Complementary roles of dimensionality and population structure in neural computations

Alexis Dubreuil[1], Adrian Valente[1], Manuel Beiran[1], Francesca Mastrogiuseppe[2], and Srdjan Ostojic[1]

[1]Laboratoire de Neurosciences Cognitives et Computationnelles, INSERM U960, Ecole Normale Superieure - PSL Research University, 75005 Paris, France
[2]Gatsby Computational Neuroscience Unit, UCL, London, Great Britain

July 3, 2020

Abstract

Neural computations are currently investigated using two competing approaches: sorting neurons into functional classes, or examining the low-dimensional dynamics of collective activity. Whether and how these two aspects interact to shape computations is currently unclear. Using a novel approach to extract computational mechanisms from networks trained with machine-learning tools on neuroscience tasks, here we show that the dimensionality of the dynamics and cell-class structure play fundamentally complementary roles. While various tasks can be implemented by increasing the dimensionality in networks consisting of a single global population, flexible input-output mappings instead required networks to be organized into several sub-populations. Our analyses revealed that the subpopulation structure enabled flexible computations through a mechanism based on gain-controlled modulations that flexibly shape the dynamical landscape of collective dynamics. Our results lead to task-specific predictions for the structure of neural selectivity and inactivation experiments.

1 Introduction

The quest to understand neural computations in the cortex currently relies on two competing paradigms. Classical works have sought to determine the computational role of individual cells by sorting them into functional classes based on their responses to sensory and behavioral variables [Hubel and Wiesel, 1959; Moser et al., 2017; Hardcastle et al., 2017]. Fast developing tools for dissecting neural circuits have opened the possibility of mapping such functional classes onto genetic and anatomic cell types, and given a new momentum to this cell-category approach [Adesnik et al., 2012; Ye et al., 2016; Kvitsiani et al., 2013; Hangya et al., 2014; Pinto and Dan, 2015; Hirokawa et al., 2019]. This viewpoint has however been challenged by observations that individual neurons often represent seemingly random mixtures of sensory and behavioral variables, especially in higher cortical areas [Churchland and Shenoy, 2007; Machens et al., 2010; Rigotti et al., 2013; Mante et al., 2013], where clear functional cell classes are often not clearly apparent [Raposo et al., 2014] (but see [Hirokawa et al., 2019]). A newly emerging paradigm has therefore posited that neural computations need instead to be understood in terms of collective dynamics in the state space of joint activity of all neurons [Buonomano and Maass, 2009; Rigotti et al., 2013; Mante et al., 2013; Gallego et al., 2017; Remington et al., 2018; Saxena and Cunningham, 2019]. Within this viewpoint, neural computations are revealed by studying properties of low-dimensional trajectories of activity in state space [Mante et al., 2013; Rajan et al., 2016; Chaisangmongkon et al., 2017; Remington et al., 2018; Wang et al., 2018; Sohn et al., 2019], while the selectivity of the individual neurons is often largely uninformative. Whether and how the two paradigms based on functional cell categories and collective dynamics can be reconciled is an open question.

A key hypothesis emerging from the collective dynamics paradigm states that the dimensionality of neural trajectories determines the complexity of the computations a network can implement [Legenstein and Maass, 2007; Buonomano and Maass, 2009; Rigotti et al., 2013; Fusi et al., 2016]. In contrast, a candidate computational role for functional cell classes within the collective dynamics framework is currently lacking. Can any task be implemented by increasing the dimensionality of the dynamics in a single population with random selectivity, or are functionally distinct sub-populations necessary for specific computations [Yang et al., 2019]? To address this fundamental computational question, we developed a new class of interpretable recurrent networks, which fully disentangle the concepts of cell populations and dimensionality of the collective dynamics. We then exploited this framework to identify the respective roles of dimensionality and sub-populations in recurrent neural networks trained on a range of systems neuroscience tasks using machine-learning [Sussillo, 2014; Barak, 2017; Yang et al., 2019]. Specifically, we first trained networks with minimal intrinsic dimensionality for each task, then determined whether several sub-populations are needed to perform the task. This approach allowed us to extract computational mechanisms from the trained networks, by reducing them to simpler interpretable networks consisting of minimal intrinsic dimension and number of sub-populations. These simplified networks performed the tasks with identical accuracy and identical collective dynamics as the original networks, but revealed the key mechanisms underlying the computations.

Altogether, our analyses demonstrate that the intrinsic dimension and sub-population structure play fundamentally different and complementary computational roles in recurrent networks. As expected from previous studies [Remington et al., 2018; Mastrogiuseppe and Ostojic, 2018], the intrinsic dimension determines the number of internal collective variables available for the network to perform a computation. The sub-population structure in contrast shapes the possible dynamics of these collective variables. While a range of tasks could be implemented by increasing the intrinsic dimension of a network consisting of a single global population with random connectivity, we found that specific tasks required the network to be organized into several statistical sub-populations. This was specifically the case for tasks requiring a flexible reconfiguration of input-output associations, a common component of many cognitive tasks [Sakai, 2008]. We show that a subpopulation structure of the network enables such flexible computations through a mechanism based on modulations of gain and effective interactions that flexibly modify the dynamical landscape of collective dynamics. Specifically, the sub-population structure allows different inputs to act either as drivers or modulators of the collective dynamics. Our results lead to direct predictions on when and where statistical structure should be present in single-neuron selectivity, as well as to specific predictions for inactivations of different sub-populations.

2 Results

2.1 Computational framework

To identify the respective roles of dimensionality and cell populations, we extended a recently introduced class of network models, low-rank recurrent neural networks [Mastrogiuseppe and Ostojic, 2018]. In line with dimensionality reduction approaches [Cunningham and Yu, 2014; Gallego et al., 2017], in this framework, the network connectivity is represented in terms of patterns over neurons (Fig. 1a, Methods section 4.1). Each feed-forward input to the network is specified by a pattern $\boldsymbol{I}^{(l)}$, and the output of the network is read out linearly through a pattern $\boldsymbol{w}$. The recurrent connectivity matrix $\boldsymbol{J}$ is of rank R, so that it is specified in terms of R modes:

$$\boldsymbol{J} = \boldsymbol{m}^{(1)}\boldsymbol{n}^{(1)T} + \ldots + \boldsymbol{m}^{(R)}\boldsymbol{n}^{(R)T}. \tag{1}$$

Each mode r consists of an output pattern $\boldsymbol{m}^{(r)}$ that determines a principal direction of activity in state space, and an input-selection pattern $\boldsymbol{n}^{(r)}$ that determines which input patterns activate the corresponding mode ([Mastrogiuseppe and Ostojic, 2018], Methods section 4.2). The population-level activity in the network can then be directly described in terms of set of internal and external collective variables κ_r and u_l, analogous to latent variables usually identified by dimensionality reduction:

$$\boldsymbol{x}(t) = \sum_{r=1}^{R} \kappa_r(t)\boldsymbol{m}^{(r)} + \sum_{l=1}^{N_{in}} u_l(t)\boldsymbol{I}^{(l)}. \tag{2}$$

The two sets of collective variables correspond to recurrent and input-driven directions in state-space [Wang et al., 2018]. One internal collective variable κ_r is associated with each connectivity mode r, and one external collective variable u_l is associated with each feed-forward input pattern, so that the dynamics is embedded in a linear subspace of dimension given by the sum of the dimensionality of feed-forward inputs and the rank R of the connectivity (Methods section 4.2). A mean-field analysis of low-rank networks provides a reduced description of the low-dimensional dynamics, in which the set of internal variables $\{\kappa_k\}$ forms a dynamical system, with effective inputs and interactions determined by the statistics of feed-forward and connectivity patterns (Fig. 1d, Methods section 4.5).

Within this framework, each neuron is characterized by a set of *loadings* that correspond to its values on each of the input, readout and connectivity patterns. Each neuron can therefore be represented as a point in *loading space*, where each axis is associated with one pattern (Methods section 4.4). For instance, if the network consists of one input pattern, one readout pattern and two connectivity modes, each neuron has six loadings, and the loading space is six-dimensional (Fig. 1b). The full network can then be represented as a set of points in the space of loadings, one point for each neuron, and within mean-field theory the statistics of this cloud of points fully determine the collective dynamics and computations performed by the network (Methods section 4.5). If the network size is large, any network generated by randomly sampling all neurons from a given statistical distribution of loadings exhibits identical collective dynamics, and therefore identical computations.

Previous work on the low-rank framework [Mastrogiuseppe and Ostojic, 2018; Schuessler et al., 2020a] considered the situation where all neurons belonged to a single cluster in the loading space. Specifically, all neurons belonged to a single Gaussian population fully specified by a matrix of covariances between input and connectivity patterns. In biological networks, neurons instead belong to several sub-populations with, in particular, distinct relations between input and connectivity [Harris and Mrsic-Flogel, 2013]. We therefore extended the low-rank framework to include P populations of neurons that corresponded to P Gaussian clusters in the space of loadings of connectivity, input and readout patterns (Fig. 1b). Each cluster was centered at the origin, but had its own matrix of covariances between patterns (Fig. 1c). Within this extended framework, the number R of connectivity modes, and the number P of populations are two independent parameters that play distinct roles: R determines the number of available internal variables, while the number of populations shapes the dynamics of these variables (Methods section 4.5).

Our goal was to understand when several distinct sub-populations of neurons are needed from a computational perspective, and what role such diversity plays in computations. To this end, we first used machine-learning tools to train low-rank RNNs on a set of systems neuroscience tasks. For each task, we specifically sought networks of minimal rank R, and identified corresponding sets of patterns that implemented the task

Behavioral task	Minimal rank	Minimal # of populations
Perceptual DM	R=1	P=1
Multi-sensory DM	R=1	P=1
Parametric WM	R=2	P=1
Context-dependent DM	R=1	P=2
Delay-Match-to-Sample (two items {A,B})	R=2	P=2

Table 1. Minimal rank and number of populations required to implement individual cognitive tasks.

(Methods section 4.6). We then performed clustering analyses on pattern loadings, and determined the covariance structure corresponding to each cluster. Specifically, we progressively increased the number of fitted clusters, and determined the minimal number of populations needed to implement the task (Table 1) by randomly sampling connectivity from the corresponding distribution of connectivity and inspecting the performance of the obtained networks (Methods section 4.7). We finally combined these analyses with mean-field theory to identify the key parameters and build reduced, minimal models of the networks in terms of collective variables (Fig. 1d) that allowed us to directly identify and interpret the dynamical mechanisms underlying the implemented computations.

2.2 Increasing dimensionality allows networks to implement increasingly complex tasks with a single population

As expected from previous work [Buonomano and Maass, 2009; Rigotti et al., 2013], we found that tasks of increasing complexity could be implemented with networks consisting of a single population by increasing the dimensionality of the internally generated activity, and therefore the number of internal variables available for computations.

We started with one of the most classical system neuroscience tasks, perceptual decision making [Gold and Shadlen, 2007]. A network received a noisy scalar stimulus along a random input pattern, and was trained to report the sign of its temporal average along a random readout pattern (Fig. 2a). A unit-rank network, consisting of a single connectivity mode was sufficient to solve the task (Supplementary Fig. S6). As expected from the theory of low-rank networks, the dynamics evolved in a two-dimensional plane spanned by the input pattern $\boldsymbol{I}$ and the output connectivity pattern $\boldsymbol{m}$, and could be described by two corresponding collective variables $u(t)$ and $\kappa(t)$ (Fig. 2e). The internal collective variable $\kappa(t)$ encoded the integrated stimulus (Fig. 2d), and therefore could be directly interpreted in terms of the computation performed by the network. The output connectivity pattern $\boldsymbol{m}$ was aligned with the readout pattern $\boldsymbol{w}$, so that the network output was directly set by κ.

As the network was specified by four patterns (the input, readout and the two connectivity patterns), the loading space was four-dimensional (Fig. 2b). Fitting a single cluster to the four dimensional distribution of loadings revealed that a single, global Gaussian population was sufficient to implement this task. Indeed, new networks generated randomly by resampling the connectivity from the fitted single-cluster covariance structure led to task accuracies indistinguishable from trained networks (Fig. 2c). We next performed a mean-field analysis of the obtained connectivity to identify the parameters in the pattern covariance structure that determined the computation. We found three key parameters: the covariance between the input pattern $\boldsymbol{I}$ and the input-selective connectivity pattern $\boldsymbol{n}$ determined the strength of the inputs integrated by the internal variable, the covariance between the readout pattern $\boldsymbol{w}$ and the output connectivity pattern $\boldsymbol{m}$ determined the strength with which the internal variable drove the readout, while the covariance between the two connectivity patterns $\boldsymbol{m}$ and $\boldsymbol{n}$ determined positive feedback on the internal variable and the integration timescale (Supplementary Fig. S1c,d). Such reduced models built by controlling only the three key parameters performed the task with an accuracy comparable to trained networks (Fig. 2c) and reproduced identical low-dimensional dynamics (Supplementary Fig. S1b).

The findings from the perceptual decision task directly extended to a multi-sensory decision-making task [Raposo et al., 2014], in which the network received two stimuli along orthogonal input patterns, and was trained to process both of them to produce the output. A unit-rank network consisting of a single population

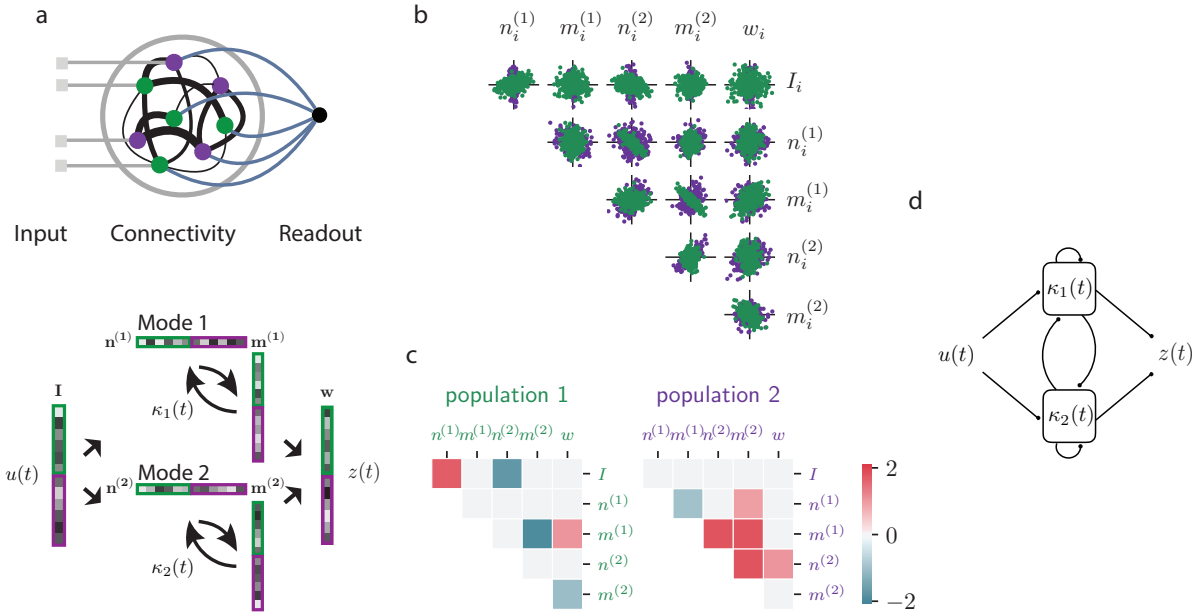

Figure 1: Multi-population, low-rank recurrent neural networks. (a) The recurrent connectivity consists of R modes ($R = 2$ in this illustration), each represented by an input-selection pattern $\boldsymbol{n}^{(r)}$ and and output pattern $\boldsymbol{m}^{(r)}$. The activity along each mode is represented by an internal collective variable κ_r. Inputs and readouts are similarly represented in terms of patterns over neurons. Each neuron in the network belongs to one of P subpopulations, each defined by different statistics of the corresponding input and connectivity sub-patterns. In the illustration $P = 2$, each population and the corresponding sub-pattern are represented by a different color (green, violet). (b) Two-dimensional projections of statistics in pattern loading space. Each neuron is characterised by its set of input, readout and connectivity pattern values, which we refer to as loadings. Each neuron therefore corresponds to a point in *loading space*, which is six-dimensional in this example. The two sub-populations form two distinct clusters of points in loading space. (c) Each population is summarized by a set of covariances between patterns, that specify the shape and orientation of the corresponding cluster in the pattern loading space. In this study, we focus on zero-mean clusters, which are all centered at the origin. (d) The dynamics in the network can be described by an effective circuit model consisting of interacting collective variables κ_r, driven by the input (see Eq. (3)). The interactions between the collective variables and the input are set by a combination of pattern covariances of the two populations shown in (c).

was sufficient to integrate several orthogonal, but congruent inputs (Supplementary Fig. S8).

We next turned to a parametric working memory task [Romo et al., 1999], where two scalar stimuli f_1 and f_2 were successively presented along an identical input pattern, interleaved by a variable delay period. The network was trained to report the difference $f_1 - f_2$ between the values of the two stimuli (Fig. 3a). We found that this task required $R = 2$ pairs of connectivity patterns (Supplementary Fig. S6), so that the dynamics were now three-dimensional and described by two internal collective variables. One internal variable integrated the first stimulus and memorized it during the delay period (Fig. 3d top), while the second one encoded stimuli transiently (Fig. 3d bottom). The final readout was obtained by combining linearly the two output directions to compute the difference between the two internal variables after the second stimulus was presented (Fig. 3a bottom).

The network was now specified by six patterns (the input and readout patterns and two pairs of connectivity patterns), so that the loading space was six-dimensional. However, again a single population was sufficient to implement the task, as fitting a single cluster to the loadings of the trained connectivity, input and readout patterns (Fig. 3b), and generating new networks by sampling from the fitted covariance structure led to networks with indistinguishable performance (Fig. 3c). The mean-field analysis allowed us to reduce the network to a simple circuit of two non-interacting collective variables (Fig. 3e), and to identify as key parameters the covariances between the input-selection and output patterns within each mode. Specifically, a large covariance between the first pair of connectivity patterns led to strong positive feedback and persistent activity in the first internal variable, while the covariance was much weaker for the second pair of patterns that encoded stimulus value transiently. The resulting reduced model performed the task with accuracy indistinguishable from trained networks (Fig. 3c), and reproduced the collective dynamics (Supplementary Fig. S2b).

2.3 Flexible tasks require multiple populations

While a variety of tasks could be implemented by increasing the dimensionality in networks consisting of a single neural population, this was not the case for all tasks we considered. In particular, several tasks required processing the same input differently in various epochs. When such flexibility was needed, we found that increasing the number of neural populations was crucial.

A first example of such a task was context-dependent decision making, where stimuli consisted of a combination of two scalar features that fluctuated in time [Mante et al., 2013]. Depending on a contextual cue, only one of the two features needed to be integrated (Fig. 4a), so that the same stimulus could require opposite responses, a hallmark of flexible input-output transformations [Fusi et al., 2016]. We implemented each stimulus feature and contextual cue as an independent input pattern over the population, so that the total input dimensionality was four. Training networks to perform this task, we found that unit-rank connectivity, consisting of a single connectivity mode and therefore a single internal variable, was sufficient (Fig. S6). As for standard decision-making, the internal variable encoded integrated evidence. However, our clustering analysis in the loading space, which was now seven-dimensional (four input, one readout and two connectivity patterns), revealed that several sub-populations were necessary to perform the computation (Fig. 4b and Supplementary Fig. S7). Indeed, generating networks from a single fitted population led to a strong degradation of the performance (Fig. 4c left). Specifically, single-population networks performed the task correctly for stimuli consisting of congruent features, but failed for incongruent stimuli for which responses needed to be flipped according to context (Fig. 4c right). This was the case even if the internal dimensionality of the networks was increased while constraining the neurons to belong to a single cluster (Supplementary Fig. S7). Instead, when we generated networks randomly by sampling from two fitted clusters with preserved covariance structure, we found they performed the task with an accuracy indistinguishable from the trained network (Fig. 4c), indicating that two sub-populations were sufficient to implement the computation.

As a second example of a task requiring flexible stimulus processing, we considered the delay-match-to-sample task [Miyashita, 1988; Engel and Wang, 2011; Chaisangmongkon et al., 2017], where two stimuli were presented interleaved by a delay period, and the network was trained to indicate in each trial whether the two stimuli were identical or different (Fig. 5a). This task involved flexible stimulus processing analogous to the context-dependent decision-making task because an identical stimulus presented in the second position required opposite responses depending on the stimulus presented in the first position (Fig. 5a,c). We found that this task required a rank two connectivity (Fig. S6), and therefore two internal variables. As in the parametric working-memory task (Fig. 3d), one internal variable maintained the first stimulus in memory during the delay period

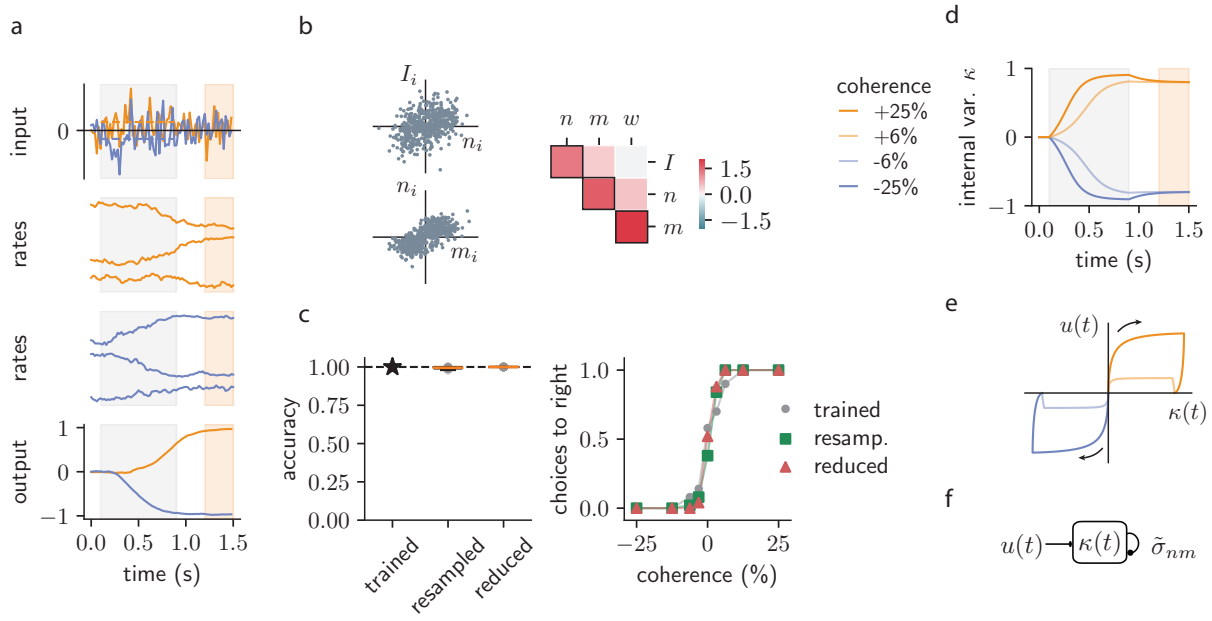

Figure 2: Perceptual decision making task. (a) A unit rank network was trained to integrate a fluctuating scalar input and report whether its average, which we denote as coherence, was positive or negative. Top to bottom: input, activity of 3 random units and output, shown for a positive coherence (orange) and a negative coherence (blue) trials. Grey and orange outlines indicate respectively the stimulus presentation and output epochs. (b) Statistics of the pattern loadings. The trained network consisted of an input pattern $\boldsymbol{I}$, connectivity patterns $\boldsymbol{m}$ and $\boldsymbol{n}$ and a readout pattern $\boldsymbol{w}$ so that the loading space was four-dimensional. Left: selected two-dimensional projections of the loading space. Each point represents the entries of a neuron on the corresponding pattern. Right: covariances between input, connectivity and readout patterns. (c) Task accuracy and psychometric curves of trained, resampled and minimal reduced networks. Ten resampled networks were generated by sampling loadings from a multivariate Gaussian distribution with a covariance matrix fitted to the trained network. The reduced model was generated by adjusting only the covariances outlined in black in (b), and corresponds to the effective circuit shown in (f). (d) Dynamics of the collective internal variable on trials with positive (orange) and negative (blue) evidence. (e) Trajectories of activity in the two-dimensional plane corresponding to the internal collective variable $\kappa(t)$ and the external collective variable $u(t)$. This subspace corresponds to the projection of activity on the output pattern $\boldsymbol{m}$ and the input pattern $\boldsymbol{I}$. (f) Circuit diagram representing the reduced model.

