

# Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor components analysis

Alex H. Williams<sup>1,\*</sup>, Tony Hyun Kim<sup>2</sup>, Forea Wang<sup>1</sup>, Saurabh Vyas<sup>2,3</sup>, Stephen I. Ryu<sup>2,11</sup>, Krishna V. Shenoy<sup>2,3,6,7,8,9</sup>, Mark Schnitzer<sup>4,5,7,9,10</sup>, Tamara G. Kolda<sup>12</sup>, Surya Ganguli<sup>4,6,7,8,†</sup>

<sup>1</sup>Neurosciences Graduate Program, <sup>2</sup>Electrical Engineering Department, <sup>3</sup>Bioengineering Department, <sup>4</sup>Applied Physics Department, <sup>5</sup>Biology Department, <sup>6</sup>Neurobiology Department, <sup>7</sup>Bio-X Program, <sup>8</sup>Stanford Neurosciences Institute, <sup>9</sup>Howard Hughes Medical Institute, <sup>10</sup>CNC Program, Stanford University, Stanford, CA 94305, USA.

<sup>11</sup>Department of Neurosurgery, Palo Alto Medical Foundation, Palo Alto, CA 94301, USA.

<sup>12</sup>Sandia National Laboratories, Livermore, CA 94551, USA

\*ahwillia@stanford.edu, †sganguli@stanford.edu

## Abstract

Perceptions, thoughts and actions unfold over millisecond timescales, while learned behaviors can require many days to mature. While recent experimental advances enable large-scale and long-term neural recordings with high temporal fidelity, it remains a formidable challenge to extract unbiased and interpretable descriptions of how rapid single-trial circuit dynamics change slowly over many trials to mediate learning. We demonstrate a simple tensor components analysis (TCA) can meet this challenge by extracting three interconnected low dimensional descriptions of neural data: *neuron factors*, reflecting cell assemblies; *temporal factors*, reflecting rapid circuit dynamics mediating perceptions, thoughts, and actions within each trial; and *trial factors*, describing both long-term learning and trial-to-trial changes in cognitive state. We demonstrate the broad applicability of TCA by revealing insights into diverse datasets derived from artificial neural networks, large-scale calcium imaging of rodent prefrontal cortex during maze navigation, and multielectrode recordings of macaque motor cortex during brain machine interface learning.

## 1 Introduction

Two of the most challenging obstacles to understanding neural circuits are their diversity of dynamical timescales and the large number of neurons that contribute to their function. For instance, circuit dynamics mediating sensory perception, decision-making, attentional shifting, motor control, and higher cognition unfold over hundreds of milliseconds, while slower processes like motivation, long-term planning, and learning vary slowly, sometimes taking days or weeks to fully manifest [1–3]. Moreover, every execution of a behavior can involve the coordinated activity of extremely large neural populations, often distributed across multiple brain regions.

Recent experimental advances enable us to monitor all aspects of this biological complexity by recording large numbers of neurons [4–7] at high temporal precision [8] over long durations [9–11]. The resulting datasets can contain thousands of neural activity traces collected over thousands of behavioral trials. The genesis of such complex, large scale datasets now present a major data-analytic challenge to the field of neuroscience. Namely, how can we develop general purpose algorithms to extract from such complex data, simple and interpretable descriptions of collective circuit dynamics that underly not only rapid sensory, motor and cognitive acts, but also describe slower signatures of long-term planning and learning? Moreover, how can these algorithms operate in an unsupervised manner, to enable the discovery of novel and unexpected cognitive states that can vary on a trial by trial basis?

27 Neuroscientists have often turned to unbiased dimensionality reduction methods to understand these complex  
28 datasets [12, 13]. However, commonly used methods focus on reducing the complexity of fast, within-trial firing rate  
29 dynamics instead of extracting interpretable slow, across-trial structure. A common approach is to average neural  
30 activity across trials [13–15], thereby precluding the possibility of understanding of how cognition and behavior  
31 change on a trial by trial basis. More recent methods, including Gaussian Process Factor Analysis (GPFA) [16] and  
32 latent dynamical system models [17, 18], identify low-dimensional firing rate trajectories *within* each trial, but do  
33 not reduce the dimensionality *across* trials by extracting analogous low-dimensional trajectories over trials. Other  
34 works have separately focused on trial-to-trial variability in neural responses [19–22], and long-term trends across  
35 many trials [1, 3, 23–26], but without an explicit focus on obtaining simple low-dimensional descriptions. Thus, while  
36 current experimental data can simultaneously capture neural dynamics underlying both fast cognitive processes as  
37 well as slower learning processes, we lack general-purpose methods for extracting unbiased descriptions of both fast  
38 cognition and slower learning.

39 The most common and fundamental method for dimensionality reduction of neural data is Principal Components  
40 Analysis (PCA) [12, 13]. Here, we explore a simple extension of PCA that enables multi-timescale dimensionality  
41 reduction of neural dynamics both within trials and across trials. The key idea is to organize neural firing rates  
42 into a third-order tensor (i.e., a three-dimensional data table) with three axes corresponding to individual neurons  
43 (index 1), time within trial (index 2), and trial number (index 3). We then fit a tensor decomposition model  
44 (CANDECOMP/PARAFAC) [27, 28] to identify a set of low-dimensional components describing variability along  
45 each of these three axes. We refer to this procedure as Tensor Components Analysis (TCA).

46 We demonstrate that TCA yields insightful descriptions of a variety of neural datasets. In particular, it enables us  
47 to move beyond trial averaging by simultaneously identifying separate low-dimensional features for rapid within-trial  
48 neural dynamics and slower across-trial neural dynamics. Furthermore, as described below, TCA possesses a set  
49 of favorable theoretical properties that translate into significant interpretational advantages when applied to neural  
50 data. In particular, the components returned by TCA are often unique [29], unlike PCA which requires a biologically  
51 unrealistic orthogonality constraint to yield unique components. Because of the uniqueness of TCA, it achieves a  
52 demixing of neural data in which *individual* components are often in one-to-one correspondence with biologically  
53 interpretable variables. For example, as we see below, in diverse datasets, individual components correspond to  
54 sensations, decisions, actions, rewards and performance.

55 Below, after introducing the method, we show that TCA is equivalent to a form of multi-dimensional gain control  
56 and so can be interpreted as a generalization of a well-studied model of cortical function [30, 31]. We then give three  
57 examples of its utility. First, in an artificial neural circuit trained to solve the well-studied motion discrimination  
58 task [32], we show that TCA yields a simple one-dimensional description of the evolving connectivity and dynamics  
59 of the circuit during learning. Next, in a maze navigation task in rodents, we show that TCA can recover several  
60 aspects of trial structure and behavior, including perceptions, decisions, rewards, and errors, in an unsupervised,  
61 data-driven fashion. Finally, for a monkey operating a brain machine interface (BMI), we show that TCA extracts a  
62 simple view of motor learning when the BMI is altered to change the relationship between neural activity and motor  
63 action.

64 Thus, this work introduces a simple and broadly applicable method for identifying interpretable structure in  
65 multi-trial neural data, thereby providing a way to attack two of the most challenging aspects of modern large-  
66 scale neural recordings: their multi-timescale nature, and their high dimensionality. While TCA is a general-purpose  
67 method [33], we provide specialized code and step-by-step instructions for applying TCA to neural data, and describe  
68 how to interpret the outcomes of TCA within the context of systems neuroscience.

## 2 Results

### 2.1 Discovering multi-timescale structure through TCA

Before describing TCA, we first review the application of PCA for analyzing large-scale recordings. Consider a recording of  $N$  neurons over  $K$  experimental trials. We assume neural activity is recorded at  $T$  timepoints within each trial, but trials of variable duration can be aligned or temporally warped to accommodate this constraint (see, e.g., [34]). This dataset is naturally represented as an  $N \times T \times K$  array of firing rates, which is known in mathematics as a third-order tensor. Each element in this tensor,  $x_{ntk}$ , denotes the firing rate of neuron  $n$  at time  $t$  within trial  $k$ . Here, the index  $n$  ranges from 1 to  $N$ ,  $t$  ranges from 1 to  $T$ , and  $k$  ranges from 1 to  $K$ .

These datasets are very challenging to interpret in their raw format. Even nominally identical trials (e.g., neural responses elicited by repeats of an identical sensory stimulus) can exhibit significant trial-to-trial variability [22]. Under the assumption that such variability is simply irrelevant noise, a common method to simplify the table is to average across trials, obtaining a two dimensional table, or matrix,  $\bar{x}_{nt}$ , which holds the trial-averaged neural firing rates for every neuron  $n$  and timepoint  $t$  (fig. 1a). Even such a matrix can be difficult to understand in large-scale experiments containing many neurons and rich temporal dynamics. PCA summarizes these data by performing a decomposition into  $R$  components such that

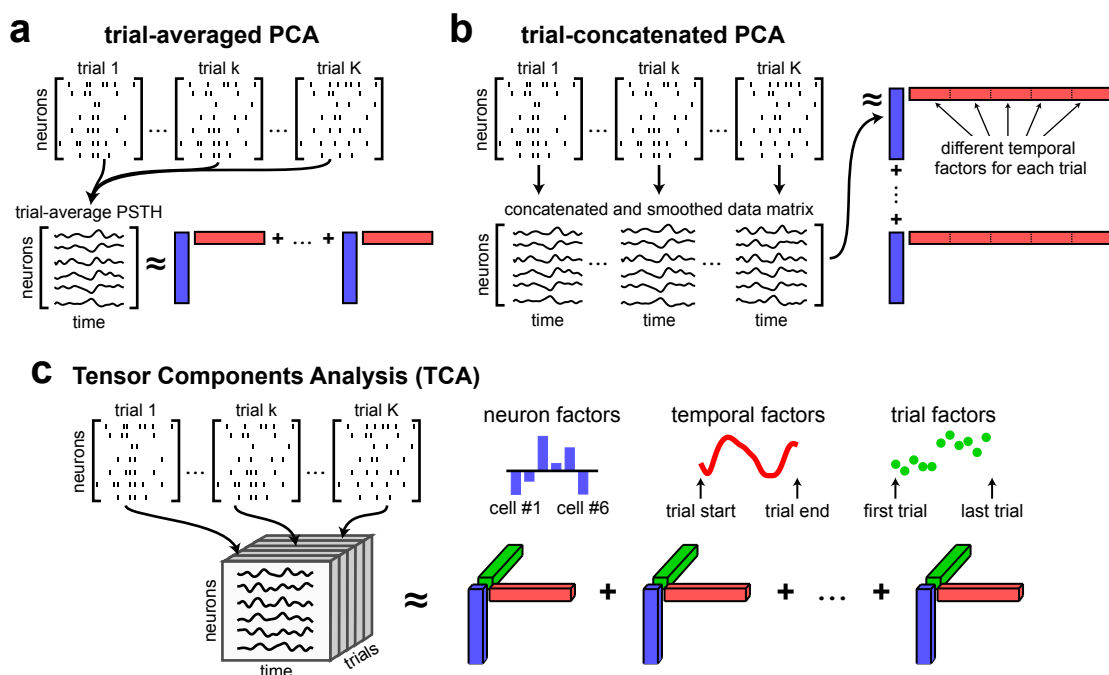
$$\bar{x}_{nt} \approx \sum_{r=1}^R w_n^r b_t^r. \quad (1)$$

This decomposition projects the high-dimensional data (with  $N$  or  $T$  dimensions) into a low-dimensional space (with  $R$  dimensions). Each component, indexed by  $r$ , contains a coefficient across neurons,  $w_n^r$ , and a coefficient across timepoints,  $b_t^r$ . These terms can be collected into vectors:  $\mathbf{w}^r$ , of length  $N$ , which we call *neuron factors* (blue vectors in fig. 1), and  $\mathbf{b}^r$ , of length  $T$ , which we call *temporal factors* (red vectors in fig. 1). The neuron factors can be thought of as an ensemble of cells that exhibit correlated firing. The temporal factors can be thought of as a trial-averaged dynamical activity pattern for each ensemble. Overall, this *trial-averaged PCA* procedure reduces the original  $N \times T \times K$  datapoints into  $R(N + T)$  values, yielding a compact, and often insightful summary of the trial-averaged data [12, 13].

However, trial-averaging is motivated by the assumption that trial-to-trial variability is irrelevant noise, which is often at odds with our understanding of neural circuits and questions of experimental interest. For instance, even under repeated sensory stimuli, trial-to-trial variability may reflect fluctuations in interesting cognitive states, like attention or arousal [20, 21]. Also, under situations in which animals are learning a task, there will be systematic changes in neural dynamics over many trials, which would be rendered invisible by trial averaging. Intriguingly, as the field moves to study more complex tasks, we may find completely unexpected structured variability across trials, corresponding to different internal brain states on different trials. Ideally, we would like unbiased, data-driven methods to extract such dynamics simply by analyzing the data tensor.

One approach to retain the variability across trials is to concatenate multiple trials rather than averaging, thereby transforming the data tensor into an  $N \times TK$  matrix, and then applying PCA to this matrix (fig. 1b). This approach, which we call *trial-concatenated PCA*, is similar to Gaussian Process Factor Analysis (GPFA) [16], another specialized technique for neural data analysis. In trial-concatenated PCA, the  $R$  temporal factors are of length  $TK$  and do not enforce any commonality across trials. It therefore achieves a less significant reduction in the complexity of the data: the  $NTK$  numbers in the original data tensor are only reduced to  $R(N + TK)$  numbers, which can be cumbersome in experiments consisting of thousands of trials.

Our proposal is to directly deal with neural data in its natural third-order tensor format by performing a dimensionality reduction of this tensor (fig. 1c), rather than first converting it to a matrix. This tensor components



**Fig 1. Tensor representation of trial-structured neural data.** (a) Schematic of trial-averaged PCA for spiking data. The raw data is represented as a sequence of  $N \times T$  matrices (top). These matrices are averaged across trials to build a matrix representation of neural firing rates. PCA approximates the trial-averaged matrix as a sum of outer products of vectors (see eq. (1)). Each outer product contains a neuron factor (blue rectangles) and a temporal factor (red rectangles). (b) Schematic of trial-concatenated PCA for spiking data. Raw data are temporally smoothed by a Gaussian filter to estimate neural firing rates before concatenating all trials along the time axis. Applying PCA produces a separate set of temporal factors for each trial (subsets of the red vectors). (c) Schematic of TCA. Raw data are smoothed and collected into a third order tensor with dimensions  $N \times T \times K$ . TCA approximates the data as a sum of outer products of three vectors, producing a third set of low-dimensional factors (trial factors, green vectors) that describe how activity changes across trials.

109 analysis (TCA) method then yields the  $R$ -component decomposition [27, 28, 33]

$$x_{ntk} \approx \sum_{r=1}^R w_n^r b_t^r a_k^r. \quad (2)$$

110 In analogy to PCA, we can think of  $\mathbf{w}^r$  as a prototypical firing rate pattern across neurons, and we can think of  
 111  $\mathbf{b}^r$  as a temporal basis function across time within trials. These neuron factors and temporal factors constitute  
 112 structure that is common across all trials. We call the third set of factors,  $\mathbf{a}^r$ , *trial factors* (green vectors in fig. 1),  
 113 which are new to TCA and not present in PCA. The trial factors can be thought of as trial-specific amplitudes for  
 114 the within-trial activity patterns identified by the neuron and temporal factors. Thus, in TCA, the trial-to-trial  
 115 fluctuations in neural activity are also embedded in  $R$ -dimensional space. TCA achieves a dramatic reduction of the  
 116 original data tensor, reducing  $NTK$  datapoints to  $R(N+T+K)$  values, while still capturing trial-to-trial variability.

117 A subtle, but critical, difference between PCA and TCA is the uniqueness of the identified factors. In order to  
 118 obtain unique factors, PCA constrains both the neuron and temporal factors to be orthogonal sets of vectors. This  
 119 assumption is motivated by mathematical convenience rather than scientific principles. In real biological circuits,  
 120 cell ensembles may overlap and temporal firing patterns may be correlated, producing non-orthogonal structure that  
 121 is missed by PCA. In contrast, the TCA model often has a unique solution without further assumptions [29]. As we  
 122 demonstrate below, TCA tends to extract non-orthogonal features of data that are more interpretable and meaningful  
 123 than those extracted by PCA. In particular, we will see that TCA not only performs dimensionality reduction, but also  
 124 demixing, by learning individual components that are in one-to-one correspondence with biologically interpretable  
 125 variables.

## 2.2 TCA as a generalized cortical gain control model.

Although TCA was originally developed as a statistical method [33], here we show that it concretely relates to a prominent theory of neural computation when applied to multi-trial datasets. In particular, performing TCA on neural data is equivalent to fitting a gain-modulated linear network model. In this network,  $N$  observed neurons (light gray circles, fig. 2a) are driven by a much smaller number of  $R$  unobserved, or latent, inputs (dark gray circles, fig. 2a) that have a fixed temporal profile but have varying amplitudes for each trial. The neuron factors of TCA,  $w_n^r$  in eq. (2), correspond to the synaptic *weights* from each latent input  $r$  to each neuron  $n$ . The temporal factors of TCA,  $b_t^r$ , correspond to *basis functions* or the activity of input  $r$  at time  $t$ . Finally, the trial factors of TCA,  $a_k^r$ , correspond to *amplitudes*, or gain, of latent input  $r$  on trial  $k$ . Such trial-to-trial fluctuations in amplitude have been observed in a variety of sensory systems [22, 35–37], and are believed to be an important and ubiquitous feature of cortical circuits [30, 31]. Furthermore, plausible cellular mechanisms for gain modulation have been examined by a number of experimental and computational studies [38–41]. The TCA model can be viewed as a higher,  $R$ -dimensional generalization of such theories. By allowing an  $R$ -dimensional space of possible gain modulations to different temporal factors, TCA can capture a rich diversity of changing multi-neuronal activity patterns across trials.

