nature research

https://doi.org/10.1038/s41593-020-0696-5

Supplementary information

Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making

In the format provided by the authors and unedited

Supplementary Math Note

1 Model description and parameter estimation

This section provides an overview of the model description and parameter estimation. A detailed treatment was provided previously[22].

2 Derivation of model likelihood

The low-dimensional description of the response is represented by a factorization of the vectors $\boldsymbol{\beta}_i^{p\top} = \mathbf{S}_p{}^{\top}\mathbf{w}_i^p$ where, if r_p is the dimensionality of the subspace for task-variable p then $\mathbf{w}_i^p \in \mathbb{R}^{r_p}$ is a neuron-specific vector of weights and $\mathbf{S}_p \in \mathbb{R}^{r_p \times T}$ is a matrix of r_p time courses shared by all neurons. The basic model structure is graphically depicted in Math Note Figure 1. If we let $\mathbf{w}_i{}^{\top} = (\mathbf{w}_i^{1\top}, \ldots, \mathbf{w}_i^{p\top})$, and $\mathbf{S}$ be a block-diagonal matrix

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_1 & & \\ & \ddots & \\ & & \mathbf{S}_P \end{pmatrix} \tag{7}$$

then we can rewrite equation (3) as

$$\mathbf{y}_{i,k} = (\mathbf{x}_k{}^{\top} \otimes I_T)\mathbf{S}^{\top}\mathbf{w}_i + \mathbf{b}_i + \boldsymbol{\epsilon}_{i,k}. \tag{8}$$

If $\mathbf{y}_{i,k}$ and $\mathbf{x}_k$ are the observed response and task variables on trial k then the collection of all observations for this neuron $\mathbf{y}_i{}^{\top} = (\mathbf{y}_{i,1}{}^{\top}, \ldots, \mathbf{y}_{i,K_i}{}^{\top})$ can be described in terms of all corresponding task variables $\mathbf{X}_i{}^{\top} = (\mathbf{x}_1, \ldots, \mathbf{x}_{K_i})$ by

$$\begin{aligned} \mathbf{y}_i &= (\mathbf{X}_i \otimes I_T)\mathbf{S}^{\top}\mathbf{w}_i + \mathbf{1}_K \otimes \mathbf{b}_i + \boldsymbol{\epsilon}_i, & (9)\\ &= \mathbf{F}_i\mathbf{w}_i + \mathbf{b}_i' + \boldsymbol{\epsilon}_i, & (10) \end{aligned}$$

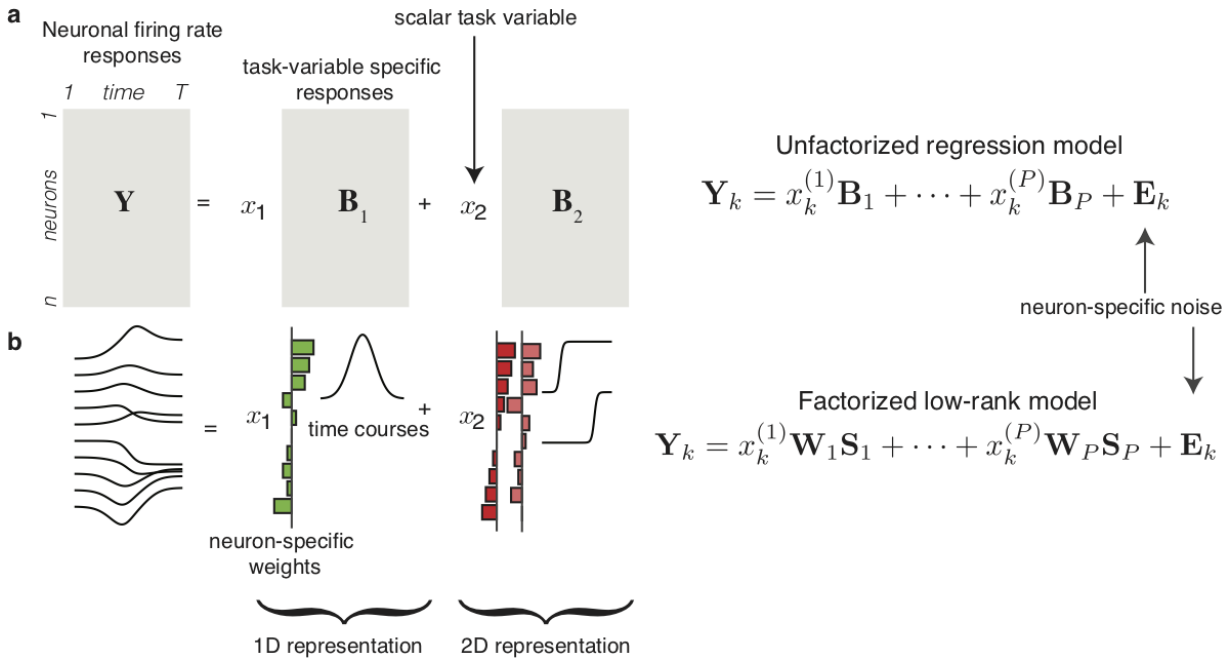
where $\mathbf{F}_i = (\mathbf{X}_i{}^{\top} \otimes I_T)\mathbf{S}^{\top}$, $\mathbf{b}_i' = \mathbf{1}_K \otimes \mathbf{b}_i$, $\mathbf{1}_{K_i}$ is a vector of 1's of length K_i, and $\boldsymbol{\epsilon}_i{}^{\top} = (\boldsymbol{\epsilon}_{i,1}{}^{\top}, \ldots, \boldsymbol{\epsilon}_{i,K_i}{}^{\top})$.

Equation (10) has the form of a standard multivariate linear regression. Therefore, if we set the noise distribution to be $\boldsymbol{\epsilon}_{i,k} \sim \mathcal{N}(0, \lambda_i^{-1} I_T)$ then we have the conditional distribution of $\mathbf{y}_i$ as

$$\mathbf{y}_i | \mathbf{S}, \mathbf{w}_i, \mathbf{D}_i \sim \mathcal{N}(\mathbf{F}_i\mathbf{w}_i + \mathbf{b}_i', \lambda_i^{-1} I_{K_i T}). \tag{11}$$

2.1 Reduced inference in terms of $\mathbf{S}$

Our strategy for accurate estimation is to focus on estimation of only one set of factors ($\mathbf{w}$'s or $\mathbf{S}$'s). While, either set of factors may be selected, the set of factors with lowest dimension should be selected to keep computational costs no higher than necessary. As described in the Methods, we integrate analytically over the $\mathbf{W}_p$'s where the elements of all $\mathbf{W}_p$s are independent and standard normally distributed. Using standard Gaussian identities[51] to marginalize over $\mathbf{w}_i$ we obtain an analytical expression for the marginal likelihood in terms of $\mathbf{s}$. For the given likelihood and prior, our marginal likelihood