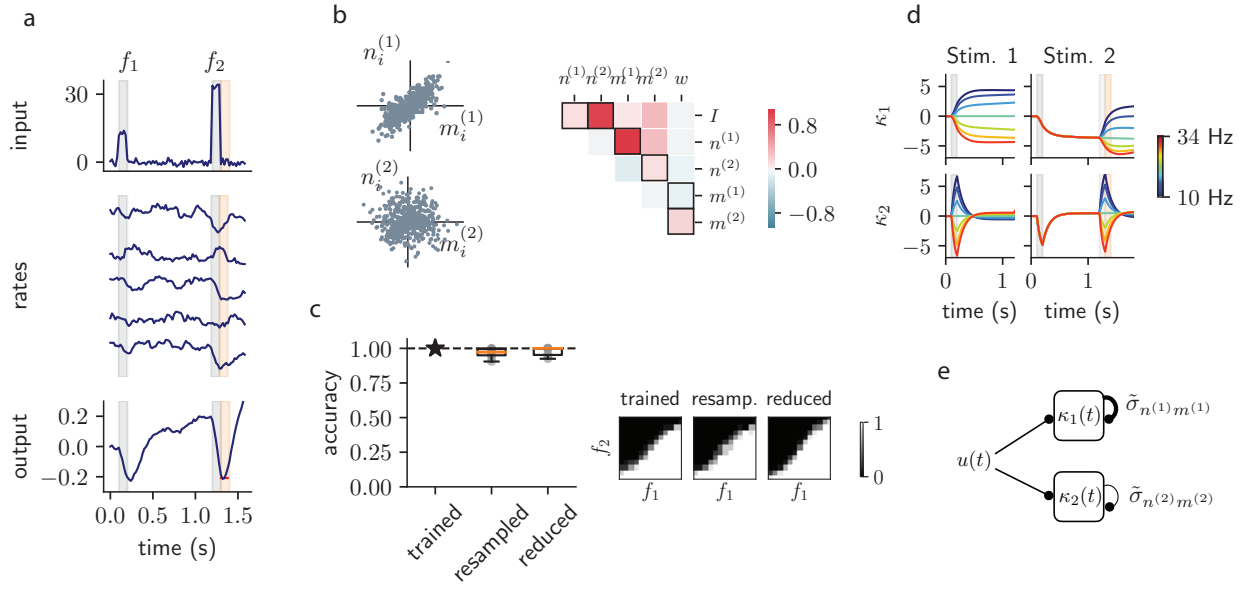

Figure 3: Parametric working memory task. (a) A rank two network received two stimuli f_1 and f_2 interleaved by a delay, along the same input pattern, and was trained to output their difference. Top to bottom: input, activity of 5 random units and output, shown for an example trial; the target value is indicated in red in the bottom panel. Grey and orange outlines indicate respectively the stimulus presentation and output epochs. (b) Statistics of the trained pattern loadings. Left: selected two-dimensional projections of the six-dimensional loading space. Right: covariances of the input, connectivity, and readout patterns. (c) Task accuracy and psychometric responses of trained, resampled and minimal, reduced networks. Left: same as figure 2c. Right: psychometric response matrices showing the proportion of positive responses in the $f1 - f2$ plane. (d) Low-dimensional dynamics of internal collective variables. Left: responses to the first stimulus (colors represent different values of f_1). Right: responses throughout the whole trial to a range of values for the second stimulation (f_1 fixed at 30Hz, colors represent different values of f_2). (e) Circuit diagram representing the reduced model.

(Fig. 5d), while the second internal variable implemented the comparison after the second stimulus (Fig. 5d). Similarly to the context-dependent decision making task, more than one population was needed to implement the task, as networks generated by resampling from a single population scrambled the performance (Fig. 5c). Fitting instead two clusters in the seven-dimensional loading space (two input, one readout and four connectivity patterns) showed that two sub-populations were sufficient (Fig. 5b), as networks generated by sampling from the fitted two-population distribution performed the task with full accuracy (Fig. 5c).

Altogether, training and randomly regenerating networks revealed that several populations were required for flexible input-output tasks. The precise role of the various populations was however not immediately clear from the low-dimensional dynamics in the trained networks.

2.4 Mechanism: reconfiguration of network dynamics by gain modulation

To unveil the mechanisms underlying flexible input-output mappings in networks with several sub-populations, we performed a mean-field analysis of the dynamics. Here we first lay out the general principles revealed by the analysis, and then apply them specifically to networks trained on the two flexible tasks described above.

For concreteness, we consider a network with $R = 2$ connectivity modes, and two input patterns $\boldsymbol{I}^A$ and $\boldsymbol{I}^B$ driven by scalar inputs $u_A(t)$ and $u_B(t)$. Such a network is described by two internal collective variables κ_1 and κ_2, and our mean-field analysis showed that the dynamics of these variables is equivalent to a simple two-unit circuit:

$$\begin{aligned}\tau\frac{d\kappa_1}{dt} &= -\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(2)}}\kappa_2 + \tilde{\sigma}_{n^{(1)}I^A}u_A(t) + \tilde{\sigma}_{n^{(1)}I^B}u_B(t)\\ \tau\frac{d\kappa_2}{dt} &= -\kappa_2 + \tilde{\sigma}_{n^{(2)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(2)}m^{(2)}}\kappa_2 + \tilde{\sigma}_{n^{(2)}I^A}u_A(t) + \tilde{\sigma}_{n^{(2)}I^B}u_B(t).\end{aligned} \tag{3}$$

The internal variables and the inputs are coupled through effective couplings $\tilde{\sigma}_{ab}$, which depend both on the hardwired input and connectivity patterns, and implicitly on the collective variables themselves, so that the dynamics of internal variables is non-linear.

For networks consisting of a single Gaussian population, the effective couplings are simply given by $\tilde{\sigma}_{ab} = \langle\Phi'\rangle\sigma_{ab}$, where σ_{ab} is the covariance between the corresponding input-selection pattern ($a = \boldsymbol{n}^{(1)}, \boldsymbol{n}^{(2)}$) and output or input patterns ($b = \boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \boldsymbol{I}^A, \boldsymbol{I}^B$), while $\langle\Phi'\rangle$ is the average gain of all the neurons, that depends implicitly both on internal variables and external inputs (see Methods section 4.5). Importantly, only input patterns having a non-zero covariance with the input-selection patterns $\boldsymbol{n}^{(1)}$ and $\boldsymbol{n}^{(2)}$ directly drive the internal variables. In contrast, inputs orthogonal to input-selection patterns do not directly drive the dynamics of internal variables, but modulate the value of the gain $\langle\Phi'\rangle$. These two types of inputs can therefore in principle play the roles of drivers and modulators [Sherman and Guillery, 1998]. Crucially however, in networks consisting of a single population, all the effective couplings are scaled by the same gain factor, which strongly limits the range of possible dynamics for the internal variables [Beiran et al., 2020], and the possible roles of modulatory inputs.

We next turn to a network in which neurons belong to P distinct sub-populations. Each connectivity or input pattern is now split into P sub-patterns of size $\alpha_p N$, one for each sub-population p, and each of the sub-populations is specified by its own set of overlaps $\sigma^{(p)}_{ab}$ between sub-patterns. A key difference with single-population networks is that now each sub-population has its own gain factor $\langle\Phi'\rangle_p$ that can be modulated independently by inputs, or internal dynamics. The collective dynamics is described by the same dynamical system as in Eq. (3), but the effective couplings are now weighted averages of connectivity overlaps for different populations:

$$\tilde{\sigma}_{ab} = \sum_{p=1}^{P} \alpha_p \langle\Phi'\rangle_p \sigma^{(p)}_{ab}. \tag{4}$$

As the gain of each sub-population can be modified independently by modulatory inputs, increasing the number of populations considerably extends the range of possible dynamics, and in fact allows a rank two network to implement in principle any two-dimensional dynamical system [Beiran et al., 2020]. In particular, modulating the gains in different trials or epochs of a task allows the sub-populations to flexibly remodel effective couplings to shape the collective dynamics, and therefore the performed computation. We next describe how this general mechanism explains the computations in the two flexible tasks of Fig. 4 and 5.

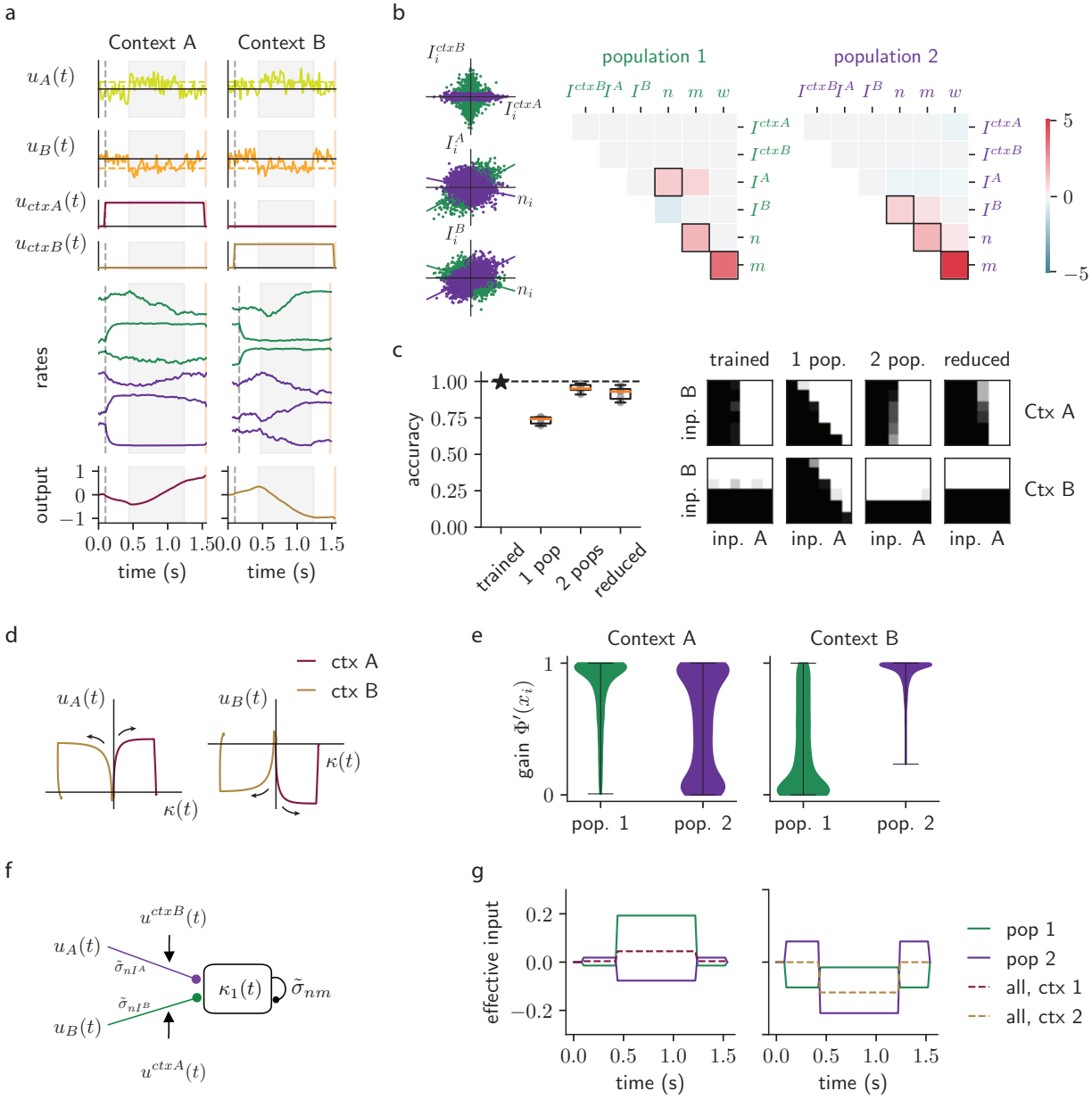

Figure 4: Context-dependent decision-making task. The input to the network consisted of a stimulus composed of two fluctuating scalar features along two different input patterns, as well as two constant contextual cues along two additional input patterns. The contextual cues indicated in each trial the identity of the stimulus feature to be integrated. The network was trained to report the sign of the average of the relevant feature. (a) Illustration of network dynamics in two trials with an identical stimulus but different contextual cues, leading to opposite responses (left and right columns). Top-bottom: contextual cues and stimulus inputs; activity of 6 random neurons, 3 from each identified population; output generated by the readout. Grey and orange outlines indicate respectively the stimulus presentation and output epochs. (b) Statistics of the trained pattern loadings, determined by a clustering analysis in the seven-dimensional loading space. Left: three two-dimensional projections of the loading space, showing the presence of two clusters that defined two different sub-populations. The regression lines indicate covariances between loadings for each cluster. Right: empirical covariances between patterns for each of the two populations.

Figure 4 *(previous page)*: (c) Accuracy and response matrix of trained, resampled and minimal reduced networks. Ten resampled networks were generated by sampling loadings from either a single-population ($p = 1$) or a two-population ($p = 2$) Gaussian distribution with covariance matrices fitted to the trained network (error bars display standard deviations). The reduced model was generated by adjusting only the covariances outlined in black in (b), and corresponds to the effective circuit shown in (f). (d) Trajectories of activity in response to the 2 trials of panel a in the two-dimensional subspaces corresponding to the internal collective variable κ and the external collective variable u_A (left) or u_B (right). These sub-spaces are spanned respectively by vectors $\boldsymbol{m}$-$\boldsymbol{I}^A$, and $\boldsymbol{m}$-$\boldsymbol{I}^B$. (e) Distribution of single neuron gains across the two populations, in the two contexts. (f) Circuit diagram representing the reduced model. (g) Contribution of the two populations to the effective input to the internal variable in the two contexts, defined as $\frac{1}{N}\sum_i n_i \phi(\sum_k I_i^{(k)})$.

For the context-dependent decision-making task, the trained network consisted of a single connectivity mode with patterns $\boldsymbol{m}$ and $\boldsymbol{n}$, and therefore a single internal variable κ, while the input consisted of two patterns $\boldsymbol{I}^A$ and $\boldsymbol{I}^B$ for the two stimulus features, and two patterns $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$ for the contextual cues. The minimal trained networks consisted of two sub-populations, so that each connectivity and input pattern was split into two sub-patterns. Combining the clustering analysis with mean-field theory revealed three key properties for these sub-patterns. First, input-feature and contextual cue patterns play repectively the roles of drivers and modulators. Indeed, the contextual cue patterns $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$ were mostly orthogonal to input-selection pattern $\boldsymbol{n}$ (Fig. 4b), and therefore modulated gains but did not drive the dynamics, while input-feature patterns had non-zero covariances with the input-selection pattern (Fig. 4b) and therefore directly drove the dynamics of the internal variable. Second, each of the two input-selection sub-patterns was correlated with only one of the input-feature patterns. Specifically, for the first population, the input-selection sub-pattern overlapped with input pattern $\boldsymbol{I}^A$ and not $\boldsymbol{I}^B$ (i.e. $\sigma^{(1)}_{nI^A} > 0$ and $\sigma^{(1)}_{nI^B} \approx 0$), and conversely the second sub-population input-selection pattern overlapped with feature input pattern $\boldsymbol{I}^B$ and not $\boldsymbol{I}^A$ ($\sigma^{(2)}_{nI^A} \approx 0$ and $\sigma^{(2)}_{nI^B} > 0$, see Fig. 4b). Third, each context-cue pattern had a strong variance on a different sub-population, and therefore the two contextual cues modulated the gains $\langle\Phi'\rangle_1$ and $\langle\Phi'\rangle_2$ of the two populations in a differential manner (Fig. 4g). Altogether, the dynamics of the internal collective variable could therefore be described by a reduced model of the form

$$\frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{mn}\kappa + \sigma^{(1)}_{nI^A}\langle\Phi'\rangle_1 u_A(t) + \sigma^{(2)}_{nI^B}\langle\Phi'\rangle_2 u_B(t) \tag{5}$$

with $\langle\Phi'\rangle_1$ and $\langle\Phi'\rangle_2$ varying in opposite directions in the two contexts. As a consequence, the effective couplings between stimulus features u_A/u_B and the internal variable were strongly modulated by contextual cues through gain-modulation, with contextual cues effectively switching off an input to κ in each context. This reduced model was analogous to classical perceptual decision making (Fig. 2c), but the internal variable selectively integrated only one stimulus feature in each context. This mechanism allowed the network to flexibly respond to non-congruent stimuli, and consequently the networks generated using this reduced model performed the task with high accuracy (Fig. 4c). Importantly, the contextual gating of the integrated stimulus feature relied on recurrent dynamics and took place only on the level of effective inputs to the internal variable, not at the level of overall inputs to the network. On the overall population level, the two stimulus features were equally represented in both contexts, but along directions orthogonal to the internal collective variable (Fig. 4d) as observed in experimental data [Mante et al., 2013]. The selective gating identified in the reduced model (Eq. (5)) is therefore not directly apparent at the level of low-dimensional dynamics of the trained network (Fig. 4d), but can be revealed by splitting the contribution from the two populations to the internal variable (compare Fig. 4g for the trained network with Fig. S3b for the reduced model).

For the delay-match-to-sample task, the trained network consisted of two connectivity modes, and therefore the internal dynamics was described by two internal variables κ_1 and κ_2. The stimuli corresponded to two patterns $\boldsymbol{I}^A$ and $\boldsymbol{I}^B$, which were activated in two trial epochs (Fig. 5a). In contrast to the context-dependent decision-making task, the input patterns were essentially orthogonal to the input-selection connectivity patterns (Fig. 5a), and therefore did not directly drive the internal collective variables, but acted instead as modulators. As a consequence, the dynamics was mostly driven by recurrent interactions between internal collective variables, and could be visualised in terms of a flow in a dynamical landscape in the $\kappa_1 - \kappa_2$ plane (Fig. 5d). The main effect of the inputs was to shape the trajectories of neural activity in this plane by modulating the dynamical

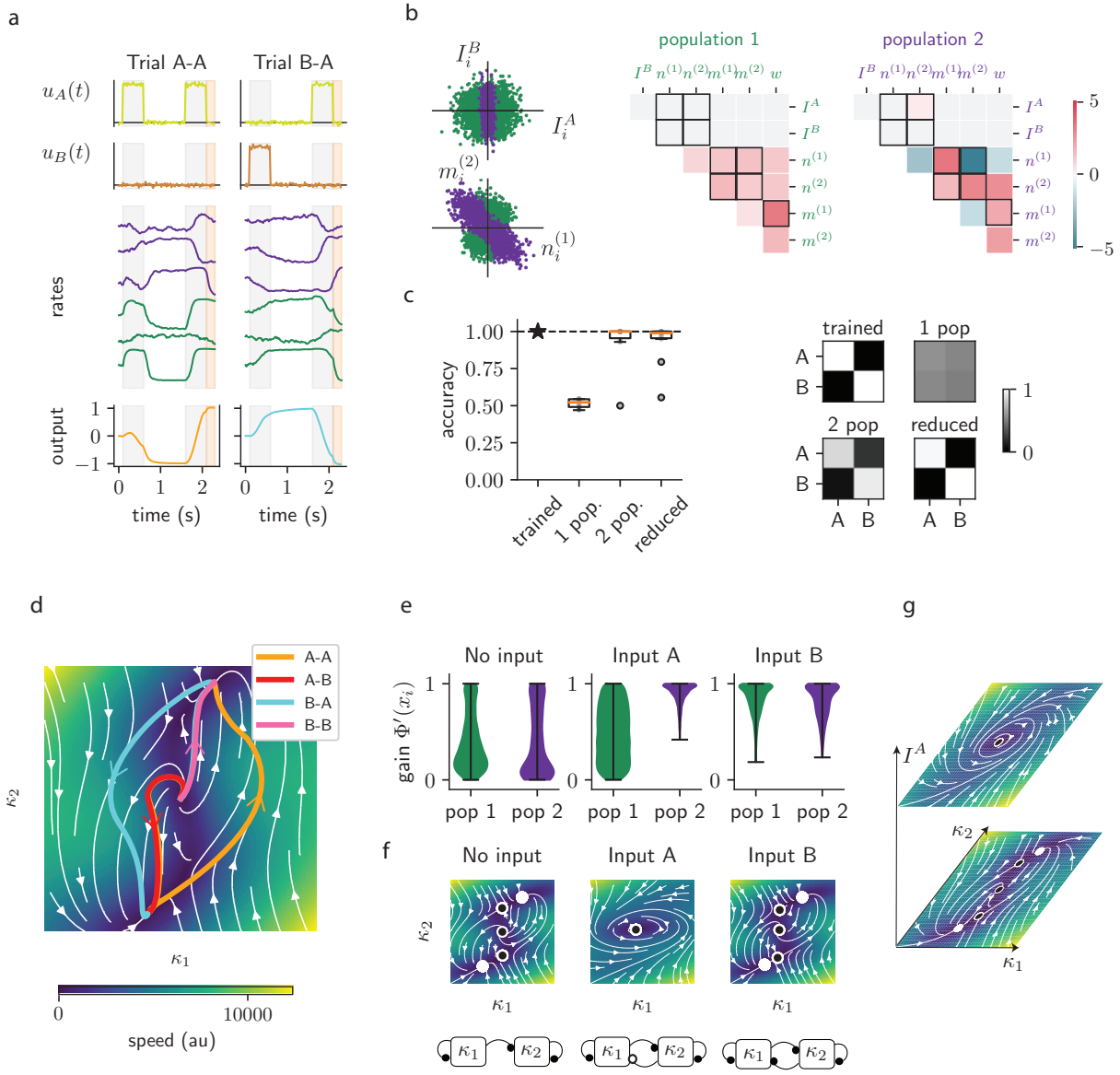

Figure 5: Delay match-to-sample task. A rank-two network received on each stimulation period one out of two stimuli A or B, represented by different input patterns, and was trained to report whether the two stimuli were identical or different. (a) Illustration of the dynamics in two trials corresponding to stimuli A-A (left) and B-A (right). Top-bottom: inputs to the network; traces of six neurons, three from each identified sub-population; readout generated by the output. (b) Statistics of the trained pattern loadings, determined by a clustering analysis in the seven-dimensional loading space. Left: two two-dimensional projections of the loading space, showing the presence of two clusters that defined two different sub-populations. Right: empirical covariances between patterns for each of the two sub-populations. (c) Task accuracy and response matrices for the trained, resampled and minimal reduced networks. Ten resampled networks were generated by sampling loadings from either a single-population ($P = 1$) or a two-population ($P = 2$) Gaussian distribution with covariance matrices fitted to the trained network (error bars display standard deviations). The reduced model was generated by adjusting only the covariances outlined in black in (b).

Figure 5 *(previous page)*: (d) Trajectories of activity in the two-dimensional subspace corresponding to the two internal collective variables. This subspace is spanned by vectors $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$. Trajectories in the four possible trials are shown in different colors. The color density plot and flow lines display the dynamical landscape corresponding to the speed and orientation of the autonomous dynamics of the network. (e) Distribution of single neuron gains across both sub-populations in presence of different inputs. (f) Dynamical landscape in the 2D subspace corresponding to the two internal collective variables, in presence of different inputs, and the associated circuit diagrams. Stable fixed points of the dynamics are indicated by filled white dots, unstable and saddle points by dots filled in black. (g) Illustration of the different cuts in the full state space corresponding to the different two-dimensional dynamical landscapes shown in (f).

landscape at different trial epochs (Fig. 5f,g and Supplementary Fig. S4c,d). These modulations of the dynamical landscape relied on the organization of the network in two sub-populations. Indeed, we found that overlaps between sub-patterns of connectivity vectors differed strongly between the two populations (Fig. 5b). As shown in Eq. (4), the effective coupling between the internal collective variables is determined by a weighted average of overlaps corresponding to the individual sub-populations, where the weights are given by the gains $\langle \Phi' \rangle_p$ of individual sub-populations. The stimuli differentially modulated the gains of the two sub-populations (Fig. 5e), so that the effective couplings interpolated between the overlaps of the two sub-populations. In the trained network described in Fig. 5, we found that the first population implemented positive feedback between the two internal variables, while the second population implemented negative feedback. In absence of inputs, positive and negative feedback balanced each other (Fig. 5f left), but individual stimuli disrupted this balance and strongly modified the dynamical landscape. In particular stimulus A strongly enhanced negative feedback (Fig. 5f middle), which led to a limit-cycle in the dynamics that opened a fast transient channel that could flip neural activity in the $\kappa_1 - \kappa_2$ plane [Chaisangmongkon et al., 2017]. Each trial in the task therefore corresponded to a different sequence of dynamical landscapes and therefore led to a different trajectory from the initial to the final state of neural activity that determined the output. A minimal, reduced model built from the observed population statistics performed the task with accuracy indistinguishable from the trained network (Fig. 5c), thus confirming the dynamical mechanisms behind the computation.

