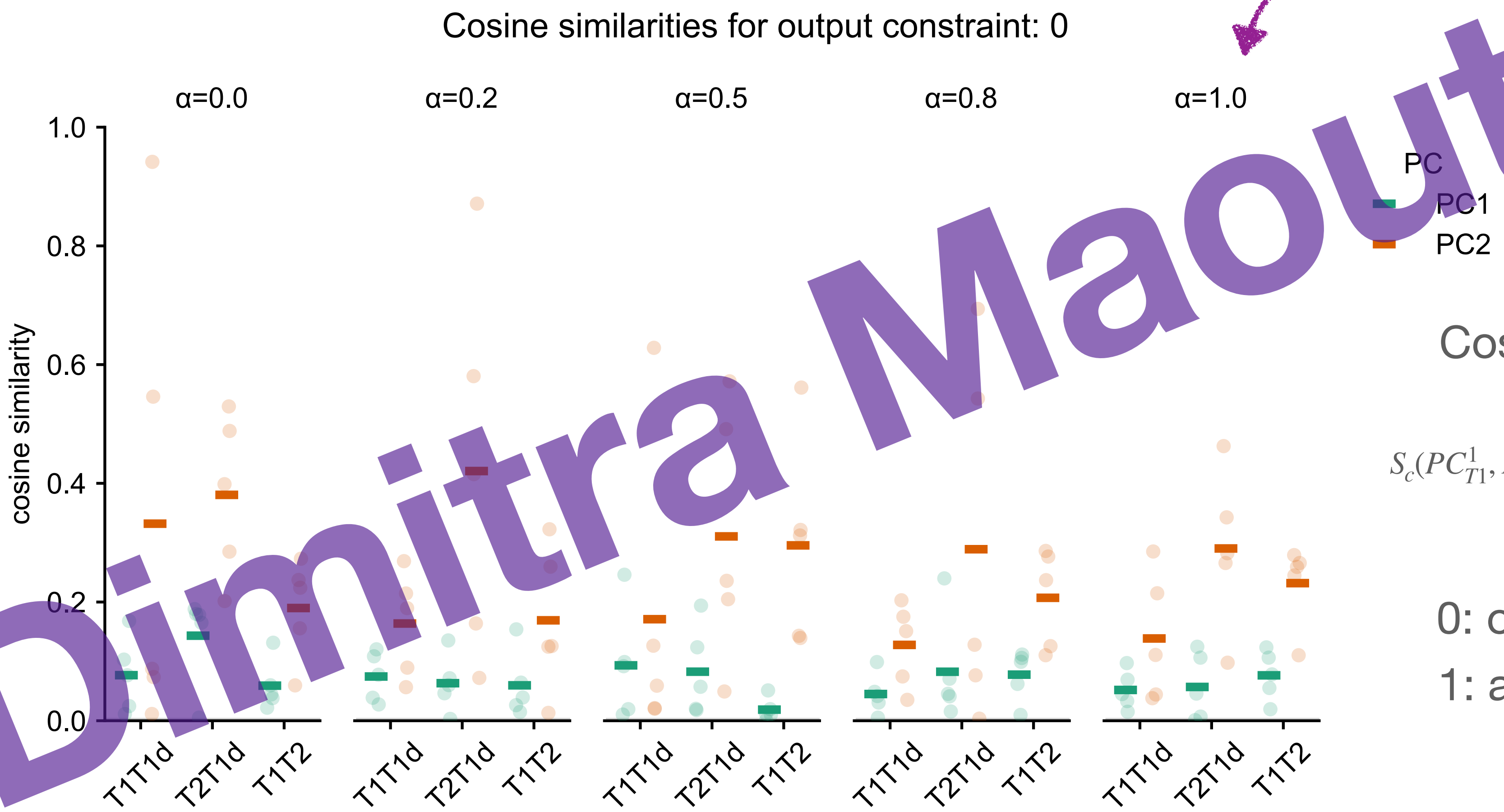






Comparing PC subspaces

all input
is separate



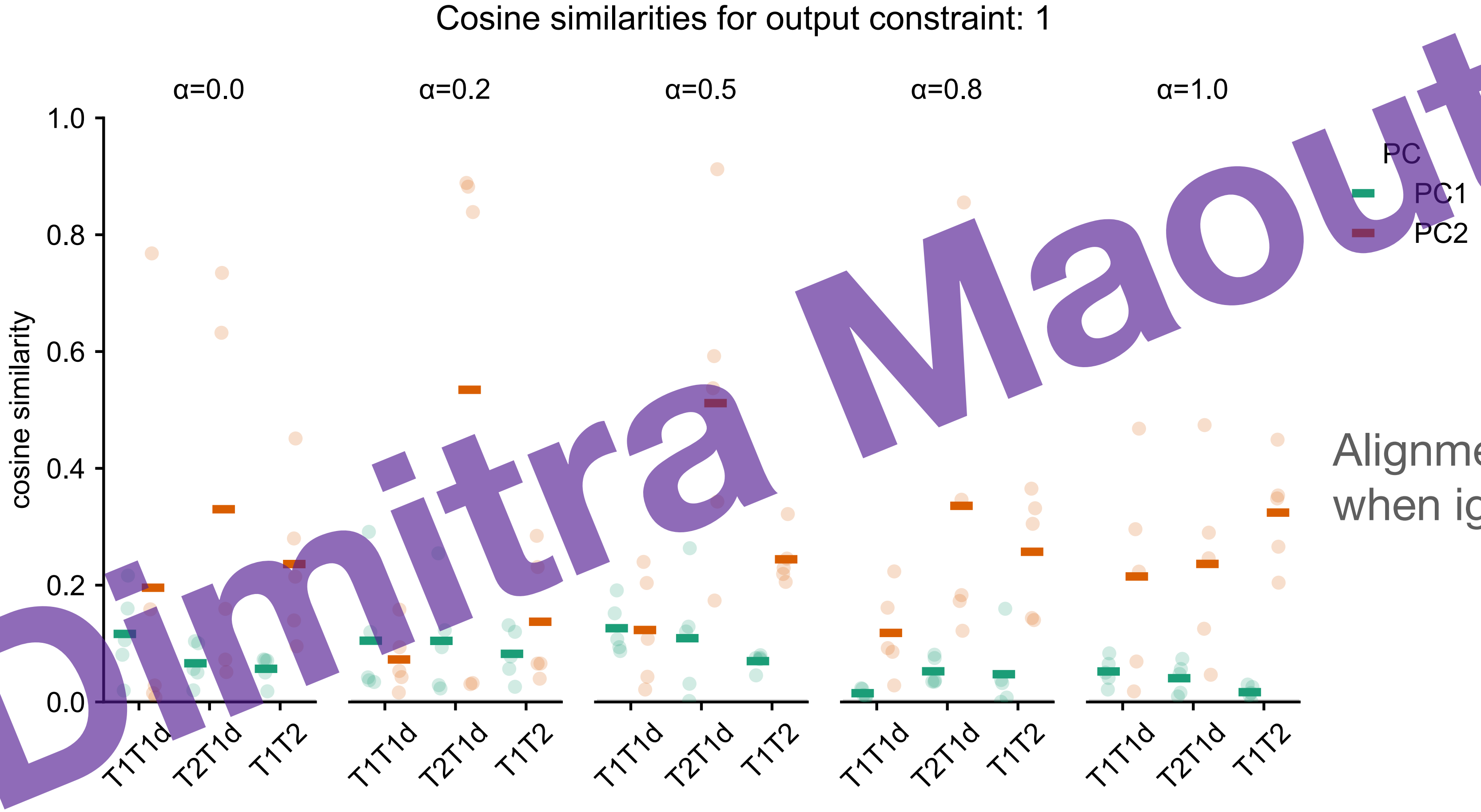
Cosine similarity

$$S_c(PC^1_{T1}, PC^1_{T2}) = \frac{PC^1_{T1} PC^1_{T2}}{||PC^1_{T1}|| \, ||PC^1_{T2}||}$$