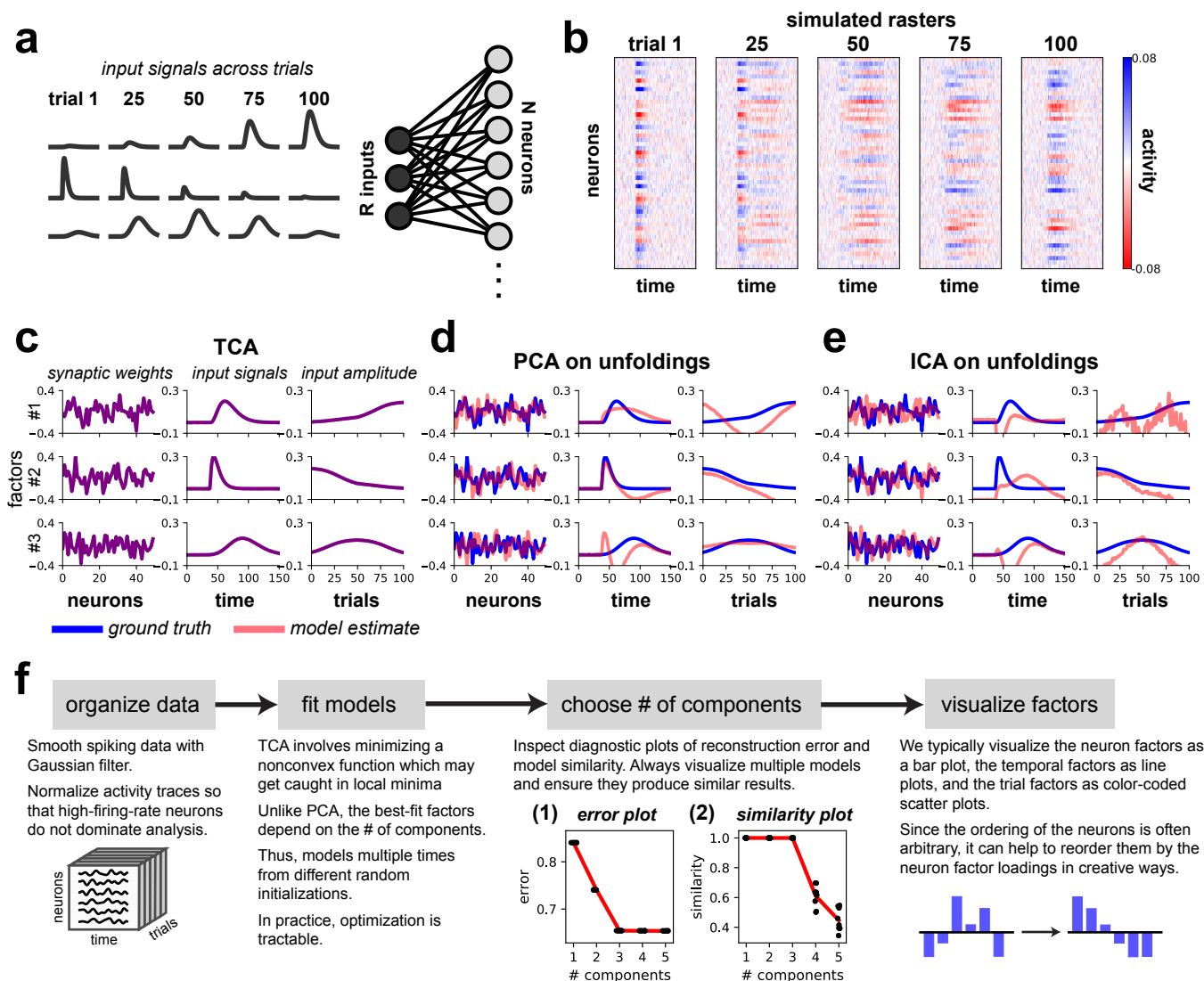
An important feature of TCA is that these network parameters can often be unambiguously identified from simulated data alone, due to the previously mentioned uniqueness property of TCA [29]. We confirmed this in a simple simulation with three latent inputs/components. In this example, the first component grows in amplitude across trials, the second component shrinks, and the third component grows and then shrinks in amplitude (fig. 2a). This model generates rich simulated population activity patterns across neurons, time, and trials as shown in (fig. 2b), where we have added Gaussian white noise to demonstrate the robustness of the method. When applied to noisy multi-neuronal traces, TCA with  $R = 3$  components precisely extracted the network parameters (fig. 2c).

In contrast, neither PCA nor independent components analysis (ICA) [42] can recover the network parameters, as demonstrated in fig. 2d and fig. 2e respectively. Unlike TCA, both PCA and ICA are fundamentally matrix, not tensor, decomposition methods. Therefore they cannot be applied directly to the data tensor, but instead must be applied to three different matrices obtained by *tensor unfolding* (fig. 2 supp. 1; [33]). In essence, the unfolding procedure generalizes the trial-concatenated representation of the data tensor (fig. 1b) to allow concatenation across neurons or timepoints. This unfolding destroys natural structure across neurons, time, and trials in the third-order data tensor, thereby precluding the possibility of finding the ground truth synaptic weights, temporal basis functions, and trial amplitudes that actually generated observed neural activity patterns.

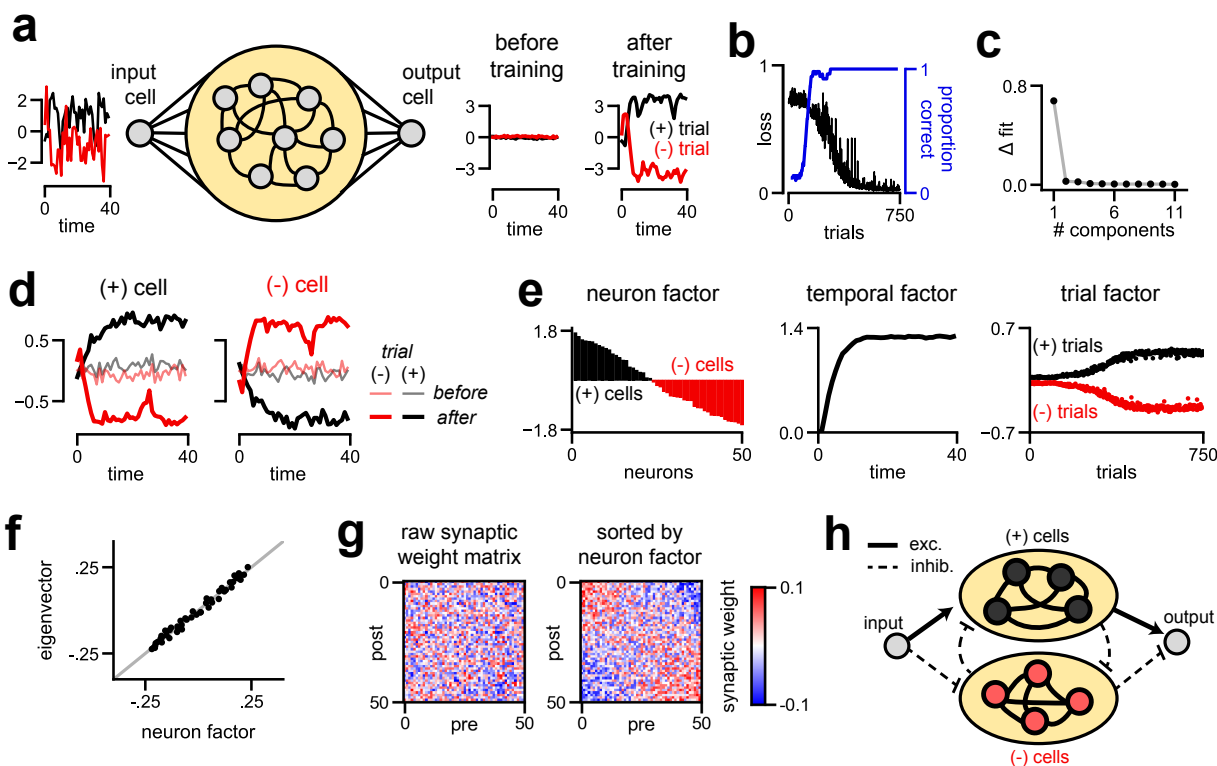
## 2.3 Choosing the number of components.

A schematic view of the process of applying TCA to neural data is shown in fig. 2f (see *Methods* for more details). As in PCA and many other dimensionality reduction methods, a critical issue is the choice of the number of components, or dimensions  $R$ . We employ two methods to inform this choice. First, we inspect an *error plot* (fig. 2f, inset), which displays the model reconstruction error as a function of the number of components  $R$ . We normalize the reconstruction error to range between zero and one as described in section 4.5.1. This provides a metric analogous to the fraction of unexplained variance, which is used in PCA. As in PCA, a kink or leveling out in this plot indicates a point of diminishing returns for including more components. Unlike PCA, we run the optimization algorithm underlying TCA at each value of  $R$  multiple times from random initial conditions, and plot the normalized reconstruction error for all such models. Such repeated optimization runs enable us to check whether some runs converge to suboptimal solutions with high reconstruction error. As shown in (fig. 2f, inset), the error plot reveals that all runs at fixed  $R$  yield the same error, and moreover, the kink in the plot unambiguously reveals  $R = 3$  as the true number of components in the generated data, in agreement with the ground truth.

A second method to assess the number of components involves generating a *similarity plot* (fig. 2f, inset), which



**Fig 2. TCA precisely recovers the parameters of a linear model network.** (a) Schematic of model network. Three input signals (dark gray) were delivered to a 1-layer, linear neural network with  $N = 50$  neurons (light gray). Gaussian noise was added to the output units. (b) Simulated activity of all neurons on example trials. (c) The factors identified by 3-component TCA precisely match the network parameters. (d-e) Applying PCA (in d) or ICA (in e) to each of the tensor unfoldings does not recover the network parameters. (f) Analysis pipeline for TCA. (f, inset 1) Error plots showing normalized reconstruction error (vertical axis) for TCA models with different numbers of components (horizontal axis). The red line tracks the minimum error (i.e., best-fit model). Each black dot denotes a model fit from different initial parameters. All models fit from different initializations had essentially identical performance. Reconstruction error did not improve after more than 3 components were included. (f, inset 2) Similarity plot showing similarity score (eq. (12); vertical axis) for TCA models with different numbers of components (horizontal axis). Similarity for each model (black dot) is computed with respect to the best-fit model with the same number of components. The red line tracks the mean similarity as a function of the number of components. Adding more than 3 components caused models to be less reliably identified.



**Fig 3. Unsupervised discovery of low-dimensional learning dynamics and mechanism in a model RNN.** (a) Model schematic. A noisy input signal is broadcast to a recurrent population of neurons with all-to-all connectivity (yellow oval). On (+)-trials the input is net positive (black traces), while on (-)-trials the input is net negative (red traces). The network is trained to output the sign of the input signal with a large magnitude. (b) Learning curve for the model, showing the objective value on each trial over learning. (c) Scree plot showing the improvement in normalized reconstruction error as more components are added to the model. (d) An example (+)-cell and (-)-cell before and after training on both trial types. Black traces indicate (+)-trials, and red traces indicate (-)-trials. (e) Factors discovered by a one-component TCA applied simulated neuron activity over training. The neuron factor identifies (+)-cells (black bars) and (-)-cells (red bars), which have opposing correlations with the input signal. These two populations naturally exist in a randomly initialized network (trial 0), but become separated after during training, as described by the trial factor. (f) The neuron factor identified by TCA closely matches the principal eigenvector of the synaptic connectivity matrix post-learning. (g) The recurrent synaptic connectivity matrix post-learning. Resorting the neurons by their order in the neuron factor in (e) uncovers competitive connectivity between the (+)-cells and (-)-cells. (h) Simplified diagram of the learned mechanism for this network.

170 displays how sensitive the recovered factors are to the initialization of the optimization procedure underlying TCA.  
 171 For each component, we compute the similarity of all fitted models to the model with lowest reconstruction error by  
 172 a similarity score bounded between zero (orthogonal factors) and one (identical factors). See section 4.5.1 for more  
 173 details. Adding more components to the model can produce lower similarity scores, which complicates exploratory  
 174 analysis since multiple low-dimensional descriptions may be consistent with the data. Like the error plot, the  
 175 similarity plot unambiguously reveals  $R = 3$  as the correct number of components, as decompositions with  $R > 3$  are  
 176 less consistent with each other (fig. 2f, inset). Notably, all models with  $R = 3$  converge to *identical* components (up  
 177 to permutations and re-scalings of factors), suggesting that only a single low-dimensional description, corresponding  
 178 to the ground truth network parameters, achieves minimal reconstruction error. TCA consistently identifies this  
 179 solution across multiple optimization runs.

## 2.4 TCA elucidates learning dynamics, circuit connectivity and computational mechanism in a nonlinear network

While TCA corresponds to a linear gain-modulated network, it can nevertheless reveal insights into the operation of more complex nonlinear networks, analogous to how PCA, a linear dimensionality reduction technique, allows visualization of low-dimensional nonlinear neural trajectories [12, 13]. We examine the application of TCA to nonlinear recurrent neural networks (RNNs), a powerful class of models that can learn to approximate any dynamical system [43]. RNNs have achieved success both in machine learning applications [44] and in modeling neural dynamics and behavior [45–48]. However, such models are so complex that they are often viewed as “black boxes.” Statistical methods that shed light on the function of RNNs and other complex computational models are therefore of great interest [49, 50]. Notably, while previous studies have focused on reverse-engineering RNNs with static parameters [51], few works have attempted to characterize how computational mechanisms in RNNs emerge over the process of learning, or optimization, of network parameters. Here we show TCA can naturally yield such a characterization for an RNN that learns to solve a simple sensory discrimination task, analogous to the well-known random dots direction-discrimination task [32].

Specifically, we trained an RNN with 50 neurons to estimate whether a noisy input signal had net positive or negative activity over a short time window, and indicate this estimate by exciting or inhibiting an output neuron (fig. 3a). We call trials with a net positive input *(+)-trials* and trials with a net negative input *(-)-trials*. The average amplitude of the input can be viewed as a proxy for the average motion energy of moving dots along a directional axis, with +/- corresponding to left/right, for example. The synaptic weights were updated by a simple gradient descent rule using backpropagation through time on a logistic loss function [52]. Within 750 trials the network performed the task with virtually 100% accuracy (fig. 3b).

Remarkably, TCA needed only a single component to capture *both* the within-trial multi-neuronal circuit dynamics of decision making *and* the across-trial dynamics of learning. Adding more components led to negligible improvements in reconstruction error (fig. 3c). A single-component TCA model makes two strong predictions about this dataset. First, within all trials, the time course of evidence integration is shared across all neurons and is not substantially effected by training. Second, across trials, the amplitude of single cell responses are simply scaled by a common factor during learning. In essence, prior to learning, all cells have some small, random preference for one of the two input types, and learning corresponds to simply amplifying these initial tunings. We visually confirmed this prediction by examining single trial responses of individual cells. We observed two cell types within this model network: *(+)-cells* which were excited on *(+)-trials* and inhibited on *(-)-trials* (fig. 3d, left), and *(-)-cells* which were excited on *(-)-trials* and inhibited on *(+)-trials* (fig. 3d, right). The response amplitudes of both cell types magnified over learning, and typically the initial tuning (pale lines) aligned with the final tuning (dark lines). These trends are verified across the full population of cells in fig. 3 Supplement 1a-b.

We then visualized the three factors of the single-component TCA model (fig. 3e). We sorted the cells by their weight in the neuron factor, and plotted this factor,  $\mathbf{w}^1$ , as a bar plot (fig. 3e; left). Neurons with a positive weight are precisely the *(+)-cells* (black bars) defined earlier, while neurons with a negative weight were *(-)-cells* (red bars). While it is conceptually helpful to discretely categorize cells, the neuron factor illustrates that the model cells actually fall along a continuous spectrum rather than two discrete groups. The temporal basis function extracted by TCA,  $\mathbf{b}^1$ , reveals a common dynamical pattern within all trials corresponding to integration to a bound (fig. 3e; middle), similar to the example cells shown in Figure 3d. Finally, the trial factor of TCA,  $\mathbf{a}^1$ , recovered two important aspects of the neural dynamics (fig. 3e; right). First, the trial amplitude is positive for *(+)-trials* (black points) and negative for *(-)-trials* (red points), thereby providing a direct readout of the input on each trial. Second, over the course of learning, these two trial types become more separated, reflecting stronger internal responses to the stimulus and a more confident prediction at the output neuron. Intriguingly, this analysis reveals that the process of learning simply involves monotonically amplifying small but random initial selectivity for the +/- stimulus into a strong final



225 selectivity.

226 This analysis also sheds light on the synaptic connectivity and computational mechanism of the RNN. To perform  
227 the task, the network must integrate evidence for the sign of the noisy stimulus over time. Linear model networks  
228 achieve this when the synaptic weight matrix has a single eigenvalue equal to one, and the remaining eigenvalues close  
229 to zero [53]. The eigenvector associated with this eigenvalue corresponds to a pattern of activity across neurons along  
230 which the network integrates evidence. The nonlinear RNN converged to a similar solution where one eigenvalue  
231 of the connectivity matrix is close to one, and the remaining eigenvalues are smaller and correspond to random  
232 noise in the synaptic connections (fig. 3, supp. 1a). Although the TCA model was fit only to the activity of the  
233 network, the prototypical firing pattern extracted by TCA in (fig. 3e; left) closely matched the principal eigenvector  
234 of the network’s synaptic connectivity matrix (fig. 3f). Thus, TCA extracted an important aspect of the network’s  
235 connectome from the raw simulated activity.

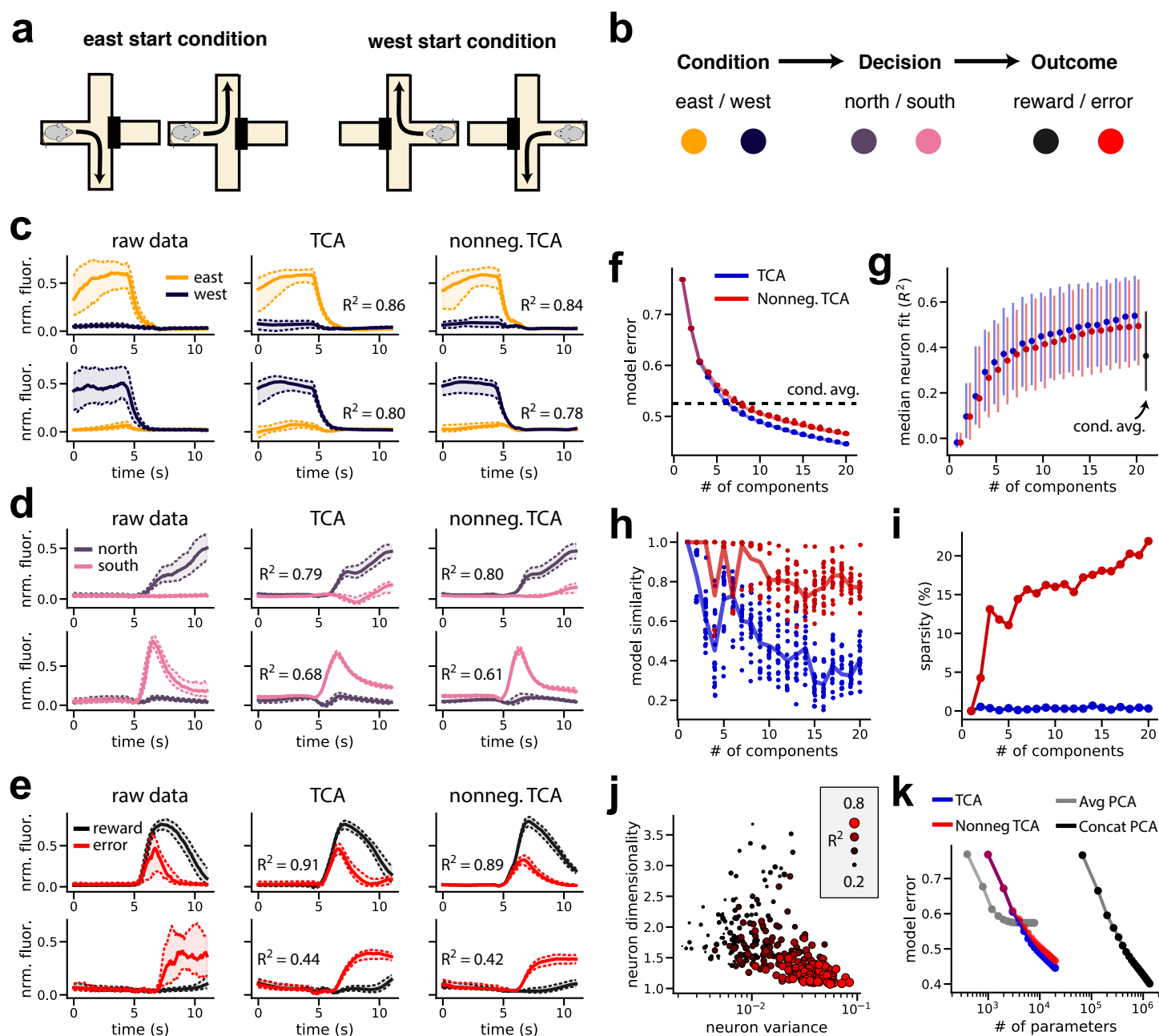
236 The neuron factor can also be used to better visualize and interpret the weight matrix itself. Since the original  
237 order of the neurons is arbitrary, the raw synaptic connectivity matrix appears to be unstructured noise (fig. 3g, left).  
238 However, re-sorting the neurons based on the neuron factor extracted by TCA, reveals a competitive connectivity  
239 between the (+)-cells and (-)-cells (fig. 3g, right). Specifically, neurons tend to send excitatory connections to cells  
240 in their same class, and inhibitory connections to cells of the opposite class. We also observed positive correlations  
241 between the neuron factor and the input and output synaptic weights of the network (fig. 3 supp. 1b-c). Taken  
242 together, these results provide a simple account of network function in which the input signal excites (+)-cells and  
243 inhibits (-)-cells on (+)-trials, and vice versa on (-)-trials. The two cell populations then compete for dominance in a  
244 winner-take-all fashion. Finally, the decision of the network is broadcast to the output cell by excitatory projections  
245 from the (+)-cells and inhibitory projections from the (-)-cells (fig. 3h).