In summary, we found that networks with multiple sub-populations implemented flexible computations by exploiting gain modulation to modify effective couplings between collective variables. The two tasks displayed in Fig. 4 and Fig. 5 illustrate two different variants of this general mechanism. In the context-dependent decision-making task, the sensory inputs acted as drivers of the internal dynamics, and contextual inputs as gain modulators that controlled effective coupling between the sensory inputs and the internal collective variable. In contrast, in the delay-match-to-sample task, sensory inputs acted as modulators of recurrent interactions, and gain modulation controlled only the effective coupling between the two internal variables. More generally, modulations of inputs and modulations of recurrent interactions could be combined to implement more complex tasks.

2.5 Implications for structure in neural selectivity

Our analyses of trained networks showed that flexible tasks required a sub-population structure in the connectivity, while simpler tasks did not. We next explored the experimental predictions of these findings. Current experimental procedures rarely allow to access the connectivity in animals trained on specific tasks. Instead, experiments typically record the activity of large neural populations during behavior, after animals have been trained on the task. We therefore examined how the experimentally accessible structure of neural activity reflects the underlying structure of connectivity.

A common approach to characterizing the relationship between neural activity and the ongoing computation is to analyze the selectivity of individual cells by performing a linear regression of activity with respect to controlled or measured task variables, such as stimulus, context or behavioral choice [Park et al., 2014; Mante et al., 2013; Raposo et al., 2014; Aoi and Pillow, 2018; Kobak et al., 2016]. For each neuron i, and at every time point t, this approach yields a set of regression coefficients $\beta_i^{(k)(t)}$, which quantify how much the activity depends on the task variable k. In our computational framework, the inputs x_i to the neurons are directly set by the input patterns $\boldsymbol{I}^{(k)}$ and connectivity patterns $\boldsymbol{m}^{(r)}$ (Eq. (2)), and regression coefficients of firing rates $\phi(x_i)$ necessarily

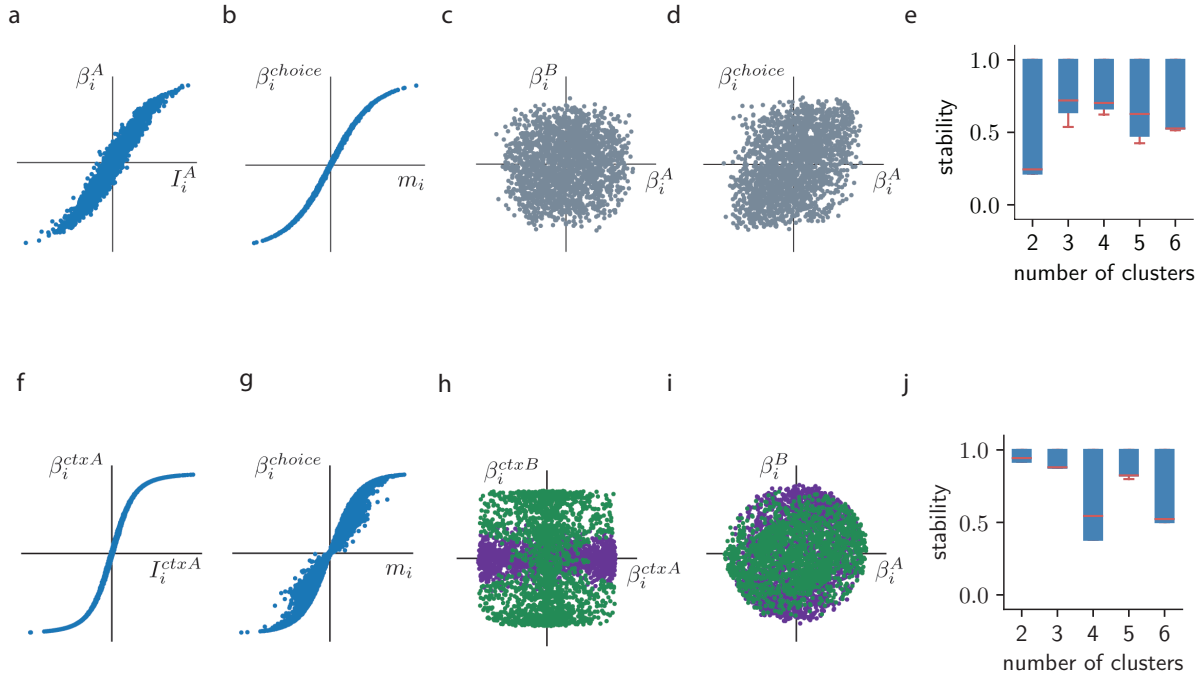

Figure 6. Connectivity structure determines selectivity to task variables. (a) - (e) Regression analysis in a network trained on the multi-sensory integration task. (a) Relationship between input mode A and regressor with respect to input A for each neuron. (b) Same for output mode m and choice regressors. (c) Relationship between input A regressors and input B regressors for each neuron. (d) Relationship between choice and input A regressors. (e) Clustering stability analysis on the 3 dimensional regressor space. No evidence is found for a particular number of clusters in this space (see methods). (f) - (g) Regression analysis in a network trained on the context-dependent decision-making task. (f) Relationship between input mode for context A and regressor with respect to context A. (g) Same for output mode m and choice regressor. (h) Relationship between context A and context B regressors, with 2 clusters found on the 5-dimensional regressor space. (i) Relationship between input A and B regressors, with the same clusters. (j) Clustering stability analysis on the 5-dimensional regressor space shows evidence for the presence of 2 clusters.

reflect this network structure, albeit non-linearly through the transfer function ϕ. In particular, during stimulus presentation, the components $I_i^{(k)}$ of the input pattern directly determine the regression coefficients with respect to the sensory input k. During the decision epoch, the regression coefficients with respect to the choice are in turn determined by the readout pattern $\boldsymbol{w}$ [Haefner et al., 2013], which in each of the tasks we studied can be approximated by a linear combination of output connectivity patterns $\boldsymbol{m}^{(r)}$. Since input patterns $\boldsymbol{I}^{(k)}$ and the output connectivity patterns $\boldsymbol{m}^{(r)}$ determine regression coefficients with respect to stimuli and choice, a sub-population structure corresponding to clusters in pattern loading space implies the existence of clusters in the regression space, where each axis corresponds to a regression coefficient with respect to a different variable. This relationship between sub-population structure in connectivity and in selectivity leads to predictions that we next illustrate for two specific tasks.

We start with the multi-sensory integration task [Raposo et al., 2014], which is an extension of the perceptual decision-making task to the situation where stimuli of two different modalities need to be integrated. Importantly, in this task, the sensory inputs corresponding to the two modalities are always congruent, i.e. they point to the same decision. We found that this task could be implemented by a unit-rank, single population network similar to perceptual decision-making (see also [Sederberg and Nemenman, 2020]), the only difference being that the two modalities correspond to two independent input patterns (Supplementary Fig. S8). The loadings of these input patterns onto individual neurons directly determine the regression coefficients with respect to the two stimulus modalities through a non-linear transform (Fig. 6a). As the network is unit-rank, it possesses a single output connectivity pattern, which fully determines the readout, and the regression coefficients with respect to choice therefore correspond to a non-linear transform of the loadings for the output connectivity pattern (Fig. 6b). Since the network consists of a single population, the input and connectivity patterns form a single cluster in loading space (Supplementary Fig. S8b). As long as the single-unit firing rates do not strongly saturate, this implies the presence of a single cluster in the space of regression coefficients, a situation referred to as fully mixed, or category-free selectivity [Raposo et al., 2014; Hirokawa et al., 2019]. To test the presence or absence of clusters in the regressor space, we ran a bootstrap stability analysis [Hirokawa et al., 2019] (Fig. 6e, Methods), essentially applying a clustering algorithm to several subsamples of the data and measuring the consistency of its results accross subsamples. This analysis showed that all clusterings in more than one population provided a poor fit to the data, thus indicating a non-clustered structure in regressor space. Detailed analyses of selectivity in neural activity recorded in this task have precisely pointed out such a lack of structure [Raposo et al., 2014], and are therefore in line with the predictions of our network models.

We next turn to the context-dependent decision-making task [Mante et al., 2013], which is essentially an extension of the multi-sensory integration task to the case where the two modalities can indicate incongruent, conflicting choices, and the relevant modality is indicated by a contextual cue. In this task, on top of sensory and choice regressors, we have also access to regression coefficients with respect to the two contextual cues, which directly reflect the corresponding contextual input patterns (Fig. 6f). As shown in previous sections, correct context-dependent responses to incongruent stimuli in trained networks require the presence of two different populations, that correspond to two clusters in loading space determined in particular by the two contextual inputs (Fig. 4b). These induce corresponding clusters in the regression space, for which our bootstrap analysis found evidence (Fig. 6j), and that strongly match the sub-populations in the connectivity loading space (F1-score=0.89). These are specifically apparent in the plane of regression coefficients to the contexts (Fig. 6h), but not along other projections (Fig. 6i). While the selectivity is still mixed and varies strongly among neurons, it is not fully random and contains structure that is key to the task performance. Note however that the strength of clustering in the regression space depends both on the strength of clustering in the loading space, and on how much individual neurons engage the non-linearity, since the regression coefficients are determined from firing rates $\phi(x_i)$. In particular, strongly non-linear activity may induce additional, spurious clusters in regression space. Functional clusters that take part in computations can then be identified either by comparing clusters in the loading and regression spaces, or by examining the effects of inactivating sub-populations as we show next.

2.6 Predictions for inactivations of specific sub-populations

In addition to implications for single-neuron selectivity, the functional sub-population structure present in networks trained on flexible input-output tasks implies specific effects on the output when clusters are silenced, and therefore leads to predictions for inactivation experiments that we illustrate here for the same two tasks as in Fig. 6.

In our network models, the input and connectivity patterns are highly distributed over individual neurons. As a consequence, the dynamics and computations in the networks are highly resilient to random, unpatterned perturbations. Inactivating a large random fraction of the neurons therefore leaves the input-output transform intact, and merely increases the effective noise in the output, irrespective of the task, and irrespective of whether the network contained a single (Fig. 7b) or multiple (Fig. 7d) subpopulations. When the computation relies on the presence of several sub-populations, inactivating instead neurons belonging to a specific sub-population produces highly specific effects that are determined by the role of that sub-population in the computation. In the context-dependent decision-making task, inactivating randomly half of the neurons within a sub-population responsible for context A switched the computation in context A from feature-selection to feature integration similar to the multi-sensory integration task, while it essentially left intact the input-output association in context B (Fig. 7e). Inactivating a specific sub-population therefore directly revealed its role in selecting the relevant stimulus feature to integrate.

The inactivations displayed in Fig. 7e,f assume that connectivity-defined sub-populations have been previously identified and made accessible for perturbations [Peron et al., 2020]. In a more realistic setting, the neurons belonging to the relevant sub-populations need first to be functionally identified, and our model provides a direct guidance for that preliminary step. As outlined above, the model predicts that neurons that specifically select the feature A or B correspond to distinct clusters in the plane defined by regression coefficients with respect to contextual cues (Fig. 6h). Since these neurons are also the ones that respond most strongly to separate contextual cues, a simple alternative for identifying the two sub-populations relevant for contextual computations is therefore to select the two groups of neurons with contextual regression coefficients larger than a threshold. Inactivating each of the obtained sub-populations then leads to the same specific disruptions of performance as inactivating the actual connectivity-defined subpopulations (Fig. 7g,h).

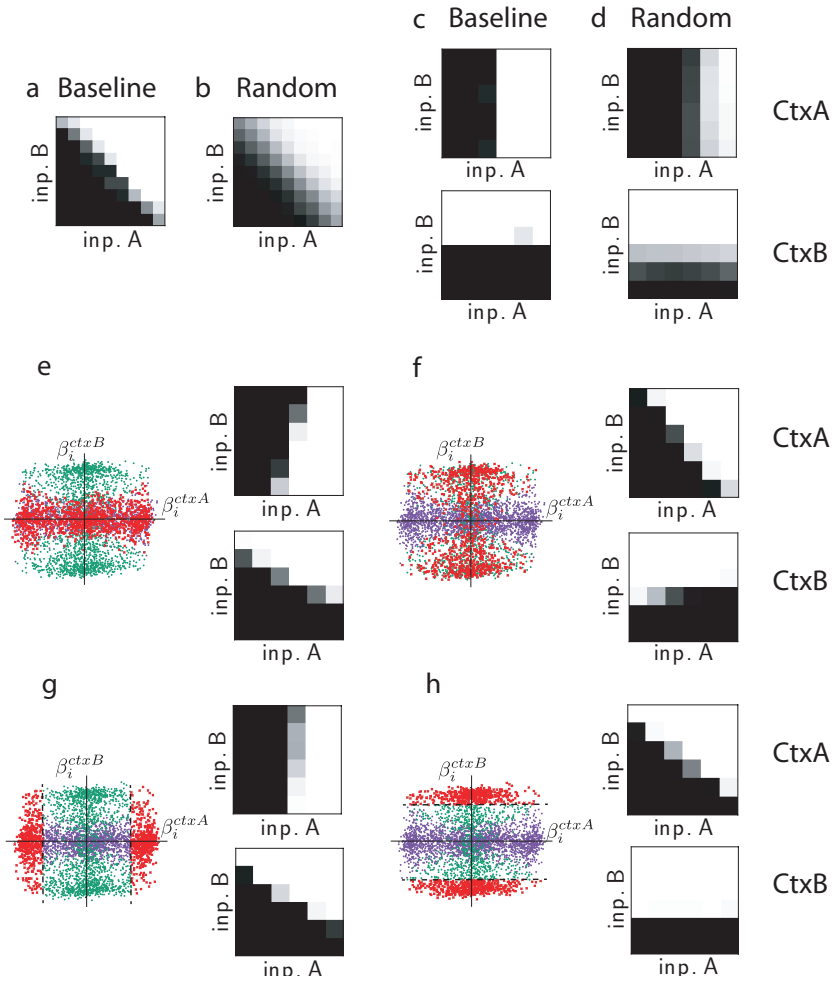

Figure 7. Population structure predicts specific effects for inactivations. (a) Baseline performance matrices for the multi-sensory integration task. (b) Performance matrices after randomly inactivating half of the network for the multi-sensory integration task. (c) Baseline performance matrices for the context-dependent decision-making task. (d) Performance matrices after randomly inactivating half of the network for the context-dependent decision-making task. (e) Left: half of the neurons in population 1 were inactivated (red-crosses). Right: Performance matrices after randomly inactivating half of population 1. (f) Same as (e) for population 2 inactivations for the context-dependent decision-making task. (g,h) Same as (e,f) but with inactivated neurons chosen based on their selectivity to context A or B.

3 Discussion

Our goal was to determine whether and when a multi-population structure is necessary for a network to perform specific computations. To address this question, we reverse-engineered recurrent neural networks trained on a set of neuroscience tasks using a new theoretical framework, in which sub-populations of neurons, and the dimensionality of the dynamics are controlled independently. Although a number of tasks could be implemented by increasing dimensionality in networks where all neurons were statistically equivalent, we found that tasks based on flexible input-output mappings instead required neurons to be structured in several sub-populations that played functionally distinct roles. It has been previously argued that the organization in sub-populations may be important in early sensory areas where individual cells perform specific computations [Hardcastle et al., 2017; Kastner et al., 2015; Tanaka et al., 2019; Ocko et al., 2018], and less prevalent in higher cortical areas where neurons typically multi-task [Raposo et al., 2014; Fusi et al., 2016]. Our results instead support the idea that some neurons need to be specialized for specific parts of the computation if a task is complex enough [Yang et al., 2019], so that multi-population structure is expected also in higher cortical areas, as found in [Hirokawa et al., 2019].

Our theoretical analysis shows that, within the collective dynamics paradigm where trajectories of activity implement computations, sub-population structure plays a fundamentally complementary role to the dimensionality of the dynamics. While the dimensionality sets the number of latent, collective variables available for computations, the sub-population structure in contrast determines the collective dynamics of these variables, and their response to inputs. Mechanistically, this role of sub-population structure can be understood from two perspectives. From the neural state-space perspective, the collective dynamics explore a low-dimensional recurrent subspace, and the sub-population structure shapes the non-linear dynamical landscape of the activity in this subspace [Sussillo and Barak, 2013]. Specifically, different inputs differentially activate different populations, and shift the recurrent sub-space into different regions of the state-space with different non-linear dynamical landscapes (Fig. 5g). A complementary picture emerges from the perspective of an effective circuit description (Fig. 1d), where the collective dynamics is described in terms of interactions between the latent, collective variables through effective couplings. In that picture, the sub-population structure allows inputs to control the effective couplings by modulating the average gain of different sub-populations. The computations then rely on two functionally distinct types of inputs: drivers that directly entrain the collective variables, and modulators that shape the gains of the different sub-populations, and thereby the interactions between collective variables. Interestingly, gain modulation has long been posited as a mechanism underlying selective attention [Rabinowitz et al., 2015], a type of processing closely related to flexible input-output tasks considered here. While patterns of gain modulation [Salinas and Thier, 2000; Ferguson and Cardin, 2020], and the distinction between drivers and modulators [Sherman and Guillery, 1998] are fundamentally physiological concepts, here we found that analogous mechanism emerge in abstract trained networks at the population level of collective variables. Note that in our framework, driver and modulators are indistinguishable at the single cell level, where they both correspond to additive inputs (in contrast to eg neuro-modulation that may multiplicatively control the gain of individual neurons, see [Stroud et al., 2018]). The functional distinction between drivers and modulators instead stems from the relation between the collective pattern of inputs, and the recurrent connectivity in the network.

Our framework is based on a highly abstract concept of sub-populations, defined as clusters in the connectivity loading space. In particular, we did not implement any explicit anatomical constraint such as Dale's law, hence sub-populations appear for purely functional, computational reasons. What could be the physiological counter-parts of the different functional sub-populations that we identified? There are at least two distinct possibilities. In the network trained on the context-dependent decision-making task, we found that the two sub-populations differed only in the relationship of their connectivity with respect to feed-forward and contextual inputs. Such sub-populations therefore bear an analogy with input- and output-defined cortical populations such as for instance defined by inputs from the thalamus [Harris and Mrsic-Flogel, 2013; Schmitt et al., 2017] or outputs to the striatum [Znamenskiy and Zador, 2013]. In the network trained on the delay-match-to sample task, the two sub-populations instead differed at the level of recurrent connectivity: one population implemented positive, and the other negative feedback, the two being in general balanced, except in response to one of the two stimuli. This situation is reminiscent of excitatory and inhibitory sub-populations, which effectively implement positive and negative feedback in biological networks. More generally, these observations pave the way for more systematic comparisons between functional and anatomical cell types, though additional biological constraints will need to be included in our network models.

In this study, for each task we explicitly sought to identify networks with the minimal rank and minimal number of sub-populations. This was achieved in particular by directly constraining the connectivity matrix to be of low-rank, and by approximating the distribution of loadings with the minimal number of Gaussian populations, an approach akin to a strong type of regularization. Remarkably, networks trained without the low-rank constraint lead to connectivities that are also based on a low-rank structure [Schuessler et al., 2020b], but this structure is generally of higher rank than found here, and correlated with the underlying full-rank initial connectivity. The solutions to the various tasks we identified here are therefore not unique - other solutions with higher rank and higher number of sub-populations appear depending on the details of training (see Supplementary section S3 and Fig. S5, S10). Our method for reducing the trained networks to simpler effective models is still applicable to trained networks with higher rank and number of sub-populations. The overall computational mechanism remains based on gain-controlled modulation of effective couplings, but the specific instantiations of this mechanism become more complex.

The fact that neurons are selective to mixtures of task variables rather than individual features has emerged as one of the defining properties of higher order areas in the mammalian cortex [Fusi et al., 2016]. Mixed selectivity however does not necessarily preclude the presence of any sub-population structure. Indeed, moving beyond the dichotomy between pure and mixed selectivity, recent works have begun to distinguish between various types of mixed selectivity. In particular, fully random mixed selectivity, where the distribution in selectivity space is fully isotropic [Raposo et al., 2014] has been contrasted with structured mixed selectivity, where cells can be assigned to different categories based on clusters in the space of selectivity to different task variables [Hirokawa et al., 2019]. Here we followed a similar approach by determining whether the distribution in the connectivity space could be approximated by a single Gaussian cluster, or requires a mixture of several Gaussians that define effective populations. Since the presence of sub-population structure in the connectivity implies the presence of clusters in selectivity space (Fig. 6h), our results predict that the expected type of mixed selectivity depends on the complexity of the performed task. For tasks requiring flexible input-output associations, we predict the presence of clusters in the selectivity space, but this statistical structure is however computationally necessary only for selectivity to specific variables, that depend on the considered task, while selectivity to other variables can be fully random. Ultimately, identifying specific signatures of computational mechanisms in the neural data therefore requires a careful comparison with recurrent network models constrained by both behavior and neural activity on a task-by-task basis.

Acknowledgements

The project was supported by the ANR project MORSE (ANR-16-CE37-0016), the CRCNS project PIND, the program “Ecoles Universitaires de Recherche” launched by the French Government and implemented by the ANR, with the reference ANR-17-EURE-0017. There are no competing interests. SO thanks Joshua Johansen and Bijan Pesaran for fruitful discussions.

Code availability

Code and trained models will be made available upon publication.

4 Methods

4.1 Low-rank networks

We considered networks of N rate units that evolve over time according to

$$\tau \frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} J_{ij}\phi(x_j) + I_i^{FF}(t) + \eta_i(t). \tag{6}$$

Here x_i represents the *activation* or total current received by the ith recurrent neuron, and $\phi(x_i) = \tanh(x_i)$ is its *activity* or firing rate. Each neuron received an independent white-noise input $\eta_i(t)$.
The connectivity matrix J was constrained to be of rank R, so that it can be represented as

$$J_{ij} = \frac{1}{N}\sum_{r=1}^{R} m_i^{(r)} n_j^{(r)} \tag{7}$$

i.e. as a sum of R modes, the mode r consisting of an outer-product of vectors $\boldsymbol{m}^{(r)} = \{m_i^{(r)}\}_{i=1\ldots N}$ and $\boldsymbol{n}^{(r)} = \{n_i^{(r)}\}_{i=1\ldots N}$. Throughout the text, we refer to the vectors $\boldsymbol{m}^{(r)}$ and $\boldsymbol{n}^{(r)}$ as the *connectivity patterns*, with $\boldsymbol{m}^{(r)}$ the r-th output pattern, and $\boldsymbol{n}^{(r)}$ the r-th input-selection pattern. Without loss of generality, we will assume that all the output patterns (and respectively all the input-selection patterns) are mutually orthogonal. Such a representation is uniquely defined by singular-value decomposition of the connectivity matrix.
The network received feedforward inputs $\boldsymbol{I}^{FF}(t)$ generated by N_{in} temporally-varying scalar stimuli $u_s(t)$, each fed into the network through a set of weights $I_i^{(s)}$:

$$I_i^{FF}(t) = \sum_{s=1}^{N_{in}} I_i^{(s)} u_s(t). \tag{8}$$

We refer to the vector $\boldsymbol{I}^{(s)} = \{I_i^{(s)}\}_{i=1\ldots N}$ as the s-th *input pattern*.
The output of the network is defined by readout values

$$z_k = \frac{1}{N}\sum_{j=1}^{N} w_j^{(k)}\phi(x_j) \quad k = 1\ldots N_{out}. \tag{9}$$

where $\boldsymbol{w}^{(k)} = \{w_i^{(k)}\}_{i=1\ldots N}$ is the k-th *readout pattern*.
The time constant of neurons was $\tau = 100$ms. For simulation and training this equation was discretized using Euler's method with a time step $\Delta t = 20$ms. The white noise η_i is simulated by drawing at each time step from a centered Gaussian distribution of standard deviation 0.05.

