

Disentangling Mixed Classes of Covariability in Large-Scale Neural Data

Arthur Pellegrino^{†,1,2,*}, Heike Stein^{†,1}, and N Alex Cayco-Gajic^{1,*}

[†]Equal Contribution

¹Laboratoire de Neurosciences Cognitives et Computationnelles, INSERM U960, Département D'Etudes Cognitives,
Ecole Normale Supérieure, PSL University, Paris, France

²Institute for Adaptive and Neural Computation, School of Informatics, University of Edinburgh, Edinburgh,
United Kingdom

*Corresponding Authors: pellegrino.arthur@ed.ac.uk, natasha.cayco.gajic@ens.fr

Abstract

Recent work has argued that large-scale neural recordings are often well described by low-dimensional ‘latent’ dynamics identified using dimensionality reduction. However, the view that task-relevant variability is shared across neurons misses other types of structure underlying behavior, including stereotyped neural sequences or slowly evolving latent spaces. To address this, we introduce a new framework that simultaneously accounts for variability that is shared across neurons, trials, or time. To identify and demix these covariability classes, we develop a new unsupervised dimensionality reduction method for neural data tensors called sliceTCA. In three example datasets, including motor cortical dynamics during a classic reaching task and recent multi-region recordings from the International Brain Laboratory, we show that sliceTCA can capture more task-relevant structure in neural data using fewer components than traditional methods. Overall, our theoretical framework extends the classic view of low-dimensional population activity by incorporating additional classes of latent variables capturing higher-dimensional structure.

1 Introduction

Neural activity varies in relation to fluctuations in the environment, slow changes in circuitry, and heterogeneous cell properties, creating variability across neurons, time, and trials. Recent work has emphasized that trial-to-trial variability is often correlated across large populations of neurons [Cunningham and Yu, 2014], generating low-dimensional representations of sensory or behavioral variables. Indeed, analyzing shared variability across neurons has led to key insights into the information encoded and computations performed by neural circuits [Panzeri et al., 2022, Jazayeri and Ostojic, 2021]. Such findings have driven an increase in the popularity of dimensionality reduction methods, such as principal component analysis (PCA), which seek to capture structure in neural data by identifying covarying population-wide patterns. More recent work has advocated instead for applying tensor-based methods, such as tensor component analysis (TCA), that distinguish between changes in neural dynamics that occur over fast (within-trial) and slow (between-trial) timescales [Williams et al., 2018, Harshman et al., 1970, Carroll and Chang, 1970]. In both of these approaches, neural activity is assumed to be constrained to a low-dimensional neural subspace (defined by a set of latent variables) that is fixed over the course of an experiment.

However, this picture of latent variables fails to account for some forms of shared variability in neural circuits. First, not all population dynamics are described by a fixed covariance structure. For example, many brain areas produce temporal sequences in which the latency of activation varies from neuron to neuron, but which are highly stereotyped across conditions [Seely et al., 2016, Pastalkova et al., 2008, Peters et al., 2014, Okubo et al., 2015, Harvey et al., 2012]. Second, the neural encoding weights for a given sensory stimulus may change over trials due to adaptation, learning, [Hennig et al., 2021], or representational drift [Rule et al., 2019, Driscoll et al., 2017, Schoonover et al., 2021]. Because methods such as PCA or TCA look for covariability across neurons, they may miss additional forms of variability that are instead shared across time or across trials.

To address this, we introduce slice tensor component analysis (sliceTCA), an unsupervised dimensionality reduction method that is able to identify and disentangle latent variables belonging to three different classes

39 of covariability (defined as variability shared across neurons, time, or trials) that are mixed within the
 40 same dataset. This property contrasts sliceTCA from matrix factorizations like PCA which capture a
 41 single covariability class at a time, and from TCA which identifies components constrained to all of them
 42 simultaneously. As a result, we show that sliceTCA can capture more structure in fewer components than
 43 either of these methods. Based on theoretical and practical considerations of the sliceTCA decomposition,
 44 we develop an analysis pipeline for model selection, optimization, and visualization that is implemented in
 45 a readily applicable Python library.

46 After validating our method on simulated data, we illustrate the utility of sliceTCA in three large-scale
 47 neural datasets. First, we demonstrate that different covariability classes encode distinct behaviorally relevant
 48 neural dynamics in motor cortical recordings in non-human primates [Churchland et al., 2012]. Next,
 49 in simultaneous imaging data from cortical and cerebellar populations during a cued motor task [Wagner
 50 et al., 2019], we show that sliceTCA untangles task-relevant manifolds by taking into account covariability
 51 across trials. Finally, we analyze a recent dataset from the International Brain Laboratory [IBL et al., 2022]
 52 and show that sliceTCA disentangles region-specific covariability classes across visual cortex, hippocampus,
 53 thalamus, and the midbrain. We then provide a geometric intuition for how neural population activity
 54 is shaped by latent variables belonging to the three different covariability classes. Together, these results
 55 demonstrate the necessity of extending the traditional view of latent variables and neural covariability to uncover
 56 higher-dimensional latent structure. With sliceTCA, we propose a novel, unsupervised dimensionality
 57 reduction method that uncovers co-existing classes of behaviorally relevant covariability in neural datasets.

58 2 Results

59 2.1 Overview of sliceTCA

60 SliceTCA is an unsupervised dimensionality reduction method that generalizes matrix factorizations, a
 61 class of methods which includes PCA and non-negative matrix factorization (NMF). Matrix factorizations
 62 approximate a data matrix \mathbf{X} as a sum of R components:

$$63 \quad \mathbf{X} \approx \hat{\mathbf{X}} = \sum_{r=1}^R \mathbf{X}^{(r)}. \quad (1)$$

64 In neuroscience, \mathbf{X} is generally a matrix of size $N \times KT$ containing the activity of N neurons recorded over
 65 K trials, each containing T timepoints. Each component $\mathbf{X}^{(r)}$ is a rank-1 matrix defined by a set of R neural
 66 factors describing different activation patterns across the population, and a set of R corresponding temporal
 67 factors describing how the strength of these patterns evolves over the course of experiment (Figure 1a). By
 68 constraining each component to a rank-1 matrix, these methods capture shared variability across neurons.

69 However, this approach to neural dimensionality reduction has two limitations. First, by concatenating
 70 trials together to structure the data into a matrix, they do not distinguish between rapid dynamics within
 71 a trial and slower changes across trials [Williams et al., 2018]. Second, not all population activity is well
 72 described by shared variability across neurons. For example, motor cortical dynamics may be better described
 73 by stereotyped sequences of neural activation which are shared across trials of the same condition [Seely et al.,
 74 2016]. These limitations can be addressed by structuring the data into an $N \times T \times K$ tensor which can
 75 be similarly decomposed following equation 1 into a low-rank tensor approximation. For this, we must
 76 generalize the concept of a rank-1 matrix to tensors. Different definitions of the tensor rank will capture
 77 different forms of structure in the data.

78 Here, we present *sliceTCA*, a novel tensor decomposition that is based on the *slice rank* [Tao and Sawin,
 79 2016] (Methods). A rank-1 matrix is defined as the outer product of two vectors, so that each column of the
 80 matrix is a scaled version of the same column vector (Figure 1a). Similarly, a slice-rank-1 tensor is defined
 81 as the outer product of a vector and a matrix (or “slice”; Figure 1b). Depending on how the tensor is sliced,
 82 these components can capture three different classes of covariability (across neurons, trials, or time).

83 To gain intuition on this, we may consider each slice type separately. First, a neuron-slicing component
 84 is described by a vector of characteristic neural weights and a matrix describing how the temporal dynamics

85 for that component changes across time and trials (Figure 1c). This component therefore captures variability
86 which is shared across neurons, but which is unconstrained across time and trials. This is exactly the same
87 class of covariability that is captured by common applications of matrix factorization methods in which the
88 data tensor is reshaped or ‘unfolded’ into a $N \times KT$ matrix (sometimes referred to as ‘trial-concatenated’
89 matrix factorization, Supplementary Figure 1a). In contrast, the other two slice types lead to different
90 assumptions about the source of covariability in the data. The trial-slicing components instead capture
91 shared variability across trials, e.g., stereotyped neuron-specific temporal dynamics that vary together in
92 amplitude over trials (Figure 1d). Finally, the time-slicing components identify shared variability over time.
93 This could represent common dynamics whose neural encoding weights change from trial to trial, e.g., due
94 to learning, adaptation, or drift (Figure 1e).

95 If only one of these three slice types were fitted, sliceTCA would be equivalent to a matrix factoriza-
96 tion on the respective unfolding of the data tensor (Supplementary Figure 1a-c). Indeed, previous work
97 has argued for performing PCA on different unfoldings of the data tensor to identify the slice type that
98 gives the best approximation [Seely et al., 2016]. Crucially, sliceTCA differs from this approach by fitting
99 all three slice types *simultaneously*, thereby demixing different covariability classes that may be combined
100 within the same dataset (Figure 1f,g). SliceTCA is also related to, yet distinct from TCA (also known as
101 CANDECOMP/PARAFAC or CP decomposition) [Williams et al., 2018, Harshman et al., 1970, Carroll and
102 Chang, 1970]. TCA constrains each component to be described by the outer product of three vectors of
103 neural, trial, and temporal factors, which requires that each component must lie in the intersection of all
104 three covariability classes (Figure 1g; Methods). As a result, sliceTCA is able to capture more structure in
105 the data with fewer components as compared to other methods.

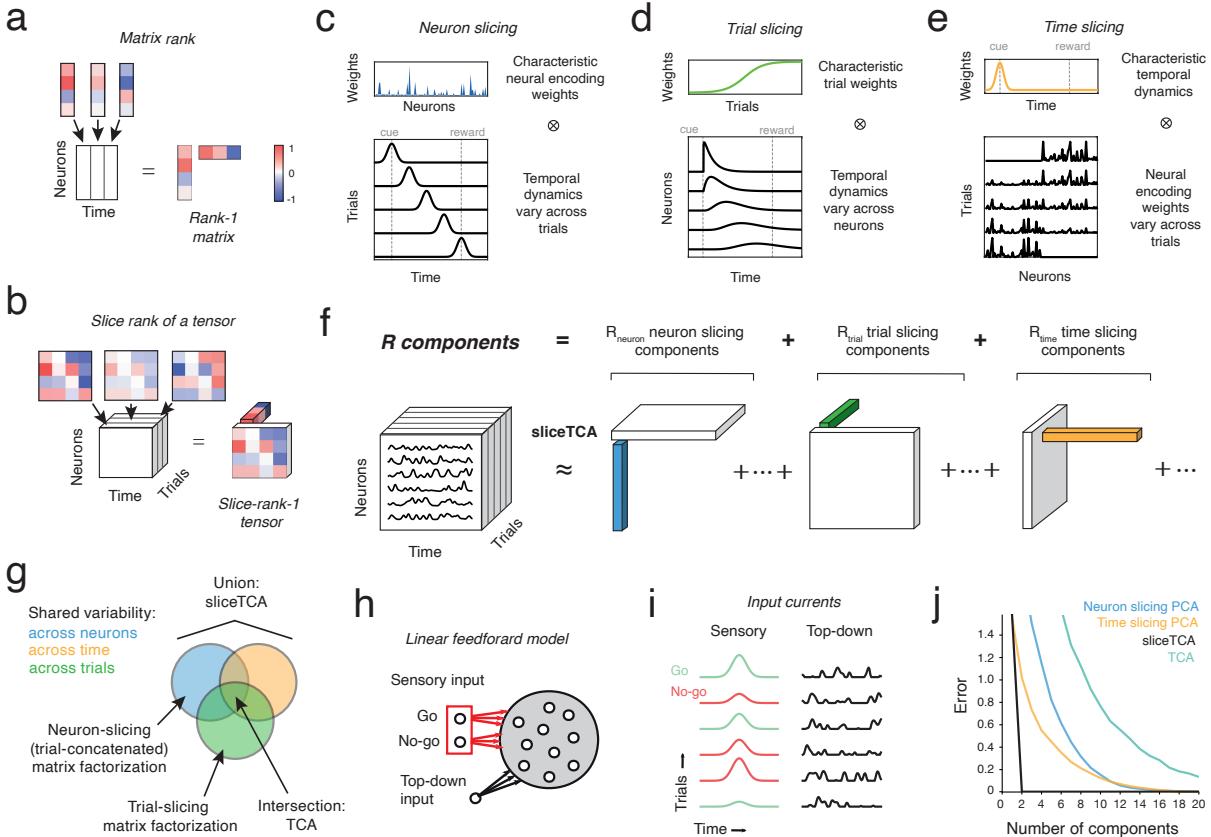


Figure 1: SliceTCA demixes shared variability across neurons, time, and trials. **a.** Schematic representation of a rank 1 matrix. Each column of the matrix is a scaled version of the same vector. Equivalently, the matrix can be written as the outer product of that same column vector and a row vector representing the scaling weights. **b.** Schematic representation of a slice-rank-1 tensor. Each “slice” of the tensor is a scaled version of the same matrix. The tensor can be written as an outer product of that matrix (a “slice”) and a vector representing the scaling weights. **c.** Example of a slice-rank-1 tensor that is an outer product of a neuron loading vector and a time-trial slice. This component represents a latent variable with a fixed neural encoding but whose temporal profile changes from trial to trial. **d.** Example slice-rank-1 tensor that is an outer product of a trial loading vector and a neuron-time slice, representing a latent variable that scales in amplitude over trials but which has neuron specific dynamics within a trial. **e.** Example slice-rank-1 tensor that is an outer product of a time loading vector and a neuron-trial slice, representing a latent variable with a characteristic temporal profile within a trial, but whose neural encoding weights change over trials. **f.** SliceTCA approximates the data tensor as a low-slice-rank approximation. Each component is described as a slice-rank-1 tensor, which can be one of three types: neuron-, trial-, or time-slicing, corresponding to the examples in (c-e). **g.** Schematic of the three covariability classes captured by sliceTCA. Matrix factorization methods like PCA only capture a single covariability class at a time, depending on how the data tensor is unfolded into matrix form. TCA requires the variability captured by each component to be shared across neurons, trials, and time simultaneously (i.e., in the intersection of the three classes). In contrast, sliceTCA represents the union of these three classes. **h.** Schematic of a toy model of perceptual learning during a Go/No-go task. On each trial, a population of linear neurons receives two inputs: (1) sensory input from one of two upstream sources representing the presentation of the Go or No-go stimulus, and (2) top-down modulation representing stimulus-independent factors. Red indicates plastic synapses (Go/No-go weights potentiate/depress over Go/No-go trials, respectively). **i.** Evolution of inputs over trials. Go/No-go inputs increase/decrease in strength over trials, while top-down inputs vary from trial to trial. Since the neurons are linear, their activities will be a linear combination of these two input sources. **j.** Loss (mean squared error) curves as a function of the number of components for different methods.

