doi:10.1038/nature12742

1. Subjects

Two adult male rhesus monkeys, A and F (14 and 12 kg) were trained on a two-alternative, forced-choice, visual discrimination task. Before training, both monkeys were prepared surgically with a head-holding device[1], and monkey A with a scleral search coil for monitoring eye movements[2]. Before electrophysiological recordings, we further implanted a recording cylinder (Crist instruments Co., Inc., Hagerstown, MD) over the arcuate sulcus. Daily access to fluids was controlled during training and experimental periods to promote behavioral motivation. All surgical and behavioral procedures conformed to the guidelines established by the National Institutes of Health and were approved by the Institutional Animal Care and Use Committee of Stanford University.

2. Behavioral task

2.1. Procedures

During both training and experimental sessions monkeys sat in a primate chair with their head restrained. Visual stimuli were presented at 96Hz refresh rate on a CRT monitor placed 43cm from the monkeys' eyes. Eye movements were monitored with a scleral eye coil (*monkey A*, C-N-C Engineering, Seattle, WA) or with an optical eye tracker (*monkeys A and F*, EyeLink 1000, SR Research, ON, Canada); the quality of the latest optical tracker systems are rapidly approaching that of the search coil system[3]. Behavioral control and stimulus presentation were managed by Apple Macintosh G5-based computers (Cupertino, CA) running the Expo software package (Peter Lennie, University of Rochester, NY; Robert Dotson, NYU, NY).

2.2. Task description

On each behavioral trial the monkeys observed a noisy, random-dots motion stimulus presented through a circular aperture. Each random dot stimulus had two properties—motion and color—that the monkey might be required to discriminate, depending upon behavioral context (below). The monkey reported either the prevalent direction of motion or the prevalent color of the stimulus with a saccadic eye movement to one of two visual targets. From trial to trial, motion coherence and color coherence (below) were varied randomly about psychophysical threshold for the discrimination task. Monkeys were rewarded for correct responses with a small quantity of juice.

On any given frame of the stimulus, a fraction of the dots was displayed with one color (color 1), while all others were displayed with a different color (color 2). The difficulty of the color discrimination was varied by parametrically changing the relative number of dots of the two colors, while keeping the total number of dots constant. The fraction of color 1 to color 2 dots, which we call 'color coherence', was fixed throughout the trial (100% coherence: only one color; 0% coherence: equal numbers of dots of the two colors). We define the sign of the color coherence to indicate the dominant color in the stimulus (Fig. 1b, *vertical axis*). In monkey A the dots were either red or green. Monkey F appeared unable to discriminate red and green dots, and was thus trained with blue and orange dots. All colors were matched in luminance.

On each trial a fraction of the dots moved coherently in one of two opposite directions, while the remaining dots were flashed transiently at random locations[4]. The difficulty of the motion discrimination was varied by parametrically changing the fraction of dots moving coherently, which corresponds to the motion coherence of the stimulus (100% coherence: all dots moving in the same direction; 0% coherence: all dots moving randomly). We define the sign of the motion coherence to indicate the direction of coherent motion in the dots (Fig 1b, *horizontal axis*). On any given trial, the motion coherence was identical for dots of color 1 and color 2. Motion and color coherence were chosen randomly on each trial, and were thus completely uncorrelated across trials.

Figure 1a illustrates the sequence of events in each trial. The monkeys initiated a trial by fixating on a small fixation spot, and were subsequently required to maintain fixation within a small window around the fixation point (1.25^{o} radius) until the go cue. The saccade targets appeared 300 ms after the initiation of fixation, and were followed after another 350 ms by the onset of the random-dots. The dots remained on the screen for 750 ms, and their offset was followed by a delay period preceding the go cue. The delay period consisted of an interval of fixed duration (0.3 s) followed by an interval whose duration was drawn from a truncated exponential distribution (mean 0.3 s, truncated at 3 s). The end of the delay period coincided with the disappearance of the fixation point, which served as the go cue, and was followed by the operant saccade to one of the two targets.

The fixation point specified what context the monkey was in. In the motion context, the fixation point was a square, and the monkeys had to discriminate the direction of motion of the dots while ignoring their color. In the color context, the fixation point was a cross, and the monkeys had to discriminate the color of the dots while ignoring their motion. Crucially, both the motion and color evidence were present in the dots on each trial, in one of 36 randomly selected combinations (Fig. 1b).

The two saccade targets varied in location and color from trial to trial (red and green in monkey A, blue and orange in monkey F). In Fig. 1a, for example, the target locations were to the right and left of the dots aperture, and the red and green targets were varied randomly between these two locations from trial to trial. In the motion context, the monkeys were rewarded for saccades to the target location corresponding to the direction of motion of the coherent dots (e.g. a saccade to the right for motion to the right). In the color context, they were rewarded for saccades to the target whose color matched the prevalent color in the dots. We never showed stimuli of 0% motion or color coherence, meaning that each trial could be unambiguously characterized as correct or incorrect. This procedure resulted in half of the trials being 'congruent' (motion and color signals indicating a saccade in the same direction) and half being 'incongruent' (motion and color signals indicating opposite saccades).

The total set of 36 stimuli consisted of all combinations of 6 signed motion coherence levels and 6 signed color coherence levels (Fig. 1b). We varied the coherence levels across monkeys and days to equate performance in the motion and color contexts (average motion coherences: 0.05, 0.15, 0.50 in monkey A, and 0.07, 0.19, 0.54 in monkey F; average color coherences: 0.06, 0.18, 0.50 in monkey A, and 0.12, 0.30, 0.75 in monkey F). For each stimulus the targets could be presented in two configurations (e.g., red target on the right vs. red target on the left) resulting in a total of 72 conditions. These conditions were presented in randomized order within blocks of 72 trials during which the context was kept constant. The end of a block was announced by a tone, and coincided with a change in context. During electrophysiology experiments, the two monkeys completed an average of 28 (monkey A) and 29 blocks per day (monkey F).

3. Electrophysiology experiments

3.1. Procedures

Neurophysiological recordings were performed with tungsten electrodes (2-4MΩ Impedance at 1kHz; FHC Inc., Bowdoin, ME) positioned with a Crist grid (Crist Instruments Co., Inc., Hagerstown, MD) and manipulated with a NAN-drive (NAN Instruments Ltd., Nazareth Illit, Israel). Spiking activity, local field potentials, eye position traces, and digitized task events were recorded using the MAP data-acquisition system (Plexon Inc., Dallas, TX). Spikes were sorted and clustered offline based on principal component analysis using the Plexon offline sorter (Plexon Inc., Dallas, TX). Each well-defined cluster was treated as a ‘unit’ for the purposes of the analyses. Clusters that did not correspond to well discriminated, single-unit activity were classified as multi-unit activity. All data were analyzed with custom scripts written in MATLAB (The MathWorks, Inc., Natick, MA).

3.2. Recording locations

In both monkeys, the majority of units were recorded in the arcuate sulcus (Extended Data Fig. 1a,f). In monkey A, the recordings also extended rostrally onto the prearcuate gyrus and cortex near and lateral to the principal sulcus. Based on anatomical criteria, a majority of the sulcal recordings most likely targeted the frontal eye fields (FEF). Indeed, in monkey A we evoked saccades with low-current electrical microstimulation at several of the recordings locations lying along the sulcus[5].

We made no systematic attempt to assign the recorded units to FEF or any of the other anatomically or functionally defined areas surrounding FEF[6,7]. All signals we studied—choice, motion, color and context—were distributed throughout the full extent of our recording sites (Extended Data Fig. 1). Moreover, even units whose activity is only weakly task modulated contribute to the signals extracted at the level of the population. We therefore combined the activity of units recorded at all locations into a single population response for each monkey, from which we extract the task related signals described below. For convenience, we refer to the entire area covered by our recordings as ‘prefrontal cortex’ (PFC).

3.3. Cell selection and task parameters

We typically recorded neural responses simultaneously from two electrodes lowered in adjacent grid holes. The electrodes were advanced until we could isolate at least one single-unit on each electrode. We first characterized the properties of all units with a visually-guided, delayed saccade task, and proceeded with the context-dependent discrimination task if the activity of units on one or both electrodes was modulated during the delay period of the delayed-saccade task. For the discrimination task, one or both saccade targets were placed in the response fields of a subset of the identified units, as characterized with the delayed-saccade task. However, all recorded units were included in the analysis, irrespective of whether they showed delay-period activity during the saccade task, and irrespective of whether one of the targets was in their response field. The random-dot aperture was positioned eccentrically and did not overlap the fixation point or the saccade targets (typical eccentricity: 8-15°, aperture diameter approximately matching the eccentricity). The average eccentricity of the targets was 16° in monkey A and 15° in monkey F.

In monkey A we recorded from 181 single-units and 581 multi-units in 139 penetrations during 80 recording sessions. On average, we recorded 1,280 trials of the context-dependent discrimination task for each unit, for a total of 163,187 behavioral trials. In monkey F we recorded from 207 single-units and 433 multi-units in 108 penetrations during 60 recording sessions. On average, we recorded 1,237 trials for each unit, for a total of 123,550 behavioral trials.

4. Analysis of behavioral data

We constructed average psychometric curves for each monkey by pooling all trials used in the analyses of the electrophysiology data (discussed below). Each trial was assigned a tag based on the strength of the motion evidence (d_1: weak; d_2: intermediate; d_3: strong) and the strength of the color evidence (c_1: weak; c_2: intermediate; c_3: strong). Trials were pooled based on these tags, rather than on the actual coherence values, which changed somewhat across recording sessions. For simplicity in plotting the behavioral data (Fig. 1, Extended Data Fig. 2a-d), we arbitrarily define one of the target locations as being on the right and the other as being on the left, even in sessions where the two targets were only separated along the vertical dimension. Likewise, we define one of the targets as being green and the other one as being red, even though these were not the colors used in monkey F.

The resulting average psychometric curves indicate that both monkeys integrate the motion and color evidence in the random dots differently in the two contexts (Fig. 1, Extended Data Fig. 2a-d). In particular, the motion evidence has a substantially stronger effect on the monkeys' choices during the motion context (Fig. 1c, Extended Data Fig. 2a) than during the color context (Fig. 1e, Extended Data Fig. 2c). Likewise, the color evidence has a substantially stronger effect on choices during the color context (Fig. 1f, Extended Data Fig. 2d) than during the motion context (Fig. 1d, Extended Data Fig. 2b). As a consequence, the monkeys' choices mostly reflect the evidence that is relevant in a given context. The irrelevant evidence is reflected in the choices as well, but weighs less towards the final decision than the relevant evidence (Fig. 1d,e, Extended Data Fig. 2b,c—the slopes are positive).

We fitted the choices of the monkeys with a simple behavioral model based on a previously published model[8] (Fig. 1c-f and Extended Data Fig. 2, curves). In the model, the motion and color inputs are weighted, summed, and then integrated towards a choice. A change in the weights for motion and color inputs across contexts is largely sufficient to account for the different pattern of choices observed in the two contexts.

5. Mathematical notation

The analysis of the electrophysiology data (section 6), as well as the description of the neural network model (section 7), both involve operations on vectors, matrices, and elements thereof. Throughout the Supplementary Information we use the following notation:

(1) *Vectors* are indicated with lower case, bold letters, for example $\boldsymbol{x}$. The k *th* element of a column vector $\boldsymbol{x}$ is then indicated with the corresponding lower case, non-bold letter, $x(k)$, with k=1 to N for a vector of length N.

(2) *Matrices* are indicated with upper case, bold letters, for example $\boldsymbol{F}$. The element of the matrix $\boldsymbol{F}$ at row k and column j is then indicated with the corresponding upper case, non-bold letter, $F(k, j)$.

