The Riemannian geometry of discrete computations on continuous manifolds

Julian Brandon[1,3] Angus Chadwick[1] Arthur Pellegrino[1,2,3]

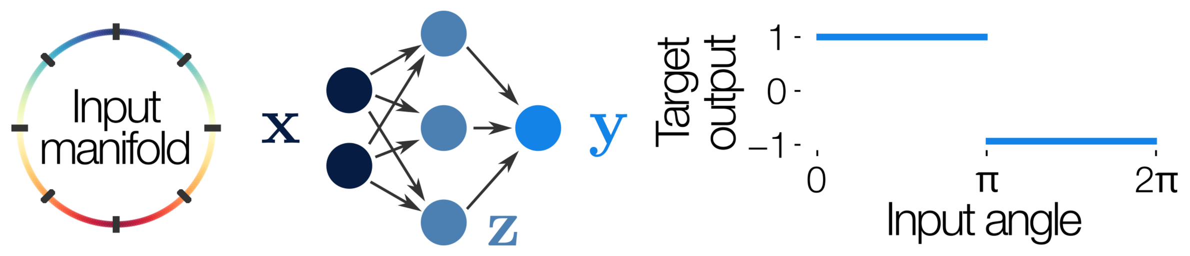
[1]School of Informatics, University of Edinburgh [2]Gatsby Unit, University College London
[3]Data Science Center, Ecole Normale Supérieure

Introduction

Input data to a neural network often lives on low-dimensional continuous manifolds, whose features are represented in the network activation. Yet, deep learning models are often trained to output discrete class labels or to perform symbolic computations. How neural networks transform continuous input manifold geometry into discrete task outputs remains elusive. **Here we show that, over learning, structure emerges in the Riemannian geometry of network activations, which reflects the discrete computations they are trained to perform on continuous input manifolds.**

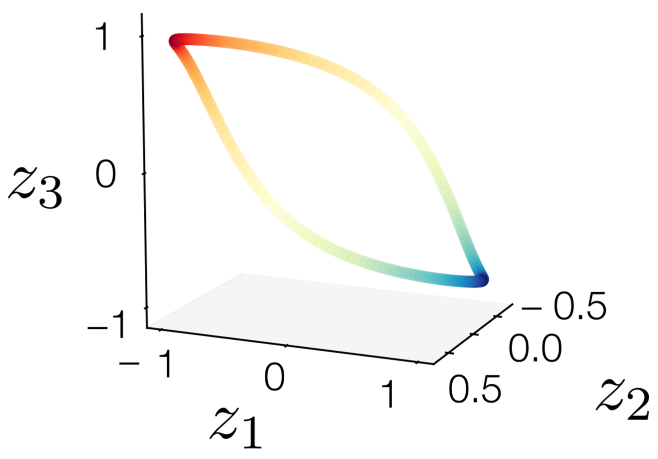
Representational geometry reflects task computations

We first gain intuition into representational geometry by considering a simple DNN.



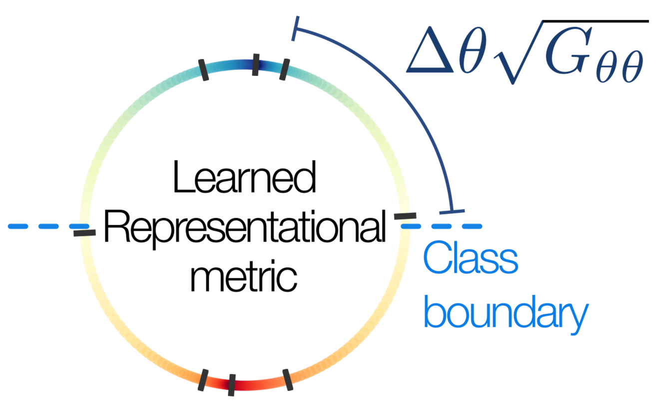
$$\mathbf{x}(\theta) = [\cos\theta, \sin\theta], \quad \mathbf{z} = \phi(W\mathbf{x}), \quad y = \mathbf{d}\cdot\mathbf{z}$$

Neural activity in the hidden layer is constrained to a manifold of the same topology as the inputs.