106 **2.2 Mixed variability in a simple model of perceptual learning**

107 Before applying sliceTCA to data, we first illustrate how mixed covariability classes could emerge in neural
108 circuits. We built a toy model of sensory cortex during a Go/No-go task (Figure 1h). In this model, a
109 population of linear cortical neurons received two sources of input in the context of a Go/No-go task (Figure
110 1i; Supplementary Figure 2a). First, all neurons received sensory input representing the presented stimulus
111 (either Go or No-go). The projection weights were plastic and subject to potentiation or depression (for
112 the Go / No-go stimuli, respectively), in line with evidence of enhanced sensitivity to target stimuli in
113 sensory cortex during perceptual learning [Poort et al., 2015]. The evolution of neural weights over trials
114 was stochastic with heterogeneous learning rates, creating variability across neurons (Supplementary Figure
115 2b; Methods). Second, all neurons also received input representing top-down modulatory processes such
116 as arousal or behavior which may vary from trial to trial, but which were not directly related to the task
117 [Vinck et al., 2015]. In this linear model, each neuron’s activity is simply the summation of its sensory and
118 top-down input currents (Figure 1i).

119 From these minimal assumptions, the two input sources represent two different classes of covariability.
120 First, the sensory input has a characteristic temporal dynamics that is time-locked to stimulus presentation
121 with neural encoding weights that vary over trials due to heterogeneity in the learning dynamics. In contrast,
122 the top-down input source is characterized by fixed, non-plastic neural encoding weights but with trial-to-
123 trial variability in the temporal dynamics. The resulting population activity has slice rank of two, as the sum
124 of one time-slicing component (sensory input) and one neuron-slicing component (top-down input). Indeed,
125 sliceTCA is able to capture the two ground truth components (Figure 1j, Supplementary Figure 2c). On
126 the other hand, PCA and TCA require significantly more components to capture the variability (Figure 1j).
127 SliceTCA outperformed PCA and TCA even when white noise was added to the data (Supplementary Figure
128 2d,e). Together these results show that mixed covariability classes can emerge from minimal assumptions
129 about heterogeneity in neural circuits, and that they can be disentangled using sliceTCA.

130 **2.3 Task relevant information is distributed across slice types in motor cortical
131 reaching dynamics**

132 Based on the results of our toy model, we predicted that different slice types could capture different kinds of
133 behaviorally relevant dynamics in neural data. We tested this hypothesis in a dataset comprising primary
134 motor cortical (M1) and premotor cortical (PMd) populations recorded simultaneously during maze reaching
135 and classic center-out (no maze) reaching tasks (Figure 2a, hand position). To quantify decoding perfor-
136 mance, we linearly mapped population activity onto hand velocity (Methods). As a benchmark, we first
137 mapped trial-averaged raw neural data on kinematic trajectories, revealing a close match between behavior
138 and neural activity (Figure 2a, trial-averaged raw data). However, when we attempted to decode hand tra-
139 jectories based on individual trials, we observed significant trial-to-trial variability that corresponded poorly
140 to kinematic data (Figure 2a, raw data).

141 We reasoned that single-trial kinematic information might be present in the data, but obscured by
142 behaviorally irrelevant neural variability. If true, then the decoder should perform significantly better on
143 properly denoised data. To test this, we first used a common approach of fitting a low-rank approximation
144 using non-negative matrix factorization (NMF, $R = 12$ components) to the $N \times (TK)$ matrix of trial-
145 concatenated neural activity ('neuron-unfolded' data). Surprisingly, this actually decreased the performance
146 of the decoder (Figure 2a, neuron-slicing NMF), suggesting that the variability discarded by this denoising
147 procedure contains information about hand kinematics. We wondered whether better performance could be
148 obtained with a method that explicitly identifies shared variability across trials. Indeed, TCA-denoised data
149 displayed a better match to the hand kinematics ($R = 12$ components, Figure 2a, TCA). Yet, by constraining
150 the decomposition to be low tensor rank and thus also discarding temporal variability across neurons, TCA
151 is not able to reconstruct neural sequences at a sufficiently high temporal resolution to allow for precise
152 behavioral readout.

153 By performing TCA and NMF on the neuron-unfolded data tensor, we have assumed that behaviourally-
154 relevant variability in the data is shared across neurons (Figure 1g). However, previous work has emphasized

that dynamics in motor regions are better described by stereotyped sequences of activation that are shared across conditions [Seely et al., 2016, Mackevicius et al., 2019]. Following this intuition, we performed the same decoding analysis on denoised trial-unfolded data, where a $T \times (NK)$ matrix is approximated using NMF ($R = 12$ components). Remarkably, this simple change in denoising strategy resulted in a significantly better match between trial-to-trial variability in the data and in the hand kinematics (Figure 2a, trial-slicing NMF). We further validated that the components obtained by trial-slicing NMF corresponded to reach-tuned sequences whose temporal orderings were reproducible across held-out data (Supplementary Figure 3). These results reveal that in this dataset, behaviorally relevant information was encoded in neural sequences shared over trials, rather than by shared variability across neurons.

Trial- and neuron-concatenated NMF constitute two special cases of non-negative sliceTCA, where either neuron-slicing components or trial-slicing components exclusively are fitted. Therefore, we next asked whether we could identify additional information in the data by demixing different classes of covariability with sliceTCA. Previous work has identified preparatory signals in the premotor cortex that indicate the dynamics of the upcoming movement [Shenoy et al., 2013]. We therefore hypothesized that we could capture preparatory signals in a time-slicing component with shared ramping dynamics, and neural encoding weights encoding reach targets and curvature on a trial-by-trial basis.

Towards this end, we used sliceTCA to add a single time-slicing component to the previous model with 12 trial-slicing components (Figure 2b; Supplementary Figure 4). In both the trial-slicing NMF model and the sliceTCA model with mixed slice types, the trial-slicing components identified sequential neural dynamics for similar reach conditions which seemed to be continuously tuned to target angles (Figure 2c; Supplementary Figure 4b). Decoding from these trial-slicing components (in either the mixed or the unmixed model) led to significantly better performance as compared to the neuron-slicing and TCA models (Figure 2e; Supplementary Figure 5). The trial-slicing partial reconstruction from sliceTCA mapped slightly better onto hand kinematics in the mixed model than in the trial-slicing only model (Figure 2e, $p < 0.001$, Wilcoxon signed rank test). Intriguingly, while the single time-slicing component mapped poorly onto hand kinematics (Supplementary Figure 4a), it identified shared dynamics that peaked around 100 ms before movement onset (Figure 2d), consistent with a preparatory movement signal.

If the time-slicing component contains motor preparatory information, we would further expect it to contain information regarding the parameters of the upcoming movement [Shenoy et al., 2013]. Indeed, the neural encoding weights in PMd (but not in M1; Supplementary Figure 6) were correlated across similar conditions and encoded both reach direction and curvature (Figure 2f-h). Therefore, while the trial-slicing components directly encoded motor sequences governing hand kinematics, the time-slicing component contained primarily preparatory information about movement parameters. Together, these results show that behaviorally relevant information in neural data can be spread across different slice types, motivating the need to demix variability classes with sliceTCA.

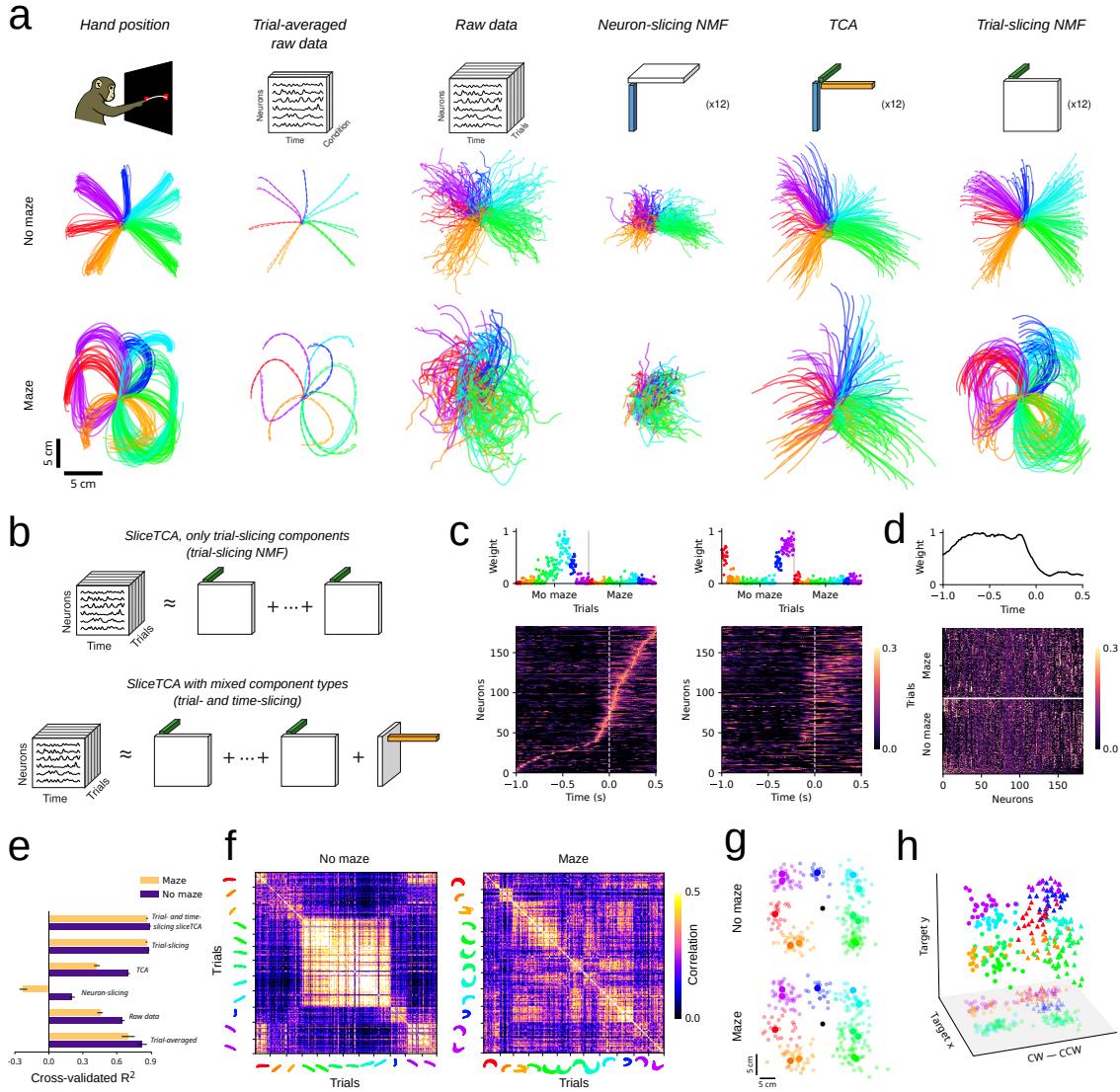


Figure 2: Time and trial slicing components identify preparatory and kinematic information in motor cortical dynamics, respectively. **a.** Behavioral and motor cortical dynamics ($n = 182$ neurons from M1/PMD) during a classic center-out reaching task with straight reaches (top) and curved maze reaches (bottom, modified from Churchland et al. [2012]). Different colours indicate different reach directions. *Hand position*. Hand positions during the experiment. *Trial-averaged raw data*. Condition-wise trial-averaged reaches (dashed lines) vs. condition-averaged neural population activity (solid lines), projected onto the 2D subspace that best matches hand trajectories. *Raw data*. Trial-by-trial mapping of raw population activity onto hand trajectories *Neuron-slicing*. Trial-by-trial mapping of denoised population activity onto hand trajectories (neuron-slicing NMF, 12 components; equivalent to NMF performed on the trial-concatenated data matrix). *TCA*. Trial-by-trial mapping of denoised population activity (TCA, 12 components). *Trial-slicing*. Trial-by-trial mapping of denoised population activity (trial-slicing NMF, 12 components) onto hand trajectories. **b.** Schematics of sliceTCA models with multiple components of the same slice type vs. a model with mixed slice types. **c.** Two example trial-slicing components, ordered by peak activation times of the first component. Sequential patterns distinguished specific reach conditions (here, upper left vs. lower right) **d.** The single time-slicing component has high temporal weight preceding movement onset, as well as condition-specific neural weights in the slice. **e.** R^2 of 5-fold cross-validated velocity decoding in each model. **f.** In PMd, correlations between neural weights on the time-slicing component were high for pairs of trials with similar reach direction and curvature, and low for dissimilar reaches. **g.** Mapping of average activity in the time-slicing component before movement onset (0.75 – 0s pre-onset) onto reach targets reveals a strong association ($R^2 = 0.95$ and $R^2 = 0.91$, center-out vs. curved reaches) **h.** Partially reconstructed activity from the time-slicing component, projected into a three-dimensional subspace identified to maximally separate clockwise vs. counter-clockwise movements and target x and y positions. Data points are clustered according to both reach direction and curvature, indicating that the time-slicing component encodes information about the dynamics of the upcoming movement (dots indicate clockwise, triangles counter-clockwise reaches).

190 **2.4 Pipeline for sliceTCA model selection and optimization**

191 Dimensionality reduction methods, while powerful, can prove challenging in practice. First, robustly identifying
192 the optimal number of components is a crucial yet challenging step in interpreting the dimensionality of
193 neural representations [Stringer et al., 2019, Lanore et al., 2021]. In many tensor and matrix decomposition
194 methods, such as NMF, different choices of the rank of the approximation may even lead to different results.
195 Moreover, even after the rank is fixed, invariances in the decomposition may lead to multiple possible solu-
196 tions. For example, matrix factorizations are known to be invariant to invertible linear transformations such
197 as rotations. Similarly, sliceTCA is invariant to such transformations within each slice type (Supplementary
198 Figure 7a, Methods). We have further identified a second class of invariant transformations that is specific
199 to sliceTCA (Supplementary Figure 7b, 8, Methods). This invariance class, when unaccounted for, prevents
200 an unambiguous attribution of covariance patterns to one of the three component types. Such invariances
201 can lead to difficulties in comparing latent representations across multiple datasets [Williams et al., 2021]
202 and are therefore crucial to address for any new method.

203 To address these concerns and to provide a user-friendly guideline for sliceTCA, we developed a full
204 data analysis pipeline for sliceTCA including data preprocessing, model selection, model optimization, and
205 visualization (Figure 3a). First, trials must be time-warped, trimmed, or masked, in order for the data to
206 be shaped into a tensor. Second, for choosing the optimal rank, we developed a rigorous cross-validation
207 procedure to identify the number of components of each slice type (Figure 3b), which we validated on
208 ground-truth data (Supplementary Figure 9). Third, to address the invariances of the decomposition, we
209 developed a hierarchical model optimization that adds additional constraints in the form of “sub-losses”
210 that must be minimized at three stages of optimization (Figure 3c; Supplementary Figure 10). One of the
211 stages of this procedure is a regularization of the reconstructed tensors of each slice type. Moreover, for non-
212 positivity-constrained sliceTCA, the same criteria used for matrix factorization methods (e.g., maximization
213 of variance and orthogonality as in PCA) can be applied to find unique solutions with respect to invertible
214 linear transformations. We further prove mathematically that a unique solution is guaranteed if each of the
215 sub-losses is unique (Supplementary Mathematical Notes). Together, employing a rigorous and standardized
216 pipeline for model selection, fitting, and optimization allows the user to make a robust, principled choice of
217 sliceTCA decomposition for further analyses and interpretation.