Math Note Figure 1: Schematic of low-rank structure in proposed regression model a) The firing rates for n neurons observed over T time point on a give trial can be concatenated into a $n \times T$ matrix $\mathbf{Y}$. A characteristic response for task variable and each neuron can also be described by a $n \times T$ matrix $\mathbf{B}_p$ where the model linearly scales the characteristic response by the task variable. This formulation is equivalent to parameterizing each time point for each neuron as a separate linear regression problem. **b)** A low-dimensional description of the neural responses is achieved by parameterized each of the characteristic response matrices $\mathbf{B}$ by a small number of temporal basis functions. In the case of a 1D representation, a single temporal basis is needed, which is weighted separately to provide the response for each neuron. In the case of a 2D representation, two linearly-independent basis functions are needed, where each basis function gets its own set of weights to construct the characteristic responses of each individual neuron. The low-dimensional description is equivalent to a low-rank matrix factorization model for each $\mathbf{B}_p$.

is given by

$$\mathbf{y}_i|\mathbf{S}, \lambda_i, \mathbf{b}_i \sim \mathcal{N}(\mathbf{b}'_i, \ \lambda_i^{-1} I_{K_i T} + \mathbf{F}_i {\mathbf{F}_i}^\top). \tag{12}$$

Assuming that noise correlations are negligible (in our case neurons are treated as having been recorded sequentially so that this is a reasonable assumption) we observe that neurons are conditionally independent given $\mathbf{S}$. Thus, both the conditional and marginal distributions for the whole population factorize across neurons. Therefore, the population log-likelihood is given by

$$\begin{aligned} \ell(\mathbf{S}, \mathbf{b}, \boldsymbol{\lambda}|\mathbf{w}, \mathbf{y}) &= \frac{1}{2}\sum_{i=1}^{n} TK_i \log|\lambda_i| \qquad (13) \\ &\quad - \lambda_i(\mathbf{y}_i - \mathbf{F}_i\mathbf{w}_i - \mathbf{b}'_i)^\top(\mathbf{y}_i - \mathbf{F}_i\mathbf{w}_i - \mathbf{b}'_i), \\ &= \frac{1}{2}\sum_{i=1}^{n} \ell_i(\mathbf{S}, \mathbf{b}_i, \lambda_i|\mathbf{w}_i, \mathbf{y}_i) \qquad (14) \end{aligned}$$

where K_i is the total number of trials observed for neuron i. The corresponding marginal log-likelihood is given by

$$\ell(\mathbf{S}, \mathbf{b}, \boldsymbol{\lambda}|\mathbf{y}) = \frac{1}{2}\sum_{i=1}^{n} \log|\lambda_i^{-1} I_{K_i T} + \mathbf{F}_i {\mathbf{F}_i}^\top| - (\mathbf{y}_i - \mathbf{b}_i')^\top (\lambda_i^{-1} I_{K_i T} + \mathbf{F}_i {\mathbf{F}_i}^\top)^{-1} (\mathbf{y}_i - \mathbf{b}_i'). \quad (15)$$

2.2 Posterior distribution of $\mathbf{w}_i$'s

Common Gaussian identities can also be used to derive the posterior distribution over weights $\mathbf{w}_i$. As above, conditioned on $\mathbf{S}$, the posterior distribution of all $\mathbf{w}_i$ factorize over neurons and we can write out each distribution independently. For the above model we find that

$$\mathbf{w}_i|\mathbf{y}_i, \mathbf{S}, \lambda_i, \mathbf{b}_i \sim \mathcal{N}(\mathbf{m}_\mathbf{w}, \mathbf{C}_i^{-1}), \quad (16)$$

where

$$\mathbf{m}_\mathbf{w} = \lambda_i \mathbf{C}_i^{-1} \mathbf{S} ({\mathbf{X}_i}^\top \otimes I_T)(\mathbf{y}_i - \mathbf{b}_i'), \quad (17)$$

and $\mathbf{C}_i = \lambda_i \mathbf{S}(\mathbf{A}_i \otimes I_T)\mathbf{S}^\top + I_{\tilde{r}}$.

3 Parameter estimation

We estimate parameters by obtaining maximum likelihood estimates of $\mathbf{s}$, $\mathbf{b}_i$, and $\boldsymbol{\lambda}$ by maximization of the marginal likelihood. The above description of the marginal likelihood $p(\mathbf{y}|\mathbf{s}, \mathbf{b}, \boldsymbol{\lambda})$, the complete data likelihood $p(\mathbf{y}, \mathbf{w}|\mathbf{s}, \mathbf{b}, \boldsymbol{\lambda}) = p(\mathbf{y}|\mathbf{w}, \mathbf{s}, \mathbf{b}, \boldsymbol{\lambda})p(\mathbf{w})$, and the posterior distribution over $\mathbf{w}$, $p(\mathbf{w}|\mathbf{y}, \mathbf{b}, \mathbf{s}, \boldsymbol{\lambda})$, allows us to derive an efficient algorithm for iteratively estimating $\mathbf{s}$, $\mathbf{b}_i$, and $\boldsymbol{\lambda}$ using exclusively closed-form updates. The algorithm is essentially a special case of the "expectation-conditional maximization, either" algorithm (ECME)[52] where parameters are block-wise estimated by either maximizing the conditional expectation of the complete data log likelihood or the marginal likelihood.

3.1 Using ECME and direct maximization

While the EMCE method described above results in accurate estimates of parameters, the ECME method converges very slowly when it gets close to a local optimum. The problem of slow convergence is well documented among EM-type algorithms for related models like factor analysis[52–54]. On the other hand, the ECME method gets close to a local optimum extremely fast.

Alternatively, we could directly maximize the marginal likelihood by gradient decent; an approach we will call maximum marginal likelihood estimation (MMLE). Although in principle the ECME method and the MMLE method should both be maximizing the marginal likelihood, they do so at different rates depending on distance from the optimum. In order to make best use of both methods we initialize using the ECME algorithm, which we parameterize with a liberal stopping criterion, and then complete the estimation procedure with MMLE. We observed this approach to provide faster convergence that either the MMLE or EMCE methods alone.

Dimensionality of each subspace was learned using a greedy algorithm described previously[22].

4 Specifying subspaces

4.1 Subspace Identifiability

We note that the factorization $\mathbf{B}_p = \mathbf{W}_p\mathbf{S}_p$ is not unique and leaves the model parameters only identifiable up to rotation and scalar multiplication. Specifically, note that we can define a orthonormal rotation matrix $\mathbf{P}$ and a scalar α to obtain a new pair of matrices $\mathbf{W}_p^* = \alpha\mathbf{W}_p\mathbf{P}$ and $\mathbf{S}_p^* = \frac{1}{\alpha}\mathbf{P}^\top\mathbf{S}_p$ such that $\mathbf{B}_p = \mathbf{W}_p\mathbf{S}_p = (\alpha\mathbf{W}_p\mathbf{P})(\frac{1}{\alpha}\mathbf{P}^\top\mathbf{S}_p) = \mathbf{W}_p^*\mathbf{S}_p^*$. This non-identifiability is identical to the type of non-identifiability inherent to other matrix factorization models such as factor analysis or probabilistic PCA[55]. Therefore, we require a way of uniquely identifying the subspace spanned by $\mathbf{W}_p$.