246 In summary, TCA extracts a simple one-dimensional description of the activity of all neurons over all trials  
247 in this nonlinear network. Moreover, each of the three factors extracted by TCA have a simple neurobiological  
248 interpretation: the neuron factor  $\mathbf{w}^1$  reveals a continuum of neurons interpolating between two cell assemblies, the  
249 temporal factor  $\mathbf{b}^1$  describes the dominant neural activity underlying decision making, namely integration to a bound,  
250 and the trial amplitudes  $\mathbf{a}^1$  reflect the trial-by-trial decisions of the network, as well as the long term amplification  
251 of stimulus selectivity underlying learning. Finally, even though the low-dimensional TCA factors were found in an  
252 unsupervised fashion from the raw neural activity, they provide direct insights into the synaptic connectivity and  
253 emergent computational mechanism underlying the network’s ability to learn and decide.

## 254 2.5 TCA compactly represents prefrontal activity during spatial navigation

255 Given the demonstrated success of TCA on an artificial nonlinear network, we next examined the performance of TCA  
256 on large-scale neurobiological datasets. We first examined the activity of cortical cells in mice performing a spatial  
257 navigation task with variable reward contingencies. A miniature microendoscope [54] was used to record fluorescence  
258 in GCaMP6m-expressing excitatory neurons in the medial prefrontal cortex while mice navigated a four-armed maze.  
259 Mice began each trial in either the east or west arm and chose to visit either the north or south arm, at which point  
260 a water reward was either dispensed or withheld (fig. 4a-b). We examined a dataset from a mouse containing  $N = 282$   
261 neurons recorded at  $T = 111$  timepoints (at 10 Hz) on  $K = 600$  behavioral trials, collected over a five day period.  
262 The rewarded navigational rules were switched periodically, prompting the mouse to explore different actions from  
263 each starting arm. Fluorescence traces for each neuron were shifted and scaled to range between zero and one in  
264 each session, and organized into a  $N \times T \times K$  tensor.

265 Neural firing in prefrontal cortical areas have previously been found to encode task variables, outcomes, value  
266 judgments, and cognitive strategies [25, 55–59]. We observed that many neurons selectively correlated with individual  
267 task variables on each trial: the initial arm of the maze (fig. 4c), the final arm (fig. 4d), and whether the mouse  
268 received a reward (fig. 4e). Notably, many of these neurons — particularly those with strong and robust coding



**Fig 4. Reconstruction of single-cell activity during spatial navigation by unconstrained and nonnegative TCA.** (a) All four possible combinations of starting and ending position on a trial. (b) Color scheme for three binary task variables (start location, end location, and reward). Each trial involves a sequential selection of these three variables. (c) Median fluorescence of example neurons that encode the starting location. Dashed lines denote upper and lower quartiles of the fluorescence. (d-e) Same as (c) but showing neurons that encode the ending location and the presence/absence of water reward. (f) Scree plot showing normalized reconstruction error for unconstrained (blue) and nonnegative (red) TCA, and the condition-averaged baseline model (black dashed line). Models were optimized from multiple initial parameters; each dot corresponds to a different optimization run. (g) Median coefficient of determination ( $R^2$ ) for neurons as a function of the number of model components for unconstrained TCA (blue), nonnegative TCA (red), and the condition-averaged baseline (black). Dots show the median  $R^2$  and the extent of the lines shows the first and third quartiles of the distribution. (h) Model similarity (section 4.5.1) as a function of model components for unconstrained (blue) and nonnegative (red) TCA. Each dot shows the similarity of a single optimization run compared to the best-fit model within each category. (i) Sparsity (proportion of zero elements) in the neuron factors of unconstrained (blue) and nonnegative decompositions. For each decomposition type, only the best-fit model is shown. (j) Neuron dimensionality (section 4.5.2) plotted against variance in activity. The size and color of the dots represent the  $R^2$  of a nonnegative decomposition with 15 components. (k) Normalized reconstruction error plotted against number of free parameters for trial-averaged PCA, trial-concatenated PCA, and TCA.

269 properties — varied most strongly in amplitude across trials, suggesting that low-dimensional gain modulation is  
270 a reasonable model for these data. A TCA model with 15 components accurately modeled the activity of these  
271 individual cells and recovered their coding properties (fig. 4c-e; middle column;  $R^2$  between 0.44 and 0.91).

272 Since the fluorescence traces were normalized to be nonnegative, we also investigated the performance of *non-*  
273 *negative TCA*. This variant of TCA constrains the neuron, temporal, and trial factors to have nonnegative elements  
274 but is otherwise identical to standard TCA. Nonnegative TCA can produce more interpretable models, since the  
275 model is constrained to reconstruct the original data only through adding, but not subtracting, components, similar  
276 to nonnegative matrix factorization [60]. Despite this additional constraint, nonnegative TCA with 15 components  
277 reconstructed the activity of individual neurons with similar fidelity to an unconstrained TCA model (fig. 4c-e; right  
278 column;  $R^2$  between 0.42 and 0.89).

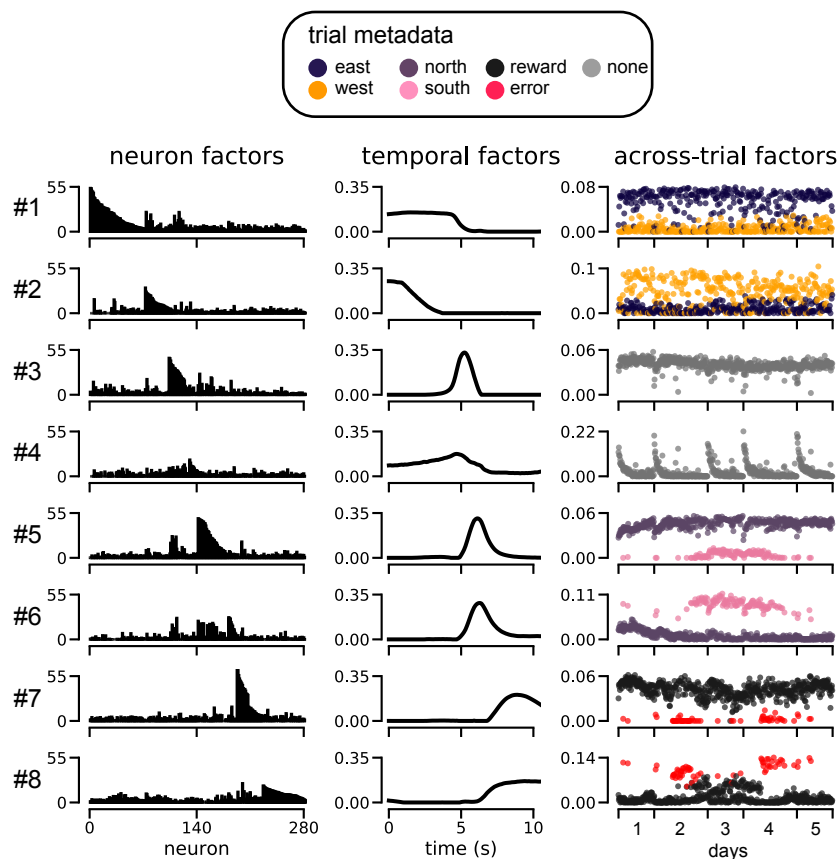
279 We then characterized the performance of TCA and nonnegative TCA across the full population of neurons. We  
280 compared both methods to a *condition-average baseline model*, which predicts the neural activity on each trial to be  
281 the trial-average population activity conditioned on the same trial trajectory (same starting arm and ending arm)  
282 and trial outcome (reward vs. error). That is, we computed the mean activity within each of the eight possible  
283 combinations of trial conditions, decisions, and outcomes, as used this to predict single-trial data. In essence, this  
284 baseline captures the average effect of all task variables, but does not account for trial-to-trial variability within each  
285 combination of task variables.

286 An error plot for unconstrained and nonnegative TCA showed three important findings (fig. 4f). First, nonnegative  
287 TCA had similar predictive performance to unconstrained TCA in terms of reconstruction error across all numbers  
288 of latent components (small gap between red and blue lines, fig. 4f). Second, both forms of TCA converged to  
289 very similar reconstruction error from twenty different random initializations, suggesting that the models did not  
290 get caught in highly suboptimal local minima during optimization (all blue points and all red points reached similar  
291 error, fig. 4f). Third, TCA models with more than 6 components matched or surpassed the condition-average baseline  
292 model, suggesting that relatively few components were needed to explain a substantial fraction of explainable variance  
293 in the dataset (dashed black line, fig. 4f). We also examined the performance of nonnegative and unconstrained TCA  
294 in terms of the  $R^2$  of individual neurons. Again, nonnegative TCA performed similarly to unconstrained TCA as  
295 judged by the median and upper/lower quartiles of the single neuron  $R^2$ , and both models surpassed the simple  
296 condition-average baseline if they included more than 7 components (fig. 4g).

297 In addition to achieving similar accuracy to unconstrained TCA, nonnegative TCA possesses two important  
298 advantages. First, a similarity plot showed that nonnegative models converged more consistently to a similar set of  
299 low-dimensional components (fig. 4h). Second, the components recovered by nonnegative TCA were more sparse,  
300 meaning that each neuron’s activity across all trials was reconstructed by a smaller and more interpretable subset of  
301 components (fig. 4i).

302 While TCA could reconstruct the activity of many neurons very well (fig. 4c-e), other neurons were more difficult  
303 to fit (fig. 4, supp. 1). However, we observed that neurons with low  $R^2$  had firing patterns that were unreliably  
304 timed across trials and did not correlate with task variables (fig. 4, supp. 1b). To visualize this, we plotted the  
305 total variance and the dimensionality of each cell’s activity against the fit of a nonnegative TCA model with 15  
306 components (fig. 4j). The dimensionality of each cell’s activity (see *Methods*, section 4.5.2) measures the trial-to-trial  
307 reliability of a cell’s firing: cells that fire consistently at the same time in each trial will be low-dimensional relative  
308 to cells that fire at different time points in each trial. First, this plot shows a negative correlation between variance  
309 and dimensionality: cells with higher variance (larger dynamic ranges in fluorescence) tended to be lower dimensional  
310 and thus more reliably timed across trials. Second, this plot shows these low-dimensional cells were well fit by TCA,  
311 suggesting that TCA summarizes the information encoded most reliably and strongly by this neural population.  
312 Moreover, outlier cells that defy a simple statistical characterization can be algorithmically identified and flagged for  
313 secondary analysis by sorting neurons by their  $R^2$  score under TCA.

314 TCA’s performance in summarizing neural population activity with very few parameters far exceeds that of trial-



**Fig 5. Nonnegative TCA of prefrontal cortical activity during spatial navigation.** Eight low-dimensional components, each containing a neuron factor (left column), temporal factor (middle column), and trial factor (right column) are shown from a 15-component model (see fig. 5, supp. 1 for the remaining seven components). For each component, the trial factor is color-coded by the task variable it is most highly correlated with.

315 averaged PCA, which has sub-par performance, and trial-concatenated PCA, which requires many more parameters  
 316 to achieve similar performance. This comparison is summarized in Figure 4k, which plots reconstruction error against  
 317 the number of free parameters over 1 to 20 low-dimensional components for each class of models. Trial-averaged PCA  
 318 (fig. 4k, gray line) has fewer parameters than TCA, but cannot account for trial-to-trial changes in activity, cannot  
 319 achieve much lower than 60% error, and by construction entirely misses trial-to-trial fluctuations in neural firing that  
 320 encode task variables. In contrast, trial-concatenated PCA (fig. 4k, black line) achieved comparable reconstruction  
 321 error to TCA but required roughly 100x more free parameters, and is therefore much less interpretable. A TCA  
 322 model with 15 components reduces the complexity of the data by 3 orders of magnitude, from  $\sim 10^7$  datapoints to  
 323  $\sim 10^4$  parameters; whereas a trial-concatenated PCA model with a comparable number of components only reduces  
 324 the number of parameters to  $\sim 10^6$ .

## 325 2.6 Individual TCA components selectively correlate with individual task variables

326 These results demonstrate that TCA accurately describes the firing rates of single cells in a highly compact manner.  
 327 We then examined whether this model identified an interpretable set of low-dimensional components. Figure 5  
 328 shows eight components from a 15-component nonnegative TCA model (the remaining seven factors carry similar  
 329 information and are shown in fig. 5, supp. 1). Each nonnegative TCA component identified a sub-population, or  
 330 assembly of cells (neuron factor; left column) with a common intra-trial temporal dynamics (temporal factor; middle  
 331 column) that was differentially activated across trials (trial factor; right column).

332 In contrast, PCA identified factors that contained complex mixtures of coding for the mouse's position, choice,

333 and reward on each trial (fig. 5, supp. 2), hampering interpretability [34]. TCA on the other hand, isolated each of  
334 these task variables into separate components: each trial factor selectively correlated with a *single* task variable, as  
335 indicated by the color-coded scatterplots in fig. 5. Overall, the TCA model uncovers, in a completely unsupervised  
336 manner, a compelling qualitative view of prefrontal dynamics in which largely distinct subsets of neurons (fig. 5, left  
337 columns) are active at successive times within a trial (fig. 5 middle column) and whose variation across trials (fig. 5  
338 right column) encodes a highly interpretable single task variable.

339 Specifically, components 1-2 uncover neurons that encode the starting location (component 1, east trials; com-  
340 ponent 2, west trials), components 5-6 encode the destination arm (component 5, north trials; component 6, south  
341 trials), and components 7-8 encode the trial outcome (component 7, rewarded trials; component 8, error trials). In-  
342 terestingly, the temporal factors indicate that these components are sequentially activated in each trial: components  
343 1-2 activated before components 5-6, which in turn activated before components 7-8, in agreement with the schematic  
344 flow diagram shown in fig. 4b.

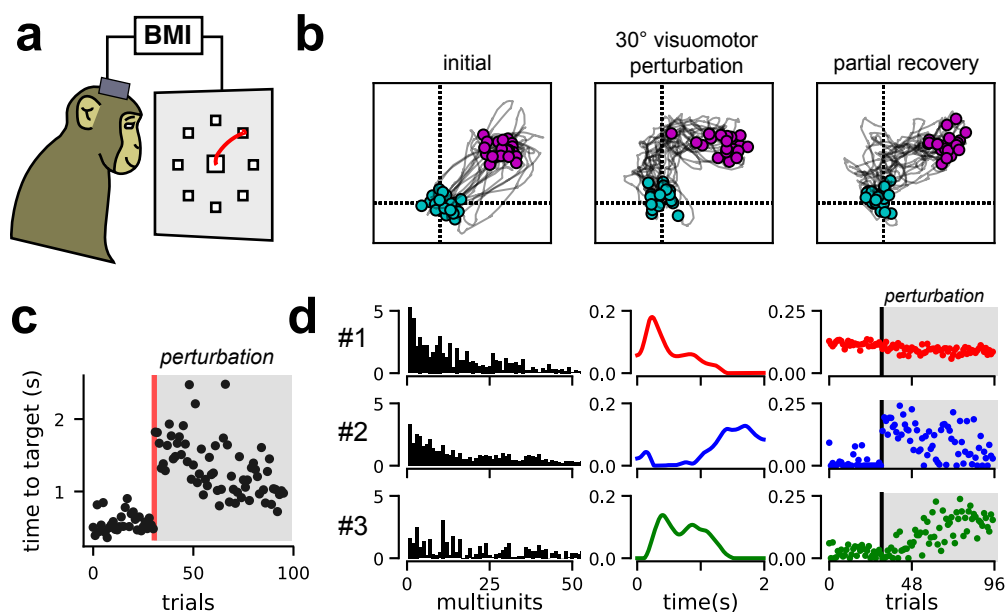
345 Intriguingly, TCA also uncovers unexpected components, like components 3-4 which activate prior to the des-  
346 tination and outcome-related components (i.e., components 5-8). Component 4 displays systematic reductions in  
347 activity across trials within each day, while component 3 is active on nearly every single trial. Component 4 could  
348 potentially correspond, for example, to a novelty or arousal signal that wanes over trials within a day. While further  
349 experiments will be required to ascertain whether this interpretation is correct, the extraction of these components  
350 illustrates the potential power of TCA as an unbiased exploratory data analysis technique to extract unobserved  
351 cognitive states and separate them from observable aspects of trial-to-trial variations in behavior.

352 It is important to emphasize that TCA is an unsupervised method that only has access to the neural data tensor,  
353 and does not receive any information about task variables like starting location, ending location, and reward. There-  
354 fore, the correspondence between TCA trial factors and behavioral information demonstrated in fig. 5, constitutes an  
355 unbiased revelation of task structure directly from neural data. Moreover, individual components extracted by TCA  
356 are in one-to-one correspondence with meaningful aspects of task structure and behavior, a property not shared by  
357 many other dimensionality reduction algorithms.

## 358 **2.7 TCA reveals two-dimensional learning dynamics in macaque motor cortex after** 359 **a BMI perturbation**

360 In the previous section we validated TCA on a dataset where the animal’s behavior decomposed into a set of discrete  
361 experimental conditions, choices, and trial outcomes. We next applied this method to a brain-machine interface  
362 (BMI) learning task, in which the behavior on each trial was quantified by a continuous path of a computer cursor.  
363 The cursor movement on each trial is never identical and is difficult to summarize concisely in a principled manner.  
364 In these more unstructured scenarios, supervised methods such as classification and regression can be difficult to  
365 construct, making unsupervised dimensionality reduction methods an important tool to explore hypotheses and let  
366 the low-dimensional structure of the data “speak for itself.”

367 Specifically, we collected multi-unit data from the pre-motor and primary motor cortices of a Rhesus macaque  
368 (*Macaca mulatta*) controlling a computer cursor in a 2D plane through a brain-machine interface (fig. 6a). Spikes  
369 were recorded when the voltage signal crossed below -4.5 times the root-mean-square voltage. The monkey was  
370 trained to make point-to-point reaches from a central position to one of eight radial targets. For simplicity, we  
371 initially investigated neural activity during 45° outward reaches. The cursor velocity was controlled by a velocity  
372 Kalman filter decoder, which was driven by non-sorted multi-unit activity (-4.5 root-mean-square threshold crossings)  
373 and fit using relations between neural activity and reaches by the monkey’s contralateral arm at the beginning of  
374 the experiment [61]. We analyzed multi-unit activity during subsequent reaches, which used this decoder as a BMI  
375 interface directly from neural activity to cursor motion. These initial reaches were accurate (fig. 6b, left) and took  
376 less than one second to execute (fig. 6c, first 30 trials).



