Extracting computational mechanisms from neural data using low-rank RNNs

Anonymous Author(s)
Affiliation
Address
email

Abstract

An influential framework within systems neuroscience posits that neural computations can be understood in terms of low-dimensional dynamics in recurrent circuits. A number of methods have thus been developed to extract latent dynamical systems from neural recordings, but inferring models that are both predictive and interpretable remains a difficult challenge. Here we propose a new method called Low-rank Inference from Neural Trajectories (LINT), based on a class of low-rank recurrent neural networks (lrRNNs) for which a link between connectivity and dynamics has been previously demonstrated. By fitting such networks to trajectories of neural activity, LINT yields a mechanistic model of latent dynamics, as well as a set of axes for dimensionality reduction and verifiable predictions for inactivations of specific populations of neurons. Here, we first demonstrate the consistency of our method and apply it to two use cases: (i) we reverse-engineer "black-box" vanilla RNNs trained to perform cognitive tasks, and (ii) we infer latent dynamics and neural contributions from electrophysiological recordings of nonhuman primates performing a similar task.

1 Introduction

As large-scale neural recordings in behaving animals are becoming commonplace, a pressing question is how computational principles can be extracted from the electrical activity of thousands of cells. An influential framework posits that neural computations rely on latent low-dimensional dynamics [46, 51] distributed across populations of neurons [57, 36]. In line with this proposal, a number of data-analysis methods have been developed to infer latent dynamics from neural recordings [56, 18, 29, 25, 9, 24, 17, 26, 7, 15, 13, 33, 34, 38, 16] (reviewed in [8]). While these statistical approaches often provide compelling descriptions of the recorded data, they generally lack a mechanistic interpretation that would, for instance, allow them to make predictions for responses to novel interventions on the underlying neural circuits. How to identify mechanistic, predictive models from neural data currently remains an important challenge.

Recurrent neural networks (RNNs) have emerged as key models for studying neural computations in this framework [55]. Indeed, RNNs can be trained to solve numerous cognitive tasks, exhibiting surprisingly similar dynamical mechanisms as those observed in neural recordings [22, 48, 4, 52, 32], and have also successfully been trained to reproduce the activity of neurons recorded *in vivo* [31, 5, 11, 28]. While potentially predictive, the obtained networks are however typically challenging to understand mechanistically [47, 21, 20, 19, 54, 49], and ongoing research directions aim at reducing

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

them to simplified, interpretable models, for example through the use of linearized dynamical systems [9, 15, 43] or through "network distillation" methods [37, 16].

Here, we exploit a particular class of interpretable RNNs, namely *low-rank RNNs* (lrRNNs) [23, 2, 6, 39, 40, 50], to develop a new method that extracts mechanistic and predictive low-dimensional models from observed neural activity, which can be applied to both artificial and biological neural networks. Our method, Low-rank Inference from Neural Trajectories (LINT) infers a minimal rank lrRNN from a dataset and exploits the theory of low-rank networks to relate the obtained connectivity with low-dimensional dynamics and computations. It produces three outputs: a set of axes for dimensionality reduction of the trajectories, an effective connectivity that implements a latent dynamical system, and predictions for interventions on specific subsets of recorded neurons (fig. 1a).

Our contributions can be summarized in three steps: first, we verify the consistency of LINT by applying it to data simulated from lrRNNs, and show that it recovers the effective part of the connectivity that reproduces the dynamics and computations. Second, we apply LINT to data generated from full-rank RNNs trained on cognitive tasks, and demonstrate that the resulting lrRNNs can capture their dynamics and offer mechanistic insights into how they function. Notably, we identify a novel population-based mechanism enabling context-dependent switching in a vanilla RNN, and verify it by targeted inactivation experiments. Finally, we apply LINT to electrophysiological recordings in nonhuman primates performing a context-dependent decision making task [22]; this reveals that a rank-one network can capture most aspects of the dynamics present in the data, and that computations in this neural circuit appear to be supported by a small proportion of all recorded neurons.

2 Approach

Here we describe our method for extracting an interpretable low-dimensional projection and dynamical model from neural trajectories by fitting a low-rank recurrent network to data.

Low-rank RNNs (lrRNNs). We start from rate-based recurrent neural networks of N units, each characterized by an activation variable x_i which follows the dynamics:

$$\tau \frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} J_{ij}\phi\left(x_j\right) + \sum_{s=1}^{N_{in}} I_i^{(s)} u_s(t) + \eta_i(t). \tag{1}$$

Here $\boldsymbol{J}$ represents the network connectivity matrix and ϕ is a nonlinear transfer function defining the neural firing rate $\phi(x_i)$, which we take here to be $\tanh$. Each network also receives N_{in} input signals $u_s(t)$ via a set of weights $\boldsymbol{I}^{(s)}$ that we refer to as *input vectors*.

Low-rank RNNs represent a subclass of models in which the connectivity matrix is constrained to be of finite rank $R \ll N$ [23, 2, 6]. In this case, $\boldsymbol{J}$ can be written as a sum of outer products of *connectivity vectors* $\boldsymbol{n}^{(r)}$ and $\boldsymbol{m}^{(r)}$:

$$J_{ij} = \frac{1}{N} \sum_{r=1}^{R} m_i^{(r)} n_j^{(r)}. \tag{2}$$

To define a unique representation, we take as $\boldsymbol{m}^{(r)}$ the left singular vectors of the connectivity matrix, and as $\boldsymbol{n}^{(r)}$ the right singular vectors multiplied by the corresponding singular value.