We can obtain a fully identifiable subspace by first reconstructing $\mathbf{B}_p$ from the estimated $\mathbf{S}_p$, where the $\mathbf{W}_p$ is estimated from the expectation of the posterior of $\mathbf{W}_p$ given in (17). Each $\mathbf{B}_p$ will then have a unique singular-value decomposition (SVD) denoted by $\mathbf{B}_p = U_p\Sigma_p{V_p}^\top$. We then take the first r_p columns of U_p, denoted $\mathbf{U}_p = (\mathbf{u}_{p,1}, \ldots, \mathbf{u}_{p,r_p})$, to define the encoding subspace of task variable p where we will refer to the jth vector in this subspace as $\mathbf{u}_{p,j}$. In this way, we obtain an orthonormal basis whose orientation gives an ordered set of vectors where the order is with respect to the variance of $\mathbf{B}_p$ explained. We refer to this orientation as the *principle components* (PC) orientation due to its relation to principle components analysis.

4.2 Orthogonalization of Subspaces

The mTDR model does not constrain encoding subspaces between task variables to be orthogonal. It is desirable for visualization to plot only the part of the encoding of each task variable that is unmixed with encoding of other task variables[27]. We therefore orthogonalize the subspaces with respect to correlated subspaces.

To do this we first obtain the PC axes $\mathbf{U}_p$ defined in Math Note 4.1. Orthogonalization of the basis $\mathbf{U}_p$ with respect to some other set of basis vectors $\mathbf{U}_q$ was achieved by the Graham-Schmidt orthogonalization. For example, if we wished to orthogonalize a stimulus subspace with respect to the choice subspace, we form the concatenated matrix $[\mathbf{U}_{\text{choice}}\mathbf{U}_{\text{stim}}]$ and orthogonalize to obtain the orthogonalized basis $[\mathbf{P}_{\text{choice}}\mathbf{P}_{\text{stim}}]$ as in

$$[\mathbf{P}_{\text{choice}}\mathbf{P}_{\text{stim}}] = \text{GS}([\mathbf{U}_{\text{choice}}\mathbf{U}_{\text{stim}}]) \tag{18}$$

where $\text{GS}(M)$ indicates performing Graham-Schmidt on the matrix M. Thus, $\mathbf{P}_{\text{stim}}$ (where “stim" = color or motion), is a set of orthonormal vectors that define the part of the stimulus subspace defined by $\mathbf{U}_{\text{stim}}$ that is orthogonal to $\mathbf{U}_{\text{choice}}$.

5 Projections onto jPCA axes

The low dimensional projections in Figure Figure 4 exhibit rotation-like dynamics. In order to verify the rotational nature of these projections and identify the plane of most rotation-like dynamics, we used jPCA[18] (calculated using Matlab code obtained from http://stat.columbia.edu/ cunningham/). Projections onto the first two jPCA axes are presented in Figure Extended Data 2.

In order to examine whether or not rotational structure was trivially present in our data we first examined

projections of shuffled versions of the data. Each neuron's PSTHs were shuffled with respect to trial type and projected onto the learned task variables axes. No clear sequential or rotational structure is observable (Figure Supplementary figure 2). We performed jPCA on these projections and similarly found no qualitative evidence for rotations (Supplementary figure 3).

To test for the presence of rotations more rigorously, we used a sampling method developed by Elsayed and Cunningham[23] in which we drew 100 samples from the maximum entropy distribution with the same second order moments as the data. We then learned a low-rank model for each sample, identified low-dimensional projections, learned a basis for the jPCA plane, and projected held-out trials onto this plane. From these projections we identified the angle of rotation and constructed a confidence interval (shown by the shaded regions in Figure 4f).

6 Decoding

6.1 Unconditional decoding

Once estimates of $\mathbf{B}_p$ and $\boldsymbol{\lambda}$ and obtained we can decode new trials using maximum likelihood. Because most neurons were not observed simultaneously, the specification of our observations $\mathbf{Y}_k$ in terms of the full set of neurons is incomplete. We accommodate non-sequential observations by specifying the true observations on each trial by $\mathbf{Z}_k = \mathbf{H}_k\mathbf{Y}_k$ where $\mathbf{H}_k$ is an observation matrix. Suppose $n_k < n$ neurons were observed on trial k, then $\mathbf{H}_k$ is a $n_k \times n$ matrix where each row is a "one hot" vector indicating that the corresponding neuron was observed.

If $\mathbf{Z}^*$, $\mathbf{H}^*$ and $\mathbf{x}^*$ are new observations of the population response, observation matrix, and task variables, then the likelihood of $\mathbf{x}^*$, conditional on $\hat{\lambda}_i$'s, and $\hat{\mathbf{B}}_p$'s is given by the data log likelihood defined by (12), which will be proportional to

$$\ell(\mathbf{x}^*|\mathbf{Z}^*, \mathbf{H}^*, \hat{\mathbf{D}}, \hat{\mathbf{B}}) \propto \text{Trace}\left[(\mathbf{Z}^* - \sum_p x^{(p)}\hat{\mathbf{B}}_p - \hat{\mathbf{B}}_0)^\top \mathbf{H}^{*\top}\mathbf{H}^*\hat{\mathbf{D}}\mathbf{H}^\top\mathbf{H}^*(\mathbf{Z}^* - \sum_p x^{(p)}\hat{\mathbf{B}}_p - \hat{\mathbf{B}}_0\right] \tag{19}$$

where $\hat{\mathbf{B}}_p = \hat{\mathbf{W}}_p\hat{\mathbf{S}}_p$.

Differentiating with respect to $x^{(p)}$ gives

$$\frac{\partial\ell(\mathbf{x}^*)}{\partial x^{(p)}} = -2\text{Trace}\left[(\mathbf{Z}^* - \hat{\mathbf{B}}_0)^\top\mathbf{H}^{*\top}\mathbf{H}^*\mathbf{D}\mathbf{H}^{*\top}\mathbf{H}^*\hat{\mathbf{B}}_p\right] + 2\sum_q x^{(q)}\text{Trace}\left[\hat{\mathbf{B}}_p{}^\top\mathbf{H}^{*\top}\mathbf{H}^*\mathbf{D}\mathbf{H}^{*\top}\mathbf{H}^*\hat{\mathbf{B}}_q\right]. \tag{20}$$

If we let $\mathbf{M}^* \equiv (I_T \otimes \mathbf{D}^{1/2}\mathbf{H}^{*\top}\mathbf{H}^*)\left(\ \text{vec}(\mathbf{B}_1), \ldots, \text{vec}(\mathbf{B}_P)\ \right)$ and $\tilde{\mathbf{y}} \equiv \text{vec}(\hat{\mathbf{D}}^{1/2}\mathbf{H}^{*\top}\mathbf{H}^*(\mathbf{Z}^* - \hat{\mathbf{B}}_0))$, then we can write the gradient of $\ell(\mathbf{x})$ in vector form as

$$\frac{\partial\ell(\mathbf{x})}{\partial\mathbf{x}} = -2\mathbf{M}^{*\top}\tilde{\mathbf{y}} + 2\mathbf{M}^{*\top}\mathbf{M}^*\mathbf{x}^*. \tag{21}$$