**Fig 6. TCA reveals two-dimensional learning dynamics in primate motor cortex during BMI cursor control.** (a) Schematic of monkey making center-out, point-to-point reaches in BMI task. (b) Cursor trajectories to a 45° target position. Twenty trials are shown at three stages of the behavioral session showing initial performance (left), performance immediately after a 30° counterclockwise visuomotor perturbation (middle), and performance after learning, at the end of the behavioral session. Cyan and magenta points respectively denote the cursor position at the beginning and end of the trial. (c) Time for the cursor to reach target for each trial in seconds. The visuomotor perturbation was introduced after 31 trials (red line). (d) An optimal 3-component nonnegative TCA on smoothed multi-unit spike trains recorded from motor cortex during virtual reaches reveals two components (2-3) that capture learning after the BMI perturbation.

377 We then perturbed the BMI decoder by rotating the output cursor velocities counterclockwise by 30° (a visuomotor  
 378 rotation). Thus, the same neural activity pattern that originally caused a motion of the cursor towards the 45°  
 379 direction, now caused a maladaptive motion in the 75° direction, yielding an immediate drop in performance: the  
 380 cursor trajectories were biased in the counterclockwise direction (fig. 6b, middle), and took longer to reach the target  
 381 (fig. 6c, trials following perturbation). These deficits were partially recovered within a single training session as the  
 382 monkey adapted to the new decoder. By the end of the session, the monkey made more direct cursor movements  
 383 (fig. 6b, right) and achieved the target more quickly (fig. 6c).

384 We applied TCA and nonnegative TCA to the raw spike trains smoothed with a Gaussian filter with a standard  
 385 deviation of 50 ms [34]. We again found that nonnegative TCA fit the data with similar reconstruction error and higher  
 386 reliability than unconstrained TCA (fig. 6 supp. 1). To examine a simple account of learning dynamics, we examined  
 387 a nonnegative TCA model with 3 components. Models with fewer than 3 components had substantially worse  
 388 reconstruction error, while models with more components had only moderately better performance and occasionally  
 389 converged to dissimilar parameters during optimization (fig. 6 supp. 1).

390 The neuron, temporal, and trial factors of the nonnegative TCA model are shown in Figure 6d. Component  
 391 1 (red) described multi-units that were active at the beginning of each trial, and were consistently active over all  
 392 trials. The other two components described multi-units that were inactive before the BMI perturbation, and became  
 393 active only after the perturbation, thereby capturing motor learning. Component 2 (blue) became active on trials  
 394 immediately after the BMI perturbation, but then slowly decayed over successive trials. Within a single trial, this  
 395 component was only active at late stages in the reach. Component 3 (green) on the other-hand was not active  
 396 on trials immediately following the BMI perturbation, but did activate slowly across successive trials. Within a  
 397 single trial, this component was active earlier in the reach. These results suggest a mode of motor learning in which  
 398 a suboptimal, late reaching-stage correction is initially used to perform the task (component 2). Over time, this

399 component is slowly traded for a more optimal early reaching-stage correction (component 3). Interestingly, motor  
400 learning did not involve extinguishing neural dynamics present before the perturbation (component 1), even though  
401 this component is maladaptive after the perturbation.

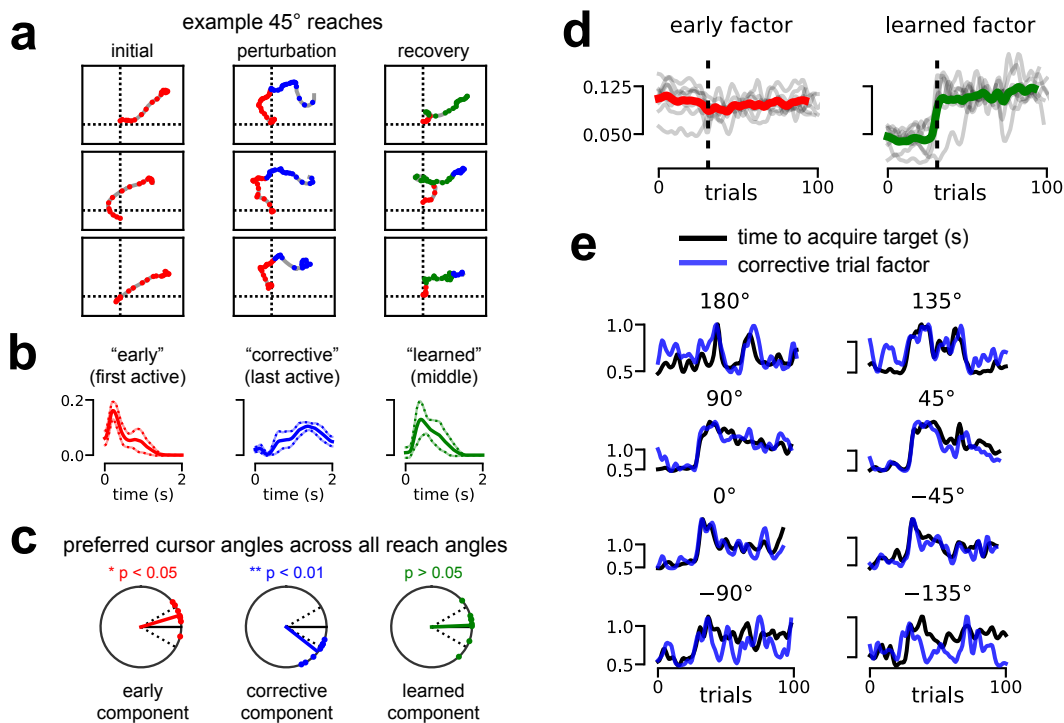
402 We were able to confirm this intuition by relating each of these components to a different phase of motor execution  
403 and learning. Figure 7a plots cursor trajectories on individual reaches before the perturbation (left), immediately  
404 following the perturbation (middle), and at the end of the behavioral session (right). Every 50ms the trajectory was  
405 colored based on the component with the largest activation at that timepoint and trial. Prior to the perturbation,  
406 component 1 (red) dominated; the other two components were nearly inactive since their TCA trial factor amplitudes  
407 were near zero before the perturbation (see fig. 6d). Immediately following the perturbation, component 1 still  
408 dominated in the early phase of each trial, producing a counterclockwise off-target trajectory. However, component  
409 2 dominated the second half of each trial at which point the monkey performed a “corrective” horizontal movement  
410 to compensate for the initial error. Finally, near the end of the training session, component 3 was most active  
411 at many stages of the reach. Typically, the cursor moved directly towards the 45° target when component 3 was  
412 active, suggesting that component 3 captured learned neural dynamics that were correctly adapted to the perturbed  
413 visuomotor environment.

414 Based on these observations, we called the component active at the beginning of each trial the *early component*  
415 (#1 in fig. 6), the component active at the end of each trial the *corrective component* (#2 in fig. 6), and the  
416 component active in the middle of each trial the *learned component* (#3 in fig. 6). These components are colored  
417 red, blue, and green respectively in both Figure 6 and Figure 7. We then fit 3-component TCA models separately  
418 to each of the eight reach angles, and operationally defined the components as *early*, *corrective*, and *learned* based  
419 on the peak magnitude of their associated within-trial temporal basis functions (fig. 7b). This very simple definition  
420 yielded similar interpretations for low-dimensional components separately fit across different reach angles.

421 Similar to computing a directional tuning curve for an individual neuron [62], we examined the *preferred cursor*  
422 *angles* of each low-dimensional component by computing the average cursor velocity weighted by activity of the  
423 component (see *Methods*, section 4.4.7). To compare across all target reach angles, we rotated the preferred angles  
424 so that the target was situated at 0° (black line, fig. 7c). All preferred angles were computed on post-perturbation  
425 trials. When the *early component* was active, the cursor typically moved at an angle counterclockwise to the target  
426 ( $p < 0.05$ , one sample test for the mean angle), reflecting our previous observation that the early component encodes  
427 pre-perturbation dynamics that are maladaptive post-perturbation (fig. 7c, left). When the *corrective component* was  
428 active, the cursor typically moved at an angle clockwise to the target ( $p < 0.01$ , one sample test for the mean angle),  
429 reflecting a late-trial compensation for the error introduced by the early component (fig. 7c, middle). Finally, the  
430 *learned component* was not significantly different from the target angle, reflecting a tuning that was better adapted  
431 for the perturbed visuomotor environment.

432 Having established a within-trial interpretation for each component, we next examined across-trial learning dy-  
433 namics. For visualization purposes, we gently smoothed all TCA trial factors by a Gaussian filter with a standard  
434 deviation of 1.5 trials. Across all reach angles, the *early component* was typically flat and insensitive to the visuomo-  
435 tor perturbation (fig. 7d, left). In contrast, the *learned component* activated soon after the perturbation was applied,  
436 although the rapidness of this onset varied across reach angles (fig. 7d, right). Together, this reinforces our earlier  
437 observation that adaptation to the visuomotor rotation typically involves the production of new neural dynamics  
438 (captured by the *learned component*), rather than the suppression of maladaptive dynamics (captured by the *early*  
439 *component*).

440 Finally, the *corrective component* was consistently correlated with the animal’s behavioral performance on all reach  
441 angles ( $p < 0.05$ , Spearman’s rho test). Since performance differed across reach angles, we separately plotted the  
442 *corrective component* (blue) against the time to acquire the target (black) for each reach angle (fig. 7e). Remarkably,  
443 in many cases, the corrective component provided an accurate trial-by-trial prediction of the reach duration, meaning  
444 that trials with a large corrective movement took longer to execute.



**Fig 7. TCA tracks performance and uncovers “corrective” dynamics in BMI adaptation task.** (a) Cursor trajectories for 45° cursor reaches. Every 50 ms, the trajectory is colored by the TCA component with the strongest activation at that timepoint and trial. Components were colored according to the definition in panel (b). Three example trajectories are shown at three stages of the experiment: reaches before the visuomotor perturbation (left), reaches immediately following the perturbation (middle), and reaches at the end of the behavioral session. (b) Average low-dimensional temporal factors identified by nonnegative TCA across all eight reach angles. The *early component* had the earliest active temporal factor (red). The *corrective component* had the last active temporal factor (blue). The *learned component* was the second active temporal factor (green). Solid and dashed lines denote mean  $\pm$  standard deviation. (c) Preferred cursor angles for each component type after the visuomotor perturbation. All data were rotated so that the target reach angle was at 0° (solid black line). Dashed black lines denote  $\pm 30^\circ$  for reference, which was the magnitude of the visuomotor perturbation. On average, the *early component* was associated with a cursor angle misaligned counterclockwise from the target (red). The *corrective component* preferred angle was aligned clockwise from the target (blue) by about  $30^\circ$ , in a way that could compensate for the  $30^\circ$  counterclockwise misalignment of the *early component*. The *learned component* preferred angle not significantly different from that of the actual target. (d) Smoothed trial factors for the *early component* and *learned component*. Colored lines denote averages across all reach angles; gray lines denote the factors for each of the eight reach conditions. Factors were smoothed with a Gaussian filter with 1.5 standard deviation for visualization purposes. (e) Smoothed trial factor for the *corrective component* (blue) and smoothed behavioral performance (black) quantified by seconds to reach target. Each subplot shows data for a different reach angle. All signals were smoothed with a Gaussian filter with 1.5 standard deviation for visualization.



445 Together, these results demonstrate that TCA can identify, in a purely unsupervised manner, both learning  
446 dynamics across trials and single trial neural dynamics. Indeed, each trial factor can be related to within-trial behav-  
447 iors, such as error-prone cursor movements and their subsequent correction. Furthermore, these basic interpretations  
448 largely replicate across all eight reach angles, despite differences in the learning rate within each of these conditions.  
449 Most intriguingly, a *single* trial factor, extracted only from neural data, can directly predict execution time on a trial  
450 by trial basis, without ever having direct access to this aspect of behavior (fig. 7e).

### 451 3 Discussion

452 Recent experimental technologies enable us to record from more neurons, at higher temporal precision, and for  
453 much longer time periods than ever before [4, 8, 11], thereby simultaneously increasing the size and complexity of  
454 datasets along three distinct modes. However, methods for multi-timescale dimensionality reduction that describe  
455 both rapid neural dynamics within trials and long-term changes in neural dynamics across-trials are still lacking. As  
456 a result, experimental investigations of neural circuits are often confined to a single timescale, even though bridging  
457 our understanding across multiple timescales is of great interest [9]. Here we demonstrated a unified approach,  
458 TCA, that simultaneously recovers low-dimensional and interpretable structure across neurons, time within trials,  
459 and trials.

460 TCA and other tensor decomposition techniques have been extensively studied from a theoretical perspective  
461 [29, 63–65], and have been applied to a variety of biomedical problems [66–68]. Several studies have applied tensor  
462 decompositions to EEG and fMRI data, most typically to model differences across subjects or Fourier/wavelet  
463 transformed signals [69–72], rather than across trials [73]. A recent study examined trial-averaged neural data across  
464 multiple neurons, conditions, and time within trials as a tensor, but they did not study trial-to-trial variability, and  
465 only examined different unfoldings of the data tensor into matrices, rather than applying TCA directly to the data  
466 tensor [74]. Other studies have modeled the receptive fields of neurons in auditory and visual cortex as third-order  
467 tensors with low-rank structure [75, 76]. We go beyond these previous studies by applying TCA to a broader class  
468 of artificial and experimental datasets, drawing a novel connection between TCA and theories of gain modulation,  
469 and demonstrating that visualization and analysis of the TCA trial factors can directly yield functional clustering of  
470 neural populations (i.e., cell assemblies) as well reveal learning dynamics on trial-by-trial basis.

471 In particular, we demonstrated that TCA reveals a simple description of learning in an artificial nonlinear neural  
472 network trained to solve the analog of a motion discrimination task [32]. TCA discovered a one-dimensional learning  
473 process in which initial, small random selectivity is monotonically amplified over time to yield the final learned  
474 decision making dynamics. Moreover, cell-type information extracted by TCA in an unsupervised manner enabled  
475 us to re-organize the network’s connectome, thereby yielding conceptual insights into how this connectome gives rise  
476 to mechanisms for decision making. Also, in calcium imaging data recorded from rodent pre-frontal cortex during a  
477 maze navigation task, TCA uncovered functional subsets of neurons that fired sequentially within trials, and whose  
478 amplitude on each trial selectively mapped onto task-relevant variables, including starting location, ending location  
479 and reward (in that order).

480 Finally, in electrophysiological recordings from macaque motor and premotor cortex, TCA revealed a simple  
481 two-dimensional learning process in response to a BMI perturbation. Interestingly, this learning process did not  
482 involve extinguishing maladaptive dynamics that were established in the pre-perturbation period (i.e., the “early  
483 component” identified by TCA). Rather, it involved the addition of two components that compensated for the  
484 maladaptive dynamics. The first “corrective” component was a suboptimal, late stage within-trial correction that  
485 was active in trials soon after the perturbation, which extinguished over trials to give rise to a second “learned”  
486 component that implemented a more optimal early stage within-trial correction. Moreover, the late stage correction  
487 could predict time to target acquisition on a trial-by-trial basis. Importantly, all of these results were discovered

488 purely from the neural data and not behavioral measurements, suggesting that TCA can uncover unexpected and  
489 otherwise unobservable neural dynamics in a data-driven, unsupervised manner.

490 In addition to the empirical success of TCA in diverse scenarios presented here, there are three other reasons  
491 we expect TCA to have widespread utility in neuroscience. First, TCA is arguably the simplest generalization of  
492 PCA that can handle trial-to-trial variability. Given the widespread adoption of PCA, we believe that TCA may  
493 also enjoy widespread adoption and success, especially as technologies enabling long-term and large-scale recordings  
494 become more accessible. Second, as we have shown, TCA has an intriguing interpretation as a network model with  
495 low-dimensional gain-modulated inputs. This model is supported by experimental evidence in many contexts [22,  
496 37, 76–78], and underlies influential theories of cortical computation [30, 31] and perceptual learning [79].

497 Third, while TCA is a simple generalization of PCA, its theoretical properties are strikingly more favorable. A  
498 fundamental limitation of PCA is that the components it recovers are restricted to be orthogonal to each other,  
499 and moreover these components can be rotated amongst each other without changing the reconstruction error. This  
500 invariance to rotations in PCA leads to a fundamental ambiguity, and so the factors identified by PCA are unlikely  
501 to be directly interpretable as biological signals (see *Methods*, section 4.4.2). In contrast, the factors identified by  
502 TCA are not invariant to many transformations [29], yielding more interpretable results. This advantage was first  
503 demonstrated in fig. 2 where the factors recovered from neural firing rates, matched the underlying parameters of the  
504 model neural network in a one-to-one fashion. Similarly, in the rodent prefrontal analysis, TCA uncovers demixed  
505 factors that individually correlate with interpretable task variables, whereas PCA does not (compare fig. 5 to fig. 5  
506 supp. 1). And finally, when applied to neural activity during BMI learning, TCA consistently found, across multiple  
507 reach angles, a “corrective factor” that significantly correlated with behavioral performance on a trial-by-trial basis  
508 (fig. 7).

509 In this paper, we examined the simplest form of TCA by making no assumptions about the temporal dynamics  
510 of neural activity within trials or the dynamics of learning across trials. As a result, we obtain extreme flexibility:  
511 for example, trial factors could be discretely activated or inactivated on each trial (fig. 5), or they might emerge  
512 incrementally over longer timescales (fig. 6). However, future work could augment TCA with additional structure  
513 and assumptions, such as a smoothness penalty or dynamical systems structure within trials [16]. Intriguingly, a  
514 dynamical system could just as easily be incorporated along the trials axis of the data tensor to potentially relate  
515 high-dimensional neural activity to low-dimensional models of learning [80].

516 Further work in this direction could connect TCA to a large body of work on fitting latent dynamical systems  
517 to reproduce within-trial firing patterns. In particular, single trial neural activity has been modeled with linear  
518 dynamics [81–84], switched linear dynamics [85, 86], linear dynamics with nonlinear observations [17], and nonlinear  
519 dynamics [18, 87]. In practice, these methods require many modeling choices, validation procedures, and post-hoc  
520 analyses. Simple linear models have a relatively constrained dynamical repertoire [12], while models with nonlinear  
521 elements often have greater predictive abilities [17, 18], but at the expense of interpretability. In all cases, the learned  
522 representation of each trial (e.g., the initial condition to a nonlinear dynamical system) is not transparently related  
523 to single trial data. In contrast, the trial factors identified by TCA have an extremely simple interpretation as  
524 introducing trial-specific linear gain modulation. Overall, we view TCA as a simple and complementary technique  
525 to identifying a full dynamical model, as has been previously suggested for PCA [12].

