RANGE NOT STATISTICAL INDEPENDENCE DRIVES MODULARITY IN BIOLOGICAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Why do biological and artificial neurons sometimes modularise, each encoding a single meaningful variable, and sometimes entangle their representation of many variables? In this work, we develop a theory of when biologically inspired networks—those that are nonnegative and energy efficient—modularise their representation of source variables (sources). We derive necessary and sufficient conditions on a sample of sources that determine whether the neurons in an optimal biologically-inspired linear autoencoder modularise. Our theory applies to any dataset, extending far beyond the case of statistical independence studied in previous work. Rather we show that sources modularise if their support is "sufficiently spread". From this theory, we extract and validate predictions in a variety of empirical studies on how data distribution affects modularisation in nonlinear feedforward and recurrent neural networks trained on supervised and unsupervised tasks. Furthermore, we apply these ideas to neuroscience data, showing that range independence can be used to understand the mixing or modularising of spatial and reward information in entorhinal recordings in seemingly conflicting experiments. Further, we use these results to suggest alternate origins of mixed-selectivity, beyond the predominant theory of flexible nonlinear classification. In sum, our theory prescribes precise conditions on when neural activities modularise, providing tools for inducing and elucidating modular representations in brains and machines.

1 INTRODUCTION

Our brains are modular. At the macroscale, different regions, such as visual or language cortex, perform specialised roles; at the microscale, single neurons often precisely encode single variables such as self-position (Hafting et al., 2005) or the orientation of a visual edge (Hubel and Wiesel, 1962). This mysterious alignment of meaningful concepts with single neuron activity has for decades fuelled hope for understanding a neuron's function by finding its associated concept. Yet, as neural recording technology has improved, it has become clear that many neurons behave in ways that elude such simple categorisation: they appear to be mixed selective, responding to a mixture of variables in linear and nonlinear ways (Rigotti et al., 2013; Tye et al., 2024). The modules vs. mixtures debate has recently been reprised in the machine learning community. Both the disentangled representation learning and mechanistic interpretability subfields are interested in when neural network representations decompose into meaningful components. Findings have been similarly varied, with some studies showing meaningful single unit response properties and others clearly showing mixed tuning (for a full discussion on related work see Appendix A). This brings us to our main research question: Why are neurons, biological and artificial, sometimes modular and sometimes mixed selective?

In this work, we develop a theory that precisely determines, for any dataset, whether the optimal learned representations will be modular or not. More precisely, in the linear autoencoder setting, we show that modularity in biologically constrained representations is governed by a *sufficient spread* condition that can roughly be thought of as measuring the extent to which the *support* of the source variables (sources, aka factors of variation) is *rectangular*. This sufficient spread condition bears resemblance to identifiability conditions derived in the matrix factorisation literature (Tatli and Erdogan, 2021a;b), though both the precise form of the problem we study and the condition we derive differ (Appendix A). This condition on the sources is much weaker than the case of mutual independence studied in previous work (Whittington et al., 2023a), and commensurately broadens the settings we can understand using our theory. For example, the loosening from statistical independence to rectangular support enables us to predict when linear recurrent neural network (RNN) representations of dynamic variables modularise.

Further, these results empirically generalise to nonlinear settings: we show that our source support conditions predict modularisation in nonlinear feedforward networks on supervised and autoencoding tasks as well as in nonlinear RNNs. We also fruitfully apply our theory to neuroscience data. We provide an explanation for why grid cells only sometimes warp in the presence of rewards, on the basis of the support independence properties of space and reward. Further, we use these results to suggest other settings in which mixed-selectivity might appear beyond the traditional nonlinear classification theory. In summary, our work contributes to the growing understanding of neural modularisation by highlighting how source support determines modularisation and explaining puzzling observations from both the brain and neural networks in a cohesive normative framework.

2 MODULARISATION IN BIOLOGICALLY INSPIRED LINEAR AUTOENCODERS

We begin with our main technical result: necessary and sufficient conditions for the modularisation of biologically inspired linear autoencoders.

2.1 PRELIMINARIES

Let $\boldsymbol{s} \in \mathbb{R}^{d_s}$ be a vector of d_s scalar source variables (sources, aka factors of variation). We are interested in how the empirical distribution, $p(\boldsymbol{s})$, affects whether the sources' neural encoding (latents), $\boldsymbol{z} \in \mathbb{R}^{d_z}$, are *modular* with respect to the sources, i.e., whether each neuron (latent) is functionally dependent on at most one source. Following Whittington et al. (2023a), we build a simplified model in which neural firing rates perfectly linearly autoencode the sources while maintaining nonnegativity (since firing rates are nonnegative):

$$\boldsymbol{z} = \boldsymbol{W}_{\text{in}}\boldsymbol{s} + \boldsymbol{b}_{\text{in}}, \quad \boldsymbol{s} = \boldsymbol{W}_{\text{out}}\boldsymbol{z} + \boldsymbol{b}_{\text{out}}, \quad \boldsymbol{z} \geq 0. \tag{1}$$

Subject to the above constraints, we study the representation that uses the least energy, as in the classic efficient coding hypothesis Barlow et al. (1961). We quantify this using the ℓ^2 norm of the firing rates and weights (other activity norms considered later):

$$\min_{\boldsymbol{W}_{\text{in}},\boldsymbol{b}_{\text{in}},\boldsymbol{W}_{\text{out}},\boldsymbol{b}_{\text{out}}} \left\langle ||\boldsymbol{z}||_2^2 \right\rangle_{p(\boldsymbol{s})} + \lambda \left(||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2 \right) \text{ s.t. (1).} \tag{2}$$

We remark that there are common analogues of representational nonnegativity and weight energy efficiency in modern machine learning (ReLU activation functions and weight decay, respectively). When the sources are statistically independent, i.e. $p(\boldsymbol{s}) = \prod_{i=1}^{d_s} p(s_i)$, then the optima of (2) have modular latents (Whittington et al., 2023a). We now improve on this result by showing necessary and sufficient conditions that guarantee modularisation for any dataset, not just those that have statistically independent sources.

2.2 INTUITION FOR SOURCE SUPPORT MODULARISATION CONDITIONS

To provide intuition, consider a hypothetical mixed selective neuron,

$$z_j = w_{j1}s_1 + w_{j2}s_2 + b_j, \tag{3}$$

that is functionally dependent on both s_1 and s_2. This neuron is mixed selective. Perhaps, however, a modular representation, in which this neuron is broken into two separate modular encodings, would be better. If so, the optimal solution would have two neurons each coding for a single source:

$$z_{j'} = w_{j1}s_1 + b_{j'}, \quad z_{j''} = w_{j2}s_2 + b_{j''}. \tag{4}$$

For simplicity, we assume the two sources are linearly uncorrelated, have mean zero, and are supported on an interval centered at zero (Figure 1a top row; we relax these assumptions in our full theory). Then, for fixed w_{j1} and w_{j2}, the optimal (energy efficient) bias should be large enough to keep the representation nonnegative, but no larger:

$$b_j = -\min_{s_1,s_2} \left[w_{j1}s_1 + w_{j2}s_2 \right], \quad b_{j'} = -\min_{s_1} w_{j1}s_1, \quad b_{j''} = -\min_{s_2} w_{j2}s_2, \tag{5}$$

where the minimizations are constrained to the support of the empirical distribution $p(s_1, s_2)$. Now that both representations are specified, we can compare their costs. When calculating the activity energy one finds that, in this simplified setting, most terms are zero or cancel, and hence the mixed selective (3) case uses less energy than the modular (4) when

$$b_j^2 < b_{j'}^2 + b_{j''}^2. \tag{6}$$

The key takeaway lies in how b_j is determined by a joint minimization over s_1 and s_2 (5). Assuming positive w_{j1} and w_{j2}; then if s_1 and s_2 take their minima simultaneously, as in the middle row of Figure 1a, then mixed bias must be large:

$$b_j = -\min_{s_1,s_2}[w_{j1}s_1 + w_{j2}s_2] = -\min_{s_1}[w_{j1}s_1] - \min_{s_2}[w_{j2}s_2] = b_{j'} + b_{j''} \tag{7}$$

And the energy of the mixed solution will always be worse than the modular, since $b_j^2 = (b_{j'}+b_{j''})^2 > b_{j'}^2 + b_{j''}^2$. Alternatively, mixing will be preferred when s_1 and s_2 *do not* take on their most negative values at the same time, as in the bottom row of Figure 1a, since then b_j does not have to be as large to maintain positivity, and the corresponding energy saving satisfies the key inequality (6).

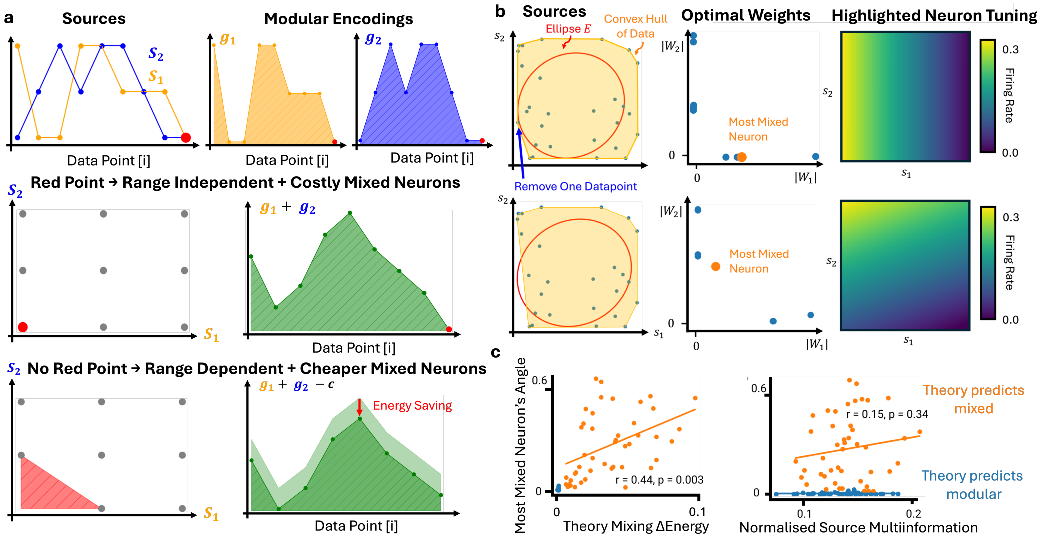


Figure 1: a) Top row: Values of two sources across a dataset and their modular encoding with associated costs (shaded regions). Middle row: If (left) the dataset includes the red datapoint then the two variables take their minimal value at the same time, and (right) a mixed encoding must include a large bias, and uses more energy. Bottom row: however (left) if the red point is missing then (right) the mixed encoding can use a smaller bias while remaining positive and save energy. b) Our modularisation conditions (Theorem 2.1) are equivalent to whether the convex hull of the data (the orange regions) encloses a data-dependent ellipse (Theorem 2.2), the red ellipse that can be easily calculated from the dataset. The precision of these conditions is validated in two experiments (rows) in which the source support differs by a single datapoint. Removing this datapoint stops the convex hull from enclosing the key ellipse, and hence the optimal representation contains a mixed-selective neuron. We plot the weight vectors of all neurons in the optimal representation (middle column), in the top row all weight vectors are axis aligned, the neurons are modular, whereas in the bottom there are mixed selective neurons. This is reiterated by extracting the most mixed selective neuron (orange dot) and (right column) plotting its response. c) Our conditions (left) predict modularisation better than measures of statistical independence such as source multiinformation (right). Here the y axis measures how mixed the representation is by quoting the largest angle between a neuron's weight vector and one of the source axes.

2.3 PRECISE CONDITIONS FOR MODULARISING BIOLOGICALLY INSPIRED LINEAR AUTOENCODERS

We now make precise the intuition developed above. All proofs are deferred to Appendix B.

Theorem 2.1. *Let* $\boldsymbol{s} \in \mathbb{R}^{d_s}$, $\boldsymbol{z} \in \mathbb{R}^{d_z}$, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{d_z \times d_s}$, $\boldsymbol{b}_{\text{in}} \in \mathbb{R}^{d_z}$, $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d_s \times d_z}$, *and* $\boldsymbol{b}_{\text{out}} \in \mathbb{R}^{d_s}$, *with* $d_z > d_s$. *Consider the constrained optimization problem*

$$\min_{\boldsymbol{W}_{\text{in}},\boldsymbol{b}_{\text{in}},\boldsymbol{W}_{\text{out}},\boldsymbol{b}_{\text{out}}} \quad \left\langle ||\boldsymbol{z}^{[i]}||_2^2 \right\rangle_i + \lambda \left(||\boldsymbol{W}_{\text{in}}||_F^2 + ||\boldsymbol{W}_{\text{out}}||_F^2 \right) \tag{8}$$
$$\text{s.t.} \quad \boldsymbol{z}^{[i]} = \boldsymbol{W}_{\text{in}}\boldsymbol{s}^{[i]} + \boldsymbol{b}_{\text{in}}, \;\; \boldsymbol{s}^{[i]} = \boldsymbol{W}_{\text{out}}\boldsymbol{z}^{[i]} + \boldsymbol{b}_{\text{out}}, \;\; \boldsymbol{z}^{[i]} \geq 0,$$

where i indexes a finite set of samples of $\boldsymbol{s}$. At the minima of this problem, the representation modularises, i.e. each row of $\boldsymbol{W}_{\text{in}}$ has at most one non-zero entry, iff the following inequality is satisfied for all $\boldsymbol{w} \in \mathbb{R}^{d_s}$:

$$\left(\min_i [\boldsymbol{w}^\top \bar{\boldsymbol{s}}^{[i]}]\right)^2 + \sum_{j,j'\neq j}^{d_s} w_j w_{j'} \left\langle \bar{s}_j^{[i]} \bar{s}_{j'}^{[i]} \right\rangle_i > \sum_{j=1}^{d_s} \left(w_j \min_i \bar{s}_j^{[i]}\right)^2, \tag{9}$$

where $\bar{\boldsymbol{s}} := \boldsymbol{s} - \left\langle \boldsymbol{s}^{[i]} \right\rangle_i$ and assuming that $\left|\min_i \bar{s}_j^{[i]}\right| \leq \max_i \bar{s}_j^{[i]} \; \forall j \in [d_s]$ w.l.o.g.

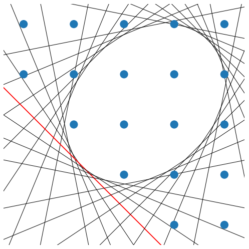
Our theory prescribes a set of inequalities (Equation 9) that determine whether an optimal representation is modular. These inequalities come from the difference in activity energy between a modular and mixed solutions, just like the intuition we built up in Section 2.2, and in particular Equation 6. If a single inequality is broken, the representation is mixed (at least in part); else, the representation is modular: each neuron's activity is a function of a single source. These inequalities depend on two key properties of the sources: the shape of the source distribution support in extremal regions, and the pairwise source correlations. Remarkably, they do not depend on the energy tradeoff parameter λ. These inequalities can be visualised, as done in the wrapped figure below. For a given dataset, $\{\boldsymbol{s}^{[i]}\}$ (blue dots), we can calculate $\left\langle \bar{s}_j^{[i]} \bar{s}_{j'}^{[i]} \right\rangle_i$ and each $\min_i \bar{s}_j^{[i]}$ as they are simple functions of the dataset. For all unit $\boldsymbol{w}$, we can draw the line

$$\boldsymbol{w}^\top \boldsymbol{x} = \sqrt{\sum_{j=1}^{d_s} \left(w_j \min_i \bar{s}_j^{[i]}\right)^2 - \sum_{j,j'\neq j}^{d_s} w_j w_{j'} \left\langle \bar{s}_j^{[i]} \bar{s}_{j'}^{[i]} \right\rangle_i} = \sqrt{\boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w}}. \tag{10}$$

Where we have defined the following matrix:

$$\boldsymbol{S} = \begin{bmatrix} (\min_i s_1^{[i]})^2 & -\langle s_1^{[i]} s_2^{[i]} \rangle_i & -\langle s_1^{[i]} s_3^{[i]} \rangle_i & \dots \\ -\langle s_1^{[i]} s_2^{[i]} \rangle_i & (\min_i s_2^{[i]})^2 & -\langle s_2^{[i]} s_3^{[i]} \rangle_i & \dots \\ -\langle s_1^{[i]} s_3^{[i]} \rangle_i & -\langle s_2^{[i]} s_3^{[i]} \rangle_i & (\min_i s_3^{[i]})^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{11}$$

Each $\boldsymbol{w}$ gives us a line, and if there exists a line (such as the red one) that bounds the source support, then an optimal representation must mix. This exercise also motivates the following equivalent statement of our conditions.



Theorem 2.2. *In the same setting as Theorem 2.1, define the set $E = \{\boldsymbol{y} : \boldsymbol{y}^T \boldsymbol{S}^{-1} \boldsymbol{y} = 1\}$. Then an equivalent statement of Theorem 2.1 is the representation modularises iff E lies within the convex hull of the datapoints.*

This therefore provides a simple test: create $\boldsymbol{S}$ and draw the set E which, if $\boldsymbol{S}$ is positive definite (as it often is), is an ellipse as shown in the wrapped figure above. The optimal representation modularises iff the convex hull of the datapoints encloses E, figure Figure 1b. If $\boldsymbol{S}$ has some positive and some negative eigenvalues, then the optimal representation must mix.

The above results apply to autoencoders trained to linearly encode and decode the pure source vector $\boldsymbol{s}^{[i]}$. In Appendix B.4 we derive similar results for linear mixtures of sources, i.e. where the data the autoencoder receives and reconstructs is $\boldsymbol{x}^{[i]} = \boldsymbol{A}\boldsymbol{s}^{[i]}$ for some mixing matrix $\boldsymbol{A}$.

2.4 RANGE INDEPENDENT VARIABLES MODULARISE

To clarify our result we present a particularly clean special case.

Corollary 2.3. *In the same setting as Theorem 2.1 the optimal representation modularises if all sources are pairwise extreme-point independent, i.e. if for all $j, j' \in [d_s]^2$:*

$$\min_i \left[s_j^{[i]} \middle| s_{j'}^{[i]} \in \left\{ \max_{i'} s_{j'}^{[i']}, \min_{i'} s_{j'}^{[i']} \right\} \right] = \min_i s_j^{[i]}. \tag{12}$$

In other words, if the joint distribution is supported on all extremal corners, the optimal representation modularises.

As presented, our theory has some limitations. It focuses on sources that are 1-dimensional, and uses a specific choice of activity norm, the L2, when others might be more reasonable. In fact, however, the core result that it is the independence of the variables' support (i.e. how rectangular they are) that drives modularity is very broadly true. In Appendix C we show that this core result generalises to other activity norms, and in Appendix D we show that it also applies to the modularisation of multi-dimensional variables such as angles on a 2D circle. Thus, our key takeaway is that the independence of support is what determines whether the optimal representation is modular.

2.5 EMPIRICAL TESTS

Validation of linear autoencoder theory. We show our inequalities correctly predict modularisation. In particular, as an illustration of the precision of our theory, we create a dataset which transitions from inducing modularising to inducing mixing via the removal of a single critical datapoint (Figure 1b). Further, we generate many datasets, create the optimal biological representation, and measure the angle θ between the most-mixed neuron's weight vector and its closest source direction, a proxy for modularity. The left of Figure 1c shows that our theory correctly predicts which datasets are modular ($\theta = 0$). Further, despite our theory being binary (will it modularise or not?), empirically we see that a good proxy for how mixed the optimal representation will be is the degree to which the inequalities in Theorem 2.1 are broken. Finally, on the right of Figure 1c we show that on the same datasets a measure of source statistical interdependence, as used in previous work, does not predict modularisation.

Predictions beyond our theory. From our theory we extract qualitative trends to empirically test in more complex settings. Since extremal points play an outsized role in determining modularisation, we consider three trends that highlight these effects. (1) Datasets from which successively smaller corners have been removed should become successively less mixed, until at a critical threshold the representation modularises. (2) It is vital that not just any data, but rather corner slices, that are removed. Removing similar amounts of random or centrally located data from the dataset should not cause as much mixing. (3) Introducing correlations into a dataset while preserving extreme-point or range independence should preserve modularity relatively well.

3 MODULARISATION IN BIOLOGICALLY INSPIRED NONLINEAR FEEDFORWARD NETWORKS

Motivated by our linear theoretical results, we explore how closely biologically constrained nonlinear networks match our predicted trends. We study nonlinear representations with linear and nonlinear decoding in supervised and unsupervised settings. Surprisingly, coarse properties predicted by our linear theory generalise empirically to these nonlinear settings (Figure 2).

Metrics for representational modularity and inter-source statistical dependence. To quantify the modularity of a nonlinear representation, we design a family of metrics called conditional information-theoretic modularity (CInfoM), an extension of the InfoM metric proposed by Hsu et al. (2023). Intuitively, a representation is modular if each neuron is informative about only a single source. We therefore calculate the conditional mutual information between a neuron's activity and each source variable given all other sources. The conditioning is necessary to remove the effect of sources leaking information about each other; prior works consider independent sources or do not account for this effect. CInfoM then measures the extent to which a neuron specialises to its favourite source, relative to its informativeness about all sources. In order to compare multiple schemes that change $p(\boldsymbol{s})$ in different ways, we report normalised source multiinformation (NSI) as a measure of source statistical interdependence. NSI involves estimating the source multiinformation (aka total correlation) $D_{\mathrm{KL}}(p(\boldsymbol{s}) \parallel \prod_{i=1}^{d_s} p(s_i))$ and normalizing by $\sum_{i=1}^{d_s} H(s_i) - \max_i H(s_i)$. We defer further exposition of the CInfoM metric to Appendix F.

What-where regression. Inspired by the modularisation of what and where into the ventral and dorsal visual streams, we present nonlinear networks with simple binary images in which one pixel is on. The network is trained to concurrently report within which region of the image ("where"), and where within that region ("what") the on pixel is found, producing two outputs which we take as our

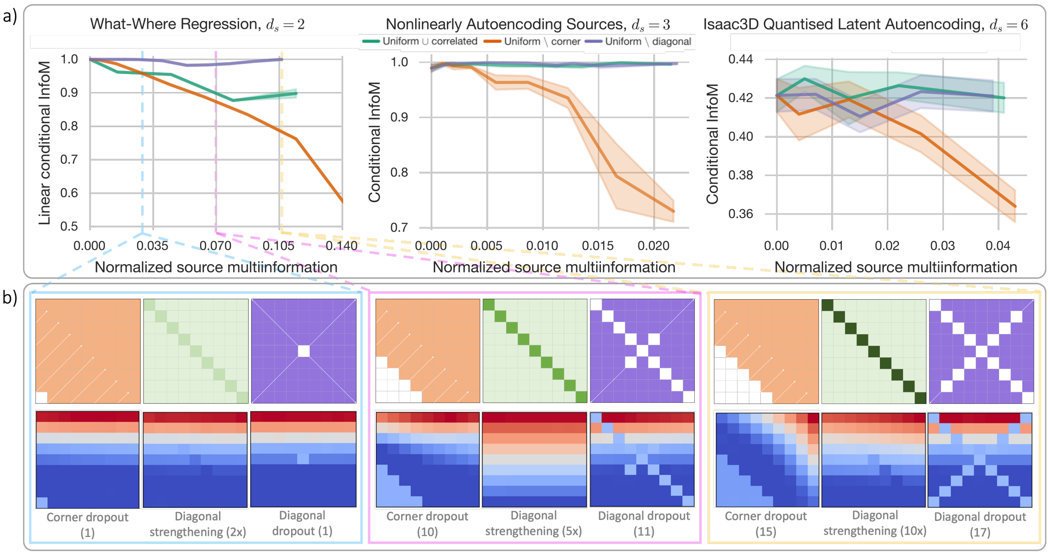


Figure 2: a) We study three tasks of varying source dimensionality d_s, model complexity, supervision signal, and input data modality (Section 3). For each task, we plot conditional InfoM, a measure of modularity, against normalised source multiinformation (NSI), a measure of the statistical dependencies in the sources. Across the three tasks, we observe that cutting corners severely degrades modularity (orange), whereas equivalently changing the NSI in ways that preserve the rectangular support boundary (green, purple) affects modularity less. Shaded regions denote standard error of the mean. b) For What-Where Regression, we visualise the source distributions at various NSI values (top) as well as the tuning of an example neuron (bottom; vertical: What; horizontal: Where).

sources, each an integer between one and nine. (More complex inputs or one-hot labels also work, see Appendix G.) We regularise the activity and weight energies, and enforce nonnegativity using a ReLU. If what and where are independent from one another, e.g. both uniformly distributed, then under our biological constraints (but not without them, see Appendix G) the hidden layer modularises what from where. Breaking the independence of what and where leads to mixed representations in patterns that qualitatively agree with our theory (Figure 2a left): cutting corners from the source support causes increasing mixing. Conversely, making other more drastic changes to the support, such as removing the diagonal, does not induce mixing. Similarly, introducing source correlations while preserving the rectangular support introduces less mixing when compared to corner cutting that induces the same amount of mutual information between what and where. The various source distributions at different NSI values are visualised in Figure 2b.