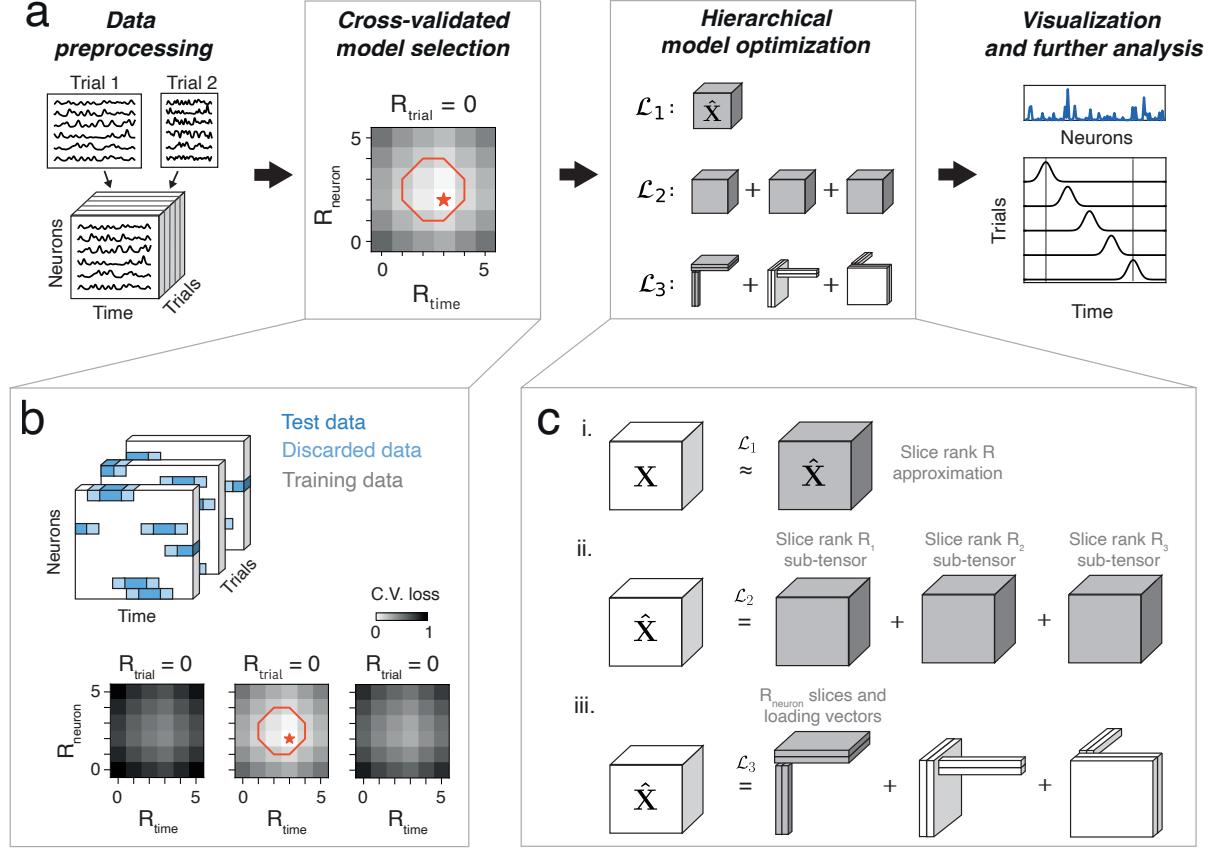


Figure 3: SliceTCA model selection, optimization, and analysis pipeline. **a.** SliceTCA data processing pipeline. First, neural data is preprocessed to form a data tensor. In experiments with variable trial length this could include temporal warping, exclusion of outlier trials, and/or trimming to the time period of interest. Second, model selection is performed to choose the number of components of each slice type ($R_{\text{neuron}}, R_{\text{trial}}, R_{\text{time}}$) based on the cross-validated mean square error (MSE) loss. Next, the hierarchical model optimization procedure is performed to identify a unique decomposition for the model. **b.** The cross-validation procedure for neural data tensors that we propose. We randomly assign blocks of consecutive time points (blue) within the data tensor as held out data. The remaining entries of the tensor are used as training data (white). To reduce correlations between the training and testing data, we discard a brief period from the ends of the held-out blocks (light blue) from both training and testing. We use only the interiors of these blocks as test data (dark blue). We run a 3D grid search on the cross-validated loss (bottom). We either choose the optimal model (red star) or a model at the “elbow” of the loss function (red circle). **c.** Hierarchical optimization over the sliceTCA invariance classes. We first fit the model on all data (optimizing the MSE or \mathcal{L}_1 loss). Then, we consecutively optimize the secondary loss functions \mathcal{L}_2 and \mathcal{L}_3 as described in (a). After this procedure, the resulting loading vectors and slices can be analyzed.

218 2.5 Denoising task-relevant manifolds during a motor task

219 With a standardized data analysis pipeline established, we next asked how behaviorally relevant latent
 220 structure sliceTCA could uncover in a novel dataset, without any prior expectation on the component types.
 221 We applied sliceTCA to a dataset consisting of simultaneously imaged cerebellar granule cells and pyramidal
 222 neurons in the premotor cortex of mice performing a motor task (Figure 4a) [Wagner et al., 2019]. Using the
 223 sliceTCA analysis pipeline, we selected a model with three trial-slicing components and three neuron-slicing
 224 components at the elbow of the cross-validated loss function (Figure 4b,c, Supplementary Figure 11; similar
 225 components observed in the optimal model, Supplementary Figure 12). The first trial-slicing component

226 captured temporally distributed cerebellar and cortical dynamics that were common to both left and right
227 correct reaches, but distinct from error reaches (Figure 4b,d). In contrast, the second trial-slicing component
228 accounted for the differential activation in left vs. right trials (Figure 4b,d). A third component decayed
229 slowly over trials, possibly representing adaptation over the course of the session (Figure 4b).

230 In addition, the three neuron-slicing components captured trial-specific population dynamics mostly
231 localized around the time of movement or reward (dashed lines, Figure 4c), with prolonged activity in error
232 trials, compared to correct trials, in the first and third component (Mann Whitney U-test, $p < 0.001$ for both
233 components). Interestingly, the second neuron-slicing component captured differences between cerebellar and
234 cortical activity (Figure 4c,d). The effect of simultaneously fitting two different covariability classes can be
235 observed by comparing sliceTCA to matrix factorization methods that do not demix neural- and trial-
236 covariability (Figure 4e, Supplementary Figure 13). While several loading vectors and slice weights of the
237 components found by PCA and FA appear similar to their corresponding sliceTCA components, sliceTCA
238 revealed more detailed structure for other components. But by contrast, without disentangling different
239 covariability classes, the slices identified by PCA and FA were of lower rank than the sliceTCA slices (Figure
240 4e, Supplementary Figure 13), and thus capture less trial- or neuron-specific dynamics and less structure in
241 the data. Together, these results show that sliceTCA identifies both task-specific (left, right, error trials) and
242 region-specific (cerebellum vs. cortex) variables, by capturing the structure of neural data across multiple
243 covariability classes.

244 Classic neural dimensionality reduction methods capture structure that is shared across neurons while
245 removing variability that is specific to individual neurons. We next illustrate how additionally modeling
246 structure that is neuron-specific but shared across trials affects the reconstruction of the data tensor (Figure
247 4f). Towards this end, we compared the neural representations of the raw data in neural space to the
248 reconstructed data from the sliceTCA model. The sliceTCA reconstruction captured the same top principal
249 components as the raw data, confirming that it was faithfully capturing the overall structure of the neural
250 representation (Supplementary Figure 14). The advantage of including both neural and trial-covariability was
251 reflected in increased behavioral interpretability of the neural representations. To show this, we projected the
252 data onto the dimension that best separated left vs. right correct trials during the period between movement
253 and reward. The axis found from the sliceTCA reconstruction revealed a more interpretable, denoised
254 representation as compared to the dimension found from raw data (Figure 4g). Similarly, the task-relevant
255 neural manifolds, found by projecting neural trajectories onto a subspace that separates activity along
256 three task-relevant dimensions (see Methods), appear significantly denoised when sliceTCA was applied,
257 compared to a direct projection of the raw data (Figure 4h; Supplementary Figure 14). We quantified this
258 denoising effect by measuring the distance between left and right trials around the time of movement onset in
259 sliceTCA reconstructions as compared to distances in raw data (Figure 4i). Our results show that sliceTCA,
260 by grouping behaviorally similar trajectories in an unsupervised manner, increases the distance between
261 trajectories of behaviorally distinct trials. Together, these results show that by demixing different classes
262 of covariability, sliceTCA is able to denoise task-relevant representations in neural data in an unsupervised
263 fashion.

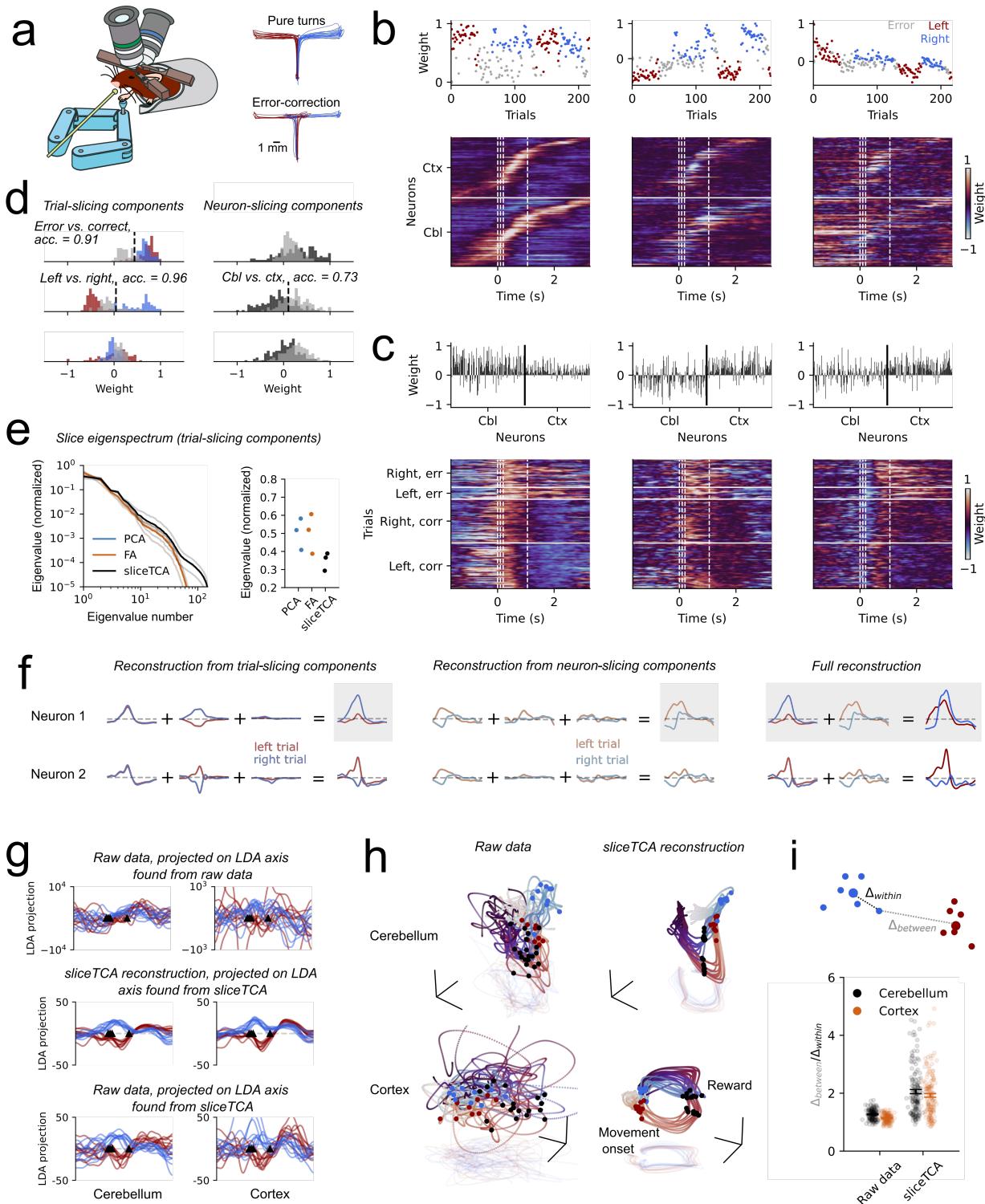


Figure 4: SliceTCA denoises task representations in simultaneously imaged cortical and cerebellar populations. **a.** Schematic of a mouse moving a manipulandum during simultaneous imaging of premotor cortex and cerebellum. Image modified from Wagner et al. [2019].

Figure 4 (previous page): **b.** The three trial-slicing components identified by sliceTCA. Weights in the loading vectors are colored according to trial type: correct left (red), correct right (blue), and error (grey). In the slices, neurons are sorted within each region (cbl, cerebellum and ctx, motor cortex) by the latency of maximum activation in the first component. White dashed lines indicate movement onset, mid-turn, movement end, and reward. **c.** Three neuron-slicing components. Loading vectors are separated into cerebellar and cortical populations. In the corresponding slices, trials are separated into left or right cued trials and into correct or error turns (corr/err). Within each block, trials are plotted in increasing order (ascending). **d.** Histograms of loading weights for the three trial- (left) and neuron-slicing (right) components, colored by trial type and region. We classified weight vectors (correct vs. incorrect and left vs. right correct trials; cerebellum vs. cortex). **e.** To show that sliceTCA results in more demixed representations with higher-rank slices than concatenated matrix factorization methods, we calculated the eigenvalues of the slices of the three trial-slicing components identified by PCA, factor analysis (FA), or sliceTCA. Left: Slice eigenspectrum, averaged over the three trial-slicing components (black; spectra for individual components in grey). Right, leading eigenvalue for each component. **f.** Example reconstructions of low slice rank approximations of individual neurons. Left: Reconstruction from the trial-slicing components. The latent dynamics for each component are neuron-specific, but shared across trials up to a scaling factor. Middle: Reconstruction from the neuron-slicing components. The latent dynamics are trial-specific but shared across neurons except for a scaling factor. Right: The full sliceTCA reconstruction is obtained by summing the contributions of all components from both slice types. Red/blue indicate dynamics on an example left/right trial. **g.** Data from ten example trials per condition, projected onto an axis that maximally separates left and right correct trials between movement onset and reward. Upper, raw data projected onto an LDA dimension found from the raw data; middle, sliceTCA reconstruction projected onto a dimension found in the sliceTCA reconstruction; lower, raw data projected onto the LDA dimension found in the sliceTCA reconstruction. **h.** Neural manifolds comprising example trajectories per trial type in an orthonormalized neural subspace found with LDA (axis 1, same as g; axis 2 that separates activity at the time of movement onset vs. reward; axis 3 that separates pre-movement vs. mid-movement) from raw data and sliceTCA reconstruction. **i.** Separation of the left vs. right trajectories from full data and data denoised with a mixed-component sliceTCA model. Δ_{within} (and $\Delta_{between}$) indicates the distance of the population vector in each trial around the time of movement onset to the center of the cluster of data points in its same (or, respectively, the opposite) trial class. Left and right trajectories are more separable after sliceTCA denoising (Wilcoxon signed-rank test, $p < .001$ both for cerebellum and motor cortex).