Network inference. We infer lrRNNs from data by training the connectivity parameters to reproduce recorded neural trajectories (either single-trial or condition-averaged trajectories). Specifically, we use back-propagation on connectivity vector parameters $n_i^{(r)}$ and $m_i^{(r)}$ and input parameters $I_i^{(s)}$ to minimize the squared difference between target and reproduced trajectories:

$$\mathcal{L} = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{t=1}^{T} (\phi(x_i^{(c)}(t)) - \phi(\tilde{x}_i^{(c)}(t)))^2 \tag{3}$$

where $\tilde{x}_{i,t,c}$ represents the target trajectory in condition c, for neuron i and timestep t and $x_{i,t,c}$ the corresponding trajectory produced by the model.

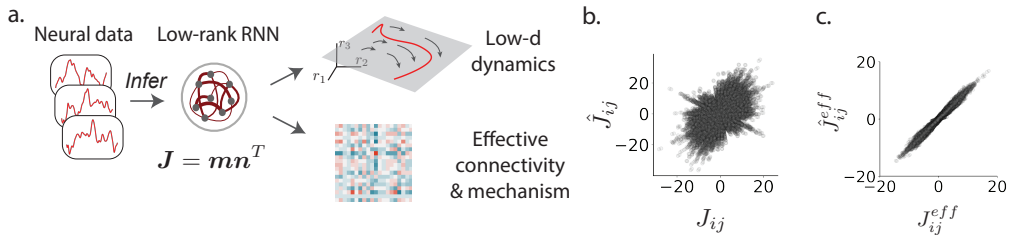

Figure 1: a. LINT pipeline: Low-rank RNNs are fitted to either simulated or recorded neural trajectories. The obtained lrRNNs provide interpretable low-dimensional dynamics as well as a computational mechanism based on an effective connectivity structure. b-c. LINT tested on trajectories simulated from a rank-one lrRNN trained to perform the CDM task (see Table 1 for results across tasks). b. For each pair of neurons, relationship between the original (J_{ij}) and inferred ($\hat{J}_{ij}$) synaptic connectivity (512^2 pairs, $r = 0.57$). c. For the same networks, relationship between the original effective connectivity (J_{ij}^{eff}, see text) and the inferred one ($\hat{J}_{ij}^{eff}$, $r = 0.99$)

Dimensionality reduction. One property of lrRNNs is that they constrain the activity vector $\boldsymbol{x}(t)$ to evolve in a low-dimensional subspace spanned by the R connectivity vectors $\boldsymbol{m}^{(r)}$ and the N_{in} input vectors $\boldsymbol{I}^{(s)}$ [2]. The trajectories therefore explore at most $R + N_{in}$ dimensions and can therefore be parametrized as:

$$\boldsymbol{x}(t) = \sum_{r=1}^{R} \kappa_r(t)\boldsymbol{m}^{(r)} + \sum_{s=1}^{N_{in}} v_s(t)\boldsymbol{I}^{(s)}, \tag{4}$$

where $v_s(t)$ are the low-pass filtered input signals $u_s(t)$ and κ_r are a set of latent variables generated by recurrent activity. Thus, lrRNNs provide by design a reduction of N-dimensional neural activity to at most $R + N_{in}$ dimensions that can be directly interpreted in terms of components on a *recurrent subspace* spanned by the connectivity vectors $\boldsymbol{m}^{(r)}$ and on an *input subspace* spanned by inputs vectors $\boldsymbol{I}^{(s)}$ [52].

Latent dynamics. In lrRNNs, the latent variables $\kappa_r(t)$ form a non-linear low-dimensional dynamical system, described by:

$$\frac{d}{dt}\boldsymbol{\kappa}(t) = F(\boldsymbol{\kappa}(t), \boldsymbol{u}(t)) \tag{5}$$

where $\boldsymbol{\kappa}(t) = \{\kappa_r(t)\}_{r=1..R}$ is an R-dimensional vector representing the activity on the recurrent subspace, $\boldsymbol{u}(t)$ represents the input signals, and F is a non-linear function that can be directly determined from network connectivity parameters [2, 6] (Appendix A). Moreover, it can be shown that rank-R networks are universal approximators of R-dimensional dynamical systems [2], and thus that the function F can approximate any non-linear mapping in R dimensions.

Task-optimized RNNs. We applied our method first to trajectories produced by task-optimized RNNs, which produce an output signal $z(t)$ from the recurrent dynamics (1) via a linear readout:

$$z(t) = \sum_{i=1}^{N} w_i \phi(x_i(t)) \tag{6}$$

In task-optimized networks, the parameters are trained with backpropagation through time to minimize the squared error between the output $z(t)$ and a target $z^*(t)$. In this work, we consider four cognitive tasks: Decision Making (DM), Working Memory (WM), Context-Dependent Decision Making (CDM, fig. 2a) and Delayed Match-to-Sample (DMS) (see Appendix B for task definitions). We then generate neural trajectories from both low-rank (in section 3.1) or full-rank (see section 3.2) task-optimized RNNs. All networks are implemented in pytorch [27], and training details can be found in Appendix C.