(3) *Sets of vectors* are indexed with one or more subscripts, for example $\boldsymbol{x}_i$. The subscript i could for instance index all the units in the population. In that case $\boldsymbol{x}_i$, i=1 to N_{unit}, corresponds to a set of vectors of equal length, one for each unit. Likewise, a set of vectors indexed by unit i and for example time t would be indicated as $\boldsymbol{x}_{i,t}$. The k *th* element of a given vector in the set is given by $x_{i,t}(k)$.

(4) *Sets of matrices*, in analogy, are also indicated with one or more subscripts, for example $\boldsymbol{F}_{i,t}$. All the matrices in the set have the same number of rows and columns. The element of the matrix $\boldsymbol{F}_{i,t}$ at row k and column j is given by $F_{i,t}(k, j)$.

6. Analysis of electrophysiology data

6.1. Pre-processing

We restrict all our analyses to neural responses occurring during the presentation of the random-dots. For each trial, we computed time-varying firing rates by counting spikes in a 50ms sliding square window (50ms steps). The first window was centered at 100ms after the onset of the random-dots stimulus, the last at 100ms after its offset. This temporal interval starts after a characteristic 'dip' in the responses that appears to precede the integration of evidence in prefrontal and parietal neurons.

6.2. Definition of choice 1 and choice 2

We defined choice 1 as the 'preferred' target for each unit based on the activity during the dots presentation. We grouped trials into two subsets based on the location of the chosen target, and compared responses between the two subsets by computing the area under the ROC curve for the corresponding response distributions[9]. We constructed these distributions by pooling responses across all time samples. We defined the target location eliciting larger responses (in terms of the ROC analysis) as choice 1, and the other target location as choice 2.

6.3. Linear regression

We used multi-variable, linear regression to determine how various task variables affect the responses of each recorded unit. We first z-scored the responses of a given unit by subtracting the mean response from the firing rate at each time and in each trial and by dividing the result by the standard deviation of the responses. Both the mean and the standard deviation were computed by combining the unit's responses across all trials and times. We then describe the z-scored responses of unit i at time t as a linear combination of several task variables:

$$r_{i,t}(k) = \beta_{i,t}(1)\, choice(k) + \beta_{i,t}(2)\, motion(k) + \beta_{i,t}(3)\, color(k) + \beta_{i,t}(4)\, context(k) + \beta_{i,t}(5), \quad (1)$$

where $r_{i,t}(k)$ is the z-scored response of unit i at time t and on trial k, $choice(k)$ is the monkey's choice on trial k (+1: to choice 1; -1 to choice 2), $motion(k)$ and $color(k)$ are the motion and color coherence of the dots on trial k, and $context(k)$ is the rule the monkey has to use on trial k (+1: motion context; -1: color context). The sign of the motion and color coherence is defined such that positive coherence values correspond to evidence pointing towards choice 1, and negative values to evidence pointing to choice 2. Thus, the sign of color coherence does not just reflect the color of the dots, but also the location of the red and green targets (which on each trial are presented randomly at one of two possible locations). Motion and color coherence are normalized such that values of -1 and +1 correspond to the largest coherence used in a given session.

The regression coefficients $\beta_{i,t}(v)$, for v=1 to 4, describe how much the trial-by-trial firing rate of unit i, at a given time t during the trial, depends on the corresponding task variable v. Here, and below, v indexes the four task variables, i.e. choice (v=1), motion (v=2), color (v=3) and context (v=4). Notably, in addition to these four task variables, the regression model also included all pairwise interaction terms (i.e. products of two task variables). Inclusion of these interaction terms did not have any substantial effects on the main regression coefficients and they are omitted here for clarity. The last regression coefficient (v=5) captures variance that is independent of the four task variables, and instead results from differences in the responses across time. The signal underlying this variance is discussed in more detail below (section 6.10).

To estimate the regression coefficients $\beta_{i,t}(v)$ we first define, for each unit i, a matrix $\boldsymbol{F_i}$ of size $N_{coef} \times N_{trial}$, where N_{coef} is the number of regression coefficients to be estimated (5), and N_{trial} is the number of trials recorded for unit i. The first four rows of $\boldsymbol{F_i}$ each contain the trial-by-trial values of one of the four task variables. The last row consists only of ones, and is needed to estimate $\beta_{i,t}(5)$. The regression coefficients can then be estimated as:

$$\boldsymbol{\beta}_{i,t} = \left(\boldsymbol{F}_i \boldsymbol{F}_i^{\boldsymbol{T}}\right)^{\boldsymbol{-1}} \boldsymbol{F}_i \boldsymbol{r}_{i,t},$$

where $\boldsymbol{\beta}_{i,t}$ is a vector of length N_{coef} with elements $\beta_{i,t}(v)$, v=1-5. Here and below we denote vectors and matrices with bold letters, and use the same letter (not bold) to refer to the corresponding entries of the vector or matrix, which in this case are indexed by v (see Mathematical Notation above).

6.4. Population average responses

We constructed population responses by combining the condition-averaged responses of units that were mostly recorded separately, rather than simultaneously[10,11]. We defined conditions based on the choice of the monkey (choice 1 or choice 2), the signed motion coherence (Fig. 1b, horizontal axis), the signed color coherence (Fig. 1b, vertical axis), context (motion- or color-relevant), and the outcome of the trial (correct or incorrect). For each unit, trials were first sorted by condition, and then averaged within conditions. We then smoothed the responses in time with a Gaussian kernel (σ = 40ms). Finally, we z-scored the average, smoothed responses of a given unit by subtracting the mean response across times and conditions, and by dividing the result by the corresponding standard deviation. We define the population response for a given condition c and time t as a vector $\boldsymbol{x}_{c,t}$ of length N_{unit} built by pooling the responses across all units for that condition and time. Therefore, the dimension of the state space corresponds to the number of units in the population.

In most figures we analyzed average population responses from correct trials only (note that the linear regression analysis described above was performed on correct *and* incorrect trials). At low coherences, where errors were plentiful, we could plot reliable trajectories for error trials as well (Extended Data Figs. 5 and 9a-e—lowest motion coherence during motion context, lowest color coherence during color context).

In the state-space plots of Fig. 2, we illustrate population responses (trajectories), measured for 36 particularly revealing combinations of these conditions (correct trials only). We first plot trajectories sorted by the relevant sensory signal in each context (Fig. 2a,b and e,f), and then re-plot data from the same trials sorted by the irrelevant sensory signal in each context (Fig. 2c,d):

> *Figures 2a,b, motion context*: 2 choices x 3 relevant motion coherences = 6 trajectories (from 'dots on' to 'dots off'). By definition, when motion is relevant, correct choices occur only when the motion input points *towards* the chosen target (3 conditions per chosen target—strong, intermediate and weak motion towards the chosen target).
>
> *Figure 2c, motion context*: 2 choices x 6 irrelevant color coherences = 12 trajectories (from 'dots on' to 'dots off'). When color is irrelevant, correct choices can occur for color input pointing *towards* or *away* from the chosen target (6 conditions per chosen target—strong, intermediate and weak color toward *either* target.)

And similarly for the color context:

> *Figures 2e,f, color context*: 2 choices x 3 relevant color coherences = 6 trajectories.
>
> *Figure 2d, color context*: 2 choices x 6 irrelevant motion coherences = 12 trajectories.

As in the linear regression analysis, trials are not sorted based on the color of the random-dots *per se*, but based on whether the color pointed towards choice 1 or choice 2.

6.5. Targeted dimensionality reduction

To understand the dynamics of PFC activity in our task, it is critical to identify the components of the population responses that are most tightly linked to the monkeys' behavior. Our ultimate goal is to define a small set of axes, within the state space of dimension N_{unit} defined by the activity of each unit, which independently account for response variance due to key task variables (for a related approach see[10,12]). The projection of the population responses onto these axes yields de-mixed estimates of the task-variables, which are mixed at the level of single neurons.

To define the axes of the subspace, we developed a 'Targeted dimensionality reduction' approach, consisting of three steps described in detail below. We start by using principal component analysis (PCA) to de-noise the population responses and focus our analyses on the subspace spanned by the first $N_{pca} = 12$ principal components (PCs). We then identify directions in this reduced subspace (the de-noised regression vectors defined below) that together account for response variance due to four task variables (choice, motion, color, and context). Finally, we orthogonalize the four identified directions to define axes that account for separate components of the variance due to the task variables.

6.6. Principal component analysis

We used PCA to identify the dimensions in state space that captured the most variance in the condition-averaged population responses. We first build a data matrix $\boldsymbol{X}$ of size $N_{unit} \times (N_{condition} \cdot T)$, whose columns correspond to the smoothed, z-scored population response vectors $\boldsymbol{x}_{c,t}$ defined above for a given condition c and time t (section 6.4). $N_{condition}$ corresponds to the total number of conditions, and T to the number of time samples. The PCs of this data matrix are vectors $\boldsymbol{v}_a$ of length N_{unit}, indexed by a from the PC explaining the most variance to the one explaining the least. We use the first N_{pca} PCs to define a de-noising matrix $\boldsymbol{D}$ of size $N_{unit} \times N_{unit}$:

$$\boldsymbol{D} = \sum_{a=1}^{N_{pca}} \boldsymbol{v}_a \boldsymbol{v}_a^T.$$

The de-noised population response for a given condition and time is defined by:

$$\boldsymbol{X}^{pca} = \boldsymbol{D}\ \boldsymbol{X},$$

with $\boldsymbol{X}^{pca}$ also of dimension of size $N_{unit} \times (N_{condition} \cdot T)$. The overall contribution of the a^{th} PC to the population response at each time point t can be quantified by first projecting the population response onto that PC, and then computing the variance across all conditions of the projection, $var(\ \boldsymbol{v}_a^T \boldsymbol{X})$ (Extended Data Fig. 4b,f).

6.7. Regression subspace

We use the regression coefficients described in Equation 1 above to identify dimensions in state space containing task related variance. For each task variable v=1-4 we first build a set of coefficient vectors $\boldsymbol{\beta}_{v,t}$ whose entries $\beta_{v,t}(i)$ correspond to the regression coefficient for task variable v, time t, and unit i. The vectors $\boldsymbol{\beta}_{v,t}$ (of length N_{unit}) are obtained by simply rearranging the entries of the vectors $\boldsymbol{\beta}_{i,t}$ (of length N_{coef}) computed above (section 6.3). This re-arrangement corresponds to the fundamental conceptual step of viewing the regression coefficients not as properties of individual units, but as the directions in state space along which the underlying task variables are represented at the level of the population. Each vector, $\boldsymbol{\beta}_{v,t}$, thus corresponds to a direction in state space that accounts for variance in the population response at time t, due to variation in task variable v.

We de-noise each vector by projecting it into the subspace spanned by the first $N_{pca} = 12$ principal components:

$$\boldsymbol{\beta}_{v,t}^{pca} = \boldsymbol{D}\ \boldsymbol{\beta}_{v,t},$$

with the set of vectors $\boldsymbol{\beta}_{v,t}^{pca}$ also of length N_{unit}. We refer to these vectors as the 'de-noised' regression coefficients (Extended Data Figs. 1 and 3e,f). This de-noising corresponds to removing from each vector $\boldsymbol{\beta}_{v,t}$ the component lying outside the subspace spanned by the first $N_{pca} = 12$ PCs.