Nonlinear autoencoding of sources. Next we study a nonlinear autoencoder trained to autoencode multidimensional source variables. Again we find (Figure 2a middle) that under biological constraints independent source variables (NSI $= 0$) result in modular latents and that source corner cutting (orange) induces far more latent mixing compared to introducing source correlations while preserving rectangular support (green), or removing central data (purple).

Disentangled representation learning of images. Finally, for an experiment involving high-dimensional image data, we turn to a recently introduced state-of-the-art disentangling method, quantised latent autoencoding (QLAE; Hsu et al. (2023; 2024)). QLAE is the natural machine learning analogue to our biological constraints. It has two components: (1) the weights in QLAE are heavily regularised, like our weight energy loss, and (2) the latent space is axis-wise quantised, introducing a privileged basis with low channel capacity. In our biologically inspired networks, nonnegativity and activity regularisation conspire to similarly structure the representation: nonnegativity creates a preferred basis, and activity regularisation encourages the representation to use as small a portion of the space as possible. We study the performance of QLAE trained to autoencode a subset of the Isaac3D dataset (Nie, 2019), a full image dataset with well defined underlying latent dimensions . We find the same qualitative patterns emerge: corner cutting is a more important determinant of mixing than range-independent correlations or the removal of centrally located data (Figure 2a right).

4 MODULARISATION IN BIOLOGICALLY INSPIRED RECURRENT NEURAL NETWORKS

Compared to feedforward networks, recurrent neural networks (RNNs) are often a much more natural setting for neuroscience. Excitingly, the core ideas of our analysis for linear autoencoders also apply to recurrent dynamical formulations, and similarly extend to experiments with nonlinear networks.

4.1 LINEAR RNNS

Linear sinusoidal regression. Linear dynamical systems can only autonomously implement exponentially growing, decaying, or stable sinusoidal functions. We therefore study linear RNNs with biological constraints trained to model stable sinusoidal signals at certain frequencies:

$$\boldsymbol{z}(t+\delta t) = \boldsymbol{W}_{\text{rec}}\boldsymbol{z}(t) + \boldsymbol{b}_{\text{rec}}, \quad \boldsymbol{W}_{\text{out}}\boldsymbol{z}(t) = \begin{bmatrix}\cos(\omega_1 t + \phi_1)\\ \cos(\omega_2 t + \phi_2)\end{bmatrix}, \quad \boldsymbol{z}(t) \geq \boldsymbol{0}. \tag{13}$$

We study the optimal nonnegative, efficient, recurrent representations, $\boldsymbol{z}_t$, and show that whether the representations of the two frequencies within $\boldsymbol{z}_t$ modularises depends on their ratio. We prove and verify empirically that if one frequency is an integer multiple of the other the encodings mix, whereas if their ratio is irrational they should modularise. Further, we show empirically, and prove in limited settings, that rational non-harmonic ratios should modularise (Appendix E). The intuition for this result is much the same as the linear autoencoding setting: the natural notions of sources are the signals $(\cos(\omega_1 t), \sin(\omega_1 t), \cos(\omega_2 t), \sin(\omega_2 t))$. Using these sources, we must simply ask: does their support allow for a reduction in activity energy via mixing? In Figure 3a, we visualise the source support for the three prototypical relationships between ω_1 and ω_2: irrational, rational (but not harmonic), and harmonic. Results of neural network verifications are in Figure 3b (details Appendix I). In the irrational case, the source support is essentially rectangular, so the model modularises. In the harmonic case, large chunks of various corners are missing from the source support, so the model mixes. In the rational case, even though the source support is quite sparse, the corners are sufficiently present such that modularising is still optimal.

Modularisation of grid cells. We now show that these spectral ideas can explain modules of grid cells (literature review: Appendix A. Grid cells are neurons in the mammalian entorhinal cortex that fire in a hexagonal lattice of positions (Hafting et al., 2005). They come in groups, called modules; grid cells within the same module have receptive fields (firing patterns) that are translated copies of the same lattice, and different modules are defined by their different lattice (Stensola et al., 2012). Current theories suggest that the grid cell system can be modelled as a RNN with activations built from linear combinations of frequencies (Dorrell et al., 2023), just like the linear RNNs considered here. Importantly, these grid cell theories use the same biological constraints as in this work and show that modules form because the optimal code contains non-harmonically related frequencies that are encoded in different neurons (Figure 3e top). This modularisation can now be theoretically justified in our framework, since non-harmonic frequencies are range-independent and so should be modularised (Figure 3e bottom).

4.2 NONLINEAR RNNS

Mixed sinusoidal regression for nonlinear RNNs. We train nonlinear ReLU RNNs with biologically inspired constraints to perform a frequency mixing task (details Appendix I). We provide a pulse input $P_\omega(t) = \mathbb{I}\left[\text{mod}_\omega(t) = 0\right]$ at two frequencies, and the network has to output the resulting "beats" and "carrier" signals:

$$\boldsymbol{z}(t+\Delta t) = \text{ReLU}\left(\boldsymbol{W}_{\text{rec}}\boldsymbol{z}(t) + \boldsymbol{W}_{\text{in}}\begin{bmatrix}P_{\omega_1}(t)\\ P_{\omega_2}(t)\end{bmatrix} + \boldsymbol{b}_{\text{rec}}\right), \; \begin{bmatrix}\cos([\omega_1 - \omega_2]t)\\ \cos([\omega_1 + \omega_2]t)\end{bmatrix} = \boldsymbol{W}_{\text{out}}\boldsymbol{z}_t + \boldsymbol{b}_{\text{out}}. \tag{14}$$

Results are in Figure 3c. Identical range-dependence properties but applied to the frequencies $\omega_1 - \omega_2$ and $\omega_1 + \omega_2$ determine whether or not the network modularises: irrational, range-independent frequencies modularise; harmonics, with their large missing corners, mix; and other rationally related frequencies are range-dependent but no sufficient corner is missing, so they modularise.

Modularisation in nonlinear teacher-student distillation. To test our predictions of when RNNs modularise, but in settings more realistic than pure frequencies, we generate training data trajectories

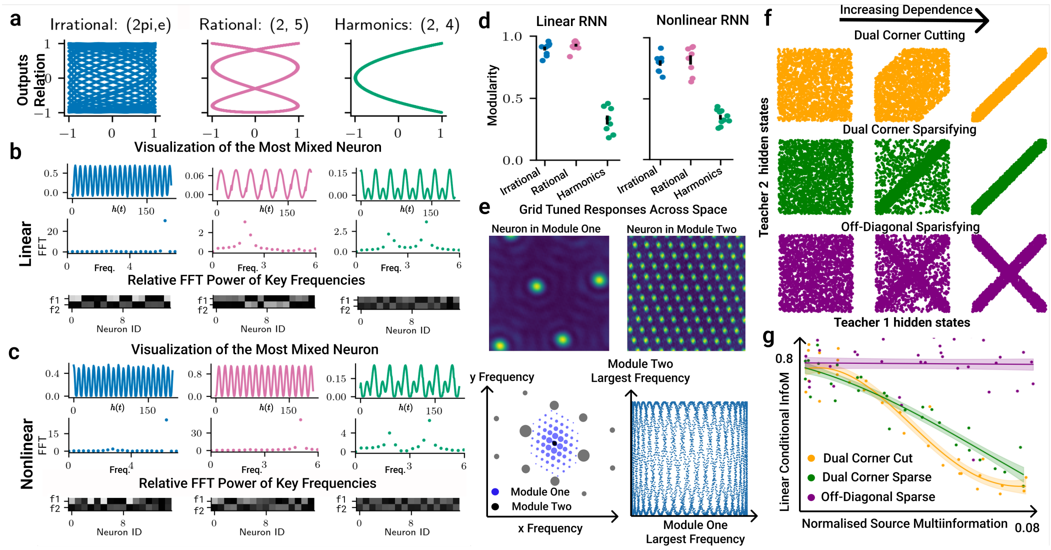


Figure 3: a) Source support visualisations: $\cos(\omega_1 t)$ vs. $\cos(\omega_2 t)$. b) The activity of the most mixed neuron in an optimal solution, the same neuron's Fourier spectrum, and the population's Fourier spectrum in the two task frequencies show clear modularisation in the irrational and rational cases, and clear mixing the in the harmonic case. c) Same as (b), but with *nonlinear* RNNs. d) Modularity of RNNs trained on 10 frequency pairs. e) Top: Two optimal grid cells from Dorrell et al. (2023) in different modules. Bottom: The modularisation of these two lattices can be understood from the range-independence of their constituent frequencies, as shown in the joint distribution of the most significant frequency in each module. f) Plot of joint distribution of hidden state activity for two teacher RNNs neurons as we increase the dependence in three different ways. g) Trends of modularity scores of student RNN in three different cases qualitatively agree with our theory.

from randomly initialised teacher RNNs with tanh activation function, and then train student RNNs (with a ReLU activation function) on these trajectories. The student's representation is constrained to be nonnegative (via its ReLU) and has its activity and weights regularised (see Appendix I.2 for details). Using carefully chosen inputs at each timestep, we are able to precisely control the teacher RNN hidden state activity (i.e., the source distribution). This allows us to change correlations/statistical independence of the hidden states, while either maintaining or breaking range independence. We consider three settings (Figure 3e). First, when statistical and range independence are progressively broken (in orange). Second, where statistical, and to a lesser extent range, independence gets progressively broken (in green). Third, where only statistical independence gets progressively broken (in purple). We qualitatively observe, as our linear theory predicts, that students learns modular representations when range independence is maintained regardless of the statistical dependence of the teacher RNN hidden states (Figure 3f).

5 MODULAR OR MIXED CODES FOR SPACE & REWARD

We now apply our results to neuroscience to understand a puzzling difference in two seemingly similar recordings from entorhinal cortex. This brain area has been thought to contain precisely modular neurons, such as the grid cell code for self-position (Hafting et al., 2005), object vector cells that fire at a particular displacement from objects (Høydal et al., 2019), and heading direction cells (Taube, 2007). Two recent papers examined the influence of rewarded locations on the grid cell code and find differing effects on the modularity of grid cells. Butler et al. (2019) find that rewards rotate the grid cells, but preserve their pure spatial coding, while Boccara et al. (2019) find that the grid cells warp towards the rewards, becoming mixed-selective to reward and space.

Whittington et al. (2023a) study this discrepancy and point to the importance of the reward distribution in these two tasks: Boccara et al. (2019) fix the positions of the possible rewards during one day, whereas Butler et al. (2019) alternate the animals between periods of free-foraging for randomly placed rewards and periods of specific rewarded locations. However, the arguments of Whittington et al. (2023a), which rely on source independence, are insufficient to explain these modularisation effects, as in neither case are reward and position independent; even in the experiments of Butler et al. (2019) there are regions of space that are much more likely to be rewarded.

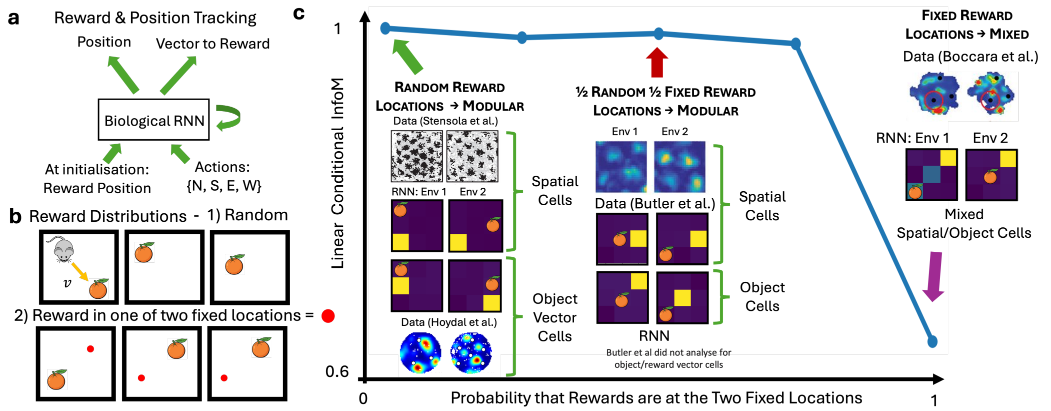


Figure 4: a, b) We train linear RNNs to integrate a sequence of actions to report their current position and displacement from a reward. In some proportion of the rooms, rewards are randomly scattered; in the rest, they are in one of two fixed locations. c) Matching the theory, if some of the rooms are random then reward and position are range-independent, and their representation modularises, as in Stensola et al. (2012) or Butler et al. (2019); if in all rooms the rewards occur in one of two fixed locations the representation mixes, with neurons encoding both reward and self position, as in Boccara et al. (2019).

However, with our improved understanding of modularisation, this makes sense. Despite Butler et al. (2019) correlating reward and position, critically, they leave them *range independent*—all combinations of reward and position are possible. On the other hand, Boccara et al. (2019) make certain combinations of reward and position impossible, making them not just correlated, but *range correlated*. As such, our theory matches this modularisation pattern. To test this we train a linear RNN with biological constraints (details Appendix J) to report both its self-position and displacement from a reward as it moves around an environment (Figure 4). We train RNNs on settings with different relationships between reward and position; for some RNNs the rewards are in fixed positions in every environment (range dependent and statistically dependent), for others reward and position are uniformly random in each room (range independent and statistically independent), and for a final group have both settings in different proportions (range independent but statistically dependent). As we vary the proportion of fixed rooms, we find the optimal representation stays modular, containing separate spatial and reward-vector cells as in Butler et al. (2019), until all the sampled rooms have fixed reward positions, at which point neurons become mixed selective to reward and self-position, as in Boccara et al. (2019). As such, our theory now covers all known modularity results about grid cells.

Missed Sources and Mixed Selectivity In contrast to the beautifully modular neurons in entorhinal cortex, recent work has highlighted neurons with mixed tuning to multiple navigational variables, such as position, heading direction, or speed Hardcastle et al. (2017). We use our theory to highlight a potential artifact of this type of analysis. A purportedly mixed selective neuron may (in part) be purely selective for another unanalysed variable that is itself correlated with the measured spatial variables or behaviour. We highlight a simple example of this effect in Figure 5a, b. Imagine a mouse is keeping track of three objects as it moves in an environment (Figure 5a); we model this as a linear RNN with biological constraints that must report the displacement of all objects from itself (Figure 5b). We make the object positions range independent but correlated. As predicted by our theory, the RNN modularises these range-independent variables such that each neuron only encodes one object (Figure 5c). However, if an experimenter were only aware of two of the three objects, they would instead analyse the neural tuning with respect to those two objects. Due to the statistical dependencies between object positions, they would find mixed selective neurons that are in reality purely coding for the missing object (Figure 5d). We do not believe that all neurons in entorhinal cortex are modular; for example, there is clear evidence that neurons co-tuned to position and velocity are meaningfully mixed-selective, implementing the path-integration updates in the grid cell system Vollan et al. (2024). Indeed, many of the neurons analysed in Hardcastle et al. (2017) *likely are* mixed selective. However, it is also seems likely that in more exploratory analyses such as these, mixed selectivity might often arise from missing encoded variables in the analysis.

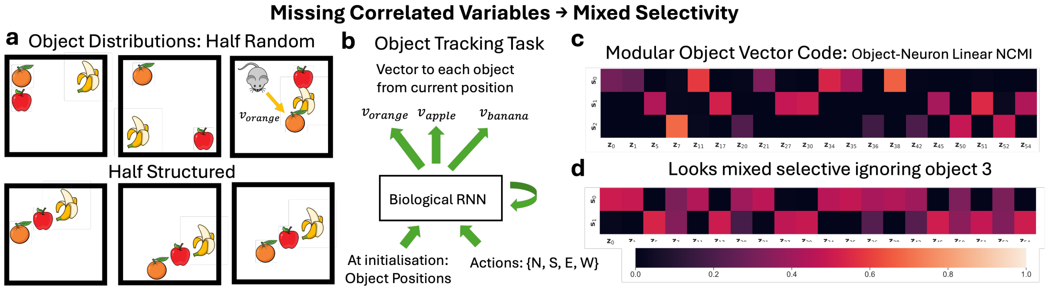

Missing Correlated Variables → Mixed Selectivity
a Object Distributions: Half Random
v_{orange}
Half Structured
b Object Tracking Task
Vector to each object from current position
v_{orange} v_{apple} v_{banana}
Biological RNN
At initialisation: Object Positions
Actions: {N, S, E, W}
c Modular Object Vector Code: Object-Neuron Linear NCMI
s_0
s_1
s_2
z_0 z_1 z_5 z_7 z_{11} z_{17} z_{20} z_{21} z_{27} z_{30} z_{34} z_{35} z_{36} z_{38} z_{42} z_{45} z_{50} z_{51} z_{52} z_{54}
d Looks mixed selective ignoring object 3
s_0
s_1
z_0
0.0
0.2
0.4
0.6
0.8
1.0

Figure 5: a, b) We train linear RNNs to report displacement to three objects as an agent moves within many rooms. In each room, object instances are either completely random (a, top) or clustered in a random line (a, bottom); object positions are therefore range independent but correlated. c) The neurons are modular: each neuron's activity is conditionally informative of only a single object given the others. d) However, if an experimenter were only aware of two of the three objects, the neurons that purely encode the disregarded object would appear mixed selective due to the statistical dependencies between objects.

6 DISCUSSION

We present precise constraints on source distributions that cause linear feedforward and recurrent networks with biological constraints to modularise. These constraints depend on the co-range properties of variables, matching patterns of modularisation in both nonlinear networks and the brain.

Limitations & future work. That said, there remain many aspects of our theory that need improvement. Deriving necessary (rather than sufficient) modularisation conditions for other norms and of multidimensional sources, extending to nonlinear settings, or to understand a less binarised notion of modularity (rather than perfectly modular or not) are attractive directions. Additionally, our model is a purposefully simple model, there are many biologically pertinent effects whose impact on modularity would be interesting to study, such as connection sparsity, anatomical constraints, noise, or requiring additional computational roles from the network. Finally, the theory needs testing beyond entorhinal cortex. In Appendix K we present preliminary analysis that suggests similar ideas of range-independence can help explain discrepancies in population orthogonalisation results from PFC (Panichello and Buschman, 2021; Xie et al., 2022), which we hope to develop further.

Mixed selectivity. In neuroscience our findings add nuance to the ongoing debate over mixed vs. modular neural coding. Theories of *nonlinear* mixed selectivity have argued that, analogously to a nonlinear kernel, such schemes permit linear readouts to decode nonlinear functions of the sources Rigotti et al. (2013). This is likely a key part of the mixed selectivity found in some brain areas, like the cerebellum Lanore et al. (2021), mushroom body (Aso et al., 2014), and perhaps certain prefrontal or hippocampal representations (Bernardi et al., 2020; Boyle et al., 2024). However, our theory raises the possibility for other explanations of both nonlinear and linear mixed selectivity that do *not* require tasks with nonlinear functions of the sources. First, purely from energy efficiency, our work analytically shows when range-dependent variables should be encoded in linearly mixed-selective representations, and empirically shows the same ideas apply to nonlinear mixed-selective representations Figure 2. Further, since in our theory even statistically dependent variables might be represented modularly, we show that this causes problems for analysis techniques that do not account for all the variables encoded in a population, leading to spurious mixed selectivity Section 5. Future work could usefully developed principled approaches to distinguishing these options.

Disentangling what? In machine learning, our work bolsters the importance of range over statistical independence. We agree with Roth et al. (2023), who argue that range independence is the more meaningful form of independence. Further, our conditions have qualitatively similarities to sufficient spread conditions from the matrix factorisation literature Tatli and Erdogan (2021a;b), suggesting a deeper correspondence. It is intriguing that the constraints inspired by biology, which hold promise for disentangling (Whittington et al., 2023a), and have links to state-of-the-art disentangling methods (Hsu et al., 2023; 2024), are sensitive to precisely this form of independence.

Reproducibility statement. For our theoretical contributions, we have included detailed proofs in the supplementary material. For our empirical results, We have included code for all of our experiments. We believe this to be sufficient for any interested party to reproduce our work.

REFERENCES

Yoshinori Aso, Daisuke Hattori, Yang Yu, Rebecca M Johnston, Nirmala A Iyer, Teri-TB Ngo, Heather Dionne, LF Abbott, Richard Axel, Hiromu Tanimoto, et al. The neuronal architecture of the mushroom body provides a logic for associative learning. *elife*, 3:e04577, 2014.

Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.

Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01), 1961.

Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Silvia Bernardi, Marcus K Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4): 954–967, 2020.

Charlotte N Boccara, Michele Nardin, Federico Stella, Joseph O'Neill, and Jozsef Csicsvari. The entorhinal cognitive map is attracted to goals. *Science*, 363(6434):1443–1447, 2019.

Matthew M Botvinick and David C Plaut. Short-term memory for serial order: a recurrent neural network model. *Psychological review*, 113(2):201, 2006.

Lara M Boyle, Lorenzo Posani, Sarah Irfan, Steven A Siegelbaum, and Stefano Fusi. Tuned geometries of hippocampal representations meet the computational demands of social memory. *Neuron*, 2024.

Bariscan Bozkurt, Cengiz Pehlevan, and Alper Erdogan. Biologically-plausible determinant maximization neural networks for blind separation of correlated sources. *Advances in Neural Information Processing Systems*, 35:13704–13717, 2022.

Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology*, 5(2):e1000291, 2009.

William N Butler, Kiah Hardcastle, and Lisa M Giocomo. Remembered reward locations restructure entorhinal spatial maps. *Science*, 363(6434):1447–1452, 2019.

William Dorrell, Peter E Latham, Timothy EJ Behrens, and James CR Whittington. Actionable neural representations: Grid cells from minimal constraints. In *The Eleventh International Conference on Learning Representations*, 2023.

Laura Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *bioRxiv*, pages 2022–08, 2022.

Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. The role of population structure in computations through neural dynamics. *Nature neuroscience*, 25(6): 783–794, 2022.

Lea Duncker, Laura Driscoll, Krishna V Shenoy, Maneesh Sahani, and David Sussillo. Organizing recurrent network dynamics by task-computation to enable continual learning. *Advances in neural information processing systems*, 33:14387–14397, 2020.

Mhairi Dunion, Trevor McInroe, Kevin Sebastian Luck, Josiah Hanna, and Stefano Albrecht. Conditional mutual information for disentangled representations in reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Mohamady El-Gaby, Adam Loyd Harris, James CR Whittington, William Dorrell, Arya Bhomick, Mark W Walton, Thomas Akam, and Tim EJ Behrens. A cellular basis for mapping behavioural structure. *bioRxiv*, pages 2023–11, 2023.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

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

Julia J Harris, Renaud Jolivet, and David Attwell. Synaptic energy use and supply. *Neuron*, 75(5): 762–777, 2012.