Table 1: Synthetic data validation results for LINT
(CC: connectivity correlation, ECC: effective connectivity correlation)

Task	Rank	Trajectory R^2	CC	ECC
Decision Making (DM)	1	0.97	0.50	0.99
Working Memory	2	0.96	0.43	0.93
Context-dependent DM	1	0.91	0.57	0.99
Delayed Match-to-Sample	2	0.98	0.39	0.63

3 Results

3.1 Validation with synthetic data and effective aspects of connectivity

We first validated the capacity of the LINT method to recover low-rank connectivity features and reproduce neural trajectories on data simulated from lrRNNs trained on cognitive tasks. For this, we first train lrRNNs with 512 neurons to produce a correct behavioral output on four systems neuroscience tasks, each time retaining the minimal rank solution [6] (see Table 1). For each task-optimized lrRNN, we then generate simulated trajectories corresponding to trials in the trained task, and apply our method to fit these trajectories using equivalent models with the same rank and number of neurons. We found that the inferred networks were able to reliably reproduce the original trajectories when fed with the same inputs, as quantified by the R^2 scores between original and fitted trajectories. Although the inferred networks were not explicitly trained on behavioral outputs, they were able to perform their task accurately when plugged to the original readout vector, implying that they also captured behavioral aspects of the original networks.

LINT provides an inferred low-rank connectivity that we could compare with the original one. We noted that the inferred connectivity weights are not identical to the original ones, although a certain degree of correlation is present (fig. 1b and Table 1, column CC for connectivity correlation). However, from a theoretical point of view, a range of different connectivity matrices can lead to identical dynamics [30, 10, 2]. The low-rank framework allows for a precise characterization of the relevant part of the connectivity, and in particular allows us to define an effective connectivity matrix $\boldsymbol{J}^{\text{eff}}$ that captures the minimal features required to obtain certain dynamics (see details in Appendix A). Our results on the synthetic data show that indeed there is a very high degree of similarity between original and inferred effective connectivities, demonstrating that LINT retrieves the relevant aspects of the neural connectivity (Table 1, ECC for Effective Connectivity Correlation, and fig. 1c).

3.2 Application to reverse-engineering full-rank RNNs

We next ask to which extent our method can infer computational mechanisms from activity generated by unconstrained, full-rank networks. Indeed, RNNs trained on simple tasks without a rank constraint often exhibit low-dimensional dynamics that can be related to the way computations unfold in the network [47, 21, 20, 19, 49]. It has been shown that such low-dimensional dynamics can be reproduced by low-rank networks [2], yet the best approach for inferring the corresponding low-rank connectivity remains to be determined. Here we show that our method vastly outperforms a direct low-rank approximation of the connectivity matrix based on truncating the SVD [40]. We then demonstrate how the inferred low-rank models can be used to interpret the computational mechanisms in the original full-rank networks.

3.2.1 Extracting low-dimensional dynamics through low-rank connectivity

We consider full-rank, vanilla RNNs trained on two complex systems neuroscience tasks, respectively context-dependent decision-making[22] (CDM) and delayed match-to-sample[4] (DMS). The full-rank RNNs reach a 100% accuracy on each task, and as expected exhibit low-dimensional dynamics. From these RNNs, we generate trajectories corresponding to trials in the trained tasks, and then use LINT to infer lrRNNs of increasing rank.

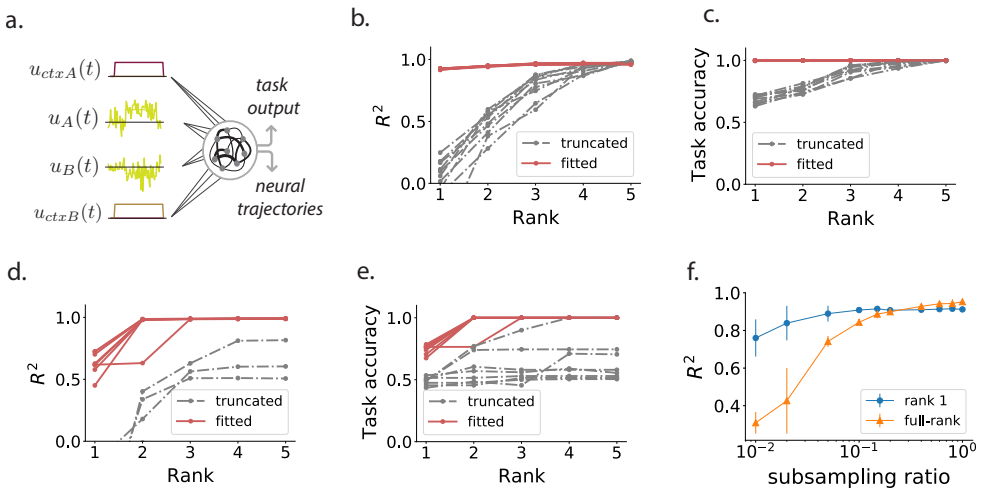

Figure 2: LINT applied to full-rank task-optimized RNNs. a. Description of the CDM task inputs and outputs. Neural trajectories correspond to the neural firing rates, choice output from a linear readout as in eq. 6. b. For the CDM task, similarity between trajectories produced by original full-rank networks and a series of fitted networks of increasing ranks (red), or low-rank networks obtained by truncating the original connectivity matrix (grey). Each line corresponds to a randomly initialized full-rank network. c. Task accuracy of the same networks when associated with the original task readout. d-e. Same as b-c. for the DMS task. f. For the CDM task, similarities between trajectories of the original and fitted networks of rank one (blue) or full-rank (orange), when networks are fitted only to a random subsample of neurons of the original network. Error bars: mean ± std over 10 random subsamples for each ratio value.