0: orthogonal

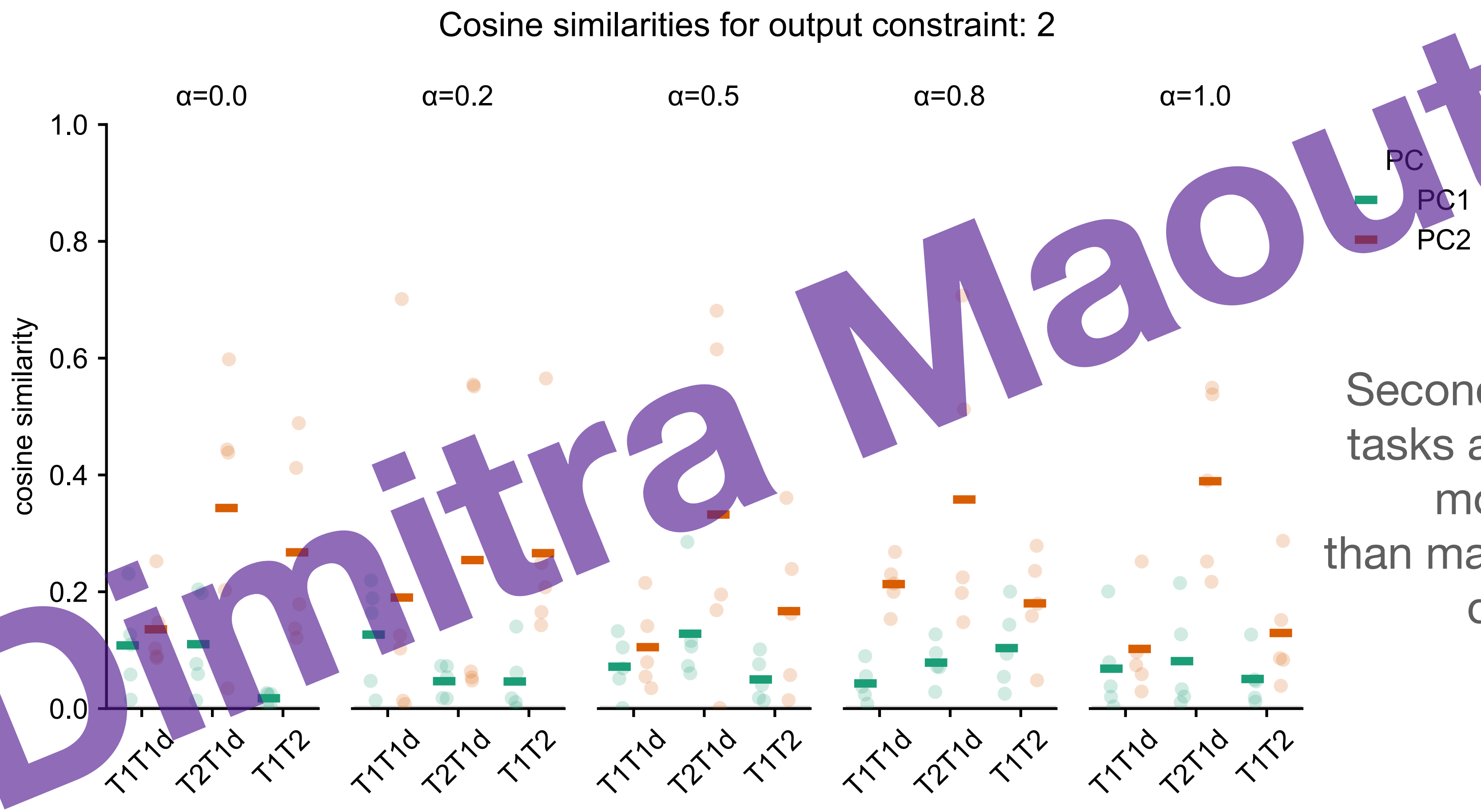
1: aligned

Comparing PC subspaces



Alignment of PC2 larger when ignoring irrelevant output

Comparing PC subspaces

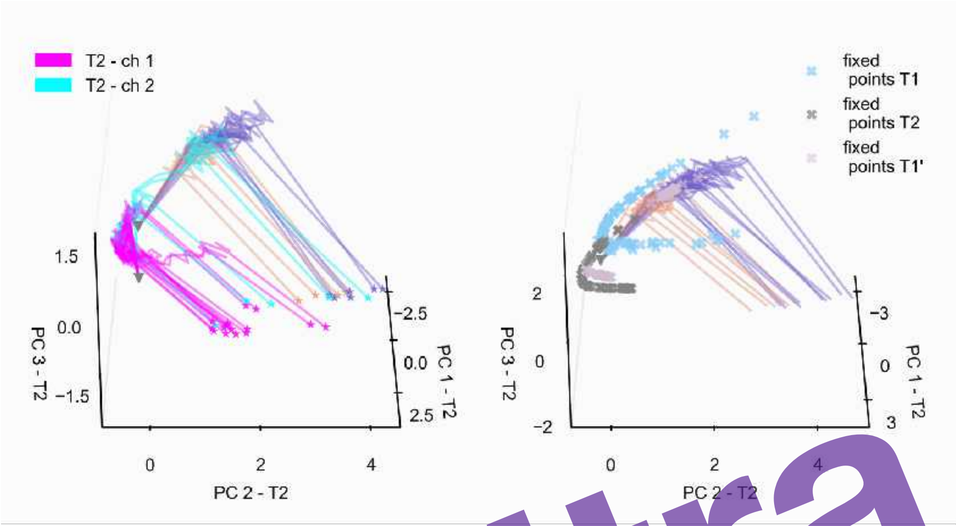


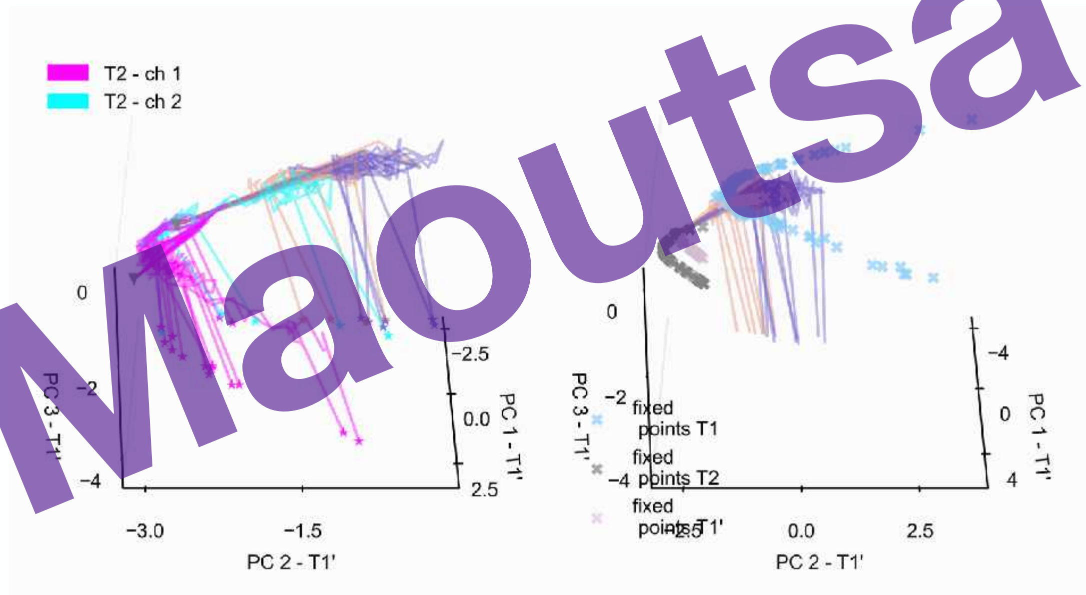
Second PCs between tasks are consistently more aligned than maximum variance directions

Summary

-Inputs with mixed selectivity promote sharing the same dynamical landscape with existing dynamical structure

a=0


T2 - ch 1
T2 - ch 2
fixed points T1
fixed points T2
fixed points T1'
PC 2 - T2
PC 3 - T2
PC 1 - T2
1.5
0.0
−1.5
0
2
4
−2.5
0.0
2.5
−2
−3
3


T2 - ch 1
T2 - ch 2
fixed points T1
fixed points T2
fixed points T1'
PC 2 - T1'
PC 3 - T1'
PC 1 - T1'
−3.0
−1.5
0.0
2.5
−2.5
0
−2
−4
4

ACC feedback to V1?

