Published in final edited form as:

*Nat Neurosci*. 2022 June ; 25(6): 783–794. doi:10.1038/s41593-022-01088-4.

# The role of population structure in computations through neural dynamics

**Alexis Dubreuil**[1,2,*,+], **Adrian Valente**[1,*,+], **Manuel Beiran**[1,3], **Francesca Mastrogiuseppe**[4,5], **Srdjan Ostojic**[1,+]

[1]Laboratoire de Neurosciences Cognitives et Computationnelles, INSERM U960, Ecole Normale Superieure - PSL Research University, 75005 Paris, France

[2]Université de Bordeaux, CNRS, IMN, UMR 5293, Bordeaux, France

[3]Center for Theoretical Neuroscience, Zuckerman Institute, Columbia University, New York, NY 10027, USA

[4]Gatsby Computational Neuroscience Unit, UCL, London, Great Britain

[5]Champalimaud Research, Lisbon, Portugal

## Abstract

Neural computations are currently investigated using two separate approaches: sorting neurons into functional sub-populations, 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 on neuroscience tasks, here we show that the dimensionality of the dynamics and sub-population structure play fundamentally complementary roles. While various tasks can be implemented by increasing the dimensionality in networks with fully random population structure, flexible input-output mappings instead require a non-random population structure that can be described in terms of multiple sub-populations. Our analyses revealed that such a sub-population structure enables flexible computations through a mechanism based on gain-controlled modulations that flexibly shape the collective dynamics. Our results lead to task-specific predictions for the structure of neural selectivity, inactivation experiments, and for the implication of different neurons in multi-tasking.

## Introduction

The quest to understand the neural bases of cognition currently relies on two disjoint paradigms[1]. Classical works have sought to determine the computational role of individual cells by sorting them into functional populations based on their responses to sensory and behavioral variables[2;3;4]. Fast developing tools for dissecting neural circuits have

[+]Corresponding authors: alexis.dubreuil@gmail.com, adrian.valente@ens.fr, srdjan.ostojic@ens.psl.eu.
[*]equal contributions

Author contributions

A.D., A.V. and S.O. designed the study. A.D., A.V., S.O. developed the training and analysis pipelines. A.D., A.V., M.B., F.M., S.O. performed research and contributed to writing the manuscript.

Competing interests

There are no competing interests.

opened the possibility of mapping such functional populations onto genetic and anatomic cell types, and given a new momentum to this cell-category approach[5;6;7;8;9;10;11]. 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[12;13;14;15;16], where sharply defined functional cell populations are often not directly apparent[15;17;4]. A newly emerging paradigm has therefore proposed that neural computations need instead to be interpreted in terms of collective dynamics in the state space of joint activity of all neurons[18;14;15;19;20;21]. This computation-through-dynamics framework[22] hence posits that neural computations are revealed by studying the geometry of low-dimensional trajectories of activity in state space[15;23;24;20;25;26], while remaining agnostic to the role of any underlying population structure.

In view of the apparent antagonism between these two approaches, two works have sought to precisely assess the presence of functional cell populations in the posterior parietal cortex (PPC)[17] and prefrontal cortex[10]. Rather than define cell populations by classical methods such as thresholding the activity or selectivity of individual neurons, these studies developed new statistical techniques to determine whether the distribution of selectivity across neurons displayed a non-random population structure[4]. Using analogous analyses, but different behavioral tasks, the two studies reached opposite conclusions. Raposo et al found no evidence for non-random population structure in selectivity, and argued that PPC neurons fully multiplex information. Hirokawa et al also observed that individual neurons responded to mixtures of task features, but in contrast to Raposo et al, they detected important deviations from a fully random distribution of selectivity, a situation they termed *non-random mixed selectivity*. By clustering neurons according to their response properties, they defined separate, though mixed-selective populations that appeared to represent distinct task variables and to reflect underlying connectivity. To resolve the apparent discrepancy, Hirokawa et al conjectured that revealing non-random population structure in higher cortical areas may require sufficiently complex behavioral tasks.

These conflicting findings therefore raise a fundamental theoretical question: do specific computational tasks require a non-random population structure, or alternatively can any task in principle be implemented with a fully random population structure as in[17]? To address this question, we trained recurrent neural networks on a range of systems neuroscience tasks[27;28;29] and examined the population structure that emerged in both selectivity and connectivity using identical methods as[17;10]. Starting from the premise that computations are necessarily determined by the underlying connectivity[30], we then developed a new approach for assessing the computational role of population structure in connectivity for each task. Together, these analyses revealed that, while a fully random population structure was sufficient to implement a range of tasks, specific tasks required a non-random population structure *in connectivity* that could be described in terms of a small number of statistically-defined sub-populations. This was in particular the case when a flexible reconfiguration of input-output associations was needed, a common component of many cognitive tasks[31] and more generally of multi-tasking[29;32;33]. To extract the mechanistic role of this population structure for computations-through-dynamics, we focused on the class of low-rank models[30;34;35] that can be reduced to interpretable latent dynamics characterized by a minimal intrinsic dimension and number of sub-populations[36]. We found

that the sub-population structure of the connectivity enables networks to implement flexible computations through a mechanism based on gain modulations[37;38] of effective interactions between latent variables. Our results lead to task-specific predictions for the statistical structure of single-neuron selectivity, for inactivations of specific sub-populations, as well as for the implication of different neurons in multi-tasking.

## Results

### Identifying non-random population structure in trained networks

We trained recurrent neural networks (RNNs) on five systems neuroscience tasks[29;39] spanning a range of cognitive components: perceptual decision-making (DM)[40], parametric working-memory (WM)[41], multi-sensory decision-making (MDM)[17], contextual decision-making (CDM)[15] and delay-match-to-sample (DMS)[42]. We then searched for evidence of non-random population structure by comparing the selectivity, connectivity and performance of the trained networks with randomized shuffles.

We first asked if training on each task led to the emergence of non-random structure in selectivity. Following Raposo et al 2014 and Hirokawa et al 2019, we represented each neuron as a point in a *selectivity space*, where each axis was given by the linear regression coefficient of neural firing rate with respect to a task variable such as stimulus, decision or context (Fig. 1a). The dimension of the selectivity space ranged from 2 to 4 depending on the task (see Methods), and each trained network led to a distribution of points in that space (Fig. 1b). For each network, we used the ePAIRS[17;10] test to compare the obtained distribution with a randomized shuffle corresponding to a multivariate Gaussian (Fig. 1b,c). A non-significant outcome suggests an isotropic distribution of single-neuron selectivity, a situation that has been denoted as non-categorical mixed selectivity[17] and we refer to it as *fully-random population structure*. A statistically significant outcome instead indicates that neurons tend to be clustered along multiple axes of the selectivity space. Following[17;10], we refer to this situation as non-random mixed selectivity, or *non-random population structure*. The ePAIRS analysis revealed the presence of non-random population structure for two out of the five tasks, the contextual decision-making and delay-match-to-sample tasks (Fig. 1d) (proportion of statistically significant networks under the ePAIRS test, $p < 0.05$, Bonferroni corrected : DM 1/100, WM 6/100, MDM 10/100, CDM 87/100, DMS 100/100, Extended Data Figure 1). In particular, we found a clear difference between the multi-sensory[17] and context-dependent[15] decision making tasks, which had an identical input structure and therefore selectivity spaces of identical dimensions, but required different mappings from inputs to outputs.

The selectivity in trained RNNs necessarily reflects the underlying connectivity[30]. We therefore next sought to determine the presence of non-random population structure directly in the connectivity of trained networks by applying an analogous analysis in a *connectivity space*. To define a connectivity space with a minimal number of parameters, we focused on RNNs constrained to have recurrent connectivity matrices $J_{ij}$ of a fixed rank $R$, a type of connectivity structure that typically emerges when training RNNs on simple tasks[35]. A matrix of rank $R$ can be written as

$$J_{ij} = m_i^{(1)} n_j^{(1)} + \cdots + m_i^{(R)} n_j^{(R)}, \quad (1)$$

so that neuron $i$ is characterized by $2R$ recurrent connectivity parameters $\{m_i^{(r)}, n_i^{(r)}\}_{r=1\ldots R}$, as well as $N_{in}$ input weights $I_i^{(s)}$ and a readout weight $w_i$ (see Methods). For each task, we determined the minimal required rank $R$ (Extended Data Figure 2). We then represented the connectivity of each neuron as a point in a $(2R+N_{in}+1)$-dimensional *connectivity space*, and described the connectivity of a full network as the corresponding distribution of points (Fig. 1e,f). Similarly to the selectivity analysis, we assessed the presence of non-random population structure by comparing connectivity distributions of trained networks with randomized shuffles corresponding to multivariate Gaussians with matching empirical means and covariances. The results were consistent with the analysis of selectivity (Fig. 1g,h), and showed a gap between the same two groups of tasks (Fig. 1h, number of networks with statistically significant clustering for each task: DM 3/100; WM 5/100; MDM 1/100; CDM 100/100; DMS 100/100; $p < 0.05$ with Bonferroni correction). In particular the MDM and CDM tasks again led to opposite results although their connectivity spaces were identical.

The analyses of selectivity and connectivity are purely correlational, and do not allow us to infer a causal role of the observed structure (see Supplementary Text 1). To determine when non-random population structure is computationally necessary, or conversely when random population structure is computationally sufficient, we therefore developed a new *resampling* analysis. For each task, we first generated new networks by sampling the connectivity parameters of each neuron from the randomized distribution used to assess structure in Fig. 1e–h, i.e. a multivariate Gaussian distribution with mean and covariance matching the trained low-rank RNNs. This procedure preserved the rank of the connectivity (Fig. 1e), and the overall correlation structure of connectivity parameters, but scrambled any non-random population structure (Fig. 1j,k). We then quantified the performance of each randomly resampled network on the original task. This key analysis revealed that the randomly resampled networks led to a near perfect accuracy for the DM, WM and MDM tasks, but not for the CDM and DMS tasks (Fig. 1l). This demonstrates that, on one hand, random population structure is sufficient to implement the DM, WM and MDM tasks, while on the other hand non-random population structure is necessary for CDM and DMS tasks. These results held independently of the constraints on the rank of the connectivity, and in particular for unconstrained, full-rank networks in which only the learned part of the connectivity was resampled (Extended Data Figure 3).

In summary, our analyses of trained recurrent neural networks revealed that certain tasks can be implemented with a fully-random population structure in both connectivity and selectivity, while others appeared to require additional organization in the connectivity that led to non-random structure in selectivity. We next sought to understand the mechanisms by which the population structure of connectivity determines the dynamics and the resulting computations. In a first step, we examined the situation in which the population structure is fully random. In a second step, we asked whether non-random population structure in the

connectivity space could be represented in terms of separate clusters or sub-populations, and how this additional organization expands the computational capabilities of the network.

## Interpreting computations in terms of latent dynamics

To unravel the mechanisms by which population structure impacts computations, we developed a method for interpreting low-rank networks in terms of underlying low-dimensional dynamics[22;36]. Here we first outline this general model reduction approach (Fig. 2), and next apply it to trained recurrent networks.

In line with methods for analyzing large-scale neural activity[18;43;19;21], we represented the dynamics as trajectories $\boldsymbol{x}(t) = \{x_i(t)\}_{i=1...N}$ in the *activity state space*, where each dimension corresponds to the activation of one neuron (Fig. 2b). As in dimensionality reduction analyses, we then parametrized these trajectories by a small number of latent variables[43;19]. Crucially, for low-rank networks this dimensionality reduction is exact, because the connectivity structure directly restricts the dynamics to lie in a low-dimensional subspace[36]. Specifically, $\boldsymbol{x}(t)$ can be decomposed into a set of internal variables $\kappa_I(t)$ and inputs $u_s(t)$ that respectively quantify activity along recurrent and input-driven directions $\boldsymbol{m}^{(I)}$ and $\boldsymbol{I}^{(s)}$ in state-space[25], where $\boldsymbol{m}^{(I)}$ and $\boldsymbol{I}^{(s)}$ are *connectivity and input vectors* obtained by grouping connectivity parameters across neurons (see Fig. 2b and Methods Eq. 23). A mathematical analysis of network dynamics then shows that the set of internal variables $\boldsymbol{\kappa} = \{\kappa_I\}_{I=1...R}$ forms a dynamical system driven by inputs $\boldsymbol{u} = \{u_s\}_{s=1...N_{in}}$, with a temporal evolution given by

$$\frac{d}{dt}\boldsymbol{\kappa}(t) = F(\boldsymbol{\kappa}(t), \boldsymbol{u}(t)) \tag{2}$$

where $F$ is a non-linear function that determines the amount of change of $\boldsymbol{\kappa}$ at every time step. In the limit of large networks, the precise shape of $F$ is set by the statistics of the connectivity parameters across neurons (Methods Eq. 32), i.e. precisely the distribution of points in the connectivity space that we previously examined in Fig. 1f. The connectivity can therefore be interpreted in two complementary ways, either in terms of directions in the activity state-space (Fig. 2b top left) or in terms of distributions in the connectivity space (Fig. 2b bottom left) and these two representations together determine the low-dimensional latent dynamics.

In summary, in line with the computation-through-dynamics framework[20;22], low-rank networks can be exactly reduced to low-dimensional, non-linear latent dynamical systems which determine the performed computations. We next examined how the population structure in trained recurrent networks impacts the resulting latent dynamical system.

## Latent dynamics for fully random population structure

Our resampling analyses of trained RNNs revealed that a range of tasks could be performed by networks in which the population structure was fully random in connectivity space (Fig. 1l). We therefore first examined the latent dynamics underlying computations in that situation. Crucially, a fully random population structure limits the available parameter space, and strongly constrains the set of achievable latent dynamics independently of their

dimensionality[36] (see Methods). We start by specifying these constraints on the dynamics, and show they nevertheless allow networks with random population structure to implement a range of tasks of increasing complexity by increasing the rank of the connectivity and therefore the dimensionality of the dynamics.

Networks with fully random population structure were defined in Fig. 1i–l as having distributions of connectivity parameters computationally equivalent to a Gaussian distribution. In such networks, the statistics of connectivity are fully characterized by a set of covariances between connectivity parameters, each of which can be directly interpreted as the alignment, or overlap between two connectivity vectors (Fig. 2b bottom left, see Eq. 10). For this type of connectivity, a mean-field analysis shows that the latent low-dimensional dynamics can be directly reduced to an effective latent circuit, where internal variables $\kappa_r$ integrate external inputs $u_s$, and interact with each other through *effective couplings* set by the overlaps between connectivity vectors multiplied by a common, activity-dependent gain factor[36]. In such reduced models, the role of individual parameters can then be analyzed in detail (Supplementary Note 2).

As a concrete example, a unit-rank network ($R = 1$) with connectivity vectors $\boldsymbol{m}$ and $\boldsymbol{n}$ and a single feed-forward input vector $\boldsymbol{I}$ ($N_{in} = 1$) leads to two-dimensional activity, fully described by a single internal variable $\kappa(t)$ and a single external variable $u(t)$ (Fig. 2b). The latent dynamics of $\kappa(t)$ are given by

$$\tau \frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{nm}\kappa + \tilde{\sigma}_{nI}u(t), \tag{3}$$

where $\tilde{\sigma}_{nm}$ and $\tilde{\sigma}_{nI}$ are effective couplings defined as $\tilde{\sigma}_{nm} = \langle \Phi' \rangle \sigma_{nm}$ and $\tilde{\sigma}_{nI} = \langle \Phi' \rangle \sigma_{nI}$, where $\sigma_{nm}$ (resp. $\sigma_{nI}$) is the fixed overlap between the vector $\boldsymbol{n}$ and the vector $\boldsymbol{m}$ (resp. $\boldsymbol{I}$). The connectivity vector $\boldsymbol{n}$ therefore selects inputs to the latent dynamics[30]: the overlap between $\boldsymbol{n}$ and $\boldsymbol{I}$ controls how strongly the latent dynamics integrate feed-forward inputs, while the overlap between $\boldsymbol{n}$ and $\boldsymbol{m}$ controls the strength of positive feedback in the latent dynamics. Crucially, all the effective couplings are scaled by the same factor $\langle \Phi' \rangle$ that represents the average gain of all neurons in the network. This gain depends on the activity in the network (see Methods Eq. 39), which makes the dynamics in Eq. 3 non-linear. The fact that all the effective couplings are scaled by the same factor however implies that, in networks with a fully random population structure, the overall form of the effective circuit is fixed by the connectivity overlaps, and this strongly limits the range of possible dynamics for the internal variables[36].

Applying this model-reduction approach to the perceptual decision-making (DM) and parametric working-memory (WM) tasks confirmed that tasks for which a fully random population structure is sufficient are those that can be implemented by a fixed effective circuit at the level of latent dynamics (Fig. 3 and Supplementary Text 2). Increasing the rank allows networks to perform tasks of increasing complexity by relying on an increasing number of latent variables. The DM task for instance relies on a single latent variable that corresponds to integrated evidence (Fig. 3c,d), while the WM task exploits two variables (Fig. 3h,i). By fixing the shape of the effective circuit, the random population structure

however limits ways in which these latent variables can interact among themselves and with inputs. As a consequence, for more complex tasks a fully random population structure was not sufficient. We next sought to further elucidate this aspect.

## Representing non-random structure with multiple populations

The resampling analysis in Fig. 1l indicated that tasks such as context-dependent decision-making and delayed-match-to-sample relied on a population structure in connectivity that was not fully random. To better understand the underlying structure and its computational role, we further examined RNNs trained on these two tasks, and asked whether their connectivity could be represented in terms of multiple populations. We first examined whether a multi-population connectivity structure is sufficient to implement the two tasks, and in a second step examined how such a structure modifies latent dynamics and expands their computational capacity.

To identify computationally-relevant populations, we took inspiration from[10], and first performed clustering analyses in the connectivity space where non-random population structure was found (Fig. 4a, see Methods). Applying a Gaussian mixture clustering algorithm on the cloud of points formed by each trained network, we partitioned the neurons into separate sub-populations. In the trained networks, all clusters were centered close to the origin, but each had a different shape and orientation that corresponded to multiple peaks in the distribution detected by the ePAIRS analysis (Fig. 1f–g). Each population was therefore characterized by a different set of covariances, or overlaps, between input, recurrent, and output connectivity vectors. We then extended our resampling approach from Fig. 1i–l, and generated new networks by first randomly assigning each neuron to a population, and then sampling its connectivity parameters from a Gaussian distribution with the fitted covariance structure. Finally, we inspected the performance of these randomly generated networks, and compared them with fully trained ones. By progressively increasing the number of fitted clusters, we determined the minimal number of populations needed to implement the task (see Methods). Within this approach, networks with a fully random population structure such as those described in Fig. 3 correspond to a single overall population in connectivity space.

We first considered context-dependent decision making, where stimuli consisted of a combination of two scalar features that fluctuated in time[15]. Depending on a contextual cue, only one of the two features needed to be integrated (Fig. 4b), so that the same stimulus could require opposite responses, a hallmark of flexible input-output transformations[44]. We found that unit-rank connectivity was sufficient (Fig. Extended Data Figure 2), and focused on such networks. The analysis in Fig. 1l showed that generating networks by resampling connectivity from a single, fully-random population led to a strong degradation of the performance, although it remained above chance. A closer inspection of psychometric matrices representing input-output transforms in different contexts revealed that the resampled single-population networks in fact generated correct responses for stimuli requiring identical outputs in the two contexts, but failed for incongruent stimuli, for which responses needed to be flipped according to context (Fig. 4c). This observation was not specific to unit-rank networks, as randomizing population structure in higher-rank (Extended

Data Figure 4) and full-rank networks (Extended Data Figure 3) led to a similar reduction in performance. We therefore performed a clustering analysis in the connectivity space. The number of clusters varied across networks (Extended Data Figure 5 and Supplementary Text 3), but the minimal required number was two. For such minimal networks, we found that randomly resampling from the corresponding Gaussian mixture distribution led to an accuracy close to the original trained connectivity (Fig. 4d). In particular, the randomly generated networks correctly switched their response to incongruent stimuli across contexts, in contrast to networks with a single population (Fig. 4c). This indicated that connectivity based on a structure in two populations was sufficient to implement the context-dependent decision-making task.

An identical analysis based on clustering and resampling connectivity parameters showed that rank-two networks with two sub-populations could perform the delayed-match-to-sample task (Extended Data Figure 6 and Supplementary Text 4). Altogether, our results therefore indicated that connectivity distributions described by a small number of populations were sufficient to implement tasks requiring flexible input-output mappings. To identify the mechanistic role of this multi-population structure, we next examined how it impacted the latent dynamics implemented by trained networks.