In the CDM task, a network receives two signal inputs which vary randomly around a positive or negative mean, as well as two binary context cues, one of which is set to 1 in each trial. The network has to output the average sign of the signal indicated by the active context, while ignoring the other signal (fig. 2a, see Appendix B for task details). Applying LINT, we found that rank-one networks are sufficient to reproduce the original trajectories with a high goodness of fit. Increasing the rank improved only marginally the fit (fig. 2b). In contrast, low-rank networks obtained by truncating the SVD of the trained full-rank connectivity matrix required a rank of 4 or 5 to reach a comparable fit (fig. 2b), or much more for different hyperparameters (sup. fig. 1). This shows that LINT captures a simplified connectivity structure that could not be trivially extracted from the original connectivity. Moreover, even though not explicitly trained on the behavioral outputs, the inferred rank-one networks performed the task correctly when plugged onto the original readout, implying that they captured well the task-related dynamics (fig. 2c).

For the DMS task, we similarly found that rank-two networks were able both to capture the dynamics of the original unconstrained network and to accurately perform the task, with no notable improvement when the rank is increased (fig. 2d-e). As for the CDM task, LINT vastly outperformed direct truncation of the original connectivity matrix. For both these tasks, reproducing the experiment over a larger number of unconstrained RNNs, trained with diverse hyperparameters and random seeds leads to similar results (sup. fig. 1).

Finally, note that the CDM and DMS tasks are particularly prone to low-dimensional implementations, which is not the case of all computations of interest in systems neuroscience [14]. To probe the capacities of LINT with higher-dimensional tasks, we applied it to full-rank networks trained on the K-back task, where a network receives a random number P of stimulations in $\{-1, 1\}$ and has to output the K-th last stimulation received. This task requires the implementation of an internal working memory with a capacity of at least K bits, and should thus require a rank at least K. We indeed observed that increase, but also noticed that implementations using a rank as low as possible

seemed to prevail (see details in appendix F and sup. fig. 8). These results show that LINT is also helpful in probing the dimensionality required by computational paradigms.

Subsampling. We then investigated the robustness of LINT to the subsampling of neural data. Indeed, in a biological setting, one cannot expect to have access to all units of a network. It is thus important to assess whether we can still recover the relevant low-dimensional dynamics and information from only a handful of neurons of the original network. We considered a full-rank network trained on the CDM task, and fitted networks to its trajectories, either without constraining their rank or by fixing it to one. Without subsampling, the inferred full-rank networks have a very slight advantage over rank-one networks in reproducing the original trajectories. Yet when considering subsamples of neurons, rank-one networks appear to be more robust than unconstrained ones, keeping a good performance until subsampling ratios as low as 1% of original neurons (fig. 2f).

3.2.2 Extracting computational mechanisms from inferred low-rank connectivity

Low-rank models of behavioral tasks open the door to mechanistic interpretations of the underlying dynamics [23, 6]. Here we show how LINT allows us to extract computational mechanisms from full-rank networks performing the CDM task.

A first output of LINT is a set of interpretable axes defining a task-related subspace for dimensionality reduction. The inferred rank-one model of the CDM task specifically yields five axes that correspond to the two stimulus inputs, the two context inputs and an internal latent variable, generated by recurrent connectivity, which represents integrated evidence and therefore choice. Projecting the activity of the full-rank network along these axes shows how low-dimensional dynamics transform inputs into the choice output (fig. 3). In this case, the axes determined by LINT are closely related to those obtained by standard targeted dimensionality reduction (TDR, sup. fig. 2) [22]. However, in contrast with TDR, the axes inferred by LINT correspond to connectivity features causing the low-dimensional dynamics, and provide a method for an unsupervised discovery of recurrently-generated latent variables. In particular, we did not *a priori* specify that the recurrent loop output $\boldsymbol{m}$ should encode choice, but rather observed that choice was generated through this recurrent mechanism.

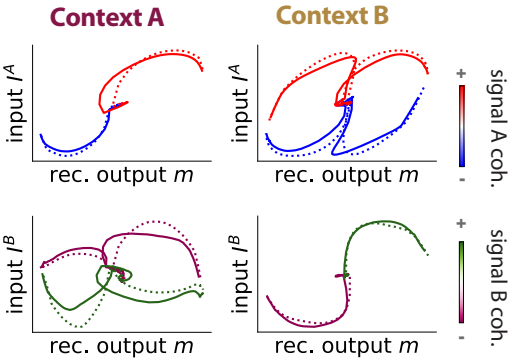

Figure 3: Low-dimensional projections produced by LINT. Trial-averaged trajectories for different task conditions, in the original full-rank network (full lines) and the inferred low-rank network (dashed lines), projected on axes obtained from the fitted lrRNN connectivity: input A and B axes, and the output axis of the rank-one recurrent connectivity $\boldsymbol{m}$, which encodes choice. All trajectories start at center.

Going beyond dimensionality reduction, LINT extracts an effective low-dimensional model of the task. For the CDM task, the inferred model is a one-dimensional, non-linear latent dynamical system, where the latent variable integrates the relevant evidence. To understand how context-dependent selection of the evidence was performed, we analyzed the connectivity parameters of the inferred rank-one RNN. Indeed, recent work has introduced a clustering method for analyzing low-rank connectivity, and has showed that in trained lrRNNs, context-dependent integration relies on a gain-modulation of latent dynamics mediated by two distinct populations of neurons [6]. Applying a similar clustering approach to rank-one networks inferred by LINT leads to a comparable mechanism. Specifically, this method identifies two populations distinguished by strong values of the fitted weights for the context-cue inputs (fig. 4a top and sup. fig. 2). In each context, these context-cue inputs place the neurons belonging to the two populations at different positions on the non-linear transform function $\phi(x)$ (fig. 4b top), thereby modulating in opposite ways their gains (defined for each neuron