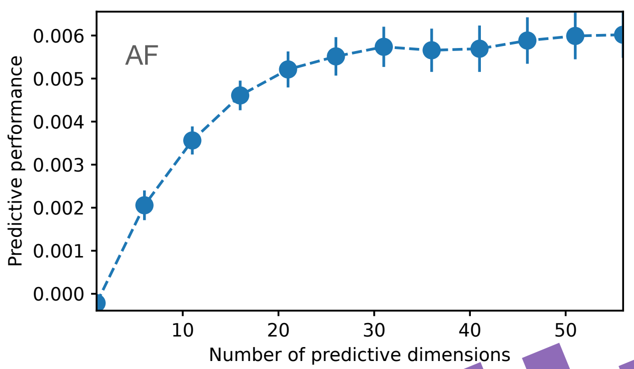
For the project of Sarah Elnozahy

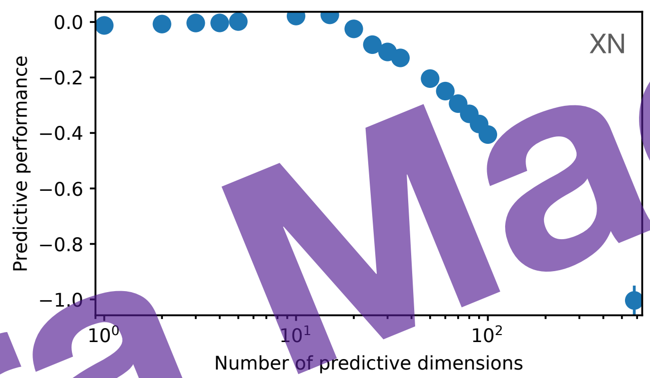
Dimitra Maoutsa

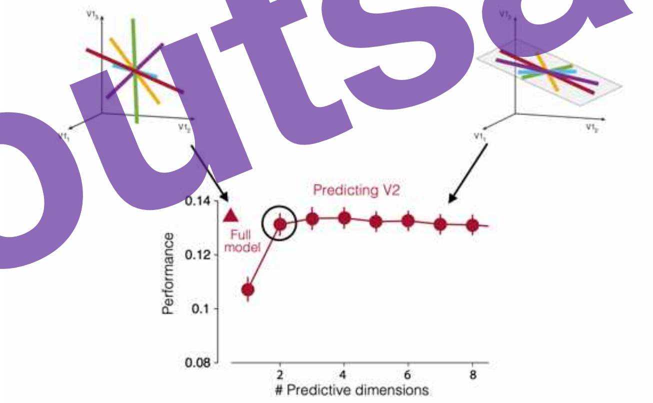
Communication subspace through reduced rank regression

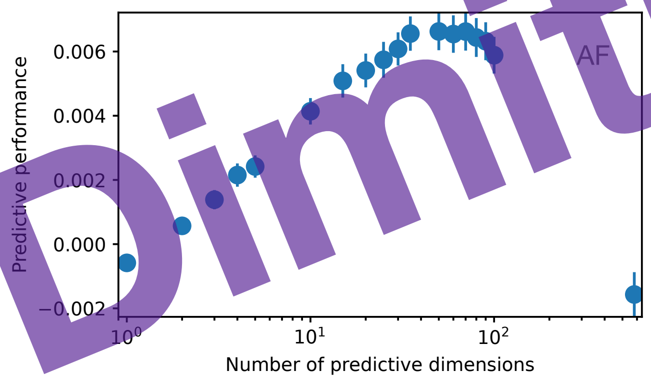
Predict ACC from V1 (V1-> ACC)

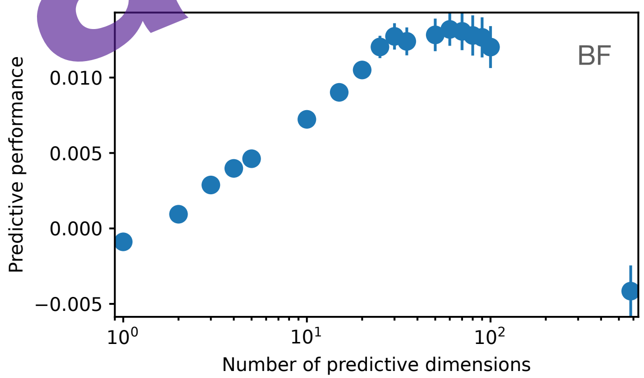
How many dimensions in V1 are required to predict ACC activity