## Gain modulation of latent dynamics

To unveil the mechanisms underlying flexible input-output mappings in networks with connectivity based on multiple populations, we examined how such a structure impacts the latent dynamics of internal variables. Here we first describe how, in contrast to a single-population, a multi-population structure allows external inputs to flexibly modulate the overall form of the circuit describing latent dynamics. We then show how this general principle applies specifically to the two flexible tasks (Fig. 4 and Extended Data Figure 6). We focus here on networks with minimal rank and minimal number of populations, and show in the next section that the inferred predictions hold more generally.

In Fig. 4 we defined sub-populations as subsets of neurons characterized by different overlaps between input, recurrent and output connectivity vectors in a network of fixed rank. In a network with a multi-population structure, the number of internal variables describing low-dimensional dynamics is determined by the rank of the recurrent connectivity, as in networks without population structure (Fig. 2a). Remarkably, a mean-field analysis[36] (see Methods) shows that the latent low-dimensional dynamics can still be represented in terms of an effective circuit where internal variables $\kappa_r$ integrate inputs and interact with each other through effective couplings (Fig. 5a). The key effect of the multi-population structure is however to modify the form of the effective couplings and endow them with much greater flexibility than in the case of a single, fully random population. Indeed, in a network with a single population, the effective couplings were given by connectivity overlaps multiplied by a single, global gain factor, and modulating the gain therefore scaled all effective couplings together. In contrast, in networks with multiple populations, each population is described by its own set of overlaps between connectivity sub-vectors (Fig. 4a), and, importantly, by its own gain, which corresponds to the average slope $\phi'(x_i)$ on the input-output nonlinearity of neurons in the population. The effective couplings between inputs and internal variables are

then given by a sum over populations of connectivity overlaps each weighted by the gain of the corresponding population (Methods Eq. (38)). As an illustration, in the case of two populations, the effective coupling between the input and the internal variable becomes

$$\tilde{\sigma}_{nI} = \sigma_{nI}^{(1)}\langle\Phi'\rangle_1 + \sigma_{nI}^{(2)}\langle\Phi'\rangle_2 \quad (4)$$

where $\sigma_{nI}^{(1)}$ and $\sigma_{nI}^{(2)}$ are the overlaps for each population between the input vector $\boldsymbol{I}$ and the input-selection vector $\boldsymbol{n}$, while $\langle\Phi'\rangle_1$ and $\langle\Phi'\rangle_2$ are the gains of the two populations, that depend implicitly both on inputs and the values of internal variables. Crucially, additional inputs restricted to a given population can modulate its gain independently of other populations by shifting the position of neurons on the non-linear input-output function. Depending on the geometry between input vectors and input-selection vectors, different sets of inputs can play distinct roles of drivers and modulators[37], allowing the network to flexibly remodel the effective circuit formed by collective variables in different trials or epochs according to the demands of the task.

We applied this model-reduction analysis to the context-dependent decision-making task, for which the minimal trained networks were of unit rank and consisted of two sub-populations (Fig. 4b). Analyzing the statistics of input and connectivity vectors for each population, we found that the input vectors $\boldsymbol{I}^A$ and $\boldsymbol{I}^B$ corresponding to the two stimulus features $u_A$ and $u_B$ had different overlaps with the input-selection vector $\boldsymbol{n}$ in the two populations (Fig. 5b right) so that the two stimulus features $u_A$ and $u_B$ acted as drivers of latent dynamics. The contextual input vectors $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$ in contrast had weak overlaps with the input-selection vector $\boldsymbol{n}$ (Extended Data Figure 7), but strongly different amplitudes on the two populations (Fig. 5b left). They therefore modified the gains of the two populations in an opposite manner (Fig. 5c bottom), and played the role of modulators that changed the form of the effective circuit describing latent dynamics in each context (Fig. 5c top). More specifically, the latent dynamics of the internal variable $\kappa$ could be approximated by (Methods and Sup. Fig. S4):

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

where $\langle\Phi'\rangle_1$ and $\langle\Phi'\rangle_2$ are the average gains of the two populations, $\sigma_{nI}^{(1)A}$ the overlap for the first population between the input vector for stimulus feature $A$ and the input-selection vector $\boldsymbol{n}$, and $\sigma_{nI}^{(2)B}$ the overlap for the second population between $\boldsymbol{n}$ and the input vector for stimulus feature $B$. By modulating the gains of the two populations in a differential manner between the two contexts (Fig. 5c bottom), the contextual cues controlled the effective couplings between stimulus inputs and the internal variable $\kappa$, and determined which feature was integrated by the internal variable in each context (Fig. 5d). This mechanism implemented an effective input gating, but only at the level of the latent dynamics of the internal variable $\kappa$ that integrated relevant evidence. Importantly, as observed in experimental data[15], on the level of the full network, the two stimulus features were instead equally represented in both contexts, but along directions in state space orthogonal to the

direction that encoded internal collective variable (Extended Data Figure 8) as observed in experimental data[15].

A similar picture was obtained for the DMS task (Fig. Extended Data Figure 9 and Supplementary Text 4), in which case the sub-population structure controlled autonomous dynamics rather than selecting a stimulus to be integrated. Altogether, our model-reduction analyses showed that networks with multiple sub-populations implemented flexible computations by exploiting gain modulation to modify in various ways the effective couplings between collective variables.

## Predictions for neural selectivity and inactivations

Analyzing networks of minimal rank and minimal number of sub-populations allowed us to identify the mechanisms underlying computations based on a multi-population structure in connectivity. We next sought to generate predictions of the identified mechanisms that are experimentally testable without access to details of the connectivity. We then tested these predictions on networks with a higher number of sub-populations or higher rank, obtained by varying the constraints used during training. We focus here specifically on the context-dependent decision-making (CDM) task, and contrast it with the multi-sensory decision-making (MDM) task, for which networks received an identical input structure, but were required to produce an output independent of context.

For the CDM task, reducing the trained networks to effective circuits revealed that the key computations relied on a differential gain modulation of separate sub-populations by contextual inputs. For each neuron, contextual cues set its functioning point on its non-linearity, and thus the gain of its response to incoming stimuli. A direct implication is that neurons more strongly modulated by contextual cues change more strongly their gain across contexts, and thereby the amplitude of their responses to stimulus features (Fig. 6a). An ensuing prediction at the level of selectivity of individual neurons is therefore that the pre-stimulus selectivity to context should be correlated with the change across contexts of regression coefficients to stimulus features (Fig. 6b). Our analyses therefore predict a specific form of multiplicative interactions, or non-linear mixed selectivity to stimulus features and context cues[14], but also imply that the two sub-populations can be identified based on their selectivity to context (Fig. 6b).

The multiplicative interaction between context and stimulus selectivity is a necessary, but not a sufficient condition for implementing context-dependent responding. A second, necessary component of the computational mechanism is that each sub-population integrates dominantly one of the two features into the latent dynamics, as seen from the overlaps between the input vectors and the input-selection vectors (Fig. 5b right). This leads to a specific prediction for inactivation experiments: inactivating separately sub-populations defined by their selectivity to context disrupts performance in one context, while leaving the other intact (Fig. 6c–d). In contrast, inactivating a random subset of neurons leads only to an overall decrease in performance independently of the context (Fig. 6c–d).

We first tested the two predictions on networks constrained to be of minimal, unit rank, but in which clustering analyses in connectivity space revealed more than two sub-populations

(Extended Data Figure 5), as in Yang et al.[29]. The two predictions for selectivity and inactivations were therefore directly borne out for such networks (Fig. 6e). We next turned to networks trained without rank constraint, and tested the two predictions without analyzing connectivity, as would be the case in experimental studies. The two predictions were again borne out (Fig. 6f), confirming that key aspects of the computational mechanisms extend to networks in which the connectivity was of higher rank, and the dynamics higher dimensional.

Finally we examined unit-rank networks trained on the MDM task. Such networks received an input structure identical to the CDM task, consisting of two stimulus features and two context cues. In contrast to the CDM task, the networks were trained to average the two stimulus features, and contextual cues were irrelevant, so that a fully random population structure was sufficient to perform the task (Fig. 1l). We therefore expected that the two predictions made for the CDM task do not necessarily hold in this case. We indeed found that training networks on the MDM task led to weaker selectivity to context, and weaker correlation between context selectivity and the change in stimulus selectivity (Fig. 6g). Specific neurons still exhibited selectivity to contextual cues, but inactivating them led to changes in performance similar to inactivating a random subset of neurons (Fig. 6g). Importantly, this finding was unchanged when we matched the strength of context selectivity between MDM and CDM task by increasing the amplitude of contextual inputs (Extended Data Figure 10).

Altogether, identical context selectivity therefore led to opposite effects of inactivations across tasks, as predicted by our minimal-rank models.

## Implications for multi-tasking

A recent study reported that multiple populations emerge in networks trained simultaneously on multiple tasks, and can be repurposed across tasks[29]. Our results more specifically suggest that a multi-population structure in connectivity is needed only when an identical stimulus requires different outputs depending on the context set by the performed task. While this is the case in many multi-tasking situations, concurrent tasks are alternatively often based on different sets of stimuli[45;46;47]. Here we show that the reduced models developed by analyzing networks trained on individual tasks can be used to build networks that perform multiple tasks in parallel (Fig. 7). More specifically, multiple tasks on an identical set of stimuli can be performed by combining and repurposing multiple sub-populations, while in contrast multiple tasks on separate sets of stimuli can be performed with a single population by relying on dynamics in orthogonal subspaces[32;48]. As a result, when identical stimuli are processed, some individual neurons exhibit task-specialisation, while for separate sets of stimuli all neurons are multi-taskers, and contribute to multiple tasks in parallel. These findings are in direct agreement with the activity of neurons in the prefrontal cortex during flexible categorisation, which show specialisation when identical stimuli are processed[49], and multi-tasking when separate stimuli sets are used[45].

To illustrate task-specialization, we first consider a network that receives stimuli composed of two sensory features, and depending on a rule cue performs one out of three different tasks on them : perceptual decision-making on the first stimulus feature, perceptual decision-

making on the second stimulus feature, or integration of the two features as in the multi-sensory decision making task (Fig. 7a). This multi-tasking setup is in fact a direct extension of context-dependent decision-making, and we implemented it using a simplified network based on the CDM task, consisting of unit-rank connectivity with three separate sub-populations (Extended Data Figure 5). In that network, each sub-population has a well defined computational role. One of them plays the role of an evidence integrator, by endowing the latent dynamics with a long timescale through strong positive feedback. That population is repurposed across all tasks (Fig. 7c orange neuron), and inactivating it leads to performance degradation on all three tasks (Fig. 7b). The other two populations relay separately the two sensory features into the latent dynamics, as in the CDM task (Fig. 5b–d). Each of them participates in only two of the three tasks, as corroborated by changes in task performance after selective inactivations (Fig. 7b). Neurons belonging to these two populations are therefore specialised for specific tasks, as seen in their task-specific responses to stimuli (Fig. 7c green and purple neurons).

We next illustrate multi-tasking in a network that performs two tasks on distinct sets of stimuli, the perceptual decision-making (DM) and the parametric working-memory (WM) tasks (Fig. 7d). Such a network can be obtained by directly superposing the connectivity matrices $\boldsymbol{J}_{DM}$ and $\boldsymbol{J}_{WM}$ of two minimal networks of rank-one and two that perform the individual tasks with random population structure (Fig. 3). The resulting connectivity $\boldsymbol{J} = \boldsymbol{J}_{DM} + \boldsymbol{J}_{WM}$ is of rank three, and has a random population structure. The corresponding latent dynamics are based on a recurrent sub-space of dimension three, and the two tasks rely on two orthogonal subspaces with one dimension implementing the DM task, and the other two implementing the WM task (Fig. 7e). Because of the random population structure, each neuron is a random combination of collective variables corresponding to different tasks, so that all neurons display multi-tasking activity (Fig. 7f).

## Discussion

The goal of this study was to determine whether and when a non-random population structure is necessary for networks to perform a specific computation based on recurrent dynamics. To address this question, we first trained recurrent neural networks on a range of standard systems neuroscience tasks, and examined the emerging population structure in the selectivity and connectivity, and its relationship with the computations. We then identified underlying mechanisms by extracting the latent low-dimensional dynamics. Although a number of tasks could be implemented with random population structure in connectivity, we found that tasks based on flexible input-output mappings instead appeared to require an additional structure that could be accurately approximated in terms of a small number of sub-populations which played functionally distinct roles.

The starting motivation of this work was the apparent discrepancy between the experimental results of Ref.[17] and Ref.[10] (see also[11]). Analyzing neural activity in the rat posterior parietal cortex during a multi-sensory decision-making task, Ref.[17] found no evidence for non-random population structure in selectivity. Applying identical analyses to the prefrontal cortex, Ref.[10] instead identified population structure in activity during a more complex task that combined perceptual and value-guided decisions. Our results suggest that the difference

between tasks provides a possible explanation for these diverging conclusions. Examining networks trained on an abstracted version of the multi-sensory integration task of Ref.[17], we found that a non-random population structure was not needed. Implementing a full version of the task used in Ref.[10] would have required reinforcement learning that falls beyond the scope of the supervised methods for training networks used here. The core component of that task was however a flexible weighing of two sensory features depending on the context set by reward history. That requirement of context-dependent weighing of input streams is in fact identical to the context-dependent decision-making task, in which all-or-none weights were assigned to the two stimulus features depending on the contextual cues. The gain-modulation mechanism underlying networks that performed the CDM task can more generally assign graded weights to each feature as required for the task of Ref.[10]. This mechanism requires multiple populations, so that our analyses predict that a non-random population structure is needed for the task used in Ref.[10].

We found that in trained networks relying on a non-random population structure, connectivity could be accurately described by a small number of sub-populations. Mechanistically, the role of such a 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 that subspace[50]. Specifically, different inputs differentially activate different sub-populations, and shift the recurrent subspace into different regions of the state-space with different non-linear dynamical landscapes. A complementary picture emerges from the perspective of the effective circuits which describe the low-dimensional latent dynamics in terms of interactions between collective variables through effective couplings (Fig. 5). 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 additive 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[51], a type of processing closely related to flexible input-output tasks considered here. While patterns of gain modulation[52;38;53], and the distinction between drivers and modulators[37] are fundamentally physiological concepts (see Supplementary Discussion 1 for a physiological interpretation of sub-populations), here we found that an analogous mechanism emerges in abstract trained networks at the collective level of latent dynamics.

Previous studies have reported that when training networks on a given task, some aspects of the solutions are invariant[54] while others depend on the details of the implementation[29;32;55]. Our analyses confirmed these observations. Our main result for the computational requirement of non-random population structure in connectivity (Fig.1l) held independently of the details of the training, and in particular in absence of constraints on the rank of the network (Extended Data Figure 3). For tasks requiring a non-random population structure, the number of sub-populations needed to approximate connectivity however varied across networks (Extended Data Figure 5). For those tasks, our results show that a single global population is insufficient (see Supplementary Discussion 2 for the relation with the universal

approximation theorem) and that fundamental computational mechanisms are conserved across a range of different networks (Fig. 6). Our analyses however do not predict the specific dimensionality or number of populations to be expected. More systematic model selection could for instance be performed by further constraining recurrent neural networks based on recorded neural activity[23;56].

The fact that neurons are selective to mixtures of task variables rather than individual features has emerged as one of the defining properties of representations in higher order areas of the mammalian cortex[44]. Moving beyond a simple dichotomy between pure and mixed selectivity, recent studies argued that mixed selectivity does not necessarily preclude the presence of a population structure, and introduced the notion of non-random mixed selectivity[17;10]. Our results predict that the expected type of structure and mixed selectivity depends on the complexity of the performed task. In particular, for tasks requiring flexible input-output associations, we predict the presence of non-random population structure. The resulting non-random mixed-selectivity however becomes apparent only in response to specific combinations of variables, while selectivity to other variables can remain fully random (Fig. 6). Ultimately, as the task complexity is increased, identifying the signatures of computational mechanisms in the neural activity requires a careful comparison with computational models on a task-by-task basis.

## Methods

### Recurrent Neural 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 $i$-th unit, and $\phi(x_i) = \tanh(x_i)$ is its firing rate. Moreover, each neuron received a feed-forward input $I_i^{FF}$ and an independent white-noise input $\eta_i(t)$ specified below.

The recurrent connectivity is set by the connectivity matrix $\boldsymbol{J} = \{J_{ij}\}_{i,j=1\ldots N}$. For full-rank networks, the coefficients $J_{ij}$ were treated as independent parameters. For low-rank networks $\boldsymbol{J}$ was constrained to be of rank $R$, and parametrized as

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

i.e. $\boldsymbol{J}$ was a sum of $R$ outer-products 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 vectors*, with $\boldsymbol{m}^{(r)}$ the $r$-th output vector, and $\boldsymbol{n}^{(r)}$ the $r$-th input-selection vector. Without loss of generality, we will assume that all the output vectors (and respectively all the input-selection vectors) are mutually orthogonal. Such a representation is uniquely defined by the singular-value

decomposition of $\boldsymbol{J}$ by taking $\mathbf{m}^{(r)}$ to be the left singular vectors, and $\mathbf{n}^{(r)}$ the right singular vectors multiplied by the corresponding singular values.

The feed-forward inputs $I_i^{FF}(t)$ were generated by $N_{in}$ temporally-varying scalar stimuli $u_s(t)$, each fed into the unit $i$ 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 $\boldsymbol{I}^{(s)} = \{I_i^{(s)}\}_{i=1\ldots N}$ as the $s$-th *input vector*.

The output of the network was defined by a readout value

$$z = \frac{1}{N}\sum_{j=1}^{N} w_j \phi(x_j), \tag{9}$$

where $\boldsymbol{w} = \{w_i\}_{\mathrm{i=1...N}}$ is the *readout vector*.

The time constant of neurons was $\tau = 100$ms. For simulation and training, equation (6) was discretized using Euler's method with a time step $\Delta t = 20$ms. The white noise $\eta_i$ was simulated by drawing at each time step a random number from a centered Gaussian distribution of standard deviation 0.05.

For any pair of $N$-dimensional vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, the *overlap* $\sigma_{ab}$ was defined as the empirical covariance of their entries:

$$\sigma_{ab} = \frac{1}{N}\sum_{i=1}^{N} a_i b_i\,. \tag{10}$$

**Network training procedure—**We used backpropagation through time[57] 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 combinations of stimulus inputs to target readouts $\hat{z}$, and then stochastically generated trials. We minimized the mean squared error loss function

$$\mathscr{L} = \sum_{k,t} M_t \left(z_{k,t} - \hat{z}_{k,t}\right)^2 \tag{11}$$

where $z_{k,t}$ and $\hat{z}_{k,t}$ are respectively the actual, and the target readout values and the indices $k$, $t$ respectively run over 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 each trial, when the readouts were required to match their target values. For each task we also define a performance measure called accuracy, defined as the fraction of test trials for which the network output has the same sign as the expected output (i.e. $\mathrm{sign}\left(\sum_t M_t \hat{z}_{k,t}\right) = \mathrm{sign}\left(\sum_t M_t z_{k,t}\right)$)

For full-rank networks (Figs. 1,6) the gradients were computed with respect to individual entries $J_{ij}$ of the connectivity matrix. For results on full-rank networks in Fig. 1 (left column) and Extended Data Figure 3, matrices $\boldsymbol{J}$ were initialized with random independent Gaussian weights of mean 0 and variance $\rho = 1/N$. For the Extended Data Figure 3, we also trained networks whose weights were initialized with a variance $\rho = 0.1/N$, since these tend to be approximated more easily by low-rank networks[35].

For low-rank networks, we specifically looked for solutions in the subspace of connectivity matrices with rank $R$. The loss functions were therefore minimized by computing gradients with respect to the elements of connectivity vectors $\{\boldsymbol{m}^{(i)}\}_{i=1...R}$, $\{\boldsymbol{n}^{(i)}\}_{i=1...R}$. Unless specified otherwise in the description of individual tasks, we did not train the entries of input vectors $\{\boldsymbol{I}^{(s)}\}_{s=1...N_{in}}$ and the readout vectors $\{\boldsymbol{w}\}$ but only an overall amplitude factor for each input and readout vector. All vectors were initialized with their entries drawn from Gaussian distributions with zero mean and unit standard deviation, except for the readout 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[58] in pytorch[59] 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 pairs of connectivity vectors $R$ was treated as a hyper-parameter. We first trained full rank networks ($R = N$) and determined the accuracy with which they solved the task. We then started training rank $R = 5$ networks, and progressively decreased the rank until there was a sharp decrease in accuracy (Extended Data Figure 2). The minimal rank $R^*$ was defined for each task such that the accuracy at $R^*$ was at least of 95%.