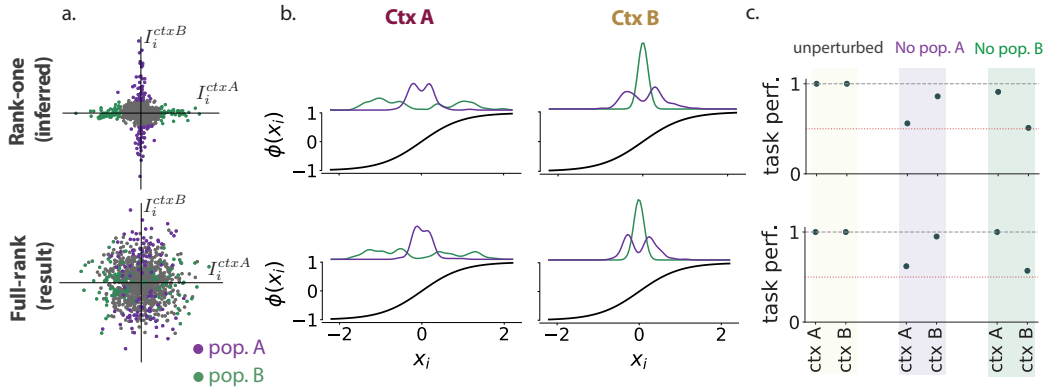

Figure 4: Extracting mechanisms and testing predictions from a low-rank network (top) inferred from a full-rank network trained on the CDM task (bottom). a. Three populations of neurons (grey, green and purple) are identified by clustering connectivity parameters in the fitted rank-one network (top). Here, we plot the same populations in joint distributions of contextual input vectors in the rank-one (top) vs full-rank (bottom) networks. The populations are not directly identifiable from input parameters in the original network (bottom). b. Neural transfer function (tanh) and smoothed distributions of mean neural activity for populations A and B in the inferred rank-one network (top) and the full-rank network (bottom). c. Inactivation experiments: task performance on each context for unperturbed networks and after inactivating populations A and B in the fitted rank-one network (top) and inactivating the same neurons in the original unconstrained network (bottom). Red dots: chance level.

as the local slope $\phi'(x)$). Combined with different statistics of the connectivity parameters on each of the populations, this modulation is sufficient to implement the desired context-switching behavior of the network (see details in Appendix D). The same pattern of gain modulation was found in the original full-rank network, indicating that it is using a similar mechanism (fig. 4b bottom).

The computational mechanism extracted using LINT produces predictions for inactivations that can be directly tested in the original full-rank model. The analysis described above assigns neurons to different populations that have specific computational roles. Specifically, in the inferred rank-one network, inactivating separately each population leads to a specific loss of performance in a single context, but not the other one (fig. 4c top and sup. fig. 2 for detailed error patterns). Since individual neurons in the original networks directly correspond to neurons in the rank-one network on a one-to-one basis, one can map those populations back onto the original network and reproduce the inactivation experiment. The predicted effects of inactivations on context-dependent performance are directly reproduced in the full-rank network, demonstrating that it relies on the identified computational mechanism (fig. 4c bottom and sup. fig. 2). One can note that although the two identified populations clearly stand out in the inferred rank-one connectivity, they are not directly apparent in the original full-rank connectivity (fig. 4a bottom), showing that although the mechanism extracted by LINT applies to the full-rank network, using an lrRNN was a necessary step in this process.

3.3 Application to neural recordings

We next apply our method to *in vivo* recordings. We consider here electrophysiological recordings from non-human primates performing a context-dependent decision-making task similar to that studied in previous sections [22]. More specifically, two macaques (designed by A and F) were presented with random dots stimuli that varied along two dimensions: the overall motion direction of the dots, and their overall color, ranging from two extremes with a set of intermediary coherences in between. At the beginning of each trial, a cue indicated a context for the trial, ordering the subject to report either the average motion or color with an eye saccade, while ignoring the irrelevant evidence. Non-simultaneous recordings were performed in the frontal eye field (FEF) an area of the

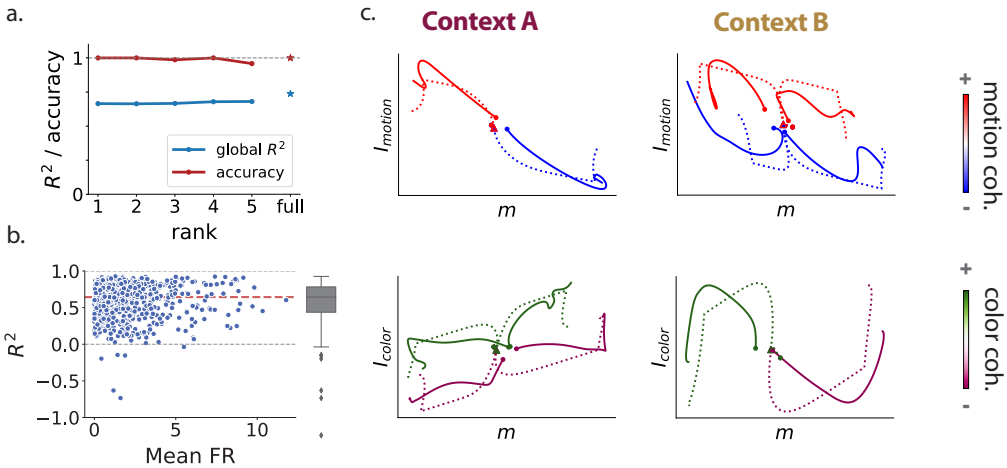

Figure 5: LINT applied to neural recordings from a non-human primate performing a context-dependent task (monkey A) [22]. a. Global R^2 of model trajectories with respect to original ones, and task accuracy of inferred networks of increasing ranks. b. For the rank-one inferred network, R^2 of each recorded neuron plotted against its average data firing rate, median R^2 in red, with marginal box-plot of R^2 values. 14 neurons with an $R^2 < -1$ are not shown, all with a mean firing rate < 2.2 (see sup. fig. 4). c. Trial-averaged trajectories in the recorded data (full lines) and for the rank-one model (dashed lines), for different task conditions (only one coherence value for each coherence sign is plotted here), projected on axes identified by LINT (motion and color input axes and the output axis $\boldsymbol{m}$ of the rank-one recurrent matrix). Trajectories start at the center.