264 **2.6 Identifying components with region-specific covariability patterns in multi-**
 265 **region recordings**

266 So far we have shown that mixed variability co-occurs within the same neural population. However, the
 267 need to consider multiple covariability classes becomes even more crucial in simultaneous recordings from
 268 many regions, as previous work has shown that different brain regions may be better described by different
 269 unfoldings of the data tensor [Seely et al., 2016]. Yet, relying on different tensor unfoldings for the analysis of
 270 distinct regions would require that these regions be analyzed separately, without leveraging the simultaneous
 271 nature of such data. We therefore asked whether sliceTCA could demix area-specific representations in
 272 distinct slice types.

273 To test this idea, we took advantage of a recently published dataset consisting of Neuropixel recordings
 274 across six brain regions during a perceptual decision-making task (Figure 5a) [IBL et al., 2022]. We selected
 275 a model with eight components: two trial-slicing components, three neuron-slicing components, and three
 276 time-slicing components (Supplementary Figure 15, 16). The two trial-slicing components identified variables
 277 related to behavioral performance (Figure 5b). The first trial-slicing component separated correct from
 278 incorrect trials (Mann Whitney-U test, $p < .001$), and the corresponding slice was characterized by reward-
 279 locked temporal response profiles in midbrain nuclei (APN and MRN), which we validated in single neuron
 280 PSTHs (Figure 5c). The second trial-slicing component instead featured more temporally heterogeneous
 281 responses in all regions and correlated inversely with log reaction times (Pearson's $r = -0.35$, $p < 0.001$,
 282 $N = 831$ trials; Figure 5b). We next asked how these components contributed to the activity of different
 283 regions. The full sliceTCA reconstruction explained 33% - 49% of neural activity, depending on the region
 284 (Figure 5d). Of this reconstructed activity, the two trial-slicing components contributed considerably to

285 neurons in APN, MRN, and thalamus (TH) ($19 \pm 10\%$, mean \pm s.d., $N = 75$ neurons; Figure 5e). Thus,
286 the trial-slicing components identified stereotyped dynamics in subcortical regions TH, APN, and MRN that
287 were linked to behavioral performance across trials.

288 In contrast, the three neuron-slicing components identified three distinct clusters of neurons corresponding
289 to cortical regions: the hippocampus (CA), dentate gyrus (DG), and visual cortex (VIS) (Figure 5f). These
290 components therefore represented population-wide covariability patterns that were specific to each of these
291 regions. The slice of the CA-preferring component was characterized by a contrast-dependent activation
292 between the sensory cue and reward (correlation of stimulus-evoked responses with contrast, Pearson's $r =$
293 0.40 , $p < 0.001$; Figure 5f,g), a feature which was less prominent in the DG and not observed in VIS-
294 preferring components ($r = 0.11$, $p = 0.002$ for DG, $R = -0.05$, $p = 0.14$ for VIS). In the DG-preferring
295 component, we observed post-reward suppression on correct (rewarded) trials which was significantly shorter
296 on error trials (Mann Whitney U-test, $p < 0.001$; Figure 5f). The final VIS-preferring component revealed
297 pre-stimulus activation that increased in strength over trials (Pearson's $r = 0.55$, $p < 0.001$, Figure 5f),
298 possibly indicating the emergence of a predictive signal of cue onset over the course of the experiment. Each
299 component contributed to a large fraction of the sliceTCA reconstruction in its respective region ($37 \pm 21\%$,
300 $N = 138$ neurons; Figure 5h). Therefore, the three neuron-slicing components represented different task-
301 relevant features that were separately encoded in CA, DG, and VIS population responses.

302 Finally, the remaining time-slicing components partitioned the task duration into three distinct periods:
303 early (pre-stimulus and stimulus onset), late (post-reward), and reward period (Figure 5i). The corresponding
304 slices revealed smooth variations of the strength of each of these components in single neurons over the course
305 of the experiment. Given the strong similarity of the three slices, we asked whether the components could sum
306 to a flat trial-varying baseline for each neuron. However, when we examined single neurons we instead saw
307 examples of a broad range of modulation patterns, with slowly varying activity that changed heterogeneously
308 over trials for the three task periods (Figure 5j as an example VIS neuron with increasing activity during
309 pre-stimulus and post-reward periods, but not during the reward period). We tested this hypothesis using a
310 linear model to compare the rate of change of the trial weights for each neuron across components (Methods).
311 A substantial proportion of neurons across all regions showed significantly different rates of change across
312 components (ANOVA, $p < 0.05$ with Bonferroni correction, $N = 221$ neurons; Supplementary Figure 17).
313 Moreover, these three components contributed significantly to the sliceTCA reconstruction across all recorded
314 regions ($62 \pm 18\%$, $N = 213$ neurons; Figure 5h). Together, these results show that by accounting for different
315 classes of covariability, sliceTCA is able to demix multi-region recording data into brain-wide representations
316 of task period, and behaviorally-relevant stereotyped dynamics, and population-wide patterns of covariability
317 encoded by individual regions.

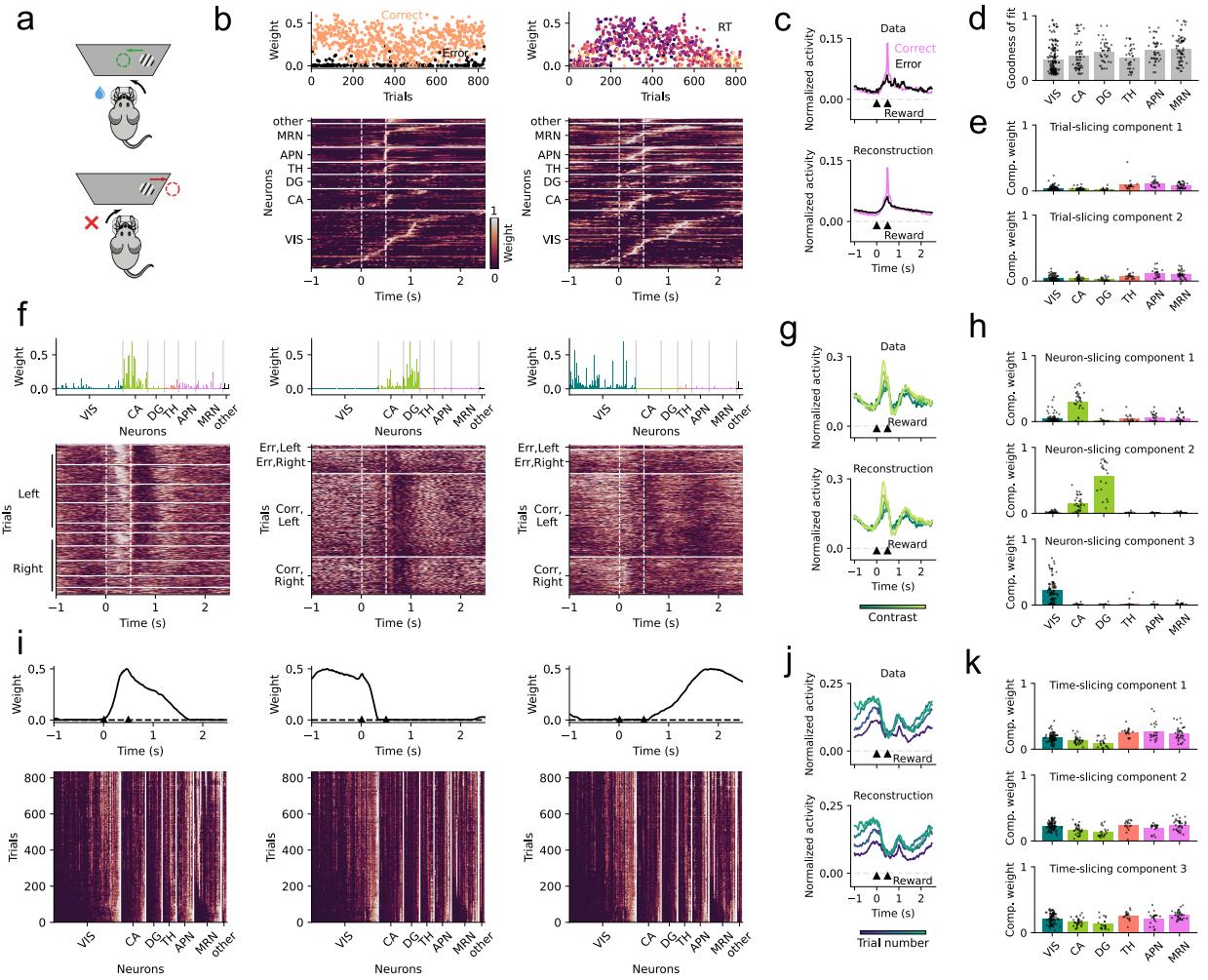


Figure 5: SliceTCA identifies region-specific sensory and behavioral variables in multi-region recordings. Schematic of perceptual decision making task from the International Brain Laboratory (IBL). Figure modified from IBL et al. [2021]. **b.** Two trial-slicing components: The loading vector of component 1 shows a separation between correct (orange) and error (black) trials. In component 2, the color scale in loading vector indicates log reaction time for each trial. In the corresponding slices: visual cortex (VIS), hippocampus (CA), dentate gyrus (DG), thalamus (TH), anterior pretectal nucleus (APN), and midbrain reticular nucleus (MRN). White dashed lines indicate stimulus onset and reward or timeout onset, respectively. Slice weights are normalized to [0,1] for each neuron separately and sorted by the latency of peak activation within each region (separately for each component). **c.** Top: PSTH of an example APN neuron with dominant reward-locked dynamics for correct (pink) and error trials (black). Bottom: PSTH built from the full sliceTCA reconstruction. Arrows indicate stimulus onset and reward. **d.** Reconstruction performance (Methods) of the full sliceTCA model, separated by region. Black dots indicate individual neurons. **e.** Contribution of each trial-slicing component to the overall reconstruction. **f.** Three neuron-slicing components: In each slice, trials are grouped into blocks separately for different components. In component 1 with dominant contribution of CA1, trials are grouped by contrast separately for left/right trials (within left/right, contrast increases from bottom to top). In components 2 (DG-related) and 3 (VIS-related), trials are grouped into blocks by left/right and correct/error. For all slices, within each block, trials are sorted in increasing order (ascending). Each slice is normalized to [0,1]. **g.** Top: PSTH of an example CA neuron for low to high contrasts (dark to light green). Bottom: PSTH built from the full sliceTCA reconstruction. **h.** Contribution of each neuron-slicing component to the overall reconstruction. **i.** Three time-slicing components: In the slices, neurons are sorted within each region according to increasing activation in early trials after normalizing weights for each neuron to [0,1] (same sorting across components). **j.** Top: PSTH of an example VIS neuron for early to late trials (blue to teal). Bottom: PSTH built from the full sliceTCA reconstruction. **k.** Contribution of each time-slicing component to the overall reconstruction.

318 2.7 Geometric interpretation of sliceTCA components

319 Recently, dimensionality reduction has been used in systems neuroscience to interpret neural population
 320 activity as trajectories embedded in a low-dimensional latent subspace within the full neural activity space.
 321 In sliceTCA, the neuron-slicing components can be interpreted in the same way due to their similarity to
 322 matrix factorization on trial-concatenated data. However, the time and trial slicing components have different
 323 interpretations as their natural bases lie within spaces in which each axis represents a different timepoint or
 324 a different trial. How then can we grasp the time- and trial-slicing components' contributions to latent
 325 representations in neural activity space?

326 We can answer this question by considering the hypothetical contribution from each slice type separately.
 327 First, note that while the neuron-slicing components are constrained to an R_{neuron} -dimensional subspace,
 328 their dynamics within that subspace are unconstrained over trials (Figure 6, neuron-slicing component). On
 329 the other hand, the dynamics of the R_{time} time-slicing components are constrained to a common temporal
 330 dynamic, but the neural weight vectors can instead vary from trial to trial. Geometrically, this means that
 331 the reconstruction from these components lies within an R_{time} -dimensional subspace that can now vary
 332 on each trial, but that the temporal dynamics within each trial-specific subspace is constrained to be the
 333 same (Figure 6, time-slicing component). Finally, the R_{trial} trial-slicing components' neural weights change
 334 at every timepoint, while trial weights are fixed. This corresponds to latent dynamics that are no longer
 335 embedded in a low-dimensional subspace, but that are built instead from stereotyped dynamical trajectories
 336 (Figure 6, trial-slicing component; but see Supplementary Figure 18). In this way, the three covariability
 337 classes that we have described can also be seen as three classes of latent dynamics in neural activity space.
 338 Together, all three classes contribute to the dynamics of the full reconstruction, which may appear more
 339 complex than any one component type (Figure 6a, reconstruction).

340 This geometric view illustrates that by fitting different classes of covariability, sliceTCA is able to capture
 341 latent dynamics that are no longer confined to a linear subspace, despite still being a multilinear method.
 342 In contrast, traditional matrix factorization methods which capture only a single covariability class are re-
 343 stricted to one of the three geometric classes of latent dynamics in neural space shown in Figure 6, while
 344 TCA constrains its components to obey the geometrical constraints of all three classes simultaneously (Sup-
 345 plementary Figure 19). In sum, sliceTCA is able to capture a broader range of covariability structure in
 346 neural data, and a broader range of latent representations in neural space, than related methods, all while
 347 remaining easily interpretable.