To ease the clustering and resampling procedure, and approach mean-field solutions, we trained large networks (of sizes 512 neurons for the networks of figures 1 and 3, 4096 neurons for the context-dependent DM and DMS task networks of Figures 5 and Extended Data Figure 6, and 1024 neurons in figure 7).

## Definition of individual tasks

### Perceptual decision making (DM) 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 $T_{decision} = 20$ms.

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

$$I_i^{FF}(t) = I_i u^{(k)}(t) \tag{12}$$

where, during the stimulation period, $u^{(k)}(t) = \bar{u}^{(k)} + \xi^{(k)}(t)$, with $\xi^{(k)}(t)$ a zero-mean Gaussian white noise with standard deviation $\sigma_u = 0.1$. The mean stimulus $\bar{u}^{(k)}$ was drawn uniformly from $\pm 0.1 \times \{1, 2, 4\}$ on each trial. The elements $I_i$ of the input vector were generated from a Gaussian distribution with zero mean and unit standard deviation, and fixed during training.

During the decision epoch, the output $z$ was evaluated through a readout vector $\boldsymbol{w} = \{w_i\}_{i=1...N}$, 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. (11)) was defined as the sign of the mean input $\bar{u}(k)$.

### Parametric working memory (WM) task

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

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

$$I_i^{FF}(t) = I_i\left(u_1^{(k)}(t) + u_2^{(k)}(t)\right) \tag{13}$$

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 $\{10, 11, \ldots, 34\}$, and $f_{min}$ = 10 and $f_{max}$ = 34. The elements $I_i$ of the input vector were generated from a Gaussian distribution with zero mean and unit standard deviation, and fixed during the training.

During the decision epoch, the output $z$ was evaluated through a readout vector $\boldsymbol{w} = \{w_i\}_{i=1...N}$, 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. (11)) was defined as $\hat{z}_{(k)} = \frac{f_1^{(k)} - f_2^{(k)}}{f_{max} - f_{min}}$.

### Context-dependent decision making (CDM) task

***Trial structure.:*** A fixation epoch of duration $T_{fix}$ = 100ms was followed by a first context-only epoch of duration $T_{ctxt1}$ = 0ms for figure 1 and 350$ms$ for Figs. 4–6 plots, followed by a stimulation epoch of duration $T_{stim}$ = 800ms, a second context-only epoch of $T_{ctxt2}$ = 500ms, and a decision epoch of $T_{decision}$ = 20ms.

***Stimuli and outputs.:*** The feed-forward input to neuron $i$ on trial $k$ was

$$I_i^{FF}(t) = u_A^{(k)}(t)I_i^A + u_B^{(k)}(t)I_i^B + u_{ctxA}^{(k)}(t)I_i^{ctxA} + u_{ctxB}^{(k)}(t)I_i^{ctxB}. \tag{14}$$

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 +0.1 (or +0.5 for Figs. 4–6), 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 $\bar{u}_A^{(k)}$ and $\bar{u}_B^{(k)}$, 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 vectors were generated from a Gaussian distribution with zero mean and unit standard deviation on both populations. For the networks presented in the main text, input vectors were trained, while for the networks reported in Supplementary Note 2.3 all the input vectors were fixed throughout training.

During the decision epoch, on trial $k$ the target $\hat{z}_{(k)}$ in the loss function (Eq. (11)) was defined as the sign of the mean $\bar{u}_X^{(k)}$ of feature $X = A$ or $B$ for which the contextual cue was activated, i. e. $u_{ctx}^{(k)} = 1$. The readout vector was fixed throughout training.

### Multi-sensory decision making (MDM) task

***Trial structure.:*** A fixation epoch of duration $T_{fix} = 100$ms was followed by a context-only period of duration $T_{ctx} = 350$ms, a stimulation epoch of duration $T_{stim} = 800$ms, a delay epoch of duration $T_{delay} = 300$ms and a decision epoch of duration $T_{decision} = 20$ms.

***Inputs and outputs.:*** The feed-forward input to neuron $i$ on trial $k$ had the same structure as for the context-dependent decision-making task, and was given by:

$$I_i^{FF}(t) = u_A^{(k)}(t) I_i^A + u_B^{(k)}(t) I_i^B + u_{ctxA}^{(k)}(t) I_i^{ctxA} + u_{ctxB}^{(k)}(t) I_i^{ctxB}. \quad (15)$$

where the two stimulus inputs $u_A^{(k)}(t)$ and $u_B^{(k)}(t)$ represent two sensory modalities, and $u_{ctxA}^{(k)}$ and $u_{ctxB}^{(k)}$ are contextual inputs. In this task, the contextual inputs were irrelevant for the output, and we included them as a control. The inputs $u_A^{(k)}(t)$ and $u_B^{(k)}(t)$ were generated as for the CDM task, with the difference that on each trial the two inputs provided congruent evidence for the output, i.e. their means were of the same sign.

Specifically in each trial a sign $s_k \in \{-1, 1\}$ is generated randomly, as well as a modality that can be A, B, or AB. Then if the modality is A or AB, a mean $\bar{u}_A^{(k)}$ is chosen from $0.1 \times s_k \times \{1, 2, 4\}$ 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. A contextual input signal $u_{ctxA}^{(k)}(t)$ is set to 0.1 from the beginning of the contextual period to the end of the trial. If the modality is B, then the signal $u_A^{(k)}(t)$ is only equal to the zero-centered gaussian white noise. The signals $u_B^{(k)}(t)$ and $u_{ctxB}^{(k)}(t)$ are set in a similar manner. During the decision epoch, the target $\hat{z}^{(k)}$ is the underlying common sign $s_k$.

The networks received input signals through input vectors $\boldsymbol{I}^A$, $\boldsymbol{I}^B$, $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$ which were trained, and output was read through a readout vector $\boldsymbol{w}$ which was fixed throughout training.

### Delayed-match-to-sample task

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

***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 vectors $\boldsymbol{I}^A$ and $\boldsymbol{I}^B$, so that the feed-forward input to neuron $i$ on trial $k$ was:

$$I_i^{FF}(t) = u_A^{(k)}(t) I_i^A + u_B^{(k)}(t) I_i^B \tag{16}$$

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. (11)) was equal to +1 if the same stimulus was received in both stimulation epochs and −1 otherwise.

**Regression analyses and selectivity space—**We used multivariate linear regression to predict time-averaged neural firing rates $r_i = \phi(x_i)$ from task variables, using a linear model :

$$\boldsymbol{r}_i = \boldsymbol{X}\boldsymbol{\beta}_i + \epsilon_i \,. \tag{17}$$

Here $\boldsymbol{r}_i = \{r_{i,1}, \ldots, r_{i,K}\}$ is a vector containing the time-averaged firing rates of neuron $i$ in trials 1 to $K$, $\boldsymbol{X}$ is the design matrix where rows correspond to different trials and columns correspond to $D$ task variables such as stimulus, context and decision in each condition (defined below for each task), $\boldsymbol{\beta}_i$ is a $D$-by-1 vector of regression coefficients, and $\boldsymbol{\epsilon}_i$ is a $K$-by-1 vector of residuals.

The regression coefficients defined the *selectivity space* (Fig. 1a–d) of dimension $D$ where each axis corresponded to the regression coefficient with respect to one task variable, and each neuron was represented as point $\beta_i$. The choice of task variables and window of time-averaging of firing rates depended on the task:

- For the DM task, two regressions were performed on different time windows, leading to $D$ = 2 two coefficients per neuron: a regression of average firing rate during the first 100ms of stimulation period against mean stimulus which defined the coefficient $\beta_i^{stim}$ and a regression of average firing rate during the decision period against network choice which defined the coefficient $\beta_i^{choice}$. This was done to avoid ill-conditioning due to correlations between choice and stimulus.
- For the WM task, the mean firing rate during the decision period was regressed against both $f_1$ and $f_2$, leading to $D$ = 2 two coefficients per neuron.

- For the MDM task and the CDM task, the average firing rate during the stimulation period was regressed against both mean stimulus features $\bar{u}_A^{(k)}$ and $\bar{u}_B^{(k)}$ and both contextual input signals $u_{ctxA}^{(k)}$ and $u_{ctxB}^{(k)}$, leading to $D = 4$ coefficients per neuron, $\beta_i^A$, $\beta_i^B$, $\beta_i^{ctxA}$ and $\beta_i^{ctxB}$ In Fig. 6, the selectivity to context was characterized by a single regression coefficient $\beta_i^{ctx}$ obtained by regressing the absolute value of the firing rate $|r_i|$, averaged over the pre-stimulus period where only the contextual cues are non-zeros, against a regressor $X$ that takes the value +1 in context A and −1 in context B. The context selectivity is extracted through the linear model for $K$ trials

$$|\boldsymbol{r}_i| = \boldsymbol{X}\beta_i^{ctx} + \boldsymbol{\epsilon} \tag{18}$$

  In order to characterize the changes in selectivity with context, we substracted the pre-stimulus firing rate to the firing rate averaged over the first 100ms of stimulus presentation, and regressed this quantity against $\bar{u}_A^{(k)}$ and $\bar{u}_B^{(k)}$ separately in each context to obtain the regression coefficients $\beta_{ctxA,i}^A$, $\beta_{ctxA,i}^B$, $\beta_{ctxB,i}^A$, $\beta_{ctxB,i}^B$. The change in selectivity is then given by

$$\Delta_{ctx}\beta_i^{A/B} = |\beta_{ctxA,i}^{A/B}| - |\beta_{ctxB,i}^{A/B}| \tag{19}$$

  In Fig. 6 the analysis is presented for feature A, similar results are obtained for feature B (not shown).

- For the DMS task, the average firing rate during the decision period was regressed against both first and second stimulus identity (with $X_{k,s} = 1$ if stimulus $s$ is A in trial $k$, 0 otherwise, $s \in \{0,1\}$), leading to $D = 2$ regression coefficients per neuron.

**Connectivity space—**For a low-rank network, the connectivity is specified by $2R+N_{in}+1$ parameters for each neuron, corresponding to its entries $\{\{n_i^{(r)}\}_{r=1\ldots R}, \{m_i^{(r)}\}_{r=1\ldots R}, \{I_i^{(s)}\}_{s=1\ldots N_{in}}, w_i\}$ on the input, connectivity and output vectors. The connectivity of each neuron can therefore be represented as a point in a space of dimension $2R + N_{in} + 1$ that we term *connectivity space*. For each network, the distribution of points in this space is analysed for randomness in Figure 1, and used in the resampling procedures. Our mean-field theory shows that in the limit of large networks, the distribution of points in this space determines the low-dimensional latent dynamics of the network (see Analysis of latent dynamics in low-rank networks).

**ePAIRS analysis—**To statistically assess the presence of non-random population structure in the selectivity and connectivity spaces of trained networks, we implemented a version of the ePAIRS statistical test[10], which is itself derived from the PAIRS test developed in[17]. We consider a point cloud $\mathbf{X} = (X_{ij})_{1\leq i\leq N, 1\leq j\leq d}$, where the rows $\mathbf{x}_i$ corresponds to different

points (here neurons) and columns correspond to different axes of the considered space (regression coefficients to different variables in the selectivity space, entries of different input, connectivity and readout vectors in the connectivity space), which is centered by removing the mean ( so that for each $j, \sum_i X_{ij} = 0$). The ePAIRS test examines the directional distribution of points, i.e. the empirical distribution of $\mathbf{x}_i/\|\mathbf{x}_i\|$, and determines whether it is statistically distinguishable from the null distribution generated by a multivariate Gaussian with a covariance matrix identical to the covariance of $\mathbf{X}$. A significant outcome indicates of the ePAIRS test that the empirical distribution presents multiple ”preferred” directions incompatible with a Gaussian.

More specifically, the analysis proceeds as follows:

1. For each point $\mathbf{x}_i$, we determine its $l$ nearest neighbors in terms of the cosine metric (*ie.* the $l$ points for which $\cos(\widehat{\mathbf{x}_i\mathbf{x}_j}) = \mathbf{x}_i^T\mathbf{x}_j/(\|\mathbf{x}_i\|\|\mathbf{x}_j\|)$ are the highest, $l$ being a hyperparameter set to 3 in our case).

2. For each neuron, we compute the mean angle $\alpha_i$ with its $l$ nearest neighbors, defining an empirical distribution $\hat{p}_{data}(\alpha)$.

3. To generate the corresponding null distribution, a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is fit to the cloud of points $\mathbf{X}$, with $\mathbf{\Sigma}$ the empirical covariance of $\mathbf{X}$, computed as $\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$. Then the steps 1–2 are applied on 500 samples of the multivariate Gaussian with the same number $N$ of data points to compute a Monte-Carlo null distribution $\hat{p}_{null}(\alpha)$.

4. Finally, the difference between the data and the null distributions is assessed using a two-sided Wilcoxon’s rank-sum test, giving a p-value, and the effect size $c$ is computed as

$$c = \frac{\mu_{null} - \mu_{data}}{\sigma_{null}}, \tag{20}$$

where $\mu$ and $\sigma$ represent the means and standard deviations of $\hat{p}_{null}(\alpha)$ and $\hat{p}_{data}(\alpha)$ An effect size $c > 0$ indicates that angles between neighbors are smaller in the data than in the resampled point clouds, meaning that points are more highly clustered than expected. On the contrary, $c < 0$ indicates that points are more regularly dispersed than expected from random.

**Resampling and clustering trained networks—**For a given trained network, we first fitted a single multivariate Gaussian to its connectivity distribution by computing the empirical covariance matrix (matrix of size $(N_{in} + 2R + 1)^2$). We then generated networks by resampling connectivity parameters from this distribution, and examined their performance (Fig. 1i and Extended Data Figure 3). In all trained networks we examined, the empirical means were close to zero, and we neglected them.

For the CDM and DMS tasks, we performed a clustering analysis in the connectivity space by fitting multivariate mixtures of Gaussians with an increasing number of clusters, and

by resampling from the obtained distributions until we found networks that were able to optimally perform the task, as defined by an accuracy higher than 95% for at least 95% of the sampled networks. We used variational inference with a gaussian prior for the mean with a precision equal to $10^5$ to enforce a zero-mean constraint for all components of the mixtures, and a Dirichlet process prior for the weights with concentration 1 divided by number of components, using the model `BayesianGaussianMixture` of the package scikit-learn[60].

Since the inference and resampling processes are susceptible to finite-size fluctuations, for the DMS task in Extended Data Figure 6 we complemented the clustering with some retraining of the covariance matrices found for each component. For this we developed a class of Gaussian mixture, low-rank RNNs, in which the covariance structure of each population is trainable. Directly training the covariance matrices is difficult given that they need to be symmetric definite positive; we therefore used a trick akin to the reparametrization trick used in variational auto-encoders[61]: the set of input, connectivity and readout vectors were defined as a linear transformation of a basis of i.i.d. normal vectors, such that for any connectivity vector $\boldsymbol{a}$:

$$a_i = (\boldsymbol{b}_a^{(p)})^T \boldsymbol{X}_i, \tag{21}$$

where $p$ is the population index of neuron $i$ (sampled from a categorical distribution with weights $\{\alpha_p\}_{p=1...P}$ i.i.d. derived by the variational inference), $\boldsymbol{X}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbb{1})$ are random normal vectors of dimension $N_{in} + 2R + 1$, 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 Supplementary Note 1 for more details). We then trained the vectors $\boldsymbol{b}_v^{(p)}$, with the population indices being sampled only once, and the $\boldsymbol{X}_i$ being resampled at each training epoch.

**Analysis of latent dynamics in low-rank networks—**Here we provide an overview of the reduction of low-rank networks to low-dimensional latent dynamics. A more complete derivation can be found in[36]. For simplicity, we consider the noise free case ($\eta_i(t) = 0$ in Eq. (6)), and we assume the initial condition $x_i = 0$ at $t = 0$ for all $i = 1 \ldots N$.

**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 $i$. When the connectivity is constrained to be of low rank, the dynamics are restricted to a low-dimensional subspace of this state-space[30]. Indeed, inserting Eqs. (7) and (8) into Eq. (6), leads to

$$\tau \frac{dx_i}{dt} = -x_i + \frac{1}{N} \sum_{r=1}^{R} m_i^{(r)} \sum_{j=1}^{N} n_j^{(r)} \phi(x_j) + \sum_{s=1}^{N_{in}} I_i^{(s)} u_s(t)\,. \tag{22}$$

At any time $t$, the right-hand-side is confined to the linear subspace spanned by the vectors $\{\boldsymbol{m}^{(r)}\}_{r=1\ldots R}$ and $\{\boldsymbol{I}^{(s)}\}_{s=1\ldots N_{in}}$ Since we assumed $x_i = 0$ at $t = 0$, the dynamics of $\boldsymbol{x}(t) = \{x_i(t)\}_{i=1\ldots N}$ remain in that subspace for all $t$. The activation vector $\boldsymbol{x}$ can therefore be expressed in terms of $R$ internal collective variables $\kappa_r$, and $N_{in}$ external collective variables $v_s$:

$$\boldsymbol{x}(t) = \sum_{r=1}^{R} \kappa_r(t)\boldsymbol{m}^{(r)} + \sum_{s=1}^{N_{in}} v_s(t)\boldsymbol{I}_{\perp}^{(s)}. \tag{23}$$

The first term on the right-hand side in Eq. (23) represents the component of the activity on the *recurrent space*[25;20] defined as the subspace spanned by the output connectivity vectors $\{\boldsymbol{m}^{(r)}\}_{r=1\ldots R}$. The corresponding internal collective variables $\kappa_r$ are defined as projections of the activation vector $\boldsymbol{x}$ on the $\boldsymbol{m}^{(r)}$:

$$\kappa_r(t) = \frac{1}{\left\|\boldsymbol{m}^{(r)}\right\|^2} \sum_{j=1}^{N} m_j^{(r)} x_j(t). \tag{24}$$

The second term on the right-hand side in Eq. (23) represents the component of the activity on the *input space* defined as the sub-space spanned by $\{\boldsymbol{I}_{\perp}^{(s)}\}_{s=1\ldots N_{in}}$, 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 $\boldsymbol{I}_{\perp}^{(s)}$:

$$v_s(t) = \frac{1}{\left\|\boldsymbol{I}_{\perp}^{(s)}\right\|^2} \sum_{j=1}^{N} I_{\perp,j}^{(s)} x_j(t). \tag{25}$$

The dimensionality of the dynamics in state 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 $N_{in}$ of the input space.

The dynamics of the internal variables $\kappa_r$ are obtained by projecting Eq. (6) onto the output connectivity vectors $\boldsymbol{m}^{(r)}$:

$$\tau \frac{d\kappa_r}{dt} = -\kappa_r(t) + \kappa_r^{rec}(t) + \frac{1}{\left\|\boldsymbol{m}^{(r)}\right\|^2} \sum_{j=1}^{N} m_j^{(r)} \sum_{s=1}^{N_{in}} I_j^s u_s(t) \tag{26}$$

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

$$\kappa_r^{rec}(t) = \frac{1}{N} \sum_{j=1}^{N} n_j^{(r)} \phi\left(x_j(t)\right). \tag{27}$$

Inserting Eq. (23) into $\kappa_r^{rec}$ leads to a closed set of equations for the $\kappa_r$:

$$\kappa_r^{rec}(t) = \frac{1}{N}\sum_{j=1}^{N} n_j^{(r)} \phi\left(\sum_{r'=1}^{R} \kappa_r'(t) m_j^{(r')} + \sum_{s=1}^{N_{in}} I^s_{\perp,j} v_s(t)\right). \tag{28}$$

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

$$\tau \frac{dv_s}{dt} = -v_s + u_s\,. \tag{29}$$

Throughout the main text, we assume for simplicity that the stimuli $u_s$ vary on a timescale slower than $\tau$, and replace $v_s$ with $u_s$. We also assume throughout the main text that input vectors are orthogonal to the output connectivity vectors, ie. $\boldsymbol{I}^{(s)} = \boldsymbol{I}_\perp^{(s)}$ for all $s$. Hence the third term on the r.h.s. of equation (26) equals zero.