Daniella Horan, Eitan Richardson, and Yair Weiss. When is unsupervised disentanglement possible? *Advances in Neural Information Processing Systems*, 34:5150–5161, 2021.

Ola Hössjer and Arvid Sjölander. Sharp lower and upper bounds for the covariance of bounded random variables. *Statistics & Probability Letters*, 182:109323, 2022.

Kyle Hsu, William Dorrell, James Whittington, Jiajun Wu, and Chelsea Finn. Disentanglement via latent quantization. *Advances in Neural Information Processing Systems*, 36, 2023.

Kyle Hsu, Jubayer Ibn Hamid, Kaylee Burns, Chelsea Finn, and Jiajun Wu. Tripod: Three complementary inductive biases for disentangled representation learning. In *International Conference on Machine Learning*, 2024.

David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

Øyvind Arne Høydal, Emilie Ranheim Skytøen, Sebastian Ola Andersson, May-Britt Moser, and Edvard I. Moser. Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752):400–404, April 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1077-7. URL `http://dx.doi.org/10.1038/s41586-019-1077-7`.

Huseyin A Inan and Alper T Erdogan. An extended family of bounded component analysis algorithms. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 442–445. IEEE, 2014.

Devon Jarvis, Richard Klein, Benjamin Rosman, and Andrew Saxe. On the specialization of neural modules. In *The Eleventh International Conference on Learning Representations*, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL `https://arxiv.org/abs/1412.6980`.

David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Frederic Lanore, N Alex Cayco-Gajic, Harsha Gurnani, Diccon Coyle, and R Angus Silver. Cerebellar granule cell axons support high-dimensional representations. *Nature neuroscience*, 24(8):1142–1150, 2021.

Simon B Laughlin. Energy as a constraint on the coding and processing of sensory information. *Current opinion in neurobiology*, 11(4):475–480, 2001.

Jin Hwa Lee, Stefano Sarao Mannelli, and Andrew Saxe. Why do animals need shaping? a theory of task composition and curriculum learning. *arXiv preprint arXiv:2402.18361*, 2024.

Laureline Logiaco, LF Abbott, and Sean Escola. Thalamic control of cortical dynamics in a model of flexible motor sequencing. *Cell reports*, 35(9), 2021.

Alexander Mathis, Andreas VM Herz, and Martin Stemmler. Optimal population codes for space: grid cells outperform place cells. *Neural computation*, 24(9):2280–2317, 2012.

Aran Nayebi, Alexander Attinger, Malcolm Campbell, Kiah Hardcastle, Isabel Low, Caitlin S Mallory, Gabriel Mel, Ben Sorscher, Alex H Williams, Surya Ganguli, et al. Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. *Advances in Neural Information Processing Systems*, 34:12167–12179, 2021.

Weili Nie. High resolution disentanglement datasets, 2019. URL `https://github.com/NVlabs/High-res-disentanglement-datasets`.

Matthew F Panichello and Timothy J Buschman. Shared mechanisms underlie the control of working memory and attention. *Nature*, 592(7855):601–605, 2021.

Cengiz Pehlevan, Sreyas Mohan, and Dmitri B Chklovskii. Blind nonnegative source separation using biological neural networks. *Neural computation*, 29(11):2925–2954, 2017.

Emilia P Piwek, Mark G Stokes, and Christopher Summerfield. A recurrent neural network model of prefrontal brain activity during a working memory task. *PLoS Computational Biology*, 19(10): e1011555, 2023.

Mark D Plumbley. Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543, 2003.

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

Brian C Ross. Mutual information between discrete and continuous data sets. *PloS one*, 9(2):e87357, 2014.

Karsten Roth, Mark Ibrahim, Zeynep Akata, Pascal Vincent, and Diane Bouchacourt. Disentanglement of correlated factors via hausdorff factorized support. In *The Eleventh International Conference on Learning Representations*, 2023.

Axel Ruhe. Perturbation bounds for means of eigenvalues and invariant subspaces. *BIT Numerical Mathematics*, 10(3):343–354, 1970.

Andrew Saxe, Shagun Sodhani, and Sam Jay Lewallen. The neural race reduction: Dynamics of abstraction in gated networks. In *International Conference on Machine Learning*, pages 19287–19309. PMLR, 2022.

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wolczyk, Alexandra Maria Proca, Johannes Von Oswald, Razvan Pascanu, Joao Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. In *The Twelfth International Conference on Learning Representations*, 2024.

Pavithraa Seenivasan and Rishikesh Narayanan. Efficient information coding and degeneracy in the nervous system. *Current opinion in neurobiology*, 76:102620, 2022.

Jianghong Shi, Eric Shea-Brown, and Michael Buice. Learning dynamics of deep linear networks with multiple pathways. *Advances in neural information processing systems*, 35:34064–34076, 2022.

Daniel Soudry and Dror Speiser. Maximal minimum for a sum of two (or more) cosines, 2015. URL `https://mathoverflow.net/questions/209071/maximal-minimum-for-a-sum-of-two-or-more-cosines`.

Hanne Stensola, Tor Stensola, Trygve Solstad, Kristian Frøland, May-Britt Moser, and Edvard I Moser. The entorhinal grid map is discretized. *Nature*, 492(7427):72–78, 2012.

Gokcan Tatli and Alper T Erdogan. Generalized polytopic matrix factorization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3235–3239. IEEE, 2021a.

Gokcan Tatli and Alper T Erdogan. Polytopic matrix factorization: Determinant maximization based criterion and identifiability. *IEEE Transactions on Signal Processing*, 69:5431–5447, 2021b.

Jeffrey S Taube. The head direction signal: origins and sensory-motor integration. *Annu. Rev. Neurosci.*, 30(1):181–207, 2007.

Kay M Tye, Earl K Miller, Felix H Taschbach, Marcus K Benna, Mattia Rigotti, and Stefano Fusi. Mixed selectivity: Cellular computations for complexity. *Neuron*, 2024.

Abraham Z Vollan, Richard J Gardner, May-Britt Moser, and Edvard I Moser. Left-right-alternating theta sweeps in the entorhinal-hippocampal spatial map. *bioRxiv*, pages 2024–05, 2024.

Xue-Xin Wei, Jason Prentice, and Vijay Balasubramanian. A principle of economy predicts the functional architecture of grid cells. *Elife*, 4:e08362, 2015.

James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263, 2020.

James CR Whittington, Will Dorrell, Surya Ganguli, and Timothy Behrens. Disentanglement with biological constraints: A theory of functional cell types. In *The Eleventh International Conference on Learning Representations*, 2023a.

James CR Whittington, William Dorrell, Timothy EJ Behrens, Surya Ganguli, and Mohamady El-Gaby. On prefrontal working memory and hippocampal episodic memory: Unifying memories stored in weights and activity slots. *bioRxiv*, pages 2023–11, 2023b.

Yang Xie, Peiyao Hu, Junru Li, Jingwen Chen, Weibin Song, Xiao-Jing Wang, Tianming Yang, Stanislas Dehaene, Shiming Tang, Bin Min, et al. Geometry of sequence working memory in macaque prefrontal cortex. *Science*, 375(6581):632–639, 2022.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020.

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

Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ica: Sparsity and beyond. *Advances in Neural Information Processing Systems*, 35:16411–16422, 2022.

A RELATED WORKS

Machine learning. The disentangled representation learning community has found inductive biases that lead networks to prefer modular or disentangled representations of nonlinear mixtures of independent variables. Examples include sparse temporal changes (Klindt et al., 2020), smoothness or sparseness assumptions in the generative model (Zheng et al., 2022; Horan et al., 2021), and axis-wise latent quantisation (Hsu et al., 2023; 2024). Relatedly, studies of tractable network models have identified a variety of structural aspects, in either the task or architecture, that lead to modularisation, including learning dynamics in gated linear networks (Saxe et al., 2022), architectural constraints in linear networks (Jarvis et al., 2023; Shi et al., 2022), compositional tasks in linear hypernetworks and nonlinear teacher-student frameworks (Schug et al., 2024; Lee et al., 2024), or task-sparsity in linear autoencoders (Elhage et al., 2022).

Neuroscience. Theories prompted by the recognition of mixed-selectivity in biological neurons have argued that nonlinearly mixed selective codes might exist to enable a linear readout to flexibly decode any required categorisation (Rigotti et al., 2013), suggesting a generalisability-flexibility tradeoff between modular and nonlinear mixed encodings (Bernardi et al., 2020). Modelling work has studied task-optimised network models of neural circuits, some of which have recovered mixed encodings (Nayebi et al., 2021). However, other models trained on a wide variety of cognitive tasks have found that networks contain meaningfully modularised components (Yang et al., 2019; Driscoll et al., 2022; Duncker et al., 2020). Dubreuil et al. (2022) study recurrent networks trained on similar cognitive tasks and find that, depending on the task structure, neurons in the trained networks can be grouped into populations defined by their connectivity patterns. To this developing body, our work and its predecessor (Whittington et al., 2023a) present understanding of when simple biological constraints would encourage encodings of many variables to be modularised into populations of single pure-tuned variables, or when mixed-selectivity would be preferred. Compared to (Whittington et al., 2023a) which only works when source variables are statistically independent, our work determines modularity for (almost) any dataset even when source variables are not independent. This is a dramatically more complete understanding of the boundary between mixed and modular codes that allows us to understand behaviour in both biological and artificial neurons that previously eluded explanation.

Independent components analysis and matrix factorization. Our linear autoencoding problem (Theorem 2.1) bears some similarity to linear independent components analysis (ICA) and matrix factorization (MF). These assume data $\boldsymbol{y}_i$ is linearly generated from a set of constrained sources, $\boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{s}_i$, and build algorithms to recover $\boldsymbol{s}_i$ from $\boldsymbol{y}_i$. The literature studies different combinations of (1) assumptions on the sources $\boldsymbol{s}_i$, e.g. independence (Bell and Sejnowski, 1995), nonnegativity(Plumbley, 2003), or boundedness Inan and Erdogan (2014); (2) assumptions on the mixing matrix $\boldsymbol{A}$, e.g. nonnegative in nonnegative matrix factorisation and (3) algorithms, often deriving identifiability results: conditions that the data must satisfy for a given algorithm to succeed. Some work even derives biological implementations of such factorisation algorithms (Pehlevan et al., 2017; Bozkurt et al., 2022). In our work we study the conditions under which a particular biologically inspired representation modularises a set of bounded sources. We provide conditions on the sources under which the optimal representation is modular, morally similar to identifiability results (indeed, other identifiability results, like ours, sometimes take the form of sufficient spread conditions (Tatli and Erdogan, 2021a;b)). We differ from previous work in the representation we study (which has only been studied before by Whittington et al. (2023a)), the form of our identifiability result (which is both necessary and sufficient, has not been derived before, and assumes only that the sources are bounded), and these developments allows us to explain both recurrent modularisation and neural data.

Grid Cell Modules One of the many surprising features of grid cells is their modular structure: each grid cell is a member of one module, of which there are a small number in rats. Grid cells from the same module have receptive fields that are translated versions of one another (Stensola et al., 2012). Previous work has built circuit models of such modules showing how they might path-integrate (Burak and Fiete, 2009), or has assumed the existence of multiple modules, then shown that they form a good code for space, and that parameter choices such as the module lengthscale ratio can be extracted from optimal coding arguments (Mathis et al., 2012; Wei et al., 2015). However, neither of these directions shows why, of all the ways to code space, a multi-modular structure is best. Dorrell et al. (2023) study a normative problem that explains the emergence of multiple modules grid cells as the best of all possible codes, but their arguments for the emergence of multiple grid modules are relatively intuitive. In this work we are able to formalise parts of their argument.

B OPTIMAL REPRESENTATIONS IN POSITIVE LINEAR AUTOENCODERS

We develop our results in three parallel ways. First, in Appendix B.1, we study simple two source problems and pedagogically derive our results. Second, in Appendix B.2, we state the full inequality constraints, Theorem 2.1, and formally prove them using the same type of argument as in the pedagogical approach. Finally, in Appendix B.4, we study the more general setting of invertible linear mixtures of sources and use a third, pertubative, approach to derive more general conditions under which a modular solution is a local optima. These conditions recover our main inequalities as a special case, but in a slightly weaker format, as they only guarantee the local optimality of modular solutions, unlike our main results. The three presentations of the result are independent, and readers may choose whichever they find more palatable. Then, in Appendix B.3, we prove the corollary Theorem 2.3. Finally, in Appendix B.5, we show the equivalence between the two characterisations of the condition, via either the set of inequalities or the convex hull.

B.1 INTUITIVE DEVELOPMENT FOR TWO SOURCES

B.1.1 PROBLEM STATEMENT

To begin, we study our simplest setting: a positive linear autoencoder that has to represent two bounded, scalar, mean-zero, sources, $x^{[i]}$ and $y^{[i]}$. These are encoded in a representation $\boldsymbol{z}^{[i]} \in \mathbb{R}^N$, where N is the number of neurons, which we will always assume to be as large as we need it to be, and in particular larger than the number of encoded variables. Our first constraint is given by the architecture. The representation is a linear function of the inputs plus a constant bias, and you must be able to decode the variables from the representation using an affine readout transformation:

$$\boldsymbol{z}^{[i]} = \boldsymbol{W}_{\text{in}} \begin{bmatrix} x^{[i]} \\ y^{[i]} \end{bmatrix} + \boldsymbol{b}_{\text{in}} \qquad \boldsymbol{W}_{\text{out}} \cdot \boldsymbol{z}^{[i]} + \boldsymbol{b}_{\text{out}} = \begin{bmatrix} x^{[i]} \\ y^{[i]} \end{bmatrix} \tag{15}$$

Where $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{b}_{\text{in}}$ are the readin weight matrix and bias, and $\boldsymbol{W}_{\text{out}}$ and $\boldsymbol{b}_{\text{out}}$ are the readout weight matrix and bias.

Our second constraint, inspired on the one hand by the non-negativity of biological neural firing rates, and on the other by the success of the ReLU activation function, is the requirement that the representation is non-negative: $\boldsymbol{z}^{[i]} \geq \boldsymbol{0}$.

Subject to these constraints we optimise the representation to minimise an energy-use loss:

$$\mathscr{L} = \frac{1}{D}\sum_{i=1}^{D} ||\boldsymbol{z}^{[i]}||^2 + \lambda(||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2) = \langle ||\boldsymbol{z}^{[i]}||^2 \rangle_i + \lambda(||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2) \tag{16}$$

This loss is inspired on the one hand by biology, in particular by the efficient coding hypothesis (Attneave, 1954; Barlow et al., 1961) and its descendants. These theories argue that neural firing should perform its functional role (e.g. encoding information) maximally energy-efficiently, for example by using the smallest firing rates possible, and has been used to understand neural responses across tasks, brain areas, individuals, and organisms (Laughlin, 2001; Seenivasan and Narayanan, 2022). Our loss can be seen as a slight generalisation of this idea, by minimising energy use both through firing rates and through synapses (Harris et al., 2012). On the other hand, this loss is similar to weight decay, a widely used regularisation technique in machine learning, that has long been linked to a simplicity bias in neural networks (Krogh and Hertz, 1991).

Our question can now be simply posed. What properties of the sources $x^{[i]}$ and $y^{[i]}$ ensure that they are optimally represented in disjoint sets of neurons? Equivalently, when does the representation modularise? The arguments of Whittington et al. (2023a) can be used to show that if the two sources are statistically independent they should optimally modularise. We will find a much weaker set of conditions are necessary and sufficient for modularisation. In particular, we derive precise conditions on the range of allowed (x, y) pairs that ensures modularising is optimal. For example, we will show that if the sources are range-symmetric ($|\min_i x^{[i]}| = \max_i x^{[i]}$) and extreme point independent, meaning $\min_i(x^{[i]} + y^{[i]}) = \min_i x^{[i]} + \min_i y^{[i]}$, they should modularise.

The structure of our argument goes as follows, by assumption the representation is a affine transformation of the inputs:

$$\boldsymbol{z}^{[i]} = \boldsymbol{u}x^{[i]} + \boldsymbol{v}y^{[i]} + \boldsymbol{b} \tag{17}$$

We will show that, for any fixed encoding sizes $||\boldsymbol{v}||$ and $||\boldsymbol{u}||$, the weight loss ($||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2$) is minimised by orthogonalising $\boldsymbol{v}$ and $\boldsymbol{u}$, in particular, by modularising the representation. We will then derive the conditions under which, for fixed encoding size, the activity loss is minimised by modularising the representation. If, whatever the encoding sizes, both losses are minimised by modularising, and the activity loss is minimised only by modularising (i.e. there are no other solutions that are equally good), the optimal representation is modular. Conversely, if the activity loss is not minimised by modularising, then you can always decrease the loss by slightly mixing the representation, hence the optimal representation is mixed.

B.1.2 CONDITIONS FOR MODULARISATION

For simplicity we will make the following assumptions, that will be relaxed later:

1. The sources are linearly uncorrelated, $\langle x^{[i]} y^{[i]} \rangle_i = \frac{1}{D} \sum_{i=1}^{D} x^{[i]} y^{[i]} = 0$
2. The sources are range-symmetric around zero, i.e. $|\min_i x^{[i]}| = \max_i x^{[i]}$ and $|\min_i y^{[i]}| = \max_i y^{[i]}$

We will consider the two losses in turn.

B.1.3 WEIGHT LOSS

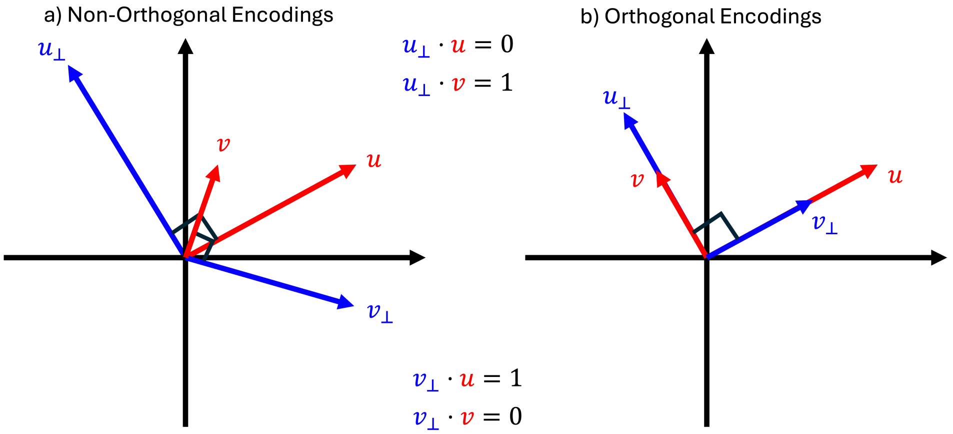


Figure 6: Schematics showing that modular representations minimise readout weight energy, $||W_{\text{out}}||_F^2$, independent of encoding size. Aligning the encoding vectors (a) necessarily requires longer pseudoinverse vectors than an orthogonal solution (b), for example a modular solution.

First, for a given linear representation of the form in equation 17, the minimum squared norm readout matrix has the following form:

$$\boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{v}_\perp^T \\ \boldsymbol{u}_\perp^T \end{bmatrix} \tag{18}$$

Where $\boldsymbol{v} \perp$ and $\boldsymbol{u}_\perp$ are the two vectors in the span of $\boldsymbol{u}$ and $\boldsymbol{v}$ with the property that $\boldsymbol{v}_\perp^T \boldsymbol{v} = 0$ and $\boldsymbol{v}_\perp^T \boldsymbol{u} = 1$, and the equivalent conditions for $\boldsymbol{u}_\perp$ (as in figure 6). To convince yourself of this consider the fact that these are the only two vectors in this plane that will produce the desired output, and you could add off-plane components to them, but that would only increase the weight loss for no gain, since:

$$||\boldsymbol{W}_{\text{out}}||_F^2 = \text{Tr}[\boldsymbol{W}_{\text{out}} \boldsymbol{W}_{\text{out}}^T] = ||\boldsymbol{u}_\perp||^2 + ||\boldsymbol{v}_\perp^2|| \tag{19}$$

These vectors, $\boldsymbol{v} \perp$ and $\boldsymbol{u}_\perp$, are the rows of the psuedoinverse, so henceforth we shall call these vectors the pseudoinverse vectors.

Then, with θ denoting the angle between $\boldsymbol{v}$ and $\boldsymbol{u}$, the readout loss can be developed using simple trigonometry:

$$||\boldsymbol{W}_{\text{out}}||_F^2 = ||\boldsymbol{v}_\perp||^2 + ||\boldsymbol{u}_\perp||^2 = \frac{1}{\sin^2(\theta)} \left(\frac{1}{||\boldsymbol{u}||^2} + \frac{1}{||\boldsymbol{v}||^2} \right) \tag{20}$$

This has two interpretable properties. First, the larger the encoding the smaller the weight cost, and, second, the more aligned the two encodings the larger the weight cost. Both make a lot of sense, the more teasing apart of the representation is needed to extract the variable, the larger the weights, and the higher the loss. One claim we'll use later is that, for a given encoding size $||\boldsymbol{u}||$ and $||\boldsymbol{v}||$, all solutions with $\boldsymbol{u}^T\boldsymbol{v} = 0$ are equally optimal. In particular, this is true of the modular solution:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}' \\ \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{v}' \end{bmatrix} \tag{21}$$

Thus far we've discussed was the output weight cost. The input weight norm is even simpler:

$$||\boldsymbol{W}_{\text{in}}||_F^2 = ||\, [\boldsymbol{u} \quad \boldsymbol{v}]\, ||_F^2 = \text{Tr}[\boldsymbol{W}_{\text{in}}^T \boldsymbol{W}_{\text{in}}] = ||\boldsymbol{u}||^2 + ||\boldsymbol{v}^2|| \tag{22}$$

So, for a fixed encoding size, i.e. fixed $||\boldsymbol{u}||^2$ and $||\boldsymbol{v}||^2$, this loss is actually fixed. It therefore won't effect the optimal alignment of the representation.

Therefore we find, as advertised, that the weight loss is minimised by a modular solution.

B.1.4 ACTIVITY LOSS AND COMBINATION

We now turn to the activity loss and study when it is minimised by modularising. We will play the following game. Let's say you have a non-modular, mixed, representation; i.e. a representation in which at least one neuron has mixed tuning to both x and y:

$$z_n^{[i]} = u_n x^{[i]} + v_n y^{[i]} + \Delta_n \tag{23}$$

Where, $\Delta_n = -\min_i [u_n x^{[i]} + v_n y^{[i]}]$ is the minimal bias required to make the representation non-negative. We will find the conditions under which, depending on the mixing coefficients (u_n and v_n), you decrease the loss by forming the modular solution:

$$\boldsymbol{z}_n^{[i]} = \begin{bmatrix} u_n x^{[i]} \\ v_n y^{[i]} \end{bmatrix} - \begin{bmatrix} |u_n| \min_i x^{[i]} \\ |v_n| \min_i y^{[i]} \end{bmatrix} = \begin{bmatrix} u_n x^{[i]} \\ v_n y^{[i]} \end{bmatrix} + \begin{bmatrix} |u_n| b_x \\ |v_n| b_y \end{bmatrix} \tag{24}$$

Where $b_x = -\min_i x^{[i]}$ is the bias required for a modular encoding of $x^{[i]}$. If, for a given $x^{[i]}$ and $y^{[i]}$, it is true that modularising decreases the loss for all mixings (u_n and v_n) then the modular representation has the optimal energy loss. If there are conditions when modularising increases the loss, then you can always usefully demodularise the representation to decrease the activity loss, and we'll find that the optimal solution is not perfectly modular.