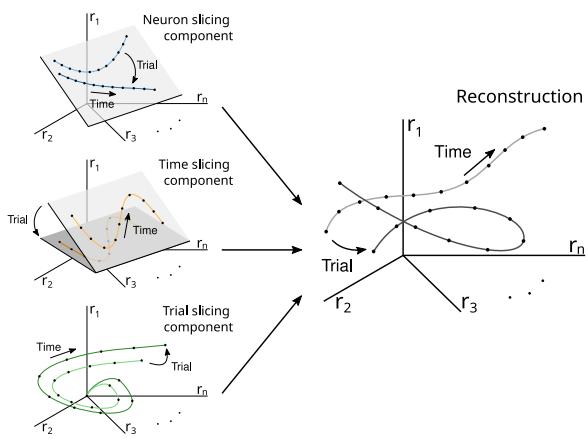


Figure 6: Different slice types capture latent variables with distinct geometric properties. *Neuron-slicing component.* Example of two neuron-slicing components visualized in neural activity space. The latent trajectories are embedded in an two-dimensional subspace, but their dynamics within that subspace are unconstrained. *Time-slicing component.* Example of two time-slicing components. These are similarly embedded within an two-dimensional subspace, but that subspace varies over trials. The latent variables are further constrained to follow the same dynamics within each latent subspace. *Trial-slicing component.* The trial-slicing components are not constrained to any latent subspace, as the neural encodings may change at every timepoint. These components describe potentially high-dimensional dynamics that are stereotyped across trials. Note that here only 1 component is shown for clarity. *Reconstruction.* After summing these components, the full latent trajectories are not necessarily limited by any of the geometric constraints that characterize individual slice types.

348 **3 Discussion**

349 Neural population dynamics are frequently interpreted as low-dimensional latent variables encoded by fixed
350 subgroups of neurons, which represent shared variability across neurons. Here, we have advocated for
351 an expansion of this view of structure in neural data which takes into account three distinct classes of
352 shared variability: across neurons, time, and trials. Towards this end, we introduced sliceTCA, a new
353 tensor decomposition method that is able to demix latent variables that belong to any of these covariability
354 classes. Through several example datasets, we demonstrated that sliceTCA can capture more task-relevant
355 covariability in neural data in fewer components, enabling the description and interpretation of complex
356 latent structure embedded in large-scale neural recordings. Finally, we illustrated how sliceTCA expands
357 the classic view of neural population dynamics towards latent variables that are not constrained to low-
358 dimensional dynamics in a fixed, linear manifold.

359 Our framework of multiple covariability classes addresses key limitations of the classic view on latent
360 dynamics, which is unable to identify several types of structure commonly found in neural data. In particular,
361 this view fails to capture neural sequences, as previously pointed out in the literature [Mackevicius et al.,
362 2019, Seely et al., 2016]. Indeed, task-relevant neural sequences are a widespread phenomenon observed across
363 brain regions during navigation, timing, value-based decision making, and motor production [Harvey et al.,
364 2012, Parker et al., 2022, Zhou et al., 2020]. Here we have emphasized the ability of the trial covariability
365 class to capture neural sequences that have shared structure across trials, e.g. choice-specific sequences.
366 However, we note that this class can capture more complex forms of neuron-specific temporal patterning
367 within a trial [Feng et al., 2015, Lakshmanan et al., 2015, Koay et al., 2022]. On the other hand, population
368 modes characterized by trial-to-trial differences in timing are captured in the neural covariability class. Such
369 variations in timing may be critical for interpretation, for example the shift of the reward prediction error
370 during temporal difference learning [Amo et al., 2022, Schultz, 1998]. Lastly, the time covariability class
371 may be well-suited for describing forms of learning or representational drift in which the latent space over
372 which neural data evolves over trials [Hennig et al., 2021, Rule et al., 2019]. Importantly, it has been argued
373 that different brain regions are better described by neural or by trial covariability [Seely et al., 2016]. Our
374 results support this hypothesis, and further show that these different classes can be demixed by sliceTCA.
375 Therefore, demixing covariability classes may be a crucial step when considering large-scale multi-region
376 recordings [IBL et al., 2022, Wagner et al., 2019, Ebrahimi et al., 2022, Ahrens et al., 2012].

377 A longstanding challenge in systems neuroscience is the difficulty of mapping neural variability to changes
378 in behavior [Renart and Machens, 2014]. This can be accomplished using supervised dimensionality reduction
379 methods that use information regarding behavior or task outcomes to identify latent variables [Kobak et al.,
380 2016, Sani et al., 2021a, Balzani et al., 2022]. Despite it being an unsupervised method, we found that
381 SliceTCA was able to disentangle behavioral and task information in each of the datasets presented. We
382 claim that this is due to two reasons: first, demixing different sources of covariability effectively “denoises”
383 components that represent task variables that would have otherwise been occluded by additional sources of
384 variability. Second, the trial-slicing components explicitly identify dynamics that are shared across trials,
385 which tend to be defined by task variables or behavioral outcomes. Indeed, in each of the three datasets, we
386 found that trial-slicing and time-slicing components correlated with behavioral variables. Moreover, in our
387 feedforward model, we suggest how sliceTCA could offer a window into the computational roles of variables
388 modeled by different slice types. We argue that the classical view on neural latents, which assumes that a
389 key part of behaviorally relevant neural variability is correlated across neurons, is overly reductionist and
390 may miss many types of neural dynamics underlying behavior.

391 Beyond tensor and matrix based methods, more sophisticated forms of nonlinear dimensionality reduction
392 can be used to identify latent variables embedded within a curved manifold [Balasubramanian and Schwartz,
393 2002, Belkin and Niyogi, 2003, McInnes et al., 2018]. Within neuroscience, several methods have been
394 proposed specifically for neural data, including methods based on neural networks [Pandarinath et al., 2018,
395 Schimel et al., 2022, Sani et al., 2021b] or manifold reconstruction based on topological features [Chaudhuri
396 et al., 2019, Rybakken et al., 2019]. While these methods are crucial for identifying nonlinearly embedded
397 latent variables, a key advantage of matrix and tensor decompositions is the simplicity of the models. Indeed,
398 the analytical tractability of the sliceTCA decomposition enabled us to characterize its invariance classes

399 and to propose a method to identify a unique solution. Identifying invariances is crucial for reproducibility
400 and interpretation, as non-unique solutions may prohibit clear comparison across datasets [Dyer et al., 2017,
401 Gallego et al., 2020]. This issue is ever more important with the recent increase in popularity of comparisons
402 of neural data to task-trained neural network models, whose representations are known to be sensitive to
403 model specifications such as architecture and inputs [Lindsay et al., 2022, Williams et al., 2021]. Going
404 forward, matrix and tensor decompositions could prove useful for comparing latent representations by virtue
405 of their interpretability and tractability.

406 SliceTCA falls into a larger class of tensor decomposition methods including TCA [Williams et al., 2018,
407 Harshman et al., 1970], which captures variability lying at the intersection of the covariability classes, and the
408 Tucker decomposition, which allows factors to interact via a core tensor [Onken et al., 2016]. Yet while tensor
409 decompositions can be viewed as generalizations of matrix factorizations, they do not always have the same
410 properties. For example, tensor-based methods are known to be generally more computationally expensive
411 [Kolda and Bader, 2009, Bläser et al., 2019]. Still, tensor decompositions are key methods in neuroscience as
412 they allow the discovery of components that can be mapped across trials or conditions. Here we have focused
413 on the classic third-order tensors (neurons \times time \times trials) that are frequently used in neuroscience. Current
414 experimental techniques are rapidly enabling the acquisition of data tensors of even higher order, by adding
415 legs that correspond to days or conditions. Future extensions of tensor methods that allow individuals to
416 be incorporated as an additional leg could help to identify the neural basis of variability across subjects
417 [Kuchibhotla et al., 2019, Smith et al., 2022]. Going forward, our framework of mixed classes of covariability
418 can help to advance our understanding of behaviorally relevant latent structure in high-dimensional neural
419 data recorded during increasingly complex tasks, across brain regions and across individual subjects.

420 4 Methods

421 4.1 Definition of sliceTCA model

422 4.1.1 Matrix rank and matrix factorization

423 Consider a data matrix consisting of N neurons recorded over T samples (timepoints): $\mathbf{X} \in \mathbb{R}^{N \times T}$. Matrix
424 factorization methods find a low-rank approximation $\hat{\mathbf{X}}$ following Eq. 1, in which each component is a rank-1
425 matrix: $\mathbf{X}^{(r)} = \mathbf{u}^{(r)} \otimes \mathbf{v}^{(r)}$, where $\mathbf{u}^{(r)} \in \mathbb{R}^N$ and $\mathbf{v}^{(r)} \in \mathbb{R}^T$ are vectors representing the neural and temporal
426 coefficients, which are chosen to minimize a loss function. In other words, the activity of neuron n at time t
427 is given by:

$$428 \quad \hat{X}_{n,t} = \sum_{r=1}^R u_n^{(r)} v_t^{(r)} \quad (2)$$

429 A common choice of loss function is the mean squared error:

$$430 \quad \mathcal{L} = \frac{1}{NT} \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 \quad (3)$$

431 Constraints may be added to the minimization of the loss, such as non-negativity of the coefficients in NMF.

432 4.1.2 Slice rank and sliceTCA

433 A d -tensor is a generalization of data matrices to d legs (i.e, a data matrix is a 2-tensor). Here we are
434 specifically concerned with 3-tensors typically used in neuroscience, in which the three legs represent neurons,
435 time, and trial/condition: $\mathbf{X} \in \mathbb{R}^{N \times T \times K}$. SliceTCA extends the matrix factorization in Eq. (1) by fitting \mathbf{X}
436 with a low *slice rank* approximation [Tao and Sawin, 2016]. A slice-rank-1 d -tensor is an outer product of a
437 vector and a $(d-1)$ -tensor. For the 3-tensors that we have been considering, this corresponds to the outer
438 product of a ‘loading’ vector and a 2-tensor, thus making this 2-tensor a *slice* of this slice-rank-1 tensor up
439 to a scalar multiple determined by the loading vector.

440 Each sliceTCA component can be one of three different slice types. For example, a neuron-slicing com-
 441 ponent can be written as $\mathbf{X}^{(r)} = \mathbf{u}^{(r)} \otimes \mathbf{A}^{(r)}$ where $\mathbf{A}^{(r)} \in \mathbb{R}^{T \times K}$ is the time-by-trial slice representing the
 442 dynamics of the component across both time and trials and the vector $\mathbf{u}^{(r)}$ represents the neural loading
 443 vector. Components of other slice types can be constructed similarly with their respective loading vectors
 444 and slices: $\mathbf{v}^{(r)} \in \mathbb{R}^T$, $\mathbf{B}^{(r)} \in \mathbb{R}^{N \times K}$ for the time-slicing components, and $\mathbf{v}^{(r)} \in \mathbb{R}^K$, $\mathbf{C}^{(r)} \in \mathbb{R}^{N \times T}$ for the
 445 trial-slicing components. Put together, this results in a decomposition of the following form:

$$446 \quad \hat{\mathbf{X}}_{n,t,k} = \sum_{r=1}^{R_{\text{neuron}}} u_n^{(r)} A_{t,k}^{(r)} + \sum_{r=1}^{R_{\text{time}}} v_t^{(r)} B_{n,k}^{(r)} + \sum_{r=1}^{R_{\text{trial}}} w_k^{(r)} C_{n,t}^{(r)} \quad (4)$$

447 Because of the different slice types, each sliceTCA model can be described by the hyperparameter 3-tuple
 448 $\mathbf{R} = (R_{\text{neuron}}, R_{\text{trial}}, R_{\text{time}})$, defining the number of neuron-, trial-, and time-slicing components, for a total
 449 of $R_{\text{neuron}} + R_{\text{trial}} + R_{\text{time}}$ components.

450 4.1.3 Relationship to TCA

451 The extension of matrix factorizations to TCA is based on a different definition of tensor rank, in which a
 452 rank-1 tensor is as an outer product of d vectors. Each component is defined by a set of vectors corresponding
 453 to trial coefficients $\mathbf{w}^{(r)} \in \mathbb{R}^K$ to each component: $\mathbf{X}^{(r)} = \mathbf{u}^{(r)} \otimes \mathbf{v}^{(r)} \otimes \mathbf{w}^{(r)}$. Then each element of the
 454 approximated data tensor can be written as:

$$455 \quad \hat{\mathbf{X}}_{n,t,k} = \sum_{r=1}^R u_n^{(r)} v_t^{(r)} w_k^{(r)} \quad (5)$$

456 In other words, a TCA component is a special case of a sliceTCA component in which the slice is a rank-1
 457 matrix. In this way, sliceTCA is more flexible than TCA as it has fewer constraints on the type of structure
 458 that is identified in the data. However, this increase in flexibility comes with a cost of an increase in the
 459 number of parameters, as sliceTCA fits all the entries of each slice. The flexibility of sliceTCA also leads
 460 to different invariance classes as discussed below. Finally, we note that the two methods can in principle be
 461 merged by incorporating TCA components into Eq. 4.

462 4.2 SliceTCA invariance classes

463 4.2.1 Transformations within a slice type

464 Matrix factorization methods are known to be invariant to invertible linear transformations, including, but
 465 not limited to, rotations of the loading vectors. For example, suppose we decompose a matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$
 466 into a the product of a matrix of weights, $\mathbf{W} \in \mathbb{R}^{N \times R}$ and a matrix of scores, $\mathbf{S} \in \mathbb{R}^{R \times T}$. Consider any
 467 invertible linear transformation $\mathbf{F} \in \mathbb{R}^{R \times R}$. Then \mathbf{Y} can be re-written as:

$$468 \quad \mathbf{Y} = \mathbf{WS} = \mathbf{WF}\mathbf{F}^{-1}\mathbf{S} = \tilde{\mathbf{W}}\tilde{\mathbf{S}} \quad (6)$$

469 where $\tilde{\mathbf{W}} = \mathbf{WF}$ and $\tilde{\mathbf{S}} = \mathbf{F}^{-1}\mathbf{S}$. As a result, matrix decompositions like factor analysis (FA) lead to not
 470 one solution, but rather an invariance class of equivalent solutions. Note that PCA avoids this problem by
 471 aligning the first component to the direction of maximum projected variance, as long as the eigenvalues of
 472 the covariance matrix are distinct. However, other methods which do not have a ranking of components
 473 are not able to use the same alignment. SliceTCA inherits this same invariance class, since all the loading
 474 vectors within a given slice type can be transformed in the same way as Eq. (6) to yield the same partially
 475 reconstructed tensor for each slice type (Supplementary Figure 7a).