4.2 Low-dimensional dynamics

The dynamics defined by Eq. (6) can be represented as a trajectory in the N-dimensional state space in which each axis corresponds to the activation x_i of unit i. In low-rank networks, the dynamics is confined to a low-dimensional subspace of this state-space [Mastrogiuseppe and Ostojic, 2018]. Inserting Eq. (7) into Eq. (6), the activation vector $\boldsymbol{x} = \{x_i\}_{i=1\ldots N}$ can be expressed in terms of R internal collective variables κ_r, and N_{in} external collective variables v_s:

$$\boldsymbol{x}(t) = \sum_{r=1}^{R} \kappa_r(t)\boldsymbol{m}^{(r)} + \sum_{s=1}^{N_{in}} v_s(t)\boldsymbol{I}_{\perp}^{(s)}. \tag{10}$$

The first term on the right-hand side in Eq. (10) represents the component of the activity on the *recurrent space* [Wang et al., 2018; Remington et al., 2018] defined as the sub-space spanned by the output connectivity patterns $\{\boldsymbol{m}^{(r)}\}_{r=1\ldots R}$. The corresponding internal collective variables κ_r are defined as projections of the activation vector $\boldsymbol{x}$ on the $\boldsymbol{m}^{(r)}$:

$$\kappa_r(t) = \frac{1}{\|\boldsymbol{m}^{(r)}\|^2} \sum_{j=1}^{N} m_j^{(r)} x_j(t). \tag{11}$$

The second term on the right-hand side in Eq. (10) represents the component of the activity on the *input space* defined as the sub-space spanned by $\{\boldsymbol{I}_\perp^{(s)}\}_{s=1...N_{in}}$, the set of input vectors orthogonalized with respect to the recurrent sub-space. The corresponding external collective variables v_s are defined as projections of the activation vector $\boldsymbol{x}$ on the $\boldsymbol{I}_\perp^{(s)}$:

$$v_s(t) = \frac{1}{\|\boldsymbol{I}_\perp^{(s)}\|^2} \sum_{j=1}^{N} I_{\perp,j}^{(s)} x_j(t). \tag{12}$$

The dimensionality of the dynamics in activation space is thus given by the sum of the dimension R of the recurrent sub-space, i.e. the rank of the connectivity, and the dimensionality of the input space $\{\boldsymbol{I}_\perp^{(s)}\}_{s=1...N_{in}}$. The dynamics of the internal variables κ_r is obtained by projecting Eq. (6) onto the output connectivity patterns $\boldsymbol{m}^{(r)}$:

$$\tau \frac{d\kappa_r}{dt} = -\kappa_r(t) + \kappa_r^{rec}(t) + \frac{1}{\|\boldsymbol{m}^{(r)}\|^2} \sum_{j=1}^{N} m_j^{(r)} \sum_{s=1}^{N_{in}} I_j^s u_s(t) \tag{13}$$

where κ_r^{rec} represents the recurrent input to the r-th collective variable, defined as the projection of the firing rate vector $\phi(\boldsymbol{x})$ onto the input-selection pattern $\boldsymbol{n}^{(r)}$:

$$\kappa_r^{rec}(t) = \frac{1}{N} \sum_{j=1}^{N} n_j^{(r)} \phi(x_j(t)). \tag{14}$$

Inserting Eq. (10) into κ_r^{rec} leads to a closed set of equations for the κ_r:

$$\kappa_r^{rec}(t) = \frac{1}{N} \sum_{j=1}^{N} n_j^{(r)} \phi\left(\sum_{r'=1}^{R} \kappa_r'(t) m_j^{(r')} + \sum_{s=1}^{N_{in}} I_{\perp,j}^s v_s(t) \right). \tag{15}$$

The dynamics of the external variables v_s is obtained by projecting Eq. (6) onto the orthogonalized input patterns $\boldsymbol{I}_\perp^{(s)}$. They are given by external inputs $u_s(t)$ filtered by the single neurons time constant τ

$$\tau \frac{dv_s}{dt} = -v_s + u_s. \tag{16}$$

Throughout the main text, we assume for simplicity that the stimuli u_s vary on a timescale slower than τ, and replace v_s with u_s.
The readout values $\{z_k\}_{k=1...N_{out}}$ can then be expressed in terms of the collective variables

$$z_k(t) = \frac{1}{N} \sum_{j=1}^{N} w_j^{(k)} \phi\left(\sum_{r'=1}^{R} \kappa_r'(t) m_j^{(r')} + \sum_{s=1}^{N_{in}} I_{\perp,j}^s v_s(t) \right). \tag{17}$$

4.3 Loading space and mean-field limit

The dynamics of the collective variables is fundamentally determined by the components of connectivity and input patterns through Eq. (15). From Eq. (10), and by analogy with factor analysis, we call *loadings* the components of different patterns on each neuron. Neuron i is therefore characterized by the $2R + N_{in} + N_{out}$ loadings

$$\{\{n_i^{(r)}\}_{r=1...R}, \{m_i^{(r)}\}_{r=1...R}, \{I_i^{(s)}\}_{s=1...N_{in}}, \{w_i^{(q)}\}_{q=1...N_{out}}\}. \tag{18}$$

Each neuron can thus be represented as a point in the *loading space* of dimension $2R + N_{in} + N_{out}$, and the connectivity of the full network can therefore be described as a set of N points in this space. Note that the

right-hand-side of Eq. (15) consists of a sum of N terms, where the term j contains only the loadings of neuron j. The loadings of different neurons therefore do not interact in κ_r^{rec}, so that the r.h.s of Eq. (15) can be interpreted as an average over the set of points corresponding to all neurons in the loading space.
Our main assumption will be that in the limit of large networks ($N \to \infty$), the set of points in the loading space can be described by a probability distribution $P(n^{(1)}, \dots, n^{(R)}, m^{(1)}, \dots, m^{(R)}, I^{(1)}, \dots, I^{(N_{in})}, w^{(1)}, \dots, w^{(N_{out})}) := P(\underline{n}, \underline{m}, \underline{I}, \underline{w})$. In this mean-field limit, the r.h.s. of Eq. (15) becomes:

$$\kappa_r^{rec}(t) = \int \mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I}\, \mathrm{d}\underline{w}\; P(\underline{n}, \underline{m}, \underline{I}, \underline{w})\; n^{(r)} \phi\left(\sum_{r'=1}^{R} \kappa_{r'}(t) m^{(r')} + \sum_{s'=1}^{N_{in}} I_\perp^{(s')} v_{s'}(t) \right) \tag{19}$$

where we have used the shorthand $\mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I}\, \mathrm{d}\underline{w} = \prod_{r'=1}^{R} \prod_{s'=1}^{N_{in}} \prod_{q'=1}^{N_{out}} \left(\mathrm{d}m^{(r')}\, \mathrm{d}n^{(r')}\, \mathrm{d}I^{(s')}\, \mathrm{d}w^{(q')} \right)$. The collective dynamics is therefore fully specified by the single-neuron distribution of pattern loadings. Once this distribution is specified, any network generated by sampling from it will have identical collective dynamics in the limit of a large number of neurons.
This also sets the values of the readouts

$$z_k(t) = \int \mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I}\, \mathrm{d}\underline{w}\; P(\underline{n}, \underline{m}, \underline{I}, \underline{w})\; w^{(k)} \phi\left(\sum_{r'=1}^{R} \kappa_{r'}(t) m^{(r')} + \sum_{s'=1}^{N_{in}} I_\perp^{(s')} v_{s'}(t) \right). \tag{20}$$

4.4 Statistics of loadings and sub-populations

To approximate any arbitrary joint distributions of loadings $P(\underline{n}, \underline{m}, \underline{I}, \underline{w})$, we used multivariate Gaussian mixture models (GMMs). This choice was based on the following considerations: (i) GMMs are able to approximate an arbitrary multi-variate distribution [Kostantinos, 2000]; (ii) model parameters can be easily inferred from data using GMM clustering; (iii) GMMs afford a natural interpretation in terms of sub-populations (iv) GMMs allow for a mathematically tractable and transparent analysis of the dynamics as shown below.
In a multivariate Gaussian mixture model, every neuron belongs to one of P sub-populations. For a neuron in sub-population p, the set of loadings $\{\{n_i^{(r)}\}_{r=1\dots R}, \{m_i^{(r)}\}_{r=1\dots R}, \{I_i^{(s)}\}_{s=1\dots N_{in}}, \{w_i^{(q)}\}_{q=1\dots N_{out}}\}$ is generated from a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_p$ and covariance $\boldsymbol{\Sigma}_p$, where $\boldsymbol{\mu}_p$ is a vector of size $2R + N_{in} + N_{out}$, and $\boldsymbol{\Sigma}_p$ is a covariance matrix of size $(2R + N_{in} + N_{out})^2$. The full distribution of loadings is therefore given by

$$P(\underline{n}, \underline{m}, \underline{I}, \underline{w}) = \sum_{p=1}^{P} \alpha_p \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p) \tag{21}$$

$$:= \sum_{p=1}^{P} \alpha_p P_p(\underline{n}, \underline{m}, \underline{I}, \underline{w}) \tag{22}$$

where the coefficients α_p define the fraction of neurons belonging to each sub-population.
Each sub-population directly corresponds to a Gaussian cluster of points in the loading space. The vector $\boldsymbol{\mu}_p$ determines the center of the p-th cluster, while the covariance matrix $\boldsymbol{\Sigma}_p$ determines its shape and orientation. For a neuron i belonging to population p, we will write as $\sigma_{ab}^{(p)}$ the covariance between two loadings a and b, with $a, b \in \{\{n^{(r)}\}_{r=1\dots R}, \{m^{(r)}\}_{r=1\dots R}, \{I^{(s)}\}_{s=1\dots N_{in}}, \{w^{(q)}\}_{q=1\dots N_{out}}\}$. Note that because the output patterns $\boldsymbol{m}^{(r)}$ (resp. input-selection patterns $\boldsymbol{n}^{(r)}$) are mutually orthogonal, the covariances between the loadings $\{m_i^{(r)}\}_{r=1\dots R}$ (respectively $\{n_i^{(r)}\}_{r=1\dots R}$) vanish.
Since every neuron belongs to a single population, the r.h.s of Eq. (15) can be split into P terms, each corresponding to an average over one population. As within each population the distribution of loadings is a joint Gaussian, Eq. (19) becomes a sum of P Gaussian integrals

$$\kappa_r^{rec}(t) = \sum_{p=1}^{P} \alpha_p \int \mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I}\, \mathrm{d}\underline{w}\; P_p(\underline{n}, \underline{m}, \underline{I}, \underline{w})\; n^{(r)}\; \phi\left(\sum_{r'=1}^{R} \kappa_{r'}(t) m^{(r')} + \sum_{s'=1}^{N_{in}} I_\perp^{(s')} v_{s'}(t) \right). \tag{23}$$

4.5 Effective dynamics of internal variables

In the following, we focus on zero-mean multivariate Gaussian mixture distributions for the loadings, and input patterns orthogonal to $\{\boldsymbol{m}^{(r)}\}_{r=1\dots R}$, as distributions with these assumptions were sufficient to describe trained networks. The more general case of Gaussian mixtures with non-zero means is treated in [Beiran et al., 2020]. Using Stein's lemma for Gaussian distributions, the dynamics of the internal collective variables can be expressed as a dynamical system (see SI section S1)

$$\frac{d\kappa_r}{dt} = -\kappa_r + \sum_{r'=1}^{R} \tilde{\sigma}_{n^{(r)}m^{(r')}}\kappa_{r'} + \sum_{s=1}^{N_{in}} \tilde{\sigma}_{n^{(r)}I^{(s)}}v_s. \tag{24}$$

For Eq. (3) in the main text, v_s were replaced by u_s which amounts to assume that inputs vary slowly with respect to the single neuron time constant τ.

In Eq. (24), $\tilde{\sigma}_{n^{(r)}m^{(r)}}$ represents the effective self-feedback of the collective variable κ_r, $\tilde{\sigma}_{n^{(r)}m^{(r')}}$ sets the interaction between the collective variables κ_r and $\kappa_{r'}$, and $\tilde{\sigma}_{n^{(r)}I^{(s)}}$ is the effective coupling between the input u_s and κ_r. These effective interactions between the internal variables are given by weighted averages over populations

$$\tilde{\sigma}_{ab} = \sum_{p=1}^{P} \alpha_p \sigma_{ab}^{(p)} \langle \Phi' \rangle_p \tag{25}$$

where $\sigma_{ab}^{(p)}$ is the covariance between loadings a and b for population p, and $\langle \Phi' \rangle_p$ is the average gain of population p, defined as

$$\langle \Phi' \rangle_p = \langle \Phi' \rangle(\Delta^{(p)}) \tag{26}$$

with

$$\langle \Phi' \rangle(\Delta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}z\, e^{-z^2/2} \phi'(\Delta z) \tag{27}$$

and

$$\Delta^{(p)} = \sqrt{\sum_{r'=1}^{R} (\sigma_{m^{(r')}}^{(p)})^2 \kappa_{r'}^2 + \sum_{s=1}^{N_{in}} (\sigma_{I^{(s)}}^{(p)})^2 v_s^2} \tag{28}$$

the standard deviation of activation variables in population p, where $\sigma_a^{(p)}$ is the variance of a pattern $\boldsymbol{a}$ on population p.

In Eq. (24), the covariances $\sigma_{ab}^{(p)}$ are set by the statistics of the hard-wired connectivity and input patterns, but the gain factors $\langle \Phi' \rangle_p$ depend on both the internal and external collective variables κ_k and v_j. As a consequence, the dynamics in Eq. (24) is non-linear, and in fact it can be shown that given a sufficient number of sub-populations, the right-hand side in Eq. (24) can approximate any arbitrary dynamical system [Beiran et al., 2020].

Eq. (24) shows that feed-forward inputs to the network can have two distinct effects on the collective dynamics of internal variables κ_r. If the input pattern s overlaps with the r-th input-selection pattern, i.e. the corresponding covariance $\sigma_{n^{(r)}I^{(s)}}^{(p)}$ is non-zero for population p, the input directly drives the dynamics of κ_r. In contrast, when all covariances between the input pattern s and the input selection patterns are zero (i.e. $\sigma_{n^{(r)}I^{(s)}}^{(p)} = 0$ for all r, p), the input can still modulate the dynamics by affecting the gain through Eq. (28) if the variance $\sigma_{I^{(s)}}^{(p)}$ of the input on some population p is non-zero. The inputs can therefore play roles of drivers and modulators of internal dynamics.

The values of the readouts (Eq. (20)) can also be expressed in terms of effective interactions

$$z_k = \sum_{r'=1}^{R} \tilde{\sigma}_{m^{(r')}w^{(k)}}\kappa_{r'} + \sum_{s=1}^{N_{in}} \tilde{\sigma}_{I^{(s)}w^{(k)}}v_s. \tag{29}$$

4.6 Network training procedure

We used backpropagation through time [Werbos, 1990] to train networks to minimize loss functions corresponding to specific tasks. For each task (see details below), we specified the temporal structure of trials and the desired mapping from stimuli u to target readouts $\hat{z}$, and then stochastically generated trials. We used the mean squared error loss function

$$\mathcal{L} = \sum_{k,i,t} M_t (z_{i,t}^{(k)} - \hat{z}_{i,t}^{(k)})^2 \tag{30}$$

where $z_{i,t}^{(k)}$ and $\hat{z}_{i,t}^{(k)}$ are respectively the actual, and the target readout values and the indices k, i, t respectively run over readout units, trials and time steps. The terms M_t are $\{0, 1\}$ masks that were non-zero only during a decision period at the end of trials, when the readouts were required to match their target values.

We specifically looked for solutions in the sub-space of rank R networks. The loss functions were therefore minimized by computing gradients with respect to the elements of connectivity patterns $\{\boldsymbol{m}^{(r)}\}_{r=1...R}, \{\boldsymbol{n}^{(r)}\}_{r=1...R}$. We didn't train the entries of input patterns $\{\boldsymbol{I}^{(s)}\}_{s=1...N_{in}}$ and readout patterns $\{\boldsymbol{w}^{(k)}\}_{k=1...N_{out}}$ but only an overall amplitude factor for each input and readout pattern (unless specified otherwise). All vectors were initialized with their entries drawn from Gaussian distributions with zero mean and unit standard deviation, except for the read-out vector, for which the standard deviation was 4. The initial network state at the beginning of each trial was always set to $\mathbf{0}$. We used the ADAM optimizer [Kingma and Ba, 2014] in pytorch [Paszke et al., 2017] with the decay rates of the first and second moments of 0.9 and 0.999, and learning rates between 10^{-3} and 10^{-2}.

To identify networks of minimal rank that performed each task, the number of connectivity patterns R was treated as a hyper-parameter. We first trained full rank networks ($R = N$) and determined the loss $\mathcal{L}_{R=N}$ with which they solved the task. We then started training rank $R = 5$ networks, and progressively decreased the rank until there was a sharp increase in trained loss (Supplementary Fig. S6). The minimal rank R^* was defined for each task such that $\mathcal{L}_{R^*} \simeq \mathcal{L}_{R=N}$ and $\mathcal{L}_{R^*-1} \gg \mathcal{L}_{R=N}$.

To ease the clustering and resampling procedure, and approach mean-field solutions, we trained large networks (of sizes 512 neurons for the perceptual DM and parametric WM tasks and 4096 neurons for the context-dependent DM and DMS tasks).

4.7 Clustering and resampling of trained networks

Following training, we approximated the obtained distributions of pattern loadings using Gaussian mixture models with zero-mean components, and then generated new networks by resampling from the obtained distributions. We specifically sought the smallest number of populations for which the network performed the task with optimal accuracy (defined for all tasks as the percentage of trials for which the signs of $z^{(k)}$ and $\hat{z}^{(k)}$ match).

For a given trained network, we first tried fitting a single multivariate Gaussian by computing the empirical covariance matrix of pattern loadings (matrix of size $(N_{in}+2R+N_{out})^2$). We then generated networks by resampling from this distribution, and if they were able to perform the task with optimal accuracy, concluded that the functionality was well explained by a single population. If not, we performed a clustering analysis in the loading space by progressively increasing the number of clusters until we found networks that were able to optimally perform the task. We used variational inference with a precision prior for the mean of 10^5 to enforce the zero-mean constraint, using the model `BayesianGaussianMixture` of the package scikit-learn [Pedregosa et al., 2011].

Since the inference and resampling processes are susceptible to finite-size fluctuations, we occasionally complemented the clustering with some retraining of the covariance matrices found for each component. For this we developed a class of trainable, Gaussian mixture, low-rank RNNs, in which the covariance structure of each population is trainable. Since directly training the covariance matrices is hard given that they need to be symmetric definite positive, we use a trick akin to the reparametrization trick used in variational auto-encoders [Kingma and Welling]: the set of input, connectivity and readout vectors are defined as a linear transformation of a basis of i.i.d. normal vectors, such that for any loading vector $\boldsymbol{a}$:

$$a_i = (\boldsymbol{b}_a^{(p)})^T \boldsymbol{X}_i, \tag{31}$$

where p is the population index of neuron i (sampled from a categorical distribution with weights $\alpha_{p_{p=1...P}}$ derived by the variational inference), $\boldsymbol{X}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbb{1})$ are random normal vectors of dimension $N_{in}+2R+N_{out}$, and the vectors $\boldsymbol{b}_a^{(p)}$ correspond to the rows of the Cholesky factorization of the covariance matrix (such that $\sigma_{ab}^{(p)} = (\boldsymbol{b}_a^{(p)})^T \boldsymbol{b}_b^{(p)}$ see SI section S1 for more details). We then trained the vectors $\boldsymbol{b}_v^{(p)}$, with the population indices being sampled only once, and the X_i being resampled at each training epoch.

The relevance of the clustering process has also been evaluated using a clustering stability measure [Hirokawa et al., 2019; Luxburg] (see SI figure S11). Formally, for a number of clusters varying from 2 to 5 we have generated 20 bootstrap subsamples containing each 80% of the neurons. The clustering algorithm is applied to each subsample and the obtained clusterings between each pair of subsamples are compared with the Adjusted Rand Index (ARI, value between 0 and 1, 1 indicating perfect agreement between clusterings, 0 indicating total randomness). For each number of clusters, a distribution of ARIs is thus obtained, for which a value consistently near 1 indicates that the corresponding number of clusters is relevant for this data.

4.8 Regression analysis

We used linear regression to relate single unit activities in networks performing the context-dependent DM task and the multi-sensory DM task to behavioral variables. For the context-dependent DM task we determined 5 regressors for each neuron, β_i^{ctxA}, β_i^{ctxB}, β_i^A, β_i^B and β_i^{choice}, while for the multi-sensory DM task only the 3 latter regressors were computed. The context regressors β_i^{ctxA} and β_i^{ctxB} were computed based on the activity during the context-only epoch, the sensory regressors β_i^A and β_i^B during the 200 first ms of the stimulation epoch, and the choice regressor β_i^{choice} during the decision epoch.
To isolate the effect of contextual inputs on neural activity and obtain the regression coefficients β_i^{ctxtA} and β_i^{ctxtB}, we fit the time-averaged neural activity $\phi(x_i)$ of neuron i obtained in 3 different contextual conditions: context A, context B (average over the first context-only time period), no context (average over the fixation period), using the linear regression model with intercept:

$$\boldsymbol{r}_i = X\beta_i + \epsilon_i. \tag{32}$$

Here $\boldsymbol{r}_i$ is a 3-by-1 vector containing the average firing rate of neuron i in the 3 conditions, X is the 3-by-3 design matrix, and β_i is a 3-by-1 vector of regression coefficients composed of β_i^{ctxA} and β_i^{ctxB} and $\beta_i^{(0)}$ (that we discard). This particular choice of separating the effect of context into two regressors was made to better retrieve the structure of input patterns to the network, as shown in figure 5.
For both the context-dependent and multi-sensory DM tasks, the selectivity to sensory inputs A and B are measured by regressing the neural activity during the first 200 ms of the stimulation period against the values of the coherence $\bar{u}_j^{(k)}$ for a set of 128 input conditions in the context-dependent task (8 values for each coherence and 2 contexts), 192 for the multi-sensory task (8 values for each coherence and 3 modalities).

4.9 Individual tasks

4.9.1 Perceptual decision making task

Trial structure. A fixation epoch of duration $T_{fix} = 100$ms was followed by a stimulation epoch of duration $T_{stim} = 800$ms, a delay epoch of duration $T_{delay} = 300$ms and a decision epoch of duration $T_{decision} = 300$ms.
Inputs and outputs. The feed-forward input to neuron i on trial k was

$$I_i^{FF}(t) = I_i u^{(k)}(t) \tag{33}$$

where, during the stimulation period, $u^{(k)}(t) = \overline{u}^{(k)} + \xi^{(k)}(t)$, with $\xi^{(k)}(t)$ a zero-mean Gaussian white noise with standard deviation $\sigma_u = 0.03$. The mean stimulus $\overline{u}^{(k)}$ was drawn uniformly from $\pm\frac{3.2}{100}\{1, 2, 4, 8, 16\}$ on each trial. The elements I_i of the input pattern were generated from a Gaussian distribution with zero mean and unit standard deviation, and fixed during training.
During the decision epoch, a single output z was evaluated through a readout pattern $\boldsymbol{w} = \{w_i\}_{i=1...N}$, the elements w_i of which were generated from a Gaussian distribution with zero mean and standard deviation of

4, and fixed during the training. On trial k, the target output value $\hat{z}^{(k)}$ in the loss function (Eq. (30)) was defined as the sign of the mean input $\overline{u}^{(k)}$.