526 An important property of TCA is that it extracts salient features of a dataset in a data-driven, unbiased fashion.  
527 Such unsupervised methods are a critical counterpart to supervised methods, such as regression, which can directly  
528 assess whether a dependent variable of interest is represented in population activity. Recently developed methods like  
529 *demixed PCA* [34] combine regression with dimensionality reduction to isolate linear subspaces that selectively code  
530 for variables of interest. Again, we view TCA as a complementary approach, with at least three points of difference.  
531 First, like trial-concatenated PCA and GPFA, demixed PCA only reduces dimensionality within trials by identifying  
532 a different low-dimensional temporal trajectory for each trial. In contrast, TCA identifies a common low-dimensional  
533 temporal trajectory (temporal factors) for all trials, which are modulated by different amplitudes (trial factors) on

534 each trial. Second, demixed PCA can separate neural dynamics in cases where trials have discrete conditions and  
535 labels, such as in the rodent prefrontal analysis in fig. 5; however, it is not designed to handle continuous dependent  
536 variables, such as those describing learning dynamics (see fig. 3 and fig. 7). Furthermore, unsupervised techniques  
537 like TCA can identify unexpected cognitive states and dynamics corresponding to unknown or difficult to measure  
538 dependent variables. Finally, the same rotation invariance of PCA is present within the linear subspaces identified  
539 by demixed PCA. Thus, both PCA and demixed PCA are fundamentally subspace identification algorithms, while  
540 TCA can often extract directly meaningful features from data, such as clusters of functional cell types or neural  
541 populations that grow or shrink in magnitude across trials.

542 An intriguing direction for future research is to expand TCA to higher-order tensors beyond those encoding  
543 neurons, timepoints, and trials. For example, we can also record across multiple subjects learning to solve the same  
544 task, yielding a fourth order data tensor, with individual subjects as the fourth index. Similarly, if an individual  
545 subject is taught multiple learning tasks, one could encode experimental task or condition as a fourth index. However,  
546 directly applying TCA to these tensors may be undesirable, since we record from different neural populations in  
547 different subjects and the learning rate may vary from subject-to-subject or from task-to-task. Instead, we could  
548 model such data via coupled tensor factorizations [88] which allow some measured tensor axes to be fit as common  
549 factors, while others are fit in a separate and unconstrained fashion. For instance, we could assign separate neuron and  
550 trial factors for each subject, but use shared temporal factors across subjects if they are hypothesized to share similar  
551 low-dimensional within trial cognitive dynamics. This scheme could extract common circuit dynamics from small  
552 numbers of neurons through increased statistical power obtained via pooling across multiple subjects. Moreover, the  
553 separate neuron factors would then provide “translations” between subjects, by revealing how the same cognitive  
554 variable is encoded in different population activity patterns in different subjects. In essence, while moving from  
555 second to third-order tensor methods provides a new window into how circuit dynamics changes across trials to  
556 mediate learning, moving additionally to fourth order tensor methods may provide new insights into how the learning  
557 dynamics itself changes across subjects and tasks.

558 Overall, this work highlights the prevalence of tensor structure in neural datasets and demonstrates that exploiting  
559 this structure can provide extremely useful insights into complex, multi-timescale, high-dimensional neural data,  
560 including the unsupervised discovery of cell assemblies, within trial neural dynamics underlying perceptions, actions  
561 and thoughts, and across trial learning dynamics. Just as PCA has become part of the standard canon of neural  
562 data analyses for trial-averaged neural recordings, the combined simplicity and power of TCA suggests it may have  
563 widespread utility in the analysis of multineuronal data at the level of single trials.

## 564 **4 Methods**

### 565 **4.1 Key Resources Table**

### 566 **4.2 Contact for Reagent and Resource Sharing**

567 Further requests for resources should be directed to and will be fulfilled by the Lead Contact, Alex H. Williams  
568 (ahwillia@stanford.edu)

### 569 **4.3 Data and Software Availability**

570 We provide specialized tools for fitting and visualizing TCA in <https://github.com/ahwillia/tensortools>. Other  
571 resources for fitting tensor decompositions include [93–95].

Reagent or Resource	Source	Identifier
Software and Algorithms		
tensortools	This paper	<a href="https://github.com/ahwillia/tensortools">https://github.com/ahwillia/tensortools</a>
Alternating Least Squares	[27]	N/A
SciPy	[89]	<a href="https://scipy.org/">https://scipy.org/</a>
Matplotlib	[90]	<a href="https://matplotlib.org/">https://matplotlib.org/</a>
scikit-learn	[91]	<a href="http://scikit-learn.org/">http://scikit-learn.org/</a>
PyTorch	none	<a href="http://pytorch.org/">http://pytorch.org/</a>
MATLAB	MathWorks	<a href="https://www.mathworks.com/products/matlab.html">https://www.mathworks.com/products/matlab.html</a>
Simulink Realtime	MathWorks	<a href="https://www.mathworks.com/products/simulink-real-time.html">https://www.mathworks.com/products/simulink-real-time.html</a>
Experimental Models: Organisms/Strains		
C57BL/6J mice	The Jackson Laboratory	000664
Rhesus macaque ( <i>Mucacca Mulatta</i> )	Wisconsin and Yerkes primate centers	N/A
Recombinant DNA		
pGP-CMV-GCamP6m	[92]	#40754, <a href="https://www.addgene.org/Douglas_Kim/">https://www.addgene.org/Douglas_Kim/</a>
Other		
Miniature fluorescence microscope	Inscopix	<a href="https://www.inscopix.com/nvista">https://www.inscopix.com/nvista</a>
Utah Microelectrode Arrays	Blackrock Microsystems	<a href="http://blackrockmicro.com/neuroscience-research-products/low-noise-ephys-electrodes/blackrock-utah-array/">http://blackrockmicro.com/neuroscience-research-products/low-noise-ephys-electrodes/blackrock-utah-array/</a>
Cerebus System	Blackrock Microsystems	<a href="http://blackrockmicro.com/neuroscience-research-products/neural-data-acquisition-systems/cerebus-daq-system/">http://blackrockmicro.com/neuroscience-research-products/neural-data-acquisition-systems/cerebus-daq-system/</a>

## 4.4 Method Details

### 4.4.1 Notation and Terminology

Colloquially, a tensor is a data array or table with multiple axes or dimensions. More formally, the axes are called *modes* of the tensor, while the *dimensions* of the tensor are the lengths of each mode. Throughout this paper we consider a tensor with three modes with dimensions  $N$  (number of neurons),  $T$  (number of timepoints in a trial), and  $K$  (number of trials).

The number of modes is called the *order* of the tensor. We denote vectors (order-one tensors) with lowercase boldface letters, e.g.,  $\mathbf{x}$ . We denote matrices (order-two tensors) with uppercase boldface letters, e.g.,  $\mathbf{X}$ . We denote higher-order tensors (order-three and higher) with boldface calligraphic letters, e.g.,  $\mathcal{X}$ . Scalars are denoted by non-boldface letters, e.g.,  $x$  or  $X$ . We use  $\mathbf{X}^T$  to denote the transpose of  $\mathbf{X}$ . We aim to keep other notation light and introduce as it is first used — readers may refer to [33] for notational conventions.

### 4.4.2 Matrix and Tensor models

Neural population activity is commonly represented as a matrix with each row holding a neuron’s activity trace [12]. Let  $\mathbf{X}$  denote an  $N \times T$  matrix dataset in which  $N$  neurons are recorded over  $T$  time steps. For spiking data,  $\mathbf{X}$  may denote trial-averaged spike counts or a single-trial spike train smoothed with a Gaussian filter. If fluorescence microscopy is used in conjunction with voltage or calcium indicators, the data entries could be normalized fluorescence ( $\Delta F/F$ ).

PCA is a special case of *matrix decomposition*. A matrix decomposition model approximates the data  $\mathbf{X}$  as a rank- $R$  matrix,  $\widehat{\mathbf{X}}$ , yielding  $R$  components. This approximation can be expressed as the product of an  $N \times R$  matrix  $\mathbf{W}$  and a  $T \times R$  matrix  $\mathbf{B}$ :

$$\mathbf{X} \approx \widehat{\mathbf{X}} = \mathbf{W}\mathbf{B}^T. \quad (3)$$

We call the columns of  $\mathbf{W}$  neuron factors, denoted  $\mathbf{w}^r$ , and the columns of  $\mathbf{B}$  temporal factors, denoted  $\mathbf{b}^r$ . The rows of  $\mathbf{W}$ , denoted  $\mathbf{w}_n$ , provide an  $R$ -dimensional description of each neuron’s activity trace. Likewise the rows of  $\mathbf{B}$ , denoted  $\mathbf{b}_t$ , provide an  $R$ -dimensional description of the full neural population activity pattern at each timepoint. In order to reduce the dimensionality of the data we chose  $R < N$  and  $R < T$ . Note that eq. (3) is equivalent to eq. (1) in the *Results*.

Perhaps the simplest matrix decomposition problem is to identify a rank- $R$  decomposition that minimizes the squared reconstruction error:

$$\underset{\mathbf{W}, \mathbf{B}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{W}\mathbf{B}^T\|_F^2. \quad (4)$$

Here,  $\|\cdot\|_F^2$  denotes the squared *Frobenius norm* of a matrix, which is simply the sum of squared matrix elements:

$$\|\mathbf{X}\|_F^2 = \sum_{n=1}^N \sum_{t=1}^T x_{nt}^2.$$

PCA provides *one* solution to eq. (4). Most critically, the PCA solution constrains the neuron factors and temporal factors to be orthogonal, meaning that  $\mathbf{W}^T\mathbf{W}$  and  $\mathbf{B}^T\mathbf{B}$  are diagonal matrices. However, this solution does not uniquely minimize the squared reconstruction error. In fact, there is a continuous manifold of matrix decompositions that solve eq. (4), since any invertible linear transformation  $\mathbf{F}$  can produce a new set of parameters,  $\mathbf{W}' = \mathbf{W}\mathbf{F}^{-1}$  and  $\mathbf{B}' = \mathbf{B}\mathbf{F}^T$  that produce an equivalent reconstruction of the data:

$$\mathbf{W}\mathbf{B}^T = \mathbf{W}\mathbf{F}^{-1}\mathbf{F}\mathbf{B}^T = \mathbf{W}'\mathbf{B}'^T = \widehat{\mathbf{X}} \quad (5)$$

This result — sometimes called the *rotation problem* — has a fundamental consequence: if the data were truly

605 generated as a combination of  $R$  low-dimensional components, then PCA *cannot not recover these ground truth*  
606 *components*. At best, PCA can only be expected to recover the same linear subspace of the true components.

607 In essence, after fitting a PCA model, one might be tempted to interpret the columns of  $\mathbf{W}$  as identifying sub-  
608 populations of neurons with firing patterns given by the columns in  $\mathbf{B}$ . However, eq. (5) shows that these putative  
609 sub-populations can be linearly mixed by a broad class of transformations, so long as the components are mixed by  
610 the appropriate inverse transformation. Thus, the latent factors identified by PCA are poorly constrained, and it  
611 is better to interpret PCA as finding an orthogonal coordinate basis for visualizing data. As reviewed below, the  
612 optimization problem addressed by TCA has superior uniqueness properties relative to eq. (4), which gives us greater  
613 license to directly interpret the TCA factors as potentially biologically meaningful neural populations and activity  
614 patterns.

615 TCA is a natural generalization of PCA to higher-order tensors. Let  $\mathcal{X}$  denote a  $N \times T \times K$  data tensor, and let  
616  $x_{ntk}$  represent the activity of neuron  $n$  at time  $t$  on trial  $k$ . For a third-order tensor, TCA finds a set of three factor  
617 matrices,  $\mathbf{W}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$ , with dimensions  $N \times R$ ,  $T \times R$ , and  $K \times R$ , respectively. As before, the columns of  $\mathbf{W}$  are  
618 the neuron factors, the columns of  $\mathbf{B}$  are the temporal factors. Analogously, the columns of  $\mathbf{A}$  are the trial factors,  
619 denoted  $\mathbf{a}^r$ , and the rows of  $\mathbf{A}$ , denoted  $\mathbf{a}_k$ , embed each trial into an  $R$ -dimensional space.

620 To reformulate eq. (2) into an equivalent matrix equation, let  $\mathbf{X}_k$  denote an  $N \times T$  matrix holding the data from  
621 trial  $k$ . TCA models each trial of neural data as:

$$\widehat{\mathbf{X}}_k = \mathbf{W} \text{Diag}(\mathbf{a}_k) \mathbf{B}^T, \quad (6)$$

where  $\text{Diag}(\mathbf{a}_k)$  embeds  $\mathbf{a}_k$  as the diagonal entries of an  $R \times R$  matrix. Again, eq. (6) is equivalent to eq. (2) in the  
*Results*. In this paper, we also employed the *nonnegative* TCA model, which simply adds a constraint that all factor  
matrices have nonnegative elements:

$$\mathbf{W} \geq 0, \mathbf{B} \geq 0, \mathbf{A} \geq 0.$$

622 Nonnegative TCA has been previously studied in the tensor decomposition literature [64, 96–98], and is a higher-  
623 order generalization of nonnegative matrix factorization (NNMF) [60, 99]. Similar to eq. (3), in this paper both  
624 unconstrained and nonnegative TCA were fit to minimize the squared reconstruction error:

$$\underset{\mathbf{W}, \mathbf{B}, \mathbf{A}}{\text{minimize}} \quad \|\mathcal{X} - \widehat{\mathcal{X}}\|_F^2 \quad (7)$$

625 Both PCA and TCA can be extended to incorporate different loss functions, such as a Poisson negative log-likelihood  
626 [100], however we do not consider these models in this paper.

627 Fitting TCA to data is a nonconvex problem. Unlike PCA, there is no efficient procedure for achieving a  
628 certifiably optimal solution [65]. We use established optimization algorithms to minimize eq. (7) from an initial  
629 guess (see section 4.4.3). Although this approach may converge to local minima in the objective function, our results  
630 empirically suggest that this is not a major practical concern. Indeed, as long we does not choose too many factors  
631 (too large an  $R$ ) and use nonnegative factors, we find that the multiple local minima yield similar parameter values  
632 and similar reconstruction error.

633 An important advantage of TCA is that the low-dimensional components it uncovers are often “essentially unique,”  
634 up to permutations and scalings. More precisely, in [29], it was proven that every local minimum of the TCA objective  
635 function is isolated in parameter space; it is not part of a continuous manifold of parameters that achieve *exactly*  
636 the same reconstruction error, as in matrix factorization described above. Instead, this continuous degeneracy, or  
637 ambiguity is replaced by a much more benign ambiguity, namely a set of solutions with the same reconstruction  
638 error related to each other simply by permutations and rescalings. For instance, the columns of  $\mathbf{W}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$  can  
639 be jointly permuted without affecting the model. Also, the columns of any pair of  $\mathbf{W}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$  can be jointly  
640 rescaled. For example, if the  $r^{\text{th}}$  column of  $\mathbf{W}$  is multiplied by a scalar  $s$ , then the  $r^{\text{th}}$  column of either  $\mathbf{B}$  or  $\mathbf{A}$  can

641 be divided by  $s$  without affecting the model’s prediction. These transformations, which are also present in PCA, are  
642 inconsequential since the direction of the latent factors and total size of any set of factors, rather than their order,  
643 are of primary interest. Thus the parameter set corresponding to the global minimum of TCA is essentially unique,  
644 up to permutations and scalings. Of course, in general we are not guaranteed to find this global minimum, but as  
645 we have shown in the main text, in situations where we do not choose too many factors, all the local minima we  
646 find using multiple runs of TCA achieve similarly low reconstruction error, and moreover are close to each other in  
647 parameter space. In such a situation, all the local minima likely cluster near the global minimum, and the resultant  
648 parameter values are likely to be biologically meaningful, or interpretable.

649 In summary, when the factors are all linearly independent (i.e.,  $\mathbf{W}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$  have full column rank), TCA  
650 is, in the sense described above, provably unique up to rescalings and permutations [29]. TCA can nevertheless  
651 be difficult to optimize if latent factors are approximately linearly dependent [101]. To quantify and monitor this  
652 possibility, we computed a similarity score between TCA models based on the angles between the extracted factors  
653 (see section 4.5.1). In practice, we did not find this to be a critical problem.

#### 654 4.4.3 Model optimization

655 TCA can be applied to neural data by a series of simple steps (Figure 2f). First, to incorporate the common  
656 assumption that latent neural firing rates are smooth in time [16], spiking data can be temporally smoothed (e.g.,  
657 with a Gaussian filter). The width of this smoothing filter affects the smoothness of the latent temporal factors  
658 recovered by TCA. Analogous smoothness hyperparameters are present in other dimensionality reduction methods.  
659 For example, in GPFA, the timescale of latent dynamics are set by the autocorrelation in the prior’s covariance  
660 matrix [16]. Depending on the dataset, it may be important to apply other common preprocessing steps, such as  
661 z-scoring the activity traces of neurons, or applying variance-stabilizing transformations such as taking the square  
662 root of spike counts [12].

663 Like many dimensionality reduction methods, TCA can only be fit by iterative optimization algorithms. While  
664 these procedures may get stuck in sub-optimal local minima, in practice we found that all optimization fits converged  
665 to similar reconstruction errors. Other techniques, such as nonnegative matrix factorization [60], also demonstrate  
666 practical success while being NP-hard in terms of worst-case analysis [102].

667 Specialized algorithms for fitting TCA are an area of active research. We used the classic method of *alternating*  
668 *least-squares* (ALS) to obtain estimates of the factor matrices. ALS is motivated by the observation that fixing  
669 two of the factor matrices and optimizing over the third in eq. (7) is a least-squares subproblem that is convex and  
670 has a closed-form solution. For illustration, consider optimizing the neuron factors  $\mathbf{W}$ , while temporarily fixing the  
671 within-trial factors,  $\mathbf{B}$ , and the trial factors  $\mathbf{A}$ . This yields the following update rule:

$$\mathbf{W} \leftarrow \underset{\tilde{\mathbf{W}}}{\operatorname{argmin}} \sum_{ntk} \left( x_{ntk} - \sum_r \tilde{w}_n^r b_t^r a_k^r \right)^2, \quad (8)$$

672 which can be solved as a linear least-squares matrix problem. In particular, with some manipulation of the indices,  
673 eq. (8) can be rearranged into a matrix equation (see [33]) and solved by standard matrix library routines. This  
674 procedure is then cyclically repeated: the temporal factors  $\mathbf{B}$  are updated while fixing  $\mathbf{W}$  and  $\mathbf{A}$ , then the trial  
675 factors  $\mathbf{A}$  are updated while fixing  $\mathbf{W}$  and  $\mathbf{B}$  and so on until the objective function converges. The ALS algorithm  
676 is available in several open-source packages [**tensor toolbox 2.6**, 94, 95], and is reviewed in [33]. For nonnegative  
677 TCA, we solved each sub-problem using a specialized nonnegative least squares solver [103], instead of standard  
678 least-squares.

#### 679 4.4.4 Linear gain-modulated model network

680 In fig. 2, we constructed a linear network model with three input neurons connected to  $N = 50$  observed neurons  
681 by random Gaussian weights. The outgoing weights of each input neuron were normalized to unit Euclidean length.  
682 Each input neuron had a different temporal firing pattern lasting  $T = 150$  time steps, parameterized as probability  
683 density functions of Gamma distributions. The trial-specific amplitude of the first two input neurons were respectively  
684 parameterized as increasing and decreasing logarithmically spaced points over  $K = 100$  trials. The amplitude of  
685 the third input neuron linearly increased for  $K < 50$  and then linearly decreased to the same starting value. All  
686 within-trial waveforms and across-trial amplitude vectors were normalized to unit Euclidean length. As described  
687 in the *Results*, the activity of all neurons is modeled by the same equations as TCA (eq. (2)). Independent and  
688 identically distributed Gaussian noise with a standard deviation of 0.01 was added to the simulated data. ICA and  
689 PCA were performed on this simulated dataset via the scikit-learn Python package [91].