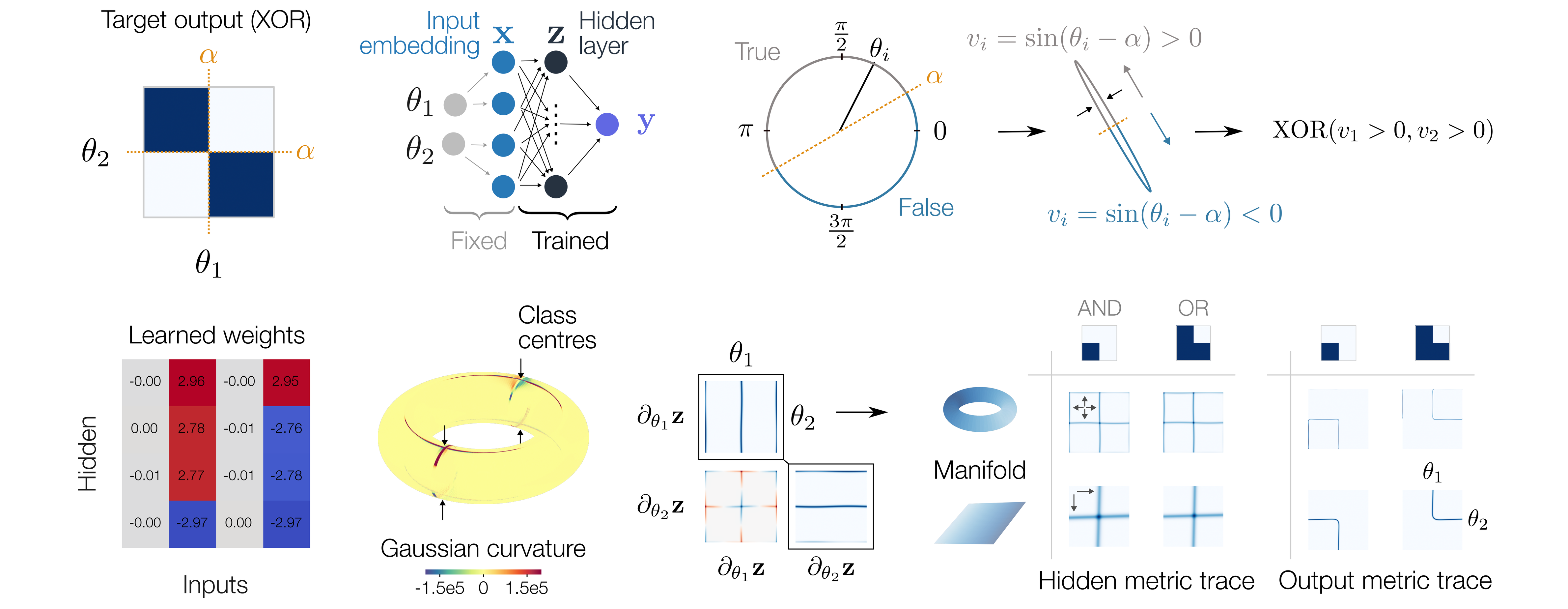
Using the pullback metric we can define distances on the original manifold using distances on the hidden layer activation.

$$G_{\theta\theta} = \partial_\theta \mathbf{z} \cdot \partial_\theta \mathbf{z}$$
$$= \partial_\theta \mathbf{x}^T \operatorname{diag} \phi'(\mathbf{x})^T W^T W \operatorname{diag} \phi'(\mathbf{x}) \partial_\theta \mathbf{x}^T$$



This highlights that, in the neural representation, space has been warped around the decision boundary.