For each task variable v, we then determine the time, t_v^{max}, for which the corresponding set of vectors $\boldsymbol{\beta}_{v,t}^{pca}$ has maximum norm, and define the *time-independent*, de-noised 'regression vectors':

$\boldsymbol{\beta}_v^{max} = \boldsymbol{\beta}_{v,t_v^{max}}^{pca}$ with

$t_v^{max} = \mathrm{argmax}_t \left\| \boldsymbol{\beta}_{v,t}^{pca} \right\|$,

where each $\boldsymbol{\beta}_v^{max}$ is of dimension N_{unit}. Finally, we obtain the orthogonal axes of choice, motion, color, and context (e.g. Fig. 2 and Extended Data Fig. 6) by orthogonalizing the regression vectors $\boldsymbol{\beta}_v^{max}$with the QR-decomposition:

$$\mathbf{B}^{max} = \boldsymbol{Q}\,\boldsymbol{R},$$

where $\mathbf{B}^{max} = [\boldsymbol{\beta}_1^{max}\ \boldsymbol{\beta}_2^{max}\ \boldsymbol{\beta}_3^{max}\ \boldsymbol{\beta}_4^{max}]$ is a matrix whose columns correspond to the regression vectors, $\boldsymbol{Q}$ is an orthogonal matrix, and $\boldsymbol{R}$ is an upper triangular matrix. The first four columns of $\boldsymbol{Q}$ correspond to the orthogonalized regression vectors $\boldsymbol{\beta}_v^{\perp}$, which we refer to as the 'task-related axes' of choice, motion, color, and context. These axes span the same 'regression subspace' as the original regression vectors, but crucially each explains distinct portions of the variance in the responses.

To study the representation of the task-related variables in PFC, we projected the average population responses onto these orthogonal axes (Fig. 2 and Extended Data Figs. 4-7):

$$\boldsymbol{p}_{v,c} = \boldsymbol{\beta}_v^{\perp\,T}\,\boldsymbol{X}_c, \qquad (2)$$

where $\boldsymbol{p}_{v,c}$ is the set of time-series vectors over all task variables and conditions, each with length T. Further, we have reorganized the data matrix, $\boldsymbol{X}$, so that separate conditions are in separate matrices, resulting in a set, $\boldsymbol{X}_c$, of $N_{condition}$ matrices of size $N_{unit}\times T$.

The interpretation of the time-series $\boldsymbol{p}_{v,c}$ depends on the exact definition of the associated axes $\boldsymbol{\beta}_v^{\perp}$. In particular, we interpret the projection of the responses onto the choice axis, $\boldsymbol{p}_{1,c}$, as the integrated relevant evidence, and the projection onto the motion axis, $\boldsymbol{p}_{2,c}$, as the momentary motion evidence (Fig. 2). As discussed below (section 7.6), we validated this interpretation on the simulated model responses, for which these quantities can be precisely defined based on the trained network connectivity (Extended Data Fig. 9h-j). Notably, the same interpretation does not hold if the order of the choice and motion regression vectors is inverted in the orthogonalization step, i.e. if the choice axis contained only the component of the choice regression vector that is orthogonal to the motion regression vector. In that case, the choice and motion axes would both represent mixtures of the integrated and momentary evidence, since the motion regression vector effectively lies along a direction that is intermediate between the one representing the integrated evidence and the one representing momentary motion evidence (Extended Data Fig. 9h-j).

Importantly, the geometric relationships between trajectories of different conditions within the regression subspace spanned by either the regression vectors $\boldsymbol{\beta}_v^{max}$, or the orthogonal axes $\boldsymbol{\beta}_v^{\perp}$, are independent of the particular choice of axes used to describe it. For instance, the effects of motion and color on the population response could have occurred along very similar directions in state space (unlike what we found, Fig. 2), even when described with respect to the orthogonal axes of motion and color. In particular, the orthogonality of the vectors in the basis set used to represent the data has no bearing on whether or not any set of trajectories will appear orthogonal in the corresponding subspace.

6.8. Stability of regression subspace

The set of time-series, $\boldsymbol{p}_{v,c}$, are easiest to interpret if a single regression subspace, spanned by the axes $\boldsymbol{\beta}_v^{\perp}$, captures a large fraction of the task-related variance in the population responses at all times and across both contexts. To assess the stability of the regression subspace across both time and contexts we performed the following two analyses.

First, to assess stability of the regression vectors across time, we estimated *time-dependent* axes $\widetilde{\boldsymbol{\beta}}_{v,t}^{\perp}$ of size N_{unit} for the task variables of motion, color, and context, and compared the ability of time-dependent and time-independent axes (section 6.8) to account for variance in the population activity. We obtained the time-dependent axes by orthogonalizing the matrix $\left[\boldsymbol{\beta}_1^{max}\ \boldsymbol{\beta}_{2,t}^{pca}\ \boldsymbol{\beta}_{3,t}^{pca}\ \boldsymbol{\beta}_{4,t}^{pca}\right]$, where the subscript indexes the four task variables. In this analysis, we held the axis of choice constant (as in section 6.7) since a time-dependent choice axis (i.e. using $\boldsymbol{\beta}_{1,t}^{pca}$ instead of $\boldsymbol{\beta}_1^{max}$ in the orthogonalization above) results in a set of four axes that mix representations of the task variables and are thus difficult to relate to the fixed axes $\boldsymbol{\beta}_v^{\perp}$. For instance, early during the dots presentation in the motion context, a time-dependent choice axis would have large projections onto the fixed axes of choice as well as motion, and thus represent a mixture of integrated and momentary motion evidence. This effect occurs because the integrated and momentary relevant evidence are approximately linearly related to each before the 'decision-boundary' is reached, and are thus difficult to de-mix based only on responses collected early during the dots presentation.

At a specific time t, the projections of the population response onto the time-dependent axes are defined by:

$$\widetilde{\boldsymbol{p}}_{v,c}(t) = \widetilde{\boldsymbol{\beta}}_{v,t}^{\perp\,T}\,\boldsymbol{X}_c(:,t),$$

again yielding a time-series of length T for each task variable and condition, but now computed with time-dependent orthogonal axes. At each time point t, we then compared the variance across conditions c in $\boldsymbol{p}_{v,c}$ (Extended Data Fig. 4d,h; *solid lines*) to the variance in $\widetilde{\boldsymbol{p}}_{v,c}$ (*dashed lines*). On average across all times, the subspace spanned by the fixed axes of motion, color, and context contains 80% (monkey A) and 78% (monkey F) of the variance captured by the corresponding subspace spanned by time-dependent axes of motion, color, and context. Moreover, the variance has similar time courses along the fixed and time-dependent axes (Extended Data Fig. 4d,h). Overall, these observations imply that the representation of the task variables is largely stable across time.

Second, to quantify the effect of context on the task-related axes, we implemented the steps between equations (1) and (2) twice, separately for responses recorded during the motion and color contexts. This yielded two sets of task-related axes $\boldsymbol{\beta}_v^{mot}$ and $\boldsymbol{\beta}_v^{col}$ (v = 1-3, v =1 is choice axis, v =2 is motion axis and v=3 is the color axis), which describe the representation of choice, motion, and color signals separately in the two contexts. We then projected each context-dependent axis into the fixed regression subspace spanned by $\boldsymbol{\beta}_v^{\perp}, f$ = 1-3 and computed its L2-norm:

$$u_v^{mot} = \sqrt{\sum_{g=1}^{3} \left(\boldsymbol{\beta}_v^{mot^T}\ \boldsymbol{\beta}_g^{\perp}\right)^2}$$

$$u_v^{col} = \sqrt{\sum_{g=1}^{3} \left(\boldsymbol{\beta}_v^{col^T}\ \boldsymbol{\beta}_g^{\perp}\right)^2}.$$

The values of u_v^{mot} and u_v^{col} (see Table 1 below) are all close to 1, indicating that the corresponding context-dependent axes lie almost entirely within the regression subspace spanned by the *fixed* axes of choice, motion, and color ($\boldsymbol{\beta}_v^{\perp}$, v = 1-3). Thus a single, fixed set of axes accurately describes the task related responses across both contexts.

	choice	*motion*	*color*
u_v^{mot}	0.98	0.97	0.98
u_v^{col}	0.98	0.98	0.97

Table 1. Overlap between the context dependent axes of choice ($\boldsymbol{v}$ =1), motion ($\boldsymbol{v}$ =2) and color ($\boldsymbol{v}$ =3) and the fixed regression subspace. Numbers correspond to the norm of the projection of a given context-dependent axis into the 3d-subspace spanned by the fixed axes of choice, motion, and color. A norm of 1 implies that a given axis lies entirely within the fixed regression subspace, a norm of 0 that it lies entirely outside.

We also directly compared the direction of the choice axes computed during the motion (u_1^{mot}) and color (u_1^{col}) contexts. These context-dependent choice axes have dot products of 0.92 in monkey A and 0.97 in monkey F, implying that in both monkeys integration of evidence occurs along a direction in state space that is largely stable between contexts.

6.9. Cross validation

We determined to what extent noise in the response of individual units affects the estimation of the regression subspace and the corresponding population trajectories by computing the underlying orthogonal axes $\boldsymbol{\beta}_v^{\perp}$ (Extended Data Fig. 4i-p) and the population trajectories $\boldsymbol{p}_{v,c}$ (Extended Data Fig. 4q,r) twice from non-overlapping subsets of trials. For each unit, we first randomly assigned each trial to one of two subsets, and estimated the corresponding linear regression coefficients separately for the two subsets. These two sets of coefficients were then used to compute two separate sets of axes $\boldsymbol{\beta}_v^{\perp}$ of the regression subspace, following the steps described above. The same two subsets of trials, and the corresponding axes, were then used to generate two sets of population trajectories $\boldsymbol{p}_{v,c}^1$ and $\boldsymbol{p}_{v,c}^2$. To quantify the similarity of trajectories computed from two trial subsets we computed the percentage of variance in the trajectories from one set that is explained by trajectories from the other set (Extended Data Fig. 4q,r, *caption*), for example:

$$100\times\left[1-\Sigma_{v,c,t}\left(\boldsymbol{p}_{v,c}^{1}(t)-\boldsymbol{p}_{v,c}^{2}(t)\right)^{2}/\Sigma_{v,c,t}\left(\boldsymbol{p}_{v,c}^{1}(t)-\langle\boldsymbol{p}_{v,c}^{1}(t)\rangle_{v,c,t}\right)^{2}\right],$$

where $\langle\cdot\rangle_{v,c,t}$ indicates the average over all task variables v, conditions c, and time t.

6.10. Urgency signal

The population trajectories in monkey F showed strong evidence for an 'urgency' signal[13,14]—an overall tendency of the population response to move leftward along the choice axis (toward 'choice 1') irrespective of the direction and strength of the sensory input. This signal has the effect of accelerating the usual leftward movement of the population response on trials in which the sensory evidence points toward choice 1 (Extended Data Fig. 7g,l, filled data points) and attenuating or even reversing the usual rightward movement on trials in which the sensory evidence points toward choice 2 (Extended Data Fig. 7g,l, open data points). By definition, units that prefer choice 2 (which we did not record from) would show equivalent effects in the opposite direction.

To compensate for this urgency signal, in monkey F we also computed 'mean-subtracted' population trajectories $\overline{\boldsymbol{p}}_{v,c}$:

$$\overline{\boldsymbol{p}}_{v,c}=\boldsymbol{p}_{v,c}-\langle\boldsymbol{p}_{v,c}\rangle_{c},$$

where $\langle\cdot\rangle_{c}$ indicates the mean over all conditions. The raw time-series, $\boldsymbol{p}_{v,c}$, are shown in Extended Data Figs. 5f-i and 7g-l, the mean subtracted responses in Extended Data Figs. 6c,d and 7a-f.

In the linear regression analysis (section 6.3) any variance due to the passage of time that is common to all conditions is captured by the last regression coefficient in Eq. 1, $\beta_{i,t}(5)$. A regression vector built from these coefficients would lie mostly outside of the subspace spanned by the task-related axes (see also[10]), with the exception of a projection onto the choice axis corresponding to the urgency signal.