macaque prefrontal cortex, with respectively 727 neurons for monkey A and 574 neurons for monkey F recorded in all 72 conditions the task presented (ignoring error trials).

For the analysis, we start by binning (5 ms) and smoothing (50 ms std Gaussian window) spike train data, and then compute the trial-averaged response of every neuron in each of the task conditions, forming a pseudo-population tensor of size $N \times T \times 72$ with N the number of neurons in each monkey and $T = 150$ the number of discrete time steps. We denoise this tensor by projecting its first axis on the subspace spanned by the top 12 principal components of the pseudo-population activity, and then project back to the high-dimensional space spanned by all neurons [22]. Finally, in order to consider only task-related neural activity, we subtract from each neuron's trajectory its condition-averaged mean. We denote the entries of the final obtained tensor by $\tilde{x}_{i,t,c}$.

We apply LINT to the obtained trajectories, training the activation of each unit in the low-rank RNN (noted $x_{i,t,c}$) to match the corresponding pre-processed activity. Importantly, the inferred networks receive inputs following the same structure than in the CDM task of previous sections, that is two noisy signal inputs and two contextual cue inputs (fig. 2a), and are left unconstrained for the first 350 ms of each trial while receiving only contextual cues, to account for the original task procedure. In particular, we make the implicit hypothesis that choice signals are generated by the recurrent activity of the network from received inputs. The quality of fits is quantified by leaving out a random subset of 8 conditions during network inference, and evaluating the R^2 of fitted networks on these left-out conditions.

We find that for both monkeys the neural activity is well reproduced by a rank-one network, with minimal improvements when the rank is increased (fig. 5a and sup. fig. 5, $R^2 = 0.66$ for monkey A, and $R^2 = 0.57$ for monkey F). Moreover, when simulated independently, the inferred rank-one networks are able to perform the task by adding a linear readout (accuracy curves in fig. 5a, sup. fig. 5). This is the case even though the recurrent and input connections were not trained on task performance, demonstrating that the reproduced trajectories contain information about the choice made by the monkey. The activity of both original and reproduced trajectories can moreover be projected on the axes identified by LINT, providing a geometrical picture of task execution (fig. 5c). These axes are

closely related to those found by targeted dimensionality reduction: notably the context, motion, and color axes identified by TDR closely match the corresponding input axes in the inferred network, while the choice TDR axis can be identified to the $\boldsymbol{m}$ axis of the rank-one recurrent connectivity (sup. figs. 4,5). The projections of the original activity on the connectivity axes (full lines on fig. 5b) therefore shows how inferred connectivity explains the geometry of recorded data.

A closer look at the distribution of inferred connectivity weights provides more information on the neural mechanisms at play (sup. figs. 6,7). In contrast to the fits of task-optimized networks, we observed for both monkeys a clearly non-normal, heavy-tailed distribution of inferred connectivity parameters (Fisher's kurtosis between 4.9 and 62.7 for different connectivity parameters), echoing some past observations on biological synaptic weights [44, 3]. Separating with a clustering algorithm (GMM) large and small-weight neurons shows that the latter can be inactivated without affecting the task performance of the network in a significant way (sup. figs. 6,7), even though they represent a majority of neurons in the circuit (570/727 neurons in monkey A, 389/574 in monkey F). This suggests that a minority of neurons with large connectivity supports the computation performed by the network.

4 Discussion

In this paper, we introduced LINT, a new method for inferring a latent dynamical system from neural recordings based on the theory of low-rank RNNs. This method yields a mechanistic, interpretable and predictive model from neural trajectories, and bridges different levels of analysis, from state-space geometry to neural connectivity. After verifying the consistency of our method, we demonstrated its potential to extract the mechanisms used by "black-box" vanilla RNNs trained to perform cognitive tasks. In particular, we found that low-rank RNNs could reproduce the dynamics of vanilla RNNs trained to perform diverse tasks, across a range of hyperparameters. Moreover, the obtained lrRNNs provided interpretable low-dimensional projections of the fitted activity, as well as predictions for the effects of inactivations of specific populations of neurons. These predictions were verified in the original network, validating computational mechanisms derived from low-rank connectivity weights. Finally, we applied LINT to electrophysiological recordings in the prefrontal cortex of nonhuman primates performing a context-dependent decision-making task. LINT was able to infer rank-one networks that reproduced both neural activity and task performance, from which low-dimensional projections and interpretable connectivity could be extracted.

These results pave the way for a better understanding of how RNNs generate their outputs, which we believe is a very relevant endeavour given the increasing importance of RNNs and related classes of networks across sciences and engineering. Applications to reverse-engineer more advanced architectures like GRUs or LSTMs, which have already been shown to use low-dimensional trajectories [21, 19] are a potentially fruitful future area for reasearch.