Setting $\frac{\partial\ell(\mathbf{x})}{\partial\mathbf{x}} = 0$ therefore, yields a closed-form solution for the maximum likelihood estimator for $\mathbf{x}^*$,

$$\hat{\mathbf{x}}^* = (\mathbf{M}^{*\top}\mathbf{M}^*)^{-1}\mathbf{M}^{*\top}\tilde{\mathbf{y}}. \tag{22}$$

This formula is intuitive as we can see that $\tilde{\mathbf{y}}$ is a precision-weighted vector of the new observations, $\mathbf{M}^{*T}\tilde{\mathbf{y}}$ is the projection of these observations onto each of the estimated task variable subspaces, and

$(\mathbf{M}^{*T}\mathbf{M}^{*})^{-1}$ serves to whiten the projection, accounting for the fact that the estimated subspaces are not necessarily orthogonal. The decoding weights are defined as $(\mathbf{M}^{*\top}\mathbf{M}^{*})^{-1}\mathbf{M}^{*\top}\mathbf{D}^{1/2}$.

Instantaneous estimates of $\mathbf{x}^*$ at time t can be obtained by simply restricting $\mathbf{B}_p$ and $\mathbf{Z}^*$ to their t^{th} columns and following the same inference procedure.

6.2 Conditional decoding

If we want to consider some elements of $\mathbf{x}^*$ to be known, then there is a straight forward way to do so. This may be the case, for example, when maximizing the log likelihood, conditioned on the animal's choice when evaluating the log likelihood ratios.

Suppose we let task variables $p = 1, \ldots, q$ be unknown and task variables $p = q+1, \ldots, P$ be known, and let $\mathbf{x}_1 \equiv (x_1, \ldots, x_q)^\top$ and $\mathbf{x}_2 \equiv (x_{q+1}, \ldots, x_P)^\top$. Furthermore, we can define matrices $\mathbf{M}_1^* = \mathbf{D}^{1/2}(\mathbf{B}_1, \ldots, \mathbf{B}_q)$ and $\mathbf{M}_2^* = \mathbf{D}^{1/2}(\mathbf{B}_{q+1}, \ldots, \mathbf{B}_P)$. The maximum likelihood estimator for $\mathbf{x}_1$, conditioned on $\mathbf{x}_2$ is then given by

$$\mathbf{x}_1^* = (\mathbf{M}_1^{*\top}\mathbf{M}_1^*)^{-1}\mathbf{M}_1^{*\top}(\tilde{\mathbf{y}} - \mathbf{M}_2\mathbf{x}_2^*). \tag{23}$$

6.3 Decoding of discrete variables by log likelihood ratio

The task variables in these data are a combination of discrete (choice, context) and continuous (color, motion) variables. It is therefore prudent to respect the domain of the discrete variable when decoding ($x_p \in \{1, -1\}$). For example, when we decode for choice, we first calculate the MLE of the continuous variables, conditioned on the two possible choices (see Math Note 6.2). This results in two vectors of task variable estimates $(\mathbf{x}^+, \mathbf{x}^-)$, one for each choice. We then evaluate the log-likelihood at each of these vectors to calculate the log-likelihood ratio (LLR), which measures the relative information in favor of the two possible categories. For data given by $\mathbf{Z}^*$, the LLR is given by

$$\text{LLR}_p = \ell(x_p = 1|\mathbf{Z}^*, \mathbf{x}^+) - \ell(x_p = -1|\mathbf{Z}^*, \mathbf{x}^-),$$

where $\ell(x_p = 1|\mathbf{Z}^*, \mathbf{x}^+)$ is the log likelihood evaluated at $x_p = 1$, and $\mathbf{x}^+$ is the MLE of all other task variables, conditioned on $x_p = 1$. The inferred probability on a given trial that the data were generated with $x_p = 1$ is therefore given by

$$P(x_p = 1) = \frac{\exp(\text{LLR}_p)}{1 + \exp(\text{LLR}_p)}.$$

The value of this approach to decoding is that we obtain a probability of a trial category at each time point, and not just a candidate category, conditioned on the neural activity. Evaluating the likelihoods with the conditional MLEs $(\mathbf{x}^+, \mathbf{x}^-)$ allows us to account for the confounding effects of the other task variables. The LLRs for context were calculated in an analogous way.

For Figure 7, Extended Data 9, Extended Data 10, Extended Data 8 we evaluated the log likelihoods with the MLE of the stimulus estimates, conditioned on the corresponding discrete variable.

7 Interpretation of projection vectors

In order to draw principled connections between the projected PSTH's and the decoded values of task variables, we adopted the following conventions for projections. We will carefully consider equation (23) and assume that the time-dependent (i.e. task-variable independent) component is given by $\mathbf{B}_P$. For simplicity, let us assume that all neurons have been observed.

First, recall from our description above on unconditional decoding that the projection of the data onto the subspace of unknown task variables at time t is given first by the projection of the normalized quantity

$$\tilde{\mathbf{y}} \equiv \mathbf{D}^{1/2}(\mathbf{y}(t) - \mathbf{B}_0(t))$$

onto the regression weights as in

$$\begin{pmatrix} \mathbf{B}_1(t)^\top \\ \vdots \\ \mathbf{B}_P(t)^\top \end{pmatrix} \mathbf{D}^{1/2}\tilde{\mathbf{y}}. \tag{24}$$

We can write this same expression along with the decomposition of $\mathbf{B}_p$, which is given by

$$\begin{pmatrix} \mathbf{s}_1(t)^\top & & \\ & \ddots & \\ & & \mathbf{s}_P(t)^\top \end{pmatrix} \begin{pmatrix} {\mathbf{W}_1}^\top \\ \vdots \\ {\mathbf{W}_P}^\top \end{pmatrix} \mathbf{D}^{1/2}\tilde{\mathbf{y}}, \tag{25}$$

where $\mathbf{s}_p(t)$ is a length-r_p vector corresponding to the collection of all r_p basis functions for task-variable encoding P at time t.

Therefore, there are two projections that take place to convert the mean-subtracted data into time-varying predictions of task variables. The first takes place by projecting the data onto the subspace defined by $(\mathbf{W}_1, \ldots, \mathbf{W}_P)^\top \mathbf{D}^{1/2}$, which does not change with respect to time and preserves the dimensionality of encoding. The second projection is onto blkdiag$(\mathbf{s}_1(t)^\top, \ldots, \mathbf{s}_P(t)^\top)$, which changes over time and reduces the dimensionality from $\sum_{p=0}^{P} r_p$ to P. Since the encoding subspace should be independent of time we therefore defined the low-dimensional trajectories by

$$\begin{pmatrix} \mathbf{v}_1(t) \\ \vdots \\ \mathbf{v}_P(t) \end{pmatrix} = \begin{pmatrix} {\mathbf{W}_1}^\top \\ \vdots \\ {\mathbf{W}_P}^\top \end{pmatrix} \mathbf{D}^{1/2}\tilde{\mathbf{y}}, \tag{26}$$

where $\mathbf{v}_p(t) \in \mathbb{R}^{r_p}$ is the low-dimensional trajectory for task variable p. Rotations of these projections, such as those plotted using seqPCA were obtained by first identifying the rotation matrix $\mathbf{R}_p$ and projecting onto the rotation as in $\mathbf{R}_p\mathbf{v}_p(t)$.