7. Neural network model

We trained a fully recurrent neural network (RNN) composed of nonlinear firing-rate units to solve a context-dependent integration task analogous to that performed by the monkeys. The recurrent feedback within the RNN generates rich dynamics that are particularly appropriate for solving dynamical problems such as selection and integration of inputs over time[15]. Our strategy was to train a randomly initialized RNN to solve the task, incorporating minimal assumptions about network architecture. We then reverse-engineered the network using fixed point analysis and linear approximation techniques to identify the mechanistic basis of the solution 'discovered' by the network[16].

7.2. Network equations

We modeled PFC responses with an RNN defined by the following equations:

$$\tau\dot{\boldsymbol{x}}=-\boldsymbol{x}+\boldsymbol{J}\boldsymbol{r}+\boldsymbol{b}^{c}u_{c}+\boldsymbol{b}^{m}u_{m}+\boldsymbol{b}^{cc}u_{cc}+\boldsymbol{b}^{cm}u_{cm}+\boldsymbol{c}^{x}+\rho_{x} \quad (3)$$

$$\boldsymbol{r}=\tanh(\boldsymbol{x})$$

$z = \boldsymbol{w}^T \boldsymbol{r} + c^z.$

The variable $\boldsymbol{x}(t)$ is a 100-dimensional vector containing the 'activation' of each neuron in the network, and $\boldsymbol{r}(t)$ are the corresponding 'firing rates', obtained by the element-wise application of the saturating nonlinearity, $\tanh$, to $\boldsymbol{x}$. Each neuron in the network has a time constant of decay defined by τ=10ms. The matrix $\boldsymbol{J}$ defines the recurrent connections in the network. The network receives 4-dimensional input, $\boldsymbol{u}(t) = [u_c(t)\, u_m(t)\, u_{cc}(t)\, u_{cm}(t)]^T$, through synaptic weights, $\boldsymbol{B} = [\boldsymbol{b}^c\, \boldsymbol{b}^m\, \boldsymbol{b}^{cc}\, \boldsymbol{b}^{cm}]$. These four inputs represent, respectively, the sensory evidence for color and motion, and the contextual cues instructing the network to integrate either the color or the motion input. Finally, the activations contain contributions from a vector of offset currents $\boldsymbol{c}^x$, and from white noise ρ_x drawn at each time step with standard deviation $3.1623 * \sqrt{\Delta t} \approx 0.1$. To read out the network activity, we defined a linear readout (the output neuron in Fig. 4), $z(t)$, as a weighted sum of the firing rates, with weights, $\boldsymbol{w}$, and bias, c^z. During training, the network dynamics were integrated for the duration T of the random-dots (T=750ms) using Euler updates with Δt=1ms. After training, model dynamics were integrated for an additional 200ms during which the sensory inputs were turned off.

7.3. Network inputs and outputs

The motion and color inputs during the context-dependent integration task were each modeled as one-dimensional, white-noise signals:

$$u_m(t) = d_m + \rho_m(t)$$

$$u_c(t) = d_c + \rho_c(t).$$

The white noise terms ρ_m and ρ_c have zero mean and standard deviation $31.623 * \sqrt{\Delta t} \approx 1$ and are added to the offsets d_m and d_c. The sign of the offset on any given trial can be positive or negative, corresponding to evidence pointing towards choice 1 or choice 2, respectively. The absolute values of the offsets correspond to the motion and color coherence. Notably, the color input is not modeled as color *per se*, but directly as color evidence towards choice 1 or choice 2 (as in the definition of the trial-averaged conditions above). During training, the offsets were randomly chosen on each trial from the range [-0.1875 0.1875]. For the simulations (Fig. 5 and Extended Data Figs. 2e-h and 9) we used 3 coherence values (0.009, 0.036, 0.15), corresponding to weak, intermediate, and strong evidence. These values were chosen to qualitatively reproduce the psychometric curves of the monkeys (Extended Data Fig. 2e-h).

To study the local, linearized dynamics of the response, we delivered transient, 1ms duration input pulses of size 2. After delivery of the pulse, the network was allowed to relax with the motion and color inputs set to zero. The pulses resulted in a deflection along the input axes of approximately the same size as the average deflection for the strongest noisy inputs in the context-dependent integration task (Fig. 5).

During both the contextual integration and the pulse experiments, the contextual inputs were constant for the duration of the trial and defined the context. In the motion context $u_{cm}(t) = 1$ and $u_{cc}(t) = 0$, while in the color context $u_{cm}(t) = 0$ and $u_{cc}(t) = 1$, at each time t.

For the purposes of training, we also defined a 'target' signal, $p(t)$, corresponding to the desired output of the network. The target was defined at only two time steps in each trial. At the first time step (i.e. the onset of the random-dots) the target was zero, i.e. $p(\Delta t) = 0$. At the last time step T (i.e. the offset of the random-dots) the target was either +1 or -1, and corresponded to the correct choice given the inputs and the context. In particular, the sign of the target at time T corresponded to the sign of the motion offset d_m in the motion context, and the sign of the color offset d_c in the color context. At all other time steps between Δt and T the target was undefined, meaning the output value of the network was completely unconstrained and had no impact on synaptic modification.

7.4. Network training

Before training, the network was initialized using standard random initialization. Specifically, the matrix elements J_{ik} were initialized from a normal distribution with zero mean and variance $1/N$, where N=100 is the number of neurons the network. The inputs $\boldsymbol{B}$ were initialized from a normal distribution with zero mean and standard deviation 0.5. The output weights $\boldsymbol{w}$ were initialized to zero. We generated $S = 160{,}000$ trials with randomized inputs to train the network.

We used Hessian-Free optimization for training recurrent neural networks[17,18] (RNNs), which utilizes back-propagation through time[19] (BPTT) to compute the gradient of the error with respect to the synaptic weights. The error was defined as:

$$\frac{1}{ST} \sum_{s \in [1..S]} \sum_{t=\Delta t, T} \left(z_s(t) - p_s(t)\right)^2,$$

where the first sum is over all S trials, and the second over the first and last time steps of the trial. The Hessian-Free method is a second-order optimization method that computes Newton steps. It was recently shown to help ameliorate the well-known vanishing gradient problem[20] associated with training RNNs using BPTT. The input to this supervised training procedure was the initialized RNN, and the input-target pairs that define the context-dependent integration. The result of the training procedure was the set of modified synaptic weights, $\boldsymbol{J}$, $\boldsymbol{B}$, and $\boldsymbol{w}$, the offset currents $\boldsymbol{c}^x$, and the bias, c^z.

The two context-dependent initial conditions of the network at the onset of the trial were not optimized as above. Nevertheless, to prevent small transient activations at the beginning of each trial, we defined the initial conditions with the following procedure. First, we trained two stable fixed points, one for each context. For this purpose, we set $u_c(t)$ and $u_m(t)$ to zero, and either $u_{cc}(t) = 1$ and $u_{cm}(t) = 0$ (motion context) or $u_{cc}(t) = 0$ and $u_{cm}(t) = 1$ (color context). During the first half of the training of the context-dependent integration we used these fixed points as initial conditions. Halfway through the training, and again at the end of training, we found the context-dependent slow points of the dynamics (see below) and reset each initial condition to the slow point resulting in the output closest to zero. Critically, responses beyond the first few time steps after stimulus onset, as well as the dynamical structure uncovered by the fixed-point analysis (see below), did not depend on the choice of initial conditions.

7.5. Fixed point analysis

To discover the dynamical structure of the trained RNN, we followed procedures established by Sussillo and Barak[16]. We found a large sample of the RNN's fixed points and slow points by minimizing the function:

$$q(x) = \frac{1}{2}|\boldsymbol{F}(\boldsymbol{x})|^2,$$

where $\boldsymbol{F}(\boldsymbol{x})$ is the RNN update equation (i.e. the right-hand side of Eq. 3). The function $q(\boldsymbol{x})$ loosely corresponds to the squared speed of the system. Since the network effectively implements two dynamical systems, one for each context, we studied the dynamics separately for the motion and color contexts (see the sine wave generator example in[16]). In each context, we first found the two stable fixed points at the end of the approximate line attractor by setting a tolerance for $q(\boldsymbol{x})$ to 1e-25. To find slow points on the line attractor, we ran the $q(\boldsymbol{x})$ optimization 75 times with a tolerance of 1.0. The identified slow points are approximate fixed points with a very slow drift, which is negligible on the time scale of the normal network operation. These slow points are referred to as fixed points throughout the main text. Any runs of the optimization that found points with $q(\boldsymbol{x})$ greater than the predefined tolerance were discarded and the optimization was run again.

We performed a linear stability analysis around each of the identified slow points, $\boldsymbol{x}^*$. We used the first order Taylor series approximation of the network update equation (Eq. 3) around $\boldsymbol{x}^*$ to create a linear dynamical system, $\delta\dot{\boldsymbol{x}} = \boldsymbol{F}'(\boldsymbol{x}^*)\delta\boldsymbol{x}$**,** and then performed an eigenvector decomposition on the matrix $\boldsymbol{F}'(\boldsymbol{x}^*)$ to obtain a set of left and right eigenvectors, $\boldsymbol{L}$ and $\boldsymbol{R}$ (see section 10 below). For all linear systems one eigenvalue was approximately zero, while all other eigenvalues had a substantially negative real part. The right and left eigenvectors associated with the zero eigenvalue correspond to the line attractor and the selection vector, respectively.

A short introduction to the theory of linear dynamical systems, as well as a detailed description of the procedures underlying the fixed-point analysis of RNNs, are provided in the first half of section 10. The specific mechanism underlying context-dependent selection and integration in the trained RNN is discussed in the second half of section 10.

7.6. Network population responses

We constructed population responses for the RNN following the same procedures as for the PFC data. To display trajectories in state space (e.g. Fig. 5), we projected the population responses onto the axes of a subspace that is analogous to the regression subspace estimated from the PFC data (Extended Data Fig. 9h-j). We found the 'model axes' by orthogonalizing the direction of the right zero eigenvectors averaged over slow points and contexts, and the input vectors, $\boldsymbol{b}^c$ and $\boldsymbol{b}^m$. These model axes closely match the axes of choice, motion, and color estimated with linear regression on the simulated model responses (Extended Data Fig. 9h-j). Unlike the estimated axes, however, the model axes can be defined exactly from the weight matrix of the network, and ultimately directly control the dynamics in the model. The population responses are built from the activations, $\boldsymbol{x}$, because they are directly related to the linear dynamics around the fixed points (see below). Population responses built from the activations are qualitatively similar to population responses built from the firing rates $\boldsymbol{r}$.

7.7. Urgency and instability models

The dynamics of responses along the choice axis in PFC differ somewhat from those observed in our neural network model. In particular, the slopes of the choice predictive signals in PFC depend less on the relevant stimulus coherence than in the model. Moreover, the effects of relevant coherence are asymmetric for the two choices in PFC, but not in the model (compare Extended Data Figs. 5b,f and 9b, *top-left* and *bottom-right*). These differences between the model and the physiological dynamics can be readily explained by previously proposed imperfections in the evidence integration process, such as 'urgency' signals[13,14] or instability in the integrator[21] (Extended Data Fig. 10).