476 4.2.2 Transformations between slice types

477 SliceTCA has an additional invariance class due to the fundamental properties of multilinear addition. For
 478 example, consider a slice-rank-2 tensor $\mathbf{Y} \in \mathbb{R}^{N \times T \times K}$ which is made of two components of different slice

479 types, which we will assume without loss of generality to be neuron- and time-slicing components with
480 corresponding slices \mathbf{V} and \mathbf{U} , such that:

$$481 \quad Y_{n,t,k} = u_n V_{t,k} + v_t U_{n,k}$$

482 Then the following transformation can be performed for arbitrary vector $\mathbf{z} \in \mathbb{R}^K$,

$$\begin{aligned} 483 \quad Y_{n,t,k} &= u_n V_{t,k} + v_t U_{n,k} + u_n v_t z_k - u_n v_t z_k \\ 484 \quad &= u_n (V_{t,k} - v_t z_k) + v_t (U_{n,k} + u_n z_k) \\ 485 \quad &= u_n \tilde{V}_{t,k} + v_t \tilde{U}_{n,k} \end{aligned}$$

486 where $\tilde{\mathbf{V}} = \mathbf{V} - \mathbf{v} \otimes \mathbf{z}$ and $\tilde{\mathbf{U}} = \mathbf{U} + \mathbf{u} \otimes \mathbf{z}$ are transformations of the original slices. This invariance class there-
487 fore corresponds to passing a tensor-rank-1 tensor between two slices of differing slice types (Supplementary
488 Figure 7b).

489 Note that two classes of transformations (within-slice-type and between-slice-type) commute (see propo-
490 sition 1.2 of Mathematical Notes), and therefore one cannot get a new transformation by, for example,
491 applying the first transformation, the second, and then the first again.

492 4.2.3 Identification of unique sliceTCA decomposition

493 In order to find a uniquely defined solution we can take advantage of natural hierarchy between the two
494 invariance classes. Specifically, let us first define the partial reconstruction $\hat{\mathbf{X}}^{\text{neuron}}$ of the low-slice-rank
495 approximation $\hat{\mathbf{X}}$ based on the neuron-slicing components, i.e.:

$$496 \quad \hat{\mathbf{X}}^{\text{neuron}} = \sum_{r=1}^{R_{\text{neuron}}} \mathbf{u}^{(r)} \otimes \mathbf{A}^{(r)}$$

497 and let $\hat{\mathbf{X}}^{\text{time}}$ and $\hat{\mathbf{X}}^{\text{trial}}$ be similarly defined, so that $\hat{\mathbf{X}} = \hat{\mathbf{X}}^{\text{neuron}} + \hat{\mathbf{X}}^{\text{time}} + \hat{\mathbf{X}}^{\text{trial}}$. Now note that the
498 within-slice-type transformations change the weights of the loading vectors and slices of all components of
499 a given slice type, without changing the partial reconstructions for each slice type. For example, applying
500 these transformations to the neuron-slicing components would change $\mathbf{u}^{(r)}$ and $\mathbf{A}^{(r)}$ but not $\hat{\mathbf{X}}^{\text{neuron}}$. On
501 the other hand, the between-slice-type transformations change the partial reconstructions $\hat{\mathbf{X}}^{\text{neuron}}$, $\hat{\mathbf{X}}^{\text{time}}$
502 and $\hat{\mathbf{X}}^{\text{trial}}$, but not the full reconstruction $\hat{\mathbf{X}}$.

503 We leveraged this hierarchy to develop a post-hoc model optimization into three steps, each with a
504 distinct loss function. The first step identifies a model that minimizes a loss function \mathcal{L}_1 defined on the
505 full reconstruction (Figure 3c i), fixing the tensor approximation $\hat{\mathbf{X}}$. Next, we use stochastic gradient
506 descent to identify the between-slice-type transformation that minimizes a new loss function \mathcal{L}_2 , which
507 fixes $\hat{\mathbf{X}}^{\text{neuron}}$, $\hat{\mathbf{X}}^{\text{time}}$ and $\hat{\mathbf{X}}^{\text{trial}}$ without affecting $\hat{\mathbf{X}}$ (Figure 3c ii). Finally, we identify the within-slice-type
508 transformation that minimizes loss \mathcal{L}_3 to arrive at the final components (loading vectors $\mathbf{u}^{(r)}, \mathbf{v}^{(r)}, \mathbf{w}^{(r)}$ and
509 slices $\mathbf{A}^{(r)}, \mathbf{B}^{(r)}, \mathbf{C}^{(r)}$) without affecting $\hat{\mathbf{X}}^{\text{neuron}}$, $\hat{\mathbf{X}}^{\text{trial}}$, and $\hat{\mathbf{X}}^{\text{time}}$ (Figure 3c iii). Each of the three loss
510 functions can in principle be chosen according to the constraints or normative assumptions most relevant
511 to the question at hand. Furthermore, we prove that if each of these objective functions leads to a unique
512 solution, then the decomposition is unique under the condition that $\text{rank } \mathbf{A}^{(r)} > R_{\text{time}} + R_{\text{trial}}$ for all
513 $r = 1, \dots, R_{\text{neuron}}$ and similarly for the other two slice types (see Theorem 1.8, Supplementary Mathematical
514 Notes).

515 4.3 Model selection, optimization and fitting

516 To fit sliceTCA for a given dataset arranged as a 3-tensor, we followed the data analysis pipeline described
517 in the main text. Below, we provide details and hyperparameters for the steps involved in the pipeline.

518 **4.3.1 Fitting sliceTCA with stochastic gradient descent**

519 For a fixed choice of \mathbf{R} , model parameters (i.e., loading vectors and slices for each component) were fitted
 520 using the optimizer Adam [Kingma and Ba, 2014] in Pytorch [Paszke et al., 2019]. Initial parameters were
 521 randomly drawn from a uniform distribution over $[-1, 1]$ or $[0, 1]$, respectively, for unconstrained and non-
 522 negative sliceTCA. Throughout, we optimized the mean-squared error (MSE) loss in Eq. (3) with a learning
 523 rate of 0.02. To introduce stochasticity in the computation of the gradient, and thus avoid local minima, we
 524 masked a fraction of tensor entries so that they are not included in the calculation of the loss. This fraction
 525 starts at 80 % and decreases exponentially during training with a decay factor of 0.5 over three (Figure 2)
 526 or five blocks of iterations (Figures 4 and 5), respectively. Within each block, the mask indices are randomly
 527 reinitialized every 20 out of a total of 150 (Figure 2), 200 (Figure 4), or 100 iterations per block (Figure
 528 5). To obtain an optimal model under a given \mathbf{R} , we repeated the fitting procedure ten times with different
 529 random seeds and chose the model with the lowest loss.

530 **4.3.2 Cross-validated grid search**

531 To choose the number of components in each slice type, we run a three-dimensional grid search to optimize
 532 the cross-validated loss. In addition to the decaying mask used during model fitting, we mask 20 % of the
 533 entries throughout the fitting procedure as held-out data. These masked entries were chosen in randomly
 534 selected 1 s (Figure 4) or 150 ms blocks (Figure 5) of consecutive timepoints in random neurons and trials.
 535 Blocked masking of held-out data (rather than salt-and-pepper masking) was necessary to avoid temporal
 536 correlations between the training and testing data due to the slow timescale of the Ca^{2+} indicator or due
 537 to smoothing effects in electrophysiological data. To further protect against spuriously high cross-validation
 538 performance due to temporal correlations, we trimmed the first and last 250 ms (Figure 4) or 40 ms (Figure
 539 5) from each block; this data was discarded from the test set, and only the remaining interior of each
 540 block was used to calculate the cross-validated loss. We repeated the grid search ten times with different
 541 random seeds for train-test-split and parameter initialization, while keeping a constant seed for different \mathbf{R} .
 542 Once the cross-validated grid search is complete, we selected \mathbf{R}^* by identifying the model with minimum
 543 or alternatively, near-optimal average test loss across seeds. Admissible models are defined as achieving a
 544 minimum of 80 % of the optimal performance for non-constrained sliceTCA, and 95 % of the optimal model
 545 performance for non-negative sliceTCA, as compared to root mean squared entries of the raw data.

546 **4.3.3 Hierarchical model optimization**

547 For the first step of the model optimization procedure, we chose the mean squared error loss for \mathcal{L}_1 :

$$548 \quad \mathcal{L}_1(\mathbf{u}, \mathbf{A}, \mathbf{v}, \mathbf{B}, \mathbf{w}, \mathbf{C}) = \frac{1}{KNT} \left\| \mathbf{X} - \left(\sum_{r=1}^{R_{\text{neuron}}} [\mathbf{u}^{(r)} \otimes \mathbf{A}^{(r)}] + \sum_{r=1}^{R_{\text{time}}} [\mathbf{v}^{(r)} \otimes \mathbf{B}^{(r)}] + \sum_{r=1}^{R_{\text{trial}}} [\mathbf{w}^{(r)} \otimes \mathbf{C}^{(r)}] \right) \right\|_F^2$$

549 as in the model selection (essentially refitting the model with the specific ranks identified with the cross-
 550 validation procedure on the entire data). For \mathcal{L}_2 we use the sum of the squared entries of the three partial
 551 reconstructions from each slice type,

$$552 \quad \mathcal{L}_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \left\| \hat{\mathbf{X}}^{\text{trial}} - \sum_{r,s} \mathbf{x}^{(r,s)} \otimes \mathbf{v}^{(s)} \otimes \mathbf{w}^{(r)} - \sum_{r,s} \mathbf{u}^{(r)} \otimes \mathbf{y}^{(r,s)} \otimes \mathbf{w}^{(s)} \right\|_F^2 \\ 553 \quad + \left\| \hat{\mathbf{X}}^{\text{time}} + \sum_{r,s} \mathbf{x}^{(r,s)} \otimes \mathbf{v}^{(s)} \otimes \mathbf{w}^{(r)} - \sum_{r,s} \mathbf{u}^{(r)} \otimes \mathbf{v}^{(s)} \otimes \mathbf{z}^{(r,s)} \right\|_F^2 \\ 554 \quad + \left\| \hat{\mathbf{X}}^{\text{neuron}} + \sum_{r,s} \mathbf{u}^{(r)} \otimes \mathbf{y}^{(r,s)} \otimes \mathbf{w}^{(s)} + \sum_{r,s} \mathbf{u}^{(r)} \otimes \mathbf{v}^{(s)} \otimes \mathbf{z}^{(r,s)} \right\|_F^2$$

555 where $\mathbf{x} \in \mathbb{R}^{R_{\text{time}} \times R_{\text{trial}} \times N}$, $\mathbf{y} \in \mathbb{R}^{R_{\text{neuron}} \times R_{\text{trial}} \times T}$, and $\mathbf{z} \in \mathbb{R}^{R_{\text{neuron}} \times R_{\text{time}} \times K}$. This can be thought as a form
 556 of L2 regularization. For \mathcal{L}_3 we chose orthogonalization and variance explained ordering through singular
 557 value decomposition.

558 4.4 Feedforward model of perceptual learning

559 We modeled a population of linear neurons receiving sensory input from upstream sources representing a Go
 560 and a No-go stimulus, as well as input representing top-down modulation which varied from trial to trial.
 561 On each trial k , either the Go or No-go stimulus was activated, with probability $p = 0.5$ of presenting the
 562 same stimulus as was presented in the previous trial. Go/No-go inputs x^{GO}, x^{NO} were assumed to follow the
 563 same bell-shaped activation function $s_t = e^{-(t-4)^2}$ on the trials during which their corresponding stimulus
 564 was presented, i.e., $x_{t,k}^{GO} = s_t$ if k was a GO trial, $x_{t,k}^{GO} = 0$ otherwise (and vice versa for No-go input).

565 The stochastic learning process of the Go and No-go weights $\mathbf{w}_k^{GO}, \mathbf{w}_k^{NO} \in \mathbb{R}^N$ over trials was modeled as
 566 a Ornstein-Uhlenbeck process, which was initialized at $\mathbf{w}_0^{GO} = \mathbf{w}_0^{NO} = 1$ and evolved independently across
 567 neurons:

$$568 \quad d\mathbf{w}_k^{GO} = \alpha(\mu^{GO} - \mathbf{w}_k^{GO})dk + \sigma d\mathbf{W}_k \\ 569 \quad d\mathbf{w}_k^{NO} = \alpha(\mu^{NO} - \mathbf{w}_k^{NO})dk + \sigma d\mathbf{W}_k$$

570 where $\alpha_n \sim \mathcal{U}([0.2, 0.8])$ are the neuron-specific learning rates, $\mu^{GO} = 2$, $\mu^{NO} = 0$, $\sigma = 1.3$. Furthermore,
 571 to keep weights non-negative and simulate their saturation, they were clamped to $[0, 2]$. The process was
 572 evaluated using a stochastic differential equation solver and sampled at K evenly spaced points in $[0, 10]$
 573 representing K trials.

574 Top-down modulation was modeled as a rectified Gaussian Process:

$$575 \quad x_{t,k}^{TD} = \max(0, \gamma(t)), \quad \gamma \sim GP(0, \kappa)$$

576 with temporal kernel:

$$577 \quad \kappa(t_1, t_2) = \exp\left(-\frac{(t_1 - t_2)^2}{2l^2}\right)$$

578 where $l = \sqrt{0.5}$. Top-down weights were non-plastic and distributed as $w_n^{TD} \sim \mathcal{U}([0, 1])$.

579 The activity of each neuron was thus given by:

$$580 \quad X_{n,t,k} = w_{n,k}^{GO} x_t^{GO} + w_{n,k}^{NO} x_t^{NO} + w_n^{TD} x_{t,k}^{TD} \\ 581 \quad = w_{n,k}^S s_t + w_n^{TD} x_{t,k}^{TD}$$

582 where the sensory input is combined into $w_{n,k}^S = w_{n,k}^{GO} \mathbb{1}_k^{GO} + w_{n,k}^{NO} (1 - \mathbb{1}_k^{GO})$ where $\mathbb{1}^{GO}$ is an indicator
 583 function that is 1 when trial k is a Go trial and 0 if it is a No-go trial. By construction, the tensor \mathbf{X} has
 584 slice rank of 2, as it can be written in the following form:

$$585 \quad \mathbf{X} = \mathbf{I}^S + \mathbf{I}^{TD}$$

586 where $I_{n,t,k}^S = w_{n,k}^S s_t$ is a time-slicing component representing the weighted, trial-specific sensory input and
 587 $I_{n,t,k}^{TD} = w_n^{TD} \gamma_{t,k}$ is a neuron-slicing component representing top-down modulatory factors that vary over
 588 trials. In our simulations, we used $K = 100$, $T = 90$, $N = 80$.

589 We fitted sliceTCA with non-negativity constraints to the synthetic dataset, using five blocks of 200
 590 iterations each with a learning rate which decayed exponentially over blocks from 0.2 to 0.0125, and a mask
 591 that decayed exponentially over blocks from 0.8 to 0.05. Masked entries changed randomly every iteration.
 592 Initial parameters were drawn uniformly over $[0, 1]$.