Therefore, decoding by maximum likelihood (Math Note 6) requires a linear transformation of the low dimensional trajectories $\mathbf{v}_p(t)$. Specifically, the decoded task variables $\mathbf{x}^*(t)$ are given by

$$\mathbf{x}^*(t) = \left(\begin{pmatrix} \mathbf{B}_1(t)^\top \\ \vdots \\ \mathbf{B}_P(t)^\top \end{pmatrix} \mathbf{D} \left(\mathbf{B}_1(t), \ldots, \mathbf{B}_P(t)\right) \right)^{-1} \begin{pmatrix} \mathbf{s}_1(t)^\top & & \\ & \ddots & \\ & & \mathbf{s}_P(t)^\top \end{pmatrix} \begin{pmatrix} \mathbf{v}_1(t) \\ \vdots \\ \mathbf{v}_P(t) \end{pmatrix} \tag{27}$$

Therefore, because the decoding weights vary in time any variation associated with the encoding can be counteracted by the decoding.

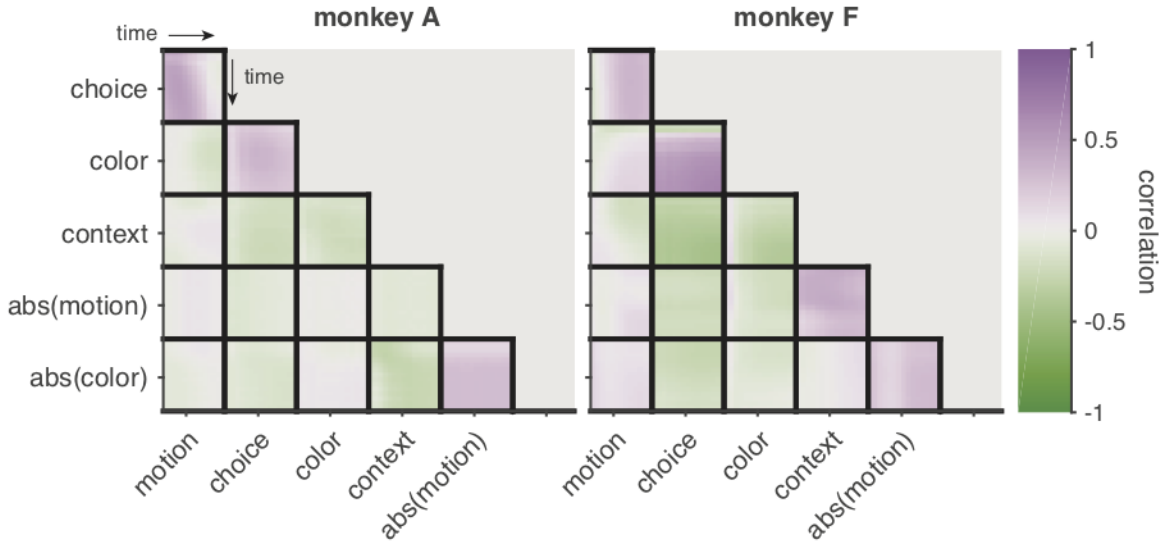
8 Relationship between subspaces

We investigated the degree to which the characteristic responses reflected coordinated activity in two ways. First, we examined the subspaces correlations and second, we examined the degree of agreement between the subspaces thenselves using cannonical correlations analysis (CCA).

8.1 Subspace correlations

Subspace correlations were calculated by taking the cross-correlations between characteristic responses ($\mathbf{B}_p$). Correlated responses imply that the population does not encode task variables independently and the encoding of task variables occurs in a (at least partially) shared subspace.

We examined the cross correlation between characteristic responses of task variables to visualize the change in correlations over time. The results of this analysis are displayed in Math Note Figure 2.

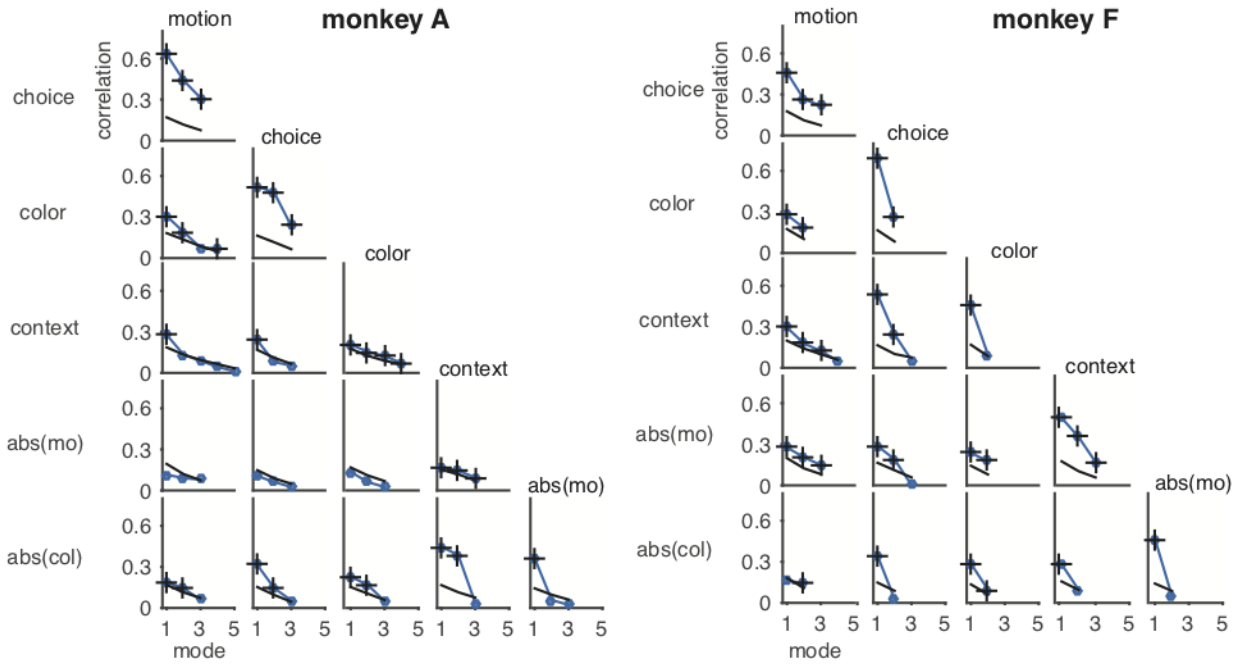


Math Note Figure 2: Cross correlation between characteristic responses of task variables. Motion and color coherence encoding appears to be positively correlated with the choice encoding for both animals.