We first studied the effects of urgency and instability on choice predictive activity by modifying a diffusion-to-bound model[22,23]. The temporal evolution of the decision variable $d(t)$ (i.e. of the integrated evidence) is modeled as:

$$d(t + \Delta t) = d(t) + \Delta t \cdot v(t)$$

where $v(t)$ is the drift rate of the diffusion process at time t. The drift rate is given by:

$$v(t) = k\big(C + \kappa + \rho_d(t)\big) + \mu + \lambda d(t)$$

where C is the coherence of the relevant input, $\mu > 0$ is the urgency signal, $\lambda d(t)$ is a drift away from the starting point ($x = 0$) that makes the integration process unstable ($\lambda > 0$), $\rho_d(t)$ is within-trial noise drawn from a normal distribution with standard deviation $\sigma/\sqrt{\Delta t}$, and κ is across-trial noise drawn from a normal distribution with standard deviation ϑ. The diffusion process ends when the decision variable reaches the bound A, i.e. when $d(t) \geq A$, $A > 0$. Here we assume that the choice is the result of a race between two such diffusion processes, one that integrates evidence in favor of choice 1, and the other in favor of choice 2. The diffusion process that first reaches the bound wins the race and determines the choice. The two processes differ only in the parameter k; for the first diffusion process $k = +\alpha$, and for the second $k = -\alpha$, $\alpha > 0$. Even though we used both processes to simulate the behavior, we then computed the integrated evidence in Extended Data Fig. 10a-d by averaging the decision variable from only one of the two processes. The four models in Extended Data Fig. 10a-d are based on the following parameters:

	$\boldsymbol{\alpha}$	$\boldsymbol{\sigma}$	$\boldsymbol{\vartheta}$	$\boldsymbol{\mu}$	$\boldsymbol{\lambda}$	$\boldsymbol{A}$
standard	0.3	0.18	0	0	0	0.05
urgency	0.3	0.17	0	0.04	0	0.05
instability	0.3	0.11	0	0	8	0.05
urgency & instability	0.3	0.10	0	0.04	8	0.05

Table 2. Diffusion model parameters.

We built neural network models that implement instability in the integration or an urgency signal using two different approaches. To build a model with instability (Extended Data Fig. 10k-p and 10s) we used the same approach as in the original model (Fig. 4 and 5), with one important exception: during training of the network we "turned off" the noise in the motion and color inputs ($\rho_m = 0$ and $\rho_c = 0$, section 7.3) as well as the noise in the internal activations of the hidden units of the RNN ($\rho_x = 0$, section 7.2).

To build a model with urgency (Extended Data Fig. 10e-j and 10r) we instead directly trained a neural network model to reproduce the output of a diffusion model with urgency. More precisely, we defined the target signal (section 7.3) as $p(t) = d(t)/A$, where $d(t)$ is the decision variable for one of the two diffusion processes, and A is the decision boundary. We simulated the diffusion model with the following parameters: $\alpha = 0.3, \sigma = 0.1, \vartheta = 0.25, \mu = 0.035, \lambda = 0, A = 0.05$. For all times between the time of the choice (i.e. the time of boundary crossing in one of the diffusion processes) and the end of the stimulus presentation ($t = 750ms$) $x(t)$ was set to its value on the last time step before the choice. The network architecture was analogous to that of the original model (Fig. 4) with the exception of an additional urgency input of constant value 1. On each trial, the relevant sensory input to the RNN (u_m during motion context, u_c during color context, section 7.3) corresponded to the input used to simulate the diffusion process $(C + \kappa + \rho_d(t))$, scaled to have a within-trial standard deviation of $3.1623 * \sqrt{\Delta}t \approx 0.1$. The irrelevant sensory input was generated in an analogous fashion, but had no bearing on $d(t)$. In this model both the motion and color inputs, as well as the urgency input, were turned off after the time of the choice. For the simulations in Extended Data Fig. 10e-j we used three coherence values C (0.12, 0.25. 0.50) that were higher than in the original model (Fig. 5). These coherence values and diffusion model parameters were chosen to achieve a qualitative match between the model predictions and the data for monkey A (both behavior and population trajectories). Different parameters or coherences result in network models with dynamical features analogous to those in Extended Data Fig. 10r.

8. Simulation of alternative population responses

To demonstrate that the properties of the population responses observed in PFC are not a trivial consequence of our analysis methods, we simulated population responses expected from the four basic mechanisms of context-dependent selection illustrated in the cartoon drawings in Fig. 3 of the main text (Extended Data Fig. 8). These simulations are based on the assumption that the responses of individual PFC neurons represent mixtures of the following four task variables: (1) the momentary motion evidence, $s^m(t)$; (2) the momentary color evidence, $s^c(t)$; (3) the integrated relevant evidence, $s^r(t)$ (integrated motion evidence in the motion context, integrated color evidence in the color context); (4) context, $s^x(t)$. Our strategy was to construct four variants of a diffusion-to-bound model of decision-making, each mimicking one of the selection mechanisms in Fig. 3 of the main text. As described below, each variant was constructed by altering the weightings of the four task variables (listed above) onto simulated single units. Each of the underlying selection mechanisms features highly heterogeneous, mixed coding like that commonly observed in PFC, yet each is characterized by a distinct, readily identifiable pattern of population activity in state space. In addition, we show that standard "single unit" regression analyses of the simulated data are singularly unhelpful in revealing the underlying mechanisms, and in one common analysis, generate conclusions that are simply wrong.

8.1. Mixtures of task variables

We simulated each task variable, $s^m(t), s^c(t)$, $s^r(t)$ and $s^x(t)$ for 500 experimental sessions of 1296 trials each (see section 8.2 below); for example, $s^r_{i,k}(t)$ is the integrated relevant evidence for session i on trial k at time t. We then simulated the responses of *sequentially* recorded neurons (one per session), by mixing the task variables from a given experimental session. Specifically, we generated the firing rate responses of neuron i by mixing the task variables for session i:

$$r_{i,k}(t) = \alpha_m(i)s_{i,k}^m(t) + \alpha_c(i)s_{i,k}^c(t) + \alpha_r(i)s_{i,k}^r(t) + \alpha_x(i)s_{i,k}^x(t) + r_b + noise,$$

where the mixing weights correspond to the components of four "mixing vectors" $\boldsymbol{\alpha}_m$, $\boldsymbol{\alpha}_c$, $\boldsymbol{\alpha}_r$, and $\boldsymbol{\alpha}_x$, r_b is the baseline response, and the $noise$ is drawn from a normal distribution of zero mean. The standard deviation of the $noise$ was chosen such that the variability in $r_{i,k}(t)$ is consistent with a point process with Fano factor of 1.

We simulated population responses corresponding to different selection mechanisms by varying the relationship between the four mixing vectors. We first built four nearly orthogonal 500-dimensional vectors $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$, $\boldsymbol{\alpha}_3$, and $\boldsymbol{\alpha}_4$ whose components were randomly drawn from a normal distribution. We then defined the mixing vectors as follows:

Observed PFC responses (Extended Data Fig. 8a-c and Fig. 3a). To simulate population responses resembling those we observed in PFC, we set $\boldsymbol{\alpha}_r = \boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}_2$, $\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_3$, and $\boldsymbol{\alpha}_x = \boldsymbol{\alpha}_4$ on all trials. Thus "mixed selectivity" is a standard feature of the simulated unit responses; all signals contribute to the responses of individual "units", with weights being randomly mixed across units.

Context-dependent early selection (Extended Data Fig. 8d-f and Fig. 3b). To simulate population responses expected by context-dependent early selection, we set $\boldsymbol{\alpha}_r = \boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_x = \boldsymbol{\alpha}_4$ on all trials, $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_c = 0.2\ \boldsymbol{\alpha}_1$ in the motion context, and $\boldsymbol{\alpha}_m = 0.2\ \boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_1$ in the color context. Again, all signals contribute to the responses of individual units, although the weights of the irrelevant momentary evidence are on average 5 times smaller than those of the relevant momentary evidence (corresponding to the ratio between the corresponding behavioral weights, h_m and h_c, see below).

Context-dependent input direction (Extended Data Fig. 8g-i and Fig. 3c). To simulate population responses expected by context-dependent input directions, we set $\boldsymbol{\alpha}_r = \boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_x = \boldsymbol{\alpha}_4$ on all trials, $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_3$ in the motion context, and $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}_2$, $\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_1$ in the color context. As above, all signals contribute to the responses of individual units, but the ensemble representation of the sensory inputs (i.e. the state space direction) varies across contexts.

Context-dependent output direction (Extended data Fig. 8j-l and Fig. 3d). To simulate population responses expected by context-dependent output direction, we set $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}_2$, $\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_3$, $\boldsymbol{\alpha}_x = \boldsymbol{\alpha}_4$ on all trials, $\boldsymbol{\alpha}_r = \boldsymbol{\alpha}_2$ in the motion context, and $\boldsymbol{\alpha}_r = \boldsymbol{\alpha}_3$ in the color context. Yet again, mixed selectivity is typical of unit responses, but the ensemble representation of the choice varies across contexts.

Importantly, the same four task variables are mixed in the neural population in all four cases—the four simulations differ only in the geometrical relationship between the corresponding mixing vectors, not in the strength of the corresponding task variables (with the exception of *early selection*, where the irrelevant momentary evidence is strongly attenuated with respect to the relevant momentary evidence).

We then analyzed these simulated population responses by applying the same methods used to analyze the responses recorded in PFC (Extended Data Fig. 8; Supp. Information, sections 6.3-6.8). Traditional single unit regression methods reveal a multitude of signals that are mixed at the level of single neurons, as expected, but provide little obvious insight into how these signals are represented in the population (compare Extended Data Fig. 8b,e,h,k). In fact, the raw regression coefficients obtained with linear regression ($\boldsymbol{\beta}_{v,t_v^{max}}$ in section 6.7) can be rather misleading. The regression coefficients of choice, for example, are correlated with the coefficients of motion (Extended Data Fig. 8b, *top left*) and color (*top middle*), even when by construction the inputs and the integrated evidence are represented along orthogonal directions in state space (i.e. $\boldsymbol{\alpha}_r$, $\boldsymbol{\alpha}_m$, and $\boldsymbol{\alpha}_c$). Moreover, estimating the overall strength of the motion and color inputs in the population by simply averaging the absolute values of the corresponding regression coefficients leads to the erroneous conclusion that the relevant input is stronger than the irrelevant input (e.g. Extended Data Fig. 8c,i,l), even when by construction the inputs are not modulated by context. The trajectories computed with targeted dimensionality reduction, on the other hand, faithfully capture the properties of the task variables as specified by the mixing vectors and clearly distinguish between the different selection mechanisms (compare Extended Data Fig. 8a,d,g,j).

Importantly, the data in Extended Data Fig. 8, unlike the cartoons in Fig. 3, reflect actual simulations of population responses based on the diffusion-to-bound model. We generated population responses expected from the four selection mechanisms by imposing different relationships between the mixing vectors. Despite the very different underlying mixing vectors, the resulting population responses are generically similar in that they all incorporate, 1) mixed selectivity at the single unit level, and 2) coding of irrelevant as well as relevant sensory information in the population (with the exception of early selection, where the irrelevant input is attenuated; see above). Critically, when analyzed with our targeted dimensionality reduction technique, each mechanism gives rise to a distinct pattern of state space trajectories that matches nicely to the corresponding cartoon pattern in Fig. 3 of the main text (compare the left column of Extended Data Fig. 8a,d,g,j, gray scale traces, to text Fig. 3a,b,c,d). Thus the structure of the observed PFC population responses (e.g. Fig. 2) is not "imposed" by our analysis methods, and is not an inevitable consequence of the existence of mixed signals in single units.