#### 690 4.4.5 Nonlinear recurrent neural network model

We simulated a discrete-time recurrent neural network with a hyperbolic tangent nonlinearity.

$$\mathbf{x}_t = \tanh(\mathbf{J}_{\text{rec}}\mathbf{x}_{t-1} + \mathbf{J}_{\text{in}}\mathbf{u}_t + \boldsymbol{\beta}) \quad (9)$$

$$\mathbf{y}_t = \mathbf{J}_{\text{out}}\mathbf{x}_t \quad (10)$$

691 Here,  $\mathbf{x}_t$  is a vector of  $N$  neural firing rates of the recurrently connected neural population at time  $t$ ,  $\mathbf{u}_t$  and  $\mathbf{y}_t$  are the  
692 inputs and outputs of the network,  $\mathbf{J}_{\text{rec}}$ ,  $\mathbf{J}_{\text{in}}$ ,  $\mathbf{J}_{\text{out}}$  are synaptic weight matrices for the recurrent, input, and output  
693 connections, and  $\boldsymbol{\beta}$  is a  $N$ -dimensional vector of bias terms. The input and output of the were one-dimensional  
694 signals, as illustrated in fig. 3a. Thus, the recurrent synaptic weights were held in a  $N \times N$  matrix,  $\mathbf{J}_{\text{rec}}$ , the input  
695 weights were held in a  $N \times 1$  matrix,  $\mathbf{J}_{\text{in}}$ , and the output weights were held in a  $N \times 1$  matrix,  $\mathbf{J}_{\text{out}}$ .

696 On each trial, the input signal to the network consisted of  $T = 40$  independent draws from a standard normal  
697 distribution with mean  $\mu = 1$  or  $\mu = -1$  (chosen randomly with equal probability on each trial). The goal of the  
698 network was to produce a positive output ( $y_t > 0$ ) when the input was net-positive, and produce a negative output  
699 ( $y_t < 0$ ) when the input was net-negative. The performance of the network on each trial was measured by a logistic  
700 loss function (applied to the output on the final time step,  $y_T$ ):

$$\ell(y_T, \mu) = \log(1 + \exp(-\mu y_T))$$

701 For each simulated trial, we used the deep learning framework PyTorch to compute the gradient of this loss function  
702 with respect to all network parameters  $\{\mathbf{J}_{\text{rec}}, \mathbf{J}_{\text{in}}, \mathbf{J}_{\text{out}}, \boldsymbol{\beta}\}$  via the backpropagation through time algorithm. A small  
703 parameter update in the direction of the negative gradient for each weight matrix was applied after each trial  
704 (stochastic gradient descent, with a learning rate of 0.005). This was repeated for  $K = 750$  trials. The activity of  
705 the recurrent units ( $\mathbf{x}_t$  in eq. (9)) over all timepoints and trials was collected into a  $N \times T \times K$  tensor for analysis.

#### 706 4.4.6 Mouse spatial navigation task

707 We injected 500 nL of AAV2/5-CaMKII $\alpha$ -GCaMP6m into the medial prefrontal cortex (AP: 1.9, ML: 0.95, DV:  
708 2.25, relative to bregma) into mice aged  $\sim 8$  weeks. Approximately one week after virus injection, we installed  
709 glass-bottom stainless steel guide tubes into the prefrontal cortex to enable deep brain optical imaging using a 1  
710 mm diameter GRIN microendoscope (1050-002176, Inscopix). Two weeks following guide tube surgery, we checked  
711 for cellular Ca<sup>2+</sup> signals with a miniaturized fluorescence microscope (nVista HD, Inscopix). Animals with robust  
712 Ca<sup>2+</sup> responses were selected for further behavioral study. Mice selected for behavioral training underwent water  
713 restriction (1 mL per day) to reach  $\sim 85\%$  of their *ad libitum* weight.



714 Mice performed spatial navigation on a custom-built elevated plus maze. The center-to-end arm length of the  
715 maze was 38 cm. By blocking one of the arms with an opaque barrier, the plus maze could be converted into a  
716 T-maze with any of the four arms as the stem. Additional gates on each of the arms (at ~15 cm from the end) could  
717 be used to confine the mouse at the arms. At the end of each arm, a proximity sensor enabled detection of the mouse  
718 and a water spout allowed for reward delivery. The maze was placed in a rectangular housing whose four side walls  
719 were uniquely defined by distinctly patterned curtains.

720 The mice performed 100-150 trials on each session. At the beginning of each trial, the experimenter placed the  
721 mouse in the stem arm of the T-maze with the corresponding gate closed. After 5 s holding time in the stem arm,  
722 the “start” gate was opened to allow the mouse to run to either end of the T-maze. Once the mouse was detected in  
723 one of the ends, the “end” gate was closed behind the mouse to confine it in the chosen arm for another 5 s. If the  
724 mouse’s choice was consistent with the reward contingency, 5-10  $\mu\text{L}$  of water was delivered to the spout. Trained  
725 mice typically made the run in 2 s; hence the typical trial was ~12 s long. At the end of each trial, the experimenter  
726 retrieved the mouse and wiped the maze with ethanol.

727 During trials, we recorded prefrontal  $\text{Ca}^{2+}$  activity at 20 Hz using the miniature fluorescence microscope. An  
728 overhead camera (DMK 23FV024, The Imaging Source) mounted above the behavioral apparatus synchronously  
729 recorded the position of the mouse on the maze. To extract cells and their activity traces from the  $\text{Ca}^{2+}$  movies,  
730 we followed a procedure previously described in [5], and we then tracked individual neurons across sessions using  
731 previously described methods [104].

732 The tensor representation of neural activity requires that the number of samples within each trial be the same  
733 for all trials, whereas the mice took a variable amount of time to complete each trial. Hence, we used the largest  
734 number of intra-trial samples common to all trials (or, equivalently, the duration of the shortest trial) as the length  
735 of the intra-trial time dimension. We chose to temporally align trials to the end of each trial, because the mice  
736 showed more consistent behavior across trials at the ends (i.e. approaching the choice arm and consuming reward,  
737 if available) rather than the beginnings (where mice could take variable time to initiate motion after opening of the  
738 start gate).

739 Along the trial dimension of the tensor, we simply concatenated trials across days. However, all  $\text{Ca}^{2+}$  activity  
740 traces were normalized to the range [0, 1] based on the cell’s minimum and maximum fluorescence values on each  
741 day. This normalization procedure was crucial for forming across-day tensors, since the exact amplitude of a  $\text{Ca}^{2+}$   
742 trace was dependent on precise, micron-level axial positioning of the microscope — which could vary randomly from  
743 session to session.

#### 744 4.4.7 Primate BMI task

745 The monkey’s hands were restrained for the full duration of the experiment. Voltage signals were band-pass filtered  
746 from each electrode (250 Hz - 7.5 KHz). A spike was recorded whenever these filtered signals crossed below a  
747 threshold of -4.5 times the root-mean-square voltage.

748 The neural recordings from PMd and M1 were used jointly and without distinction to train a BMI decoder by  
749 the recalibrated feedback-intention trained Kalman filter (ReFIT) procedure [61]. At the start of each session, the  
750 monkey observed 600 trials of automated cursor movements from the center of the workspace to one of 8 radially  
751 arranged targets at a distance of 12 cm. During these observation trials, the cursor velocity began at 8 cm/s, and  
752 increased by 2 cm/s every 200 trials. Under the premise that the monkey is imagining the intended task during these  
753 observation trials, we used the neural activity and cursor kinematics to fit a Kalman filter decoder. The velocity  
754 gain of the decoder was calibrated by the experimenter to help the monkey achieve fast reaches (improved by high  
755 gain) while still holding the cursor steady (improved by low gain).

756 The monkey then executed instructed-delay cursor movements to indicated radial target locations, before return-  
757 ing to the center position and repeating the cursor movement to another target. This essential behavioral paradigm

758 has been previously described [105]. Each target position and the center position were indicated on the screen.  
 759 Monkeys started by holding the cursor on the central target continuously for 500 ms. After a randomized delay  
 760 (sampled uniformly from 400-800 ms), monkeys moved the cursor within a 4 x 4 cm acceptance window of the cued  
 761 target. This target also had to be held continuously for 500 ms. The target changed color to signify the hold period.  
 762 If the cursor left the acceptance window, the timer was reset, but the trial was not immediately failed. Monkeys had  
 763 2 s to acquire the target. Success was accompanied with a liquid reward, along with a success tone. Failure resulted  
 764 in no reward, and a failure tone. The center target was then presented, which the monkeys also had to acquire and  
 765 hold.

766 For our analysis, we collected the non-sorted spiking activity of all  $N = 192$  multiunit recordings during all center  
 767 to outward cursor reaches (reaches back to the center were not analyzed). Spike times were aligned to the end of the  
 768 delay period ( $t = 0$ ) and ended at the time of first target acquisition or after two seconds had elapsed and the target  
 769 was still not required. The data tensor was zero padded to ensure a consistent trial length of two seconds. Data  
 770 were smoothed within each trial with a Gaussian filter with a standard deviation of 50 ms (same as in [34]). Using  
 771 a smaller filter did not qualitatively effect the trial factors extracted by TCA, but resulted in less smooth temporal  
 772 factors.

## 773 4.5 Quantification and Statistical Analysis

### 774 4.5.1 TCA model analysis

775 Unlike PCA (but similar to ICA and other methods), TCA needs to be iteratively optimized to minimize a cost  
 776 function. In theory, each optimization run may converge to a sub-optimal local minimum. Additionally, the number  
 777 of components in the model can affect the final result [63]. This is different from PCA where the largest components  
 778 do not change by adding additional components (a consequence of the Eckert-Young theorem; [106]). Thus, we fit  
 779 all TCA models from multiple initial parameters and with different numbers of low-dimensional factors. We then  
 780 inspect this ensemble of models for a consistent and interpretable summary of the data.

781 The most basic metric to compare models is the squared reconstruction error, since this is what TCA aims to  
 782 minimize. For interpretability, we normalize the reconstruction error on a scale of zero to one:

$$\frac{\|\mathcal{X} - \hat{\mathcal{X}}\|_F^2}{\|\mathcal{X}\|_F^2}. \quad (11)$$

783 We typically visualize reconstruction error as a function of the number of model components (see, e.g., fig. 4g), which  
 784 we call an “error plot.”

As discussed in section 4.4.2, TCA is invariant to permutations and rescalings of the factors. In PCA, the components are often normalized to unit Euclidean length and ordered by variance explained. An analogous procedure exists for TCA [33]. First, rescale the columns of  $\mathbf{W}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$  to be unit length, and absorb these scalings into  $\lambda_r$  for each component  $r$ . Then the estimate of the data becomes:

$$\hat{x}_{ntk} = \sum_{r=1}^R \lambda_r w_n^r b_t^r a_k^r,$$

785 If desired, the components can be sorted by decreasing  $\lambda_r$ .

786 To quantify the similarity of two fitted TCA models, we used a similarity score based on the angles between  
 787 latent factors [107]. Formally, for two TCA models,  $\{\mathbf{W}, \mathbf{B}, \mathbf{A}\}$  and  $\{\mathbf{W}', \mathbf{B}', \mathbf{A}'\}$ , the similarity score is:

$$\min_{\omega \in \Omega} \frac{1}{R} \sum_{r=1}^R \left[ \left( 1 - \frac{|\lambda_r - \lambda_{\omega(r)}|}{\max(\lambda_r, \lambda_{\omega(r)})} \right) (\mathbf{w}_r^T \mathbf{w}'_{\omega(r)} \cdot \mathbf{b}_r^T \mathbf{b}'_{\omega(r)} \cdot \mathbf{a}_r^T \mathbf{a}'_{\omega(r)}) \right] \quad (12)$$

788 Where  $\Omega$  denotes the set of all permutations of the factors, and  $\omega$  is a particular permutation. For example, for a three  
789 component model ( $R = 3$ ) the score is computed for all possible permutations,  $\omega = \{1, 2, 3\}$ ,  $\{2, 1, 3\}$ ,  $\{3, 2, 1\}$ , and  
790  $\{3, 1, 2\}$ , and the lowest score is taken. For TCA models with more than 10 components, enumerating all permutations  
791 can be computationally prohibitive. In these cases we match factors in a greedy fashion to identify a permutation  
792 that provides a good (though not certifiably optimal) alignment of the models. Note that this measurement of  
793 model similarity is quite severe, since the distance of each pair of factors are multiplied – if any single dimension is  
794 orthogonal, For our datasets, models with similarity scores above 0.8 were qualitatively similar and led to similar  
795 quantitative results in post-hoc analyses. Models with similarity scores within the 0.6 – 0.8 range also appeared quite  
796 similar in our applications.

#### 797 4.5.2 Mouse spatial navigation task

798 We quantified the *dimensionality* of a single neuron across trials by the following quantity:

$$\dim(\mathbf{X}^{(n)}) = \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2}, \quad (13)$$

799 where  $\lambda_i$  are the eigenvalues of the covariance matrix; i.e.,  $\lambda_i = \sigma_i^2$  where  $\sigma_i$  are the singular values of  $\mathbf{X}^{(n)}$ , which is  
800 a  $K \times T$  matrix holding the activity of neuron  $n$  across all trials. This is a continuous measure of dimensionality used  
801 in condensed matter physics, and was previously applied to analyze neural circuits [108]. For example, (13) reduces  
802 to  $N$  when all the  $\lambda_i$  are evenly distributed and take the same value, and reduces to 1 if only one  $\lambda_i$  is nonzero. For  
803 uneven distributions of  $\lambda_i$ , this measure sensibly interpolates between these two extremes.

#### 804 4.5.3 Primate BMI task

805 In fig. 7, statistical tests on the mean preferred angle of TCA components were performed using PyCircStat (<https://github.com/circstat/pycircstat>). Statistical tests on Spearman’s rho were computed using the SciPy statistics  
806 module (<https://docs.scipy.org/doc/scipy-0.14.0/reference/stats.html>).  
807

## 808 4.6 Experimental model and subject details

### 809 4.6.1 Mice

810 The Stanford Administrative Panel on Laboratory Animal Care approved all mouse procedures. We used male  
811 C57BL/6 mice, aged ~8 weeks at start. Throughout the entire protocol, we monitored the weight daily and looked  
812 for signs of distress (e.g., unkempt fur, hunched posture). Mice were habituated to experimenter handling and the  
813 behavioral apparatus for ~2 weeks prior to the five day behavioral protocol.

### 814 4.6.2 Monkey

815 Recordings were made from motor cortical areas of an adult male monkey, R (*Macaca mulatta*, 15 kg, 12 years old),  
816 performing an instructed delay cursor task. The monkey had two chronic 96-electrode arrays (1 mm electrodes,  
817 spaced 400  $\mu\text{m}$  apart, Blackrock Microsystems), one implanted in the dorsal aspect of the premotor cortex (PMd)  
818 and one implanted in the primary motor cortex (M1). The arrays were implanted 5 years prior to these experiments.  
819 Animal protocols were approved by the Stanford University Institutional Animal Care and Use Committee.

## 820 Acknowledgments

821 The authors thank Jeff Seely (Cognescent Corporation) and Casey Battaglini (Georgia Tech) for discussions pertain-  
822 ing to this work. A.H.W. was supported by the Department of Energy Computational Science Graduate Fellowship  
823 program. T.H.K. was supported by a Stanford Graduate Fellowship in Science & Engineering. F.W. was supported  
824 by a National Science Foundation Graduate Research Fellowship. S.V. was supported by a National Science Found-  
825 ation Graduate Research Fellowship, a Ric Weiland Stanford Graduate Fellowship, National Institutes of Health  
826 F31 training grant, and the Stanford Center for Mind, Brain and Computation. K.V.S. was supported by the US  
827 National Institutes of Health Director’s Pioneer Award 8DP1HD075623, US National Institutes of Health Director’s  
828 Transformative Research Award (TR01) from the NIMH #5R01MH09964703, Defense Advanced Research Projects  
829 Agency NeuroFAST award from BTO #W911NF-14-2-0013, the Simons Foundation, and the Howard Hughes Med-  
830 ical Institute. M.S. was supported by the National Institutes of Health (#1R21NS104833-01), the National Science  
831 Foundation (#1707261), and the Howard Hughes Medical Institute. Work by T.G.K. was supported by the U.S.  
832 Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics  
833 program in a grant to Sandia National Laboratories, a multimission laboratory managed and operated by National  
834 Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc.,  
835 for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA-0003525.  
836 S.G. was supported by the Burroughs Wellcome Foundation, the McKnight Foundation, the James S. McDonnell  
837 Foundation, the Simons Foundation, and the Office of Naval Research.

## 838 References

- 839 [1] JA Kleim, S Barbay, and RJ Nudo. “Functional Reorganization of the Rat Motor Cortex Following Motor  
840 Skill Learning”. *J Neurophysiol* 80.6 (1998), pp. 3321–3325.
- 841 [2] CT Law and JI Gold. “Neural correlates of perceptual learning in a sensory-motor, but not a sensory, cortical  
842 area”. *Nat Neurosci* 11.4 (2008), pp. 505–513.
- 843 [3] AJ Peters, SX Chen, and T Komiyama. “Emergence of reproducible spatiotemporal activity during motor  
844 learning”. *Nature* 510.7504 (2014), pp. 263–267.
- 845 [4] A Marblestone, B Zamft, Y Maguire, M Shapiro, T Cybulski, J Glaser, D Amodei, PB Stranges, R Kalhor, D  
846 Dalrymple, D Seo, E Alon, M Maharbiz, J Carmena, J Rabaey, E Boyden\*\*, G Church\*\*, and K Kording\*\*. “Physical principles for scalable neural recording”. *Front Comput Neurosci* 7 (2013), p. 137.  
847
- 848 [5] TH Kim, Y Zhang, J Lecoq, JC Jung, J Li, H Zeng, CM Niell, and MJ Schnitzer. “Long-Term Optical Access  
849 to an Estimated One Million Neurons in the Live Mouse Cortex”. *Cell Reports* 17.12 (2016), pp. 3385–3394.
- 850 [6] JP Seymour, F Wu, KD Wise, and E Yoon. “State-of-the-art MEMS and microsystem tools for brain research”.  
851 *Microsys Nanoeng* 3 (2017),
- 852 [7] M Pachitariu, C Stringer, M Dipoppa, S Schröder, LF Rossi, H Dagleish, M Carandini, and KD Harris.  
853 “Suite2p: beyond 10,000 neurons with standard two-photon microscopy”. *bioRxiv* (2017).
- 854 [8] MZ Lin and MJ Schnitzer. “Genetically encoded indicators of neuronal activity”. *Nat Neurosci* 19.9 (2016),  
855 pp. 1142–1153.
- 856 [9] H Lütcke, DJ Margolis, and F Helmchen. “Steady or changing? Long-term monitoring of neuronal population  
857 activity”. *Trends in Neurosciences* 36.7 (2013), pp. 375–384.
- 858 [10] AK Dhawale, R Poddar, E Kopelowitz, V Normand, S Wolff, and B Olveczky. “Automated long-term recording  
859 and analysis of neural activity in behaving animals”. *bioRxiv* (2016).