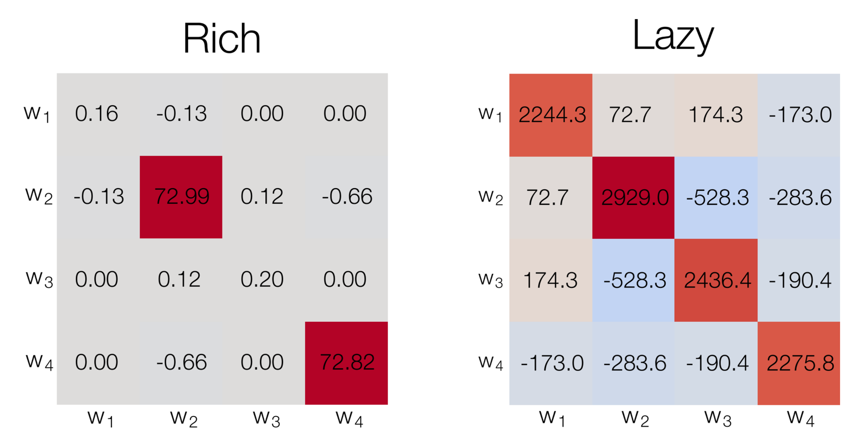
Hidden layer geometry encodes features of discrete computations on the manifold



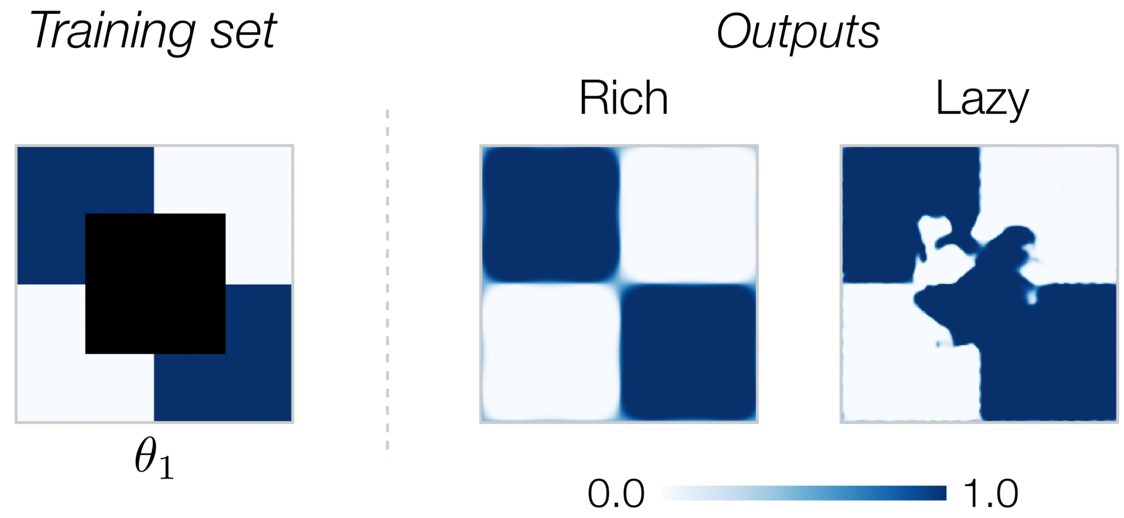
Feature learning involves discretising the input manifold

We varied the initial weight variance $W_{ij} \sim \mathcal{N}(0, \sigma^2)$ to induce rich or lazy learning [1].

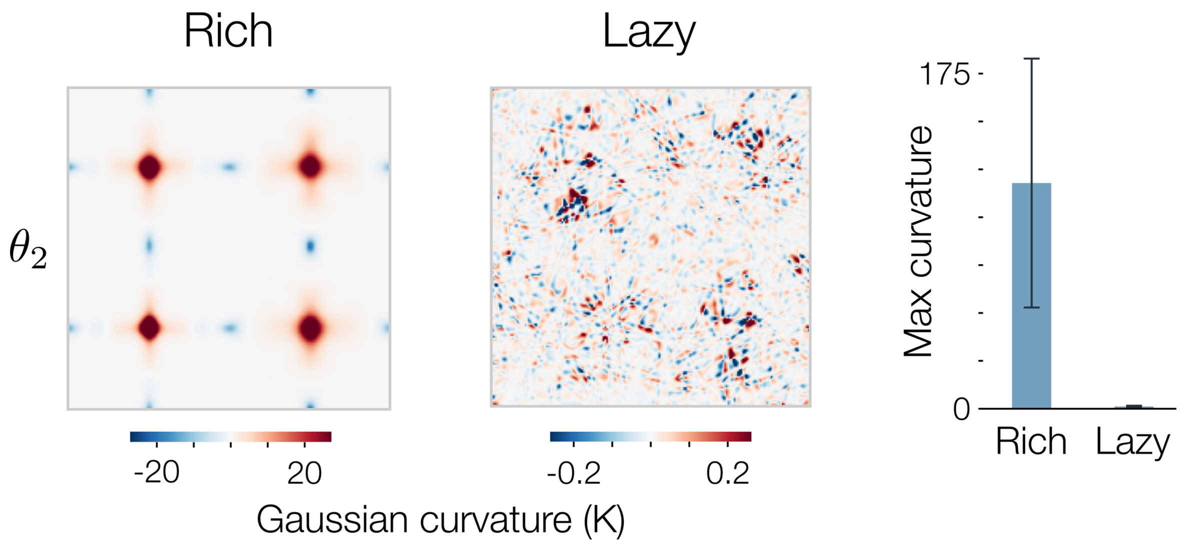
Gram matrix $W^T W \in \mathbb{R}^{n_{\text{in}} \times n_{\text{in}}}$ shows that rich networks learn low-d representations.