Using Eq. (23), the readout value $z$ can be expressed in terms of the collective variables as

$$z(t) = \frac{1}{N}\sum_{j=1}^{N} w_j \phi\left(\sum_{r'=1}^{R} \kappa_r'(t) m_j^{(r')} + \sum_{s=1}^{N_{in}} I^s_{\perp,j} v_s(t)\right). \tag{30}$$

**Connectivity space and mean-field limit:** The dynamics of the collective variables are fundamentally determined by the components of connectivity and input vectors through Eq. (28). Neuron $i$ is therefore characterized by the $2R + N_{in} + 1$ parameters

$$\{\{n_i^{(r)}\}_{r=1\ldots R}, \{m_i^{(r)}\}_{r=1\ldots R}, \{I_i^{(s)}\}_{s=1\ldots N_{in}}, w_i\}\,. \tag{31}$$

Each neuron can thus be represented as a point in the *connectivity space* of dimension $2R + N_{in} + 1$, 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. (28) consists of a sum of $N$ terms, where the term $j$ contains only the connectivity parameters of neuron $j$. The connectivity parameters of different neurons therefore do not interact in $\kappa_r^{rec}$, so that the r.h.s of Eq. (28) can be interpreted as an average over the set of points corresponding to all neurons in the connectivity space.

Our main assumption will be that in the limit of large networks ($N \to \infty$), the set of points in the connectivity space is described by a probability distribution $P\left(n^{(1)}, \ldots, n^{(R)}, m^{(1)}, \ldots, m^{(R)}, I^{(1)}, \ldots, I^{(N_{in})}, w\right) := P(\underline{n}, \underline{m}, \underline{I}, w)$. In this mean-field limit, the r.h.s. of Eq. (28) becomes:

$$\kappa_r^{rec}(t) = \int \mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I}\, \mathrm{d}w\; P(\underline{n}, \underline{m}, \underline{I}, w) n^{(r)} \phi\left( \sum_{r'=1}^{R} \kappa_{r'}(t) m^{(r')} + \sum_{s'=1}^{N_{in}} I_{\perp}^{(s')} v_{s'}(t) \right), \quad (32)$$

where we have used the shorthand $\mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I} = \prod_{r'=1}^{R} \prod_{s'=1}^{N_{in}} \left( \mathrm{d}m^{(r')} \mathrm{d}n^{(r')} \mathrm{d}I^{(s')} \right)$. The collective dynamics are therefore fully specified by the single-neuron distribution of connectivity parameters. 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.

The joint distribution of connectivity parameters $P(\underline{n}, \underline{m}, \underline{I}, w)$ also determines the values of the readout:

$$z(t) = \int \mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I}\, \mathrm{d}w\; P(\underline{n}, \underline{m}, \underline{I}, w) w \phi\left( \sum_{r'=1}^{R} \kappa_{r'}(t) m^{(r')} + \sum_{s'=1}^{N_{in}} I_{\perp}^{(s')} v_{s'}(t) \right). \quad (33)$$

**Statistics of connectivity and sub-populations:** To approximate any arbitrary joint distributions of connectivity parameters $P(\underline{n}, \underline{m}, \underline{I}, \underline{w})$, we used multivariate Gaussian mixture models (GMMs). This choice was based on the following considerations: (i) GMMs are able to approximate an arbitrary multivariate distribution[62]; (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[36].

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 parameters $\{\{n_i^{(r)}\}_{r=1\ldots R}, \{m_i^{(r)}\}_{r=1\ldots R}, \{I_i^{(s)}\}_{s=1\ldots N_{in}}, w_i\}$ 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 $2R+N_{in}+1$, and $\boldsymbol{\Sigma}_p$ is a covariance matrix of size $(2R + N_{in} + 1)^2$. The full distribution of connectivity parameters is therefore given by

$$P(\underline{n}, \underline{m}, \underline{I}, \underline{w}) = \sum_{p=1}^{P} \alpha_p \mathcal{N}\left(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p\right) \quad (34)$$

$$:= \sum_{p=1}^{P} \alpha_p P_p(\underline{n}, \underline{m}, \underline{I}, \underline{w}) \quad (35)$$

where the coefficients $\alpha_p$ define the fraction of neurons belonging to each sub-population.

Each sub-population directly corresponds to a Gaussian cluster of points in the connectivity 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 connectivity parameters $a$ and $b$, with $a, b \in \{\{n^{(r)}\}_{r=1\ldots R}, \{m^{(r)}\}_{r=1\ldots R}, \{I^{(s)}\}_{s=1\ldots N_{in}}, w\}$. Note that because the output vectors $\boldsymbol{m}^{(i)}$ (resp. input-selection vectors $\boldsymbol{n}^{(i)}$) are mutually orthogonal, the covariances between the parameters $\{m_i^{(r)}\}_{r=1\ldots R}$ (respectively $\{n_i^{(r)}\}_{r=1\ldots R}$ vanish.

Since every neuron belongs to a single population, the r.h.s of Eq. (28) can be split into P terms, each corresponding to an average over one population. As within each population the distribution is a joint Gaussian, Eq. (32) becomes a sum of $P$ Gaussian integrals

$$\kappa_r^{rec}(t) = \sum_{p=1}^{P} \alpha_p \int \mathrm{d}\underline{m}\, \mathrm{d}\underline{n}\, \mathrm{d}\underline{I}\, \mathrm{d}w\; P_p(\underline{n}, \underline{m}, \underline{I}, w \\ )n^{(r)}\phi\left(\sum_{r'=1}^{R} \kappa_{r'}(t) m^{(r')} + \sum_{s'=1}^{N_{in}} I_{\perp}^{(s')} v_{s'}(t)\right). \tag{36}$$

**Effective circuit description of latent dynamics:** In the following, we focus on zero-mean multivariate Gaussian mixture distributions for the connectivity parameters, and input vectors orthogonal to $\{\boldsymbol{m}^{(i)}\}_{i=1\ldots R}$, 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[36]. Using Stein's lemma for Gaussian distributions, the dynamics of the internal collective variables can be expressed as a dynamical system (see Supplementary Note 1)

$$\frac{d\kappa_r}{dt} = -\kappa_r + \sum_{r'=1}^{R} \tilde{\sigma}_{n^{(r)}m^{(r')}} \kappa_{r'} + \sum_{s=1}^{N_{in}} \tilde{\sigma}_{n^{(r)}I^{(s)}} v_s\,. \tag{37}$$

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 $\tau$.

In Eq. (37), $\tilde{\sigma}_{n^{(r)}m^{(r)}}$ represents the effective self-feedback of the collective variable $\kappa_r$, $\tilde{\sigma}_{n^{(r)}m^{(r')}}$ sets the interaction between the collective variables $\kappa_r$ and $\kappa_r'$, and $\tilde{\sigma}_{n^{(r)}I^{(s)}}$ is the effective coupling between the input $u_s$ and $\kappa_r$. These effective interactions between the internal variables are given by weighted averages over populations

$$\tilde{\sigma}_{ab} = \sum_{p=1}^{P} \alpha_p \sigma_{ab}^{(p)} \langle \Phi' \rangle_p \tag{38}$$

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

$$\langle \Phi' \rangle_p = \langle \Phi' \rangle \left(\Delta^{(p)}\right) \tag{39}$$

with

$$\langle \Phi' \rangle(\Delta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}z e^{-z^2/2} \phi'(\Delta z) \tag{40}$$

and

$$\Delta^{(p)} = \sqrt{\sum_{r'=1}^{R} \left(\sigma^{(p)}_{m(r')}\right)^2 \kappa_{r'}^2 + \sum_{s=1}^{N_{in}} \left(\sigma^{(p)}_{I(s)}\right)^2 v_s^2} \tag{41}$$

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

In Eq. (37), the covariances $\sigma_{ab}^{(p)}$ are set by the statistics of the connectivity and input vectors, but the gain factors $\langle \Phi' \rangle_p$ in general depend both on internal and external collective variables $\kappa_k$ and $v_j$. As a consequence, the dynamics in Eq. (37) is non-linear, and in fact it can be shown that given a sufficient number of sub-populations, the right-hand side in Eq. (37) can approximate any arbitrary dynamical system[36].

In the special case of linear networks (i.e. $\Phi(x) = x$), the gain is constant so that the effective couplings $\tilde{\sigma}_{ab}$ in Eq. 38 are equal to the overlaps $\sigma_{ab}$ of vectors $a$ and $b$ over the full population, as defined in Eq. 10. The population structure therefore only plays a role for non-linear networks.

The value of the readout (Eq. (33)) can also be expressed in terms of effective interactions as

$$z = \sum_{r'=1}^{R} \tilde{\sigma}_{m(r')w} \kappa_{r'} + \sum_{s=1}^{N_{in}} \tilde{\sigma}_{I(s)w(k)} v_s \,. \tag{42}$$

**Drivers and modulators of latent dynamics:** Eq. (37) shows that feed-forward inputs to the network can have two distinct effects on the collective dynamics of internal variables $\kappa_r$. If the input vector $I^{(s)}$ overlaps with the $r$-th input-selection vector $n^{(r)}$, i.e. the corresponding covariance $\sigma^{(p)}_{n(r)I(s)}$ is non-zero for population $p$, the input directly drives the latent dynamics, in the sense that $v_s$ acts as an effective external input to the dynamics of $\kappa_r$ in Eq. (37).

In contrast, when all covariances between the input vectors and the input selection vectors are zero (i.e. $\sigma^{(p)}_{n(r)I(s)} = 0$ for all $r$, $p$), the corresponding input does not drive the latent dynamics, but can still modulate them by modifying the gain through Eq. (41) if the variance $\sigma^{(p)}_{I(s)}$ of the input on some population $p$ is non-zero.

The inputs can therefore play roles of drivers and modulators of latent dynamics, depending on whether the corresponding input vectors overlap or not with the input selection vectors $n^{(r)}$.

## Reduced models of latent dynamics for individual tasks

**Perceptual decision making task:** We found that computations in the rank-one, single population trained networks could be reproduced by a reduced model with two non-zero covariances $\sigma_{nI}$ and $\sigma_{nm}$ (Sup. Fig. S2a). For this reduced model, the dynamics of the internal collective variable is given by

$$\frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{nm}\kappa + \tilde{\sigma}_{nI}v(t), \tag{43}$$

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. (39), and the effective population variance $\Delta$ given by:

$$\Delta = \sqrt{\sigma_m^2\kappa^2 + \sigma_I^2 v^2}\,. \tag{44}$$

Here $v(t)$ corresponds to the integrated input $u(t)$, see Eq. (29).

An analysis of nonlinear dynamics defined by Eq. (43) showed that adjusting these parameters was sufficient to implement the task, as additional parameters only modulate the overall gain. In particular the value of $\sigma_{nm}$ determines the qualitative shape of the dynamical landscape on which the internal variable evolves and sets the timescale on which it integrates inputs (see Supplementary Note 2.1for more details).

**Parametric working memory task:** 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 $\sigma_n{}^{(2)}{}_I$ (Sup. Fig. S3a). In particular covariances $\sigma_n{}^{(1)}{}_m{}^{(2)}$, $\sigma_n{}^{(2)}{}_m{}^{(1)}$ across the two vectors could be set to zero without performance impairment. For this reduced model, the dynamics of the two internal collective variables is given by:

$$\begin{aligned}\frac{d\kappa_1}{dt} &= -\kappa_1 + \tilde{\sigma}_n{}^{(1)}{}_m{}^{(1)}\kappa_1 + \tilde{\sigma}_n{}^{(1)}{}_I v(t)\\ \frac{d\kappa_2}{dt} &= -\kappa_2 + \tilde{\sigma}_n{}^{(2)}{}_m{}^{(2)}\kappa_2 + \tilde{\sigma}_n{}^{(2)}{}_I v(t)\end{aligned} \tag{45}$$

where $\tilde{\sigma}_{ab} = \sigma_{ab}\langle\Phi'\rangle(\Delta)$ with $\langle\Phi'\rangle(\Delta)$ defined in Eq. (39), and the effective noise $\Delta$ given by:

$$\Delta = \sqrt{(\sigma_m{}^{(1)})^2\kappa_1^2 + (\sigma_m{}^{(2)})^2\kappa_2^2 + \sigma_I^2 v(t)^2}\,. \tag{46}$$

Here $v(t)$ corresponds to the integrated input $u(t)$, see Eq. (29).

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 $\kappa_1$ 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 $\kappa_2$, such that the first stimulus is forgotten during the delay and that the second stimulus is represented during the decision epoch (see Supplementary Note 2.2 for more details).

**Context-dependent decision making task:** We found that the computations in the unit rank, two populations network relied on the following conditions for the covariances in the two populations (Sup. Fig. S4a): (i) $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$ were essentially orthogonal to the input-selection vector **n**, implying that $\sigma^{(p)}_{nI^{ctxA}} \simeq 0$ and $\sigma^{(p)}_{nI^{ctxB}} \simeq 0$ for both populations $p$ = 1,2; (ii) on each population, each of the two input-selection vectors was correlated with only one of the input-feature vectors, 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 vector 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

$$\frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{nm}\kappa + \tilde{\sigma}_{nI^A}v_A(t) + \tilde{\sigma}_{nI^B}v_B(t) \tag{47}$$

with effective couplings

$$\tilde{\sigma}_{nI^A} = \frac{1}{2}\sigma^{(1)}_{nI^A}\langle\Phi'\rangle_1 \tag{48}$$

$$\tilde{\sigma}_{nI^B} = \frac{1}{2}\sigma^{(2)}_{nI^B}\langle\Phi'\rangle_2 . \tag{49}$$

The averaged gains for each population were given by equations (40), with the standard deviations of currents onto each population

$$\begin{aligned}\Delta^{(1)} &= \sqrt{\left(\sigma^{(1)}_m\right)^2\kappa^2 + \left(\sigma^{(1)}_{I^A}\right)^2 v_A^2 + \left(\sigma^{(1)}_{I^B}\right)^2 v_B^2 + \left(\sigma^{(1)}_{I^{ctxB}}\right)^2 c_B^2} \\ \Delta^{(2)} &= \sqrt{\left(\sigma^{(2)}_m\right)^2\kappa^2 + \left(\sigma^{(2)}_{I^A}\right)^2 v_A^2 + \left(\sigma^{(2)}_{I^B}\right)^2 v_B^2 + \left(\sigma^{(2)}_{I^{ctxA}}\right)^2 c_A^2} .\end{aligned} \tag{50}$$

Here $v_A(t)$ and $v_B(t)$ correspond to the integrated inputs $u_A(t)$ and $u_B(t)$, see Eq. (29).

As for the perceptual decision making task, the value of $\sigma_{mn}$ determines the qualitative shape of the dynamical landscape on which the internal variable evolves and sets the timescale on which it integrates inputs. Large values of the variances $\sigma^{(1)}_{I^{ctxtB}}$ and $\sigma^{(2)}_{I^{ctxtA}}$

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 Supplementary Note 2.3 for more details).

**Delayed-match-to-sample task:** We found that the computations in the rank-two, two population network relied on the following conditions for the covariances in the two populations (Sup. Fig. S5a): (i) on one population, the two connectivity modes were coupled through $\sigma^{(1)}_{n^{(1)}m^{(2)}}$, $\sigma^{(1)}_{n^{(2)}m^{(1)}} \neq 0$, with a specific condition on their values to induce a limit cycle (that the difference $|\sigma^{(1)}_{n^{(1)}m^{(2)}} - \sigma^{(1)}_{n^{(2)}m^{(1)}}|$ is large, see Supplementary Text 4 and[30;36]); (ii) on the other population, the covariances were in contrast set to counter-balance the first population, and cancel the rotational dynamics $\sigma^{(2)}_{n^{(1)}m^{(2)}} \simeq -\sigma^{(1)}_{n^{(1)}m^{(2)}}$ and $\sigma^{(2)}_{n^{(2)}m^{(1)}} \simeq -\sigma^{(1)}_{n^{(2)}m^{(1)}}$; (iii) the input-selection and output vectors for the second connectivity mode on the second population had a strong overlap $\frac{1}{2}\sigma^{(2)}_{n^{(2)}m^{(2)}} > 1$ that led to strong positive feedback; (iv) the input vectors $I^A$ had a strong variance on population 2, $\sigma^{(2)}_{I^A} \gg 1$ while other input sub-vectors had small variances $\sigma^{(1)}_{I^A}$, $\sigma^{(1)}_{I^B}$,$\sigma^{(2)}_{I^B} \simeq 0$.

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

$$\begin{aligned}\frac{d\kappa_1}{dt} &= -\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(1)}m^{(2)}}\kappa_2 \\ \frac{d\kappa_2}{dt} &= -\kappa_2 + \tilde{\sigma}_{n^{(2)}m^{(1)}}\kappa_1 + \tilde{\sigma}_{n^{(2)}m^{(1)}}\kappa_2 + \tilde{\sigma}_{n^{(2)}I^A}\upsilon_A + \tilde{\sigma}_{n(2)I^B}\upsilon_B,\end{aligned} \tag{51}$$

with the effective couplings mediating inputs

$$\tilde{\sigma}_{n^{(2)}I^A} = \frac{1}{2}\sigma^{(2)}_{n^{(2)}I^A}\langle\Phi'\rangle_2 \tag{52}$$

$$\tilde{\sigma}_{n^{(2)}I^B} = \frac{1}{2}\sigma^{(2)}_{n^{(2)}I^B}\langle\Phi'\rangle_2, \tag{53}$$

and effective couplings governing the autonomous dynamics:

$$\tilde{\sigma}_{n^{(1)}m^{(1)}} = \frac{1}{2}\sigma^{(1)}_{n^{(1)}m^{(1)}}\langle\Phi'\rangle_1 \tag{54}$$

$$\tilde{\sigma}_{n^{(1)}m^{(2)}} = \frac{1}{2}\sigma^{(1)}_{n^{(1)}m^{(2)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(1)}m^{(2)}}\langle\Phi'\rangle_2 \tag{55}$$

$$\tilde{\sigma}_{n^{(2)}m^{(1)}} = \frac{1}{2}\sigma^{(1)}_{n^{(2)}m^{(1)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(2)}m^{(1)}}\langle\Phi'\rangle_2 \tag{56}$$

$$\tilde{\sigma}_{n^{(2)}m^{(2)}} = \frac{1}{2}\sigma^{(1)}_{n^{(2)}m^{(2)}}\langle\Phi'\rangle_1 + \frac{1}{2}\sigma^{(2)}_{n^{(2)}m^{(2)}}\langle\Phi'\rangle_2 . \quad (57)$$

The average gains are given by Eq. (40), with standard deviations of currents onto each population

$$\begin{aligned}\Delta^{(1)} &= \sqrt{(\sigma^{(1)}_{m^{(1)}})^2\kappa_1^2 + (\sigma^{(1)}_{m^{(2)}})^2\kappa_2^2 + (\sigma^1_{IA})^2 v_A^2} \\ \Delta^{(2)} &= \sqrt{(\sigma^{(2)}_{m^{(1)}})^2\kappa_1^2 + (\sigma^{(2)}_{m^{(2)}})^2\kappa_2^2} .\end{aligned} \quad (58)$$

Here $v_A(t)$ and $v_B(t)$ correspond to the integrated inputs $u_A(t)$ and $u_B(t)$, see Eq. (29).

Conditions (i) to (iv) on the covariances allow to implement the dynamical landscape modulation of Extended Data Figure 9f (see Sup. Fig. S5d). When stimulus A is present ($u_A = 1$), the gain of population 2 is set to $\langle\Phi'\rangle_2 \simeq 0$ because of $\sigma^{(2)}_{IA} \gg 1$ (see Eq. (58)), and the specific values of covariances for sub-vectors in population 1 induce a limit cycle (see Supplementary Note 2.5). In absence of inputs, or when input B was present, gains were approximately equal for the two populations (Sup. Fig. S5c), 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.

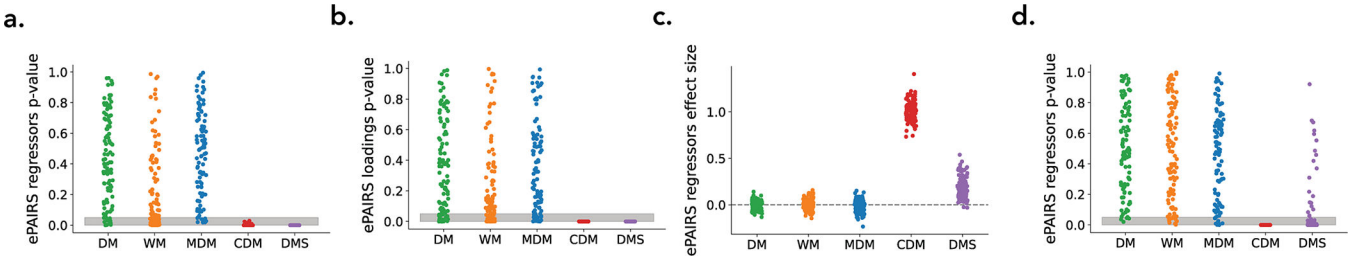
## Code availability

All codes used to train models and generate figures are available at https://github.com/adrian-valente/populations_paper_code.