Let's analyse the activity loss of these two representations, for the modular representation (eq: 24):

$$\mathscr{L}_G^M = u_n^2(\langle (x^{[i]})^2 \rangle_i + b_x^2)) + v_n^2(\langle (y^{[i]})^2 \rangle + b_y^2) \tag{25}$$

And compare it to the mixed (De-modularised) solution:

$$\mathscr{L}_G^D = u_n^2 \langle (x^{[i]})^2 \rangle_i + v_n^2 \langle (y^{[i]})^2 \rangle_i + \Delta_n^2 \tag{26}$$

Mixing is prefered over modularity, for a given u_i and v_i, when:

$$(\min_i [u_n x^{[i]} + v_n y^{[i]}])^2 = \Delta_n^2 < u_n^2 b_x^2 + v_n^2 b_y^2 = u_n^2 \min_i (x^{[i]})^2 + v_n^2 \min_i (y^{[i]})^2 \tag{27}$$

Let's get some intuition for what this is saying. Take the simple setting when $v_n \approx 0$, i.e. we're considering a mixed neuron that mainly codes for x, with a little bit of y. Then two equivalent modular neurons are worse than this hypothetical mixed neuron if, to second order in v_n, $\Delta_n < |u_n| b_x + \frac{v_n^2 b_y^2}{|u_n| b_x}$. Since Δ_n is the mixed bias, this means that mixing a small amount of y into the representation of x is preferable when doing so effectively does not increase the required bias, relative to the bias for only encoding x ($|u_n| b_x$). For this to be true $u_n x^{[i]}$ must have the same minima as $u_n x^{[i]} + v_n y^{[i]}$. For small enough v_n, an equivalent phrasing of this condition is that at the times at which $u_n x^{[i]}$ takes its minimal value, $y^{[i]}$ is non-negative.

We can visualise this conditions by plotting the allowed range of $x^{[i]}$ and $y^{[i]}$, Figure 7. In grey is the bounding box given by the minima and maxima of x and y. Assume for now that u_n and v_n are positive. For a fixed small value of v_n the minima of $u_n x^{[i]} + v_n y^{[i]}$ remains $u_n \min_i x^{[i]}$ if there are

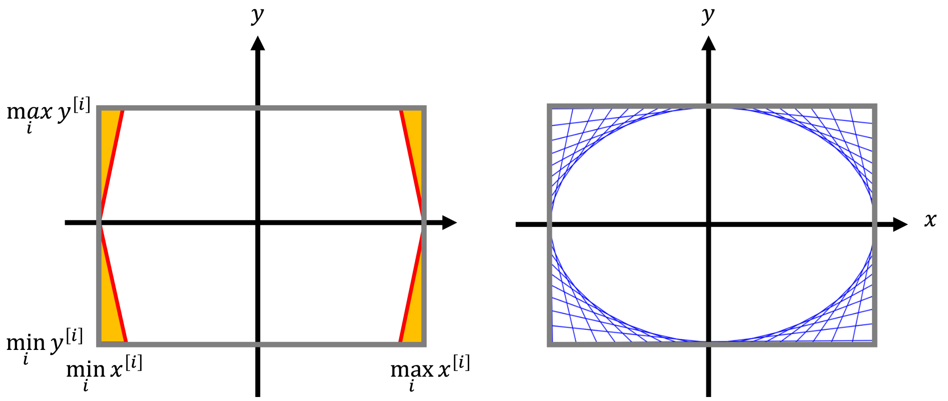


Figure 7: Visualising constraints via corner-cutting. Left: The optimal representation will prefer a mixture containing a small amount of y over modularised if any of the orange shaded regions are missing from the co-range of x and y. Right: Drawing these lines for all mixings creates a set of infinitely many lines, each corresponds to the data that must be missing for a particular mixed neuron to be preferable to the equivalent modular solution.

no datapoints below the line $u_n \min_i x^{[i]} = u_n x + v_n y$. Drawing this onto the square, this means there are no datapoints below the red line, in the bottom-left orange region. If this condition is met the representation should mix because this mixed neuron (given by the particular u_n and v_n ratio) is better than an equivalent modular solution.

That was for positive u_n and v_n. If either of them are negative then you draw a different line corresponding to that sign combination, e.g.: $u_n \min_i x^{[i]} = |u_n|x - |v_n|y$. This produces the top-left red line and orange area, again, if there are no datapoints in this shaded region a mixed neuron with mixture coefficients, $|u_n|$ and $-|v_n|$ is better than the equivalent modular solution. Repeating this for the other sign combinations fills in the other four corners. Hence, we find that a mixed solution in which a small amount of y is mixed into a neuron encoding x is better when there are no datapoints in any of the four corner regions.

That was for one particular pair of u_n and v_n, this hypothetical mixed neuron that encodes mainly x with a pinch of y. But to be optimal the modular solution has to be better than all possible mixed neurons. As such we get a family of such constraints: for each mixing of the sources we get a different inequality on the required bias, as described by equation 27. Each of these can be similarly visualised as a corner missing from the co-range of x and y, the blue lines in Figure 7 right. If any one of these corners is missing from the co-range of (x, y) then the mixed solution is better, else the modular solution is preferred. We can already see the shape of the ellipse emerging that will be the basis of our later version of the theorem, see Appendix B.5.

B.1.5 COMBINING TWO ANALYSES

We studied the two loss terms separately, how do they interact?

If the inequality conditions, (27), are satisfied, then the activity loss prefers a modular solution over all mixed representations with the same fixed encodings sizes, $||\boldsymbol{v}||$ and $||\boldsymbol{u}||$. Further, the weight loss is also minimised, for a fixed encoding size, by modularising. Therefore the optimal solution will be modular.

Conversely, if one of the inequalities is broken then the activity loss is not minimised at the modular solution. We can move towards the optimal mixed solution by creating a new mixed encoding neuron with infinitesimally low activity, and mixed in proportions given by the broken inequality. This neuron can be created by taking small pieces of the modular encoding of each source, in proportion to the mixing. Doing this will decrease the activity loss. Further, the weight loss is minimised at the modular solution, so, to first order, moving away from modular solution doesn't change it. Therefore, for each modular representation, we have found a mixed solution that is better, so the optimal solution is not modular. Therefore, if one of the inequalities is broken, the optimal representation will be mixed.

B.1.6 RELAXING ASSUMPTIONS

We now show how the modularisation behaviour of variables that break the two assumptions listed in B.1.2 can be similarly understood.

Correlated Variables Correlated variables are easy to deal with, they simply introduce an extra term into the difference in activity losses, equation 26 minus equation 25:

$$\mathscr{L}_G^{\text{Modular}} - \mathscr{L}_G^{\text{Mixed}} = u_n^2 b_x^2 + v_n^2 b_y^2 - 2u_n v_n \langle x^{[i]} y^{[i]} \rangle_i - \Delta_n^2 \tag{28}$$

This leads to a slightly updated inequality, the solution mixes if:

$$\Delta_n < \sqrt{u_n^2 b_x^2 + v_n^2 b_y^2 - 2v_n u_n \langle x^{[i]} y^{[i]} \rangle_i} \tag{29}$$

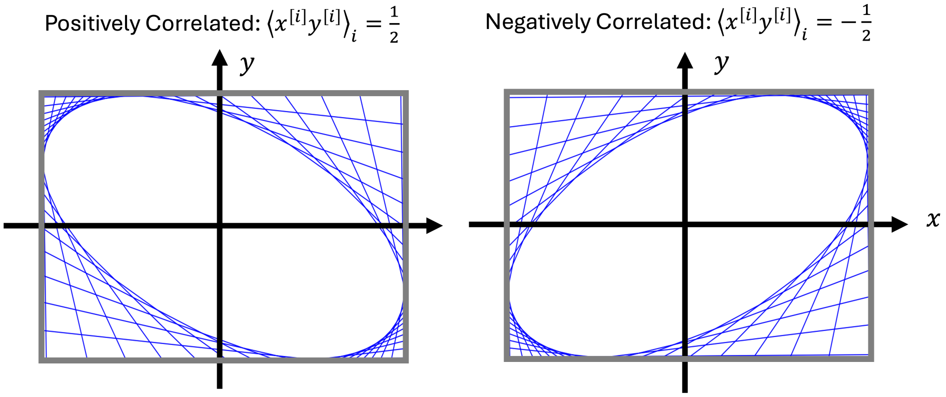

Positively Correlated: $\langle x^{[i]} y^{[i]} \rangle_i = \frac{1}{2}$
Negatively Correlated: $\langle x^{[i]} y^{[i]} \rangle_i = -\frac{1}{2}$
y
y
x

Figure 8: Visualising how correlations change corner-cutting. Left: It is energetically costly to align the encodings of positively correlated variables

The effect of this change is to shift the corner-cutting lines that determine whether a modular of mixed solution is optimal. This can be understood intuitively. If the variables are positively correlated then positively aligning their representations is energetically costly, and negatively aligning them is energy saving. This is most simply seen in one line of maths. The energy cost contains the following term:

$$\langle (\boldsymbol{u}x^{[i]} + \boldsymbol{v}y^{[i]})^2 \rangle_i = ||\boldsymbol{u}||^2 \langle (x^{[i]})^2 \rangle_i + ||\boldsymbol{v}||^2 \langle (y^{[i]})^2 \rangle_i + \boldsymbol{u}^T \boldsymbol{v} \langle x^{[i]} y^{[i]} \rangle_i \tag{30}$$

So if the correlation is positive, $\boldsymbol{u}^T\boldsymbol{v}$ should be negative to decrease the energy, i.e. the encodings should be anti-aligned. Hence, correlated variables have a preferred orientation.

Similarly, the corner cutting has an associated 'direction of alignment'. If the bottom left corner is missing that means that building a neuron whose representation is a mix of positively aligned x and y is preferable, as it has a smaller minima, requiring less bias. Similarly if the bottom right corner is missing the representation can take advantage of that by building neurons that encode x negatively and y positively, i.e. anti-aligning them.

If these two directions, one from the correlations and one from the range-dependent positivity effects, agree, then it is 'easier' to de-modularise, i.e. less of a corner needs to be missing from the range to de-modularise. Conversely, if they mis-align it is harder. This is why the correlations shift the corner-cutting lines up or down depending on the corner. We can see this in figure Figure 8, when the variables are positively correlated it becomes 'easier' to mix in the anti-aligned corners, at bottom-right or top-left, and harder at the aligned pair, while the converse is true of the negatively correlated example.

Of interest is a result we can find that guarantees modularisation regardless of the correlation. For these bounded variables the covariances, $\langle x^{[i]} y^{[i]} \rangle_i$, are themselves bounded. The most postively co-varying two variables can be is to be equal, then the covariance becomes the variance. Further, the bounded, symmetric, mean-zero variable with the highest variance spends half its time at its max, and half at its min. Similar arguments but with a minus sign for the most anti-covarying representations tell us that:

$$-b_y b_x \leq \langle x^{[i]} y^{[i]} \rangle_i \leq b_x b_y \tag{31}$$

These results are similar to those in Hössjer and Sjölander (2022). Putting this into (29), the maximal bounding bias for the most positively or negatively correlated variables to prefer modularising:

$$\Delta_n > |u_n|b_x + |v_n|b_y \tag{32}$$

This bound is the development of equation 29 in two cases. First in the maximally positively correlated case when the sign of u_i and v_i are opposite. Second in the maximally negatively correlated case where the coefficient signs are the same.

From this we can derive the following interesting result, the pedagogical version of Theorem B.1. We will say that if, across the dataset, when $x^{[i]}$ takes its maximum (or minimum) value, $y^{[i]}$ takes both its maximum and minimum values for different datapoints, the two variables are extreme point independent. Another way of saying this is that there is non-zero probability of each of the corners of the rectangle. Now, no matter the value of the correlation, extreme point independence variables should modularise. This is because:

$$\Delta_n = -\min_i \left[u_n x^{[i]} + v_n y^{[i]} \right] = -|u_n| \min_i x^{[i]} - |v_n| \min_i y^{[i]} = |u_n|b_x + |v_n|b_y \tag{33}$$

Hence, we find that even the loosest inequality possible, eqaution 32, is never satisfied. Further, since the variables are extreme point independent their covariance never achieves the equalities in equation 31, so inequality 32 is not satisfied and the variables should modularise.

This means that symmetric extreme point independent variables should always modularise, regardless of how correlated they are. Conversely, for variables that are not quite extreme point independent (perhaps they are missing some corner), there is always a value of their correlation that would de-modularise them.

Range Asymmetric Variables Thus far we have assumed $|\min_i x^{[i]}| = \max_i x^{[i]}$, for simplicity. This is not necessary. If this is not the case than there is a 'direction' the variable wants to be encoded. Consider encoding a single variable either positively ($z_+^{[i]} = |u|x^{[i]} + b$) or negatively ($z_-(t) = -|u|x^{[i]} + b$), the losses will differ:

$$\langle (z_+^{[i]})^2 \rangle_i = |u|^2 \big(\langle (x^{[i]})^2 \rangle_i + (\min_i x^{[i]})^2 \big) \qquad \langle (z_-^{[i]})^2 \rangle_i = |u|^2 \big(\langle (x^{[i]})^2 \rangle_i + (\max_i x^{[i]})^2 \big) \tag{34}$$

This is simply expressing the fact that if a variable is mean-zero but asymmetrically bounded then more of the data must be bunched on the smaller side, and you should choose the encoding where most of this data stays close to the origin.

Now, assume without loss of generality that for each variable $b_x = |\min_t x(t)| \leq \max_t x(t)$. Then the identical argument to before carries through for the modularisation behaviour around the lower left corner. Around the other corners however, things are more complicated. There might be a corner missing from the lower right quadrant, but to exploit it you would have to build a mixed neuron that is positively tuned to y but negatively tuned to x. This would incur a cost, as suddenly the x variable is oriented in its non-optimal direction. Therefore the calculation changes, because the modular solution can always choose to 'correctly' orient variables, so doesn't have to pay this cost. This is all expressed in the same inequalities, eqn. 29:

$$\Delta_n < \sqrt{u_n^2 b_x^2 + v_n^2 b_y^2 - 2 v_n u_n \langle x^{[i]} y^{[i]} \rangle_i} \tag{35}$$

This effect can also be viewed in the required range plots, where it translates to a much larger corner being cut off in order to pay the cost of switching around the variables.

Remaining Assumptions There are two remaining assumptions about the variables, that they are mean-zero and that they are bounded. Neither of these are really assumptions. The first since we don't penalise biases in the weight losses you can add or remove a constant from the variables for no cost. That means adding or removing the mean from the variables doesn't change the problem, so our results generalise to non-mean-zero variables, but all the conditions apply to the mean-subtracted versions of the variables. The second boundedness constraint appears more fundamental, but for any finite dataset the variables will have a maximum and minimum value. These values will be the important ones that determine modularisation.

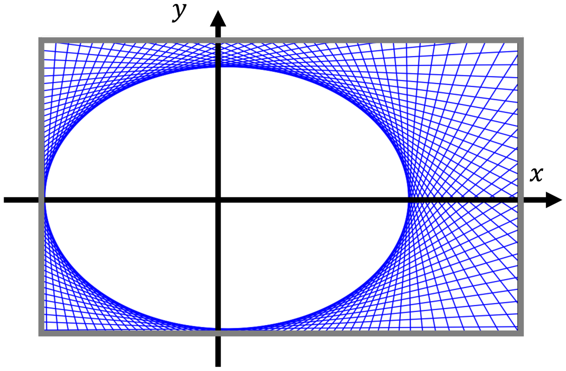


Figure 9: Visualising how asymmetric ranges change corner-cutting. When the ranges are asymmetric much larger corners must be cut off the 'longer' sides before mixing becomes preferable to modularising.

B.2 FULL THEORETICAL TREATMENT

Now we shall prove our main result in generality.

Theorem B.1 (Main Result)**.** *Let $\boldsymbol{s} \in \mathbb{R}^{d_s}$, $\boldsymbol{z} \in \mathbb{R}^{d_z}$, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{d_z \times d_s}$, $\boldsymbol{b}_{\text{in}} \in \mathbb{R}^{d_z}$, $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d_s \times d_z}$, and $\boldsymbol{b}_{\text{out}} \in \mathbb{R}^{d_s}$, with $d_z >> d_s$. Consider the constrained optimization problem*

$$\begin{aligned} \min_{\boldsymbol{W}_{\text{in}}, \boldsymbol{b}_{\text{in}}, \boldsymbol{W}_{\text{out}}, \boldsymbol{b}_{\text{out}}} \quad & \left\langle ||\boldsymbol{z}^{[i]}||_2^2 \right\rangle_i + \lambda \left(||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2 \right) \\ \text{s.t.} \quad & \boldsymbol{z}^{[i]} = \boldsymbol{W}_{\text{in}} \boldsymbol{s}^{[i]} + \boldsymbol{b}_{\text{in}}, \ \boldsymbol{s}^{[i]} = \boldsymbol{W}_{\text{out}} \boldsymbol{z}^{[i]} + \boldsymbol{b}_{\text{out}}, \ \boldsymbol{z}^{[i]} \geq 0, \end{aligned} \tag{36}$$

where i indexes samples of $\boldsymbol{s}$. At the minima of this problem, each row of $\boldsymbol{W}_{\text{in}}$ has at most one non-zero entry, i.e. the representation modularises, iff the following inequality is satisfied for all $\boldsymbol{w} \in \mathbb{R}^{d_s}$:

$$\left(\min_i [\boldsymbol{w}^\top \boldsymbol{s}^{[i]}] \right)^2 + \sum_{j, j' \neq j}^{d_s} w_j w_{j'} \left\langle \bar{s}_j^{[i]} \bar{s}_{j'}^{[i]} \right\rangle_i > \sum_{j=1}^{d_s} \left(w_j \min_i \bar{s}_j^{[i]} \right)^2, \tag{37}$$

where $\bar{\boldsymbol{s}} := \boldsymbol{s} - \left\langle \boldsymbol{s}^{[i]} \right\rangle_i$ and assuming that $\left| \min_i \bar{s}_j^{[i]} \right| \leq \max_i \bar{s}_j^{[i]} \ \forall j \in [d_s]$ w.l.o.g.

To show this statement we begin by showing a lemma: that, for a fixed encoding size, the weight loss is minimised by orthogonalising the encoding of each source, the multidimensional analogue of Appendix B.1.3. Writing the representation as an affine function of the de-meaned sources:

$$\boldsymbol{z}^{[i]} = \sum_{j=1}^{d_s} \boldsymbol{u}_j s_j^{[i]} + \boldsymbol{b} = \sum_{j=1}^{d_s} \boldsymbol{u}_j (s_j^{[i]} - \langle s_j^{[k]} \rangle_k) + \boldsymbol{b}' = \sum_{j=1}^{d_s} \boldsymbol{u}_j \bar{s}_j^{[i]} + \boldsymbol{b}' \tag{38}$$

then by fixed encoding size we mean that each $|\boldsymbol{u}_j|$ is fixed. We express things in terms of mean-zero variables $\bar{s}_j^{[i]}$ since it is easier and we can freely shift by a constant offset.

Theorem B.2 (Modularising Minimises Weight Loss)**.** *The weight loss, $||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2$, is minimised, for fixed encoding magnitudes, $|\boldsymbol{u}_j|$, by modularising the representation.*

Proof. First, we know that $\boldsymbol{W}_{\text{in}}$ must be of the following form:

$$\boldsymbol{W}_{\text{in}} = [\boldsymbol{u}_1 \quad \dots \quad \boldsymbol{u}_{d_s}] \tag{39}$$

And therefore for a fixed encoding size the input weight loss is constant:

$$||\boldsymbol{W}_{\text{in}}||_F^2 = \text{Tr}[\boldsymbol{W}_{\text{in}}^T \boldsymbol{W}_{\text{in}}] = \sum_{j=1}^{d_s} |\boldsymbol{u}_j|^2 \tag{40}$$

So let's study the output weights, these are defined by the way they map the representation back to the sources:

$$\boldsymbol{W}_{\text{out}} \cdot \boldsymbol{z}^{[i]} + \boldsymbol{b}_{\text{out}} = \begin{bmatrix} s_1^{[i]} \\ \vdots \\ s_{d_s}^{[i]} \end{bmatrix} \tag{41}$$

The min-norm $\boldsymbol{W}_{\text{out}}$ with this property is the Moore-Penrose Pseudoinverse, i.e. the matrix:

$$\boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{U}_1^T \\ \vdots \\ \boldsymbol{U}_{d_s}^T \end{bmatrix} \tag{42}$$

Where each pseudoinverse $\boldsymbol{U}_i$ is defined by $\boldsymbol{U}_i^T \boldsymbol{u}_j = \delta_{ij}$ and lying completely within the span of the encoding vectors $\{\boldsymbol{u}_j\}$. We can calculate the norm of this matrix:

$$|\boldsymbol{W}_{\text{out}}|_F^2 = \text{Tr}[\boldsymbol{W}_{\text{out}} \boldsymbol{W}_{\text{out}}^T] = \sum_{j=1}^{d_s} |\boldsymbol{U}_j|^2 \tag{43}$$

Now, each of these capitalised pseudoinverse vectors must have some component along its corresponding lower case vector, and some component orthogonal to that:

$$\boldsymbol{U}_j = \frac{1}{|\boldsymbol{u}_j|^2} \boldsymbol{u}_j + \boldsymbol{u}_{j,\perp} \tag{44}$$

We've chosen the size of the component along $\boldsymbol{u}_j$ such that $\boldsymbol{U}_j^T \boldsymbol{u}_j = 1$, and the $\boldsymbol{u}_{j,\perp}$ is chosen so that $\boldsymbol{U}_j^T \boldsymbol{u}_k = \delta_{jk}$. Now, for a fixed size of $|\boldsymbol{u}_j|$, this sets a lower bound on the size of the weight matrix:

$$|\boldsymbol{W}_{\text{out}}|_F^2 = \sum_{j=1}^{d_s} \frac{1}{|\boldsymbol{u}_j|^2} + |\boldsymbol{u}_{j,\perp}|^2 \geq \sum_{j=1}^{d_s} \frac{1}{|\boldsymbol{u}_j|^2} \tag{45}$$

And this lower bound is achieved whenever the $\{\boldsymbol{u}_j\}_{j=1}^{d_s}$ vectors are orthogonal to one another, since then $\boldsymbol{u}_{j,\perp} = 0$. Therefore, we see that, for a fixed size of encoding, the weight loss is minimised when the encoding vectors are orthogonal, and that is achieved when the code is modular. ☐

Now we will turn to studying the activity loss, and find conditions under which a modular representation is better than a mixed one. We compare two representations, a mixed neuron:

$$z_n^{[i]} = \sum_{j=1}^{d_s} u_{nj} \bar{s}_j^{[i]} + \Delta_n = \sum_{j=1}^{d_s} u_{nj} \bar{s}_j^{[i]} - \min_i [\sum_{j=1}^{d_s} u_{nj} \bar{s}_j^{[i]}] \tag{46}$$

And another representation in which we break this neuron apart into its modular form, preserving the encoding size of each source:

$$\boldsymbol{z}_n^{[i]} = \begin{bmatrix} u_{n1} \bar{s}_1^{[i]} \\ \vdots \\ u_{nd_s} \bar{s}_{d_s}^{[i]} \end{bmatrix} - \begin{bmatrix} |u_{n1}| \min_j \bar{s}_1^{[j]} \\ \vdots \\ |u_{nd_s}| \min_j \bar{s}_{d_s}^{[i]} \end{bmatrix} \tag{47}$$

The activity loss difference between these two representations is:

$$\mathscr{L}_G^{\text{Modular}} - \mathscr{L}_G^{\text{Mixed}} = \sum_j u_{nj}^2 (\min_i \bar{s}_j^{[i]})^2 - (\min_i [\sum_{j=1}^{d_s} u_{nj} \bar{s}_j^{[i]}])^2 - \sum_{j=1, k \neq j}^{d_s} u_{nj} u_{nk} \langle \bar{s}_j^{[i]} \bar{s}_k^{[i]} \rangle_i \tag{48}$$

So the modular loss is smaller if, for all $\boldsymbol{u}_n \in \mathbb{R}^{d_s}$:

$$(\min_i [\sum_{j=1}^{d_s} u_{nj} \bar{s}_j^{[i]}])^2 > \sum_j u_{nj}^2 (\min_i \bar{s}_j^{[i]})^2 - \sum_{j, k \neq j} u_{nj} u_{nk} \langle \bar{s}_j^{[i]} \bar{s}_k^{[i]} \rangle_i \tag{49}$$

Further, we can see that if this equation holds for one vector, $\boldsymbol{u}$, it also holds for any scaled version of that vector, therefore we can equivalently require the inequality to be true only for unit norm vectors.