- 860 [11] R Chen, A Canales, and P Anikeeva. “Neural recording and modulation technologies”. *Nature Reviews Ma-*  
861 *terials* 2 (2017),
- 862 [12] JP Cunningham and BM Yu. “Dimensionality reduction for large-scale neural recordings”. *Nat Neurosci* 17.11  
863 (2014), pp. 1500–1509.
- 864 [13] P Gao and S Ganguli. “On simplicity and complexity in the brave new world of large-scale neuroscience”.  
865 *Curr Opin Neurbiol* 32 (2015), pp. 148–155.
- 866 [14] MB Ahrens, JM Li, MB Orger, DN Robson, AF Schier, F Engert, and R Portugues. “Brain-wide neuronal  
867 dynamics during motor adaptation in zebrafish”. *Nature* 485.7399 (2012), pp. 471–477.
- 868 [15] MM Churchland, JP Cunningham, MT Kaufman, JD Foster, P Nuyujukian, SI Ryu, and KV Shenoy. “Neural  
869 population dynamics during reaching”. *Nature* 487.7405 (2012), pp. 51–56.
- 870 [16] BM Yu, JP Cunningham, G Santhanam, SI Ryu, KV Shenoy, and M Sahani. “Gaussian-Process Factor  
871 Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity”. *J Neurophysiol* 102.1  
872 (2009), pp. 614–635.
- 873 [17] Y Gao, EW Archer, L Paninski, and JP Cunningham. “Linear dynamical neural population models through  
874 nonlinear embeddings”. *Advances in Neural Information Processing Systems* 29. Ed. by DD Lee, M Sugiyama,  
875 UV Luxburg, I Guyon, and R Garnett. Curran Associates, Inc., 2016, pp. 163–171.
- 876 [18] C Pandarinath, DJ O’Shea, J Collins, R Jozefowicz, SD Stavisky, JC Kao, EM Trautmann, MT Kaufman,  
877 SI Ryu, LR Hochberg, JM Henderson, KV Shenoy, LF Abbott, and D Sussillo. “Inferring single-trial neural  
878 population dynamics using sequential auto-encoders”. *bioRxiv* (2017).
- 879 [19] BB Averbeck, PE Latham, and A Pouget. “Neural correlations, population coding and computation”. *Nat*  
880 *Rev Neurosci* 7.5 (2006), pp. 358–366.
- 881 [20] MR Cohen and JHR Maunsell. “A Neuronal Population Measure of Attention Predicts Behavioral Performance  
882 on Individual Trials”. *J Neurosci* 30.45 (2010), pp. 15241–15253.
- 883 [21] MR Cohen and JHR Maunsell. “When Attention Wanders: How Uncontrolled Fluctuations in Attention Affect  
884 Performance”. *J Neurosci* 31.44 (2011), pp. 15802–15806.
- 885 [22] RLT Goris, JA Movshon, and EP Simoncelli. “Partitioning neuronal variability”. *Nat Neurosci* 17.6 (2014),  
886 pp. 858–865.
- 887 [23] K Ganguly and JM Carmena. “Emergence of a Stable Cortical Map for Neuroprosthetic Control”. *PLOS Biol*  
888 7.7 (2009), pp. 1–13.
- 889 [24] KB Clancy, AC Koralek, RM Costa, DE Feldman, and JM Carmena. “Volitional modulation of optically  
890 recorded calcium signals during neuroprosthetic learning”. *Nat Neurosci* 17.6 (2014), pp. 807–809.
- 891 [25] MJ Siniscalchi, V Phoumthipphavong, F Ali, M Lozano, and AC Kwan. “Fast and slow transitions in frontal  
892 ensemble activity during flexible sensorimotor behavior”. *Nat Neurosci* advance online publication (2016),
- 893 [26] LN Driscoll, NL Pettit, M Minderer, SN Chettih, and CD Harvey. “Dynamic Reorganization of Neuronal  
894 Activity Patterns in Parietal Cortex”. *Cell* (2017).
- 895 [27] JD Carroll and JJ Chang. “Analysis of individual differences in multidimensional scaling via an n-way gener-  
896 alization of “Eckart-Young” decomposition”. *Psychometrika* 35.3 (1970), pp. 283–319.
- 897 [28] RA Harshman. “Foundations of the PARAFAC procedure: Models and conditions for an explanatory multi-  
898 modal factor analysis”. *UCLA Working Papers in Phonetics* 16 (1970), pp. 1–84.
- 899 [29] JB Kruskal. “Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic  
900 complexity and statistics”. *Linear Algebra and its Applications* 18.2 (1977), pp. 95–138.

- 901 [30] E Salinas and P Thier. “Gain Modulation: A Major Computational Principle of the Central Nervous System”.  
902 *Neuron* 27.1 (2000), pp. 15–21.
- 903 [31] M Carandini and DJ Heeger. “Normalization as a canonical neural computation”. *Nat Rev Neurosci* 13.1  
904 (2012), pp. 51–62.
- 905 [32] K Britten, M Shadlen, W Newsome, and J Movshon. “The analysis of visual motion: a comparison of neuronal  
906 and psychophysical performance”. *J Neurosci* 12.12 (1992), pp. 4745–4765.
- 907 [33] TG Kolda and BW Bader. “Tensor Decompositions and Applications”. *SIAM Review* 51.3 (2009), pp. 455–  
908 500.
- 909 [34] D Kobak, W Brendel, C Constantinidis, CE Feierstein, A Kepecs, ZF Mainen, XL Qi, R Romo, N Uchida,  
910 and CK Machens. “Demixed principal component analysis of neural population data”. *eLife* 5 (2016), e10989.
- 911 [35] I Dean, NS Harper, and D McAlpine. “Neural population coding of sound level adapts to stimulus statistics”.  
912 *Nat Neurosci* 8.12 (2005), pp. 1684–1689.
- 913 [36] CM Niell and MP Stryker. “Modulation of visual responses by behavioral state in mouse visual cortex”.  
914 *Neuron* 65.4 (2010), pp. 472–479.
- 915 [37] HK Kato, SN Gillet, AJ Peters, JS Isaacson, and T Komiyama. “Parvalbumin-Expressing Interneurons Lin-  
916 early Control Olfactory Bulb Output”. *Neuron* 80.5 (2013), pp. 1218–1231.
- 917 [38] FS Chance, L Abbott, and AD Reyes. “Gain modulation from background synaptic input”. *Neuron* 35.4  
918 (2002), pp. 773–782.
- 919 [39] SA Prescott and Y De Koninck. “Gain control of firing rate by shunting inhibition: Roles of synaptic noise  
920 and dendritic saturation”. *Proc Natl Acad Sci USA* 100.4 (2003), pp. 2076–2081.
- 921 [40] JA Cardin, LA Palmer, and D Contreras. “Cellular mechanisms underlying stimulus-dependent gain modu-  
922 lation in primary visual cortex neurons in vivo”. *Neuron* 59.1 (2008), pp. 150–160.
- 923 [41] FR Fernandez and JA White. “Gain control in CA1 pyramidal cells using changes in somatic conductance”.  
924 *J Neurosci* 30.1 (2010), pp. 230–241.
- 925 [42] AJ Bell and TJ Sejnowski. “An information-maximization approach to blind separation and blind deconvol-  
926 ution”. *Neural computation* 7.6 (1995), pp. 1129–1159.
- 927 [43] K Funahashi and Y Nakamura. “Approximation of dynamical systems by continuous time recurrent neural  
928 networks”. *Neural Networks* 6.6 (1993), pp. 801–806.
- 929 [44] A Graves, A r. Mohamed, and G Hinton. “Speech recognition with deep recurrent neural networks”. *2013  
930 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 6645–6649.
- 931 [45] V Mante, D Sussillo, KV Shenoy, and WT Newsome. “Context-dependent computation by recurrent dynamics  
932 in prefrontal cortex”. *Nature* 503.7474 (2013), pp. 78–84.
- 933 [46] R Laje and DV Buonomano. “Robust timing and motor patterns by taming chaos in recurrent neural net-  
934 works”. *Nat Neurosci* 16.7 (2013), pp. 925–933.
- 935 [47] HF Song, GR Yang, and XJ Wang. “Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive  
936 Tasks: A Simple and Flexible Framework”. *PLOS Comput Biol* 12.2 (2016), pp. 1–30.
- 937 [48] HF Song, GR Yang, and XJ Wang. “Reward-based training of recurrent neural networks for cognitive and  
938 value-based tasks”. *eLife* 6 (2017). Ed. by TE Behrens, e21492.
- 939 [49] D Sussillo. “Neural circuits as computational dynamical systems”. *Curr Opin Neurobiol* 25 (2014), pp. 156–163.
- 940 [50] E Jonas and KP Kording. “Could a Neuroscientist Understand a Microprocessor?” *PLOS Comput Biol* 13.1  
941 (2017), pp. 1–24.

- 942 [51] D Sussillo and O Barak. “Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent  
943 Neural Networks”. *Neural Comput* 25.3 (2013), pp. 626–649.
- 944 [52] PJ Werbos. “Backpropagation through time: what it does and how to do it”. *Proceedings of the IEEE* 78.10  
945 (1990), pp. 1550–1560.
- 946 [53] HS Seung. “How the brain keeps the eyes still”. *Proc Natl Acad Sci USA* 93.23 (1996), pp. 13339–13344.
- 947 [54] KK Ghosh, LD Burns, ED Cocker, A Nimmerjahn, Y Ziv, AE Gamal, and MJ Schnitzer. “Miniaturized  
948 integration of a fluorescence microscope”. *Nat Meth* 8.10 (2011), pp. 871–878.
- 949 [55] EL Rich and M Shapiro. “Rat Prefrontal Cortical Neurons Selectively Code Strategy Switches”. *J Neurosci*  
950 29.22 (2009), pp. 7208–7219.
- 951 [56] D Durstewitz, NM Vittoz, SB Floresco, and JK Seamans. “Abrupt Transitions between Prefrontal Neural  
952 Ensemble States Accompany Behavioral Transitions during Rule Learning”. *Neuron* 66.3 (2010), pp. 438–448.
- 953 [57] JD Wallis and SW Kennerley. “Heterogeneous reward signals in prefrontal cortex”. *Curr Opin Neurobiol* 20.2  
954 (2010), pp. 191–198.
- 955 [58] M Rigotti, O Barak, MR Warden, XJ Wang, ND Daw, EK Miller, and S Fusi. “The importance of mixed  
956 selectivity in complex cognitive tasks”. *Nature* 497.7451 (2013), pp. 585–590.
- 957 [59] MG Stokes, M Kusunoki, N Sigala, H Nili, D Gaffan, and J Duncan. “Dynamic Coding for Cognitive Control  
958 in Prefrontal Cortex”. *Neuron* 78.2 (2013), pp. 364–375.
- 959 [60] DD Lee and HS Seung. “Learning the parts of objects by non-negative matrix factorization”. *Nature* 401.6755  
960 (1999), pp. 788–791.
- 961 [61] V Gilja, P Nuyujukian, CA Chestek, JP Cunningham, BM Yu, JM Fan, MM Churchland, MT Kaufman, JC  
962 Kao, SI Ryu, and KV Shenoy. “A high-performance neural prosthesis enabled by control algorithm design”.  
963 *Nat Neurosci* 15.12 (2012), pp. 1752–1757.
- 964 [62] A Georgopoulos, J Kalaska, R Caminiti, and J Massey. “On the relations between the direction of two-  
965 dimensional arm movements and cell discharge in primate motor cortex”. *J Neurosci* 2.11 (1982), pp. 1527–  
966 1537.
- 967 [63] TG Kolda. “A Counterexample to the Possibility of an Extension of the Eckart–Young Low-Rank Approx-  
968 imation Theorem for the Orthogonal Rank Tensor Decomposition”. *SIAM J Matrix Anal Appl* 24.3 (2003),  
969 pp. 762–767.
- 970 [64] LH Lim and P Comon. “Nonnegative approximations of nonnegative tensors”. *J Chemometrics* 23.7-8 (2009),  
971 pp. 432–441.
- 972 [65] CJ Hillar and LH Lim. “Most Tensor Problems Are NP-Hard”. *J. ACM* 60.6 (2013), 45:1–45:39.
- 973 [66] L Omberg, GH Golub, and O Alter. “A tensor higher-order singular value decomposition for integrative  
974 analysis of DNA microarray data from different studies”. *Proc Natl Acad Sci USA* 104.47 (2007), pp. 18371–  
975 18376.
- 976 [67] DA Cartwright, SM Brady, DA Orlando, B Sturmfels, and PN Benfey. “Reconstructing spatiotemporal gene  
977 expression data from partial observations”. *Bioinformatics* 25.19 (2009), p. 2581.
- 978 [68] V Hore, A Vinuela, A Buil, J Knight, MI McCarthy, K Small, and J Marchini. “Tensor decomposition for  
979 multiple-tissue gene expression experiments”. *Nat Genet* advance online publication (2016),
- 980 [69] M Mørup, LK Hansen, CS Herrmann, J Parnas, and SM Arnfred. “Parallel factor analysis as an exploratory  
981 tool for wavelet transformed event-related EEG”. *NeuroImage* 29.3 (2006), pp. 938–947.

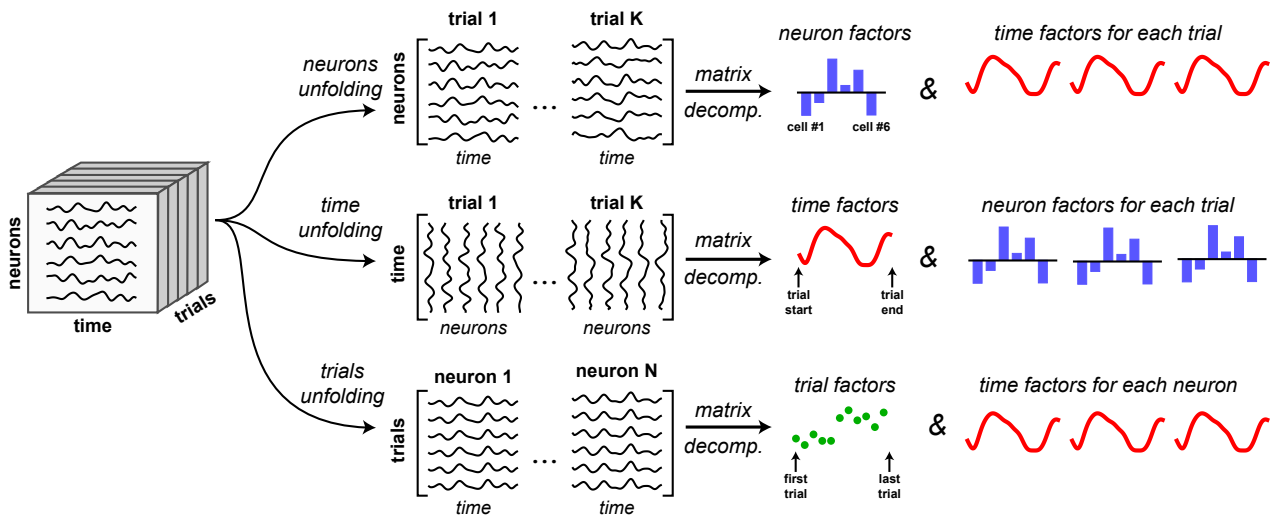
- 982 [70] E Acar, C Aykut-Bingol, H Bingol, R Bro, and B Yener. “Multiway analysis of epilepsy tensors”. *Bioinformatics* 23.13 (2007), pp. i10–i18.  
983
- 984 [71] F Cong, QH Lin, LD Kuang, XF Gong, P Astikainen, and T Ristaniemi. “Tensor decomposition of EEG  
985 signals: A brief review”. *J Neurosci Methods* 248 (2015), pp. 59–69.
- 986 [72] B Hunyadi, P Dupont, W Van Paesschen, and S Van Huffel. “Tensor decompositions and data fusion in  
987 epileptic electroencephalography and functional magnetic resonance imaging data”. *Wiley Interdisciplinary  
988 Reviews: Data Mining and Knowledge Discovery* 7.1 (2017), e1197–n/a.
- 989 [73] AH Andersen and WS Rayens. “Structure-seeking multilinear methods for the analysis of fMRI data”. *NeuroImage* 22.2 (2004), pp. 728–739.  
990
- 991 [74] JS Seely, MT Kaufman, SI Ryu, KV Shenoy, JP Cunningham, and MM Churchland. “Tensor Analysis Reveals  
992 Distinct Population Structure that Parallels the Different Computational Roles of Areas M1 and V1”. *PLoS  
993 Comput Biol* 12.11 (2016), pp. 1–34.
- 994 [75] MB Ahrens, JF Linden, and M Sahani. “Nonlinearities and Contextual Influences in Auditory Cortical Re-  
995 sponses Modeled with Multilinear Spectrotemporal Methods”. *J Neurosci* 28.8 (2008), pp. 1929–1942.
- 996 [76] NC Rabinowitz, RL Goris, M Cohen, and EP Simoncelli. “Attention stabilizes the shared gain of V4 popula-  
997 tions”. *eLife* 4 (2015). Ed. by M Carandini, e08998.
- 998 [77] JJ Letzkus, SBE Wolff, EMM Meyer, P Tovote, J Courtin, C Herry, and A Luthi. “A disinhibitory microcircuit  
999 for associative fear learning in the auditory cortex”. *Nature* 480.7377 (2011), pp. 331–335.
- 1000 [78] BV Atallah, W Bruns, M Carandini, and M Scanziani. “Parvalbumin-Expressing Interneurons Linearly Trans-  
1001 form Cortical Responses to Visual Stimuli”. *Neuron* 73.1 (2012), pp. 159–170.
- 1002 [79] H Makino, EJ Hwang, NG Hedrick, and T Komiyama. “Circuit Mechanisms of Sensorimotor Learning”.  
1003 *Neuron* 92.4 (2016), pp. 705–721.
- 1004 [80] RS Sutton and AG Barto. *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge, 1998.
- 1005 [81] AC Smith and EN Brown. “Estimating a State-Space Model from Point Process Observations”. *Neural Comput*  
1006 15.5 (2003), pp. 965–991.
- 1007 [82] JH Macke, L Buesing, JP Cunningham, BM Yu, KV Shenoy, and M Sahani. “Empirical models of spiking  
1008 in neural populations”. *Advances in Neural Information Processing Systems* 24. Ed. by J Shawe-Taylor, RS  
1009 Zemel, PL Bartlett, F Pereira, and KQ Weinberger. Curran Associates, Inc., 2011, pp. 1350–1358.
- 1010 [83] L Buesing, JH Macke, and M Sahani. “Spectral learning of linear dynamics from generalised-linear observa-  
1011 tions with application to neural population data”. *Advances in neural information processing systems*. 2012,  
1012 pp. 1682–1690.
- 1013 [84] JC Kao, P Nuyujukian, SI Ryu, MM Churchland, JP Cunningham, and KV Shenoy. “Single-trial dynamics  
1014 of motor cortex and their applications to brain-machine interfaces”. *Nat Commun* 6 (2015),
- 1015 [85] B Petreska, BM Yu, JP Cunningham, G Santhanam, SI Ryu, KV Shenoy, and M Sahani. “Dynamical seg-  
1016 mentation of single trials from population neural data”. *Advances in Neural Information Processing Systems*  
1017 24. Ed. by J Shawe-Taylor, RS Zemel, PL Bartlett, F Pereira, and KQ Weinberger. Curran Associates, Inc.,  
1018 2011, pp. 756–764.
- 1019 [86] S Linderman, M Johnson, A Miller, R Adams, D Blei, and L Paninski. “Bayesian Learning and Inference  
1020 in Recurrent Switching Linear Dynamical Systems”. *Proceedings of the 20th International Conference on  
1021 Artificial Intelligence and Statistics*. Ed. by A Singh and J Zhu. Vol. 54. Proceedings of Machine Learning  
1022 Research. Fort Lauderdale, FL, USA: PMLR, 2017, pp. 914–922.