## Data availability

Trained models are available at https://github.com/adrian-valente/populations_paper_code.

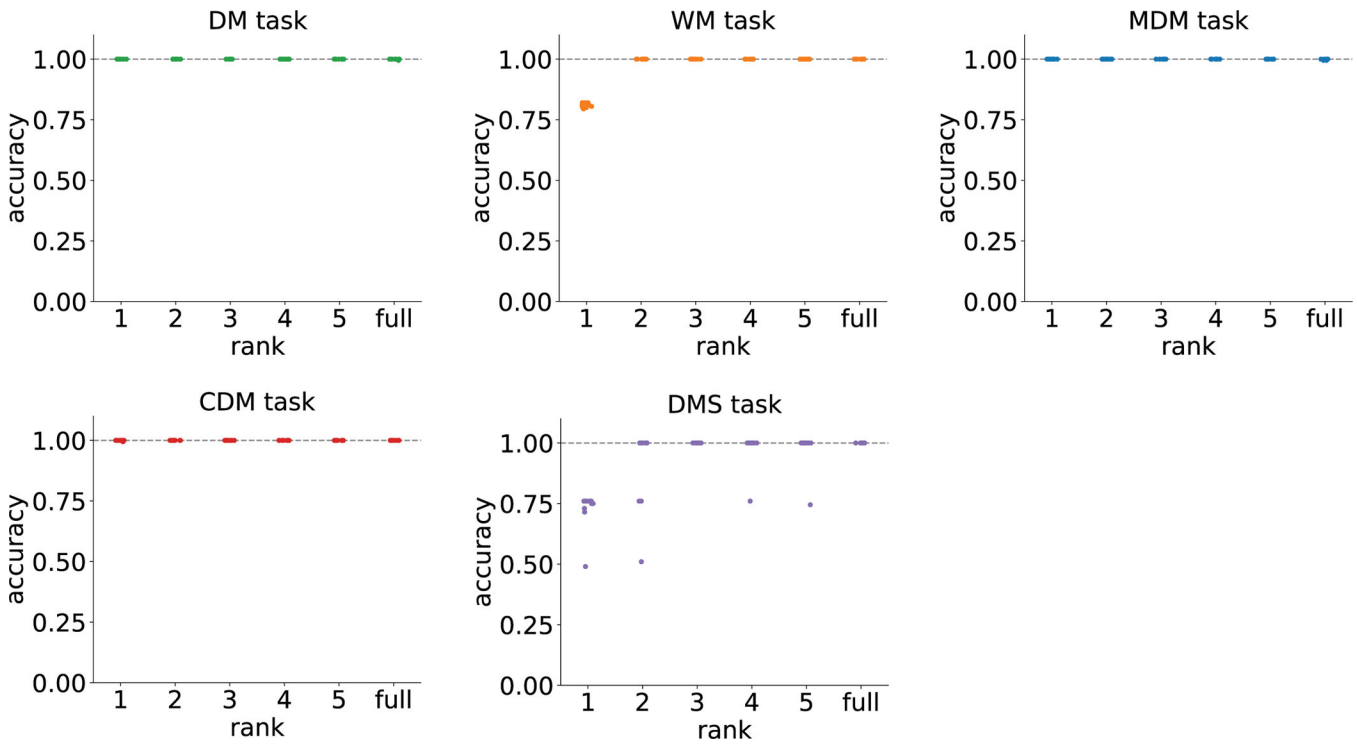
# Extended Data




**Extended Data Figure 1. Additional ePAIRS results.**

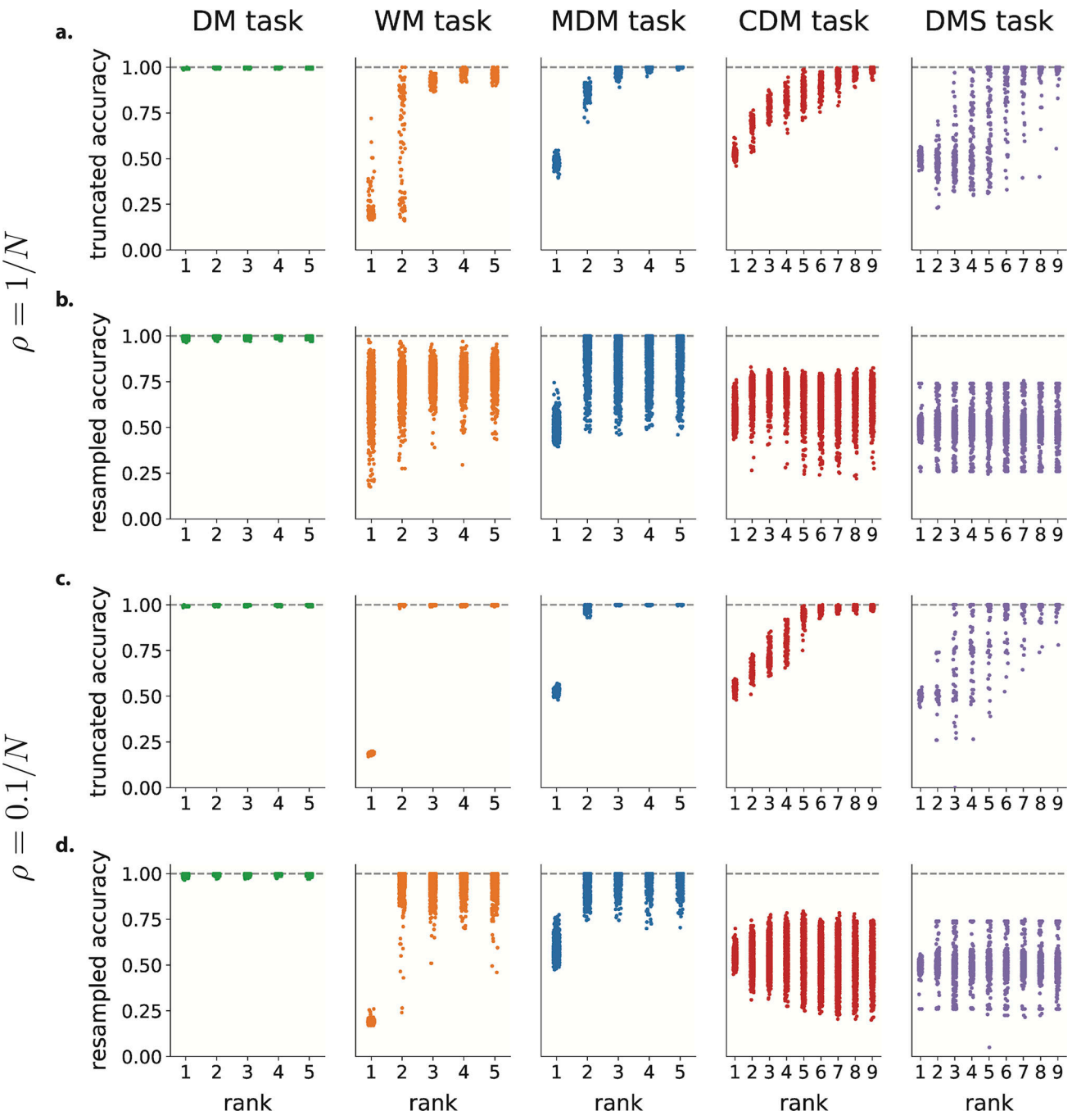
(a) p-values given by the ePAIRS test on selectivity spaces for the full-rank networks displayed in Fig. 1d (two-sided ePAIRS test, 100 networks per task, $n = 512$ neurons for each network). (b) p-values given by the ePAIRS test on connectivity spaces for the low-rank networks displayed in Fig 1h (two-sided ePAIRS test, 100 networks per task, $n = 512$

neurons for each network). (c) ePAIRS effect sizes on the selectivity space for the same low-rank networks (two-sided ePAIRS test, 100 networks per task, $n = 512$ neurons for each network). (d) Corresponding ePAIRS p-values.




**Extended Data Figure 2. Determination of the minimal rank for each task.**

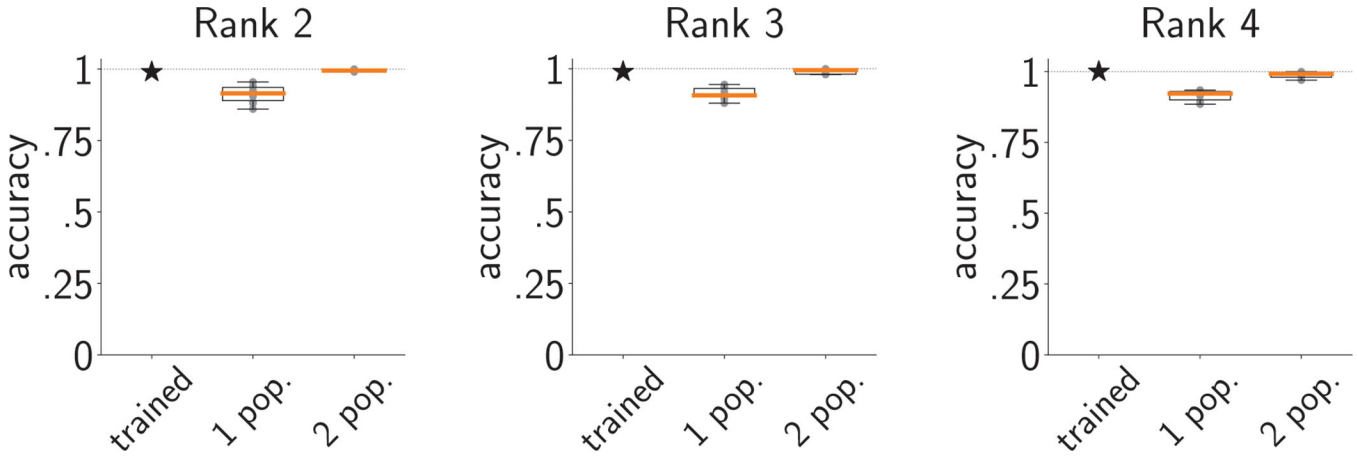
For each task and each rank $R$ between 1 and 5, ten rank-$R$ networks were trained with different random initial connectivity. For each task, a panel displays the performance of trained networks as function of their rank.




**Extended Data Figure 3. Analysis of trained full-rank networks.**

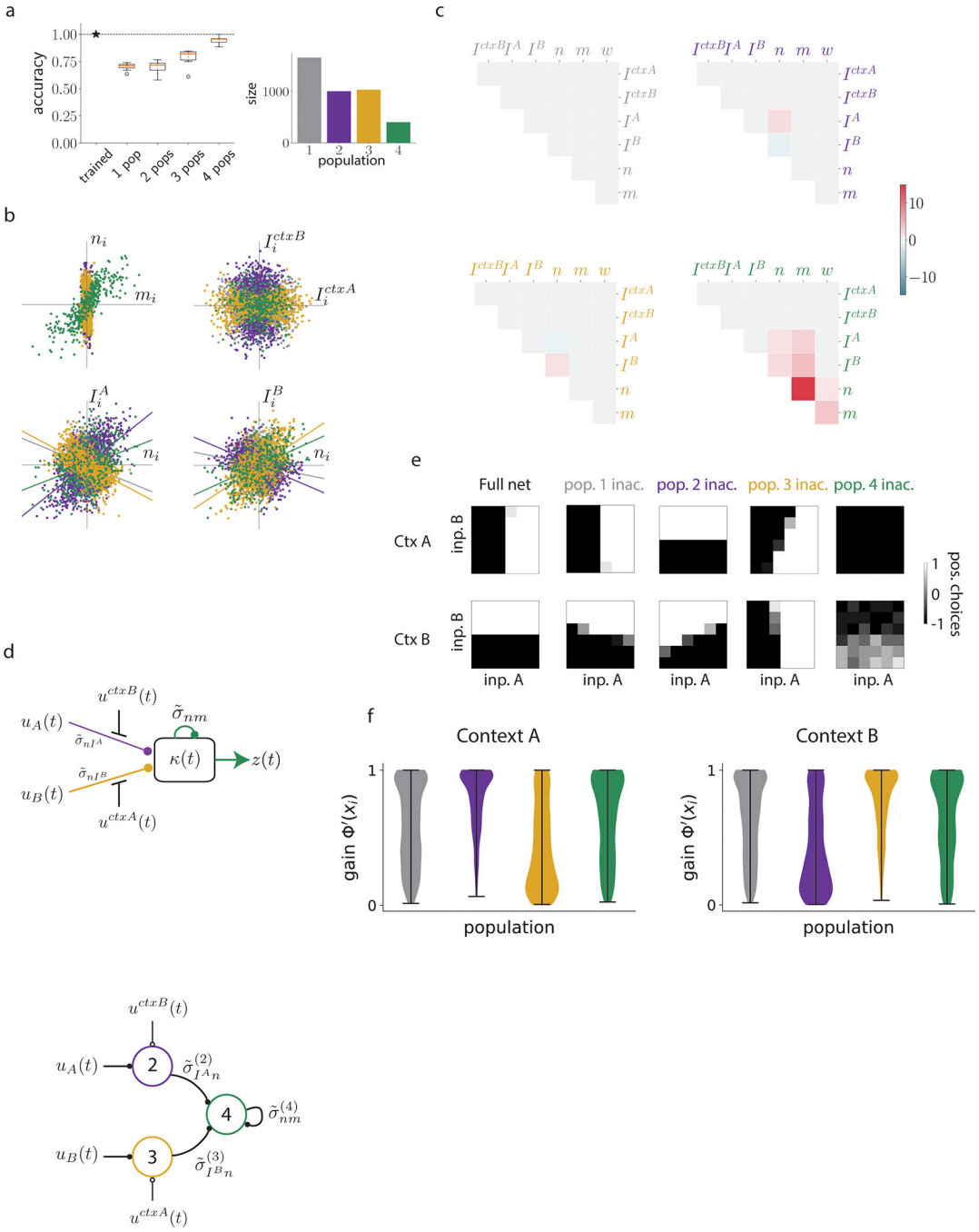
(a)-(b) Analysis of full-rank networks trained with initial connectivity weights of variance $1/N$ (100 networks for each task). (a) Performance of truncated-rank networks. Following[35], we extract from full-rank networks the learned part of the connectivity $\mathbf{\Delta J} = \boldsymbol{J} - \boldsymbol{J_0}$ defined as the difference between the final connectivity $\boldsymbol{J}$ and the initial connectivity $\boldsymbol{J_0}$. We then truncate $\mathbf{\Delta J}$ to a given rank via singular value decomposition, and add it back to $\boldsymbol{J_0}$. For each task, a panel displays the performance of the obtained networks as function of the rank used for the truncation. (b) Resampling analysis of truncated networks. Starting

from the truncated networks in (a) we fit multivariate Gaussians to the distribution of their $\mathbf{\Delta}\boldsymbol{J}$ in the corresponding connectivity spaces. We then generate new networks by resampling from this distribution, as done on the trained low-rank networks for Fig. 1i–l. For each task, a panel displays the performance of the obtained resampled network as function of the rank used for the truncation. (c)-(d) Same analyses as (a)-(b) for sets of networks trained with initial connectivity weights of variance $0.1/N$ (100 networks for each task, for DMS 49/100 networks that had an accuracy < 95% after training and were ignored). Networks with weaker initial connectivity are better approximated by their resampled low-rank connectivity. This is due to the fact that larger initial connectivities induce correlations between $\mathbf{\Delta}\boldsymbol{J}$ and $\boldsymbol{J_0}$[35]. The resampling destroys both this correlation and the population structure, leading to performance impairments even when the population structure is potentially irrelevant.




**Extended Data Figure 4. Increasing the rank maintains the requirement for population structure.**

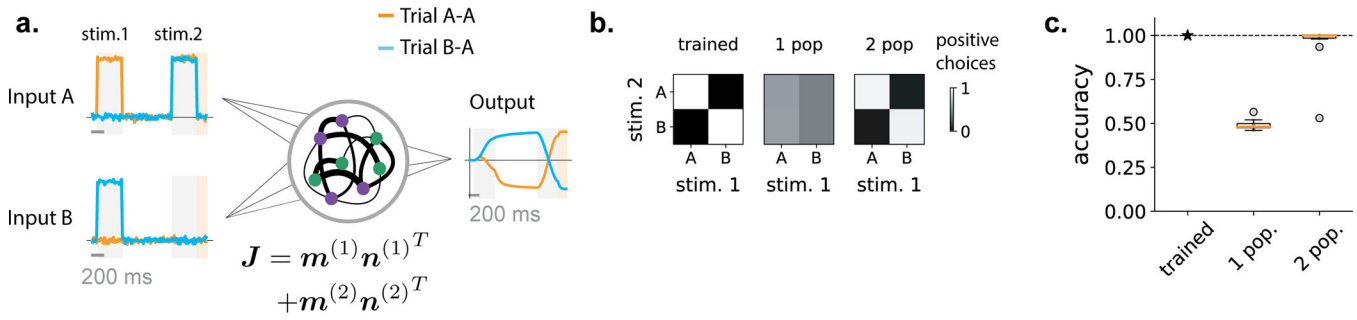
For this figure we have trained low-rank networks with a rank higher than 1 on the CDM task, fitted a single Gaussian or a mixture of 2 Gaussians to the obtained connectivity space, and retrained the obtained distribution (Methods ) to obtain resampled networks with a performance as high as possible. Even with this additional layer of retraining of the fitted distributions (which is only present in the main text for the DMS task) the obtained single-population networks fell short of performing the CDM task with a good accuracy. Here, 10 draws of a single network for each combination of rank and number of populations are shown (line: median, box: quartiles, whiskers: range, in the limit of median ± 1.5 interquartile range, points: outliers).




**Extended Data Figure 5. Alternative implementation of the CDM task.**

A network trained with different hyperparameters offers an example of an alternative solution for the CDM task, using 3 effective population and a fourth one accounting for neurons that are not involved in the task (see Supplementary Text 3)). (a) Left: for each number of sub-populations, a boxplot shows the performance of 10 networks with connectivity resampled from a Gaussian Mixture Model (GMM) fitted to the trained network (line: median, box: quartiles, whiskers: range, in the limit of median ± 1.5 interquartile range, points: outliers). Right: for the GMM with four sub-populations, size

of each component found by the clustering procedure. (b) Four 2d projections of the 7-dimensional connectivity space. (c) Upper-right triangle of the empirical covariance matrices for each of the four populations. (d) Illustration of the mechanism used by the network at the level of latent dynamics. Populations 2 – 4 control one effective coupling each, indicated by the matching color. (e) Psychometric matrices similar to those shown in Fig. 4 after inactivation of each sub-population. (f) Violin plots showing the gain distributions of neurons in each of the four sub-populations in each context.