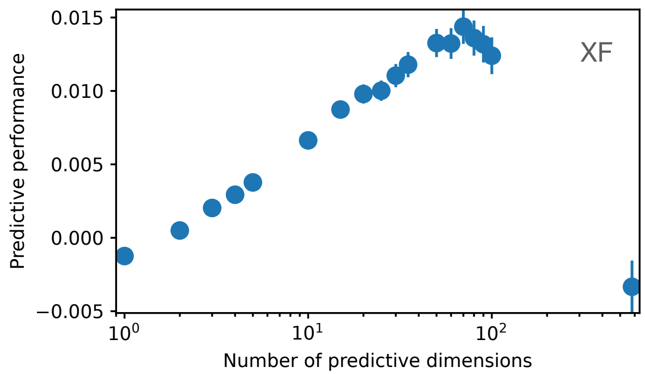








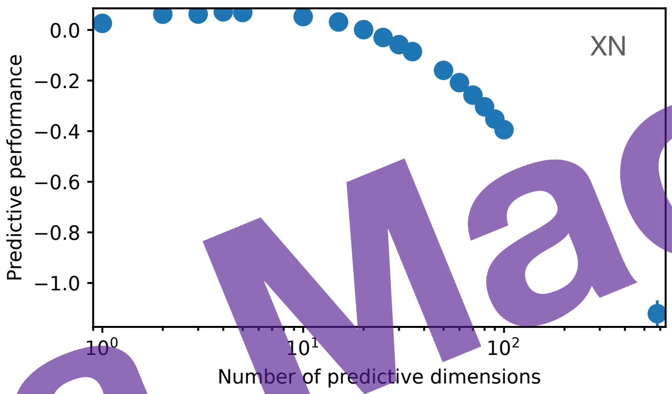


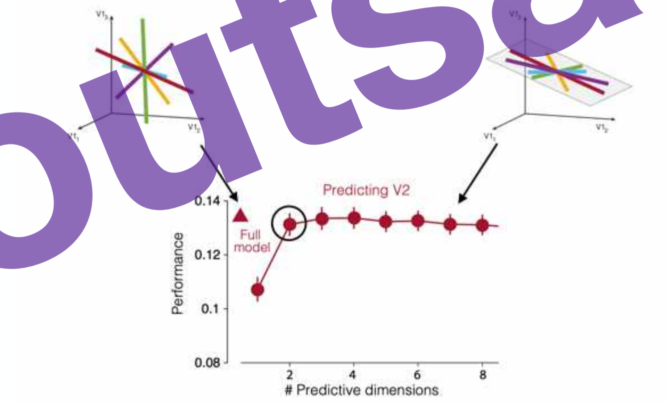


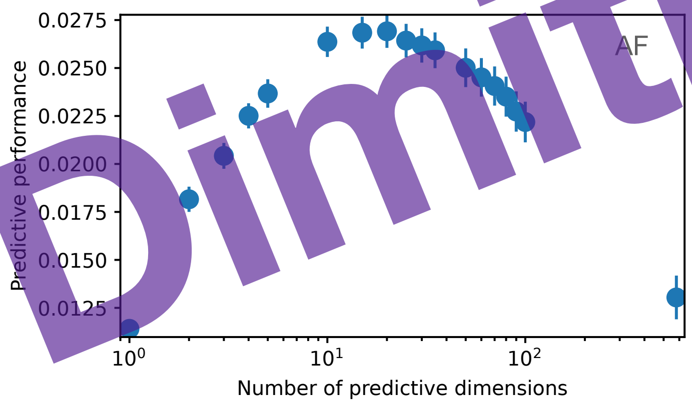
Communication subspace through reduced rank regression

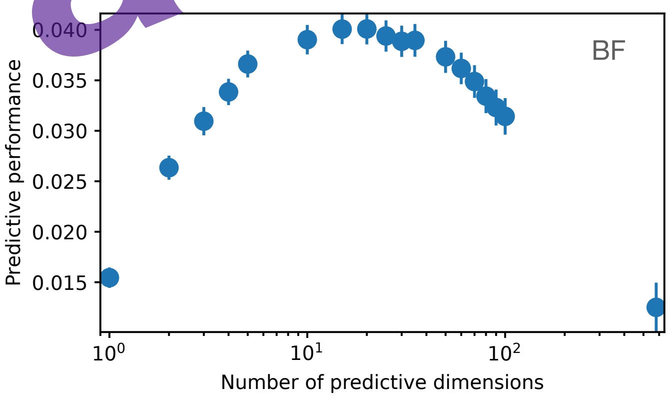
Predict V1 from ACC (ACC-> V1)