8.2. Generation of task variables

We generated the four task variables by simulating a diffusion model (see also section 7.7). The integrated evidence is based on the time-dependent decision variable, $s^r(t) = d(t)$, where:

$$d(t + \Delta t) = d(t) + \Delta t \cdot v(t).$$

The underlying drift of the diffusion process $v(t)$ here is computed as:

$$v(t) = h_m s^m(t) + h_c s^c(t) + \mu,$$

where $s^m(t)$ and $s^c(t)$ are the momentary motion and color evidence, respectively, and $\mu = 0.1$ is the urgency signal. The diffusion process ends when the decision variable reaches the bound $A = 0.1$. The motion gain h_m is 0.8 during the motion context, and 0.16 during the color context. Likewise, the color gain h_c is 0.16 during the motion context, and 0.8 during the color context. The momentary motion and color evidence are defined as:

$$s^m(t) = C_m + \kappa_m + \rho_m(t)$$

and

$s^c(t) = C_c + \kappa_c + \rho_c(t)$.

The within-trial noise $\rho_m(t)$ and $\rho_c(t)$ is drawn from normal distributions with standard deviation $\sigma/\sqrt{\Delta t}$, $\sigma = 0.05$, and the across-trials noise κ_m and κ_c is drawn from normal distributions with standard deviation $\vartheta = 0.25$. Finally, the context signal is a constant, $s^x(t) = 1$ in the motion context, and $s^x(t) = -1$ in the color context. We simulated the diffusion process for all combinations of 6 motion coherences C_m (±0.03, ±0.12, ±0.5), 6 color coherences C_c (±0.03, ±0.12, ±0.5), and two contexts (motion and color context), for a total of 6x6x2=72 conditions. We simulated each condition 18 times within each experimental session, for a total of 1296 trials per session.

The diffusion model variables $s^m(t)$, $s^c(t)$, $s^r(t)$ and $s^x(t)$ for session i and trial k are then scaled to obtain the task variables $s_{i,k}^m(t)$, $s_{i,k}^c(t)$, $s_{i,k}^r(t)$ and $s_{i,k}^x(t)$ (see section 8.1 above), which are mixed to obtain the simulated neural firing rates. The task variables are defined as: $s_{i,k}^m(t) = 0.04 \cdot s^m(t)$, $s_{i,k}^c(t) = 0.04 \cdot s^c(t)$, $s_{i,k}^r(t) = 1 \cdot s^r(t)$, $s_{i,k}^x(t) = 0.02 \cdot s^x(t)$. These scaling factors are fixed across the four selection mechanisms, and result in simulated population responses (Extended Data Fig. 8a-c) with across-condition variance along the task related axes of choice, motion, color, and context that qualitatively match those observed in PFC (e.g. Fig. 2 and Extended Data Fig. 6).

9. Supplementary references

1 Evarts, E. V. A technique for recording activity of subcortical neurons in moving animals. *Electroencephalogr Clin Neurophysiol* **24**, 83-86, (1968).

2 Judge, S. J., Richmond, B. J. & Chu, F. C. Implantation of magnetic search coils for measurement of eye position: an improved method. *Vision Res* **20**, 535-538, (1980).

3 Kimmel, D. L., Mammo, D. & Newsome, W. T. Tracking the eye non-invasively: simultaneous comparison of the scleral search coil and optical tracking techniques in the macaque monkey. *Front Behav Neurosci* **6**, 49, (2012).

4 Britten, K. H., Shadlen, M. N., Newsome, W. T. & Movshon, J. A. The analysis of visual motion: a comparison of neuronal and psychophysical performance. *J Neurosci* **12**, 4745-4765, (1992).

5 Bruce, C. J., Goldberg, M. E., Bushnell, M. C. & Stanton, G. B. Primate frontal eye fields. II. Physiological and anatomical correlates of electrically evoked eye movements. *J Neurophysiol* **54**, 714-734, (1985).

6 Petrides, M. Lateral prefrontal cortex: architectonic and functional organization. *Philos Trans R Soc Lond B Biol Sci* **360**, 781-795, (2005).

7 Schall, J. D. in *Cerebral Cortex* Vol. 12 (eds K.S. Rockland, J. H. Kaas, & A. Peters) (Plenum Press, 1997).

8 Gold, J. I. & Shadlen, M. N. The influence of behavioral context on the representation of a perceptual decision in developing oculomotor commands. *J Neurosci* **23**, 632-651, (2003).

9 Shadlen, M. N. & Newsome, W. T. Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey. *J Neurophysiol* **86**, 1916-1936, (2001).

10 Machens, C. K., Romo, R. & Brody, C. D. Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex. *J Neurosci* **30**, 350-360, (2010).

11 Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487**, 51-56, (2012).

12 Machens, C. K. Demixing population activity in higher cortical areas. *Front Comput Neurosci* **4**, 126, (2010).

13 Churchland, A. K., Kiani, R. & Shadlen, M. N. Decision-making with multiple alternatives. *Nat Neurosci* **11**, 693-702, (2008).

14 Reddi, B. A. & Carpenter, R. H. The influence of urgency on decision time. *Nat Neurosci* **3**, 827-830, (2000).

15 Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**, 544-557, (2009).

16 Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput* **25**, 626-649, (2013).

17 Martens, J. & Sutskever, I. Learning recurrent neural networks with hessian-free optimization. *Proceedings of the 28th International Conference on Machine Learning.* (2011).

18 Martens, J. Deep learning via Hessian-free optimization. *Proceedings of the 27th International Conference on Machine Learning.* (2010).

19 Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE.* 1550-1560 (1990).

20 Bengio, Y., Simard, P. & Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks.* 157-166 (1994).

21 Brunton, B. W., Botvinick, M. M. & Brody, C. D. Rats and humans can optimally accumulate evidence for decision-making. *Science* **340**, 95-98, (2013).

22 Smith, P. L. & Ratcliff, R. Psychology and neurobiology of simple decisions. *Trends Neurosci* **27**, 161-168, (2004).

23 Palmer, J., Huk, A. C. & Shadlen, M. N. The effect of stimulus strength on the speed and accuracy of a perceptual decision. *J Vis* **5**, 376-404, (2005).

10. Mathematical explanation of selective integration in the RNN

Introduction

The standard equations to define an RNN are given by

$$\tau \dot{x}_i = -x_i + \sum_k^N J_{ik} r_k + \sum_k^I B_{ik} u_k + b_i^x \tag{1}$$

$$r_i = \mathrm{h}\left(x_i\right) \tag{2}$$

$$z = \sum_k^N w_k r_k + b^z, \tag{3}$$

where x_i is the "activation" of the i^{th} neuron, and $r_i = \mathrm{h}(x_i)$ is the associated "firing rate", defined as the application of the saturating nonlinear function. Each of the N neurons in the network has a time constant of decay defined by τ, and so in isolation an individual neuron acts as a low-pass filter. The recurrence of the network is defined by the matrix $\mathbf{J}$ and it is this feedback that gives the RNN its power, along with the nonlinearity on the firing rates. The matrix elements, J_{ik}, are initialized from a normal distribution with zero mean and variance, $1/N$. The network receives I-dimensional input, $\mathbf{u}(t)$, through synaptic weights $\mathbf{B}$. Finally, there is a vector of offset currents, $\mathbf{b}^x$. In order to read out the network solution to a given problem, it is common to define a linear readout, z, defined as a weighted sum of the firing rates with weights $\mathbf{w}$, plus a bias, b^z.

These networks are not spiking networks. One typically thinks of a single firing rate variable, r_i, as being the population averaged firing rate of many spiking neurons[1]. In an RNN, the activation and firing rate variables, x_i and r_i, can take analog values, again like a population average of spiking neurons. Further, the network is continuous in time. The natural time scales of the network are related to both τ and the nature of the feedback as defined by $\mathbf{J}$.

The selective integrating RNN (siRNN) employed in this paper parametrizes equation 1 with $N = 100$, $I = 4$ (the two sensory inputs of color and motion as well as the two contextual inputs for color and motion, $\mathbf{B} = [\mathbf{b}^c\ \mathbf{b}^m\ \mathbf{b}^{cc}\ \mathbf{b}^{cm}]$), $\tau = 10\text{ms}$ and $\mathrm{h}() = \tanh()$. The values of the weights and biases were set via an optimization approach described in the Suppl. Information section 7.4.

Motivation of our approach

Most often in computational studies in neuroscience a network model is designed by hand to implement a specific function, such as a decision[2], or auto completion of memories[3].

This study was different. Instead we took a machine learning approach and trained the RNN with a powerful optimization technique, specifically the Hessian-Free optimization technique recently proposed for neural networks by Martens and Sutskever[4]. We make no claims about the biological validity of the training approach. Rather, our goal was to study solutions to the problem of selective integration that were nonlinear, dynamical and distributed (i.e. implemented by the interactions of simple units), and where the solution was not explicitly built into the network. We trained many networks (around 100) from different initializations of the weights and biases, and each time the network solved the problem in the same qualitative way. This points to the fact that the selective integration task (see Fig. 1 in main text) placed strong constraints on the optimization process.

The Hessian-Free optimization technique is a supervised learning algorithm, which means that the training routine compares the actual outputs of the RNN to the desired outputs and changes the synaptic weights and biases to reduce this error. Because the supervised training algorithm specifies *which* function to perform without specifying *how* to perform it, the precise mechanisms underlying an RNN's solution are often completely opaque. Such a network is often referred to as a "black box", implying that it fundamentally cannot be understood. However, new techniques have recently been developed for understanding RNN functionality[5], and we employ these approaches extensively in the current paper. Our aim was to "crack" open the network and potentially discover novel solutions to the problem of selective integration.

In what follows, we explain how the trained siRNN functions. We proceed in a general analytic sequence: defining fixed points and line attractors, linearization around fixed points, and using the eigenvector decomposition to understand dynamics of linear systems. This very general approach to elucidating network mechanism is mandated by our original decision to train the siRNN without building in any specific solution to the computational problem. Achieving a post hoc understanding of the trained siRNN, or indeed any system trained in this manner, is greatly facilitated by this sequence of analyses, which are described individually in the next several sections. Readers who are already familiar with these techniques may wish to skip ahead to the section entitled, "Understanding selective integration", which provides a concise explanation of how the siRNN works. We omit precise details of the siRNN training procedure, which are provided in Suppl. Information section 7.4.

Understanding how the network functions

Fixed points and line attractors

As shown in the main article, the siRNN creates two approximate line attractors, each bounded at both ends by a stable attractor. The line attractors are defined by the context input, meaning only one line attractor ever exists during a given trial. Depending on the contextual input (i.e. the motion context or the color context), a line attractor implements an accumulation of noisy evidence of one or the other input streams. Once the RNN has accumulated enough evidence, and in so doing, moved along the line far enough in either direction, the network dynamics become fixed in one of two attractor states at either end of the line attractor, representing the decision made by the RNN.

Surprisingly, given the relative simplicity of the system, to good approximation we can understand the behavior of the selective integrator in terms of simple linear algebra, discarding most nonlinear and dynamical aspects of the RNN. To begin, we step away from the siRNN for a moment, and instead consider a generic dynamical system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \tag{4}$$

where the state is defined by the N-dimensional vector, $\mathbf{x}$, and the update rules are defined by $\mathbf{F}$, a vector function. Fixed points are vectors $\mathbf{x}^*$ such that

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}^*) = \mathbf{0}, \tag{5}$$

so the system is at equilibrium at such a point. Fixed points are either stable or unstable. For a stable fixed point, if the state of the system is started near the fixed point, the state converges to it. For an unstable fixed point, the state diverges away. Stable fixed points are also called attractors, and are the mechanism underlying memories in Hopfield networks[3]. Unstable fixed points are not observed in the siRNN.

Finally, one can also have a line attractor, which is a 1-dimensional manifold (a line, possibly curved) of fixed points with the property that there is zero motion in the direction of the line and decaying dynamics towards the line. A famous example of a line attractor in neuroscience is given by Seung[6], where he explains how the eyes can simultaneously take many positions and are nevertheless kept still. For the siRNN, the purpose of a line attractor is to represent the amount of accumulated evidence towards one choice or another. Since there are two contexts in the siRNN, there are two contextually defined line attractors.