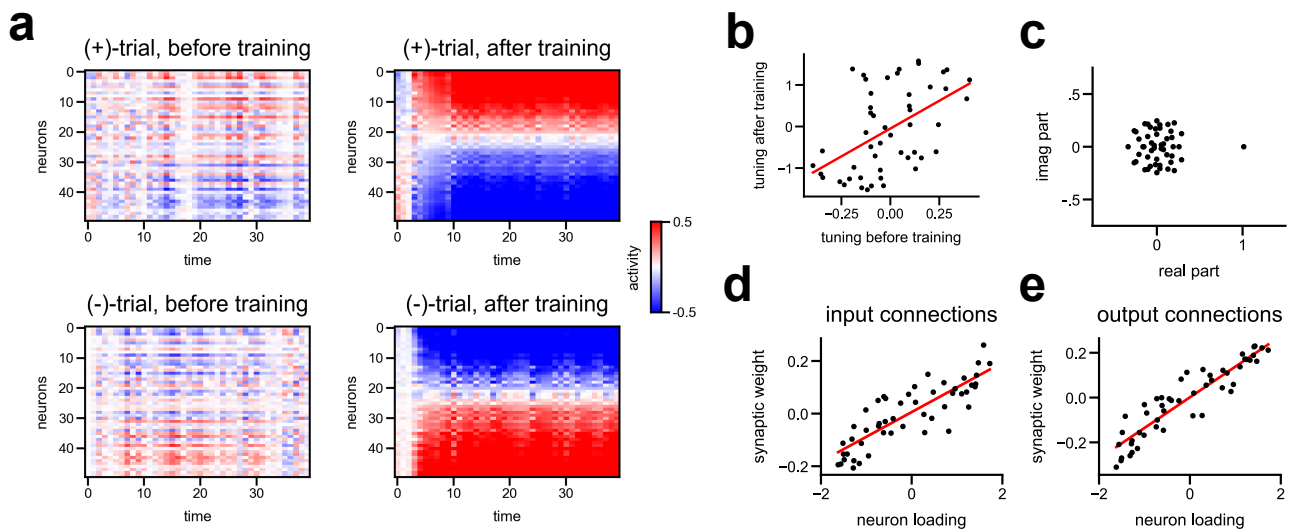
- 1023 [87] Y Zhao and IM Park. “Interpretable Nonlinear Dynamic Modeling of Neural Trajectories”. *Advances in Neural*  
1024 *Information Processing Systems 29*. Ed. by DD Lee, M Sugiyama, UV Luxburg, I Guyon, and R Garnett.  
1025 Curran Associates, Inc., 2016, pp. 3333–3341.
- 1026 [88] E Acar, R Bro, and AK Smilde. “Data Fusion in Metabolomics Using Coupled Matrix and Tensor Factoriza-  
1027 tions”. *Proceedings of the IEEE* 103.9 (2015), pp. 1602–1620.
- 1028 [89] E Jones, T Oliphant, P Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–.
- 1029 [90] JD Hunter. “Matplotlib: A 2D Graphics Environment”. *Computing in Science Engineering* 9.3 (2007), pp. 90–  
1030 95.
- 1031 [91] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss,  
1032 V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. “Scikit-learn:  
1033 Machine Learning in Python”. *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- 1034 [92] TW Chen, TJ Wardill, Y Sun, SR Pulver, SL Renninger, A Baohan, ER Schreiter, RA Kerr, MB Orger, V  
1035 Jayaraman, LL Looger, K Svoboda, and DS Kim. “Ultrasensitive fluorescent proteins for imaging neuronal  
1036 activity”. *Nature* 499.7458 (2013), pp. 295–300.
- 1037 [93] BW Bader, TG Kolda, et al. *MATLAB Tensor Toolbox*. 2017. URL: [www.tensortoolbox.org](http://www.tensortoolbox.org).
- 1038 [94] N Vervliet, O Debals, L Sorber, M Van Barel, and L De Lathauwer. *Tensorlab 3.0*. 2016. URL: [http://www.](http://www.tensorlab.net)  
1039 [tensorlab.net](http://www.tensorlab.net).
- 1040 [95] J Kossaifi, Y Panagakis, and M Pantic. “TensorLy: Tensor Learning in Python”. *ArXiv e-print* (2016).
- 1041 [96] R Bro and S De Jong. “A fast non-negativity-constrained least squares algorithm”. *J Chemometrics* 11.5  
1042 (1997), pp. 393–401.
- 1043 [97] P Paatero. “A weighted non-negative least squares algorithm for three-way ‘PARAFAC’ factor analysis”.  
1044 *Chemometrics and Intelligent Laboratory Systems* 38.2 (1997), pp. 223–242.
- 1045 [98] M Welling and M Weber. “Positive Tensor Factorization”. *Pattern Recognition Letters* 22.12 (2001), pp. 1255–  
1046 1261.
- 1047 [99] P Paatero and U Tapper. “Positive matrix factorization: A non-negative factor model with optimal utilization  
1048 of error estimates of data values”. *Environmetrics* 5.2 (1994), pp. 111–126.
- 1049 [100] EC Chi and TG Kolda. “On Tensors, Sparsity, and Nonnegative Factorizations”. *SIAM J Matrix Anal Appl*  
1050 33.4 (2012), pp. 1272–1299.
- 1051 [101] P Comon, X Luciani, and ALF de Almeida. “Tensor decompositions, alternating least squares and other  
1052 tales”. *J Chemometrics* 23.7-8 (2009), pp. 393–405.
- 1053 [102] SA Vavasis. “On the Complexity of Nonnegative Matrix Factorization”. *SIAM Journal on Optimization* 20.3  
1054 (2010), pp. 1364–1377.
- 1055 [103] J Kim and H Park. “Fast Nonnegative Matrix Factorization: An Active-Set-Like Method and Comparisons”.  
1056 *SIAM Journal on Scientific Computing* 33.6 (2011), pp. 3261–3281.
- 1057 [104] BF Grewe, J Gründemann, LJ Kitch, JA Lecoq, JG Parker, JD Marshall, MC Larkin, PE Jercog, F Grenier,  
1058 JZ Li, A Lüthi, and MJ Schnitzer. “Neural ensemble dynamics underlying a long-term associative memory”.  
1059 *Nature* 543.7647 (2017), pp. 670–675.
- 1060 [105] KV Shenoy, M Sahani, and MM Churchland. “Cortical Control of Arm Movements: A Dynamical Systems  
1061 Perspective”. *Annual Review of Neuroscience* 36.1 (2013), pp. 337–359.
- 1062 [106] C Eckart and G Young. “The approximation of one matrix by another of lower rank”. *Psychometrika* 1.3  
1063 (1936), pp. 211–218.

- 1064 [107] G Tomasi and R Bro. “A comparison of algorithms for fitting the PARAFAC model”. *Computational Statistics*  
1065 *& Data Analysis* 50.7 (2006), pp. 1700–1734.
- 1066 [108] A Litwin-Kumar, KD Harris, R Axel, H Sompolinsky, and LF Abbott. “Optimal Degrees of Synaptic Con-  
1067 nectivity”. *Neuron* 93.5 (2017), 1153–1164.e7.

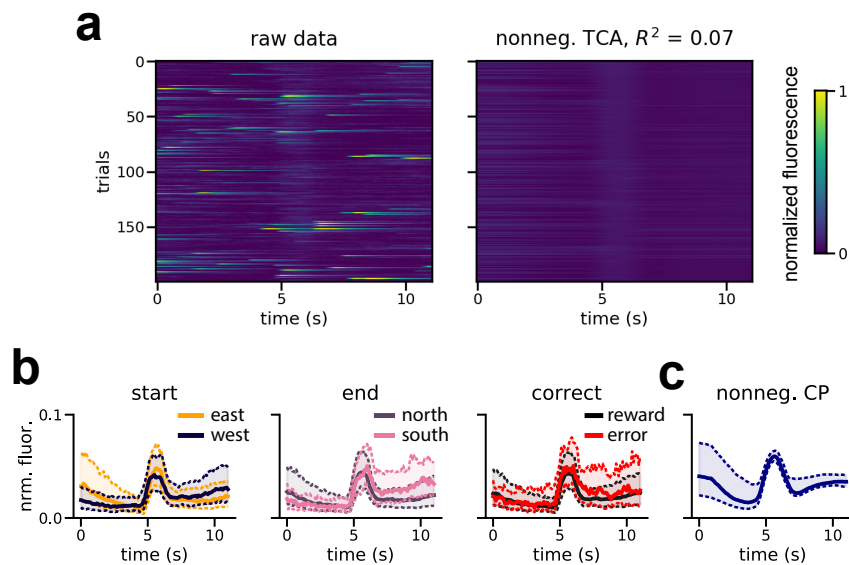
1068 **Figure Supplements**



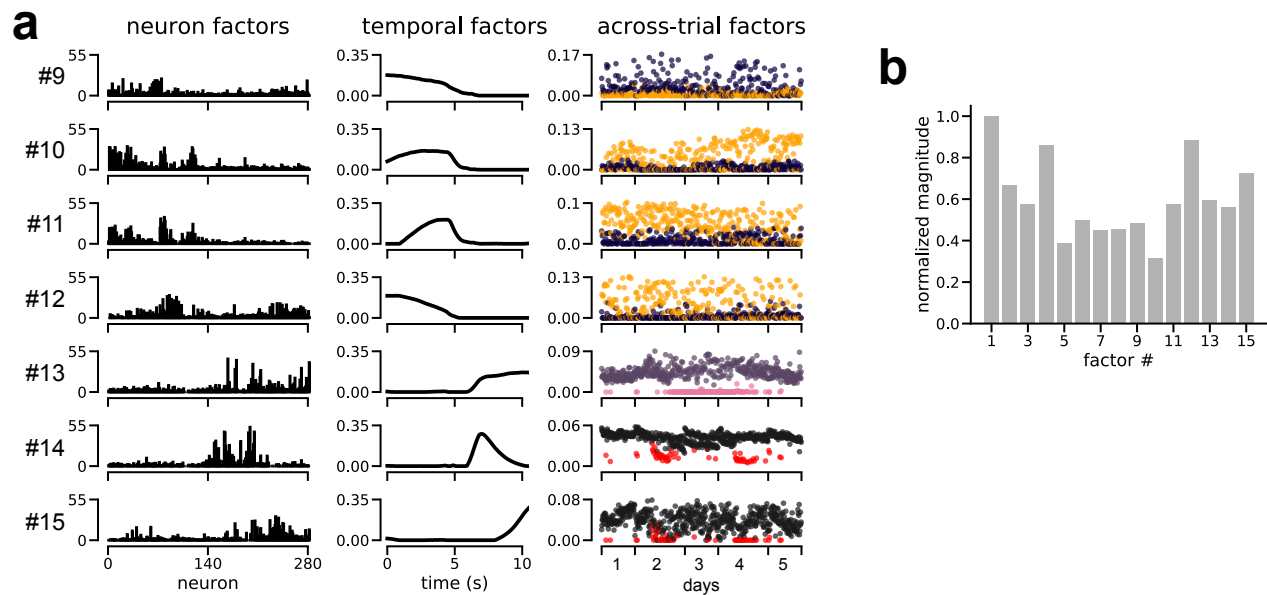
**Figure 2, Supplement 1.** Illustration of *tensor unfolding* for applying matrix decompositions to tensor datasets. A  $N \times T \times K$  dimensional tensor can be reshaped into three different matrices: a “neurons unfolding” with dimensions  $N \times TK$ , a “time unfolding” with dimensions  $T \times NK$ , and a “trials unfolding” with dimensions  $K \times NT$ . Applying PCA or other matrix decomposition methods to each unfolding yields a different set of low-dimensional factors.



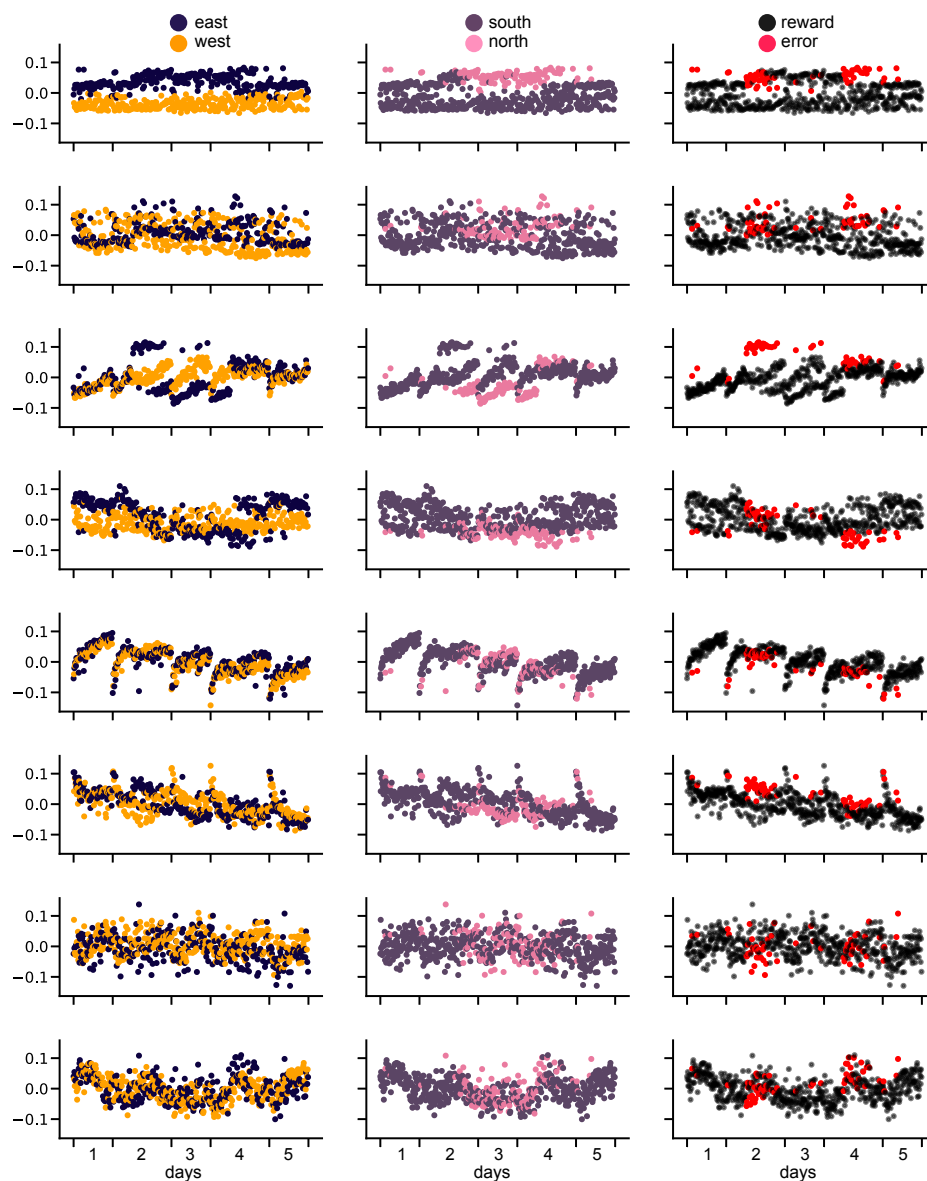
**Figure 3, Supplement 1.** Cell tuning and synaptic connectivity properties in a nonlinear RNN trained on a stimulus discrimination task. **(a)** Activity of all cells on (+)-trials and (-)-trials before and after training. Cells were sorted by the low-dimensional neuron factor,  $w_n^1$ , as in fig. 3e. **(b)** Cell tuning quantified as peak activity on (+)-trials minus peak activity on (-)-trials before and after training (averaged over ten trials). Cells with positive tuning scores are (+)-cells, while cells with negative tuning scores are (-)-cells. The initial tuning was positively correlated with final tuning for each cell. **(c)** Eigenvalues of the synaptic connectivity matrix after training. Similar to the solution in linear networks [53], the connectivity matrix has a single eigenvalue near  $1 + 0i$ ; and all other eigenvalues are small in magnitude. **(d-e)** The neuron factor identified by a 1-component TCA model is positively correlated with the input-to-network synaptic weights **(d)**, and the network-to-output weights **(e)**.



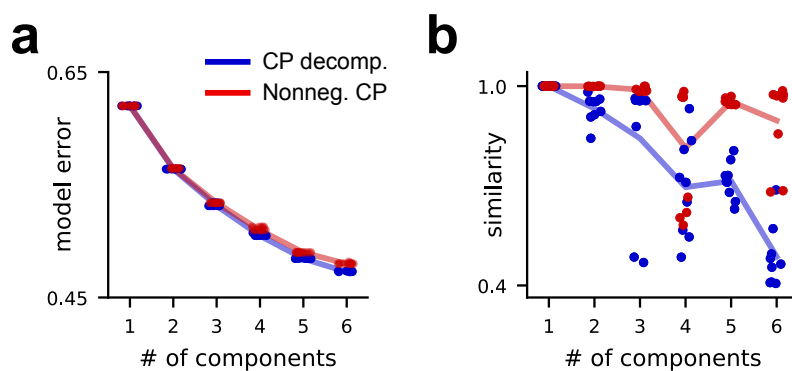
**Figure 4, Supplement 1.** An example cell with low  $R^2$ . **(a)** Raster heatmaps showing the cell’s fluorescence over the 200 most active trials (left), and the estimate of a 15-component nonnegative TCA model on these trials (right). On a small subset of trials the cell is active, but at variable phases of the trial. Note that on the remaining trials, the cell was hardly active at all (not shown). **(b)** Median fluorescence traces averaged over various task variables (start location, end location, and reward delivery). The cell does not, on average, show a preference for any task variable. Dashed lines denote the first and third quartiles of the fluorescence trace. **(c)** Median estimated fluorescence of the 15-component nonnegative TCA model for this cell. The estimate is closely matched to the median firing rates shown in panel b. Dashed lines denote first and third quartiles.



**Figure 5, Supplement 1.** Additional detail on the decomposition of mouse prefrontal cortex dynamics. **(a)** Remaining seven TCA factors from the 15-component decomposition shown in fig. 5. **(b)** The magnitude (Euclidean length) of each factor in the decomposition, a metric analogous to the variance explained by each component (see *Methods*, section 4.5.1).



**Figure 5, Supplement 2.** PCA components in trial-space do not cleanly encode individual task variables, in line with previous observations [34]. Each row shows a principal component, ordered by variance explained. Each column shows a different coloring of that principal component by a different task variable. With few exceptions (notably the top component), any single coloring does not yield a simple interpretation of the component.



**Figure 6, Supplement 1.** Diagnostic plots for TCA models fit to 45° reaches in the primate BMI dataset. **(a)** Scree plot for unconstrained (blue) and nonnegative (red) TCA. As elsewhere in this manuscript, each dot denotes a model fit from different initial parameters, demonstrating that neither model got caught in appreciably sub-optimal local minima during optimization. Nonnegative decomposition provided similar explanatory power to unconstrained decompositions. **(b)** Similarity plot for unconstrained (blue) and nonnegative (red) CP decompositions. As elsewhere in this manuscript, each dot denotes the similarity score between a model and the best-fit model with the same number of components. Nonnegative decomposition had larger similarity scores, suggesting that the latent factors were more reliably identified and less sensitive to initialization.