So, if these inequalities are satisfied, both the weight and activity loss are minimised at fixed encoding size by modularising, and this holds for all encoding sizes, therefore the optimal solution must be modular. Hence these conditions are sufficient for modular solutions to be optimal.

Conversely, if one of the inequalities is broken then the activity loss is not minimised at the modular solution. We can move towards the optimal mixed solution by creating a new mixed encoding neuron with infinitesimally low activity, and mixed in proportions given by the broken inequality. This neuron can be created by taking small pieces of the modular encoding of each source, in proportion to the mixing. Doing this will decrease the activity loss. Further, the weight loss is minimised at the modular solution, so, to first order, moving away from modular solution doesn't change it. Therefore, for each modular representation, we have found a mixed solution that is better, so the optimal solution is not modular.

For intuition on how corner-cutting relates to the above inequalities see the pedagogical derivation in section Appendix B.1.

B.3 RANGE-INDEPENDENT VARIABLES

We will now prove this corollary, that was pedagogically discussed in Appendix B.1:

Corollary B.3. *In the same setting as Theorem 2.1 the optimal representation modularises if all sources are pairwise extreme-point independent, i.e. if*

$$\min_i \left[s_j^{[i]} \middle| s_{j'}^{[i]} \in \left\{ \max_{i'} s_{j'}^{[i']}, \min_{i'} s_{j'}^{[i']} \right\} \right] = \min_i s_j^{[i]} \tag{50}$$

for all $j, j' \in [d_s]^2$.

Proof. If the variables are extreme-point independent then we can take the min inside the sum in the definition of the mixed neuron bias:

$$z_n^{[i]} = \sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]} + \Delta_n = \sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]} - \min_i [\sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]}] = \sum_{j=1}^{d_s} \left(u_{nj}\bar{s}_j^{[i]} - \min_i [u_{nj}\bar{s}_j^{[i]}] \right) \tag{51}$$

Now we can compute the activity loss:

$$\langle (z_n^{[i]})^2 \rangle_i = \sum_{j=1}^{d_s} \langle \left(u_{nj}\bar{s}_j^{[i]} - \min_i [u_{nj}\bar{s}_j^{[i]}] \right)^2 \rangle_i + \sum_{j,j'=1, j\neq j}^{d_s} \langle \left(u_{nj}\bar{s}_j^{[i]} - \min_i [u_{nj}\bar{s}_j^{[i]}] \right) \left(u_{nj}\bar{s}_{j'}^{[i]} - \min_i [u_{nj}\bar{s}_{j'}^{[i]}] \right) \rangle_i \tag{52}$$

The cross terms at the end are strictly positive, and each of the squared terms on the left is the same or larger than its corresponding modular neuron, since either u_{nj} is positive and it is the same, or u_{nj} is negative and it is greater than or equal to the equivalent modular neuron. Therefore the mixed energy cost is larger than the modular energy cost for extreme-point independent variables. ☐

B.4 PERTURBATIVE APPROACHES TO MODULARISATION OF LINEARLY MIXED SOURCES

Now we consider a more general setting than that stated in the main paper. We allow the input data, $\boldsymbol{x}^{[i]}$, to be a linear mixture of the sources: $\boldsymbol{x}^{[i]} = \boldsymbol{A}\boldsymbol{s}^{[i]}$. We then similarly ask when should the optimal nonnegative, linear, efficient, representation be modular?

Theorem B.4. *Let* $\boldsymbol{s} \in \mathbb{R}^{d_s}$, $\boldsymbol{x} \in \mathbb{R}^{d_x}$, $\boldsymbol{z} \in \mathbb{R}^{d_z}$, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{d_z \times d_x}$, $\boldsymbol{b}_{\text{in}} \in \mathbb{R}^{d_x}$, $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d_x \times d_z}$, $\boldsymbol{b}_{\text{out}} \in \mathbb{R}^{d_x}$, *and* $\boldsymbol{A} \in \mathbb{R}^{d_x \times d_s}$ *with* $d_z > d_s$ *and* $d_x \geq d_s$, *and* $\boldsymbol{A}$ *is rank* d_s. *Consider the constrained optimization problem*

$$\begin{aligned} \min_{\boldsymbol{W}_{\text{in}}, \boldsymbol{b}_{\text{in}}, \boldsymbol{W}_{\text{out}}, \boldsymbol{b}_{\text{out}}} \quad & \left\langle ||\boldsymbol{z}^{[i]}||_2^2 \right\rangle_i + \lambda \left(||\boldsymbol{W}_{\text{in}}||_F^2 + ||\boldsymbol{W}_{\text{out}}||_F^2 \right) \\ \text{s.t.} \quad & \boldsymbol{z}^{[i]} = \boldsymbol{W}_{\text{in}}\boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{in}}, \ \boldsymbol{x}^{[i]} = \boldsymbol{W}_{\text{out}}\boldsymbol{z}^{[i]} + \boldsymbol{b}_{\text{out}}, \ \boldsymbol{z}^{[i]} \geq 0, \end{aligned} \tag{53}$$

where i indexes a finite set of samples of $\boldsymbol{s}$, and $\boldsymbol{x}^{[i]} = \boldsymbol{A}\boldsymbol{s}^{[i]}$. There is a modular representation that is a local minima of this problem, i.e. one in which each latent, $z_j^{[i]}$, is a function of a single source, iff the following inequality is satisfied for all $\boldsymbol{w} \in \mathbb{R}^{d_s}$:

$$\langle (\boldsymbol{w}^T \boldsymbol{s}^{[i]})^2 \rangle_i + (\min_i \boldsymbol{w}^T \boldsymbol{s}^{[i]})^2 + \lambda \boldsymbol{w}^T (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{w} > \lambda \boldsymbol{w}^T \boldsymbol{D}^{-2} (\boldsymbol{A}^T \boldsymbol{A}) \boldsymbol{D}^{-2} \boldsymbol{w} \tag{54}$$

where $\bar{\boldsymbol{s}} := \boldsymbol{s} - \left\langle \boldsymbol{s}^{[i]} \right\rangle_i$, assuming that $\left| \min_i \bar{s}_j^{[i]} \right| \leq \max_i \bar{s}_j^{[i]} \; \forall j \in [d_s]$ w.l.o.g., and where $\boldsymbol{D}$ is a diagonal matrix with diagonal elements:

$$D_{jj} = \sqrt[4]{\frac{\lambda((\boldsymbol{A}^T\boldsymbol{A})_{jj})^2}{\langle (s_j^{[i]})^2 \rangle_i + (\min_i s_j^{[i]})^2 + \lambda(((\boldsymbol{A}^T\boldsymbol{A})^{-1})_{jj})^2}} \tag{55}$$

Using the method as for the original statement, this too can be turned into an visually easier claim about the convex hull and bounding a particular set:

Theorem B.5. *In the same setting as Theorem B.4 form the matrix $\boldsymbol{S}$:*

$$\boldsymbol{S} = \lambda \boldsymbol{D}^{-2} (\boldsymbol{A}^T \boldsymbol{A}) \boldsymbol{D}^{-2} - \lambda (\boldsymbol{A}^T \boldsymbol{A})^{-1} - \langle \boldsymbol{s}^{[i]} (\boldsymbol{s}^{[i]})^T \rangle_i \tag{56}$$

then an equivalent condition to (54) is that the convex hull of the datapoints encloses the set $E = \{\boldsymbol{y} : \boldsymbol{y}^T \boldsymbol{S}^{-1} \boldsymbol{y} = 1\}$.

We cannot attack this with the same techniques as before, as, unless the columns of $\boldsymbol{A}$ are orthogonal, the weight loss is no longer minimised by modularising. We therefore take a different approach, we study the linear stability of the optimal modular solution and show that iff the inequalities in theorem Theorem B.4 are satisfied the optimal modular solution is a local minima. Notice, however, that this approach does not guarantee that the modular solution is the global optima. That said, we empirically verify that this inequality governs when the optimal solution modularises in the cases we have tested, such as in Figure 10, suggesting that this local optimal is in fact global. Future work could usefully clarify whether this is indeed true or there are counterexamples.

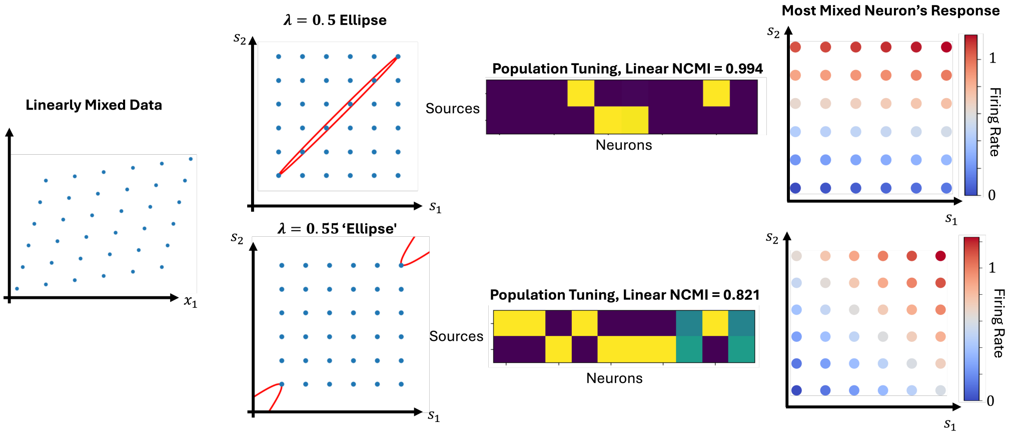


Figure 10: We linear mix range-independent data through some matrix A and consider the optimisation problem for two different values of λ. In one case (top row) the matrix $\boldsymbol{S}$ is positive definite, meaning E is an ellipse, and the convex hull encloses it. Thus, theory predicts that the representation should modularise and the empirics match this. Yet, (bottom row), increasing λ by a small amount makes the matrix $\boldsymbol{S}$ indefinite, and E becomes a parabola that no finite dataset can enclose in its convex hull. Thus we get a mixed optimal representation.

B.4.1 OPTIMAL MODULAR SOLUTION

First we need to find the optimal modular solution. This can be implemented by d_S neurons (since the loss is rotationally invariant adding more neurons makes no difference). Let's write this solution as:

$$\boldsymbol{g}_M^{[i]} = \sum_{j=1}^{d_S} d_j (s_j^{[i]} - \min_i s_j^{[i]}) \boldsymbol{e}_j = \sum_{j=1}^{d_S} d_j \bar{s}_j^{[i]} \boldsymbol{e}_j \tag{57}$$

Where $\boldsymbol{e}_j$ are the canonical basis vectors, d_j represents the strength of the encoding of each source, and the subscript M denotes the fact it is modular. Stacking the d_j elements into a diagonal matrix $\boldsymbol{D}$ we can derive each of the three loss terms. We know that:

$$\boldsymbol{g}^{[i]} = \boldsymbol{W}_{\text{in}}\boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{in}} = \boldsymbol{W}_{\text{in}}\boldsymbol{A}\boldsymbol{s}^{[i]} + \boldsymbol{b}_{\text{in}} = \boldsymbol{D}\boldsymbol{s}^{[i]} + \boldsymbol{b}_{\text{in}} \tag{58}$$

Hence:

$$\boldsymbol{W}_{\text{in}} = \boldsymbol{D}\boldsymbol{A}^{\dagger} \qquad \boldsymbol{W}_{\text{out}} = \boldsymbol{W}_{\text{in}}^{\dagger} = \boldsymbol{A}\boldsymbol{D}^{-1} \tag{59}$$

where † denotes the pseudoinverse. Hence:

$$||\boldsymbol{W}_{\text{in}}||_F^2 = \text{Tr}[\boldsymbol{W}_{\text{in}}^T\boldsymbol{W}_{\text{in}}] = \text{Tr}[\boldsymbol{D}^2\boldsymbol{A}^{\dagger}(\boldsymbol{A}^{\dagger})^T] = \text{Tr}[\boldsymbol{D}^2\boldsymbol{O}_A^{-1}] \tag{60}$$

$$||\boldsymbol{W}_{\text{out}}||_F^2 = \text{Tr}[\boldsymbol{D}^{-2}\boldsymbol{O}_A] \tag{61}$$

Where we've defined $\boldsymbol{O}_A = \boldsymbol{A}^T\boldsymbol{A}$, and then $\boldsymbol{A}^{\dagger}(\boldsymbol{A}^{\dagger})^T = \boldsymbol{O}_A^{-1}$, as can be seen using the SVD of $\boldsymbol{A}$ and $\boldsymbol{A}^{\dagger}$.

Denote with α_j^2 the diagonal elements of $\boldsymbol{O}_A$, they represent how strongly a particular source is encoded in the data. Further, denote with $(\alpha'_j)^2$ the diagonal elements of $\boldsymbol{O}_A^{-1}$, these represent the lengths of the 'pseudoinverse vectors' in Figure 6. If the encodings of each source are orthogonal they are just the inverse of the encoding sizes, α_j, if not, they also tell you how 'entangled' a source's representation is. In the limit of colinearity of sources, α'_j go to infinity representing the fact that the modular solution cannot exist. In terms of these variables we can write the whole loss as:

$$\mathscr{L} = \sum_{j=1}^{d_s} d_j^2[\langle(s_j^{[i]})^2\rangle_i + (\min_i s_j^{[i]})^2 + \lambda(\alpha'_j)^2] + \frac{\lambda\alpha_j^2}{d_j^2} \tag{62}$$

Then the optimal choice of encoding sizes is:

$$(d_j^*)^4 = \frac{\lambda\alpha_j^2}{\langle(s_j^{[i]})^2\rangle_i + (\min_i s_j^{[i]})^2 + \lambda(\alpha'_j)^2} \tag{63}$$

B.4.2 PERTURBING THE OPTIMAL MODULAR SOLUTION

Now let's add the following perturbation to the optimal representation:

$$\boldsymbol{g}^{[i]} = \sum_{j=1}^{d_s}(d_j^* - \delta_j)(s_j^{[i]} - \min_i s_j^{[i]})\boldsymbol{e}_j + \boldsymbol{e}_{d_s+1}(\boldsymbol{w}^T\boldsymbol{s}^{[i]} - \min_i \boldsymbol{w}^T\boldsymbol{s}^{[i]}) \tag{64}$$

This represents an arbitrary perturbation to the modular solution combined with adding a new mixed neuron with mixing $\boldsymbol{w}$. We could add more mixed neurons, but to the order we consider they contribute independently, allowing you to consider them one-by-one. We will assume all the δ_j and $\boldsymbol{w}$ are small, and expand the loss to second order in them. If this perturbation decreases the loss for any $\boldsymbol{w}$ then the optimal representation is mixed, else the optimal modular solution is a local minima. First let's create some new matrices:

$$\tilde{\boldsymbol{D}} = \begin{bmatrix} d_1^* - \delta_1 & 0 & \dots & 0 \\ 0 & d_2^* - \delta_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_{d_s}^* - \delta_{d_S} \\ 0 & 0 & \dots & 0 \end{bmatrix} \qquad \boldsymbol{E} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ w_1 & w_2 & \dots & w_{d_s} \end{bmatrix} \tag{65}$$

Then I claim the min-norm $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_{\text{out}}$ are given by:

$$\boldsymbol{W}_{\text{in}} = (\tilde{\boldsymbol{D}} + \boldsymbol{E})\boldsymbol{A}^{\dagger} \qquad \boldsymbol{W}_{\text{out}} = \boldsymbol{A}(\tilde{\boldsymbol{D}} + \boldsymbol{E})^{\dagger} \tag{66}$$

These are the min-norm solutions, because they are pseudoinverses and they're min-rank, i.e. rank d_s. The form of the losses will again be things like:

$$||\boldsymbol{W}_{\text{in}}||_F^2 = \text{Tr}[(\tilde{\boldsymbol{D}} + \boldsymbol{E})^T(\tilde{\boldsymbol{D}} + \boldsymbol{E})\boldsymbol{O}_A^{-1}] = \text{Tr}[\boldsymbol{O}_D\boldsymbol{O}_A^{-1}] \qquad ||\boldsymbol{W}_{\text{out}}||_F^2 = \text{Tr}[\boldsymbol{O}_D^{-1}\boldsymbol{O}_A] \tag{67}$$

We can see the following:

$$\boldsymbol{O}_D = (\tilde{\boldsymbol{D}} + \boldsymbol{E})^T(\tilde{\boldsymbol{D}} + \boldsymbol{E}) = \tilde{\boldsymbol{D}}^2 + \boldsymbol{w}\boldsymbol{w}^T \tag{68}$$

Then, using Sherman-Morrison:

$$\boldsymbol{O}_D^{-1} = \tilde{\boldsymbol{D}}^{-2} - \frac{\tilde{\boldsymbol{D}}^{-2}\boldsymbol{w}\boldsymbol{w}^T\tilde{\boldsymbol{D}}^{-2}}{1 + \boldsymbol{w}^T\tilde{\boldsymbol{D}}^{-2}\boldsymbol{w}} \tag{69}$$

We're only going to be working to second order, so we can simplify this expression significantly:

$$\boldsymbol{O}_D^{-1} = \tilde{\boldsymbol{D}}^{-2} - \boldsymbol{D}^{-2}\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{D}^{-2} + \mathscr{O}(\delta^3) \tag{70}$$

Now it is a bit of algebra to derive the three losses:

$$||\boldsymbol{W}_{\text{in}}||_F^2 = \text{Tr}[\boldsymbol{D}^2\boldsymbol{O}_A^{-1}] - 2\,\text{Tr}[\text{diag}(d_j^*\delta_j)\boldsymbol{O}_A^{-1}] + \text{Tr}[\text{diag}(\delta_j^2)\boldsymbol{O}_A^{-1}] + \boldsymbol{w}^T\boldsymbol{O}_A^{-1}\boldsymbol{w} \tag{71}$$

$$||\boldsymbol{W}_{\text{out}}||_F^2 = \text{Tr}[\boldsymbol{D}^{-2}\boldsymbol{O}_A] + 2\,\text{Tr}[\text{diag}(\frac{\delta_j}{(d_j^*)^3})\boldsymbol{O}_A] + \text{Tr}[\text{diag}(\frac{\delta_j^2}{(d_j^*)^4})\boldsymbol{O}_A] - \boldsymbol{w}^T\boldsymbol{D}^{-2}\boldsymbol{O}_A\boldsymbol{D}^{-2}\boldsymbol{w} \tag{72}$$

$$\langle||\boldsymbol{g}^{[i]}||^2\rangle_i = \sum_{j=1}^{d_s}(\langle(s_j^{[i]})^2\rangle_i+(\min_i s_j^{[i]})^2)((d_j^*)^2-2d_j^*\delta_j+\delta_j^2)+\langle(\boldsymbol{w}^T\boldsymbol{s}^{[i]})^2\rangle_i+(\min_i \boldsymbol{w}^T\boldsymbol{s}^{[i]})^2 \tag{73}$$

There are first order terms in δ_j, but you can check that at the optimal modular solution these are 0. Then we have to consider the second order terms, there are some proportional to δ_j^2:

$$\sum_{j=1}^{M}\delta_j^2(\lambda(\alpha_j')^2 + \frac{3\lambda\alpha_j^2}{(d_j^*)^4} + \langle(s_j^{[i]})^2\rangle_i + (\min_i s_j^{[i]})^2) \tag{74}$$

But, you will notice that this term is always positive! Therefore, it always increases the loss, so we might as well set $\delta_j = 0$, and move on to the only term that might cause mixing to be optimal, if this term is positive then modular solutions are locally the best!

$$\langle(\boldsymbol{w}^T\boldsymbol{s}^{[i]})^2\rangle_i + (\min_i \boldsymbol{w}^T\boldsymbol{s}^{[i]})^2 + \lambda\boldsymbol{w}^T\boldsymbol{O}_A^{-1}\boldsymbol{w} - \lambda\boldsymbol{w}^T\boldsymbol{D}^{-2}\boldsymbol{O}_A\boldsymbol{D}^{-2}\boldsymbol{w} > 0 \tag{75}$$

This is the linearly mixed equivalent of our inequality, and proves the original statement, Theorem B.4.

B.4.3 RE-DERIVING ORIGINAL RESULT

As a corollary we can rederive our main result, Theorem 2.1. To do this we assume $\boldsymbol{x}^{[i]} = \boldsymbol{s}^{[i]}$. This occurs when $\boldsymbol{A} = I$, $\boldsymbol{O}_A = \boldsymbol{O}_A^{-1} = I$; then:

$$(d_j^*)^4 = \frac{\lambda}{\lambda + \langle(s_j^{[i]})^2\rangle_i + (\min_i s_j^{[i]})^2} \qquad (d_j^*)^{-4} = 1 + \frac{1}{\lambda}(\langle(s_j^{[i]})^2\rangle_i + (\min_i s_j^{[i]})^2) \tag{76}$$

Then you can rewrite (75) as:

$$\begin{gathered}\langle(\boldsymbol{w}^T\boldsymbol{s}^{[i]})^2\rangle_i + (\min_i \boldsymbol{w}^T\boldsymbol{s}^{[i]})^2 + \lambda\boldsymbol{w}^T\boldsymbol{w} - \lambda\boldsymbol{w}^T\boldsymbol{D}^{-4}\boldsymbol{w} = \\ \langle(\boldsymbol{w}^T\boldsymbol{s}^{[i]})^2\rangle_i + (\min_i \boldsymbol{w}^T\boldsymbol{s}^{[i]})^2 - \sum_{j=1}^{M} w_j^2(\langle(s_j^{[i]})^2\rangle_i + (\min_i s_j^{[i]})^2) > 0\end{gathered} \tag{77}$$

Which exactly recovers the previous inequalities!

This shows that, when these inequality conditions are met, the best modular solution is a local optima, but unlike the other arguments, it does not show they are a global optima.

B.5 EQUIVALENCE OF CONVEX HULL FORMULATION

In this section we prove Theorem 2.2, an equivalent formulation of the infinitely many inequalities in Theorem 2.1 in terms of the convex hull, restated below:

Theorem B.6. *In the setting as Theorem 2.1, define the following matrix:*

$$\boldsymbol{S} = \begin{bmatrix}(\min_i s_1^{[i]})^2 & -\langle s_1^{[i]}s_2^{[i]}\rangle_i & \dots \\ -\langle s_1^{[i]}s_2^{[i]}\rangle_i & (\min_i s_2^{[i]})^2 & \dots \\ \vdots & \vdots & \ddots\end{bmatrix} \tag{78}$$

Define the set $E = \{\boldsymbol{y} : \boldsymbol{y}^T\boldsymbol{S}^{-1}\boldsymbol{y} = 1\}$. Then an equivalent statement of the modularisation inequalities (9), is the representation modularises iff E lies within the convex hull of the datapoints.

Notice that the matrix $\boldsymbol{S}$ is not even guaranteed to be positive semi-definite, meaning E is not necessarily an ellipse. In these cases no convex hull can contain the unbounded quadric it forms, and the optimal solution is mixed regardless of the particular distribution of the datapoints.