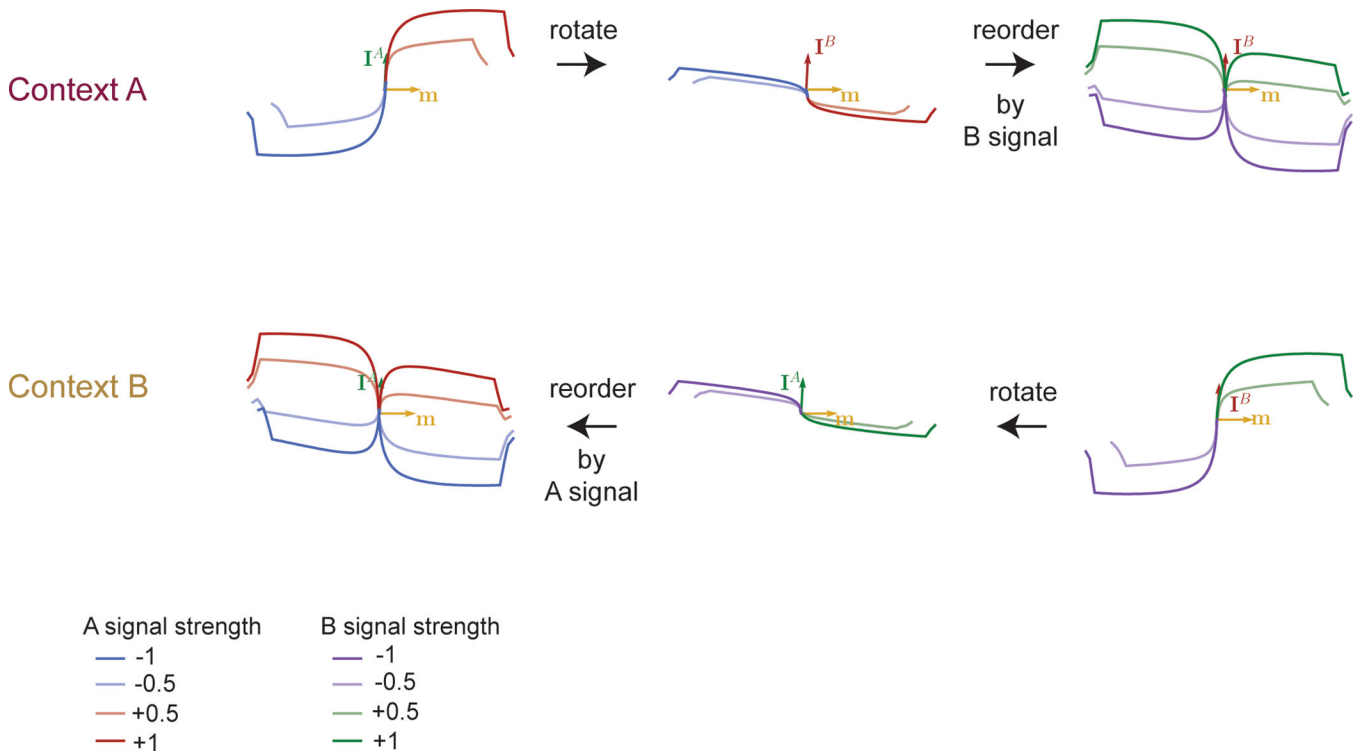



**Extended Data Figure 6. Multi-population analysis of networks performing the delayed match-to-sample (DMS) task.**

(a) Networks received a sequence of two stimuli during two stimulation periods (in light gray) separated by a delay. Each stimulus belonged to one out two categories (A or B), each represented by a different input vector. Rank-two networks were trained to produce an output during a response period (in light orange) with a positive value if the two stimuli were identical, and a negative value otherwise. Here we illustrate two trials with stimuli A-A and B-A respectively. (b) Psychometric response matrices. Fraction of positive responses for each combination of first and second stimuli, for a trained network (left) and for networks generated by resampling connectivity from a single population (middle) or two populations (right). (c) Average accuracy of a trained network and for 10 draws of resampled single-population and two-population networks (line: median, box: quartiles, whiskers: range, in the limit of median ± 1.5 interquartile range, points: outliers).

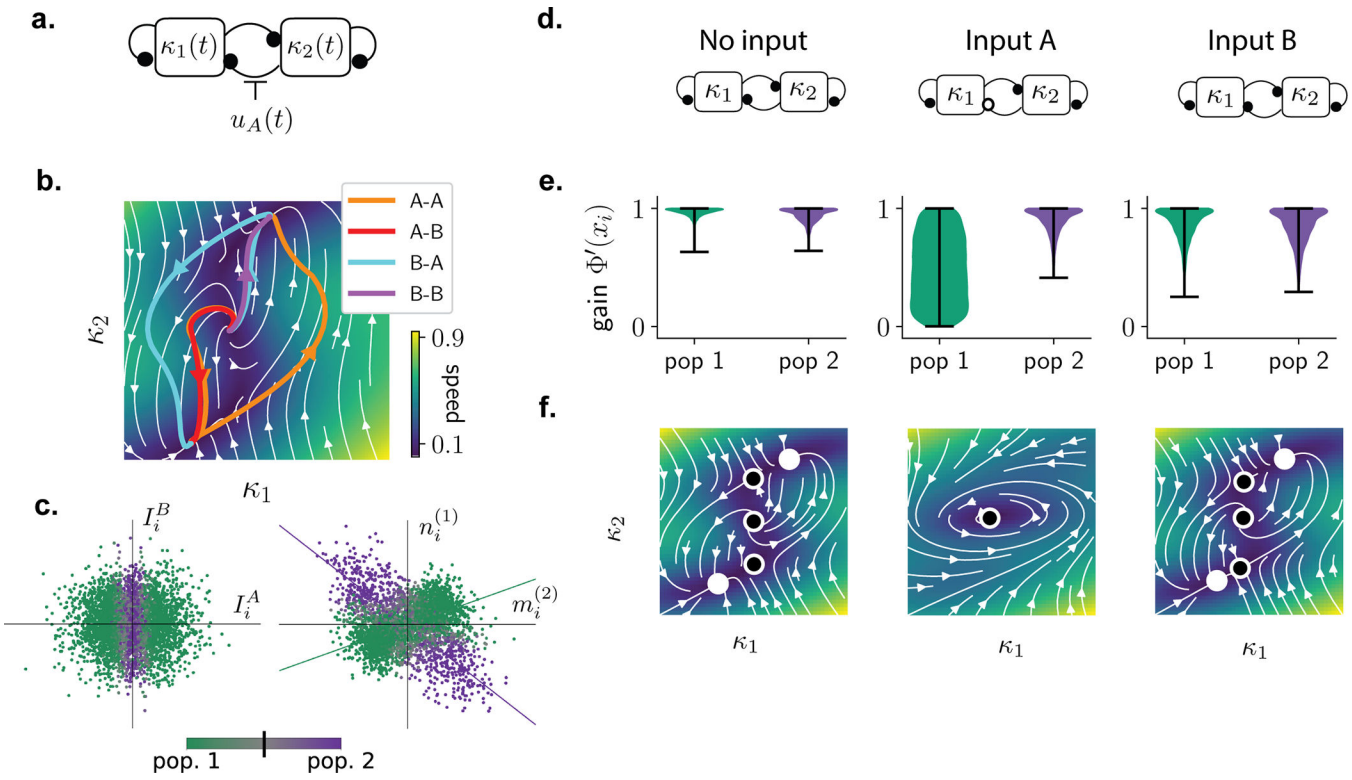

**Extended Data Figure 7. Statistics of connectivity in trained networks.**

Upper left corner of the empirical covariance matrix between connectivity vectors for networks trained on each task, after clustering neurons in two populations for tasks CDM and DMS. These covariance matrices are then used for resampling single-population and two-population networks that successfully perform each task.



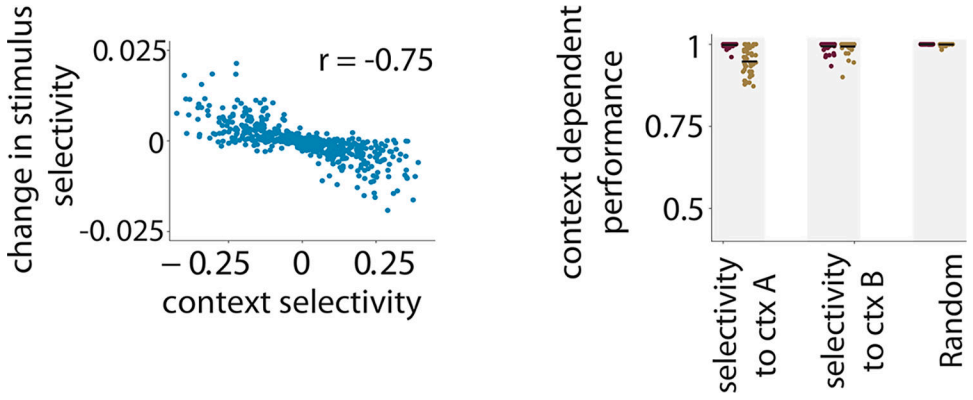

**Extended Data Figure 8. Context-dependent decision-making state-space dynamics.**

Here we reproduce figures akin to those presented in[15] for the trained low-rank network used in figures 4 and 5. We generate 32 conditions corresponding to different combinations of context, signal A coherence and signal B coherence and then project condition-averaged trajectories either on the plane spanned by the recurrent connectivity vector $\boldsymbol{m}$ (which corresponds to the choice axis) and the input vector $\boldsymbol{I}^A$, or on the $\boldsymbol{m} - \boldsymbol{I}^B$ plane. Similarly to what was observed in[15], signal A strength is encoded along the $\boldsymbol{I}^A$ axis, even when it is irrelevant (lower left corner), and signal B strength is encoded along the $\boldsymbol{I}^B$ axis, even when it is irrelevant (top right corner).




**Extended Data Figure 9. Low-dimensional latent dynamics in networks performing the delayed match-to-sample (DMS) task.**

(a) Circuit diagram representing latent dynamics for a minimal network trained on the DMS task (Eq. 51). The network was of rank two, so that the latent dynamics were described by two internal variables $\kappa_1$ and $\kappa_2$. Input A acts as a modulator on the recurrent interactions between the two internal variables. (b) Dynamical landscape for the autonomous latent dynamics in the $\kappa_1 - \kappa_2$ plane (*ie.* the $\boldsymbol{m}^{(1)}$-$\boldsymbol{m}^{(2)}$ plane). Colored lines depict trajectories corresponding to the 4 types of trials in the task (see Sup. Fig. S6 for details of trajectories). Background color and white lines encode the speed and direction of the dynamics in absence of inputs. (c) Two 2d projections of the seven-dimensional connectivity space, with colors indicating the two sub-populations and lines corresponding to linear regressions for each of them on the right panel. (d) Effective circuit diagrams in absence of inputs (left), and when input A (middle) or input B (right) are present (see Supplementary Note 2.4). Filled circles denote positive coupling, open circles negative coupling. Input A in particular induces a negative feedback from $\kappa_2$ to $\kappa_1$. (e) Distribution of neural gains for each populations (pop. 1: $n = 3050$, pop. 2: $n = 1046$), in the three situations described above. The gain of population 1 (green) is specifically modulated by input A. (f) Dynamical landscapes in the 3 situations described above (see Methods). Filled and empty circles indicate respectively stable and unstable fixed points. The negative feedback induced by input A causes a limit cycle to appear in the latent dynamics.




**Extended Data Figure 10. Control for the strength of context cues in the MDM task.**

Here the context input vectors have been multiplied by a factor five compared to the network analyzed in Fig. 6g. (a) Context cues are thus able to set the functioning point of some neurons closer to the saturating part of the transfer function, leading to the observation of non-linear mixed-selectivity between context and changes in sensory representation with context. (b) As opposed to the CDM task, this particular feature of selectivity is not functional as revealed by specifically inactivating neurons with a high selectivity to context A or B, showing a similar decrease in behavioral performance as for randomly selected neurons.

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgements

The project was supported by the NIH Brain Initiative project U01-NS122123 (SO, AV), ANR project MORSE (ANR-16-CE37-0016) (SO, FM), the CRCNS project PIND (SO, AD, MB), the program “Ecoles Universitaires de Recherche” launched by the French Government and implemented by the ANR, with the reference ANR-17-EURE-0017 (SO). There are no competing interests. SO thanks Joshua Johansen and Bijan Pesaran for fruitful discussions.

## References

[1]. Barack David L and Krakauer John W. Two views on the cognitive brain. Nature Reviews Neuroscience, pages 1–13, 2021.

[2]. Hubel David H and Wiesel Torsten N. Receptive fields of single neurones in the cat’s striate cortex. The Journal of physiology, 148(3):574–591, 1959. [PubMed: 14403679]

[3]. Moser Edvard I, Moser May-Britt, and McNaughton Bruce L. Spatial representation in the hippocampal formation: a history. Nature neuroscience, 20(11):1448, 2017. [PubMed: 29073644]

[4]. Hardcastle Kiah, Maheswaranathan Niru, Ganguli Surya, and Giocomo Lisa M. A multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex. Neuron, 94(2):375–387, 2017. [PubMed: 28392071]

[5]. Adesnik Hillel, Bruns William, Taniguchi Hiroki, Huang Z Josh, and Scanziani Massimo. A neural circuit for spatial summation in visual cortex. Nature, 490(7419):226–231, 2012. [PubMed: 23060193]

[6]. Ye Li, Allen William E, Thompson Kimberly R, Tian Qiyuan, Hsueh Brian, Ramakrishnan Charu, Wang Ai-Chi, Jennings Joshua H, Adhikari Avishek, Halpern Casey H, et al. Wiring and

molecular features of prefrontal ensembles representing distinct experiences. Cell, 165(7):1776–1788, 2016. [PubMed: 27238022]

[7]. Kvitsiani D, Ranade S, Hangya B, Taniguchi H, Huang JZ, and Kepecs A. Distinct behavioural and network correlates of two interneuron types in prefrontal cortex. Nature, 498(7454):363–366, 2013. [PubMed: 23708967]

[8]. Hangya Balázs, Pi Hyun-Jae, Kvitsiani Duda, Ranade Sachin P, and Kepecs Adam. From circuit motifs to computations: mapping the behavioral repertoire of cortical interneurons. Current opinion in neurobiology, 26:117–124, 2014. [PubMed: 24508565]

[9]. Pinto Lucas and Dan Yang. Cell-type-specific activity in prefrontal cortex during goal-directed behavior. Neuron, 87(2):437–450, 2015. [PubMed: 26143660]

[10]. Hirokawa Junya, Vaughan Alexander, Masset Paul, Ott Torben, and Kepecs Adam. Frontal cortex neuron types categorically encode single decision variables. Nature, 576(7787):446–451, 2019. [PubMed: 31801999]

[11]. Hocker David L, Brody Carlos D, Savin Cristina, and Constantinople Christine M. Sub-populations of neurons in lofc encode previous and current rewards at time of choice. eLife, 10:e70129, 2021. [PubMed: 34693908]

[12]. Churchland Mark M and Shenoy Krishna V. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. Journal of neurophysiology, 97(6):4235–4257, 2007. [PubMed: 17376854]

[13]. Machens Christian K, Romo Ranulfo, and Brody Carlos D. Functional, but not anatomical, separation of “what” and “when” in prefrontal cortex. Journal of Neuroscience, 30(1):350–360, 2010. [PubMed: 20053916]

[14]. Rigotti Mattia, Barak Omri, Warden Melissa R, Wang Xiao-Jing, Daw Nathaniel, Miller Earl K, and Fusi Stefano. The importance of mixed selectivity in complex cognitive tasks. Nature, 497(7451):585–590, 2013. [PubMed: 23685452]

[15]. Mante Valerio, Sussillo David, Shenoy Krishna V, and Newsome William T. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature, 503(7474):78–84, 2013. [PubMed: 24201281]

[16]. Park Il Memming, Meister Miriam LR, Huk Alexander C, and Pillow Jonathan W. Encoding and decoding in parietal cortex during sensorimotor decision-making. Nature neuroscience, 17(10):1395, 2014. [PubMed: 25174005]

[17]. Raposo David, Kaufman Matthew T, and Churchland Anne K. A category-free neural population supports evolving demands during decision-making. Nature neuroscience, 17(12):1784, 2014. [PubMed: 25383902]

[18]. Buonomano Dean V and Maass Wolfgang. State-dependent computations: spatiotemporal processing in cortical networks. Nature Reviews Neuroscience, 10(2):113–125, 2009. [PubMed: 19145235]

[19]. Gallego Juan A, Perich Matthew G, Miller Lee E, and Solla Sara A. Neural manifolds for the control of movement. Neuron, 94(5):978–984, 2017. [PubMed: 28595054]

[20]. Remington Evan D, Narain Devika, Hosseini Eghbal A, and Jazayeri Mehrdad. Flexible sensorimotor computations through rapid reconfiguration of cortical dynamics. Neuron, 98(5):1005–1019, 2018. [PubMed: 29879384]

[21]. Saxena Shreya and Cunningham John P. Towards the neural population doctrine. Current opinion in neurobiology, 55:103–111, 2019. [PubMed: 30877963]

[22]. Vyas Saurabh, Golub Matthew D, Sussillo David, and Shenoy Krishna V. Computation through neural population dynamics. Annual Review of Neuroscience, 43:249–275, 2020.

[23]. Rajan Kanaka, Harvey Christopher D, and Tank David W. Recurrent network models of sequence generation and memory. Neuron, 90(1):128–142, 2016. [PubMed: 26971945]

[24]. Chaisangmongkon Warasinee, Swaminathan Sruthi K, Freedman David J, and Wang Xiao-Jing. Computing by robust transience: how the fronto-parietal network performs sequential, category-based decisions. Neuron, 93(6):1504–1517, 2017. [PubMed: 28334612]

[25]. Wang Jing, Narain Devika, Hosseini Eghbal A, and Jazayeri Mehrdad. Flexible timing by temporal scaling of cortical responses. Nature neuroscience, 21(1):102–110, 2018. [PubMed: 29203897]

[26]. Sohn Hansem, Narain Devika, Meirhaeghe Nicolas, and Jazayeri Mehrdad. Bayesian computation through cortical latent dynamics. Neuron, 103(5):934–947, 2019. [PubMed: 31320220]

[27]. Sussillo David. Neural circuits as computational dynamical systems. Current opinion in neurobiology, 25: 156–163, 2014. [PubMed: 24509098]

[28]. Barak Omri. Recurrent neural networks as versatile tools of neuroscience research. Current opinion in neurobiology, 46:1–6, 2017. [PubMed: 28668365]

[29]. Yang Guangyu Robert, Joglekar Madhura R, Song H Francis, Newsome William T, and Wang Xiao-Jing. Task representations in neural networks trained to perform many cognitive tasks. Nature neuroscience, 22 (2):297–306, 2019. [PubMed: 30643294]

[30]. Mastrogiuseppe Francesca and Ostojic Srdjan. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. Neuron, 99(3):609–623, 2018. [PubMed: 30057201]

[31]. Sakai Katsuyuki. Task set and prefrontal cortex. Annu. Rev. Neurosci, 31:219–245, 2008. [PubMed: 18558854]

[32]. Duncker Lea, Driscoll Laura, Shenoy Krishna V, Sahani Maneesh, and Sussillo David. Organizing recurrent network dynamics by task-computation to enable continual learning. Advances in Neural Information Processing Systems, 33, 2020.

[33]. Masse Nicolas Y, Grant Gregory D, and Freedman David J. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. Proceedings of the National Academy of Sciences, 115 (44):E10467–E10475, 2018.

[34]. Schuessler Friedrich, Dubreuil Alexis, Mastrogiuseppe Francesca, Ostojic Srdjan, and Barak Omri. Dynamics of random recurrent networks with correlated low-rank structure. Physical Review Research, 2(1): 013111, 2020.

[35]. Schuessler Friedrich, Mastrogiuseppe Francesca, Dubreuil Alexis, Ostojic Srdjan, and Barak Omri. The interplay between randomness and structure during learning in rnns, 2020.

[36]. Beiran Manuel, Dubreuil Alexis, Valente Adrian, Mastrogiuseppe Francesca, and Ostojic Srdjan. Shaping dynamics with multiple populations in low-rank recurrent networks. Neural Computation, 33(6):1572–1615, 2021. [PubMed: 34496384]

[37]. Sherman S Murray and Guillery RW. 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.

[38]. Ferguson Katie A and Cardin Jessica A. Mechanisms underlying gain modulation in the cortex. Nature Reviews Neuroscience, 21(2):80–92, 2020. [PubMed: 31911627]

[39]. Yang Guangyu Robert and Wang Xiao-Jing. Artificial neural networks for neuroscientists: A primer. Neuron, 107(6):1048–1070, 2020. [PubMed: 32970997]

[40]. Gold Joshua I and Shadlen Michael N. The neural basis of decision making. Annual review of neuroscience, 30, 2007.

[41]. Romo Ranulfo, Brody Carlos D, Hernández Adrián, and Lemus Luis. Neuronal correlates of parametric working memory in the prefrontal cortex. Nature, 399(6735):470–473, 1999. [PubMed: 10365959]

[42]. Miyashita Yasushi. Neuronal correlate of visual associative long-term memory in the primate temporal cortex. Nature, 335(6193):817–820, 1988. [PubMed: 3185711]

[43]. Cunningham John P and Yu Byron M. Dimensionality reduction for large-scale neural recordings. Nature neuroscience, 17(11):1500–1509, 2014. [PubMed: 25151264]

[44]. Fusi Stefano, Miller Earl K, and Rigotti Mattia. Why neurons mix: high dimensionality for higher cognition. Current opinion in neurobiology, 37:66–74, 2016. [PubMed: 26851755]

[45]. Cromer Jason A, Roy Jefferson E, and Miller Earl K. Representation of multiple, independent categories in the primate prefrontal cortex. Neuron, 66(5):796–807, 2010. [PubMed: 20547135]

[46]. Fritz Jonathan B, David Stephen V, Radtke-Schuller Susanne, Yin Pingbo, and Shamma Shihab A. Adaptive, behaviorally gated, persistent encoding of task-relevant auditory information in ferret frontal cortex. Nature neuroscience, 13(8):1011, 2010. [PubMed: 20622871]

[47]. Elgueda Diego, Duque Daniel, Radtke-Schuller Susanne, Yin Pingbo, David Stephen V, Shamma Shihab A, and Fritz Jonathan B. State-dependent encoding of sound and behavioral meaning in a

tertiary region of the ferret auditory cortex. Nature neuroscience, 22(3):447–459, 2019. [PubMed: 30692690]

[48]. Zenke Friedemann, Poole Ben, and Ganguli Surya. Continual learning through synaptic intelligence. International Conference on Machine Learning, pages 3987–3995, 2017.