8.2 Subspace agreement

We analyzed the overlap between task variable subspaces by performing CCA. We used CCA because it allowed us to identify alignment between subspaces that do not have the same dimensionality. The result is a sequence of correlation coefficients that describe mutually orthogonal directions where the subspaces are at least partially aligned. The results of this analysis are presented in Math Note Figure 3.

Signifiant, multi-dimensional overlap for both monkeys were observed between the motion-choice and color-choice subspace pairs. Smaller, but still significant overlap was also observed for motion-color, abs(color)-context, and abs(color)-abs(motion), subspace pairs. Monkey F showed stronger correlations across subspaces than monkey A. Monkey A showed no overlap between the abs(mo) subspace and the motion ,color, or choice subspaces.



Math Note Figure 3: Canonical correlations between task variable subspaces. Canonical correlations measure the degree to which the subspaces overlap. Black lines indicate 95% confidence limit for canonical correlations from 200 randomly permuted axes from the measured subspaces. Markers with "+" indicate the measured canonical correlations that are significantly larger than expected by chance (permutation test, controlling for false discovery rate at .01 level).

9 Sequential PCA (seqPCA)

The goal of seqPCA is to identify a subspace orientation that best describes the data via a sequence of axes in which the order of the axes describes the order in time that each axis dominates the variance of the data.

The basis for seqPCA is constructed as follows: Suppose we have D-dimensional observations $\mathbf{y}_t$ at each time t. We can arrange all of the observations up to time t into a $D \times t$ matrix $\mathbf{Y}^c_{1:t}$ where the index c may refer to trials or conditions. We can arrange the data for all $c = 1, \ldots, C$, up to time t into a $D \times tC$ matrix $\mathbf{Y}_{1:t} = [\mathbf{Y}^1_{1:t}, \ldots \mathbf{Y}^C_{1:t}]$. If the j^{th} singular value of $\mathbf{Y}_{1:t}$ is denoted by $\sigma_{j,t}$ then the fraction of variance that the j^{th} singular vector describes for the first t time points is given by

$$p_{j,t} = \frac{\sigma^2_{j,t}}{\sum_{i=1}^{D} \sigma^2_{i,t}}. \tag{28}$$

If the singular values are ordered such that $\sigma_{1,t} \geq \sigma_{2,t} \geq \cdots \geq \sigma_{D,t}$, then the largest possible variance captured by any single linear dimension at time t is given by $p_{1,t}$.

This construction evokes a sequence of proportions such that $p_{1,t}$ will vary in characteristic ways according to the specific dynamics of the data. For example, if the data project perfectly onto a single dimension then $p_{1,t} = 1$ for all t. If the data are unstructured then $p_{1,t}$ will decrease monotonically until it converges to $p_{1,t} = 1/D$. However, if the data are structured such that the sequential observations progress linearly along a single direction up to time t' and then change direction, then $p_{1,t}$ will increase

up to time t' and then begin to decrease.

In the latter case we can identify the point at which the data begin to change direction as a peak in the $p_{1,t}$ sequence. If t' is the time of this peak then we can identify the first basis vector as the first singular vector at time t' ($\mathbf{u}_{1,t'}$). To identify the next sequential element to this basis we can subtract off the projection onto the first basis as in

$$\mathbf{Y}'_{1:t} = (\mathbf{I} - \mathbf{u}_{1,t'}\mathbf{u}_{1,t'}{}^{\top})\mathbf{Y}_{1:t}, \tag{29}$$

and repeat the process on $\mathbf{Y}'_{1:t'+1}$. The process may be repeated $D-1$ times with the D^{th} vector being completely determined. An orthonormal basis can be constructed from this collection of vectors by Graham-Schmidt orthogonalization.

While the method will always produce a basis for the data, the data never needs to load exclusively onto a single axis at each time point. This is a crucial detail in that exclusive loading onto a single seqPCA axis is a feature of the data, not the method.

We will illustrate the seqPCA method with the following 3D example. A sequential data set was generated by first generating 3 orthogonal vectors that were added. The coordinates of 10 points uniformly spaced on each line were jittered by adding Gaussian noise (Math Note Figure 4a).

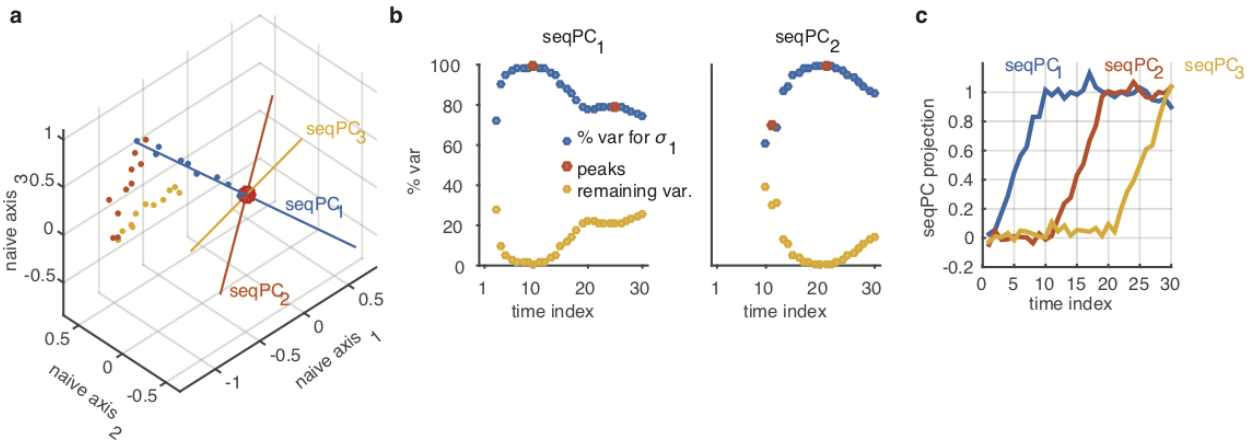
The seqPCA algorithm starts by calculating the variance explained by the first singular vector. As the number of data points increases, the first singular vector explains more of the variance until it reaches the 10th data point, after which it decreases, followed a a second, smaller peak (Math Note Figure 4b, left). This first peak represents the point at which the data matrix includes all of the first ten data points, shown as the blue points in Math Note Figure 4a. After these first 10 points all other points necessarily lie in an orthogonal subspace and the amount of variance explained by any one axis necessarily decreases. Therefore, the first peak represents the number of datapoints to include to calculated the first seqPCA axies (seqPC$_1$), represented by the blue line in Math Note Figure 4a. The index of this peak serves as the temporal boundary between the seqPC$_1$ and seqPC$_2$ axes.

After the seqPC$_1$ has been identified, the projection of the data onto seqPC$_1$ is subtracted from the data as described in (29) and the algorithm picks up the analysis using the residuals from the first temporal boundary. We find a second peak around time index 20 (Math Note Figure 4b, right). This peak correctly identifies the transition between orange and yellow data points in Math Note Figure 4a.