Collective dynamics and reduced model. We found that computations in the rank one, single population trained networks could be reproduced by a reduced model with two non-zero covariances σ_{nI} and σ_{nm} (Supplementary Fig S1a). For this reduced model, the dynamics of the two internal collective variables is given by

$$\frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{nm}\kappa + \tilde{\sigma}_{nI}v(t), \tag{34}$$

where $\tilde{\sigma}_{nm} = \sigma_{nm}\langle\Phi'\rangle(\Delta)$ and $\tilde{\sigma}_{nI} = \sigma_{nI}\langle\Phi'\rangle(\Delta)$ with $\langle\Phi'\rangle(\Delta)$ defined in Eq. (26), and the effective population variance Δ given by:

$$\Delta = \sqrt{\sigma_m^2\kappa^2 + \sigma_I^2 v^2}. \tag{35}$$

Here $v(t)$ corresponds to the integrated input $u(t)$, see Eq. (16).

An analysis of nonlinear dynamics defined by Eq. (34) showed that adjusting these parameters was sufficient to implement the task, as additional parameters only modulate the overall gain (see SI section S2.1). In particular the value of σ_{mn}, determines the qualitative shape of the dynamical landscape on which the internal variable evolves and sets the time scale on which it integrates inputs (see SI S2.1 for more details).

4.9.2 Parametric working memory task

Trial structure. A fixation epoch of duration $T_{fix} = 100$ms was followed by a first stimulation epoch of duration $T_{stim1} = 100$ms, a delay epoch of duration T_{delay} drawn from a uniform distribution between 500 and 2000ms, a second stimulation epoch of duration $T_{stim2} = 100$ms and a decision epoch of duration $T_{decision} = 100$ms.

Inputs and outputs. The feed-forward input to neuron i on trial k was

$$I_i^{FF}(t) = I_i\left(u_1^{(k)}(t) + u_2^{(k)}(t)\right) \tag{36}$$

where $u_1^{(k)}(t)$ and $u_2^{(k)}(t)$ were non-zero during the first and second stimulation epochs respectively. On trial k and during the corresponding stimulation epoch, the values of these inputs were $u_{1,2}^{(k)} = \frac{1}{f_{max}-f_{min}}(f_{1,2}^{(k)} - \frac{f_{max}+f_{min}}{2})$, with $f_1^{(k)}$ and $f_2^{(k)}$ drawn uniformly from $\{10, 14, 18, 22, 26, 30, 34\}$, and $f_{min} = 10$ and $f_{max} = 34$. The elements I_i of the input pattern were generated from a Gaussian distribution with zero mean and unit standard deviation, and fixed during the training.

During the decision epoch, a single output z was evaluated through a readout pattern $\boldsymbol{w} = \{w_i\}_{i=1\ldots N}$, the elements w_i of which were generated from a Gaussian distribution with zero mean and standard deviation of 4, and fixed during the training. On trial k, the target output value $\hat{z}^{(k)}$ in the loss function (Eq. (30)) was defined as $\hat{z}^{(k)} = \frac{f_1^{(k)}-f_2^{(k)}}{f_{max}-f_{min}}$.

Collective dynamics and reduced model. We found that computations in the rank two, single population trained networks could be reproduced by a reduced model with four non-zero covariances $\sigma_{n^{(1)}m^{(1)}}$, $\sigma_{n^{(2)}m^{(2)}}$, $\sigma_{n^{(1)}I}$ and $\sigma_{n^{(2)}I}$ (Supplementary Fig. S2a). In particular covariances $\sigma_{n^{(1)}m^{(2)}}$, $\sigma_{n^{(2)}m^{(1)}}$ across the two patterns could be set to zero without performance impairment.

For this reduced model, the dynamics of the two internal collective variables is given by:

$$\begin{aligned}\frac{d\kappa_1}{dt} &= -\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(1)}I}v(t)\\ \frac{d\kappa_2}{dt} &= -\kappa_2 + \tilde{\sigma}_{n^{(2)}m^{(2)}}\kappa_2 + \tilde{\sigma}_{n^{(2)}I}v(t)\end{aligned} \tag{37}$$

where $\tilde{\sigma}_{ab} = \sigma_{ab}\langle\Phi'\rangle(\Delta)$, with $\langle\Phi'\rangle(\Delta)$ defined in Eq. (26), and the effective noise Δ given by:

$$\Delta = \sqrt{(\sigma_{m^{(1)}})^2\kappa_1^2 + (\sigma_{m^{(2)}})^2\kappa_2^2 + \sigma_I^2 v(t)^2}. \tag{38}$$

Here $v(t)$ corresponds to the integrated input $u(t)$, see Eq. (16).
The two internal collective variables are therefore effectively uncoupled, and integrate the incoming feed-forward inputs at two different timescales due to different levels of positive feedback. For the first collective variable, a strong, fine-tuned positive feedback $\sigma_{m^{(1)}n^{(1)}} \simeq 1$ leads to an approximate line attractor along κ_1 that persistently encodes the first stimulus throughout the delay and the sum of the two stimuli at the decision epoch. For the second internal variable, a weaker positive feedback $\sigma_{m^{(2)}n^{(2)}} \lesssim 1$ leads to a shorter timescale of a transient response to stimuli along κ_2, such that the first stimulus is forgotten during the delay and that the second stimulus is represented during the decision epoch.

4.9.3 Context-dependent decision making task

Trial structure. A fixation epoch of duration $T_{fix} = 100$ms was followed by a first context-only epoch of duration $T_{ctxt1} = 350$ms, followed by a stimulation epoch of duration $T_{stim} = 800$ms, a second context-only epoch of $T_{ctxt2} = 500$ms, and a decision epoch of $T_{decision} = 20$ms.
Stimuli and outputs. The feed-forward input to neuron i on trial k was

$$I_i^{FF}(t) = u_A^{(k)}(t) I_i^A + u_B^{(k)}(t) I_i^B + u_{ctxA}^{(k)}(t) I_i^{ctxA} + u_{ctxB}^{(k)}(t) I_i^{ctxB}. \tag{39}$$

Here $u_{ctxA}^{(k)}$ and $u_{ctxB}^{(k)}$ correspond to contextual cues. On each trial, during the context-only and the stimulation epochs, one of the two cues took a value +1, while the other was 0. The inputs $u_A^{(k)}(t)$ and $u_B^{(k)}(t)$ represent two sensory features of the stimulus. They were non-zero only during the stimulation epoch, and took the same form as in the perceptual decision-making task, with means $\overline{u}_A^{(k)}$ and $\overline{u}_B^{(k)}$, and fluctuating parts $\xi_A^{(k)}(t)$ and $\xi_B^{(k)}(t)$ drawn independently for each feature, on each trial. The elements of the input patterns were generated from a Gaussian distribution with zero mean and unit standard deviation on both populations. For the solution presented in the main text, only the contextual input patterns I_i^{ctxA} and I_i^{ctxB} had their entries trained, while for the other solution reported in supplementary section S2.3 all the input patterns were fixed throughout training. During the decision epoch, on trial k the target $\hat{z}^{(k)}$ in the loss function (Eq. (30)) was defined as the sign of the mean $\overline{u}_X^{(k)}$ of feature $X = A$ or B for which the contextual cue was activated, i. e. $u_{ctx}^{(k)} = 1$.

Collective dynamics and reduced model. We found that the computations in the unit rank, two populations network relied on the following conditions for the pattern covariances in the two populations (Supplementary Fig. S3a): (i) $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$ were essentially orthogonal to the input-selection pattern $\mathbf{n}$, implying that $\sigma^{(p)}_{nI^{ctxA}} \simeq 0$ and $\sigma^{(p)}_{nI^{ctxB}} \simeq 0$ for both populations $p = 1, 2$; (ii) each of the two input-selection sub-patterns was correlated with only one of the input-feature patterns, i.e. $\sigma^{(1)}_{nI^A} > 0$ and $\sigma^{(2)}_{nI^{(B)}} > 0$, while $\sigma^{(1)}_{nI^B} \approx 0$ and $\sigma^{(2)}_{nI^{(A)}} \approx 0$; (iii) each context-cue pattern had a strong variance on a different sub-population, i.e. for the first population $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$ had respectively weak and strong variance (i.e. $\sigma^{(1)}_{I^{ctxA}} \approx 0$ and $\sigma^{(1)}_{I^{ctxB}} > 1$), and conversely for the second population $\sigma^{(2)}_{I^{ctxA}} > 0$ and $\sigma^{(2)}_{I^{ctxB}} \approx 0$.
The computation could therefore be described by a reduced model, in which the covariances $\sigma^{(1)}_{nI^{(B)}}, \sigma^{(2)}_{nI^{(A)}} \sigma^{(2)}_{I^{ctxtB}}, \sigma^{(2)}_{I^{ctxtB}}$ were set to zero. The dynamics of the internal variable was then given by

$$\frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{nm}\kappa + \tilde{\sigma}_{nI^A} v_A(t) + \tilde{\sigma}_{nI^B} v_B(t) \tag{40}$$

with effective couplings

$$\tilde{\sigma}_{nI^A} = \frac{1}{2}\sigma^{(1)}_{nI^A}\langle\Phi'\rangle_1 \tag{41}$$

$$\tilde{\sigma}_{nI^B} = \frac{1}{2}\sigma^{(2)}_{nI^B}\langle\Phi'\rangle_2. \tag{42}$$

The averaged gains for each population were given by equations (27), with the standard deviations of currents onto each population

$$\Delta^{(1)} = \sqrt{(\sigma_m^{(1)})^2\kappa^2 + (\sigma_{I^A}^{(1)})^2 v_A^2 + (\sigma_{I^B}^{(1)})^2 v_B^2 + (\sigma_{I^{ctxB}}^{(1)})^2 c_B^2}$$
$$\Delta^{(2)} = \sqrt{(\sigma_m^{(2)})^2\kappa^2 + (\sigma_{I^A}^{(2)})^2 v_A^2 + (\sigma_{I^B}^{(2)})^2 v_B^2 + (\sigma_{I^{ctxA}}^{(2)})^2 c_A^2}. \tag{43}$$

Here $v_A(t)$ and $v_B(t)$ correspond to the integrated inputs $u_A(t)$ and $u_B(t)$, see Eq. (16).
As for the perceptual decision making task, the value of σ_{mn}, determines the qualitative shape of the dynamical landscape on which the internal variable evolves and sets the time scale on which it integrates inputs. Large values of the variances $\sigma_{I^{ctxtB}}^{(1)}$ and $\sigma_{I^{ctxtA}}^{(2)}$ allow the contextual cues to differentially vary the gain of the two populations in the two contexts, leading to an effective gating of the inputs integrated by the internal collective variable (see SI section S2.3 for more details).

4.9.4 Delay-match-to-sample task

Trial structure. A fixation epoch of duration $T_{fix} = 100$ms was followed by a first stimulus epoch of duration $T_{stim1} = 500$ms, a delay epoch of a duration drawn uniformly between 500ms and 3000ms, a second stimulus epoch of duration $T_{stim2} = 500$ms, and a decision epoch of duration $T_{decision} = 1000$ms.

Stimuli and outputs. During each stimulus epoch, the network received one of two stimuli A or B, which were randomly and independently chosen on each trial and stimulus epoch. These two stimuli were represented by two input patterns $\boldsymbol{I}^A$ and $\boldsymbol{I}^B$, so that the feed-forward input to neuron i on trial k was:

$$I_i^{FF}(t) = u_A^{(k)}(t) I_i^A + u_B^{(k)}(t) I_i^B \tag{44}$$

where the inputs $u_A^{(k)}(t)$ and $u_B^{(k)}(t)$ were non-zero only when stimuli A or B are respectively received, in which case they were equal to one.
During the decision epoch, the target output value $\hat{z}$ in the loss function (Eq. (30)) was equal to +1 if the same stimulus was received in both stimulation epochs and -1 otherwise.

Collective dynamics and reduced model. We found that the computations in the rank two, two populations network relied on the following conditions for the pattern covariances in the two populations (Supplementary Fig. S4a): (i) on one population, the two connectivity modes were coupled through $\sigma_{n^{(1)}m^{(2)}}^{(1)}, \sigma_{n^{(2)}m^{(1)}}^{(1)} \neq 0$, with a specific condition on their values to induce a limit cycle (that the difference $|\sigma_{n^{(1)}m^{(2)}}^{(1)} - \sigma_{n^{(2)}m^{(1)}}^{(1)}|$ is large, see SI and [Mastrogiuseppe and Ostojic, 2018; Beiran et al., 2020]); (ii) on the other population, the covariances were in contrast set to counter-balance the first population, and cancel the rotational dynamics $\sigma_{n^{(1)}m^{(2)}}^{(2)} \simeq -\sigma_{n^{(1)}m^{(2)}}^{(1)}$ and $\sigma_{n^{(2)}m^{(1)}}^{(2)} \simeq -\sigma_{n^{(2)}m^{(1)}}^{(1)}$; (iii) the input-selection and output patterns for the second connectivity mode on the second population had a strong overlap $\frac{1}{2}\sigma_{n^{(2)}m^{(2)}}^{(2)} > 1$ that led to strong positive feedback; (iv) the input patterns I^A has a strong variance on population 2, $\sigma_{I^A}^{(2)} \gg 1$ while other input sub-patterns have small variances $\sigma_{I^A}^{(1)}, \sigma_{I^B}^{(1)}, \sigma_{I^B}^{(2)} \simeq 0$.
For this reduced model, the dynamics of the two internal collective variables is given by:

$$\begin{aligned}
\frac{d\kappa_1}{dt} &= -\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(2)}}\kappa_2 \\
\frac{d\kappa_2}{dt} &= -\kappa_2 + \tilde{\sigma}_{n^{(2)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(2)}m^{(2)}}\kappa_2 + \tilde{\sigma}_{n^{(2)}I^A} v_A + \tilde{\sigma}_{n^{(2)}I^B} v_B,
\end{aligned} \tag{45}$$

with the effective couplings mediating inputs

$$\tilde{\sigma}_{n^{(2)}I^A} = \frac{1}{2}\sigma_{n^{(2)}I^A}^{(2)} \langle \Phi' \rangle_2 \tag{46}$$
$$\tilde{\sigma}_{n^{(2)}I^B} = \frac{1}{2}\sigma_{n^{(2)}I^B}^{(2)} \langle \Phi' \rangle_2, \tag{47}$$

and effective couplings governing the autonomous dynamics:

$$\tilde{\sigma}_{n^{(1)}m^{(1)}} = \frac{1}{2}\sigma^{(1)}_{n^{(1)}m^{(1)}}\langle\Phi'\rangle_1 \quad (48)$$

$$\tilde{\sigma}_{n^{(1)}m^{(2)}} = \frac{1}{2}\sigma^{(1)}_{n^{(1)}m^{(2)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(1)}m^{(2)}}\langle\Phi'\rangle_2 \quad (49)$$

$$\tilde{\sigma}_{n^{(2)}m^{(1)}} = \frac{1}{2}\sigma^{(1)}_{n^{(2)}m^{(1)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(2)}m^{(1)}}\langle\Phi'\rangle_2 \quad (50)$$

$$\tilde{\sigma}_{n^{(2)}m^{(2)}} = \frac{1}{2}\sigma^{(1)}_{n^{(2)}m^{(2)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(2)}m^{(2)}}\langle\Phi'\rangle_2. \quad (51)$$

The average gains are given by equations (27), with standard deviations of currents onto each population

$$\begin{aligned}\Delta^{(1)} &= \sqrt{(\sigma^{(1)}_{m^{(1)}})^2\kappa_1^2 + \sigma^{(1)}_{m^{(2)}})^2\kappa_2^2 + (\sigma^1_{I^A})^2 v_A^2} \\ \Delta^{(2)} &= \sqrt{(\sigma^{(2)}_{m^{(1)}})^2\kappa_1^2 + \sigma^{(2)}_{m^{(2)}})^2\kappa_2^2}.\end{aligned} \quad (52)$$

Here $v_A(t)$ and $v_B(t)$ correspond to the integrated inputs $u_A(t)$ and $u_B(t)$, see Eq. (16).
Conditions (i) to (iv) on the covariances allow to implement the dynamical landscape modulation of Fig. 4h (see Supplementary Fig. S4d). When stimulus A is present ($u_A = 1$), the gain of population 2 is set to $\langle\Phi'\rangle_2 \simeq 0$ because of $\sigma^{(2)}_{I^A} \gg 1$ (see Eq. (52)), and the specific values of covariances for sub-patterns in population 1 induce a limit cycle (see SI section S2.4). In absence of inputs, or when input B was present, gains were approximately equal for the two populations (Supplementary Fig. S4c), leading to a cancellation of the cross effective couplings $\tilde{\sigma}_{n^{(1)}m^{(2)}}$ and $\tilde{\sigma}_{n^{(2)}m^{(1)}}$, while positive feedback implemented through $\sigma^{(2)}_{n^{(2)}m^{(2)}}$ shaped a dynamical landscape with two fixed-points.

4.9.5 Multi-sensory decision making task

Trial structure. A fixation epoch of duration $T_{fix} = 100$ms was followed by a stimulation epoch of duration $T_{stim} = 800$ms, a delay epoch of duration $T_{delay} = 100$ms and a decision epoch of duration $T_{decision} = 500$ms.

Inputs and outputs. The feed-forward input to neuron i on trial k was given by:

$$I_i^{FF}(t) = I_i^A u_A^{(k)}(t) + I_i^B u_B^{(k)}(t) \quad (53)$$

where the two inputs signals $u_A^{(k)}(t)$ and $u_B^{(k)}(t)$ correspond to two sensory modalities, that provide congruent evidence for the output. Specifically a sign $s_k \in \{-1, 1\}$ is chosen for each trial, as well as a modality that can be A, B, or AB. Then if the modality is A or AB, a mean $\overline{u}_A^{(k)}$ is chosen from $\frac{1}{10}s_k\{1, 2, 3, 4, 8\}$ and the signal $u_A^{(k)}(t)$ during the stimulation period is set to that mean plus a gaussian white noise as in the perceptual decision making task. If the modality is B, then the signal $u_A^{(k)}(t)$ is only equal to the zero-centered gaussian white noise. The signal $u_B^{(k)}(t)$ is set in a similar manner. During the decision epoch, the target $\hat{z}^{(k)}$ is the underlying common sign s_k.

References

H. Adesnik, W. Bruns, H. Taniguchi, Z. J. Huang, and M. Scanziani. A neural circuit for spatial summation in visual cortex. *Nature*, 490(7419):226–231, 2012.

M. Aoi and J. W. Pillow. Model-based targeted dimensionality reduction for neuronal population data. In *Advances in neural information processing systems*, pages 6690–6699, 2018.

O. Barak. Recurrent neural networks as versatile tools of neuroscience research. *Current Opinion in Neurobiology*, 46:1–6, Oct 2017.

M. Beiran, A. Dubreuil, A. Valente, F. Mastrogiuseppe, and S. Ostojic. Shaping dynamics with multiple populations in low-rank recurrent networks, 2020.

D. V. Buonomano and W. Maass. State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113–125, 2009.

W. Chaisangmongkon, S. K. Swaminathan, D. J. Freedman, and X.-J. Wang. Computing by robust transience: How the fronto-parietal network performs sequential, category-based decisions. *Neuron*, 93(6):1504–1517.e4, Mar 2017.

M. M. Churchland and K. V. Shenoy. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *Journal of neurophysiology*, 97(6):4235–4257, 2007.

J. P. Cunningham and B. M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, Nov. 2014.

T. A. Engel and X.-J. Wang. Same or Different? A Neural Circuit Mechanism of Similarity-Based Pattern Match Decision Making. *Journal of Neuroscience*, 31(19):6982–6996, May 2011.

K. A. Ferguson and J. A. Cardin. Mechanisms underlying gain modulation in the cortex. *Nature Reviews Neuroscience*, Jan 2020.

S. Fusi, E. K. Miller, and M. Rigotti. Why neurons mix: high dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37:66–74, Apr 2016.

J. A. Gallego, M. G. Perich, L. E. Miller, and S. A. Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

J. I. Gold and M. N. Shadlen. The neural basis of decision making. *Annual review of neuroscience*, 30, 2007.

R. M. Haefner, S. Gerwinn, J. H. Macke, and M. Bethge. Inferring decoding strategies from choice probabilities in the presence of correlated variability. *Nature neuroscience*, 16(2):235, 2013.

B. Hangya, H.-J. Pi, D. Kvitsiani, S. P. Ranade, and A. Kepecs. From circuit motifs to computations: mapping the behavioral repertoire of cortical interneurons. *Current opinion in neurobiology*, 26:117–124, 2014.

K. Hardcastle, N. Maheswaranathan, S. Ganguli, and L. M. Giocomo. A multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex. *Neuron*, 94(2):375–387, 2017.

K. D. Harris and T. D. Mrsic-Flogel. Cortical connectivity and sensory coding. *Nature*, 503(7474):51–58, Nov 2013. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature12654.

J. Hirokawa, A. Vaughan, P. Masset, T. Ott, and A. Kepecs. Frontal cortex neuron types categorically encode single decision variables. *Nature*, Dec. 2019.

D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

D. B. Kastner, S. A. Baccus, and T. O. Sharpee. Critical and maximally informative encoding between neural populations in the retina. *Proceedings of the National Academy of Sciences*, 112(8):2533–2538, 2015.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. Kobak, W. Brendel, C. Constantinidis, C. E. Feierstein, A. Kepecs, Z. F. Mainen, X.-L. Qi, R. Romo, N. Uchida, and C. K. Machens. Demixed principal component analysis of neural population data. *eLife*, 5, Apr. 2016.

N. Kostantinos. Gaussian mixtures and their applications to signal processing. *Advanced signal processing handbook: theory and implementation for radar, sonar, and medical imaging real time systems*, pages 3–1, 2000.

D. Kvitsiani, S. Ranade, B. Hangya, H. Taniguchi, J. Huang, and A. Kepecs. Distinct behavioural and network correlates of two interneuron types in prefrontal cortex. *Nature*, 498(7454):363–366, 2013.

R. Legenstein and W. Maass. Edge of chaos and prediction of computational performance for neural circuit models. *Neural networks*, 20(3):323–334, 2007.

U. Von Luxburg. *Clustering stability: An overview*. Now Publishers Inc, 2010.

C. K. Machens, R. Romo, and C. D. Brody. Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex. *Journal of Neuroscience*, 30(1):350–360, 2010.

V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, Nov. 2013.

F. Mastrogiuseppe and S. Ostojic. Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks. *Neuron*, July 2018.

Y. Miyashita. Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature*, 335(6193):817–820, 1988.

E. I. Moser, M.-B. Moser, and B. L. McNaughton. Spatial representation in the hippocampal formation: a history. *Nature neuroscience*, 20(11):1448, 2017.

S. Ocko, J. Lindsey, S. Ganguli, and S. Deny. The emergence of multiple retinal cell types through efficient coding of natural movies. In *Advances in Neural Information Processing Systems*, pages 9389–9400, 2018.

I. M. Park, M. L. Meister, A. C. Huk, and J. W. Pillow. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature neuroscience*, 17(10):1395, 2014.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

S. Peron, R. Pancholi, B. Voelcker, J. D. Wittenbach, H. F. Ólafsdóttir, J. Freeman, and K. Svoboda. Recurrent interactions in local cortical circuits. *Nature*, pages 1–4, 2020.

L. Pinto and Y. Dan. Cell-type-specific activity in prefrontal cortex during goal-directed behavior. *Neuron*, 87 (2):437–450, 2015.

N. C. Rabinowitz, R. L. Goris, M. Cohen, and E. P. Simoncelli. Attention stabilizes the shared gain of v4 populations. *Elife*, 4:e08998, 2015.

K. Rajan, C. D. Harvey, and D. W. Tank. Recurrent network models of sequence generation and memory. *Neuron*, 90(1):128–142, 2016.

D. Raposo, M. T. Kaufman, and A. K. Churchland. A category-free neural population supports evolving demands during decision-making. *Nature Neuroscience*, 17(12):1784–1792, Dec. 2014.

E. D. Remington, D. Narain, E. A. Hosseini, and M. Jazayeri. Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics. *Neuron*, 98(5):1005–1019.e5, June 2018.