Line attractors require perfect tuning in order to have zero motion along the direction of the line. In practice, such fine tuning does not exist, so a series of fixed points that approximate a line attractor are in fact what we find in the trained siRNN (see Fig. 5 in the main text). On such an approximate line attractor, there are a few true fixed points, i.e. $\dot{\mathbf{x}} = \mathbf{0}$, but mostly there are slow points, i.e. $\dot{\mathbf{x}} \approx \mathbf{0}$, such that there is very mild drift along the line.

Linear approximations

The main reason fixed points are important when trying to understand a nonlinear dynamical system, such as an RNN, is that a region always exists around a fixed point, sometimes small and sometimes large, where the system can be understood in essentially linear terms, i.e. as a linear dynamical system. We can see this with just a few lines of math involving the Taylor series expansion of the update equations, $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$. Consider the Taylor expansion of $\mathbf{F}(\mathbf{x})$ around a fixed point in state space, $\mathbf{x}^*$:

$$\dot{(\mathbf{x}^* + \boldsymbol{\delta}\mathbf{x})} = \mathbf{F}(\mathbf{x}^* + \boldsymbol{\delta}\mathbf{x}) = \mathbf{F}(\mathbf{x}^*) + \mathbf{F}'(\mathbf{x}^*)\boldsymbol{\delta}\mathbf{x} + \frac{1}{2}\boldsymbol{\delta}\mathbf{x}\mathbf{F}''(\mathbf{x}^*)\boldsymbol{\delta}\mathbf{x} + \ldots \tag{6}$$

Here we have defined the nonlinear system up to second order. Since the system is at a fixed point, the zero order term, $\mathbf{F}(\mathbf{x}^*)$, is equal to $\mathbf{0}$, giving

$$\mathbf{F}(\mathbf{x}^* + \boldsymbol{\delta}\mathbf{x}) = \mathbf{F}'(\mathbf{x}^*)\boldsymbol{\delta}\mathbf{x} + \frac{1}{2}\boldsymbol{\delta}\mathbf{x}\mathbf{F}''(\mathbf{x}^*)\boldsymbol{\delta}\mathbf{x} + \ldots \tag{7}$$

If we ensure that $\boldsymbol{\delta}\mathbf{x}$ is small, we can safely ignore second and higher order terms, yielding

$$\dot{(\mathbf{x}^* + \boldsymbol{\delta}\mathbf{x})} = \mathbf{F}'(\mathbf{x}^*)\boldsymbol{\delta}\mathbf{x} \tag{8}$$

$$\dot{\boldsymbol{\delta}}\mathbf{x} = \mathbf{F}'(\mathbf{x}^*)\boldsymbol{\delta}\mathbf{x} \tag{9}$$

and by simply renaming variables, $\mathbf{y} \equiv \boldsymbol{\delta}\mathbf{x}$ and $\mathbf{M} \equiv \mathbf{F}'(\mathbf{x}^*)$, we end up with the familiar linear form

$$\dot{\mathbf{y}} = \mathbf{M}\mathbf{y}. \tag{10}$$

Thus for small perturbations, $\boldsymbol{\delta}\mathbf{x}$, around a fixed point, $\mathbf{x}^*$, any nonlinear system behaves like a linear system. The fixed points act as a scaffolding for the nonlinear dynamics, allowing us, at least in simple cases, to decompose a hard nonlinear problem into smaller, linear sub-problems. This process is called linearization around a fixed point.

For the siRNN, the matrix $\mathbf{M}(\mathbf{x}^*)$ is obtained by computing $\mathbf{F}'(\mathbf{x}^*)$ for equation (1). Concretely, it is the derivative of $\mathrm{F}_i()$ with respect to x_j, i.e. $\frac{\partial \mathrm{F}_i}{\partial x_j}$, giving

$$M_{ij}(\mathbf{x}^*) = -\delta_{ij} + J_{ij}\ \mathrm{h}'(x_j^*), \tag{11}$$

where δ_{ij} is defined to be 1 if $i = j$ and otherwise 0 [*], and $\mathrm{h}'()$ is the derivative of the nonlinearity $\mathrm{h}()$ with respect to its input. Since this matrix derives from $\mathbf{F}(\mathbf{x})$, it is related to the feedback matrix, $\mathbf{J}$, but it is not $\mathbf{J}$. Instead, $\mathbf{M}$ defines the linear network that approximates the RNN around the point $\mathbf{x}^*$.

Going forward, our notation will drop the explicit dependency of $\mathbf{M}$ on $\mathbf{x}^*$, with the understanding that each locally linear system is still defined in terms of a particular fixed point.

[*] The notation δ_{ij} is the identity matrix written using indices and shouldn't be confused with $\boldsymbol{\delta}\mathbf{x}$.

Combining local linear systems to understand an RNN

As shown in Sussillo and Barak[5], the approach to understanding a trained RNN is to find as many fixed points and approximate fixed points of the system as possible. After finding these points, one linearizes the dynamics around them to understand the local dynamics. One then pieces all the linear solutions together to garner a semi-quantitative view of how the RNN functions. This means that there is always a local approximate linear system in consideration as well as the global, nonlinear system and one should keep these two systems separate conceptually. In what follows, we focus on a single, generic fixed point on a line attractor. Thus the arguments hold for all the fixed points on the line attractor. For the siRNN, this approach is adequate to explain how the system works.

An aside concerning approximate fixed points

Following Sussillo and Barak[5], linearization is appropriate around not only fixed points, but any sufficiently slow point, where a slow point is defined by a small nonzero value of the function $q(\mathbf{x}) = \frac{1}{2} |\mathbf{F}(\mathbf{x})|^2$. The function $q(\mathbf{x})$ defines the squared speed of the system divided by two. The understanding that one can treat slow points (with care) in the same way as true fixed points is important here since the line attractors in the siRNN are approximate, meaning they are lines of mostly slow points with only a few true fixed points. To simplify the explanations in what follows, we will ignore the distinction between a true fixed point and a slow point with the understanding that dynamics around slow points are qualitatively similar to those around true fixed points. Most importantly, the main assumption that linear dynamics are a good local approximation of the nonlinear dynamics still holds around slow points.

Linear systems

Linear dynamical systems can do four things: expand, contract, oscillate, and integrate an input. The last can technically only happen under perfect tuning. Normally, after an input is injected into the system, one thinks of very slow expansion or contraction as approximate integration.

The primary method one uses to understand what a linear system is doing is by diagonalizing the interaction matrix, $\mathbf{M}$, using an eigenvector decomposition. This decomposition is useful because it defines a basis in which certain patterns of activity, i.e. activity in special directions in state space, evolve separately from each other. A right eigenvector, $\mathbf{v}$, satisfies $\mathbf{M}\ \mathbf{v} = \lambda\ \mathbf{v}$, thus the matrix acts on these special vectors in a particularly straightforward way by scaling them by the amount, λ, called the eigenvalue. So the behavior of a linear dynamical system, $\dot{\mathbf{y}} = \mathbf{M}\ \mathbf{y}$, which involves the repeated application of $\mathbf{M}$, becomes easy to understand as, for example, the expansion (repeated scaling up) or contraction (repeated scaling down) of these vectors. The eigenvectors are a property of the matrix, and for a matrix defined by equation 11, the eigenvector decomposition is

$$\mathbf{M} = \mathbf{R}\;\mathbf{E}\;\mathbf{L} = \sum_{a}^{N} \lambda_a\, \mathbf{r}^a\, \mathbf{l}^a, \tag{12}$$

where λ_a is the a^{th} eigenvalue, $\mathbf{r}^a$ is the a^{th} right eigenvector (a column of $\mathbf{R}$) and $\mathbf{l}^a$ is the a^{th} left eigenvector (a row of $\mathbf{L}$). The matrix $\mathbf{R}$ is the matrix of right eigenvectors collected as columns, $\mathbf{L}$ is the matrix of left eigenvectors collected as rows with the property that $\mathbf{L} = \mathbf{R}^{-1}$. The matrix $\mathbf{E}$ is a diagonal matrix of eigenvalues.[†]

Looking forward, we are interested in the linearized dynamics around a fixed point in the full nonlinear system. To study those linear dynamics, we study $\mathbf{M}$, defined by equation (11), that derives from the original nonlinear system. The way to make sense of $\mathbf{M}$ is to use the eigenvector decomposition, defined by equation (12).

In the basis of the left eigenvectors, the local linear system is diagonalized, meaning the dynamics of the N modes evolve independently of each other. Diagonalizing the local network dynamics around a fixed point proceeds (again with $\mathbf{y} \equiv \mathbf{x} - \mathbf{x}^*$) as follows

$$\dot{\mathbf{y}} = \mathbf{M}\;\mathbf{y} \tag{13}$$

$$\dot{\mathbf{y}} = (\mathbf{R}\;\mathbf{E}\;\mathbf{L})\;\mathbf{y} \tag{14}$$

$$\mathbf{L}\;\dot{\mathbf{y}} = \mathbf{E}\;\;(\mathbf{L}\;\mathbf{y}) \tag{15}$$

$$\dot{\alpha_a} = \lambda_a \alpha_a, \tag{16}$$

where α_a is the a^{th} component of the vector $\boldsymbol{\alpha} \equiv (\mathbf{L}\;\mathbf{y})$. The independent modes, $\alpha_a(t)$, show how the different patterns, $\mathbf{r}^a$, evolve through time to create the overall population response.

Assuming all the eigenvalues are distinct, the linear dynamical system is trivially solved in this basis, giving

$$\alpha_a(t) = e^{\lambda_a t} \tag{17}$$

$$\alpha_a(t) = e^{\sigma_a t}\;\cos(\omega_a t), \tag{18}$$

where we have split the eigenvalue λ_a into its real and imaginary parts, $\lambda_a = \sigma_a + i\omega_a$, and ignored the constant of integration[‡]. Thus the eigenvalues explain whether or not a particular pattern, $\mathbf{r}^a$, expands - $\sigma_a > 0$, contracts - $\sigma_a < 0$, oscillates - $\omega_a \neq 0$, or integrates - $\sigma_a = 0, \omega_a = 0$.[§]

[†] A right eigenvector satisfies $\mathbf{M}\;\mathbf{r}^i = \lambda_i\;\mathbf{r}^i$ and a left eigenvector satisfies $\mathbf{l}^i\;\mathbf{M} = \lambda_i\;\mathbf{l}^i$.

[‡] The full solution for a complex root is $c_1 e^{\sigma_a t}\;\cos(\omega_a t) + c_2 e^{\sigma_a t}\;\sin(\omega_a t)$, for constant coefficients c_1 and c_2.

[§] Note that the fine-tuning of an integrator that is implemented as a line attractor is expressed in the requirement that both the real and imaginary parts of the integrating dimension must be exactly zero.

So now we know how the system will behave, based on examining the eigenvalues. However, our description of the dynamics is not in the basis of individual neuron activations. The final step is to put everything back in this basis. To get the modes of the system, we applied $\mathbf{L}$ to $\dot{\mathbf{y}}(t)$ to get $\dot{\boldsymbol{\alpha}}(t)$ and then integrated the modes separately. In order to get back the local network state, $\mathbf{y}(t)$, we apply the $\mathbf{R}$ matrix to $\boldsymbol{\alpha}(t)$, since $\mathbf{R} = \mathbf{L}^{-1}$. This gives

$$\mathbf{y}(t) = \mathbf{R}\ \boldsymbol{\alpha}(t). \tag{19}$$

Understanding selective integration

Due to the nature of the local linear systems on the color-context and motion-context line attractors, we can simplify the linearized dynamics far beyond the general eigenvector decomposition given in the last section. The eigenvalue spectra of the local linearized systems all have a common motif: there is a single eigenvalue that is very near to zero, which we call λ_0, and the rest of the eigenvalues have a large negative real part, $\sigma_a \equiv \mathrm{Re}(\lambda_a) \ll 0$, indicating that the respective modes will decay very quickly. A sensible indexing of the modes is to index a from 0 to N-1, since the zero mode turns out to be the only mode of interest.