10 Relationship between early/middle/late axes and TDR axes

We compared projections obtained through mTDR and the TDR method proposed previously[1]. While the steps that lead to acquiring a projection axis in the two methods differ substantially, most of these steps are aimed at denoising and regressing the data. The key features of each method is in the selection of the subspace to be analyzed once regression weights have been identified. The TDR analysis chose a single axis for each subspace, corresponding to the regression coefficients at the time index with maximum norm, and then performed an ordered orthogonalization of these axes. Formally, if $\mathbf{B}_p(t)$ is the vector of regression coefficients for task variable p at time index t, then the non-orthogonalized axes are identified by

$$\mathbf{b}_p = \frac{\mathbf{D}^{1/2}\mathbf{B}_p(t_p^{\max})}{\|\mathbf{D}^{1/2}\mathbf{B}_p(t_p^{\max})\|_2} \tag{30}$$



Math Note Figure 4: **a**) Example data (dots) and estimated seqPCA axes (colored axes). **b**) Example of seqPCA vector selection process using motion subspace projections. Blue markers indicate the fraction of variance explained by the first left singular vector ($p_{1,t}$), compared to all remaining dimensions, at each time index. **c**) Projection of data onto the estimated seqPC's.

where

$$t_p^{\text{max}} = \arg\max_t \|\mathbf{D}^{1/2}\mathbf{B}_p(t)\|_2$$

and $\mathbf{D} = \text{diag}(\boldsymbol{\lambda})$ is the diagonal matrix of noise precisions (see Math Note 2). The axes are orthogonalized by first arranging the vectors into a matrix as $[\mathbf{b}_{\text{choice}}\ \mathbf{b}_{\text{motion}}\ \mathbf{b}_{\text{color}}\ \mathbf{b}_{\text{context}}]$ and then orthogonalized by the Graham-Schmidt algorithm. We normalize the regression coefficients by $\mathbf{D}^{1/2}$ to reflect the fact that the neurons were Z-scored prior to regression in the previous analysis[1].

We obtained projection weights for the mTDR method first by identifying the low-rank matrices of regression coefficients by maximum likelihood (ML) as described in previous sections of this supplement (Sections 3, 4.2, 7, and 9), performed an ad hoc orthogonalization on the stimulus and context subspaces (see caption of Fig.Figure 4) and then rotated them (ML + rotation) to obtain the early, middle, and late seqPCA axes for each subspace.

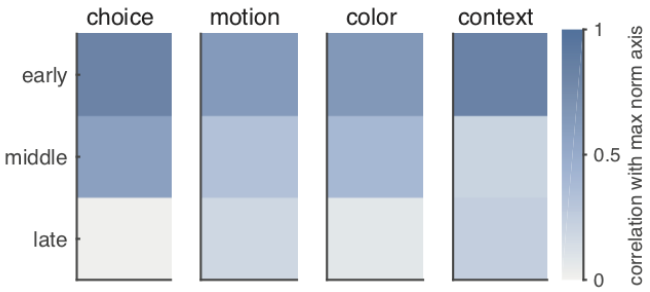
10.1 1D TDR versus multidimensional mTDR and projection magnitudes

Encoding magnitudes were compared (Figure 4e, Extended Data 4) by comparing the projections obtained from the TDR encoding axes described in the previous section with those of the mTDR method where the mTDR projection was summed across early, middle, and late axes. This is appropriate since the three seqPCA axes are orthogonal to each other. While Figure 4e, Extended Data 4 only display the strongest encoding strengths, statistical testing was conducted using pseudotrials drawn for all stimulus strengths. Paired, left-tailed Wilcoxon signed-rank test was used to test whether mTDR more strongly encoded (i.e. projections are further from zero) that those of TDR. The positive false discovery rate[45] (pFDR, controlled at .01) was used to control for multiple comparisons.

10.2 Correlations between TDR and mTDR axes

We examined the correlation (i.e. the normalized inner product) between the early/middle/late axes and the axis selected by the TDR max-norm approach described by equation (30). The correlations

are presented graphically in Math Note Figure 5. Math Note Figure 5 shows that while the TDR axes are weakly correlated with all three seqPCA axes, they are best aligned with the early axis, quantitatively confirming the qualitative similarity between the trajectories from previous TDR analysis and the trajectories presented in Figure 4.



Math Note Figure 5: Correlations between max-norm axes and early/middle/late axes for each subspace For all subspaces the maximum correlation between the max-norm axis and the early axis is larger than the correlation between the max-norm axis and the middle and late axes. $n = 762$ neurons.

Supplementary References

[51] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York:, 2006.

[52] Chuanhai Liu and Donald B Rubin. The ecme algorithm: a simple extension of em and ecm with faster monotone convergence. *Biometrika*, 81(4):633–648, 1994.

[53] Xiao-Li Meng and Donald B Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[54] J-H Zhao, LH Philip, and Qibao Jiang. Ml estimation for factor analysis: Em or non-em? *Statistics and computing*, 18(2):109–123, 2008.

[55] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 611–622, 1999.