This leads to better generalization to unseen inputs and increased noise robustness.



This was afforded by symmetries in the intrinsic geometry of the hidden layer manifold.

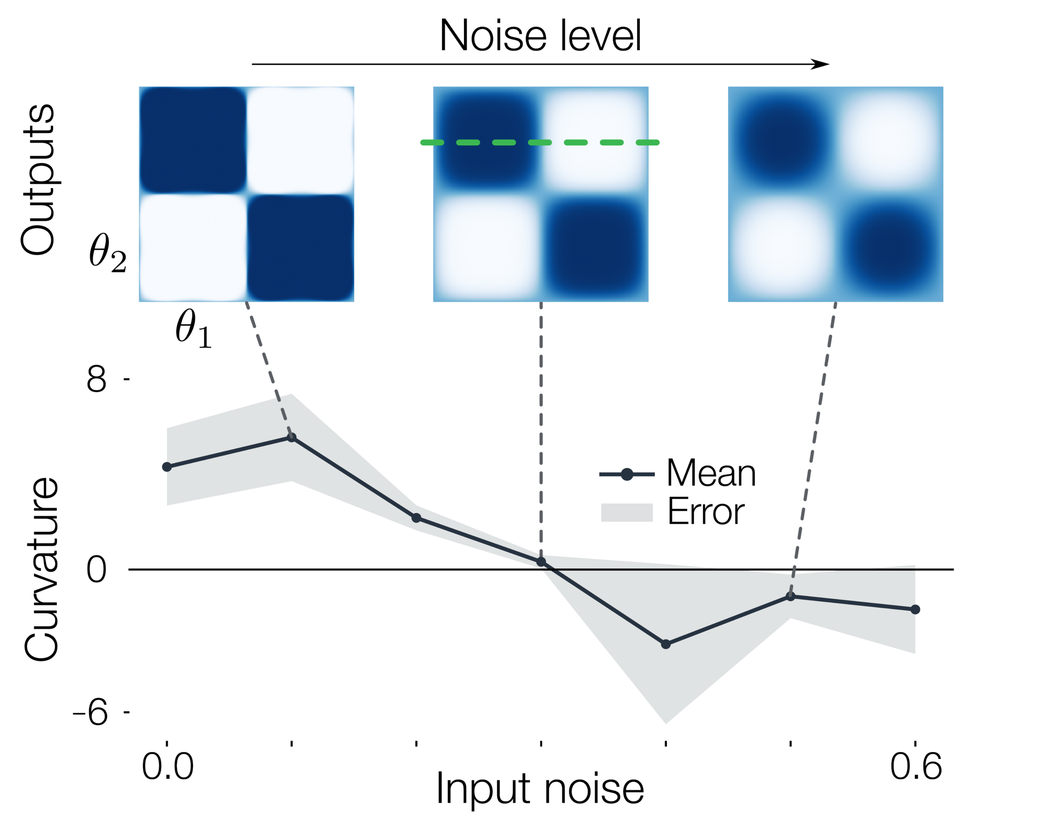


Bayesian computations smooths manifold geometry

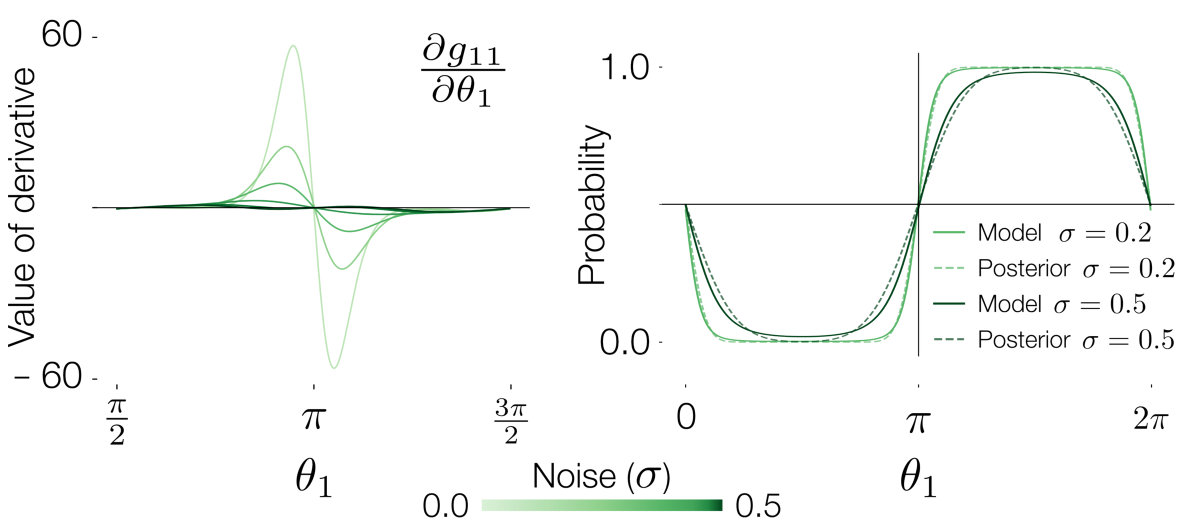
Next, we trained the model to output the posterior distribution of an input with noise:

$$\mathbf{x} = \boldsymbol{\mu}(\theta_1, \theta_2) + \sigma\xi, \quad \xi \sim \mathcal{N}(0, I)$$

We found that at higher noise, models learned flatter hidden-layer representations.