[49]. Roy Jefferson E, Riesenhuber Maximilian, Poggio Tomaso, and Miller Earl K. Prefrontal cortex activity during flexible categorization. Journal of Neuroscience, 30(25):8519–8528, 2010. [PubMed: 20573899]

[50]. Sussillo David and Barak Omri. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. Neural computation, 25(3):626–649, 2013. [PubMed: 23272922]

[51]. Rabinowitz Neil C, Goris Robbe L, Cohen Marlene, and Simoncelli Eero P. Attention stabilizes the shared gain of v4 populations. Elife, 4:e08998, 2015. [PubMed: 26523390]

[52]. Salinas Emilio and Thier Peter. Gain modulation: a major computational principle of the central nervous system. Neuron, 27(1):15–21, 2000. [PubMed: 10939327]

[53]. Stroud Jake P, Porter Mason A, Hennequin Guillaume, and Vogels Tim P. Motor primitives in space and time via targeted gain modulation in cortical networks. Nature neuroscience, 21(12):1774–1783, 2018. [PubMed: 30482949]

[54]. Maheswaranathan Niru, Williams Alex H, Golub Matthew DGanguli Surya, and Sussillo David. Universality and individuality in neural dynamics across large populations of recurrent networks. Advances in neural information processing systems, 2019:15629, 2019. [PubMed: 32782422]

[55]. Flesch Timo, Juechems Keno, Dumbalska Tsvetomira, Saxe Andrew, and Summerfield Christopher. Rich and lazy learning of task representations in brains and neural networks. bioRxiv, 2021.

[56]. Aoi Mikio C, Mante Valerio, and Pillow Jonathan W. Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making. Nature neuroscience, 23(11):1410–1420, 2020. [PubMed: 33020653]

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

[58]. Kingma Diederik P and Ba Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[59]. Paszke Adam, Gross Sam, Chintala Soumith, Chanan Gregory, Yang Edward, DeVito Zachary, Lin Zeming, Desmaison Alban, Antiga Luca, and Lerer Adam. Automatic differentiation in pytorch. 2017.

[60]. Pedregosa Fabian, Varoquaux Gäel, Gramfort Alexandre, Michel Vincent, Thirion Bertrand, Grisel Olivier, Blondel Mathieu, Prettenhofer Peter, Weiss Ron, Dubourg Vincent, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

[61]. Kingma Diederik P and Welling Max. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[62]. Kostantinos N. 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.

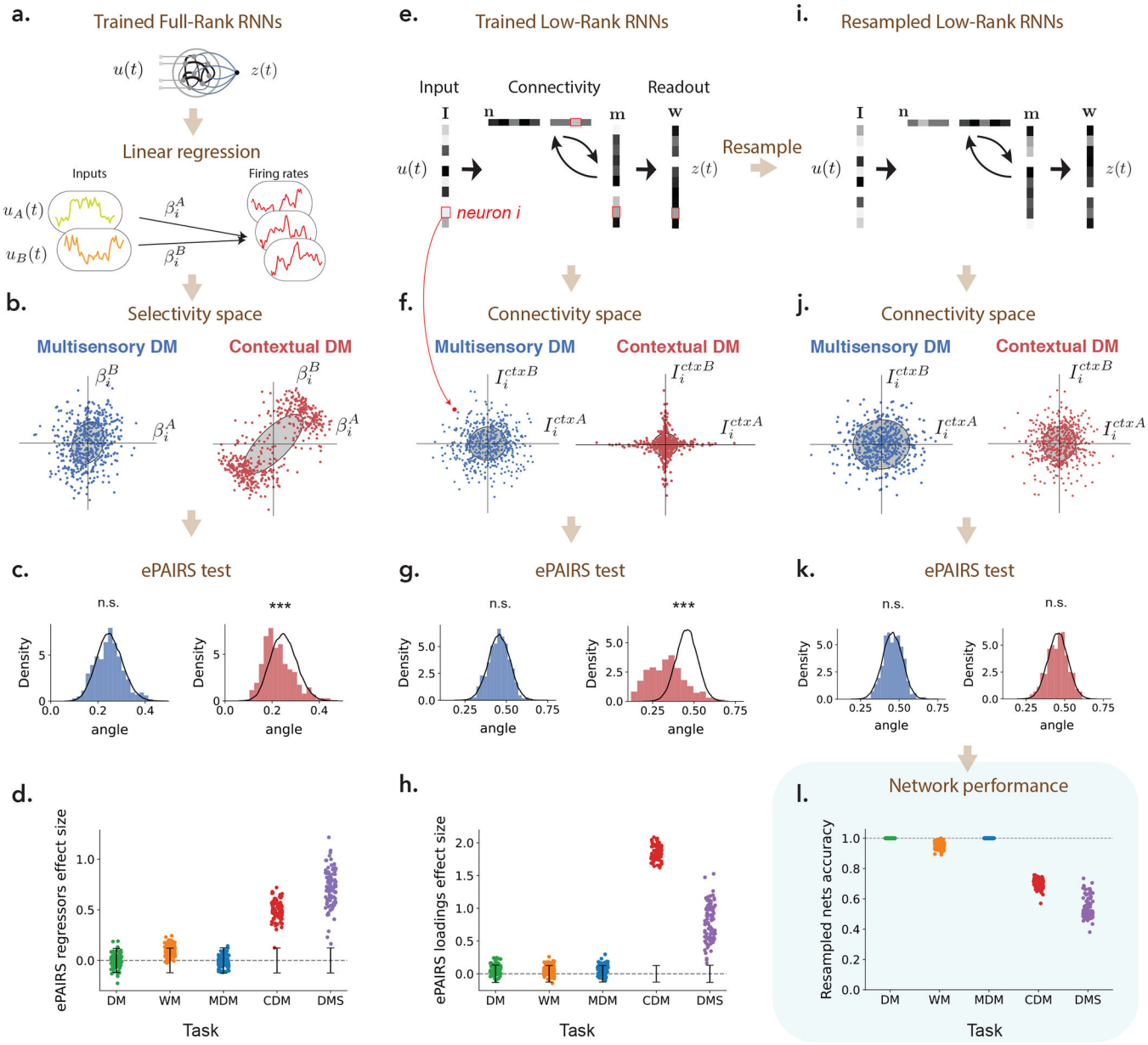



**Figure 1. Identifying non-random population structure in selectivity, connectivity and computations.**

(a) Recurrent neural networks (RNNs) were trained separately on five tasks. For each task, and each trained RNN, selectivity was first quantified by computing linear regression coefficients $\beta_i^{var}$ for each neuron $i$ with respect to task-defined variables such as stimulus features or decision (see Methods ). Each neuron was then represented as a point in a selectivity space where each axis corresponds to the regression coefficient with respect to one variable. For each network, we then compared the resulting distribution of points with a random shuffle corresponding to a multivariate Gaussian with matching empirical covariance. (b) Illustration of the distribution of regression coefficients in selectivity space for two networks trained on respectively the multi-sensory (MDM) and context-dependent

decision-making (CDM) tasks which received identical inputs (two stimuli $A$ and $B$ and two contextual cues) but required different outputs. The full selectivity space was four dimensional. The plots show two-dimensional projections of the selectivity distribution onto the plane defined by regression coefficients with respect to stimuli $A$ and $B$. Gray ellipses correspond to the 1 s.d. ellipse of a Gaussian distribution with matching mean and covariance. (c) Distribution of angles between each point and its nearest neighbor in the selectivity space illustrated in panel b (colored histograms), compared with that of a matching multivariate Gaussian (null distribution, black line). The mismatch between the two distributions was quantified using the ePAIRS test[17;10] (two-sided, see Methods ). The mismatch was significant for the CDM task ($p = 3 \times 10^{-25}$, effect size $c = 0.58$ $n =$ 512 neurons; ***: $p < 0.001$), but not for the MDM task ($p = 0.61$, $c = 0.01$, $n = 512$). (d) Population structure in the selectivity space across networks and tasks: effect size of the ePAIRS test on the selectivity space for 100 networks trained on each of the tasks (see Extended Data Figure 1 for p-values). Black bars represent 95% confidence intervals for null distributions, centered around mean null effect size. (e) To assess for population structure in connectivity, we focused on low-rank networks, where connectivity is fully specified by vectors over neurons[30]. Each neuron is characterized by one parameter on each vector (illustrated in grayscale, entries for a specific neuron are outlined in red), and can be represented as a point in a connectivity space where each axis corresponds to the parameters on one vector. We assessed the presence of non-random population structure in that space using a procedure identical to the analysis of selectivity (c-d). (f) Illustration of the distribution of parameters in connectivity space for the two networks trained on respectively the MDM and CDM tasks. For these tasks, minimal trained networks were of rank $R = 1$ (Extended Data Figure 2), so that the connectivity space was of dimension 7 (four inputs, two recurrent vectors and one readout). The plots show two-dimensional projections of the full connectivity distribution onto the plane defined by input parameters of contextual cues $A$ and $B$. (g) Comparison of nearest-neighbor angle distributions in connectivity space for trained networks and the randomized shuffles as in c. The difference is significant for the CDM task ($p = 2 \times 10^{-142}$, $c = 1.89$, $n = 512$), but not for the MDM task ($p = 0.72$, $c = 0.005$, $n = 512$). (h) Population structure in the connectivity space across networks and tasks: effect size of the ePAIRS test on the connectivity space for 100 networks trained on each of the five studied tasks (see Extended Data Figure 1 for p-values). (i) To identify the causal role of population structure on computations, we randomly generated new networks by resampling from the null distribution in connectivity space that preserved the mean and covariance structure but scrambled any non-random population structure. (j-k) In randomly resampled networks, the statistics of connectivity are by design identical to shuffles used for the ePAIRS test (MDM: $p = 0.08$, $c = 0.05$, $n = 512$; CDM: $p = 0.68$, $c = -0.01$, $n = 512$). (l) Performance of each randomly resampled network on its corresponding task as measured by accuracy.

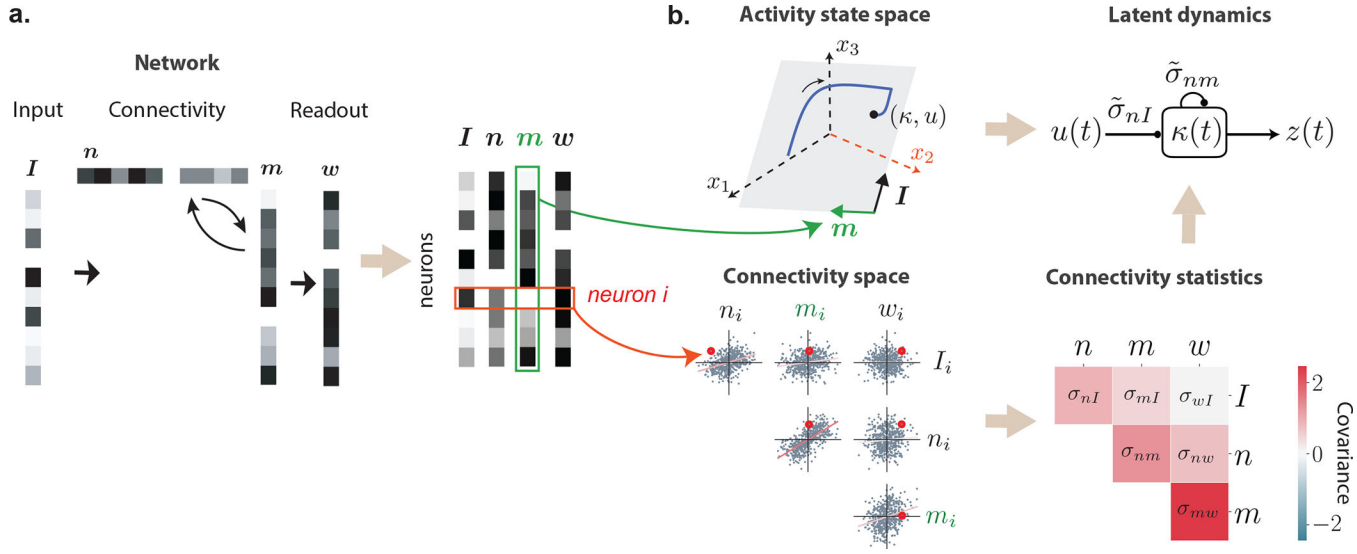



**Figure 2. Reducing low-rank networks to low-dimensional latent dynamics to explain computations in low-rank RNNs.**

(a) The connectivity parameters in a low-rank RNN (left) can be grouped in a matrix where each row contains the input, recurrent and output parameters (values illustrated in grayscale) of a given neuron (right). (b) The connectivity can therefore be represented in two complementary manners that together determine low-dimensional dynamics. Top-left: Columns of the matrix in (a) define specific directions (illustrated as arrows) in the activity state space, where each axis is the activity $x_i$ of neuron $i$. The connectivity constrains the trajectories of activity to lie in a low-dimensional subspace spanned by input vectors $\boldsymbol{I}^{(s)}$ and recurrent vectors $\boldsymbol{m}^{(r)}$. The activity trajectory (illustrated in blue) is parametrized along those directions by input variables $u_s$ and internal variables $\kappa_r$. Bottom left: Each row of the matrix in (a) defines a point in the connectivity space (specific example in red), where each axis corresponds to entries along each connectivity vector. The full network is described by the distribution of the cloud of points. Here we illustrate a four-dimensional distribution by its pairwise two-dimensional projections. Bottom right: A Gaussian distribution in connectivity space is specified by its covariance matrix that describes the shape of the point cloud (regression lines shown in bottom left). Top right: The latent dynamics can be reduced to an effective circuit (Eq. 3), in which each internal variable is represented as a unit that receives external inputs, and interacts with itself (and other internal variables) through a set of effective couplings determined by the connectivity covariances illustrated in the bottom-left panel.

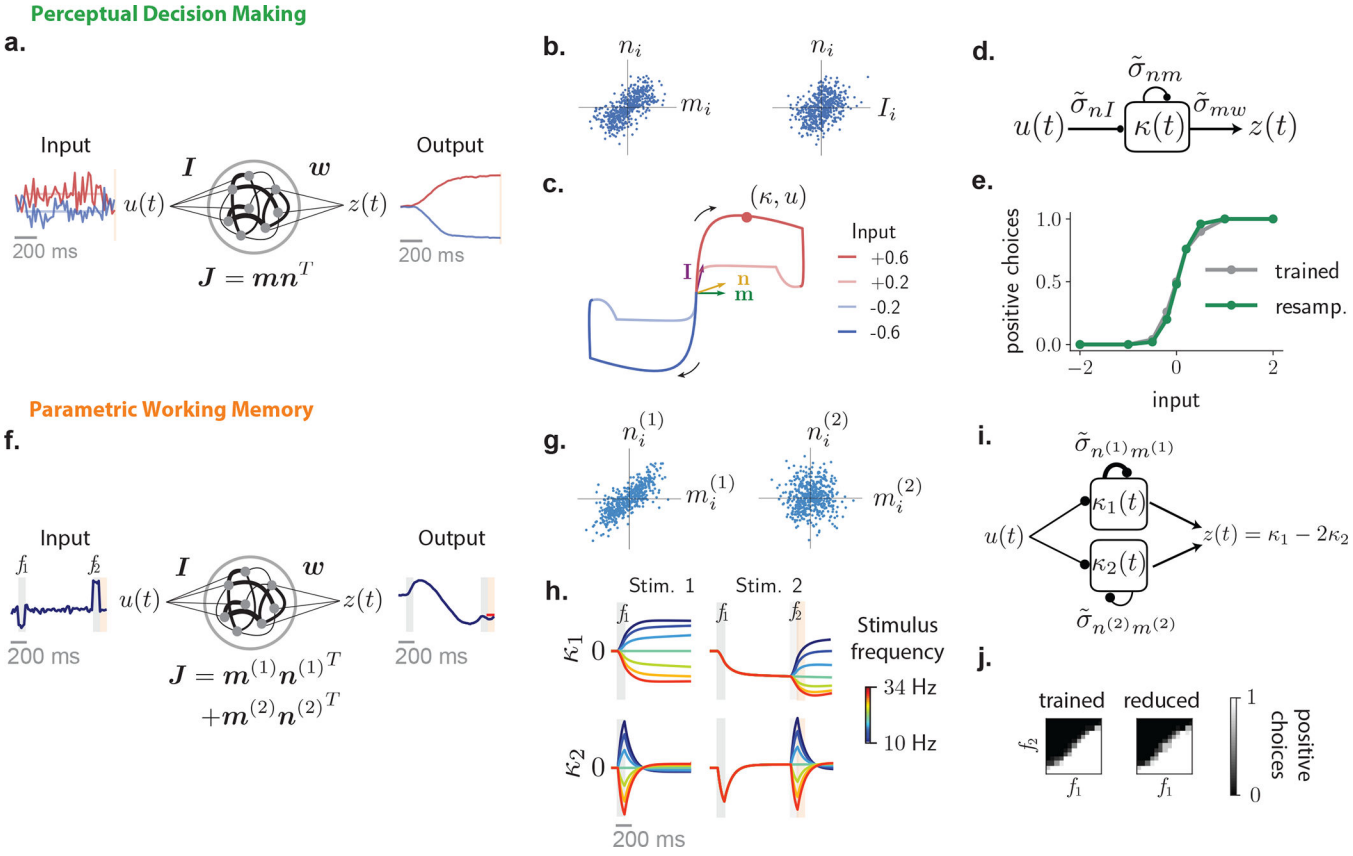



**Figure 3. Low-dimensional latent dynamics in networks with a random population structure.**

(a)-(e) Perceptual decision making task. (a) A rank-one network was trained to output the sign of the mean of a noisy input signal. Two example trials for a positive (red) and a negative (blue) input mean. (b) Two two-dimensional projections of the obtained four-dimensional connectivity space. Each point represents the connectivity parameters of one neuron. (c) Low-dimensional trajectories in the two-dimensional subspace spanned by vectors $\boldsymbol{m}$ and $\boldsymbol{I}$ for four trials. (d) The latent dynamics are equivalent to an effective circuit governed by 2 effective couplings (Eq. 3), which are determined by the overlaps $\sigma_{nI}$ and $\sigma_{nm}$ of the vector $\boldsymbol{n}$ with $\boldsymbol{I}$ and $\boldsymbol{m}$ (see vectors in panel c). The readout from the network is set by the overlap $\sigma_{mw}$ between the vectors $\boldsymbol{m}$ and $\boldsymbol{w}$. (e) Psychometric function showing the fraction of positive outputs for the trained network, and for a reduced network generated by controlling only three parameters corresponding to the effective couplings in f (see Supplementary Note 2.1). (f)-(j) Parametric working memory task. (f) A rank-two network was trained to compute the difference between two stimuli $f_1$ and $f_2$ separated by a variable delay. (g) Two projections of the obtained six-dimensional connectivity space. (h) Since the network is rank-two, the recurrent activity is parametrized by two internal variables, $\kappa_1$ and $\kappa_2$ that correspond to activity along connectivity vectors $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$. The variable $\kappa_1$ acts as an integrator that encodes the stimuli persistently: it encodes $f_1$ following the first stimulus, and $f_1 + f_2$ following the second one. The variable $\kappa_2$ responds transiently to each stimulus, and therefore encodes $f_2$ at the decision time. (i) The latent dynamics are described by an effective circuit where the two internal variables evolve independently, with different amounts of positive feedback (Eq. 45). (j) Psychometric response matrix for

the trained network, and a reduced network generated by controlling only six parameters corresponding to the effective couplings in i (see Supplementary Note 2.2). Each matrix displays the fraction of positive responses for each combination of stimuli $f_1$ and $f_2$.