We'll begin by rewriting the inequalities, then we'll show each direction of the equivalence.

B.5.1 REWRITE INEQUALITY CONSTRAINTS

The current form of the inequality is that, $\forall \boldsymbol{w} \in \mathbb{R}^n$:

$$(\min_i \boldsymbol{w}^T \boldsymbol{s}^{[i]})^2 + \sum_{j,j' \neq j}^{d_s} w_j w_{j'} \langle s_j^{[i]} s_{j'}^{[i]} \rangle_i \geq \sum_j w_j^2 (\min_i s_j^{[i]})^2 \tag{79}$$

If this equation holds for one vector, $\boldsymbol{w}$, it also holds for all positively scaled versions of the vector, $\boldsymbol{v} = \lambda \boldsymbol{w}$ for $\lambda > 0$, so we can equivalently state that this condition must only hold for unit vectors, $\{\boldsymbol{w} \in \mathbb{R}^n : |\boldsymbol{w}| = 1\}$.

Construct the following matrix that depends on the correlations and the dimension-wise minima:

$$\boldsymbol{S} = \begin{bmatrix} (\min_i s_1^{[i]})^2 & -\langle s_1^{[i]} s_2^{[i]} \rangle_i & \dots \\ -\langle s_1^{[i]} s_2^{[i]} \rangle_i & (\min_i s_2^{[i]})^2 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{80}$$

Then this condition can be rewritten:

$$(\min_i \boldsymbol{w}^T \boldsymbol{s}^{[i]})^2 \geq \boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w} \quad \forall \boldsymbol{w} \in \mathbb{R}^n, |\boldsymbol{w}| = 1 \tag{81}$$

Further, notice that we can swap the minima in the inequality for a maxima. We can do this because if the inequality is satisfied for a particular value of $\boldsymbol{w}$ then the following inequality is satisfied for $-\boldsymbol{w}$:

$$(\max_i \boldsymbol{w}^T \boldsymbol{s}^{[i]})^2 \geq \boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w} \quad \forall \boldsymbol{w} \in \mathbb{R}^n, |\boldsymbol{w}| = 1 \tag{82}$$

Since (81) is satisfied for all $\boldsymbol{w}$, and all $\boldsymbol{w}$ have their $-\boldsymbol{w}$ unit vector pairs, so is (82). Similarly, if (82) is satisfied for all unit vectors $\boldsymbol{w}$, (81) is satisfied but using $-\boldsymbol{w}$. Hence the two are equivalent.

B.5.2 SUFFICIENCY OF CONDITION

First, let's show the convex hull enclosing E is a sufficient condition. Choose an arbitrary unit vector $\boldsymbol{w}$. Choose the point within E that maximises the dot product with $\boldsymbol{w}$:

$$\boldsymbol{y}_w = \arg\max_{\boldsymbol{y} \in E} \boldsymbol{w}^T \boldsymbol{y} \tag{83}$$

We can find $\boldsymbol{y}_w$ by lagrange optimisation under the constraint that $\boldsymbol{y}^T \boldsymbol{S}^{-1} \boldsymbol{y} = 1$:

$$\boldsymbol{y}_w = \frac{\boldsymbol{S}\boldsymbol{w}}{\sqrt{\boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w}}} \tag{84}$$

By the definition of E and its enclosure in the convex hull of the points we can write:

$$\boldsymbol{y}_w = \sum_i \lambda_i \boldsymbol{s}^{[i]} \quad \lambda_i > 0, \sum_i \lambda_i \leq 1 \tag{85}$$

Now:

$$\boldsymbol{w}^T \boldsymbol{y} = \sqrt{\boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w}} = \sum_i \lambda_i \boldsymbol{w}^T \boldsymbol{s}^{[i]} \leq \max_i \boldsymbol{w}^T \boldsymbol{s}^{[i]} \sum_i \lambda_i = \max_i \boldsymbol{w}^T \boldsymbol{s}^{[i]} \tag{86}$$

Therefore, we get our our main result:

$$(\max_i \boldsymbol{w}^T \boldsymbol{s}^{[i]})^2 \geq \boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w} \tag{87}$$

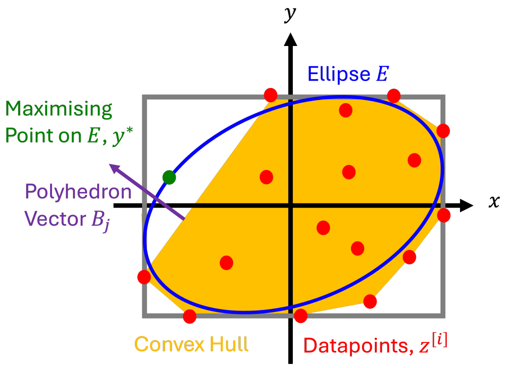


Figure 11: If the convex hull does not enclose the ellipse then you can always find a unit vector that breaks the inequality. In particular, this is true for the unit vector that defines the faces of the convex hull, $\boldsymbol{B}_j$.

B.5.3 NECESSITY OF CONDITION

Now we will show the necessity of this condition. We will do this by showing that if one of the points in E is not within the convex hull of the datapoints then we can find one of the inequalities in (82) that is broken.

First, get all the datapoints that for some unit vector $\boldsymbol{w}$ maximise $\boldsymbol{w}^T \boldsymbol{s}^{[i]}$ over the dataset. Construct the convex hull of these points, the polyhedra P. You can equivalently characterise any polyhedra by a set of linear inequalities: $P = \{\boldsymbol{x} \in \mathbb{R} : \boldsymbol{B}\boldsymbol{x} \leq \boldsymbol{b}\}$. Denote with $\boldsymbol{B}_j$ the jth row of $\boldsymbol{B}$. We can always choose to write the inequalities in such a way that the rows of $\boldsymbol{B}$ are unit length, by appropriately rescaling each b_j. Assume we have done this.

Assume the convex hull does not contain the entire ellipse. Therefore there is at least one point on the ellipse that breaks one of the inequalities defining the polyhedra. These points are charactersied by being both on the ellipse, $\boldsymbol{y}^T \boldsymbol{S}^{-1} \boldsymbol{y} = 1$, but at least one of the inequalities defining the polyhedra is broken: $\boldsymbol{B}_j^T \boldsymbol{y} > b_j$. Let's call the set of points on the ellipse for which this inequality is broken $\mathscr{B}_j = \{\boldsymbol{y} \in \mathbb{R}^n : \boldsymbol{y}^T \boldsymbol{S}^{-1} \boldsymbol{y} = 1 \quad \text{and} \quad \boldsymbol{B}_j^T \boldsymbol{y} > b_j\}$.

Now, choose the element of $\mathscr{B}_j$ that maximises $\boldsymbol{B}_j^T \boldsymbol{y}$, call it $\boldsymbol{y}^*$. This will also be the element of E that maximises $\boldsymbol{B}_j^T \boldsymbol{y}$, which by the previous lagrange optimisation, (84), has the following dot product: $\boldsymbol{B}_j^T \boldsymbol{y}^* = \sqrt{\boldsymbol{B}_j^T \boldsymbol{S} \boldsymbol{B}_j}$.

But now we derive a contradiction. This point is outside the polyhedra, $\boldsymbol{B}_j^T \boldsymbol{y} > b_j$, whereas all points inside the polyhedra, including all the datapoints, satisfy all inequalities, hence, $\boldsymbol{B}_j^T \boldsymbol{s}^{[i]} \leq b_j \forall \boldsymbol{s}^{[i]}$. But, $\boldsymbol{B}_j^T \boldsymbol{y}^* = \sqrt{\boldsymbol{B}_j^T \boldsymbol{S} \boldsymbol{B}_j} > b_j$, therefore:

$$\boldsymbol{B}_j^T \boldsymbol{s}^{[i]} \leq b_j < \sqrt{\boldsymbol{B}_j^T \boldsymbol{S} \boldsymbol{B}_j} \quad \forall \boldsymbol{s}^{[i]} \tag{88}$$

And hence we've found a unit vector $\boldsymbol{B}_i$ such that:

$$\max_i \boldsymbol{B}_j^T \boldsymbol{s}^{[i]} < \sqrt{\boldsymbol{B}_j^T \boldsymbol{S} \boldsymbol{B}_j} \quad \rightarrow \quad (\max_i \boldsymbol{B}_j^T \boldsymbol{s}^{[i]})^2 < \boldsymbol{B}_j^T \boldsymbol{S} \boldsymbol{B}_j \tag{89}$$

Hence one of the inequalities in (82) is broken, showing the necessity.

B.5.4 GENERALISATION TO LINEARLY MIXED SOURCES

Using an exactly analogous strategy, one can show that, in the setting of Theorem B.4, an equivalent condition for the set of inequalities to be satisfied, is for the convex hull of the datapoints to contain the ellipse $E = \{\boldsymbol{y} : \boldsymbol{y}^T \boldsymbol{S}^{-1} \boldsymbol{y}\}$, where:

$$\boldsymbol{S} = \lambda \boldsymbol{D}^{-2}(\boldsymbol{A}^T \boldsymbol{A})\boldsymbol{D}^{-2} - \lambda(\boldsymbol{A}^T \boldsymbol{A})^{-1} - \langle \boldsymbol{s}^{[i]}(\boldsymbol{s}^{[i]})^T \rangle_i \tag{90}$$

C Extension to all Activity Norms

We now extend our theoretical results to the same setting as the main theorem, Theorem B.1, but with other activity norms rather than L2. As shown in theorem Theorem B.2, the weight loss is minimised by orthogonalising the coding of different variables, so we just focus on the activity loss and will recombine later. We will show that, for all norms, modular solutions are optimal if the variables are range independent.

Consider a set of modular neurons:

$$\boldsymbol{z}_n^{[i]} = \begin{bmatrix} |u_{n1}|(\bar{s}_1^{[i]} - \min_j \bar{s}_1^{[j]}) \\ \vdots \\ |u_{nd_s}|(\bar{s}_{d_s}^{[i]} - \min_j \bar{s}_{d_s}^{[i]}) \end{bmatrix} \geq \mathbf{0} \tag{91}$$

And let's see if we could usefully mix these encodings together for any set of mixing coefficients $\boldsymbol{u}_n \in \mathbb{R}^{d_s}$:

$$z_n^{[i]} = \sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]} + \Delta_n = \sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]} - \min_i [\sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]}] \geq 0 \tag{92}$$

If the variables are range-independent then we can move the min inside the sum:

$$\Delta_n = -\min_i [\sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]}] = -\sum_{j=1}^{d_s} \min_i u_{nj}\bar{s}_j^{[i]} \tag{93}$$

Then the Lp norm is:

$$\langle |z_n^{[i]}|^p \rangle_i = \langle \left| \left(\sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]} + \Delta_n \right) \right|^p \rangle_i = \langle \left| \left(\sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]} - \min_i u_{nj}\bar{s}_j^{[i]} \right) \right|^p \rangle_i \tag{94}$$

Now using the convexity of the Lp norm:

$$\langle \left| \left(\sum_{j=1}^{d_s} u_{nj}\bar{s}_j^{[i]} - \min_i u_{nj}\bar{s}_j^{[i]} \right) \right|^p \rangle_i \geq \sum_{j=1}^{d_s} \langle |u_{nj}\bar{s}_j^{[i]} - \min_i u_{nj}\bar{s}_j^{[i]}|^p \rangle_i \tag{95}$$

with equality only if p is equal to 1, or if at most one of the entries of $\boldsymbol{u}$ are 0.

Finally, since we orient our variables such that $|\min_i \bar{s}_j^{[i]}| < \max_i \bar{s}^{[i]}$, without loss of generality,:

$$\langle |u_{nj}\bar{s}_j^{[i]} - \min_i u_{nj}\bar{s}_j^{[i]}|^p \rangle_i \geq \langle |u_{nj}|^p \langle |\bar{s}_j^{[i]} - \min_i \bar{s}_j^{[i]}|^p \rangle_i \tag{96}$$

So we get our final result that the mixed activity loss is greater than or equal to the modular activity loss if the variables are range independent:

$$\langle |z_n^{[i]}|^p \rangle_i \geq \sum_{j=1}^{d_s} |u_{nj}|^p \langle |\bar{s}_j^{[i]} - \min_i \bar{s}_j^{[i]}|^p \rangle_i = \langle |\boldsymbol{z}_n^{[i]}|^p \rangle_i \tag{97}$$

So, as long as $p > 1$, the activity loss is lowered by modularising the representation of range-independent variables. Further, since the weight loss is also minimised, the optimal solution will be modular.

Two caveats: first, in the particular case of the L1 norm all that we can guarantee is that the activity loss of a modular solution is equal to that of a mixed solution. This means that, even in the case of range-independent variables, there might be orthogonal but mixed solutions that are equally good as the modular solution. For $p > 1$ this caveat is not needed.

Second, this is a set of sufficient conditions, range-independent variables are optimally encoded modularly. There will be other nearly range-independent variables that are also optimally encoded modularly, just as we found for the L2 case.

D EXTENSION TO MULTIDIMENSIONAL VARIABLES

Many variables in the real world, such as angles on a circle, are not 1-dimensional. Ideally our theory would tell us when such multi-dimensional sources were modularised. Unfortunately, precise necessary and sufficient conditions are not an easy step from our current theory because our proof strategies make extensive use of the optimal modular encoding. For one-dimensional sources this is very easy to find, simply align each source correctly in a neuron, and make the representation positive. Finding similar optimal modular solutions for multidimensional sources is much harder, potentially as hard as simply solving the optimisation problem.

However, we can make progress on a simpler problem: showing that range-independence is a sufficient condition for a set of multi-dimensional sources to modularise. Consider two multidimensional sources $\boldsymbol{x}^{[i]} \in \mathbb{R}^{d_x}$ and $\boldsymbol{y}^{[i]} \in \mathbb{R}^{d_y}$, and an encoding in which they are mixed in single neurons:

$$z_n^{[i]} = \sum_{j=1}^{d_x} u_{nj} x_j^{[i]} + \sum_{j=1}^{d_y} v_{nj} y_j^{[i]} + \Delta_n \tag{98}$$

Whatever choice of coefficient vectors, $\boldsymbol{u}$ and $\boldsymbol{v}$, or activity norm (as long as $p > 1$), we can consider breaking this mixed coding into two modular neurons. Define new variables $a^{[i]} = \sum_{j=1}^{d_x} u_{nj} x_j^{[i]}$ and $b^{[i]} = \sum_{j=1}^{d_y} v_{nj} y_j^{[i]}$. If the two multidimensional sources are range-independent, then so too are a and b. Define the modular representation:

$$\boldsymbol{z}_n^{[i]} = \begin{bmatrix} a^{[i]} - \min_j a^{[j]} \\ b^{[i]} - \min_j b^{[j]} \end{bmatrix} \tag{99}$$

All of our previous arguments apply to argue that this modular encoding will be preferred if the multidimensional variables are range-independent.

We just have to show that the weight loss is also reduced by modularising multidimensional sources, a small extension of Theorem B.2, which we shall now show. Write a modular code as:

$$\boldsymbol{z}^{[i]} = \boldsymbol{U}\boldsymbol{x}^{[i]} - \min_j [\boldsymbol{U}\boldsymbol{x}^{[j]}] + \boldsymbol{V}\boldsymbol{y}^{[i]} - \min_j [\boldsymbol{V}\boldsymbol{y}^{[j]}] \tag{100}$$

With the encoding of $\boldsymbol{x}$ and $\boldsymbol{y}$ spanning two different spaces: $\boldsymbol{U}^T\boldsymbol{V} = \boldsymbol{0}$.

Then define the minimal L2 norm readout weight matrix using the pseudoinverse:

$$\boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{U}^\dagger \\ \boldsymbol{V}^\dagger \end{bmatrix} \tag{101}$$

Such that $\boldsymbol{W}_{\text{out}}\boldsymbol{z}^{[i]} = \begin{bmatrix} \boldsymbol{x}^{[i]} \\ \boldsymbol{y}^{[i]} \end{bmatrix}$ up to some constant offset that can be removed by the bias term in the readout. Now consider mixing the representation of the two multidimensional sources using an orthogonal matrix that preserves the encoding size of each variable, $\boldsymbol{O}$:

$$\boldsymbol{z}^{[i]} = \boldsymbol{U}\boldsymbol{x}^{[i]} + \boldsymbol{O}\boldsymbol{V}\boldsymbol{y}^{[i]} \qquad \boldsymbol{W}_{\text{out}} = \boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{U}^\dagger + \boldsymbol{A} \\ \boldsymbol{V}^\dagger\boldsymbol{O}^T + \boldsymbol{B} \end{bmatrix} \tag{102}$$

Then in the subspace spanned by the columns of $\boldsymbol{U}$ $\boldsymbol{W}_{\text{out}}$ must be the same as $\boldsymbol{U}^\dagger$. However, in order to offset the newly aligned matrix $\boldsymbol{O}\tilde{\boldsymbol{V}}$ there must be an additional component orthogonal, $\boldsymbol{A}$:

$$\begin{aligned} \boldsymbol{W}_{out}\boldsymbol{U} = \boldsymbol{U}^\dagger\boldsymbol{U} + \boldsymbol{A}\boldsymbol{U} = \mathbb{I} \qquad \boldsymbol{A}\boldsymbol{U} = \boldsymbol{0} \\ \boldsymbol{W}_{out}\boldsymbol{O}\boldsymbol{V} = \boldsymbol{U}^\dagger\boldsymbol{O}\boldsymbol{V} + \boldsymbol{A}\boldsymbol{O}\boldsymbol{V} = \boldsymbol{0} \qquad \boldsymbol{A}\boldsymbol{O}\boldsymbol{V} = -\boldsymbol{U}^\dagger\boldsymbol{O}\boldsymbol{V} \end{aligned} \tag{103}$$

So $\boldsymbol{A}$ and $\boldsymbol{U}^\dagger$ live in orthogonal subspaces, and $\boldsymbol{A} \neq 0$. The same applies to $\boldsymbol{V}$ and $\boldsymbol{B}$, therefore:

$$||\boldsymbol{W}_{out}||_F^2 = ||\boldsymbol{U}^\dagger||_F^2 + ||\boldsymbol{V}^\dagger||_F^2 + ||\boldsymbol{A}||_F^2 + ||\boldsymbol{B}||_F^2 \geq ||\boldsymbol{U}^\dagger||_F^2 + ||\boldsymbol{V}^\dagger||_F^2 \tag{104}$$

And hence orthogonalising the subspaces reduces the loss in this case as well.

Combining these two, the optimal representation of range-independent multi-dimensional variables for activity losses using an Lp norm for $p > 1$ is modular.

E POSITIVE LINEAR RECURRENT NEURAL NETWORKS

One of the big advantages of changing the phrasing of the modularising constraints from statistical properties (as in Whittington et al. (2023a)) to range properties is that it naturally generalises to recurrent dynamical formulations, and recurrent networks are often a much more natural setting for neuroscience. To illustrate this we'll study positive linear RNNs. Linear dynamical systems can only produce mixtures of decaying or growing sinusoids, so we therefore ask the RNN to produce different frequency outputs via an affine readout:

$$\boldsymbol{W}_{\text{out}}\boldsymbol{g}(t) + \boldsymbol{b}_{\text{out}} = \begin{bmatrix} \cos(\omega_1 t) \\ \cos(\omega_2 t) \end{bmatrix} \tag{105}$$

Then we'll assume the internal dynamical structure is a standard linear RNN:

$$\boldsymbol{W}_{\text{rec}}\boldsymbol{g}(t) + \boldsymbol{b} = \boldsymbol{g}(t + \Delta t) \tag{106}$$

We will study this two frequency setting and ask when the representation learns to modularise the two frequencies. These results could be generalised to multiple variables as in section B.2. Again, our representation must be non-negative, $\boldsymbol{g}(t) \geq \boldsymbol{0}$, and we minimise the following energy loss:

$$\mathscr{L} = \langle ||\boldsymbol{g}(t)||^2 \rangle + \lambda_W ||\boldsymbol{W}_{\text{rec}}||_F^2 + \lambda_R ||\boldsymbol{W}_{\text{out}}||_F^2 \tag{107}$$

We will show that, if λ_W is sufficiently large and the RNN solves the task for infinite time, the optimal representation contains activity cycling at only the two frequencies, ω_1 and ω_2. Further we'll show that these two parts should modularise if the two frequencies are irrational ratios of one another, and should mix if one frequency is an integer multiple of the other, i.e. the frequencies are harmonics. Finally, we conjecture, show empircally, and present proof in limited cases, that non-harmonic rational frequency ratios should modularise. If the task runs for finite time then a smudge factor is allowed, the frequencies can be near integer multiples, and how close you have to be to mix is governed by the length of the task. As long as during the task the frequencies are 'effectively' integer multiples, the representation will mix.

To show this we will first study the weight losses, show that they are minimised if each frequency is modularised from the others. We will then argue that when λ_W is sufficiently large the representation will contain only two frequencies. Finally, we will study when the activity loss is minimised. Then, as in Appendix B.1, when all losses agree on their minima, the representation modularises, else it mixes.

E.1 STRUCTURE OF REPRESENTATION - ACT I

Due to the autonomous linear system architecture, the representation can only contain a linear mixture of growing, decaying, or stable frequencies, and a constant. The growing and decaying modes cannot help you solve the task (since if they do at one point in time, they later won't), and will either cost infinite energy (the growing modes) or will cost some energy without helping to reduce the bias (since, again, the bias has to be set large enough such that after the decaying mode has disappeared the representation is still positive). Hence,

$$\boldsymbol{g}(t) = \sum_{i=1}^{F} \boldsymbol{a}_i \cos(\omega_i t) + \boldsymbol{b}_i \sin(\omega_i t) + \boldsymbol{b}_0 \tag{108}$$

Where F is smaller than half the number of neurons to ensure the autonomous linear system can propagate each frequency (each frequency 'uses' 2 dimensions, as will be obvious from Appendix E.3.

E.2 READOUT LOSS

The readout loss is relatively easy, again create some capitalised pseudoinverse vectors $\{\boldsymbol{A}_i, \boldsymbol{B}_i\}_{i=1}^F$ defined by being the min-norm vectors with the property that $\boldsymbol{A}_i^T \boldsymbol{a}_j = \delta_{ij}$ and $\boldsymbol{A}_i^T \boldsymbol{b}_j = 0$, and the same for $\boldsymbol{B}_i$. Then the min-norm readout matrix is:

$$\boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \end{bmatrix} \tag{109}$$

And the readout loss is:

$$|\boldsymbol{W}_{\text{out}}|_F^2 = \text{Tr}[\boldsymbol{W}_{\text{out}}\boldsymbol{W}_{\text{out}}^T] = |\boldsymbol{A}_1|^2 + |\boldsymbol{A}_2|^2 \tag{110}$$

Each vector has the following form, $\boldsymbol{A}_i = \frac{1}{\boldsymbol{a}_i^2}\boldsymbol{a}_i + \boldsymbol{a}_{i,\perp}$, where $\boldsymbol{a}_{i,\perp}$ is orthogonal to $\boldsymbol{a}_i$ and is included to ensure the correct orthogonality properties hold. So, for a fixed encoding size, the readout loss is minimised if $\boldsymbol{a}_{i,\perp} = 0$. This occurs when the encodings are orthogonal (i.e. $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are orthogonal from one another, all other $\boldsymbol{a}_i$ vectors, and from each of the $\boldsymbol{b}_i$ vectors). This happens if the two frequencies are modularised from one another, and additionally the sine and cosine vectors for each frequency are orthogonal. I.e. a modularised solution with this property has the minimal readout weight loss for a given encoding size.