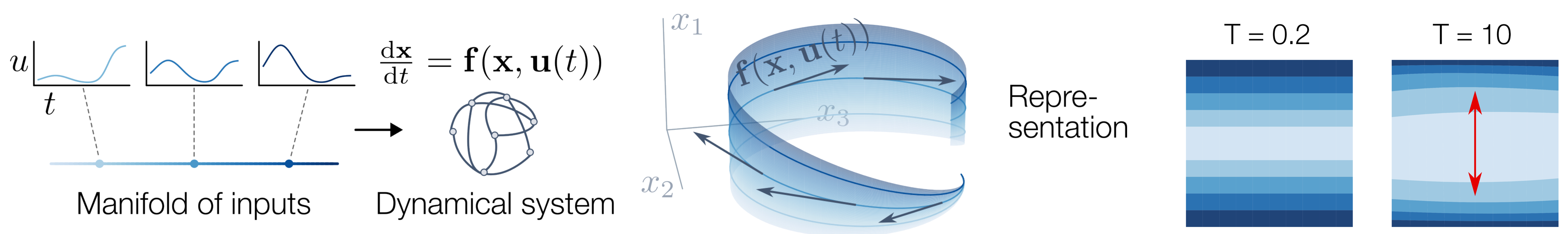
Corresponding to learning flatter posteriors.



Hence, hidden layer activation carries noise-level specific Riemannian geometry corresponding to learning Bayesian computations.

NEURAL INFORMATION PROCESSING SYSTEMS

Extension to dynamical systems receiving time-varying inputs



RNNs perform task computations by dynamically warping neural representations
Pellegrino & Chadwick, NeurIPS 2025.

References

[1] Saxe et al. From Lazy to Rich: Exact learning dynamics in deep linear networks, *ICLR*, 2025.

arthur.pellegrino@ens.fr