593 **4.5 Dataset 1: Motor cortical recordings during a center-out and maze reaching**
594 **task**

595 **4.5.1 Description of the dataset**

596 We analyzed a dataset of motor cortical (M1, $N = 90$) and premotor cortical electrophysiological recordings
597 (PMd, $N = 92$) [Churchland et al., 2012], which is curated and publicly available as part of the 'Neural
598 Latents Benchmark' project [Pei et al., 2021]. Briefly, monkeys were trained to perform a delayed center-out
599 reach task to one of 27 locations in both maze conditions (in which barriers were placed on the screen,
600 leading to curved optimal reach trajectories) and in no maze conditions with matched target locations
601 (classic center-out task leading to straight optimal reach trajectories). The go signal for movement initiation
602 appeared $0 - 1000ms$ after target onset and $1000 - 2600ms$ after the trial started with a fixation cue. We
603 analyzed data from one animal (monkey J) in a single session and randomly subselected 12 target locations,
604 resulting in $K = 246$ single-target trials in the maze reach conditions and $K = 265$ single-target trials in the
605 12 center-out reach conditions with matched target locations.

606 **4.5.2 Additional preprocessing**

607 We calculated firing rates for bins of 10 ms which we then smoothed with a Gaussian filter with $\sigma = 20$ ms
608 and rescaled to minimum and maximum values of 0 and 1 over the course of the experiment for each neuron
609 separately. We selected a time period starting 1 s before movement onset (thus including a substantial part
610 the motor preparation period) and ending 0.5 s after movement onset, when the monkey had successfully
611 reached the target position in the majority of trials. We did not time-warp the data. The resulting data
612 tensor had dimensions of $N = 182$, $T = 150$, and $K = 511$.

613 **4.5.3 Supervised mapping of neural population activity onto kinematic data**

614 To identify the neural subspace from which 2D hand trajectories could be read out (Figure 2a), we used
615 ordinary least squares (OLS). Specifically, we found weights that project the neuron-unfolded data from the
616 full neural space onto a 2D subspace that best maps onto x/y hand velocity with a time delay of 100 ms
617 to account for the lag between neural activity and movement. When testing the decoding analysis after
618 dimensionality reduction we instead applied OLS to the reconstruction (or partial reconstruction, i.e., from
619 only a single slice type) after reshaping it into a $N \times KT$ matrix. We also used OLS to project time-
620 averaged pre-movement activity onto target locations (Figure 2g). For Figure 2h, we used LDA to identify
621 the dimension that best separates pre-movement averaged activity in clockwise vs. counter-clockwise curved
622 reaches in the maze condition. To plot activity in a 3D neural subspace that contained information about the
623 upcoming movement, we then orthogonalized the two axes which map neural activity onto target locations
624 to the axis that distinguishes clockwise and counter-clockwise movements.

625 For all decoding analyses, we calculated R^2 values on left-out trials in a 5-fold cross-validation procedure
626 performed on 100 permutations of the trials. Decoding was performed on data from period spanning 250 ms
627 before to 450 ms after movement onset. For trial-resolved data (Figure 2a, raw data, neuron-slicing, TCA,
628 trial-slicing.), we averaged trial-wise R^2 values, and for pre-movement information on target positions, we
629 calculated a single R^2 value across trials for center-out and maze reaching conditions. For trial-averaged data
630 (Figure 2a, trial-averaged), we performed 2-fold cross-validation by averaging hand and neural trajectories
631 separately for each fold, and then calculating R^2 values averaged over conditions and folds.

632 **4.5.4 Visualization of sliceTCA weights**

633 The results of fitting non-negative sliceTCA are shown in Figure 2c,d and Supplementary Figure 3. Each
634 component consists of a weight vector and a slice of corresponding weights on the other two variables. Along
635 the trial dimension, we sorted trials by the angle of the target position and whether trials belonged to
636 center-out or maze reaching conditions. Along the neuron dimension of trial-slicing components, neurons

637 were sorted by the peak latency of neural activity in the first component. For the time-slicing component,
638 neurons were sorted according to their mean activity in the first reaching condition.

639 **4.5.5 Correlation matrices**

640 To assess the encoding similarity of movement preparation in the time-slicing component, we calculated
641 the $K \times K$ correlation matrix of the neural encoding weights (i.e., the rows of the slice in Figure 2d) for
642 different pairs of trials, separately for center-out and maze reach conditions, and for PMd (Figure 2f) and M1
643 (Supplementary Figure 6). We sorted the resulting correlation matrices by the angle of the target location
644 (Figure 2f).

645 **4.6 Dataset 2: Cortico-cerebellar calcium imaging during a motor task**

646 **4.6.1 Description of the dataset**

647 We analyzed recently published calcium imaging data consisting of simultaneously recorded cerebellar granule
648 cells ($N = 134$) and premotor cortical L5 pyramidal cells ($N = 152$) from a head-fixed mouse performing a
649 motor task in which a manipulandum had to be moved forward and left- or rightward for a reward [Wagner
650 et al., 2019]. After a correct movement was completed, a water reward was delivered with a 1 s delay,
651 followed by an additional 3.5 s inter-trial interval. Left vs. right rewarded turn directions were alternated
652 without a cue after 40 successful trials. We analyzed data from one sessopm of a mouse in an advanced stage
653 of learning, comprising a total of $K = 218$ trials. The data was sampled at a 30 Hz frame rate. Calcium
654 traces were corrected for slow drifts, z-scored and low-pass filtered [Wagner et al., 2019].

655 **4.6.2 Additional preprocessing**

656 Due to the freely timed movement period, we piecewise linearly warped data to the median interval lengths
657 between movement onset, turn, and movement end, respectively. The remaining trial periods were left
658 unwarped and cut to include data from 1.5 s before movement onset until 2.5 s after reward delivery,
659 resulting in a preprocessed $N \times T \times K$ data tensor with $N = 286$, $T = 150$, and $K = 218$.

660 **4.6.3 Visualization of sliceTCA weights**

661 In Figure Figure 4b,c, we show the results of a fitted sliceTCA model. We further reordered trials in the trial-
662 time slices according to trial type, and the neurons in neuron-time slices according to the peak activity in the
663 first trial-loading component. This allows for a visual comparison of tiling structure across components. We
664 used Mann Whitney U-tests on time-averaged activity between reward and end of trial in trial-time slices.
665 We used LDA to determine the classification accuracy for neuron identity (cerebellum vs. cortex) based on
666 the loading vector weights of the three neuron-slicing components found by sliceTCA. We similarly reported
667 classification accuracy of trial identity (error vs. correct, left vs. right) based on the loading vector weights
668 of the trial-slicing components.

669 **4.6.4 Matrix rank of slices**

670 To determine whether sliceTCA finds components with higher matrix rank than methods that do not demix
671 slice types (neuron-slicing PCA and factor analysis (FA) with neuron loadings, neuron-time-concatenated
672 PCA and FA with trial loadings), we performed singular value decomposition (SVD) on the six slices of
673 the sliceTCA model shown in Figure Figure 4b, as well as on the scores of either trial-slicing or neuron-
674 slicing PCA and FA, after refolding the resulting scores into $N \times T$ or $K \times T$ matrices, respectively. We
675 then compare these to the normalized eigenvalue spectra of the slices of the trial-slicing (Figure 4e) or
676 neuron-slicing components (Supplementary Figure 13d). Factor analysis was performed using the Python
677 package “sklearn” [Pedregosa et al., 2011], which uses an SVD-based solver. For comparability with PCA
678 and sliceTCA solutions, no factor rotations were performed.

679 **4.6.5 Manifolds from sliceTCA reconstructions**

680 To analyze the geometry of neural data, we reconstructed the low-slice-rank approximation of neural activity
681 from the sliceTCA model separately for the cerebellum and for the premotor cortex. We then used LDA on
682 both raw and reconstructed data to find the three axes that maximally separate left vs. right correct trials
683 between movement onset and reward (axis 1, shown in Figure 4g), movement onset time vs. the time of
684 reward in all correct trials (axis 2), and the time of motor preparation vs. motor execution (trial start vs.
685 mid-movement, axis 3). We orthonormalized the three axes and projected raw and reconstructed data onto
686 the new, three-dimensional basis Figure 4h.

687 We then measured the distance ratio between trials of the same vs. between trials of a distinct trial class
688 (left vs. right) in the full neural space. For the reconstructed vs. the full data set, we averaged neural activity
689 over a 650 ms window centered at movement onset and measured the Euclidean distance of the population
690 response in each trial to the trial-averaged population response in its own trial type, compared to the
691 Euclidean distance to the average population response of the respective other trial type: $\Delta_{\text{between}}/\Delta_{\text{within}}$,
692 where $\Delta_{\text{within}} = d(x_k^L, \bar{x}^L)$ is the Euclidean distance between population vectors in each left trial to the
693 mean population vector across all left trials (and vice versa for right trials), and $\Delta_{\text{between}} = d(x_k^L, \bar{x}^R)$ is
694 the Euclidean distance of population vectors in each left trial to the mean population vector across all right
695 trials (and vice versa for right trials).

696 **4.7 Dataset 3: Electrophysiology across many brain regions during perceptual
697 decision making**

698 **4.7.1 Description of the dataset**

699 The third analyzed dataset comprised recently published multi-region Neuropixel recordings ($N = 303$) in a
700 mouse performing a perceptual decision making task [IBL et al., 2021]. In the task, mice were presented a
701 grating patch image with varying contrast (0%, 25%, 35%, 50% or 100%), shown on the left or right sides of
702 a screen. The mice were trained to move the image to the center of the screen using a steering wheel within
703 a 60 s period in order to receive a sugary water reward. A correct response was registered if the stimulus
704 was moved to the center, and an incorrect response if the stimulus was moved to the border of the screen.
705 We selected a single example mouse (subject CSHL049 from the openly released ephys data repository).

706 **4.7.2 Additional preprocessing**

707 We binned single-neuron spiking events in 10 ms windows. Due to the variable response times across trials,
708 we piecewise linearly warped data between stimulus onset and reward delivery or respectively, timeout onset,
709 to correspond to the median interval length, and clipped the trial period to start 1 s before stimulus onset and
710 to end 2 s after reward delivery or timeout onset. We smoothed data with a Gaussian filter with $\sigma = 20ms$
711 and rescaled the activity of each neuron to a minimal and maximal value of 0 and 1 over all trials. We
712 excluded neurons with mean firing rates below 0.2 Hz, leading to a total of $N = 221$ neurons analyzed out of
713 $N = 303$ neurons recorded. Brain regions included visual cortex (VIS: anterior layers 2/3, 4, 5, 6a and 6b as
714 well as anteromedial layers 2/3, 4, 5, and 6a; $N = 85$ neurons), hippocampal regions CA1 ($N = 32$ neurons)
715 and dentate gyrus (DG: molecular, polymorph, and granule cell layers; $N = 21$ neurons), thalamus (TH,
716 including posterior limiting nucleus and lateral posterior nucleus; $N = 18$ neurons) and the anterior pretectal
717 and midbrain reticular nucleus (APN, $N = 22$ neurons, and MRN, $N = 35$ neurons) of the midbrain. In
718 total, the resulting data tensor had dimensions $N = 221$, $T = 350$, and $K = 831$.

719 **4.7.3 Visualization of sliceTCA weights**

720 In Figure 5b, we scaled the rows of the neuron-time slices to a [0,1] interval to highlight differences in the
721 timing of peak activity between neurons. We then reordered neuron-time slices by peak activity within
722 each region for each slice type separately, to show characteristic differences between neural correlates of
723 behavioral variables. Trial-time slices were regrouped by trial type to show region-specific representations

724 of task variables. Finally, neuron-trial slices were reordered by average weights across the first 100 trials for
725 each neuron within a region.

726 4.7.4 Reconstruction performance and component weights

727 For each neuron, we estimated the goodness of fit of the sliceTCA reconstruction as:

$$728 1 - \frac{\sum_{n,t,k} (X_{n,t,k} - \hat{X}_{n,t,k})^2}{\sum_{n,t,k} X_{n,t,k}^2}$$

729 We then quantified the contribution of the neuron-slicing components on the total sliceTCA reconstruction
730 for each neuron n as the following ratio:

$$731 f_n^{\text{neuron}} = \frac{\sum_{t,k} \hat{X}_{n,t,k}^{\text{neuron}}}{\sum_{t,k} \hat{X}_{n,t,k}}$$

732 where \hat{X}^{neuron} describes the partial reconstruction of the data tensor from only the neuron-slicing compo-
733 nents. We similarly defined the contributions of the time- and trial-slicing components to the sliceTCA
734 reconstruction of each neuron n as f_n^{time} and f_n^{trial} .

735 4.8 Code availability

736 A GPU accelerated Python library for the sliceTCA data analysis pipeline (including preprocessing, model
737 selection, model optimization, and visualization of components) will be made available upon publication. In
738 addition, the code necessary for reproducing main analyses will be published in a separate Github repository,
739 also upon publication.

740 4.9 Acknowledgements

741 We thank Joao Barbosa, Matthias Hennig, Kishore Kuchibhotla, Ashok Litwin-Kumar, Arno Onken, Yann
742 Sweeney, and the members of the Cayco Gajic lab for comments on the manuscript. We additionally thank
743 Angus Chadwick for helpful discussions on an early stage of the manuscript, and Mark Wagner for sharing
744 his data. We are also grateful to the IBL, the Churchland/Shenoy labs, and the Neural Latents Benchmark
745 project for making their processed and curated data freely available. This work was funded by EMBO
746 (H.S., ALTF 471-2021) and the Agence Nationale de la Recherche (N.A.C.G., ANR-20-CE37-0004; ANR-
747 17-EURE-0017).

748 4.10 Author contributions

749 A.P., H.S., and N.A.C.G. conceptualized the project. A.P. and H.S. designed the data analysis pipeline.
750 A.P. and H.S. performed data analysis investigations. A.P., H.S., and N.A.C.G. designed the feedforward
751 model. A.P. wrote the mathematical notes. N.A.C.G. wrote an initial draft of the manuscript, which all
752 authors reviewed and revised. N.A.C.G. supervised the project.

753
754

755 References

756 M. B. Ahrens, J. M. Li, M. B. Orger, D. N. Robson, A. F. Schier, F. Engert, and R. Portugues. Brain-wide
757 neuronal dynamics during motor adaptation in zebrafish. *Nature*, 485(7399):471–477, 2012.