E.3 RECURRENT LOSS WITHOUT BIAS

First we consider the slightly easier case where there is no bias in the recurrent dynamics:

$$\boldsymbol{W}\boldsymbol{g}(t) = \boldsymbol{g}(t+1) \tag{111}$$

Then we will write down a convenient decomposition of the min-norm $\boldsymbol{W}$. Call the matrix of stacked coefficient vectors, $\boldsymbol{X}$:

$$\boldsymbol{X} = [\boldsymbol{a}_1 \quad \boldsymbol{b}_1 \quad \dots \quad \boldsymbol{a}_F \quad \boldsymbol{b}_F \quad \boldsymbol{b}_0] \tag{112}$$

Similarly, call the matrix of stacked normalised psuedo-inverse vectors $\boldsymbol{X}^\dagger$:

$$\boldsymbol{X}^\dagger = [\boldsymbol{A}_1 \quad \boldsymbol{B}_1 \quad \dots \quad \boldsymbol{A}_F \quad \boldsymbol{B}_F \quad \boldsymbol{B}_0] \tag{113}$$

And finally create an ideal rotation matrix:

$$\begin{aligned}\boldsymbol{R} = &\begin{bmatrix} \cos(\omega_1\Delta t) & -\sin(\omega_1\Delta t) & \dots & 0 & 0 & 0 \\ \sin(\omega_1\Delta t) & \cos(\omega_1\Delta t) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \cos(\omega_F\Delta t) & -\sin(\omega_F\Delta t) & 0 \\ 0 & 0 & \dots & \sin(\omega_F\Delta t) & \cos(\omega_F\Delta t) & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}_1 & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{R}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \boldsymbol{R}_F & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}\end{aligned} \tag{114}$$

Where each $\boldsymbol{R}_i$ is a 2×2 rotation matrix at frequency ω_i. Now:

$$\boldsymbol{W} = \boldsymbol{X}\boldsymbol{R}\boldsymbol{X}^{\dagger,T} \tag{115}$$

We can then calculate the recurrent weight loss:

$$|\boldsymbol{W}|_F^2 = \text{Tr}[\boldsymbol{W}\boldsymbol{W}^T] = \text{Tr}[\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{R}\boldsymbol{X}^{\dagger,T}\boldsymbol{X}^\dagger\boldsymbol{R}^T] \tag{116}$$

$\boldsymbol{X}^T\boldsymbol{X}$ is a symmetric $2F+1 \times 2F+1$ positive-definite matrix, and its inverse is $\boldsymbol{X}^{\dagger,T}\boldsymbol{X}^\dagger$, another symmetric positive-definite matrix. To see that these matrices are inverses of one another perform the singular value decomposition, $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, and $\boldsymbol{X}^\dagger = \boldsymbol{U}\boldsymbol{S}^{-1}\boldsymbol{V}^T$. Then $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T$ and $\boldsymbol{X}^{\dagger,T}\boldsymbol{X}^\dagger = \boldsymbol{V}\boldsymbol{S}^{-2}\boldsymbol{V}^T$, which are clearly inverses of one another.

Introducing a new variable, $\boldsymbol{Y} = \boldsymbol{X}^T\boldsymbol{X}$:

$$|\boldsymbol{W}|_F^2 = \text{Tr}[\boldsymbol{Y}\boldsymbol{R}\boldsymbol{Y}^{-1}\boldsymbol{R}^T] \tag{117}$$

We then use the following trace inequality, from Ruhe (Ruhe, 1970). For two positive semi-definite symmetric matrices, $\boldsymbol{E}$ and $\boldsymbol{F}$, with ordered eigenvalues, $e_1 \geq \dots \geq e_n \geq 0$ and $f_1 \geq \dots \geq f_n \geq 0$

$$\text{Tr}[\boldsymbol{E}\boldsymbol{F}] \geq \sum_{i=1}^{n} e_i f_{n-i+1} \tag{118}$$

Now, since $\boldsymbol{R}\boldsymbol{Y}^{-1}\boldsymbol{R}^T$ and $\boldsymbol{Y}^{-1}$ are similar matrices, they have the same eigenvalues, and $\boldsymbol{Y}^{-1}$ is the inverse of $\boldsymbol{Y}$ so its eigenvalues are the inverse of those of $\boldsymbol{Y}$. Therefore:

$$|\boldsymbol{W}|_F^2 = \mathrm{Tr}[\boldsymbol{Y}\boldsymbol{R}\boldsymbol{Y}^{-1}\boldsymbol{R}^T] \geq \sum_{i=1}^{2F+1} \frac{\lambda_i}{\lambda_i} = 2F+1 \tag{119}$$

Then we can show that this lower bound on the weight loss is achieved when the coefficient vectors are orthogonal, hence making the modular solution optimal. If all the coefficient vectors are orthogonal then $\boldsymbol{Y}$ and $\boldsymbol{Y}^{-1}$ are diagonal, so they commute with any matrix, and:

$$|\boldsymbol{W}|_F^2 = \mathrm{Tr}[\boldsymbol{Y}\boldsymbol{R}\boldsymbol{Y}^{-1}\boldsymbol{R}^T] = \mathrm{Tr}[\boldsymbol{Y}\boldsymbol{Y}^{-1}\boldsymbol{R}^T\boldsymbol{R}] = \mathrm{Tr}[\mathbb{I}_{2F+1}] = 2F+1 \tag{120}$$

E.4 RECURRENT LOSS WITH BIAS

Now we return to the case of interest:

$$\boldsymbol{W}\boldsymbol{g}(t) + \boldsymbol{b} = \boldsymbol{g}(t+1) \tag{121}$$

Our energy loss, equation 107, penalises the size of the weight matrix $|\boldsymbol{W}|_F^2$ and not the bias. Therefore, if we can make $\boldsymbol{W}$ smaller by assigning some of its job to $\boldsymbol{b}$ then we should. We can do this by setting $\boldsymbol{b}_0 = \boldsymbol{b}$ (recall the definition of $\boldsymbol{b}_0$ from equation 108) and constructing the following, smaller, min-norm weight matrix:

$$\boldsymbol{W} = [\boldsymbol{a}_1 \quad \boldsymbol{b}_1 \quad \dots \quad \boldsymbol{a}_F \quad \boldsymbol{b}_F] \begin{bmatrix} \boldsymbol{R}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{R}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{R}_F \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{B}_1^T \\ \vdots \\ \boldsymbol{A}_F^T \\ \boldsymbol{B}_F^T \end{bmatrix} = \hat{\boldsymbol{X}}\hat{\boldsymbol{R}}\hat{\boldsymbol{X}}^\dagger \tag{122}$$

Using the definitions in the previous section. This slightly complicates our previous analysis because now $\hat{\boldsymbol{X}}^{\dagger,T}\hat{\boldsymbol{X}} = \hat{Y}^\dagger$ is not the inverse of $\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}} = \hat{\boldsymbol{Y}}$. If $\boldsymbol{b}$ is orthogonal to all the vectors $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^F$, then it is the inverse, and the previous proof that modularity is an optima goes through.

Fortunately, it is easy to generalise to this setting. $\boldsymbol{X}^\dagger$ is not quite the pseudoinverse of $\boldsymbol{X}^\dagger$ because its vectors have to additionally be orthogonal to $\boldsymbol{b}$. This means we can break down each of the vectors into two components, for example:

$$\boldsymbol{A}_1 = \hat{\boldsymbol{A}}_1 + \hat{\boldsymbol{A}_{1,\perp}} \tag{123}$$

The first of these vectors is the transpose of the equivalent row of the pseudoinverse of $\hat{\boldsymbol{X}}$, it lives in the span of the vectors $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^F$. The second component is orthogonal to this span and ensures that $\boldsymbol{A}_1^T\boldsymbol{b} = 0$. Hence, the previous claim. If $\boldsymbol{b}$ is orthogonal to the span of $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^F$, this is the standard pseudoinverse and the previous result goes through.

We can express the entire $\hat{\boldsymbol{X}}^\dagger$ matrix in these two components:

$$\hat{\boldsymbol{X}}^\dagger = \hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^\dagger \tag{124}$$

Then:

$$\hat{\boldsymbol{Y}}^\dagger = (\hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^\dagger)^T(\hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^\dagger) = \hat{\boldsymbol{X}}_0^{\dagger,T}\hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^{\dagger,T}\hat{\boldsymbol{X}}_\perp^\dagger = \hat{\boldsymbol{Y}}^{-1} + \hat{\boldsymbol{Y}}_\perp^\dagger \tag{125}$$

Both of these new matrices are positive semi-definite matrices, since they are formed by taking the dot product of a set of vectors. Then:

$$|\boldsymbol{W}|_F^2 = \mathrm{Tr}[\hat{\boldsymbol{Y}}\hat{\boldsymbol{R}}\hat{\boldsymbol{Y}}^{-1}\hat{\boldsymbol{R}}^T] + [\hat{\boldsymbol{Y}}\hat{\boldsymbol{R}}\hat{\boldsymbol{Y}}_\perp^\dagger\hat{\boldsymbol{R}}^T] \tag{126}$$

Now, the first term is greater than or equal than $2F$, as in the previous setting. And since $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{Y}}_\perp^\dagger$ are positive semi-definite, and so therefore is $\hat{\boldsymbol{R}}\hat{\boldsymbol{Y}}_\perp^\dagger\hat{\boldsymbol{R}}^T$, this second term is greater than or equal to 0. Hence, $|\boldsymbol{W}|_F^2 \geq 2F$, and orthogonal encodings achieve this bound, therefore it is an optimal solution according to the weight loss.

E.5 STRUCTURE OF REPRESENTATION - ACT II - LARGE λ_W

We saw that our representation takes the following form.

$$\boldsymbol{g}(t) = \sum_{i=1}^{F} \boldsymbol{a}_i \cos(\omega_i t) + \boldsymbol{b}_i \sin(\omega_i t) + \boldsymbol{b}_0 \tag{127}$$

Further, and we saw in the previous section that each added frequency increases the recurrent weight loss. To solve the task we only need two frequencies, why would we have more than two? The only possibility is that by including additional frequencies we might save activity energy, at the cost of weight energy. We will simplify our life for now by saying λ_W is very large, so our representation only has two frequencies. We now turn to our activity loss and ask which frequencies should mix, and which modularise.

E.6 FREQUENCIES

Consider a mixed frequency neuron:

$$g_i(t) = (\boldsymbol{a}_1)_i \cos(\omega_1 t) + (\boldsymbol{b}_1)_i \sin(\omega_1 t) + (\boldsymbol{a}_2)_i \cos(\omega_2 t) + (\boldsymbol{b}_2)_i \sin(\omega_2 t) + b_n \tag{128}$$

By rescaling and shifting time, we can rewrite all such mixed neurons in a simpler form:

$$g_i(t) = \alpha_i \cos(t) + \beta_i \cos(\omega t + \phi) + b_i \quad \omega > 1 \tag{129}$$

Where ω is the ratio of the larger to the smaller frequencies. We will show that the activity of this mixed neuron is lower than its corresponding modular counterpart when ω is an integer, and higher if ω is irrational. Finally, we show empirically, and some theoretical evidence, that the same is true of rational, non-integer ω. Hence, we find the activity energy is minimised by modularising unless one of the frequencies is an integer multiple of the others. Further, since if the mixed neuron $g_i(t)$ is preferred over its modular counterpart, then so too are positively scaled versions, $\mu g_i(t)$, we will consider responses of the form:

$$g(t) = \cos(t) + \delta \cos(\omega t + \phi) + \Delta \quad \omega > 1 \tag{130}$$

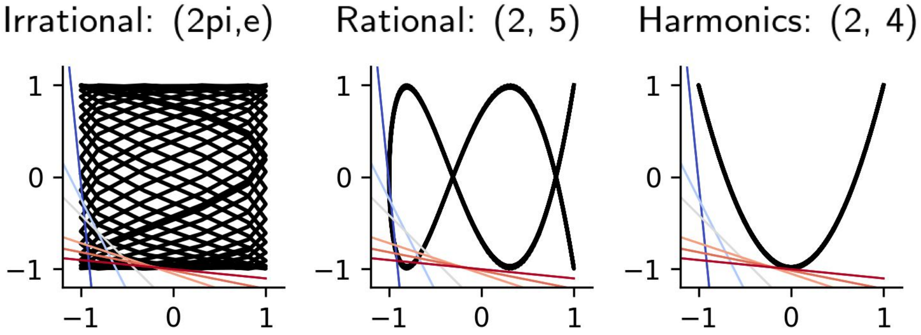


Figure 12: Schematics showing irrational and rational ratio case the data are range dependent beyond the modular-mixed boundary while in de-modularising harmonics case, the two periodic waves are range independent.

Irrational Frequencies Modularise If ω is irrational then, by Kronecker's theorem, you can find a value of t for which $\cos(t)$ and $\cos(\omega t + \phi)$ take any pair of values. This makes the two frequencies, among other things, extreme point independent, and therefore no mixing is better than modularising.

Even Integer Multiples Mix According to Theorem 2.1, to modularise, for all δ and ϕ values:

$$\Delta > \sqrt{1 + \delta^2} = 1 + \mathscr{O}(\delta^2) \tag{131}$$

We will therefore try to find a δ and ϕ which breaks this inequality. Consider $\delta < 0$ and very small and $\phi = 0$, then, writing $\omega = 2m$ for integer m:

$$\Delta = -\min_i[\cos(t) + \delta\cos(2mt)] \tag{132}$$

$\cos(t) - |\delta|\cos(2mt)$ takes its extreme values when t is an integer multiple of π, and the smallest it can be is $-1 + |\delta|$. Hence:

$$\Delta = 1 - \delta < 1 + \mathscr{O}(\delta^2) \tag{133}$$

This is smaller than the critical value, so the representation should mix, since at least one mixing inequality was broken.

Odd Integer Multiples Mix If ω is an odd integer then we can instead mix cosine with sine by choosing $\phi = \frac{\pi}{2}$:

$$g_i(t) = \cos(t) + \delta\sin(\omega t) + \Delta \tag{134}$$

Then the same argument goes through as above.

Other Rational Multiples Modularise Now consider $\omega = \frac{p}{q}$ for two integers p and $q \neq 1$. We were inspired in this section by the mathoverflow post of Soudry and Speiser (2015). So we are considering the mixed encoding:

$$g(t) = \cos(t) + \delta\cos(\frac{p}{q}t + \phi) + \Delta \tag{135}$$

And we have to show that for all δ and ϕ, $\Delta^2 > 1 + \delta^2$. For general ϕ we have been unable to do this, but for $\phi = 0$ we present proof below in the hope this can be generalised.

We break down the problem into four cases based on the sign of δ, and whether p and q are both odd, or only one is. The easiest is if both are odd, and $\delta > 0$. Then take $t = q\pi$:

$$g(t) = \cos(q\pi) + \delta\cos(p\pi) + \Delta = -1 - \delta + \Delta > 0 \tag{136}$$

Therefore Δ is at least $1 + \delta$, which is larger than $\sqrt{1 + \delta^2}$, hence modularising is better.

Conversely, if one of p or q is even (let's say p w.l.o.g.) and $\delta < 0$ then choose $t = p\pi$ again:

$$g(t) = -1 + \delta\cos(p\pi) + \Delta = -1 - |\delta| + \Delta > 0 \tag{137}$$

And the same argument holds.

Now consider the case where $\delta > 0$ and one of p or q is odd. Then there exists an odd integer k such thatSoudry and Speiser (2015):

$$kp = q + 1 \mod 2q \tag{138}$$

Therefore take $t = k\pi$:

$$g(t) = \cos(k\pi) + \delta\cos(\pi(\frac{kp}{q})) + \Delta = -1 + \delta\cos(\pi(\frac{q + 1 + 2nq}{q})) + \Delta \tag{139}$$

For some integer n. Developing:

$$g(t) = -1 - \delta\cos(\frac{\pi}{q}) + \Delta \tag{140}$$

The key question is whether for any Δ below the critical value this representation is nonnegative. For this to be true:

$$-1 - \delta\cos(\frac{\pi}{q}) + \sqrt{1 + \delta^2} \geq 0 \tag{141}$$

Developing this we get a condition on the mixing coefficient δ:

$$\delta \geq \frac{2\cos(\frac{\pi}{q})}{1 - \cos^2(\frac{\pi}{q})}. \tag{142}$$

We can apply a similar argument with $t = \frac{qk\pi}{p}$ to get:

$$-\cos(\frac{\pi}{p}) - \delta + \Delta \geq 0 \tag{143}$$

This leads us to:

$$\delta \leq \frac{1-\cos^2(\frac{\pi}{p})}{2\cos(\frac{\pi}{p})} \tag{144}$$

We squeeze inequalities eqn.142,144 and get

$$1 \leq \frac{(1-\cos^2(\pi/p))(1-\cos^2(\pi/q))}{4\cos(\pi/p)\cos(\pi/q)} \tag{145}$$

For any even, odd integer pair of $p, q \neq 1$ and $\delta > 0$, $\cos(\pi/p)$ and $\cos(\pi/q)$ are bounded in $(0, 1]$. In this case, the maximum possible value of the righthand side is achieved with the smallest possible pair of integers $(3, 4)$, for which the inequality does not hold. Thus the inequalities eqn.144, 142 cannot hold at the same time which let us to conclude that there is no Δ below the critical value satisfying nonnegativity constraint and leads to modular representation.

The last case is where $\delta < 0$ and both p, q are odd. We can extend the relation shown in eqn.138 as following by simple substitution of oddity of the variables. With given odd integers p, q, there exists an *even* integer k satisfying eqn.138. Similar to above, take $t = \frac{qk\pi}{p}$ for the mixed encoding eqn.135 and we develop:

$$g_i = -\cos(\frac{\pi}{p}) + \delta + \Delta \tag{146}$$

, which leads to the following inequality:

$$\delta \geq \frac{1-\cos^2(\pi/p)}{-2\cos(\pi/p)} \tag{147}$$

Now take $t = k\pi$, and we get:

$$g_i = 1 - \delta\cos(\pi/q) + \Delta \tag{148}$$

which leads to the in equality,

$$\delta \leq \frac{-2\cos(\pi/q)}{1-\cos^2(\pi/q)} \tag{149}$$

Again, the above two inequalities cannot hold together with the odd integers $p, q > 1$.

We also empirically show on Fig 12 that in rational multiple frequencies, there is no modular-mixed boundary possible to break the modularisation.

F METRICS FOR REPRESENTATIONAL MODULARITY AND SOURCE STATISTICAL INTERDEPENDENCE

We seek to design a metric to measure how much information a latent contains about a source. However, we would like to ignore information that could be explained through the latent's tuning to a different source. For example, perhaps two variables are correlated, if a latent encodes one of them, then it will also be slightly informative about the other despite not being functionally related. We therefore seek to condition on all of the sources bar one, and measure how much information the latent contains about the remaining source.

Dunion et al. (2023) leverage conditional mutual information in a similar way to this in a reinforcement learning context, but for training rather than evaluation, and therefore resort to a naive Monte Carlo estimation scheme that scales poorly. Instead, we leverage the identity

$$I(\boldsymbol{z}_j; \boldsymbol{s}_i | \boldsymbol{s}_{-i}) = I(\boldsymbol{z}_j; \boldsymbol{s}) - I(\boldsymbol{z}_j; \boldsymbol{s}_{-i}), \tag{150}$$

where $\boldsymbol{s}_{-i}$ is a shorthand for $\{\boldsymbol{s}_{i'} \mid i' \neq i\}$. Since this involves computing mutual information with multiple sources, we restrict ourselves to considering discrete sources and use a continuous-discrete KSG scheme (Ross, 2014) to estimate information with continuous neural activities. We normalise conditional mutual information by $H(\boldsymbol{s}_i|\boldsymbol{s}_{-i})$ to obtain a measure in $[0, 1]$. We then arrange the pairwise quantities into a matrix $\boldsymbol{C} \in \mathbb{R}^{d_s \times d_z}$ and compute the normalised average "max-over-sum" in a column as a measure of sparsity, following Hsu et al. (2023):

$$\text{CInfoM}(\boldsymbol{s}, \boldsymbol{z}) := \left(\frac{1}{d_z} \sum_{j=1}^{d_z} \frac{\max_i \boldsymbol{C}_{ij}}{\sum_{i=1}^{d_s} \boldsymbol{C}_{ij}} - \frac{1}{d_s} \right) \Bigg/ \left(1 - \frac{1}{d_s} \right). \tag{151}$$

CInfoM is appropriate for detecting arbitrary functional relationships between a source and a neuron's activity. However, in some of our experiments, the sources are provided as supervision for a linear readout of the representation. In such cases, the network cannot use information that is nonlinearly encoded, so the appropriate meausure is the degree of linearly encoded information. Operationalising this we leverage the predictive $\mathscr{V}$-information framework of Xu et al. (2020). We specify the function class $\mathscr{V}$ as linear and calculate

$$\begin{aligned} I_{\mathscr{V}}(\boldsymbol{z}_j \to \boldsymbol{s}_i | \boldsymbol{s}_{-i}) &= I_{\mathscr{V}}(\boldsymbol{z}_j \to \boldsymbol{s}) - I_{\mathscr{V}}(\boldsymbol{z}_j \to \boldsymbol{s}_{-i}) \\ &= H_{\mathscr{V}}(\boldsymbol{s}) - H_{\mathscr{V}}(\boldsymbol{s}|\boldsymbol{z}_j) - H_{\mathscr{V}}(\boldsymbol{s}_{-i}) + H_{\mathscr{V}}(\boldsymbol{s}_{-i}|\boldsymbol{z}_j), \end{aligned} \tag{152}$$

followed by a normalisation by $H_{\mathscr{V}}(\boldsymbol{s}_i|\boldsymbol{s}_{-i})$. Each predictive conditional $\mathscr{V}$-entropy term is estimated via a standard maximum log-likelihood optimization over $\mathscr{V}$, which for us amounts to either logistic regression or linear regression, depending on the treatment of the source variables as discrete or continuous. The pairwise linear predictive conditional information quantities are reduced to a single linear conditional InfoM quantity by direct analogy to (151).

Finally, to facilitate comparisons across different source distributions, we report CInfoM against the normalised multiinformation of the sources:

$$\text{NI}(\boldsymbol{s}) = \frac{\sum_{i=1}^{d_s} H(s_i) - H(\boldsymbol{s})}{\sum_{i=1}^{d_s} H(s_i) - \max_i H(s_i)}. \tag{153}$$

This allows us to test the following null hypothesis: that breaking statistical independence, rather than range independence, is more predictive of mixing. On the other hand, if range independence is more important, then source distributions that retain range independence while admitting nonzero multiinformation will better induce modularity compared to those that break range independence.

G WHAT-WHERE TASK

G.1 EXPERIMENTAL SETUP

G.1.1 DATA GENERATION

The network modularises when exposed to both simple and complex shapes. Simple shapes, used in all main paper figures, are 9-element one-hots, reflecting the active pixel's position in a 3×3 grid. Formally, for a shape at position (i, j) in the grid, the corresponding vector $s \in \mathbb{R}^9$ is given by:

$$s_k = \begin{cases} 1 & \text{if } k = 3(i-1) + j \\ 0 & \text{otherwise} \end{cases}$$

where $i, j \in \{1, 2, 3\}$.

For complex shapes, each shape is a binary vector $c \in \mathbb{R}^9$, with exactly 5 elements set to 1 and 4 elements set to 0, representing the active and inactive pixels, respectively. Each complex shape takes the (approximate) shape of a letter (13) and can be shifted to any of the 9 positions in the 3×3 grid. The standard training and testing datasets include one of every shape-position pair. The correlations introduced later include duplicates of a subset of these data points.