M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, May 2013.

R. Romo, C. D. Brody, A. Hernández, and L. Lemus. Neuronal correlates of parametric working memory in the prefrontal cortex. *Nature*, 399(6735):470–473, 1999.

K. Sakai. Task set and prefrontal cortex. *Annu. Rev. Neurosci.*, 31:219–245, 2008.

E. Salinas and P. Thier. Gain modulation: a major computational principle of the central nervous system. *Neuron*, 27(1):15–21, 2000.

S. Saxena and J. P. Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.

L. I. Schmitt, R. D. Wimmer, M. Nakajima, M. Happ, S. Mofakham, and M. M. Halassa. Thalamic amplification of cortical connectivity sustains attentional control. *Nature*, 545(7653):219–223, May 2017.

F. Schuessler, A. Dubreuil, F. Mastrogiuseppe, S. Ostojic, and O. Barak. Dynamics of random recurrent networks with correlated low-rank structure. *Physical Review Research*, 2(1):013111, 2020a.

F. Schuessler, F. Mastrogiuseppe, A. Dubreuil, S. Ostojic, and O. Barak. The interplay between randomness and structure during learning in rnns, 2020b.

A. Sederberg and I. Nemenman. Randomly connected networks generate emergent selectivity and predict decoding properties of large populations of neurons. *PLOS Computational Biology*, 16(5):e1007875, 2020.

H. S. Seung. How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences of the United States of America*, 93(23):13339–13344, Nov 1996.

S. M. Sherman and R. Guillery. On the actions that one nerve cell can have on another: distinguishing "drivers" from "modulators". *Proceedings of the National Academy of Sciences*, 95(12):7121–7126, 1998.

H. Sohn, D. Narain, N. Meirhaeghe, and M. Jazayeri. Bayesian computation through cortical latent dynamics. *Neuron*, 103(5):934–947, 2019.

J. P. Stroud, M. A. Porter, G. Hennequin, and T. P. Vogels. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature neuroscience*, 21(12):1774–1783, 2018.

D. Sussillo. Neural circuits as computational dynamical systems. *Current Opinion in Neurobiology*, 25:156–163, Apr 2014.

D. Sussillo and O. Barak. Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks. *Neural Computation*, 25(3):626–649, Mar. 2013.

H. Tanaka, A. Nayebi, N. Maheswaranathan, L. McIntosh, S. Baccus, and S. Ganguli. From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction. In *Advances in Neural Information Processing Systems*, pages 8537–8547, 2019.

J. Wang, D. Narain, E. A. Hosseini, and M. Jazayeri. Flexible timing by temporal scaling of cortical responses. *Nature neuroscience*, 21(1):102–110, 2018.

P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10): 1550–1560, 1990.

G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, and X.-J. Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, Jan. 2019.

L. Ye, W. E. Allen, K. R. Thompson, Q. Tian, B. Hsueh, C. Ramakrishnan, A.-C. Wang, J. H. Jennings, A. Adhikari, C. H. Halpern, et al. Wiring and molecular features of prefrontal ensembles representing distinct experiences. *Cell*, 165(7):1776–1788, 2016.

P. Znamenskiy and A. M. Zador. Corticostriatal neurons in auditory cortex drive decisions during auditory discrimination. *Nature*, 497(7450):482–485, 2013.

Supplementary information

S1 Parametrization and collective dynamics for mixture of Gaussians loadings

In this section we show how connectivity patterns with loadings drawn from mixtures of multivariate Gaussians can be constructed from independent Gaussians, as mentionned in Eq. (31). We then derive the dynamics of the internal collective variables (Eq. (24)) in this setting.

We considered distributions of loadings characterized by P covariance matrices $\mathbf{\Sigma}_p$, and zero means $\boldsymbol{\mu}_p = \mathbf{0}$, $p = 1, \dots, P$. For a neuron i belonging to population p, each pattern loading $a_i \in \{n_i^{(r)}, m_i^{(r)}, I_i^{(s)}, w_i^{(q)}\}$ is constructed as a linear transformation of the same set of values $\{X_i^{(d)}\}_{d=1\dots N_{in}+2R+N_{out}}$

$$a_i = \sum_{d=1}^{N_{in}+2R+N_{out}} b_{a,d}^{(p)} X_i^{(d)}. \tag{S1}$$

Here the $X_i^{(d)}$ are drawn from $\mathcal{N}(0,1)$, independently for each i and d. The linear coefficients $\{b_{a,d}\}_{d=1\dots N_{in}+2R+N_{out}}$ are different for each pattern $a \in \{n^{(r)}, m^{(r)}, I^{(s)}, w^{(q)}\}$, but identical across neurons within a given population. These sets of coefficients therefore determine the covariance $\sigma_{ab}^{(p)}$ between pattern loadings within a given population p:

$$\sigma_{ab}^{(p)} = \sum_{d=1}^{D} b_{a,d}^{(p)} b_{b,d}^{(p)} \tag{S2}$$

The row-vectors $\boldsymbol{b_a^{(p)}}$ in fact correspond to the rows of the Cholesky factorization of the covariance matrix.
We next turn to the derivation of Eq. (24). With the parametrization of pattern loadigns defined in Eq. (S1), the recurrent inputs to the r-th internal collective variable Eq. (19) can be written as

$$\kappa_r^{rec} = \sum_{p=1}^{P} \alpha_p \int \left(\prod_{d=1}^{D} DX^{(d)}\right) \sum_{d=1}^{D} b_{n^{(r)},d}^{(p)} X^{(d)} \phi\left(\sum_{r'=1}^{R} \kappa_{r'} \sum_{d=1}^{D} b_{m^{(r')},d}^{(p)} X^{(d)} + \sum_{s=1}^{N_{in}} v_s \sum_{d=1}^{D} b_{I^{(s)},d}^{(p)} X^{(d)}\right) \tag{S3}$$

with $DX^{(d)} = \frac{dX^{(d)}}{\sqrt{2\pi}} e^{-(X^{(d)})^2/2}$. For a given p, we then compute each of the D integrals $\int \left(\prod_{d=1}^{D} DX^{(d)}\right) b_{n^{(r)},d}^{(p)} X^{(d)} \phi(\dots)$ applying successively Stein's lemma

$$\int Dz\, z f(z) = \int Dz f'(z), \tag{S4}$$

and using the fact that a sum of Gaussians is a Gaussian with variance given by the sum of variances, so that

$$\int DxDy \dots f(\alpha x + \beta y + \dots) = \int Dz f(\sqrt{\alpha^2 + \beta^2 + \dots} z). \tag{S5}$$

This leads to

$$\begin{aligned} \kappa_r^{rec} &= \sum_{p=1}^{P} \alpha_p \sum_{d=1}^{D} b_{n^{(r)},d}^{(p)} \left(\sum_{r'=1}^{R} b_{m^{(r')},d}^{(p)} \kappa_{r'} + \sum_{s=1}^{N_{in}} b_{I^{(s)},d}^{(p)} v_s\right) \int Dz \phi'\left(\Delta^{(p)} z\right) \\ &= \sum_{p=1}^{P} \alpha_p \left(\sum_{r'=1}^{R} \sigma_{n^{(r)} m^{(r')}}^{(p)} \kappa_{r'} + \sum_{s=1}^{N_{in}} \sigma_{n^{(r)} I^{(s)}}^{(p)} v_s\right) \int Dz \phi'\left(\Delta^{(p)} z\right) \end{aligned} \tag{S6}$$

with

$$\Delta^{(p)} = \sqrt{\sum_{r'=1}^{R} (\sigma^{(p)}_{m^{(r')}m^{(r')}})^2 \kappa_{r'}^2 + \sum_{s=1}^{N_{in}} (\sigma^{(p)}_{I^{(s)}I^{(s)}})^2 u_s^2}. \tag{S7}$$

Inverting the sums on p and r', s indices and assuming that input patterns $\boldsymbol{I}^{(s)}$ are orthogonal to the output patterns $\{\boldsymbol{m}^{(r)}\}_{r=1,\dots,R}$ (as in all the reduced models described in the section below), we get the compact description in terms of effective couplings for the dynamics of internal collective variables Eq. (24)

$$\frac{d\kappa_r}{dt} = -\kappa_r + \sum_{r'=1}^{R} \tilde{\sigma}_{n^{(r)}m^{(r')}} \kappa_{r'} + \sum_{s=1}^{N_{in}} \tilde{\sigma}_{n^{(r)}I^{(s)}} v_s \tag{S8}$$

with, for any two patterns $\boldsymbol{a}, \boldsymbol{b}$, the effective couplings

$$\tilde{\sigma}_{ab} = \sum_{p=1}^{P} \sigma^{(p)}_{ab} \langle \Phi' \rangle_p \tag{S9}$$

and averaged gains

$$\langle \Phi' \rangle_p = \int Dz \phi'(\Delta^{(p)} z). \tag{S10}$$

S2 Theoretical analysis of reduced models

Here we examine reduced network models, that were minimally parametrized to solve each task by relying on the same network dynamics as the trained networks presented in the main text. The minimal parameter sets correspond to subsets of covariances between input and connectivity patterns (outlined in Figs. 2b,3b,4b,5b). These parameters were first set by hand and then, if necessary fine-tuned with the ADAM optimizer to solve the task with optimal accuracy (see Methods section 4.6). We first report how to parametrize input and connectivity patterns to build these networks. We then examine the effects of these parameters on mean-field collective dynamics and show their implication in task solving.

S2.1 Perceptual decision-making network

The network trained on this task was of unit rank, and consisted of a single population. Such a network can be minimally parametrized using three covariances σ_{nm}, σ_{nI} and σ_{mw} (Fig.S1a). This can be obtained with an input pattern $I_i = X_i^{(1)}$ and connectivity patterns given by:

$$\begin{aligned} n_i &= \sigma_{nI} X_i^{(1)} + \sqrt{\sigma_{nm}} X_i^{(2)} \\ m_i &= \sqrt{\sigma_{nm}} X_i^{(2)} + \sqrt{\sigma_{mm} - \sigma_{nm}} X_i^{(3)} \end{aligned} \tag{S11}$$

for $i = 1, \dots, N$, with $X_i^{(d)}$'s drawn independently from zero-mean Gaussian distributions of unit variance. The readout components were taken as

$$w_i = \frac{\sigma_{mw}}{\sqrt{\sigma_{mm} - \sigma_{nm}}} X_i^{(3)}. \tag{S12}$$

The dynamics of the single internal collective variable is then given by

$$\frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{mn}\kappa + \tilde{\sigma}_{nI} v(t) \tag{S13}$$

with effective couplings given by equation (S9), i.e. the covariances scaled by the global gain factor

$$\langle \Phi' \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz e^{-z^2/2} \phi'(\sqrt{\sigma_{mm}\kappa^2 + \sigma_{II} v^2} z) \tag{S14}$$

This dynamics can be graphically summarized as in Fig.2c and leads to network dynamics that matches the one of trained networks (Fig.S1b).

The autonomous dynamics of the network is determined by the parameter σ_{nm} that controls (i) the qualitative shape of the dynamical landscape, with a transition from a single stable fixed-point ($\sigma_{nm} < 1$) to two symmetric fixed-points ($\sigma_{nm} > 1$) and (ii) the time-scale $\tau_{rec} = \frac{1}{|1-\sigma_{nm}|}$ with which the network state relaxes or diverges from the initial condition $\boldsymbol{x} = \mathbf{0}$ at the beginning of each trial (Fig.S1c,d, [Mastrogiuseppe and Ostojic, 2018]).
The integration of the filtered input $v(t)$ by κ is controlled by σ_{nI}, the covariance between the input pattern $\boldsymbol{I}$ and the input-selection pattern $\boldsymbol{n}$. For instance for $\sigma_{nI} = 0$, $v(t)$ is projected on a direction orthogonal to the input-selection pattern and is not integrated by the recurrent activity (Fig. S1g light shade line).
Finally, the covariance σ_{mw} between the output pattern $\boldsymbol{m}$ and the readout pattern $\boldsymbol{w}$ controls the extent to which the readout is driven by κ, with no drive of the readout in case of orthogonal output pattern and readout pattern, $\sigma_{mw} = 0$ (Fig. S1f light shade line).
The network connectivity of equation (S11), also involved the variance of the individual connectivity pattern σ_{mm}. Changing σ_{mm} influences the autonomous dynamics of the network (Fig. S1c) by influencing the gain of neurons (see Eq. (S18)).
For the reconstructed network shown in the main text, the non-zero covariances were: $\sigma_{nm} = 1.4, \sigma_{nI} = 2.6$ and $\sigma_{mw} = 2.1$.

S2.2 Parametric working-memory network

The network trained on this task was of rank two, and consisted of a single population. A minimal parametrization of this network involves four covariances $\sigma_{n^{(1)}I}$, $\sigma_{n^{(1)}m^{(1)}}$, $\sigma_{n^{(2)}I}$ and $\sigma_{n^{(2)}m^{(2)}}$ (Fig. S2a). This can be obtained with an input pattern $I_i = X_i^{(1)}$ and two connectivity modes:

$$\begin{aligned} n_i^{(1)} &= \sigma_{n^{(1)}I} X_i^{(1)} + \sqrt{\sigma_{n^{(1)}m^{(1)}}} X_i^{(2)} \\ m_i^{(1)} &= \sqrt{\sigma_{n^{(1)}m^{(1)}}} X_i^{(2)} + \sqrt{\sigma_{m^{(1)}m^{(1)}} - \sigma_{n^{(1)}m^{(1)}}} X_i^{(3)} \\ n_i^{(2)} &= \sigma_{n^{(2)}I} X_i^{(1)} + \sqrt{\sigma_{n^{(2)}m^{(2)}}} X_i^{(4)} \\ m_i^{(1)} &= \sqrt{\sigma_{n^{(2)}m^{(2)}}} X_i^{(4)} + \sqrt{\sigma_{m^{(2)}m^{(2)}} - \sigma_{n^{(2)}m^{(2)}}} X_i^{(5)} \end{aligned} \tag{S15}$$

for $i = 1, \ldots, N$, with $X_i^{(a)}$'s drawn from zero-mean Gaussian distributions of unit variance. The readout components were taken as

$$w_i = \frac{\sigma_{m^{(1)}w}}{\sqrt{\sigma_{m^{(1)}m^{(1)}} - \sigma_{n^{(1)}m^{(1)}}}} X_i^{(3)} + \frac{\sigma_{m^{(2)}w}}{\sqrt{\sigma_{m^{(2)}m^{(2)}} - \sigma_{n^{(2)}m^{(2)}}}} X_i^{(5)}. \tag{S16}$$

The dynamics of the two internal collective variables is then given by:

$$\begin{aligned} \frac{d\kappa_1}{dt} &= -\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(1)}} \kappa_1 + \tilde{\sigma}_{n^{(1)}I} v(t) \\ \frac{d\kappa_2}{dt} &= -\kappa_2 + \tilde{\sigma}_{n^{(2)}m^{(2)}} \kappa_2 + \tilde{\sigma}_{n^{(2)}I} v(t) \end{aligned} \tag{S17}$$

with effective couplings given by equation (S9), i.e. the covariances scaled by the global gain factor

$$\langle \Phi' \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz e^{-z^2/2} \phi'(\Delta z) \tag{S18}$$

with

$$\Delta = \sqrt{\sigma_{m^{(1)}m^{(1)}} \kappa_1^2 + \sigma_{m^{(2)}m^{(2)}} \kappa_2^2 + \sigma_{II} v^2} \tag{S19}$$

This dynamics can be graphically summarized as in Fig. 3e and reproduces the dynamics of trained networks as shown in Fig.S2b. Fig.S2c shows the dynamical phase portrait on which recurrent activity evolves. It

approximates a line attractor [Seung, 1996] on the direction $\boldsymbol{m}^{(1)}$ as the covariance $\sigma_{n^{(1)}m^{(1)}} \simeq 1$ sets the network close to the bifurcation point of Fig.S1c. On the second direction $\boldsymbol{m}^{(2)}$ the dynamics relaxes with a time scale set by the covariance $\sigma_{n^{(2)}m^{(2)}}$. For the reconstructed network shown in the main text, the non-zero covariances were: $\sigma_{n^{(1)}m^{(1)}} = 1.0, \sigma_{n^{(2)}m^{(2)}} = 0.5, \sigma_{n^{(1)}I} = 0.5, \sigma_{n^{(2)}I} = 1.9, \sigma_{m^{(1)}w} = 2.8$ and $\sigma_{m^{(2)}w} = -2.2$.

S2.3 Context-dependent decision-making network

The networks trained on this task were of unit rank and consisted of either two or three populations depending on the training procedure (see methods section 4.9.3 and supplementary section S3).

Two-population network Such a network can be minimally parametrized using 4 non-zero variances/covariances on each population. This can be obtained with the two sensory input patterns generated independently $I_i^A = X_i^{(1)}$, $I_i^B = X_i^{(2)}$, irrespective of the population structure. The connectivity patterns are structured in two sub-patterns. For i in population 1:

$$\begin{aligned} n_i &= \sigma^{(1)}_{nI^A} X_i^{(1)} + \sqrt{\sigma^{(1)}_{nm}} X_i^{(3)} \\ m_i &= \sqrt{\sigma^{(1)}_{nm}} X_i^{(3)} + \sqrt{\sigma^{(1)}_{mm} - \sigma^{(1)}_{nm}} X_i^{(4)} \end{aligned} \tag{S20}$$

while for i in population 2:

$$\begin{aligned} n_i &= \sigma^{(2)}_{nI^B} X_i^{(2)} + \sqrt{\sigma^{(2)}_{nm}} X_i^{(3)} \\ m_i &= \sqrt{\sigma^{(2)}_{nm}} X_i^{(3)} + \sqrt{\sigma^{(2)}_{mm} - \sigma^{(2)}_{nm}} X_i^{(4)} \end{aligned} \tag{S21}$$

with $X_i^{(a)}$'s drawn from centered Gaussian distributions of unit variance. The readout pattern is taken as

$$w_i = \frac{\sigma^{(1)}_{mw}}{\sqrt{\sigma^{(1)}_{mm} - \sigma^{(1)}_{nm}}} X_i^{(4)} \tag{S22}$$

for i in population 1 and

$$\frac{\sigma^{(2)}_{mw}}{\sqrt{\sigma^{(2)}_{mm} - \sigma^{(2)}_{nm}}} X_i^{(4)} \tag{S23}$$

for i in population 2. Importantly the contextual input patterns are also structured in two sub-patterns, such that for i in population 1:

$$\begin{aligned} I_i^{ctxA} &= 0 \\ I_i^{ctxB} &= \sigma^{(1)}_{I^{ctxB} I^{ctxB}} X_i^{(5)} \end{aligned} \tag{S24}$$

while for i in population 2:

$$\begin{aligned} I_i^{ctxA} &= \sigma^{(2)}_{I^{ctxA} I^{ctxA}} X_i^{(5)} \\ I_i^{ctxB} &= 0 \end{aligned} \tag{S25}$$

with $\sigma^{(1)}_{I^{ctxB} I^{ctxB}}, \sigma^{(2)}_{I^{ctxA} I^{ctxA}} \gg 1$.

The recurrent activity is then described by a single internal collective variable, graphically summarized in Fig.4f:

$$\frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{mn}\kappa + \tilde{\sigma}_{nI^A} v_A(t) + \tilde{\sigma}_{nI^B} v_B(t) \tag{S26}$$

The time evolution of the internal collective variable is coupled to the two inputs through the two effective couplings $\tilde{\sigma}_{nI^A}, \tilde{\sigma}_{nI^B}$, each supported by one of the two populations:

$$\tilde{\sigma}_{nI^A} = \frac{1}{2}\sigma^{(1)}_{nI^A}\langle\Phi'\rangle_1 \tag{S27}$$

$$\tilde{\sigma}_{nI^B} = \frac{1}{2}\sigma^{(2)}_{nI^B}\langle\Phi'\rangle_2 \tag{S28}$$

The recurrent dynamics is supported equally by the two populations:

$$\tilde{\sigma}_{nm} = \frac{1}{2}\sigma^{(1)}_{nm}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{nm}\langle\Phi'\rangle_2 \tag{S29}$$

with averaged gains given by equations (S10) and standard deviations of currents onto each population

$$\begin{aligned}\Delta^1 &= \sqrt{(\sigma^{(1)}_{mm})^2\kappa^2 + (\sigma^{(1)}_{I^AI^A})^2 v_A^2 + (\sigma^{(1)}_{I^BI^B})^2 v_B^2 + (\sigma^{(1)}_{I^{ctxtB}I^{ctxtB}})^2 c_B^2} \\ \Delta^2 &= \sqrt{(\sigma^{(2)}_{mm})^2\kappa^2 + (\sigma^{(2)}_{I^AI^A})^2 v_A^2 + (\sigma^{(2)}_{I^BI^B})^2 v_B^2 + (\sigma^{(2)}_{I^{ctxtA}I^{ctxtA}})^2 c_A^2}.\end{aligned} \tag{S30}$$

The obtained dynamics is similar to the trained networks displayed in Fig. 4d,e, with contextual inputs controlling the gain of each of the two populations (Fig.S3b). This control relies on the large amplitude of the weights of contextual input patterns, $\sigma^{(2)}_{I^{ctxtA}I^{ctxtA}}, \sigma^{(1)}_{I^{ctxtB}I^{ctxtB}} \gg 1$, as illustrated in Fig.S3c where we show the effect of varying these parameters on the network readout. In this implementation, each of the two populations selectively integrates one of the two sensory inputs thanks to the non-zero covariances between input pattern and input-selection modes $\sigma^{(1)}_{nI^A}, \sigma^{(2)}_{nI^B}$, as illustrated in Fig.S3d.
The non-zero covariances for the implementation of the solution presented in the main text are given by $\sigma^{(1)}_{nm} = 2.2, \sigma^{(2)}_{nm} = 2.3, \sigma^{(1)}_{nI^A} = 2.9, \sigma^{(2)}_{nI^B} = 3.1, \sigma^{(1)}_{mw} = 4.6, \sigma^{(2)}_{mw} = 5.0, \sigma^{(2)}_{I^{ctxtA}I^{ctxtA}} = 100, \sigma^{(1)}_{I^{ctxtB}I^{ctxtB}} = 100$.

Three-population network For the context-dependent decision-making task, we also examined a network relying on three populations. In this network, two populations selectively gate inputs as in the two-population network, but the recurrent interactions that implement evidence integration are segregated to a third population. Here we describe the corresponding reduced model.
As for the two-population network, the two sensory input patterns are generated independently $I_i^A = X_i^{(1)}$, $I_i^B = X_i^{(2)}$, irrespective of the population structure. The connectivity mode is structured in three sub-populations. For i in population 1:

$$\begin{aligned} n_i &= \sigma^{(1)}_{nI^A} X_i^{(1)} \\ m_i &= 0 \end{aligned} \tag{S31}$$

for i in population 2:

$$\begin{aligned} n_i &= \sigma^{(2)}_{nI^B} X_i^{(2)} \\ m_i &= 0 \end{aligned} \tag{S32}$$

and for i in population 3:

$$\begin{aligned} n_i &= \sqrt{\sigma^{(3)}_{nm}} X_i^{(3)} \\ m_i &= \sqrt{\sigma^{(3)}_{nm}} X_i^{(3)} + \sqrt{\sigma^{(3)}_{mm} - \sigma^{(3)}_{nm}} X_i^{(4)} \end{aligned} \tag{S33}$$

for $i = 1, \ldots, N$, with $X_i^{(a)}$'s drawn independently from centered Gaussian distributions of unit variance. The readout pattern reads only from the third population:

$$w_i = \frac{\sigma^{(3)}_{mw}}{\sqrt{\sigma^{(3)}_{mm} - \sigma^{(3)}_{nm}}} X_i^{(4)} \tag{S34}$$

The contextual inputs are the same as in the two-population network. The overall expression for the time evolution of the internal collective variable is unchanged compared to the two populations solution Eq. (S26). Each of the effective couplings between κ and inputs is supported by one of two populations

$$\tilde{\sigma}_{nI^A} = \frac{1}{3}\sigma^{(1)}_{nI^A}\langle\Phi'\rangle_1 \tag{S35}$$

$$\tilde{\sigma}_{nI^B} = \frac{1}{3}\sigma^{(2)}_{nI^B}\langle\Phi'\rangle_2 \tag{S36}$$

and the self-coupling of the internal collective variable is supported by the third population

$$\tilde{\sigma}_{nm} = \frac{1}{3}\sigma^{(3)}_{nm}\langle\Phi'\rangle_3 \tag{S37}$$

with averaged gains given by equations (S10) and standard deviations of currents onto each population by

$$\begin{aligned}
\Delta^1 &= \sqrt{(\sigma^{(1)}_{I^AI^A})^2 v_A^2 + (\sigma^{(1)}_{I^BI^B})^2 v_B^2 + (\sigma^{(1)}_{I^{ctxtB}I^{ctxtB}})^2 c_B^2} \\
\Delta^2 &= \sqrt{(\sigma^{(2)}_{I^AI^A})^2 v_A^2 + (\sigma^{(2)}_{I^BI^B})^2 v_B^2 + (\sigma^{(2)}_{I^{ctxtA}I^{ctxtA}})^2 c_A^2} \\
\Delta^3 &= \sqrt{(\sigma^{(3)}_{mm})^2 \kappa^2 + (\sigma^{(2)}_{I^AI^A})^2 v_A^2 + (\sigma^{(2)}_{I^BI^B})^2 v_B^2}
\end{aligned} \tag{S38}$$

In this three-population implementation, the contextual inputs do not control the gains of neurons in the third population and thus modulate only the effective couplings that mediate the influence of sensory inputs. The non-zero covariances for an implementation of this solution are given by $\sigma^{(3)}_{nm} = 3.6, \sigma^{(1)}_{nI^A} = 3.1, \sigma^{(2)}_{nI^B} = 2.8, \sigma^{(3)}_{mw} = 9.8, \sigma^{(2)}_{I^{ctxtA}I^{ctxtA}} = 100, \sigma^{(1)}_{I^{ctxtB}I^{ctxtB}} = 100$.