- 758 R. Amo, S. Matias, A. Yamanaka, K. F. Tanaka, N. Uchida, and M. Watabe-Uchida. A gradual temporal shift
759 of dopamine responses mirrors the progression of temporal difference error in machine learning. *Nature*
760 *Neuroscience*, pages 1–11, 2022.
- 761 M. Balasubramanian and E. L. Schwartz. The isomap algorithm and topological stability. *Science*, 295
762 (5552):7–7, 2002.
- 763 E. Balzani, J. P. Noel, P. Herrero-Vidal, D. E. Angelaki, and C. Savin. A probabilistic
764 framework for task-aligned intra- and inter-area neural manifold estimation, 2022. URL
765 <https://arxiv.org/abs/2209.02816>.
- 766 M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural*
767 *computation*, 15(6):1373–1396, 2003.
- 768 M. Bläser, C. Ikenmeyer, V. Lysikov, A. Pandey, and F. Schreyer. Variety membership testing,
769 algebraic natural proofs, and geometric complexity theory. *CoRR*, abs/1911.02534, 2019. URL
770 <http://arxiv.org/abs/1911.02534>.
- 771 J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way
772 generalization of “eckart-young” decomposition. *Psychometrika*, 35(3):283–319, 1970.
- 773 R. Chaudhuri, B. Gercek, B. Pandey, A. Peyrache, and I. Fiete. The intrinsic attractor manifold and
774 population dynamics of a canonical cognitive circuit across waking and sleep. *Nature neuroscience*, 22(9):
775 1512–1520, 2019.
- 776 M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V.
777 Shenoy. Neural population dynamics during reaching. *Nature*, 487:51–56, 2012.
- 778 J. P. Cunningham and B. M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.
- 780 L. N. Driscoll, N. L. Pettit, M. Minderer, S. N. Chettih, and C. D. Harvey. Dynamic reorganization of
781 neuronal activity patterns in parietal cortex. *Cell*, 170(5):986–999, 2017.
- 782 E. L. Dyer, M. Gheshlaghi Azar, M. G. Perich, H. L. Fernandes, S. Naufel, L. E. Miller, and K. P. Körding.
783 A cryptography-based approach for movement decoding. *Nature biomedical engineering*, 1(12):967–976,
784 2017.
- 785 S. Ebrahimi, J. Lecoq, O. Rumyantsev, T. Tasçi, Y. Zhang, C. Irimia, J. Li, S. Ganguli, and M. J. Schnitzer.
786 Emergent reliability in sensory cortical coding and inter-area communication. *Nature*, 605(7911):713–721,
787 2022.
- 788 T. Feng, D. Silva, and D. J. Foster. Dissociation between the experience-dependent development
789 of hippocampal theta sequences and single-trial phase precession. *Journal of Neuroscience*, 35(12):4890–4902,
790 2015. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2614-14.2015. URL
791 <https://www.jneurosci.org/content/35/12/4890>.
- 792 J. A. Gallego, M. G. Perich, R. H. Chowdhury, S. A. Solla, and L. E. Miller. Long-term stability of cortical
793 population dynamics underlying consistent behavior. *Nature neuroscience*, 23(2):260–270, 2020.
- 794 R. A. Harshman et al. Foundations of the parafac procedure: Models and conditions for an “explanatory”
795 multimodal factor analysis. 1970.
- 796 C. D. Harvey, P. Coen, and D. W. Tank. Choice-specific sequences in parietal cortex during a virtual-
797 navigation decision task. *Nature*, 484(7392):62–68, 2012. ISSN 1476-4687. doi: 10.1038/nature10918.
798 URL <https://doi.org/10.1038/nature10918>.

- 799 J. A. Hennig, E. R. Oby, D. M. Losey, A. P. Batista, M. Y. Byron, and S. M. Chase. How learning unfolds
800 in the brain: toward an optimization view. *Neuron*, 109(23):3720–3735, 2021.
- 801 I. B. L. IBL, V. Aguiillon-Rodriguez, D. Angelaki, H. Bayer, N. Bonacchi, M. Carandini, F. Cazettes, G. Cha-
802 puis, A. K. Churchland, Y. Dan, E. Dewitt, M. Faulkner, H. Forrest, L. Haetzel, M. Häusser, S. B. Hofer,
803 F. Hu, A. Khanal, C. Krasniak, I. Laranjeira, Z. F. Mainen, G. Meijer, N. J. Miska, T. D. Mrsic-Flogel,
804 M. Murakami, J.-P. Noel, A. Pan-Vazquez, C. Rossant, J. Sanders, K. Socha, R. Terry, A. E. Urai,
805 H. Vergara, M. Wells, C. J. Wilson, I. B. Witten, L. E. Wool, and A. M. Zador. Standardized and re-
806 producible measurement of decision-making in mice. *eLife*, 10:e63711, may 2021. ISSN 2050-084X. doi:
807 10.7554/eLife.63711.
- 808 I. B. L. IBL, K. Banga, J. Benson, N. Bonacchi, S. A. Bruijns, R. Campbell, G. A. Cha-
809 puis, A. K. Churchland, M. F. Davatolhagh, H. D. Lee, M. Faulkner, F. Hu, J. Hunterberg,
810 A. Khanal, C. Krasniak, G. T. Meijer, N. J. Miska, Z. Mohammadi, J.-P. Noel, L. Panin-
811 ski, A. Pan-Vazquez, N. Roth, M. Schartner, K. Socha, N. A. Steinmetz, M. Taheri, A. E.
812 Urai, M. Wells, S. J. West, M. R. Whiteway, and O. Winter. Reproducibility of in-vivo elec-
813 trophysiological measurements in mice. *bioRxiv*, 2022. doi: 10.1101/2022.05.09.491042. URL
814 <https://www.biorxiv.org/content/early/2022/05/09/2022.05.09.491042.1>.
- 815 M. Jazayeri and S. Ostojic. Interpreting neural computations by examining intrinsic and embedding dimen-
816 sionality of neural activity. *Current opinion in neurobiology*, 70:113–120, 2021.
- 817 D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. URL
818 <https://arxiv.org/abs/1412.6980>.
- 819 S. A. Koay, A. S. Charles, S. Y. Thibierge, C. D. Brody, and D. W. Tank. Sequential and efficient neural-population coding of complex task information. *Neuron*, 110(2):328–
820 349.e11, 2022. ISSN 0896-6273. doi: <https://doi.org/10.1016/j.neuron.2021.10.020>. URL
821 <https://www.sciencedirect.com/science/article/pii/S0896627321008357>.
- 823 D. Kobak, W. Brendel, C. Constantinidis, C. E. Feierstein, A. Kepecs, Z. F. Mainen, X.-L. Qi, R. Romo,
824 N. Uchida, and C. K. Machens. Demixed principal component analysis of neural population data. *Elife*,
825 5:e10989, 2016.
- 826 T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- 827 K. V. Kuchibhotla, T. Hindmarsh Sten, E. S. Papadoyannis, S. Elnozahy, K. A. Fogelson, R. Kumar,
828 Y. Boubenec, P. C. Holland, S. Ostojic, and R. C. Froemke. Dissociating task acquisition from expression
829 during learning reveals latent knowledge. *Nature communications*, 10(1):1–13, 2019.
- 830 K. C. Lakshmanan, P. T. Sadler, E. C. Tyler-Kabara, A. P. Batista, and B. M. Yu. Extracting low-
831 dimensional latent structure from time series in the presence of delays. *Neural Comput.*, 27(9):1825–1856,
832 sep 2015. ISSN 0899-7667. doi: 10.1162/NECO_a_00759.
- 833 F. Lanore, N. A. Cayco-Gajic, H. Gurnani, D. Coyle, and R. A. Silver. Cerebellar granule cell axons support
834 high-dimensional representations. *Nature Neuroscience*, 24(8):1142–1150, 2021.
- 835 G. W. Lindsay, T. D. Mrsic-Flogel, and M. Sahani. Bio-inspired neural networks implement different recurrent
836 visual processing strategies than task-trained ones do. *bioRxiv*, 2022.
- 837 E. L. Mackevicius, A. H. Bahle, A. H. Williams, S. Gu, N. I. Denisenko, M. S. Goldman, and M. S. Fee. Unsu-
838 pervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience.
839 *Elife*, 8:e38471, 2019.
- 840 L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension
841 reduction. *arXiv preprint arXiv:1802.03426*, 2018.

- 842 T. S. Okubo, E. L. Mackevicius, H. L. Payne, G. F. Lynch, and M. S. Fee. Growth and splitting of
843 neural sequences in songbird vocal development. *Nature*, 528(7582):352–357, 2015. ISSN 1476-4687. doi:
844 10.1038/nature15741. URL <https://doi.org/10.1038/nature15741>.
- 845 A. Onken, J. K. Liu, P. C. R. Karunasekara, I. Delis, T. Gollisch, and S. Panzeri. Using matrix and tensor
846 factorizations for the single-trial analysis of population spike trains. *PLoS computational biology*, 12(11):
847 e1005189, 2016.
- 848 C. Pandarinath, D. J. O’Shea, J. Collins, R. Jozefowicz, S. D. Stavisky, J. C. Kao, E. M. Trautmann,
849 M. T. Kaufman, S. I. Ryu, L. R. Hochberg, et al. Inferring single-trial neural population dynamics using
850 sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.
- 851 S. Panzeri, M. Moroni, H. Safaai, and C. D. Harvey. The structures and functions of correlations in neural
852 population codes. *Nature Reviews Neuroscience*, pages 1–17, 2022.
- 853 N. F. Parker, A. Baidya, J. Cox, L. M. Haetzler, A. Zhukovskaya, M. Murugan, B. Engelhard,
854 M. S. Goldman, and I. B. Witten. Choice-selective sequences dominate in cortical relative to thalamic inputs to nac to support reinforcement learning. *Cell Reports*, 39
855 (7):110756, 2022. ISSN 2211-1247. doi: <https://doi.org/10.1016/j.celrep.2022.110756>. URL
856 <https://www.sciencedirect.com/science/article/pii/S2211124722005204>.
- 858 E. Pastalkova, V. Itskov, A. Amarasingham, and G. Buzsáki. Internally generated cell assembly sequences
859 in the rat hippocampus. *Science*, 321(5894):1322–1327, 2008. doi: 10.1126/science.1159775. URL
860 <https://www.science.org/doi/abs/10.1126/science.1159775>.
- 861 A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga,
862 A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang,
863 J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances
864 in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- 865 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer,
866 R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay.
867 Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- 868 F. Pei, J. Ye, D. M. Zoltowski, A. Wu, R. H. Chowdhury, H. Sohn, J. E. O’Doherty, K. V. Shenoy, M. T.
869 Kaufman, M. Churchland, M. Jazayeri, L. E. Miller, J. Pillow, I. M. Park, E. L. Dyer, and C. Pandarinath.
870 Neural latents benchmark ’21: Evaluating latent variable models of neural population activity. In *Advances
871 in Neural Information Processing Systems (NeurIPS), Track on Datasets and Benchmarks*, 2021. URL
872 <https://arxiv.org/abs/2109.04463>.
- 873 A. J. Peters, S. X. Chen, and T. Komiyama. Emergence of reproducible spatiotemporal activity during
874 motor learning. *Nature*, 510(7504):263–267, 2014. ISSN 1476-4687. doi: 10.1038/nature13235. URL
875 <https://doi.org/10.1038/nature13235>.
- 876 J. Poort, A. G. Khan, M. Pachitariu, A. Nemri, I. Orsolic, J. Krupic, M. Bauza, M. Sahani, G. B. Keller,
877 T. D. Mrsic-Flogel, et al. Learning enhances sensory and multiple non-sensory representations in primary
878 visual cortex. *Neuron*, 86(6):1478–1490, 2015.
- 879 A. Renart and C. K. Machens. Variability in neural activity and behavior. *Current opinion in neurobiology*,
880 25:211–220, 2014.
- 881 M. E. Rule, T. O’Leary, and C. D. Harvey. Causes and consequences of representational drift. *Current
882 Opinion in Neurobiology*, 58:141–147, 2019.
- 883 E. Rybakken, N. Baas, and B. Dunn. Decoding of neural data using cohomological feature extraction. *Neural
884 computation*, 31(1):68–93, 2019.

- 885 O. G. Sani, H. Abbaspourazad, Y. T. Wong, B. Pesaran, and M. M. Shanechi. Modeling behaviorally
 886 relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):
 887 140–149, 2021a.
- 888 O. G. Sani, B. Pesaran, and M. M. Shanechi. Where is all the nonlinearity: flexible nonlinear
 889 modeling of behaviorally relevant neural dynamics using recurrent neural networks. *bioRxiv*, 2021b. doi:
 890 10.1101/2021.09.03.458628. URL
 891 <https://www.biorxiv.org/content/early/2021/09/06/2021.09.03.458628>.
- 892 M. Schimel, T.-C. Kao, K. T. Jensen, and G. Hennequin. ilqr-vae: control-based learning of input-driven
 893 dynamics with applications to neural data. *bioRxiv*, pages 2021–10, 2022.
- 894 C. E. Schoonover, S. N. Ohashi, R. Axel, and A. J. Fink. Representational drift in primary olfactory cortex.
 895 *Nature*, 594(7864):541–546, 2021.
- 896 W. Schultz. Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 80(1):1–27, 1998.
- 897 J. S. Seely, M. T. Kaufman, S. I. Ryu, K. V. Shenoy, J. P. Cunningham, and M. M. Churchland. Tensor
 898 analysis reveals distinct population structure that parallels the different computational roles of areas m1
 899 and v1. *PLoS computational biology*, 12(11):e1005164, 2016.
- 900 K. V. Shenoy, M. Sahani, M. M. Churchland, et al. Cortical control of arm movements: a dynamical systems
 901 perspective. *Annu Rev Neurosci*, 36(1):337–359, 2013.
- 902 M. A.-Y. Smith, K. S. Honegger, G. Turner, and B. de Bivort. Idiosyncratic learning performance in flies.
 903 *Biology Letters*, 18(2):20210424, 2022.
- 904 C. Stringer, M. Pachitariu, N. Steinmetz, M. Carandini, and K. D. Harris. High-dimensional geometry of
 905 population responses in visual cortex. *Nature*, 571(7765):361–365, 2019.
- 906 T. Tao and W. Sawin. Notes on the “slice rank” of tensors, 08 2016. URL
 907 <https://terrytao.wordpress.com/2016/08/24/notes-on-the-slice-rank-of-tensors/>.
- 908 M. Vinck, R. Batista-Brito, U. Knoblich, and J. A. Cardin. Arousal and locomotion make distinct contribu-
 909 tions to cortical activity patterns and visual encoding. *Neuron*, 86(3):740–754, 2015.
- 910 M. J. Wagner, T. H. Kim, J. Kadmon, N. D. Nguyen, S. Ganguli, M. J. Schnitzer, and L. Luo. Shared
 911 cortex-cerebellum dynamics in the execution and learning of a motor task. *Cell*, 177(3):669–682, 2019.
- 912 A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda,
 913 and S. Ganguli. Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple
 914 timescales through tensor component analysis. *Neuron*, 98(6):1099–1115, 2018.
- 915 A. H. Williams, E. Kunz, S. Kornblith, and S. Linderman. Generalized shape metrics on neural representa-
 916 tions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in
 917 Neural Information Processing Systems*, volume 34, pages 4738–4750. Curran Associates, Inc., 2021. URL
 918 <https://proceedings.neurips.cc/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf>.
- 919 S. Zhou, S. C. Masmanidis, and D. V. Buonomano. Neural sequences as an op-
 920 timal dynamical regime for the readout of time. *Neuron*, 108(4):651–658.e5,
 921 2020. ISSN 0896-6273. doi: <https://doi.org/10.1016/j.neuron.2020.08.020>. URL
 922 <https://www.sciencedirect.com/science/article/pii/S0896627320306516>.