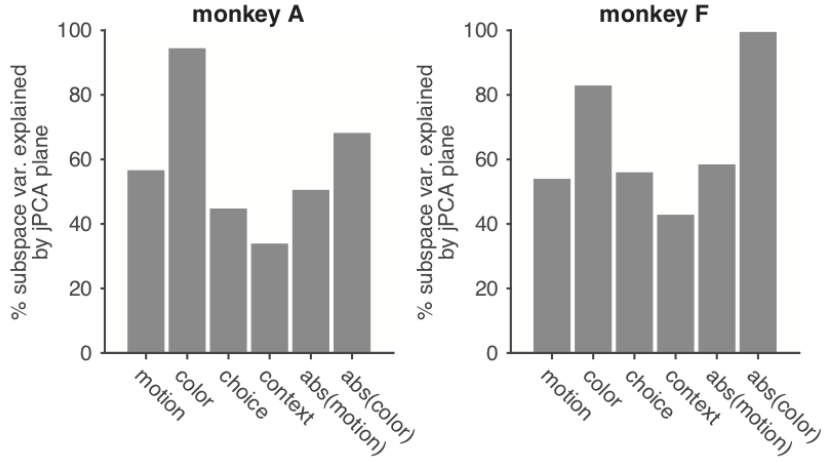
Supplementary Information

Supplementary Table

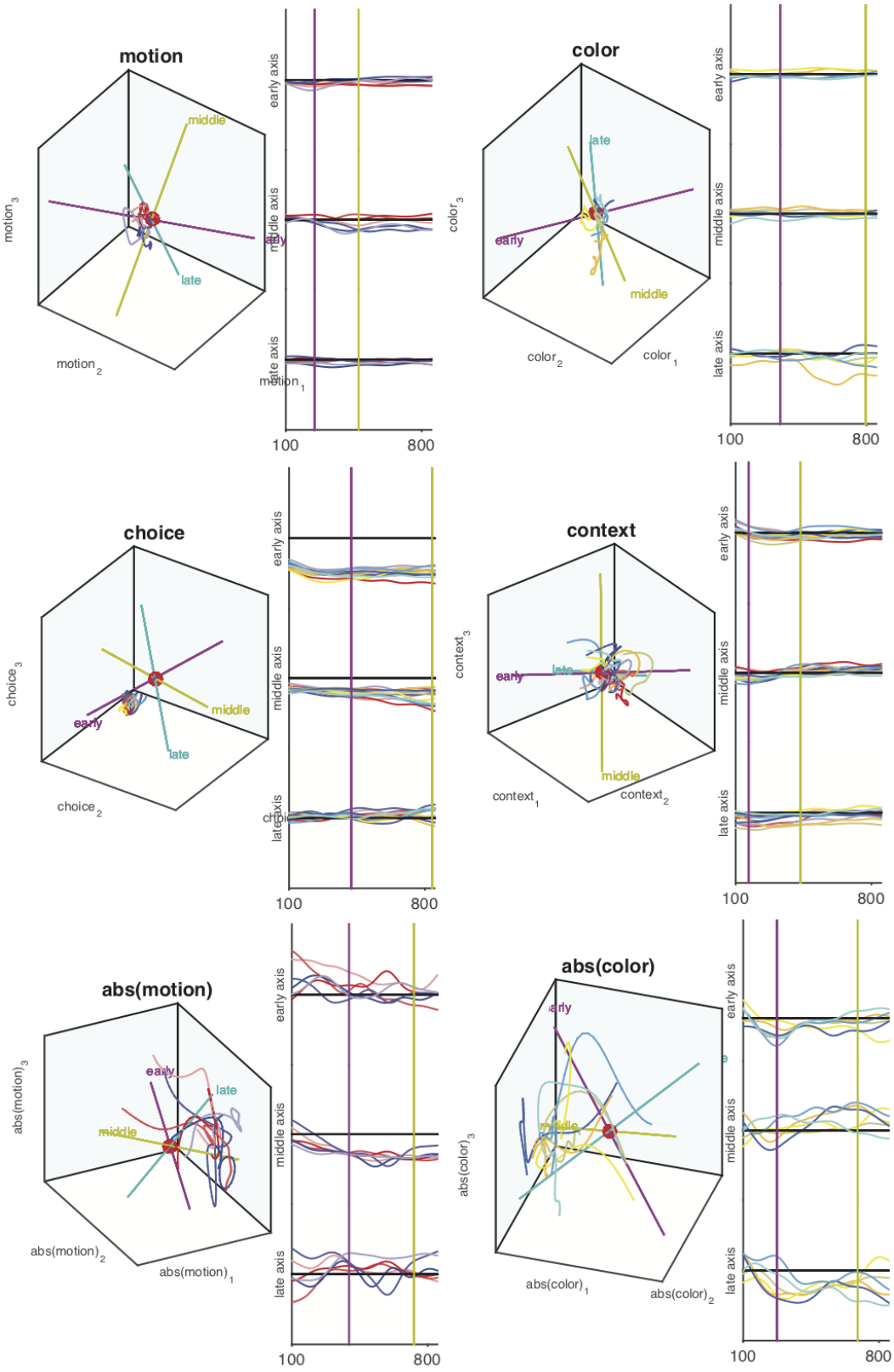
animal	motion	color	choice	context	abs(motion)	abs(color)
monkey A	5	4	3	5	3	3
monkey F	5	2	3	4	3	2

Supplementary Table 1: Summary of estimated dimensionality of each task variable subspace.

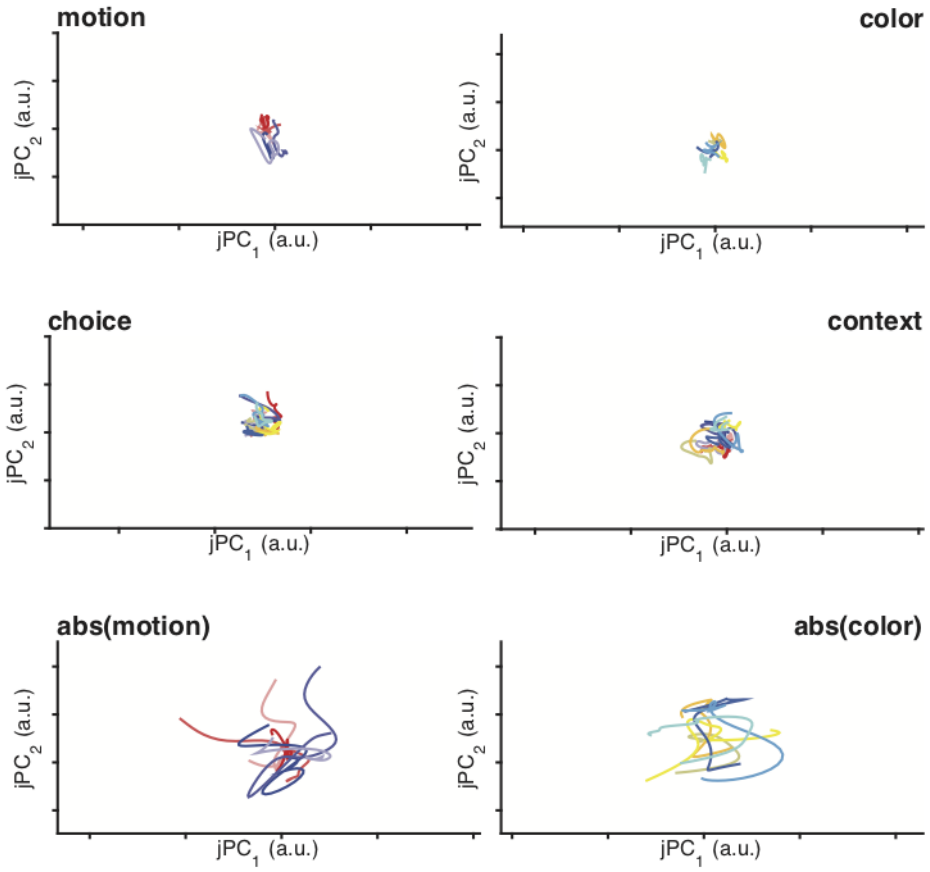
Supplementary Figures



Supplementary figure 1: Variance explained by jPCA axes. Variance was calculated from the same set of PSTHs used in projections in Figure Figure 4 and Extended Data 3. n is different for each task variable. If T is the number of time steps, C_p the number of conditions used for task variable p and and r_p is the dimension of the corresponding subspace, then $n = r_p \cdot C_p \cdot T$.



Supplementary figure 2: Projections of shuffled population PSTH's onto task variable subspaces. Monkey A Characteristic example of projections from the same analysis as in Figure Figure 4 but with conditions of the PSTHs randomly permuted.



Supplementary figure 3: Projections of shuffled population PSTH's onto jPCA axes. Monkey A. Characteristic example of projections onto the jPCA plane from the same analysis as in Figure Extended Data 2 but with conditions of the PSTHs randomly permuted.