It is worth noting that other statistical methods as mTDR [1] can provide a better fit to this data although it does not provide a dynamical system generating the trajectories. Our method is unique in that it bridges that standard state-space geometrical approach of methods like mTDR with a predictive model of the dynamics that methods like LFADS [26] can offer, as well as understanding at the neural connectivity level of how the exhibited dynamics emerge from recurrent interactions.

Methods for inferring latent dynamics from biological data often rely on non-obvious hypotheses about the inputs received by a circuit, and here we hypothesized that choice was recurrently generated within the recorded area. This raises the question of disentangling whether a latent variable is recurrently generated or externally fed to a circuit, which is a particularly important challenge in systems neuroscience, although methods have recently been proposed to address this issue [13, 38, 35]. LINT potentially provides a new approach for tackling this question, by comparing inferences made with different possible input structures, by exploiting the potential of single-trial simultaneous recordings which we have not used here, or by building multi-area models from wide-scale recordings that are increasingly available nowadays [42, 41, 45, 28]. On a longer time-scale, this work could contribute to the development of more powerful brain-machine interfaces.

References

[1] Mikio C Aoi, Valerio Mante, and Jonathan W Pillow. Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making. *Nature neuroscience*, 23(11):1410–1420, 2020.

[2] Manuel Beiran, Alexis Dubreuil, Adrian Valente, Francesca Mastrogiuseppe, and Srdjan Ostojic. Shaping dynamics with multiple populations in low-rank recurrent networks. *Neural computation*, 33(6):1572–1615, 2021.

[3] György Buzsáki and Kenji Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278, 2014.

[4] Warasinee Chaisangmongkon, Sruthi K Swaminathan, David J Freedman, and Xiao-Jing Wang. Computing by robust transience: how the fronto-parietal network performs sequential, category-based decisions. *Neuron*, 93(6):1504–1517, 2017.

[5] Zach Cohen, Brian DePasquale, Mikio C Aoi, and Jonathan W Pillow. Recurrent dynamics of prefrontal cortex during context-dependent decision-making. *bioRxiv*, 2020.

[6] Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. The role of population structure in computations through neural dynamics. *bioRxiv*, pages 2020–07, 2021.

[7] Lea Duncker, Gergo Bohner, Julien Boussard, and Maneesh Sahani. Learning interpretable continuous-time models of latent stochastic dynamical systems. In *International Conference on Machine Learning*, pages 1726–1734. PMLR, 2019.

[8] Lea Duncker and Maneesh Sahani. Dynamics on the manifold: Identifying computational dynamical activity from neural population recordings. *Current opinion in neurobiology*, 70:163–170, 2021.

[9] Daniel Durstewitz. A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. *PLoS computational biology*, 13(6):e1005542, 2017.

[10] Chris Eliasmith and Charles H Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2003.

[11] Arseny Finkelstein, Lorenzo Fontolan, Michael N Economo, Nuo Li, Sandro Romani, and Karel Svoboda. Attractor dynamics gate cortical information flow during decision-making. *Nature Neuroscience*, 24(6):843–850, 2021.

[12] Timo Flesch, Keno Juechems, Tsvetomira Dumbalska, Andrew Saxe, and Christopher Summerfield. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 2022.

[13] Aniruddh R Galgali, Maneesh Sahani, and Valerio Mante. Residual dynamics resolves recurrent contributions to neural computation. *bioRxiv*, 2021.

[14] Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *BioRxiv*, page 214262, 2017.

[15] Joshua Glaser, Matthew Whiteway, John P Cunningham, Liam Paninski, and Scott Linderman. Recurrent switching dynamical systems models for multiple interacting neural populations. *Advances in neural information processing systems*, 33:14867–14878, 2020.

[16] Christopher Langdon and Tatiana A Engel. Latent circuit inference from heterogeneous neural responses during cognitive tasks. *bioRxiv*, 2022.

[17] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922. PMLR, 2017.

[18] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. *Advances in neural information processing systems*, 24, 2011.

[19] Niru Maheswaranathan and David Sussillo. How recurrent networks implement contextual processing in sentiment analysis. *arXiv preprint arXiv:2004.08013*, 2020.

[20] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. *Advances in neural information processing systems*, 32, 2019.

[21] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. *Advances in neural information processing systems*, 32, 2019.

[22] Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84, 2013.

[23] Francesca Mastrogiuseppe and Srdjan Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.

[24] Marcel Nonnenmacher, Srinivas C Turaga, and Jakob H Macke. Extracting low-dimensional dynamics from multiple large-scale neural population recordings by learning to predict correlations. *Advances in Neural Information Processing Systems*, 30, 2017.

[25] Marius Pachitariu, Biljana Petreska, and Maneesh Sahani. Recurrent linear models of simultaneously-recorded neural populations. *Advances in neural information processing systems*, 26, 2013.

[26] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[28] Matthew G Perich, Charlotte Arlt, Sofia Soares, Megan E Young, Clayton P Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H Rudebeck, Christopher D Harvey, et al. Inferring brain-wide interactions using data-constrained recurrent neural network models. *bioRxiv*, pages 2020–12, 2021.

[29] Biljana Petreska, Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Dynamical segmentation of single trials from population neural data. *Advances in neural information processing systems*, 24, 2011.

[30] Eli Pollock and Mehrdad Jazayeri. Engineering recurrent neural networks from task-relevant manifolds and dynamics. *PLoS computational biology*, 16(8):e1008128, 2020.