S2.4 Delay-match-to-sample network

Networks trained on this task were of rank two and consisted of two populations. Here we propose a minimally parametrized network (Fig. S4a) that, similarly to the trained network presented in the main text, relies on the ability of inputs to control the autonomous dynamics of the network. The connectivity modes defined on the first population are coupled to each other through covariances $\sigma^{(1)}_{n^{(1)}m^{(2)}}$ and $\sigma^{(1)}_{n^{(2)}m^{(2)}}$ [Mastrogiuseppe and Ostojic, 2018; Beiran et al., 2020]:

$$\begin{aligned}
n_i^{(1)} &= \sigma^{(1)}_{n^{(1)}m^{(1)}} X_i^{(1)} + \sigma^{(1)}_{n^{(1)}m^{(2)}} X_i^{(2)} \\
m_i^{(1)} &= X_i^{(1)} + X_i^{(3)} \\
n_i^{(2)} &= \sigma^{(1)}_{n^{(2)}m^{(1)}} X_i^{(3)} + \sigma^{(1)}_{n^{(2)}m^{(2)}} X_i^{(4)} \\
m_i^{(2)} &= X_i^{(2)} + X_i^{(4)}
\end{aligned} \tag{S39}$$

with covariances chosen such that the trivial fixed-points $\boldsymbol{x} = \mathbf{0}$ is an unstable spiral point, and the dynamics defined by the first sub-population generate a limit cycle. As shown by a linear stability analysis of the dynamical equation for internal collective variables, this dynamical feature arises when the covariances are such that the following matrix has complex eigenvalues with positive real-parts [Mastrogiuseppe and Ostojic, 2018; Beiran et al., 2020]

$$\boldsymbol{J} = \begin{pmatrix} \sigma^{(1)}_{n^{(1)}m^{(1)}} - 1 & \sigma^{(1)}_{n^{(1)}m^{(2)}} \\ \sigma^{(1)}_{n^{(2)}m^{(1)}} & \sigma^{(1)}_{n^{(2)}m^{(2)}} - 1 \end{pmatrix}. \tag{S40}$$

This first population is coupled to a second population which, in the absence of inputs, cancels the rotational dynamics, through the relationships $\sigma^{(1)}_{n^{(1)}m^{(2)}} = -\sigma^{(2)}_{n^{(1)}m^{(2)}}$ and $\sigma^{(1)}_{n^{(2)}m^{(1)}} = -\sigma^{(2)}_{n^{(2)}m^{(1)}}$. The second population also implements a pair of fixed-points that will be used to store the identity of the first stimulus throughout the delay and report the match/non-match decision. The connectivity sub-pattern on the second population can then be written as:

$$\begin{aligned} n_i^{(1)} &= \sigma^{(2)}_{n^{(1)}m^{(2)}} X_i^{(2)} \\ m_i^{(1)} &= X_i^{(3)} \\ n_i^{(2)} &= \sigma^{(2)}_{n^{(2)}I^A} X_i^{(5)} - |\sigma^{(2)}_{n^{(1)}I^B}| X_i^{(6)} + \sigma^{(2)}_{n^{(2)}m^{(1)}} X_i^{(3)} + \sigma_{n^{(2)}m^{(2)}} X_i^{(4)} \\ m_i^{(2)} &= X_i^{(2)} + X_i^{(4)} + X_i^{(7)} \end{aligned} \tag{S41}$$

The readout pattern reads only from the second population:

$$w_i = \sigma^{(2)}_{m^{(2)}w} X_i^{(7)} \tag{S42}$$

The input pattern $\boldsymbol{I}^B$ also stimulates only the second population, pushing the dynamics towards one fixed point on the direction $\boldsymbol{m}^{(2)}$

$$I_i^B = X_i^{(6)} \tag{S43}$$

while the input pattern $\boldsymbol{I}^A$ activates the two populations. For units in the first population

$$I_i^A = X_i^{(8)} \tag{S44}$$

pushing the dynamics towards the other fixed point on the direction $\boldsymbol{m}^{(2)}$, while for i in the second population

$$I_i^A = \sigma^{(2)}_{I^A I^A} X_i^{(5)}, \tag{S45}$$

with $\sigma^{(2)}_{I^A I^A} \gg 1$

Such a connectivity leads to the dynamical equation for the two internal collective variables

$$\begin{aligned} \frac{d\kappa_1}{dt} &= -\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(2)}}\kappa_2 \\ \frac{d\kappa_2}{dt} &= -\kappa_2 + \tilde{\sigma}_{n^{(2)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(2)}m^{(2)}}\kappa_2 + \tilde{\sigma}_{n^{(2)}I^A} v_A + \tilde{\sigma}_{n^{(2)}I^B} v_B \end{aligned} \tag{S46}$$

with the effective couplings mediating inputs of the form

$$\tilde{\sigma}_{n^{(2)}I^A} = \frac{1}{2}\sigma^{(2)}_{n^{(2)}I^A}\langle\Phi'\rangle_2 \tag{S47}$$

$$\tilde{\sigma}_{n^{(2)}I^B} = \frac{1}{2}\sigma^{(2)}_{n^{(2)}I^B}\langle\Phi'\rangle_2 \tag{S48}$$

and effective couplings governing the autonomous dynamics:

$$\tilde{\sigma}_{n^{(1)}m^{(1)}} = \frac{1}{2}\sigma^{(1)}_{n^{(1)}m^{(1)}}\langle\Phi'\rangle_1 \tag{S49}$$

$$\tilde{\sigma}_{n^{(1)}m^{(2)}} = \frac{1}{2}\sigma^{(1)}_{n^{(1)}m^{(2)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(1)}m^{(2)}}\langle\Phi'\rangle_2 \tag{S50}$$

$$\tilde{\sigma}_{n^{(2)}m^{(1)}} = \frac{1}{2}\sigma^{(1)}_{n^{(2)}m^{(1)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(2)}m^{(1)}}\langle\Phi'\rangle_2 \tag{S51}$$

$$\tilde{\sigma}_{n^{(2)}m^{(2)}} = \frac{1}{2}\sigma^{(1)}_{n^{(2)}m^{(2)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(2)}m^{(2)}}\langle\Phi'\rangle_2 \tag{S52}$$

$$\tag{S53}$$

with averaged gains given by equations (S10) and standard deviations of currents onto each population

$$\begin{aligned}\Delta^1 &= \sqrt{(\sigma^{(1)}_{m^{(1)}m^{(1)}})^2\kappa_1^2 + \sigma^{(1)}_{m^{(2)}m^{(2)}})^2\kappa_2^2 + (\sigma^{(1)}_{I^AI^A})^2 v_A^2} \\ \Delta^2 &= \sqrt{(\sigma^{(2)}_{m^{(1)}m^{(1)}})^2\kappa_1^2 + \sigma^{(2)}_{m^{(2)}m^{(2)}})^2\kappa_2^2 + (\sigma^{(2)}_{I^AI^A})^2 v_A^2 + (\sigma^{(2)}_{I^BI^B})^2 v_B^2}\end{aligned} \tag{S54}$$

This dynamics can be graphically summarized as in Fig.5f bottom. It reproduces the dynamics of trained rank two networks presented in the main text (Fig.S4b), by relying on the same network mechanism, with input A controlling the gains of neurons in population one (Fig.S4c, middle) and thus shaping the dynamical landscape on which internal collective variables evolve (Fig.S4d). The important non-zero covariances of the reduced model are given by: $\sigma^{(1)}_{n^{(1)}m^{(1)}} = 0.34, \sigma^{(1)}_{n^{(2)}m^{(2)}} = 3.7, \sigma^{(1)}_{n^{(1)}m^{(2)}} = -3.9, \sigma^{(1)}_{n^{(2)}m^{(1)}} = 4.2$ for the first population and $\sigma^{(2)}_{n^{(2)}m^{(2)}} = 3.1, \sigma^{(2)}_{n^{(1)}m^{(2)}} = 4.4, \sigma^{(2)}_{n^{(2)}m^{(1)}} = -3.8, \sigma^{(2)}_{n^{(2)}I^A} = 0.2, \sigma^{(2)}_{n^{(2)}I^B} = -0.2, \sigma^{(2)}_{m^{(2)}w} = 1.0, \sigma^{(2)}_{I^AI^A} = 10.$

S3 Non-uniqueness of network implementation for a given task

We observed that trainings on a given task can lead to various network implementations. We identified three factors that contribute to such variability.

A first factor is the determination of the network parameters that are trained (e.g. number of connectivity modes R, train input patterns or not, scaling of trained parameters with network size, etc.). An example of this is provided by training a rank-one network on the context-dependent decision-making task, without training any of the input patterns (while the contextual-input $\boldsymbol{I^{ctxA}}$ and $\boldsymbol{I^{ctxB}}$ patterns are trained for the rank-one networks presented in the main text). Fig. S5 reports the analysis of such a trained network, showing that training leads to a network with three functional populations, whose implication in the computations are reproduced and detailed in a reduced model (section S2.3), and which is reminiscent of the one found in [Yang et al., 2019]. Another such example concerns the number of connectivity modes allowed during training. For instance if training a rank-two networks on the perceptual decision-making task, one could exhibit networks with a ring-like slow manifold [Mastrogiuseppe and Ostojic, 2018], which gives rise to a single, non-linear collective variable embedded in a two-dimensional subspace.

A second factor is task parametrization. For instance we observed that training on the parametric-working memory task with fixed delays between the two stimuli, while they are drawn randomly here, leads to solutions that exploit network oscillations rather than a line attractor (not shown). Another such example can be put forward for the context-dependent decision-making task. Here we trained networks on a two-alternatives forced choice version of this task in which every stimulus requires one out of two responses (section 4.9.3) and found that multiple populations were required for the implementation (Fig. S7). In a Go-Nogo setting, where the alternatives are to either respond or not, flexible input-output associations can be implemented with a single population, through a mechanism based on biasing the response threshold rather than modulating the gain [Mastrogiuseppe and Ostojic, 2018].

A third factor is the stochastic nature of the training procedure, with initial conditions of trained parameters randomly drawn for each training, as well as the stochastic split of training examples into batches inherent to stochastic-gradient-descent based methods as used here. In Fig. S10, we show the dynamics of a network trained on the delay-match-to-sample task obtained for the same task parametrization and the same trained parameters.

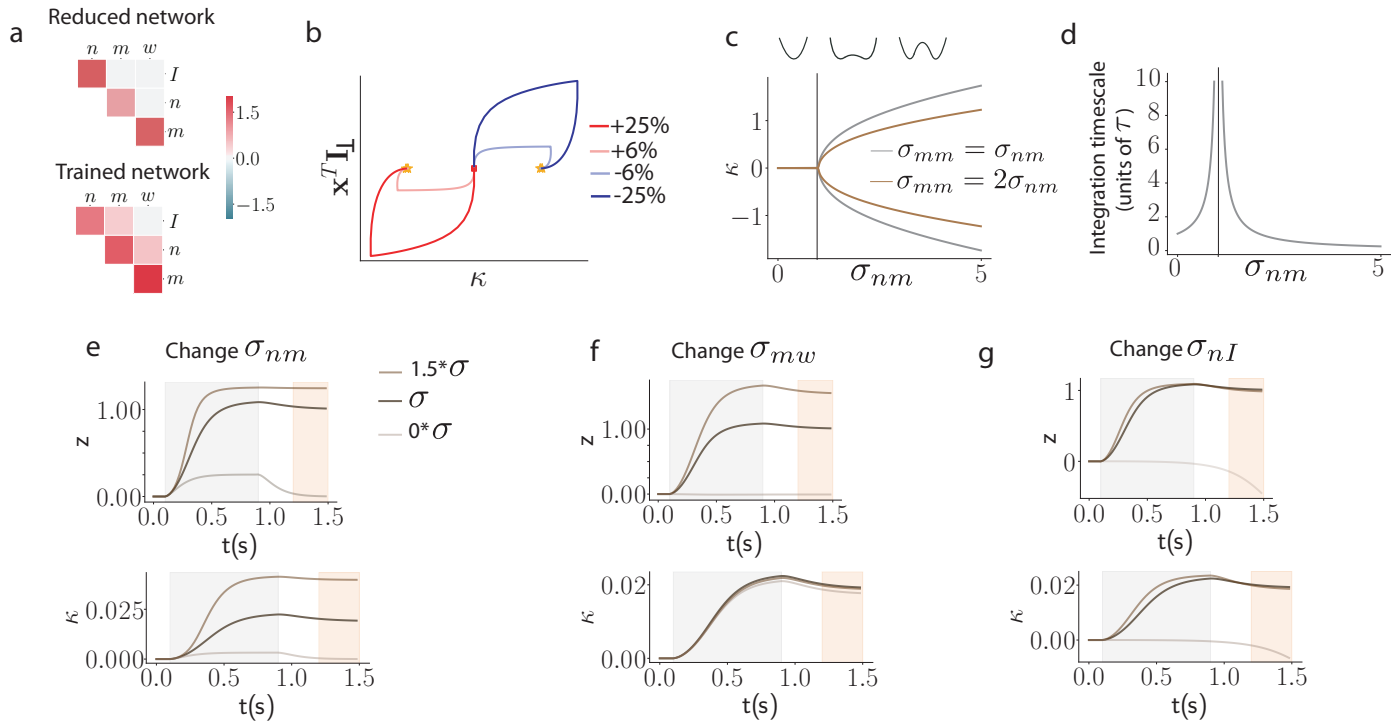

Figure S1. Theoretical analysis of reconstructed networks for the perceptual DM task. (a) Covariances between patterns of reduced and trained networks. (b) Dynamics throughout trials. Left: dynamics in the recurrents and input subspaces. Top-right: readouts. Bottom-right: neural gains averaged over the whole population. At the beginning of the trials, the network state is around $\boldsymbol{x} = 0$ and integrates inputs with $\Phi' \simeq 1$, and then commits to a decision by flowing to one of two fixed points where $\Phi' \simeq 0.6$. (c) Bifurcation analysis of the autonomous dynamics showing the value of the internal collective variable κ^* at the stable fixed-points of the network. (d) Time-scale of network dynamics around the network state at trial start $\boldsymbol{x} = \mathbf{0}$ for $\sigma_{mm} = \sigma_{mn}$. (e,f,g) Change in readout and internal collective variable dynamics as network connectivity is varied.

Similarly to the solution described in the main text, it relies on gain modulations through external inputs to shape the autonomous dynamics of the network. However, such solution switches between two dynamical landscapes with different sets of fixed points and separatrix between fixed points, while the solution in the main text switches between a dynamical landscape with two fixed-points and one with a limit cycle.

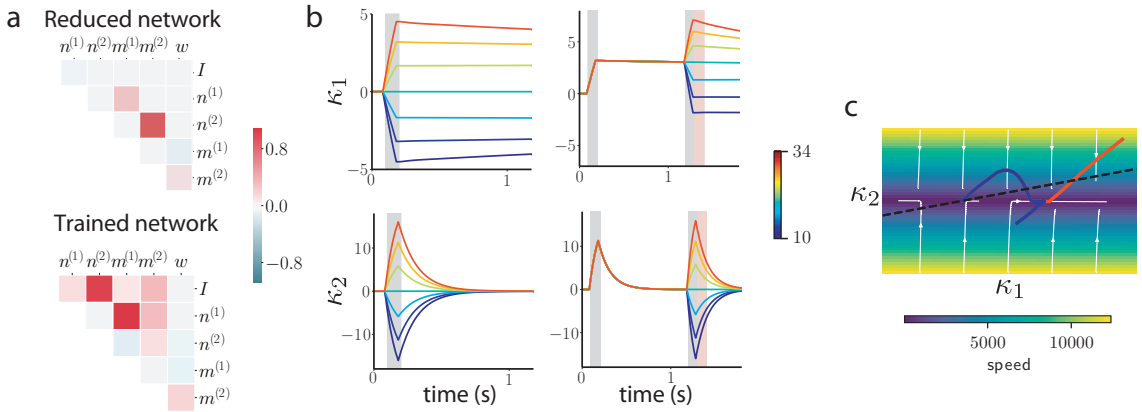

Figure S2. Theoretical analysis of reconstructed networks for the parametric working memory task. (a) Covariances between patterns of reduced and trained networks. (b) Low-dimensional dynamics of internal collective variables. Left: responses to the first stimulus (colors represent different values of f_1). Right: responses throughout the whole trial to a range of values for the second stimulation (f_1 fixed at 30Hz, colors represent different values of f_2). (c) Dynamical landscape on which the two internal collective variables evolve. From yellow to blue color, decreasing norm of the flow field $\sqrt{\dot{\kappa}_1(v=0)^2 + \dot{\kappa}_2(v=0)^2}$. Full lines depict two trajectories corresponding to $f_1 = 22$Hz and $f_2 = 30$Hz (blue) and $f_2 = 14$Hz (orange) and the dashed line represents the direction of the readout pattern $\boldsymbol{w}$.

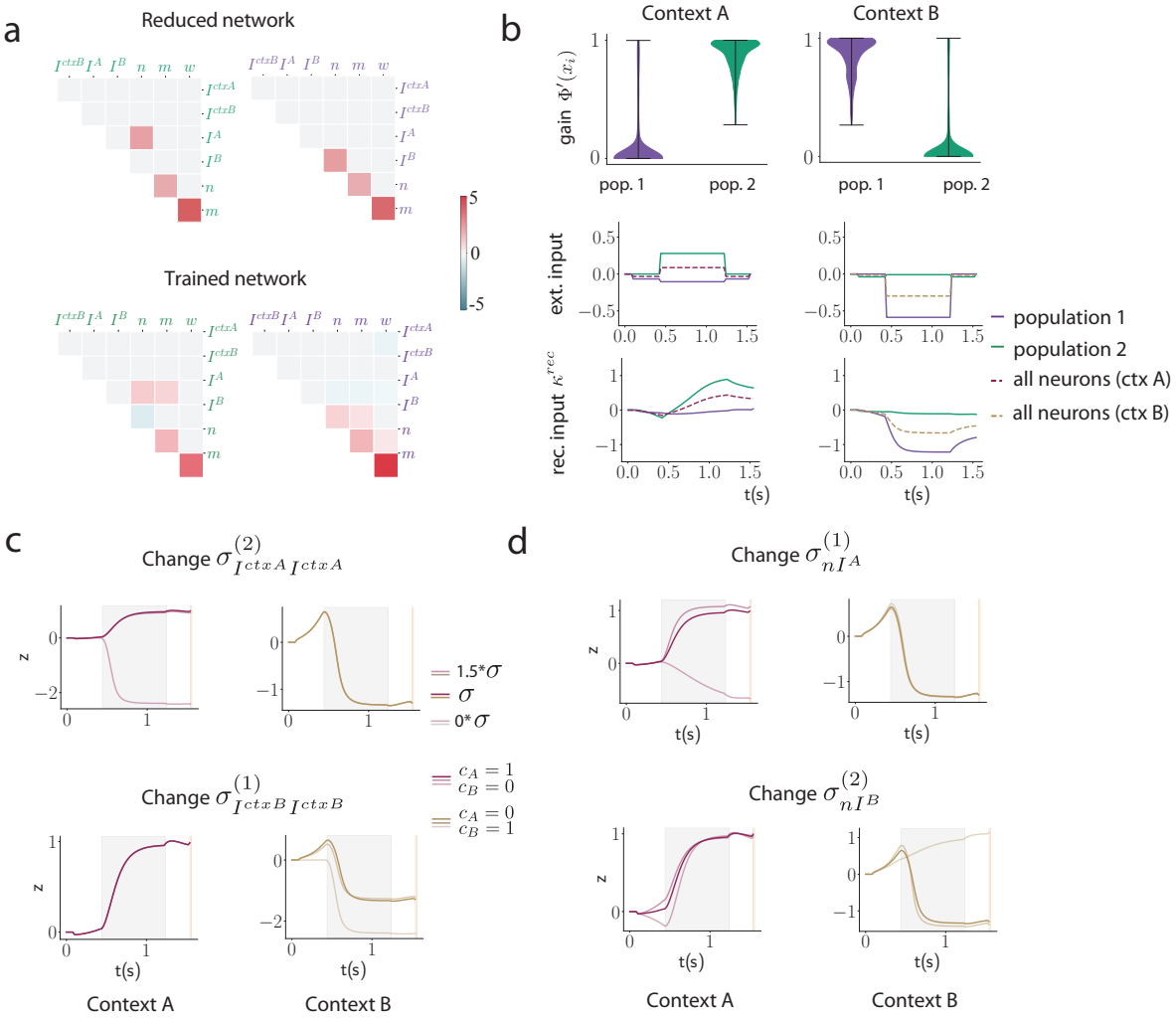

Figure S3. Theoretical analysis of reconstructed networks for the context-dependent task. (a) Covariances between patterns of reduced and trained networks. (b) Top row: Distribution of single neuron gains across the two populations, in the two contexts. Middle-bottom row: contribution of the two populations to the dynamics of the internal collective variable in the two contexts. Middle: contributions to the external inputs to the internal collective variable. Bottom: contribution to the recurrent feedback on the internal collective variable. (c,d) Changes in readout dynamics as network connectivity is varied.

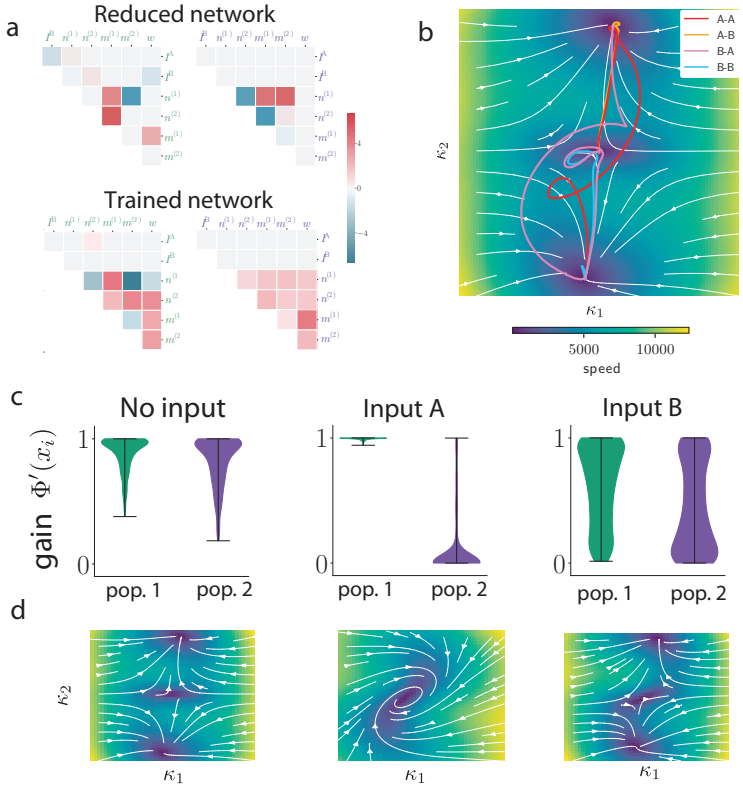

Figure S4. Theoretical analysis of reconstructed networks for the delay-match-to-sample task. (a) Covariances between patterns of reduced and trained networks. (b) Trajectories of activities in the 2-dimensional space spanned by internal collective variables. (c) Distributions of individual neuronal gains in each of the two populations in the present of inputs. (d) Dynamical landscape in which the internal collective variables evolve in the various stimulation conditions of the task.