Imagine the network was instructed to integrate an instantaneous pulse of color input (see cartoon in Fig. 6b in main text), and the system was on the color-context line attractor at a fixed point, $\mathbf{x}^*$. Then the color input pulse to the siRNN pushes the system off the line attractor in the direction of the color input vector. We want to know how much of the pulse of color is integrated, or stated graphically, how far along the color-context line attractor the system travels after a sufficient amount of time for the network to relax back to the line attractor, call it t_∞. Further, we want to understand how a pulse of irrelevant motion input is ignored while the relevant color pulse is integrated. How is it that the pulse of motion input does not move the system along the color-context line attractor?

In this scenario, each mode of the system will be affected by the color pulse according to the degree of projection of the input vector[¶] onto the associated left eigenvector. So the color pulse, $u_c(t)$, which comes into the system through weights, $\mathbf{b}^c$, has the projection onto a given left eigenvector, $\mathbf{l}^a$, of size $\mathrm{dot}(\mathbf{l}^a, \mathbf{b}^c u_c(t))$. We add this input to the differential equation for the independent modes, equation (16), and solve it. Assuming the network

[¶] We present the argument by using the color (or motion) input vector as a simplified proxy for the location of the system after a pulse of color (or motion) input arrives. To be strictly correct, we should instead use the network state after the pulse of input is finished to handle any nonlinear effects of a strong input. For our siRNNs, this mattered little so we continue using the color and motion input vectors.

state is initialized to the local origin, this gives the standard solution[||]

$$\alpha_a(t) = \mathrm{dot}\,(\mathbf{l}^a,\ \mathbf{b}^c)\, e^{\lambda_a t}. \tag{20}$$

For a single, instantaneous pulse of color input at time 0, all the modes with index $a > 0$ will be transiently perturbed according to the number, $\mathrm{dot}(\mathbf{l}^a, \mathbf{b}^c)$, and then quickly decay to zero. So in the long run, we have $\alpha_a(t_\infty) = 0$ for $a > 0$. Since the non-zero modes decay quickly, one can ignore their dynamics altogether. However, $\mathbf{l}^0$, the mode with (approximately) zero eigenvalue does not decay quickly. Instead, $\mathbf{l}^0$ (approximately) integrates the pulse by adding the value $\mathrm{dot}(\mathbf{l}^0, \mathbf{b}^c)$ to the previous value of the mode, which was zero since the system started at $\mathbf{x}^*$, the local origin. So for the current time interval, we have

$$\alpha_0(t) = \mathrm{dot}\left(\mathbf{l}^0,\ \mathbf{b}^c\right) e^{0t} \tag{21}$$

$$\alpha_0 = \mathrm{dot}\left(\mathbf{l}^0,\ \mathbf{b}^c\right). \tag{22}$$

Thus, the local linear dynamics around each fixed point can be approximated as 1-dimensional and static, having only a single mode that integrates the projection of the relevant input onto the selection vector, $\mathbf{l}^0$. We call $\mathbf{l}^0$ the *selection vector* because it is the projection of the input, either color or motion, onto this vector that determines whether or not that input will be integrated or dynamically deleted. We reemphasize that there is no decay along this vector. Thus if an input projects onto it, it will remain in the system. Clearly, the orientation of such a vector is a very powerful determinant in deciding what inputs are, or are not, relevant. If the projection of an input is large, then the amount of integration of that input will be large, if the projection is zero, the amount of integration of that input will be zero.

To understand the final state of the local system in response to a pulse of color input, we project back into the original local space by multiplying with $\mathbf{r}^0$, the local *line attractor*, giving

$$\mathbf{y}(t_\infty) = \mathbf{r}^0\, \mathrm{dot}\left(\mathbf{l}^0,\ \mathbf{b}^c\right). \tag{23}$$

In words, equation (22) states that the *amount* of integration is given by the projection of the input onto the selection vector, yielding a number. Equation (23) states that this amount is *represented* by the local system by advancing along the line attractor by exactly that amount.

Equation (23) determines the amount of integration of a pulse of color input and its representation in state space, while entirely ignoring the linear (or nonlinear) dynamics. Of course, the transient dynamic of the system from its deflection caused by the color pulse, back to the color-context line attractor, $\mathbf{r}^0$, results from the decay of activity on $\mathbf{r}^a$, for $\alpha > 0$.

[||] The full solution is $\alpha_a(t) = \alpha_a(0)\ e^{\lambda_a t} + \int_0^t e^{\lambda_a(t-t')}\ (\mathbf{l}^a\ \mathbf{b}^c)\ u_c(t')\ dt'$. We set the initial condition, $\alpha_a(0)$, to 0 since we are interested in a pulse from the current fixed point, $\mathbf{x}^*$, which is the origin of the current local linear system. The input pulse is treated as a Dirac delta function at time 0, yielding equation 20.

Note that the long-time solution for a generic color input is

$$\mathbf{y}(t_\infty) = \mathbf{r}^0 \int_0^{t_\infty} \mathrm{dot}\left(\mathbf{l}^0, \mathbf{b}^c\right)\ u_c(t)\ dt, \tag{24}$$

which makes clear that the selection vector determines the integrand and the line attractor determines the direction of the integral in state space.

Finally, to get back the absolute position in state space, we "leave" the local linear system by adding back the local origin, $\mathbf{x}^*$. The new absolute position on the global color-context line attractor is $\mathbf{x}(t_\infty) = \mathbf{x}^* + \mathbf{y}(t_\infty)$ [**]. This results in a new position on the global color-context line attractor. The change in the local state given by equation (23) represents a good approximation of the total change in state of the full nonlinear siRNN resulting from the integration of a single pulse of color information.

Note that if the matrix of linearized dynamics around the fixed point were normal (e.g. a symmetric matrix, with $\mathbf{M} = \mathbf{M}^\mathrm{T}$, is normal), then both $\mathbf{R}$ and $\mathbf{L}$ would be composed of orthonormal vectors and $\mathbf{L} = \mathbf{R}^\mathrm{T}$. Equation 24 would instead be

$$\mathbf{y}(t_\infty) = \mathbf{r}^0 \int_0^{t_\infty} \mathrm{dot}\left(\mathbf{r}^0, \mathbf{b}^c\right)\ u_c(t)\ dt. \tag{25}$$

Equation 25 corresponds to the familiar notion that in order to integrate an input, (i) the input should project onto the line attractor, $\mathbf{r}^0$; (ii) that the amount of integration corresponds to the size of the projection onto the line, $\mathrm{dot}(\mathbf{r}^0, \mathbf{b}^c)$; and (iii) that the representation in state space of this integrated input is a deflection along the line, $\mathbf{r}^0\,\mathrm{dot}(\mathbf{r}^0, \mathbf{b}^c)$. While intuitive, this is not true for the linear systems in the siRNN, because equation (11) does not in general generate normal matrices.

In the general case, which is applicable to the siRNN, the only requirements on the pair $(\mathbf{r}^0, \mathbf{l}^0)$ is that they are not orthogonal and their dot product equals 1. This leads to an additional degree of freedom that the network has regarding which direction in state space it chooses to integrate. The network architecture may be configured such that $\mathbf{l}^0$ can point in any direction, so long as its not orthogonal to $\mathbf{r}^0$[††]. Any input in the direction of $\mathbf{l}^0$ will be integrated on $\mathbf{r}^0$. Selective integration as implemented here is as simple as making sure that $\mathbf{l}^0$ is pointed towards the input to be integrated and orthogonal to the input to be ignored. The counterintuitive part is that the neural activations reflect this integration using a different vector, $\mathbf{r}^0$. See Fig. 6b in the main text for an illustration of the interaction of the left and right eigenvectors to achieve selective integration.

Putting it all together, for the global line attractor defined by the color context, the $\mathbf{l}^0$ vectors of the local linear systems are pointed towards the color input vector and are roughly

[**] This statement is true only if the linear approximation is a perfect description of the global dynamics. Otherwise, there will be some small error.

[††] While arranging that $\mathbf{l}^0$ be nearly orthogonal to $\mathbf{r}^0$ (e.g. 89.9°) is possible, it is a very poor choice. The requirement that $\mathrm{dot}(\mathbf{l}^0, \mathbf{r}^0) = 1$ would dictate that the norm of $\mathbf{l}^0$ be gigantic, leading to many problems, such as integrating noise in other dimensions.

orthogonal to the motion input vector. This simultaneously explains both the integration of color and the dynamic deletion of the irrelevant motion input. The correct amount of color input is projected onto a mode with no decay, and motion input is projected exclusively into directions of fast decay. In the motion context, the global motion-context line attractor is active. In this case the $\mathbf{l}^0$ vectors associated with the motion line attractor are pointed towards the motion input vector and are approximately orthogonal to the color input vector. In either context, due to the flexibility of the local color and motion $\mathbf{l}^0$ vectors, the two global line attractors need not be precisely aligned to their respective relevant input vectors (see Fig. 6c in main text).

Finally, we address whether treating the nonlinear RNN as a set of linear systems around fixed points is sufficient for explaining its integration and gating mechanisms. We examine this question quantitatively by computing the relative sizes of the zero-order, $|\mathbf{F}(\mathbf{x}^*)|$, first-order, $|\mathbf{F}'(\mathbf{x}^*)\delta\mathbf{x}|$, and second-order, $|\frac{1}{2}\delta\mathbf{x}\mathbf{F}''(\mathbf{x}^*)\delta\mathbf{x}|$, parts of the Taylor series expansion around all the fixed points on the color-context line attractor. For the the color pulse (magnitude of 2.0) in the color context, averaged across all fixed points, the values of the Taylor series terms are (mean ± std) 0.006 ± 0.004, 0.575 ± 0.002, 0.009 ± 0.001, respectively. For a motion pulse in the color context, the values are 0.006 ± 0.004, 0.660 ± 0.004, 0.013 ± 0.001, respectively. So for both motion and color pulses, which result in a deflection in state space of the same order of magnitude as the noisy input during normal operation, the linear part of the Taylor expansion is far larger than either the zero-order or second-order terms. This demonstrates that our linear systems approach is sufficient to explain the operation of the RNN. Analogous results hold for the motion context.

In summary, analysis of the siRNN suggests that a simplified (but still useful) view of how RNNs work is that of a state space tiled with linear systems. These linear systems are responsible for locally linear dynamic computations and have large volumes where the linearity assumption is valid. The nonlinearity of the system is then activated by inputs or internal dynamics that drive the system from one linear system to another.

References for the mathematical explanation

1. Wilson, H. & Cowan, J. Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons. *Biophysical Journal* **12**, 124 (1972).
2. Wang, X. J. Decision making in recurrent neuronal circuits. *Neuron* **60**, 215234 (2008).
3. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci USA* **79**, 25542558 (1982).
4. Martens, J. & Sutskever, I. Learning recurrent neural networks with hessian-free optimization. Proceedings of the 28th International Conference on Machine Learning *(ICML)* (2011).
5. Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation* **25**, 626649 (2013).
6. Seung, H. S. How the brain keeps the eyes still. *Proc Natl Acad Sci USA* **93**, 1333913344 (1996).