[31] Kanaka Rajan, Christopher D Harvey, and David W Tank. Recurrent network models of sequence generation and memory. *Neuron*, 90(1):128–142, 2016.

[32] Evan D Remington, Devika Narain, Eghbal A Hosseini, and Mehrdad Jazayeri. Flexible sensorimotor computations through rapid reconfiguration of cortical dynamics. *Neuron*, 98(5):1005–1019, 2018.

[33] Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):140–149, 2021.

[34] Omid G Sani, Bijan Pesaran, and Maryam M Shanechi. Where is all the nonlinearity: flexible nonlinear modeling of behaviorally relevant neural dynamics using recurrent neural networks. *bioRxiv*, 2021.

[35] Britton A Sauerbrei, Jian-Zhong Guo, Jeremy D Cohen, Matteo Mischiati, Wendy Guo, Mayank Kabra, Nakul Verma, Brett Mensh, Kristin Branson, and Adam W Hantman. Cortical pattern generation during dexterous movement is input-driven. *Nature*, 577(7790):386–391, 2020.

[36] Shreya Saxena and John P Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.

[37] Rylan Schaeffer, Mikail Khona, Leenoy Meshulam, Ila Fiete, et al. Reverse-engineering recurrent neural network solutions to a hierarchical inference task for mice. *Advances in Neural Information Processing Systems*, 33:4584–4596, 2020.

[38] Marine Schimel, Ta-Chu Kao, Kristopher T Jensen, and Guillaume Hennequin. ilqr-vae: control-based learning of input-driven dynamics with applications to neural data. *bioRxiv*, 2021.

[39] Friedrich Schuessler, Alexis Dubreuil, Francesca Mastrogiuseppe, Srdjan Ostojic, and Omri Barak. Dynamics of random recurrent networks with correlated low-rank structure. *Physical Review Research*, 2(1):013111, 2020.

[40] Friedrich Schuessler, Francesca Mastrogiuseppe, Alexis Dubreuil, Srdjan Ostojic, and Omri Barak. The interplay between randomness and structure during learning in rnns. *Advances in neural information processing systems*, 33:13352–13362, 2020.

[41] João D Semedo, Amin Zandvakili, Christian K Machens, M Yu Byron, and Adam Kohn. Cortical areas interact through a communication subspace. *Neuron*, 102(1):249–259, 2019.

[42] Markus Siegel, Timothy J Buschman, and Earl K Miller. Cortical information flow during flexible sensorimotor decisions. *Science*, 348(6241):1352–1355, 2015.

[43] Jimmy Smith, Scott Linderman, and David Sussillo. Reverse engineering recurrent neural networks with jacobian switching linear dynamical systems. *Advances in Neural Information Processing Systems*, 34, 2021.

[44] Sen Song, Per Jesper Sjöström, Markus Reigl, Sacha Nelson, and Dmitri B Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS biology*, 3(3):e68, 2005.

[45] Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, 2019.

[46] David Sussillo. Neural circuits as computational dynamical systems. *Current opinion in neurobiology*, 25:156–163, 2014.

[47] David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.

[48] David Sussillo, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature neuroscience*, 18(7):1025–1033, 2015.

[49] Elia Turner, Kabir V Dabholkar, and Omri Barak. Charting and navigating the space of solutions for recurrent neural networks. *Advances in Neural Information Processing Systems*, 34:25320–25333, 2021.

[50] Adrian Valente, Srdjan Ostojic, and Jonathan Pillow. Probing the relationship between linear dynamical systems and low-rank recurrent neural network models. *arXiv preprint arXiv:2110.09804*, 2021.

[51] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

[52] Jing Wang, Devika Narain, Eghbal A Hosseini, and Mehrdad Jazayeri. Flexible timing by temporal scaling of cortical responses. *Nature neuroscience*, 21(1):102–110, 2018.

[53] Yang Xie, Peiyao Hu, Junru Li, Jingwen Chen, Weibin Song, Xiao-Jing Wang, Tianming Yang, Stanislas Dehaene, Shiming Tang, Bin Min, et al. Geometry of sequence working memory in macaque prefrontal cortex. *Science*, 375(6581):632–639, 2022.

[54] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

[55] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: A primer. *Neuron*, 107(6):1048–1070, 2020.

[56] Byron M Yu, Afsheen Afshar, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Extracting dynamical structure embedded in neural activity. *Advances in neural information processing systems*, 18, 2005.

[57] Rafael Yuste. From the neuron doctrine to neural networks. *Nature reviews neuroscience*, 16(8):487–497, 2015.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [N/A]
- Did you include the license to the code and datasets? [No]
- Did you include the license to the code and datasets? [Yes]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The three contributions exposed in the introduction are detailed in the results section. The specific scope is detailed in the introduction and the discussion.
   (b) Did you describe the limitations of your work? [Yes] Limitations are exposed in the discussion, notably with respect to hypotheses made about the inputs.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See discussion.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] No novel theoretical results are exposed here, the theoretical results on low-rank RNNs already exist in the literature, notably in [2]
   (b) Did you include complete proofs of all theoretical results? [N/A] Same answer.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Code will be made available upon publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the supplemental material, Appendix C
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Figure 2 contains whole distributions for the fitting experiments, standard deviation for the subsampling experiment, and the material exposed in section 3.3 and figure 5 has been fully reproduced on data from a second macaque subject. Moreover, additional hyperparameters are explored in sup. fig. 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the supplemental material, Appendix C
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Providers of the dataset [22] cited in section 3.3
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] Trained network weights will be provided upon publication.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No human participants.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No human participants not offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No human participants.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No human participants.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No human participants.