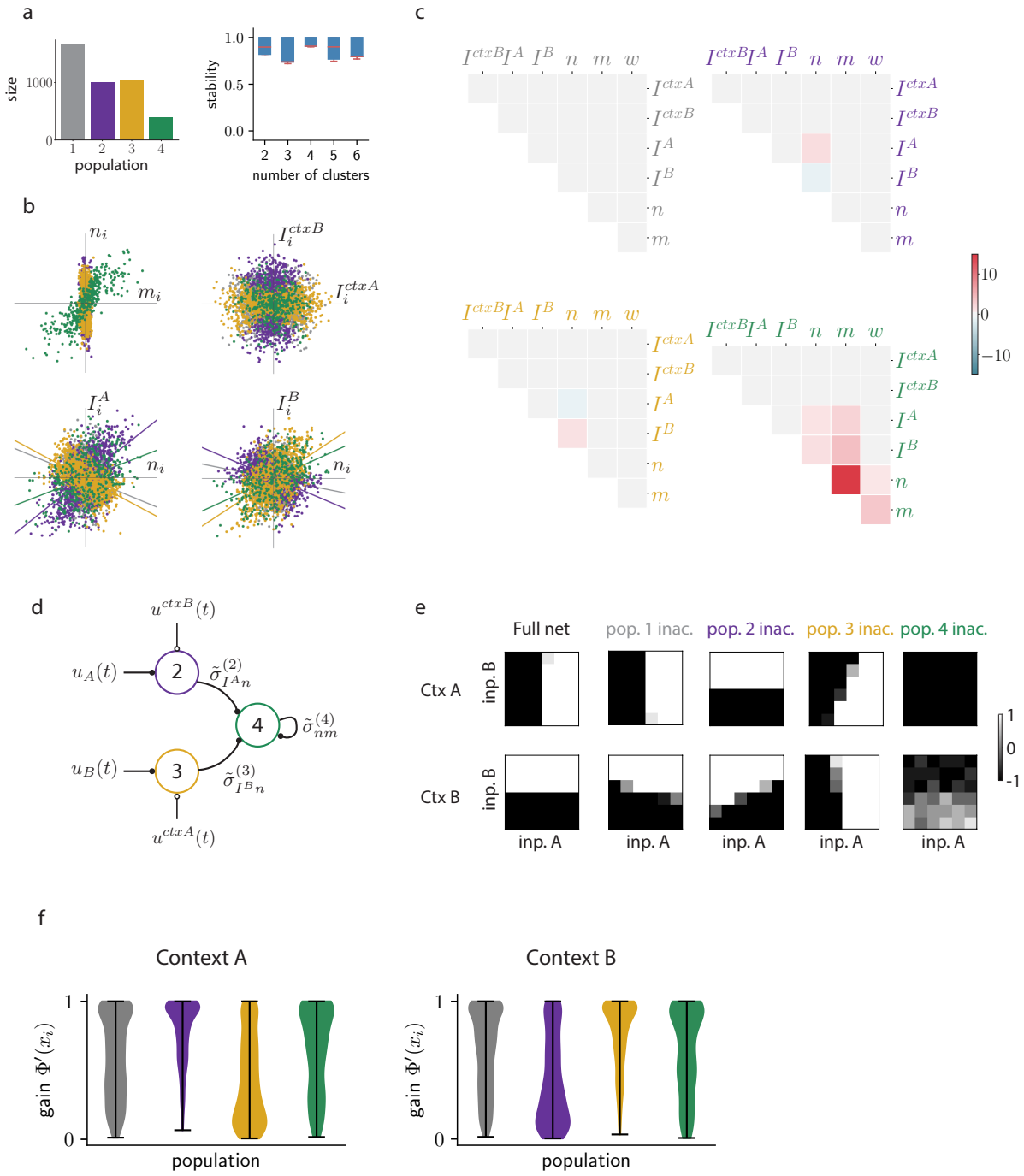

Figure S5 *(previous page)*: Context-dependent decision-making: alternative implementation. A network of $N = 4096$ neurons was trained on the same task as for figure 4, the only difference being that contextual input patterns were not trained. The solution found here exhibits a mechanism based on 3 populations, although the loading space statistics are better explained by taking into account a fourth population of neurons not involved in the computations (here in gray). In the whole figure, population 4 implements the effective line attractor for integration of evidence (similarly to the network exhibited in figure 2), while the populations 2 and 3 in violet and yellow act as relays, respectively of inputs B and A, modulated by the contextual inputs. (a) Sizes of the populations, and clustering stability analysis on loading space. (b) Some 2-dimensional projections of the loading space. The N points are color coded according to the clustering analysis. (c) Empirical covariances between patterns for each of the four populations. (d) Functional circuit explaining the role of the different populations. (e) Directly inactivating the whole populations found on the loading space reveals their functional role. Here, psychometric matrices are shown for the network with each of the populations inactivated. (f) Distribution of neuronal gains in both contexts accross populations.

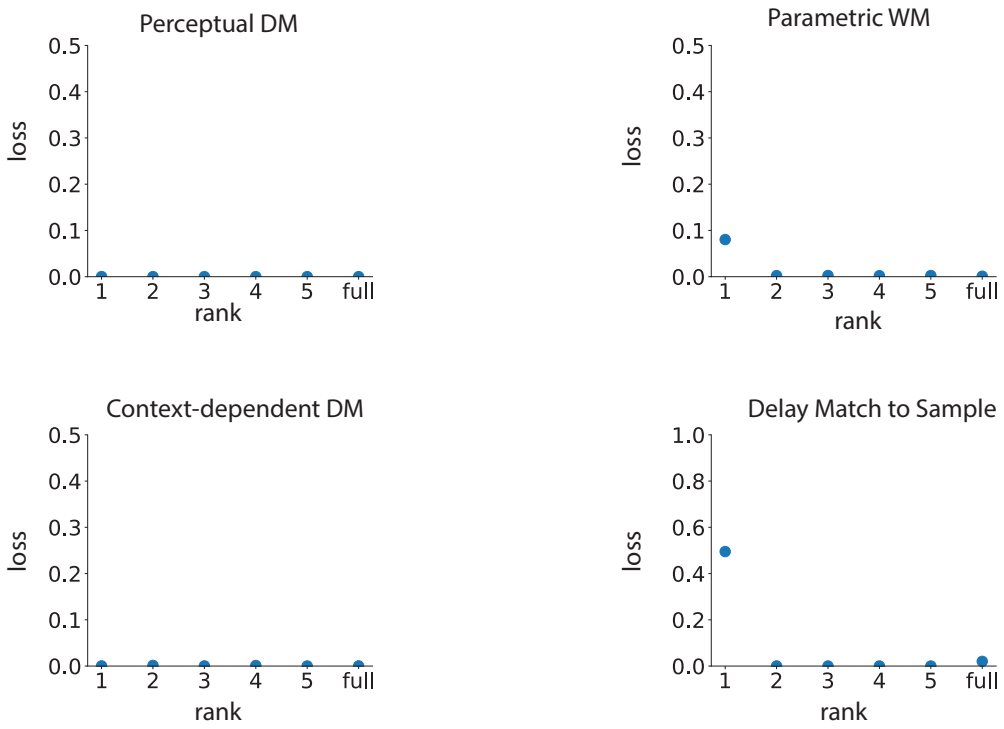

Figure S6. Loss of trained networks as a function of rank for 4 different tasks. For each task, low-rank networks in an exactly equivalent way, with a learning rate of 10^{-2} and a number of epochs much higher than what is normally needed (around 200). Full-rank networks required a different learning rate (10^{-4}).

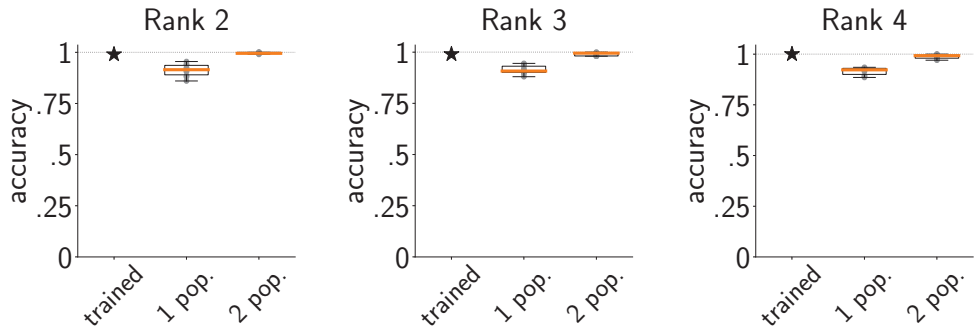

Figure S7. Performance of resampled networks extracted from higher rank solutions for the context-dependent decision making tasks, as in figure 3c of the main text. Here, networks of rank respectively 2, 3, and 4 have been trained on the same task, the statistics of the resulting connectivities have been extracted to define corresponding mixture-of-gaussians low-rank RNNs. Those have been retrained to make sure that single population networks would still not be able to recover the performance of both trained and 2 population networks, in spite of an increased rank. Note that an increase in the performance of the 1 population networks with respect to figure 4c can be seen, probably because the network can take advantage of the higher rank to implement a diagonal separatrix similarly to the contextual Go-NoGo task [Mastrogiuseppe and Ostojic, 2018]. However this is not sufficient for a perfect performance in the forced choice task. Increasing the rank beyond 2 does not seem to bring any more changes.

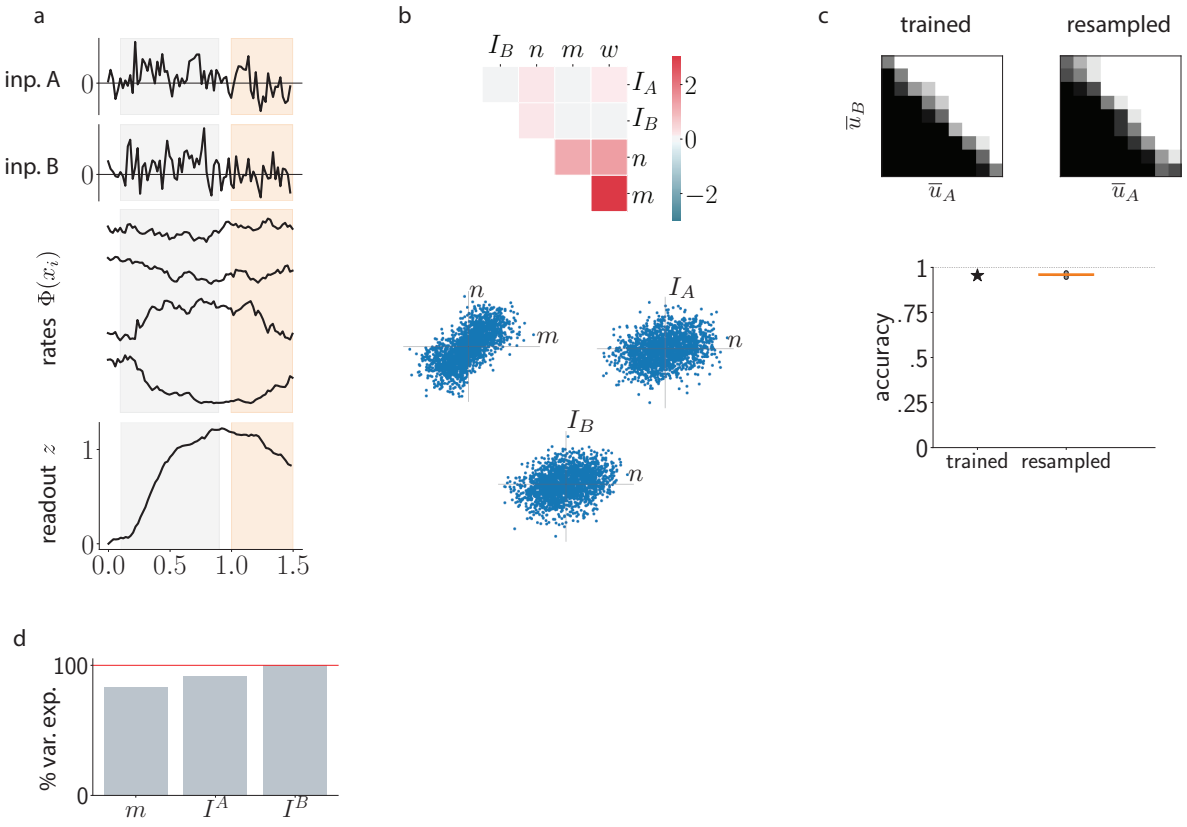

Figure S8. Multi-sensory decision making task. A network of size $N = 2048$ was trained on an implementation of a multi-sensory decision making task [Raposo et al., 2014], as described in the methods. (a) Top-bottom: both input signals to the network in an example trial with positive evidence, activity of 4 randomly selected neurons and of the readout unit to this stimulation. Grey background: stimulation period, brown background: response period. (b) Top: covariances between input, connectivity and readout patterns. Bottom: Selected two-dimensional projections of the loading space. (c) Top: psychometric performance of a trained rank one solution and of a resampled network with one gaussian population. Bottom: accuracies of the trained network and of ten resampled networks with one gaussian population (whisker plot). (d) Cumulative percentage of variance of the network activity explained by projecting successively onto the patterns m, I_A and I_B

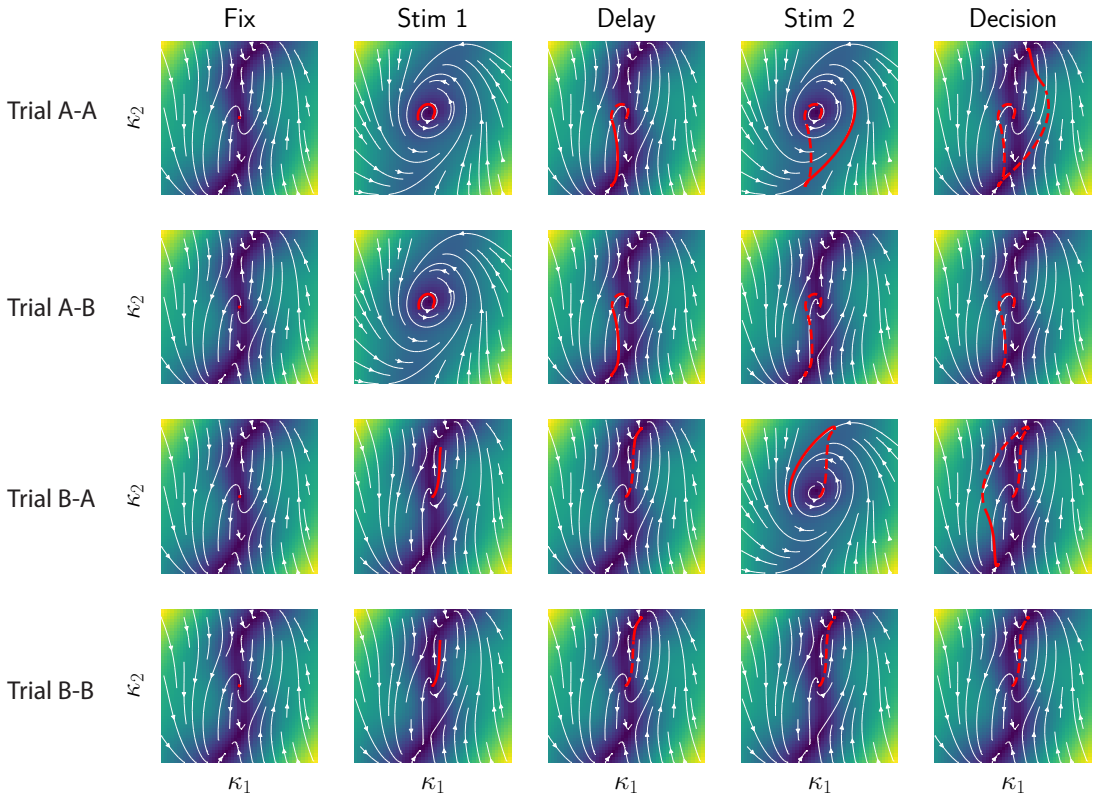

Figure S9. Dynamics of the internal collective variables κ_1 and κ_2 for the rank 2 network of figure 5 on 4 different trial types and for each epoch of the task. The trajectories are superposed on the flow field of the network as visible on an affine plane of the state space (see main figure 4i), dependent on the input present at each epoch. Dotted lines indicate the parts of the trajectory from previous epochs.

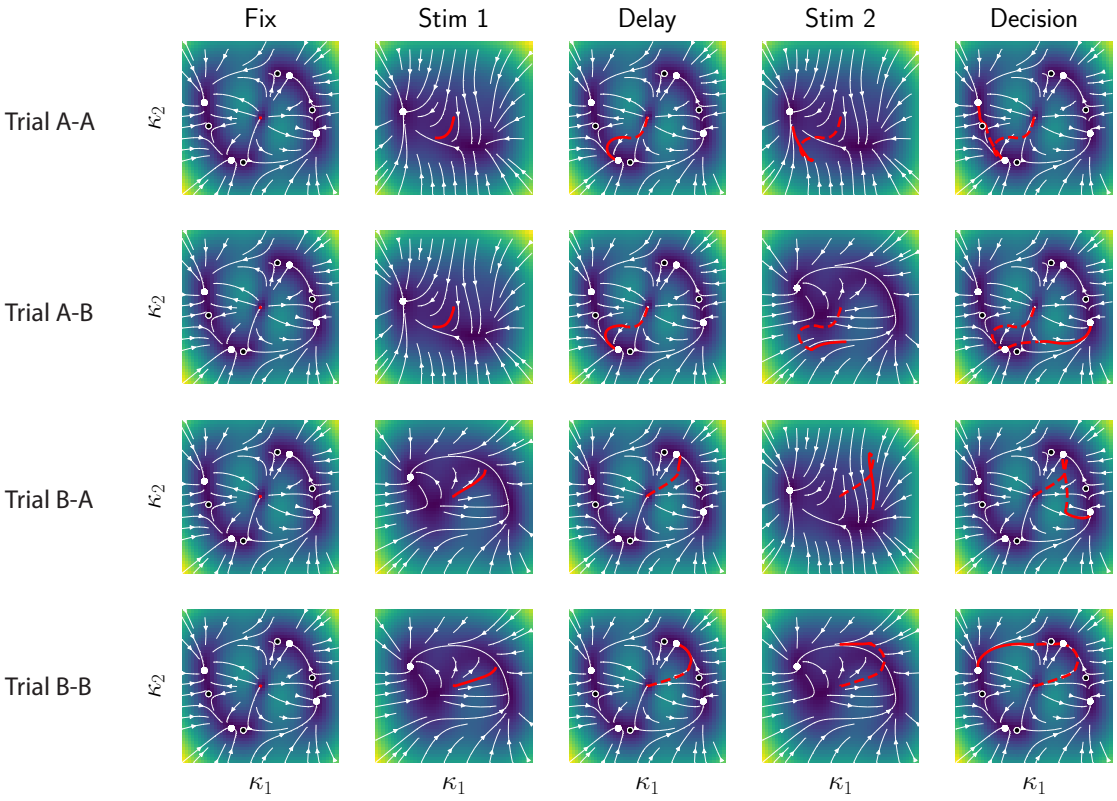

Figure S10. Alternative rank 2 solution to the Delay Match-to-Sample task. Here, a network of $N = 500$ units was trained on the same task as the network of figure 5, and exhibits a different mechanism, based on 4 stable fixed points. During the first stimulation period, the network reaches one of the 2 vertical fixed points, which hence encode the memory of the first stimulation. Depending on the second stimulation, the network then reaches one of the 2 horizontal fixed points, which thus encode the match or non-match categories.

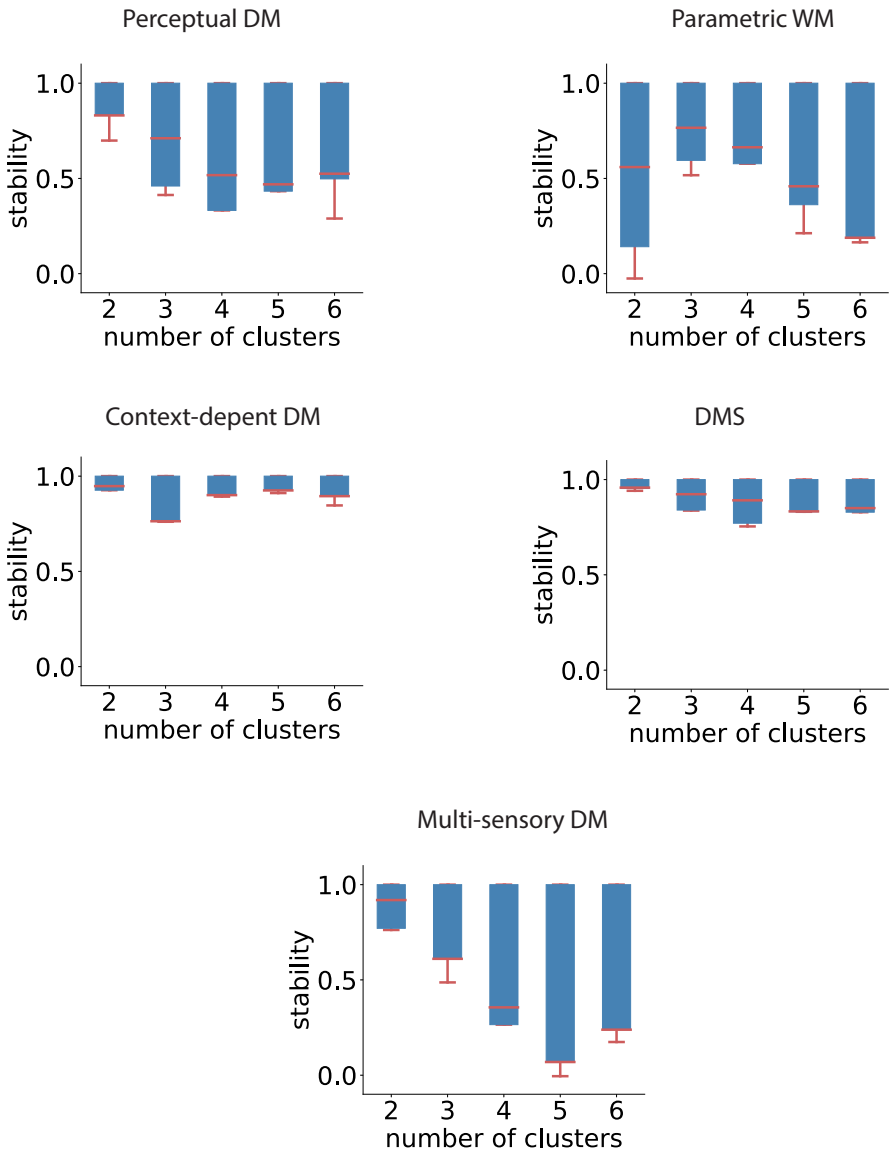

Figure S11. Measure of clustering stability to evaluate the presence and number of clusters in the loading spaces of low-rank RNNs trained on 5 different tasks. For each number of clusters, a clustering algorithm is applied on 20 bootstrap samples of the same data, and stability between the obtained results accross samples is measured. A value consistently near 1 implies that the given number of clusters is a good fit to the data (see Methods).