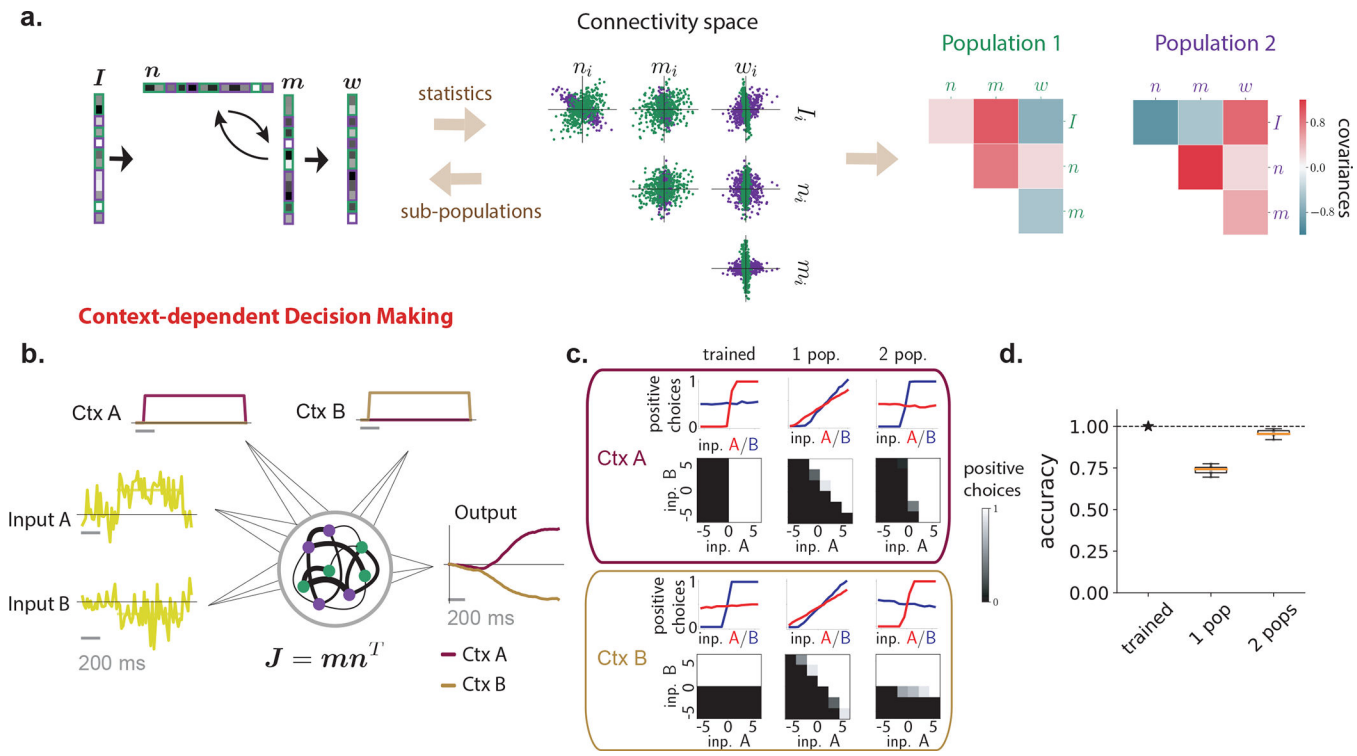



**Figure 4. Multi-population connectivity structure captures the computational requirements for context-dependent tasks.**

(a) Method for representing a low-rank connectivity structure in terms of multiple sub-populations. The connectivity vectors (left) are represented as a set of points in connectivity space, each point corresponding to connectivity parameters of one neuron. The center panel shows an illustration of two-dimensional projections of the full distribution in connectivity space, which in this example is four dimensional. A mixture of Gaussians clustering algorithm assigns every neuron to a sub-population based on the full distribution in connectivity space. The green and purple colors denote the two identified sub-populations, which in this illustration have identical centers but different shapes. Each sub-population is therefore defined by a different set of covariances (right panel), that correspond to overlaps between vectors shown in green and purple colors in the left panel. (b)-(d) Application to the context-dependent decision making task. (b) Networks received stimulus inputs consisting of two noisy features along two different input vectors, together with one of two contextual cues in each trial. Unit-rank networks were trained to output the sign of the mean of the cued feature. Here we illustrate two example trials sharing the same stimulus inputs and opposite contextual cues (context A activated in dark red, context B in pale brown), leading to opposite outputs. (c) Psychometric functions and response matrices. Each psychometric matrix displays the fraction of positive responses for each combination of stimulus features. Each psychometric function represents the fraction of positive responses for the value of one feature, averaging over the other. The two rows show psychometric functions and matrices in different contexts, for a trained network (left column), and for networks generated by resampling connectivity from a single population (middle column) or two sub-populations (right column). (d) Average accuracy of a trained network and for 10 draws of resampled

single-population and two-population networks (line: median, box: quartiles, whiskers: range, in the limit of median ± 1.5 interquartile range, points: outliers).

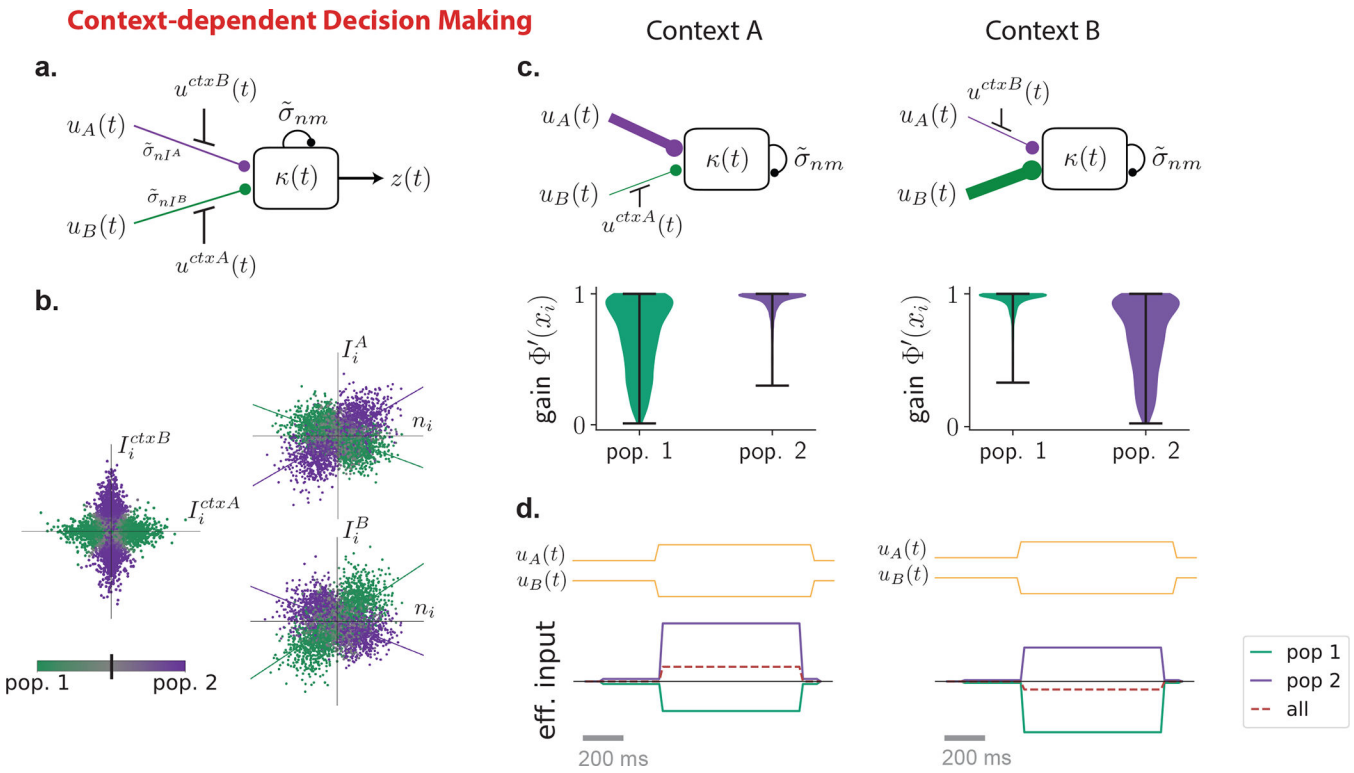



**Figure 5. Mechanisms of computations based on a multi-population connectivity.**

(a) Circuit diagram representing latent dynamics in the reduced model of context-dependent decision-making task (Eq. 5 and Supplementary Note 2.3). The internal variable $\kappa$ is represented as a unit that integrates the two stimulus features $u_A$ and $u_B$ through effective couplings $\tilde{\sigma}_{nI}{}^A$ and $\tilde{\sigma}_{nI}{}^B$ Contextual inputs $u^{ctxA}$ and $u^{ctxB}$ modulate the gains of the two populations and therefore the effective couplings that govern which stimulus feature is integrated. Lines with round ends represent effective couplings, lines with straight ends represent gain modulation. (b) Projections of the six-dimensional connectivity space for a network trained on the task. Each point represents the parameters of one neuron, with the color shade indicating the probability that it belongs to each sub-population, as found by the clustering procedure (Fig. 4a). For the remaining analysis, the two sub-populations are defined by a hard threshold at 0.5 on this probability. Left: plane defined by components of the contextual-cue vectors $\boldsymbol{I}^{ctxA}$ and $\boldsymbol{I}^{ctxB}$; right: two planes defined by components on the input-selection vector $\boldsymbol{n}$ and the two stimulus feature vectors $\boldsymbol{I}^A$ and $\boldsymbol{I}^B$ (lines show linear regressions for each population). (c) Effective circuits in each context (top) and corresponding gains of neurons in each population (bottom). For each neuron $i$, the gain is defined as the slope of $\phi(x_i)$ during stimulation period. Violin plots show the distribution of gains for all neurons in each population (pop. 1: $n = 2028$, pop. 2: $n = 2068$) in context A (left) and B (right). In context A, the average gain of neurons in population 1 (green) is lower than population 2 (purple), which decreases the effective connectivity between input feature B and the latent variable (top left circuit). The opposite happens in context B (top right circuit). (d) Effective inputs to the latent variable $\kappa$ in the two contexts (bottom) in response to the same stimulus input (top). Solid lines show inputs mediated by each

population (Methods Eq. (36)), the dashed line shows the total input, which changes signs between the two contexts, leading to opposite responses.

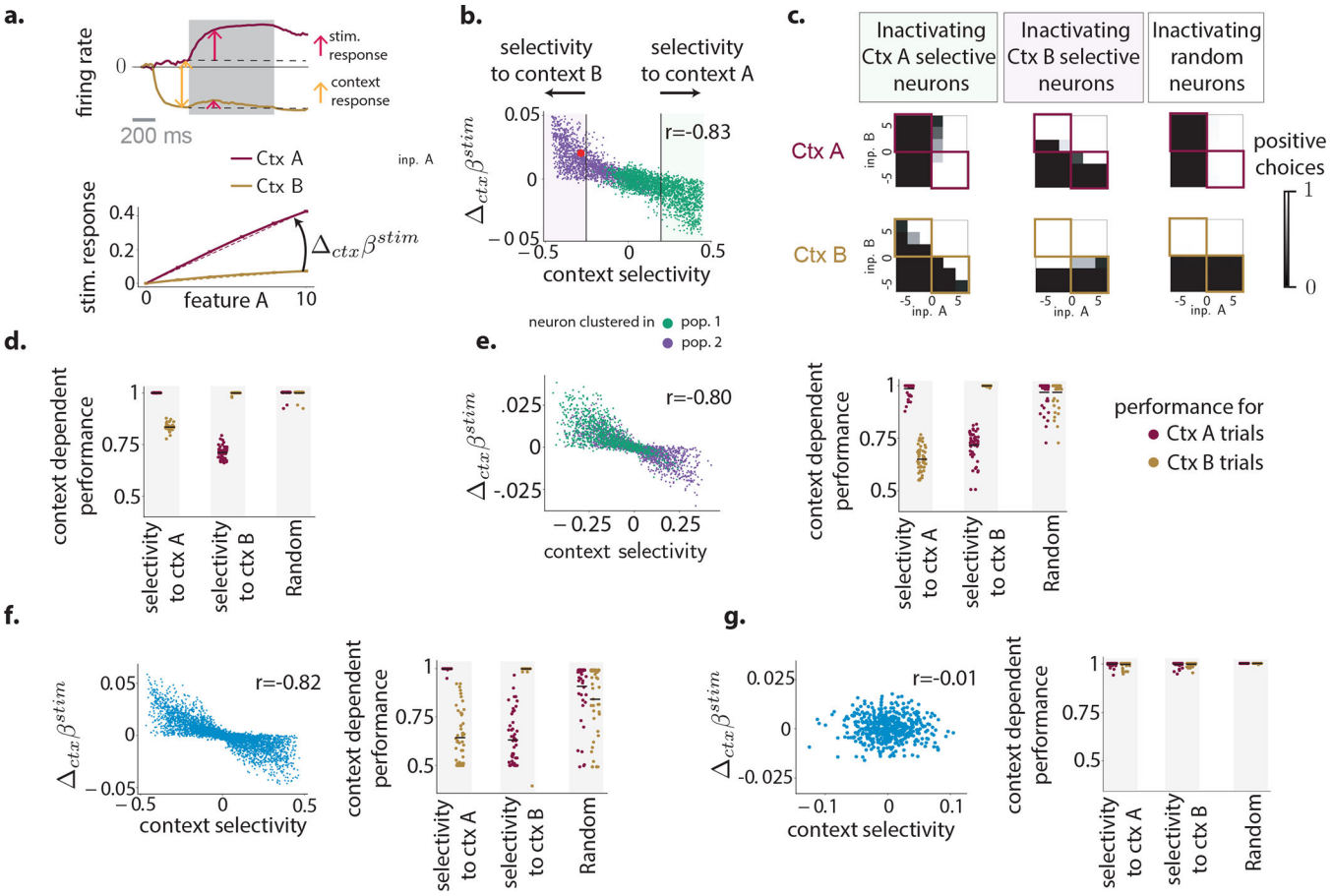



**Figure 6. Predictions for neural selectivity and inactivations.**

(a-d) Predictions for the context-dependent decision-making task based on the minimal unit-rank, two-populations network (Fig. 5a). (a) Context-dependent stimulus response for an example neuron. Top: response to an identical stimulus in two contexts (gray box: stimulus-presentation period). The context response was defined as the change of pre-stimulus baseline across contexts (orange arrow). The stimulus response was defined in each context as the deviation from the pre-stimulus baseline (red arrows). Bottom: context-dependent responses of the same neuron to stimuli with increasing strength of feature *A*. In each context, we computed the regression coefficient with respect to feature strength (dashed lines), and the corresponding change in stimulus selectivity $\Delta_{ctx}\beta^{stim}$ (Methods Eq. (19)). (b) Interaction between context selectivity and the change in stimulus selectivity across neurons. Each point shows the change in stimulus selectivity versus selectivity to context for one neuron (see Methods Eq. (18)). Dot color corresponds to population determined from clustering procedure (Fig. 5). Red dot: example neuron in (a). (c) Inactivations based on context selectivity lead to specific performance deficits. Psychometric response matrices when inactivating the 256 out of 1024 neurons with highest positive context selectivity (left), highest negative context selectivity (middle) or randomly chosen across the whole network (right). (d) Summary of the effects of inactivations: average performance over incongruent stimuli corresponding to colored squares of the psychometric matrix in (c). Each dot represents an inactivation of a random subset of 256 out of 1024 neurons. Inactivated

neurons are chosen randomly among the neurons with either positive context selectivity (left column), negative context selectivity (middle column) or without constraint (right column). (e-g) Tests of the predictions for selectivity (left panels) and inactivations (right panels) on: (e) a unit-rank network consisting of three populations (Extended Data Figure 5); (f) a network trained without a rank constraint; (g) a network trained on the multi-sensory decision-making (MDM) task.

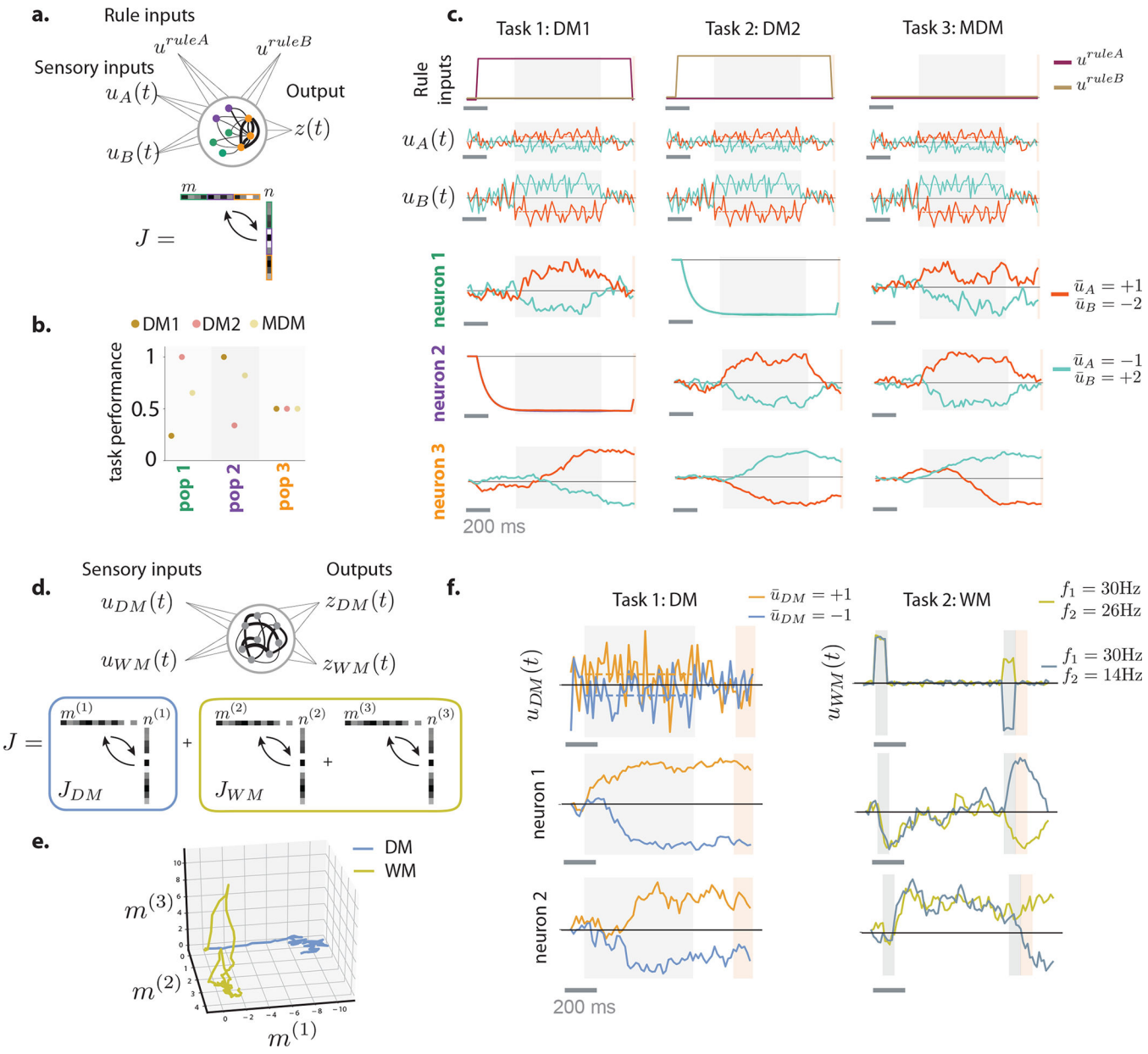



**Figure 7. Implications of multi-population structure for multi-tasking.**

(a) A network performing three different tasks on the same set of stimuli consisting of two features $u_A$ and $u_B$: decision-making based on $u_A$ (DM1), decision-making based on $u_B$ (DM2), decision-making based on integrating $u_A$ and $u_B$ (MDM). The model is obtained from the unit-rank network performing the CDM task based on three populations indicated in color. (b) Effects on the performance of individual tasks when specific populations are inactivated. In each case one third of the neurons in the network is inactivated, corresponding to one of the three populations. (c) Illustration of task specialization of different populations. The orange population plays the role of an integrator, and participates to all tasks. Green and purple populations respectively relay $u_A$ and $u_B$. Different columns correspond to different tasks. Top three rows display stimulus and rule inputs. Bottom

three rows display single unit activities of three selected neurons (one in each population) in two trials of each task. (d) A network performing two different tasks on distinct sets of stimuli, the decision-making (DM) task on $u_{DM}$, and the working-memory task on $u_{WM}$. This network is obtained by superposing the low-rank recurrent connectivity matrices corresponding to the two tasks (illustrated at the bottom). (e) The two tasks rely on neural activity in orthogonal subspaces of the state space. Each subspace is determined by the input connectivity vectors of the corresponding task. (f) Illustration of multi-tasking of two example neurons.