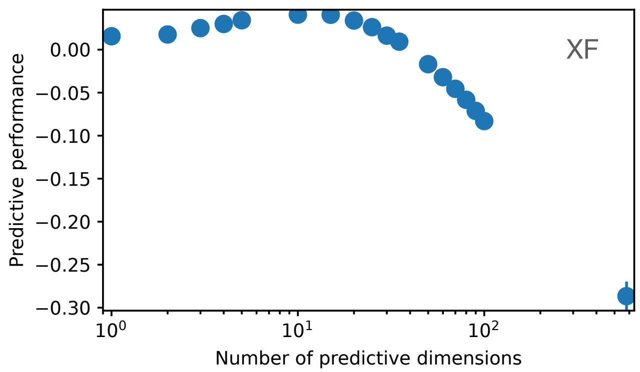
How many dimensions in V1 are required to predict ACC activity

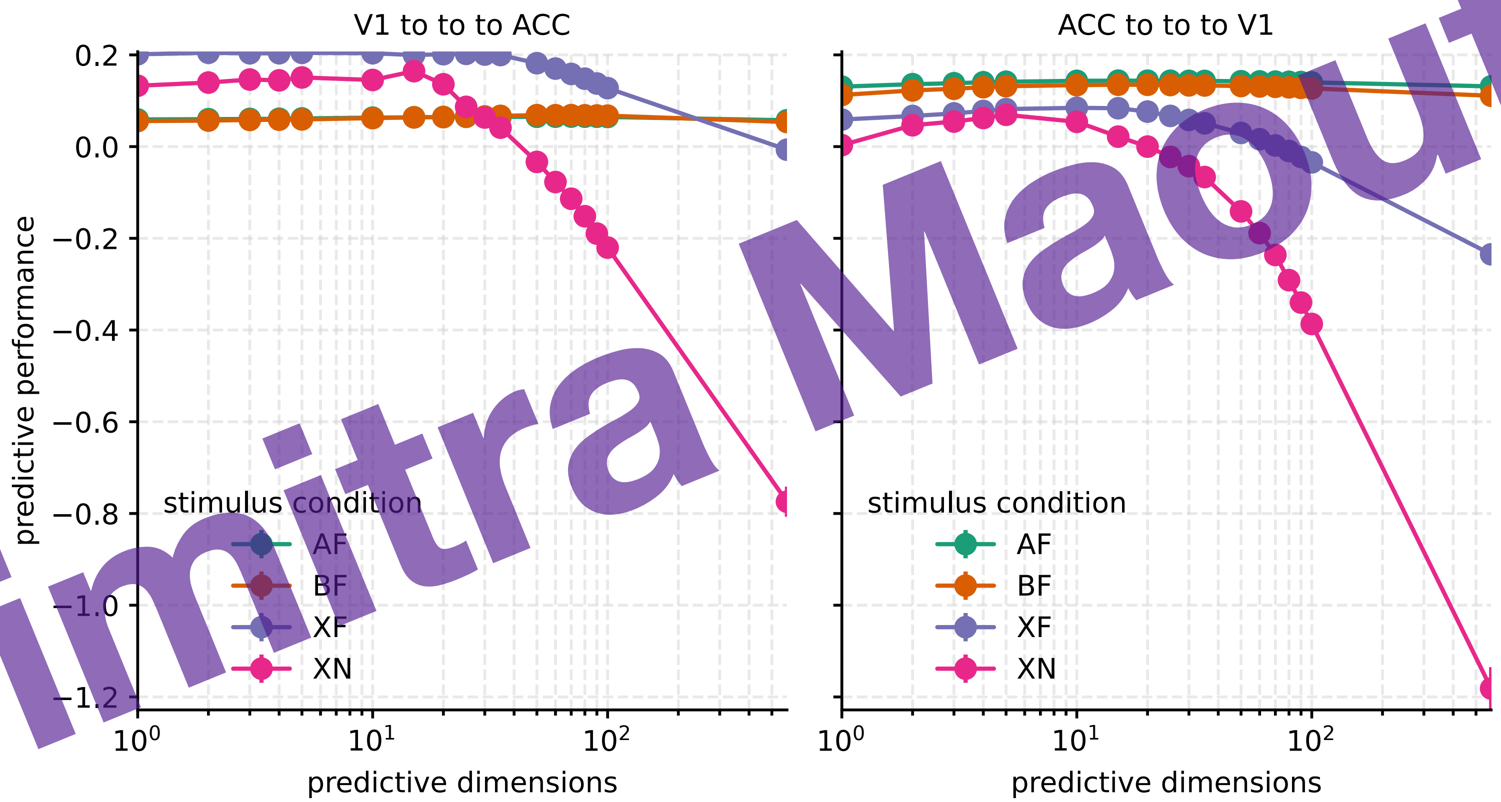






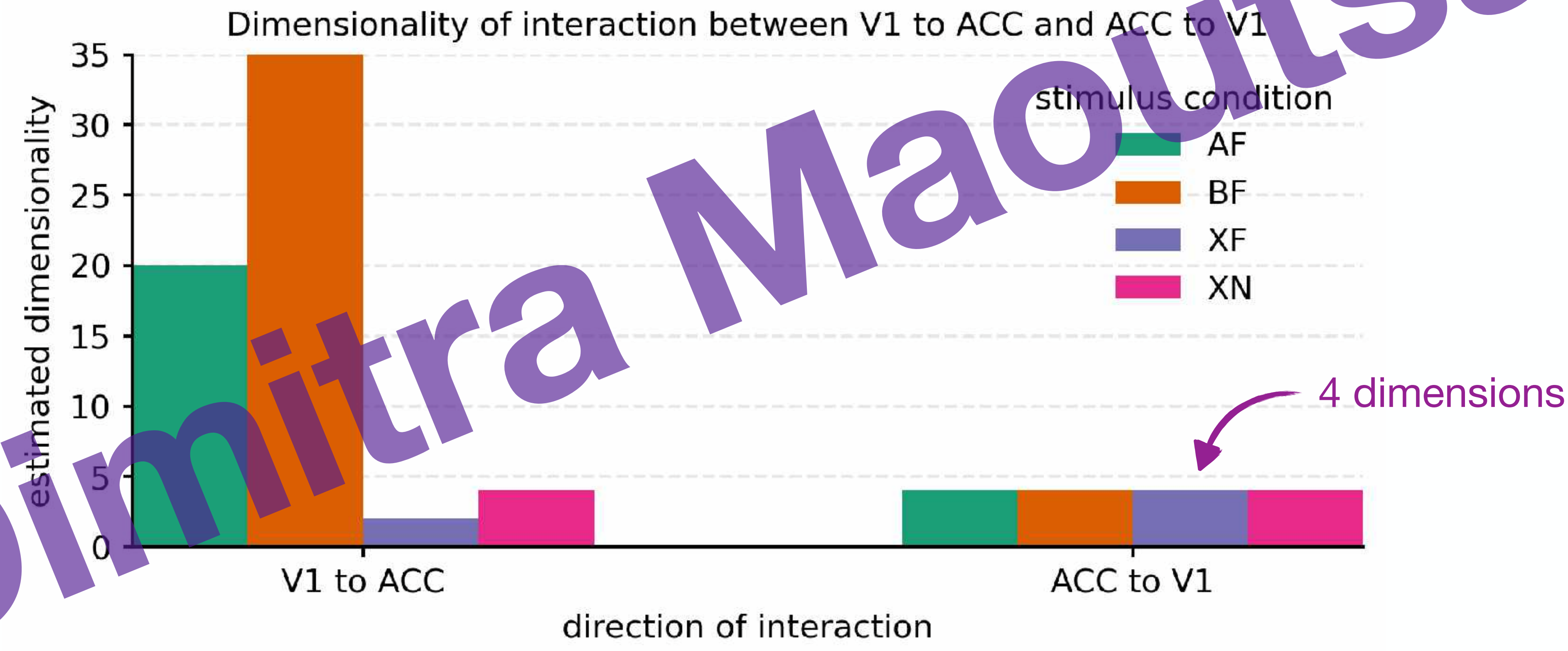






V1 to to to ACC
ACC to to to V1
predictive performance
predictive dimensions
stimulus condition
AF
BF
XF
XN
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
10^0
10^1
10^2

Reduced rank regression predicts low-dimensional feedback from ACC to V1



Summary

- reduced rank regression predicted **high dimensional input from V1 to ACC** and **low-dimensional input from ACC to V1**
- Attempts to fit Generalised Linear models to predict future population responses were not particularly fruitful
- Simultaneous recordings from V1 neurons and ACC axons may improve the performace of the GLM