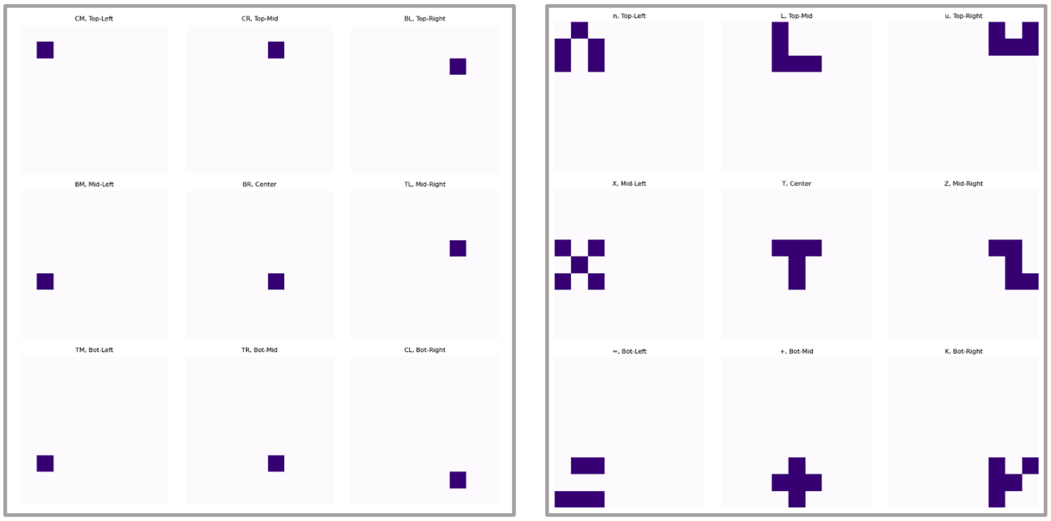


Figure 13: Different possible inputs to the network. Both one-hot (left) and letter-based (right) shapes can be outputted as either a concatenation of one-hots or as a 2D variable.

G.1.2 NETWORK ARCHITECTURE

The network we use here is designed in PyTorch and takes as input am 81-element vector, flattened to a 9×9 image (13).The network architecture is formally defined as follows:

$$\begin{aligned} &\text{Input:} \quad x \in \mathbb{R}^{81} \\ &\text{Hidden layer:} \quad \mathbf{h} = \phi(\mathbf{W}_1 x + \mathbf{b}_1), \quad \mathbf{W}_1 \in \mathbb{R}^{25 \times 81}, \quad \mathbf{b}_1 \in \mathbb{R}^{25} \\ &\text{Output layer:} \quad \mathbf{y} = \mathbf{W}_2 h + \mathbf{b}_2, \quad \mathbf{W}_2 \in \mathbb{R}^{2 \times 25}, \quad \mathbf{b}_2 \in \mathbb{R}^{2} \end{aligned}$$

where ϕ is the activation function, either ReLU or $\tanh$, depending on the experiment. Weights $\mathbf{W}_1$ and $\mathbf{W}_2$ are initialised with a normal distribution $\mathcal{N}(0, 0.01)$, and biases $\mathbf{b}_1$ and $\mathbf{b}_2$ are initialised to zero.

G.1.3 TRAINING PROTOCOLS

The network uses the Adam optimiser (Kingma and Ba (2014)), with learning rate and other hyperparameters varying from experiment to experiment. The mean squared error (MSE) is calculated separately for ‘what’ and ‘where’ tasks, and then combined along with the regularisation terms. The total loss function L_{total} is a combination of the task-specific losses and regularisation terms:

$$L_{total} = L_{what} + L_{where} + \lambda_R(||\mathbf{W}_1||_2^2 + ||\mathbf{W}_2||_2^2 + ||\mathbf{h}||_2^2)$$

where L_{task} for ‘what’ and ‘where’ tasks is defined as:

$$L_{task}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Where λ_R is the regularisation hyperparameter, typically set to 0.01 unless specified otherwise. The network is trained using the Adam optimiser with a learning rate ranging between 0.001 and 0.01, adjusted as needed. Experiments are run for order 10^4 epochs on 5 random seeds. Each experiment was executed using a single consumer PC with 8GB of RAM and across all settings, the networks achieve negligible loss.

G.2 MODULARITY OF WHAT & WHERE

G.2.1 BIOLOGICAL CONSTRAINTS ARE NECESSARY FOR MODULARITY

To understand the effect of biological constraints on modularisation, consider the optimisation problem under positivity and energy efficiency constraints. The activation function $\phi(x)$ ensures non-negativity:

$$\phi(x) = \max(0, x)$$

The energy efficiency is enforced by adding an L2 regularisation term to the loss function. By minimising the combined loss function L_{total}, the network encourages sparse and low-energy activations, leading to a separation of neurons responding to different tasks (‘what’ and ‘where’).

To illustrate the necessity of biological constraints in the modularisation of these networks, we show below the weights and activity responses of networks for networks where these constraints aren’t present. As discussed above, positivity is introduced via the ReLU activation function, and energy efficiency is defined as an L2 regularisation on the hidden weights and activities.

As shown in the neural tuning curves, the unconstrained networks do encode task features, but they do so such that each neuron responds to the specific values of both input features, they are mixed selective. Compare these results to the weights and activity responses when positivity and energy efficiency constraints are introduced. In this setting, there is a clear separation of ‘what’ and ‘where’ tasks, into two distinct sub-populations of neurons.

G.3 DROPOUT & CORRELATION

We use two different approaches to dropout in order to illustrate the importance of range independence.

Diagonal Dropout: The first approach removes example data points (i.e., shape-position pairs) from elements along the diagonals, starting in the middle and extending out to all four corners. This has the effect of increasing the mutual information between data sources but does not significantly affect their range dependence. Formally, let D be the set of all shape-position pairs. In the diagonal dropout setting, we remove pairs (s_i, p_i) where i lies along the diagonal of the input grid:

$$D' = D \setminus \{(s_i, p_i) \mid i \in \text{diagonal positions}\}$$

In the most extreme case, only one data point from each corner is removed, and this is not sufficient to force mixed-selectivity in the neurons.

Corner Dropout: The second approach removes data points from one corner of the distribution. This also increases the mutual information between sources but changes their extreme-point dependence. Specifically, we remove pairs (s_i, p_i) where i lies in the bottom-left corner of the input grid:

$$D'' = D \setminus \{(s_i, p_i) \mid i \in \text{bottom-left corner positions}\}$$

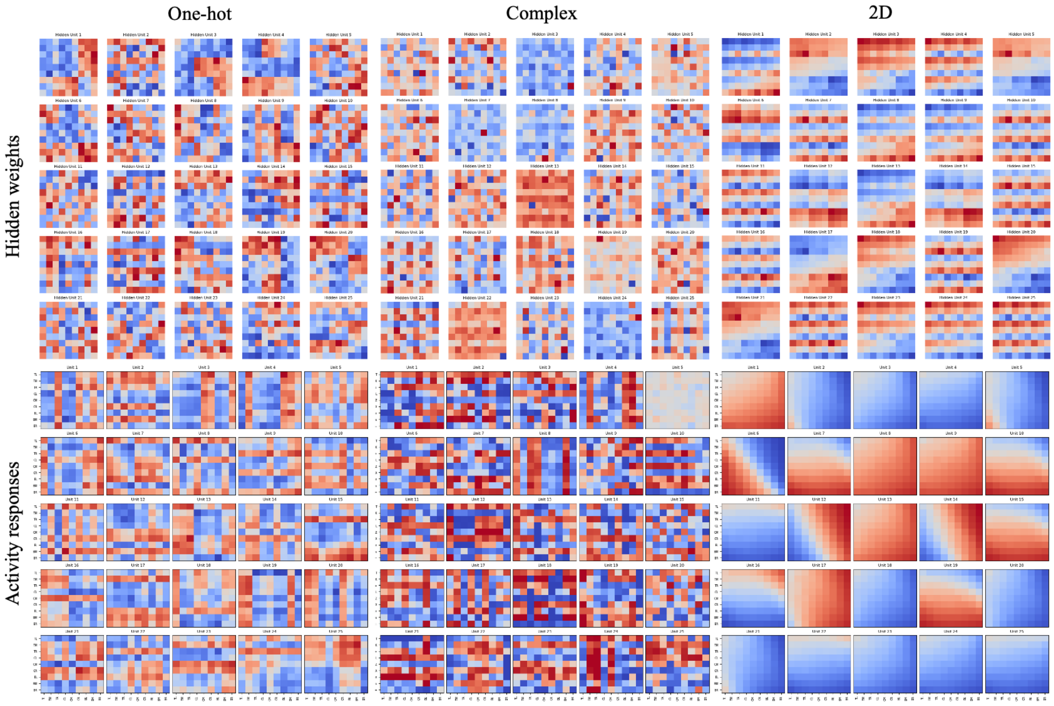


Figure 14: Hidden weights and activity responses of FF networks without biologically-inspired constraints. The one-hot and letter-like binary cases are shown, as well the 2D output setting with one-hot inputs.

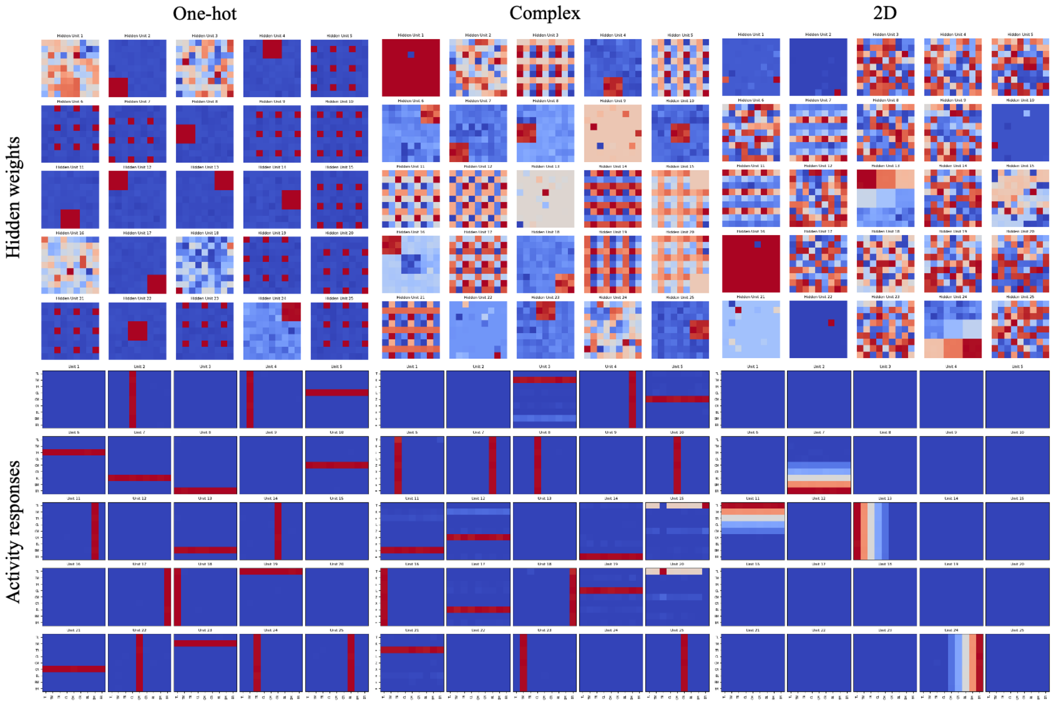


Figure 15: Weights and activity response of networks under biological constraints. The one-hot and letter-like binary cases are shown, as well the 2D output setting with one-hot inputs.

In this setting, a single data point removed from the corner is insufficient for breaking modularity; however, removing more than this causes mixed selective neurons to appear.

Correlation: To correlate sources, we duplicate data points that appear along the diagonal. This increases the mutual information between sources without affecting their range independence:

$$D''' = D \cup \{(s_i, p_i) \mid i \in \text{diagonal positions}\}$$

The mutual information $I(X;Y)$ between the shape X and position Y is calculated as follows:

$$I(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

where $P(x,y)$ is the joint probability distribution of X and Y.

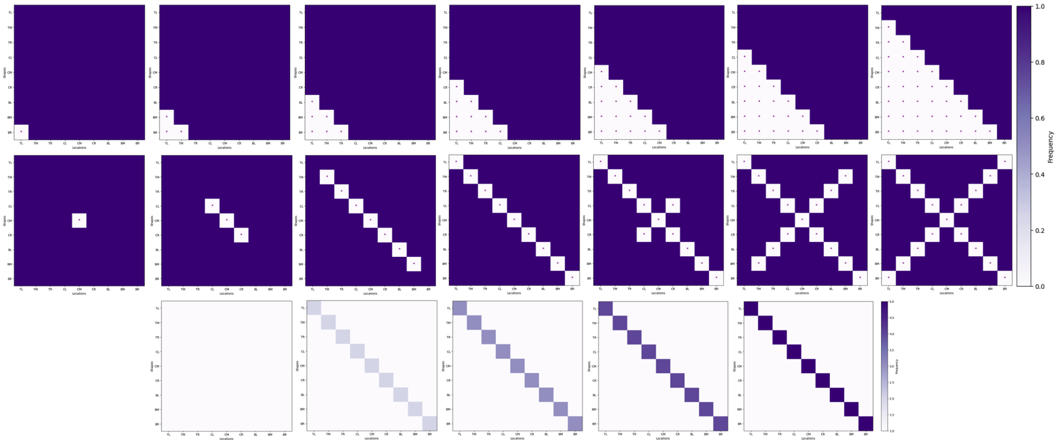


Figure 16: Increasing dropout (left to right) for both the corner-cutting (top) and diagonal (middle) cases, as well as correlation distributions (bottom). Note: asterisk denotes the absence of training data for this pair of input features.

H NONLINEAR AUTOENCODERS

H.1 AUTOENCODING SOURCES

Three-dimensional source data is sampled from $[0,1]^3$ and discretised to 21 values per dimension. The encoder and decoder are each a two-layer MLP with hidden size 16 and ReLU activation. The latent bottleneck has dimensionality six. All models use $\lambda_{\text{reconstruct}} = 1$, $\lambda_{\text{activity energy}} = 0.01$, $\lambda_{\text{activity nonnegativity}} = 1$, and $\lambda_{\text{weight energy}} = 0.0001$. Models are initialised from a He initialisation scaled by 0.3 and optimized with Adam using learning rate 0.001. Each experiment was executed using a single consumer GPU on a HPC with 2 CPUs and 4GB of RAM.

H.2 AUTOENCODING IMAGES

We subsample 6 out of the 9 sources in the Isaac3D dataset, fixing a single value for the other 3. This yields a dataset of size $12,288$. We use an expressive convolutional encoder and decoder taken from the generative modeling literature. We use 12 latents, each quantised to take on 10 possible values. We use a weight decay of 0.1 and a learning rate of 0.0002 with the AdamW optimizer. Each experiment was executed using a single consumer GPU on a HPC with 8 CPUs and 8GB of RAM.

I RECURRENT NEURAL NETWORKS

I.1 LINEAR AND NONLINEAR PERIODIC WAVE RNN EXPERIMENT DETAILS

For linear RNN, we considered a task where the RNN gets a periodic pulse input (2D delta function of frequencies w_1, w_2 as input x, i.e. $x_k(t) = \delta(\cos(w_k t) - 1), k \in \{1, 2\}$) and learns to generate a 2D cosine wave of the same frequencies, $y_k(t) = \cos(w_k t)$.

For the nonlinear RNN, we designed a two frequency mixing task. Given two source frequencies, the network must generate two cosine waves whose frequencies are the sum and difference of two input frequencies. The network requires non-linearities in order to approximate the multiplication: $\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b), \cos(a - b) = \cos(a)\cos(b) + \sin(a)\sin(b)$. In the task, the RNN receives 2D periodic delta pulse of frequency $\frac{w1+w2}{2}$ and $\frac{w1-w2}{2}$ and learns to generate a trajectory of $\cos(w_1)$ and $\cos(w_2)$.

We use the following recurrent neural network,

$$\mathbf{g}(t+1) = f(\mathbf{W}_{\text{rec}}\mathbf{g}(t) + \mathbf{W}_{\text{in}} + \mathbf{b}_{\text{rec}}) \tag{154}$$

$$\mathbf{y}(t) = \mathbf{R}\mathbf{g}(t) + \mathbf{b}_{\text{out}} \tag{155}$$

in our linear RNNs $f(\cdot)$ is identity whereas in the nonlinear ones we used $ReLU$ activation to enforce positivity condition.

For irrational output frequency ratio case, we used

$$w_1 = p\pi, w_2 = \sqrt{q}, p \sim \mathscr{U}(0.5, 4), q \ \sim \mathscr{U}(1, 10). \tag{156}$$

For rational case, we sampled

$$w_1, w_2 \in [1, 20] \cap \mathbb{Z}. \tag{157}$$

Where $\mathbb{Z}$ is the set of integers. For harmonics:

$$w_1 \in [1, 10] \cap \mathbb{Z}, w_2 = 2w_1. \tag{158}$$

We trained RNN with the trajectory of length $T = 200$, bin size 0.1, and hidden dimension 16. For the linear RNN, we used learning rate 1e-3, 30k training iterations and $\lambda_{\text{target}} = 1$, $\lambda_{\text{activity}} = 0.5$, $\lambda_{\text{positivity}} = 5$, and $\lambda_{\text{weight}} = 0.02$. For the nonlinear RNN, we used learning rate 7.5e-4, 40k training iterations and $\lambda_{\text{target}} = 5$, $\lambda_{\text{activity}} = 0.5$ and $\lambda_{\text{weight}} = 0.01$. In both case, we initialised the weights to be orthogonal and the biases at zero, and used Adam optimiser.

To assess the modularity of the trained RNNs, we performed a Fast Fourier Transform(FFT) on each neuron's activity and measured the relative power of the key frequencies $w1, w2$ with respect to the sum of total power spectrum

$$C_{\text{neuron}_i, w_j} = \frac{|FFT(g_i; w_j)|}{\sum_f |FFT(g_i; f)|} \tag{159}$$

and used it as a proxy of mutual information for modularity metric introduced in eqn. 151.

I.2 NONLINEAR TEACHER-STUDENT RNNS

Network details. The Teacher network has input, hidden, and output dimensions of 2, and has orthogonal recurrent weights, input weights, and output weights. The Student RNN has input dimension 2, output dimension 2, and hidden dimension 64. It is initialised as per PyTorch default settings. The Teacher network dynamics is a vanilla RNN: $\boldsymbol{h}_t = tanh(\boldsymbol{W}_{rec}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{in}\boldsymbol{i}_t)$, and each teacher predicts a target via $\boldsymbol{o}_t = \boldsymbol{W}_{out}\boldsymbol{h}_t$. The Student RNN has identical dynamics (but with a ReLU activation function, and different weight matrices etc).

Generating training data. The teacher RNN generates training data for the Student RNN. We want to tightly control the Teacher RNN hidden distribution (for corner cutting or correlation analyses), i.e., tightly control the source distribution for the training data. To control the distribution of hidden activities of the Teacher RNN, we use the following procedure. 1) We sample a randomly initialised Teacher RNN. 2) $\boldsymbol{h}_0$ is initialised as a vector of zeros. 3) With a batch size of N, we take a **single** step of the Teacher RNN (starting from 0 hidden state) assuming $\boldsymbol{i}_t = 0$. This produces network activations (for each batch), $\boldsymbol{p}_t = tanh(\boldsymbol{W}_{rec}\boldsymbol{h}_{t-1}$. 4) We then sample from idealised distribution of the teacher hidden states, $\boldsymbol{h}_t$. For example a uniform distribution, or a cornet cut distribution. At this point these are just i.i.d. random variables, and not recurrently connected. 5) To recurrently connect these points, we optimise the input to the RNN, $\boldsymbol{i}_t$, such that the RNN prediction, $\boldsymbol{p}_t$, becomes, $\boldsymbol{h}_t$. We then repeat steps 3-5) for all subsequent time-steps, i.e., we find what the appropriate inputs are to produce hidden states as if they were sampled from an idealised distribution. To prevent the Teacher RNN from being input driven, on step 4), we solve a linear sum assignment problem across all batches (N batches), so the RNN (on average) gets connected to a sample ,$\boldsymbol{h}_t$, that is close to its initial prediction, $\boldsymbol{p}_t$. This means the input, $\boldsymbol{i}_t$, will be as small as possible and thus the Teacher RNN dynamics are as unconstrained as possible.

Training. We train the Student RNN on 10000 sequences generated by the Teacher RNN. The learning objective is a prediction loss $|\boldsymbol{o}_t^{teacher} - \boldsymbol{o}_t^{student}|^2$ plus regularisation of the squared activity of each neuron as well as each synapse (both regularisation values of 0.1). We train for 60000 gradient updates, with a batch size of 128. We use the Adam optimiser with learning rate 0.002.

J RNN MODELS OF ENTORHINAL CORTEX

We now talk through the linear RNNs used in Section 5. Our model is inspired by the entorhinal literature and linear network models of it Dorrell et al. (2023); Whittington et al. (2020; 2023b;a). In each trial the agent navigates a 3x3 periodic environment. The RNN receives one special input only at timepoint 0 that tells it the layout of the room (format described later). Otherwise at each timestep it is told which action (north, south, east, west), the agent took. It uses this action to linearly update its hidden state:

$$\boldsymbol{g}_t = \boldsymbol{W}_{a_t}\boldsymbol{g}_{t-1} + \boldsymbol{b}_{a_t} \tag{160}$$

At each timestep it has to output, via an affine readout, a particular target depending on the task.

In the spatial-reward task, Figure 4, at timepoint 0 the agent starts at a consistent position and receives an input that is a 9-dimensional 1-hot code telling the agent the relative position of the reward. At each timestep the agent then has to output two 9-dimensional 1-hot codes, one signalling its position in the room, the other the relative displacement of the reward from its position.

The agent experiences many rooms. Across rooms the rewards are either randomly sampled (random rooms), or sampled from one of two fixed positions (fixed rooms). Each agent experiences a different mixture of randomly sampled or fixed rooms. We then optimise the network to perform the task with non-negative neural activities while penalising the L2 norm of the activity and all weight matrices.

In the mixed-selectivity task, Figure 5, there are three objects in each room and the agent has to report the relative displacement of each of them. The input at the start of each trial is a 27-dimensional code signalling where the three objects are relative to its own position. As such, as the agent moves around the room it has to keep track of the three objects. This task was harder to train so we moved from the mean squared error to the cross-entropy loss and found it worked well, apart from that all details of loss and training are the same.

Between trials we randomise the position of the objects. Some portion (0.8) of the time these positions are drawn randomly (including objects landing in the same position). The rest of the time the objects were positioned so that the first object was one step north-east of the second, which itself was one step north-east of the third. This introduced correlations between the positions of the objects, while preserving their range independence - all objects could occur in all combinations.

We measured the linear NCMI between the neural activity and the 27-dimensional output code and found that each neuron was informative about a single source, as expected, it had modularised Figure 5C. However, pretend we did not know the third object existed. Instead we would calculate the linear NCMI between each neuron and those objects we know to exist. We would still find modular codes for these objects, but we would also find that the neurons that in reality code for third object, due to the correlations between object placements, are actually informative about the first and second object, so they look mixed selective! The position of the objects is always informative about each other, but by conditioning on each object we are able to remove this effect with our metric and uncover the latent modularity. But without knowing which latents to condition on we cannot proceed, and instead get lost in correlations.

K PREFRONTAL MODELLING

Here we present preliminary results using our theory to explain a puzzling difference in monkey prefrontal neurons in two seemingly similar working memory tasks. In both tasks (Xie et al., 2022; Panichello and Buschman, 2021), items are presented to an animal, and, after a delay, must be recalled according to the rules of the task. Similarly, in both tasks, as well as in neural networks trained to perform these tasks (Botvinick and Plaut, 2006; Piwek et al., 2023; Whittington et al., 2023b), the neural representation during the delay period consists of multiple subspaces, each of which encodes a memory of one of the items. Bizarrely, in one task these subspaces are orthogonal to one another (Panichello and Buschman, 2021), a form of modularisation in the representation of different memories sans a preferred basis, while in the other they are not (Xie et al., 2022).

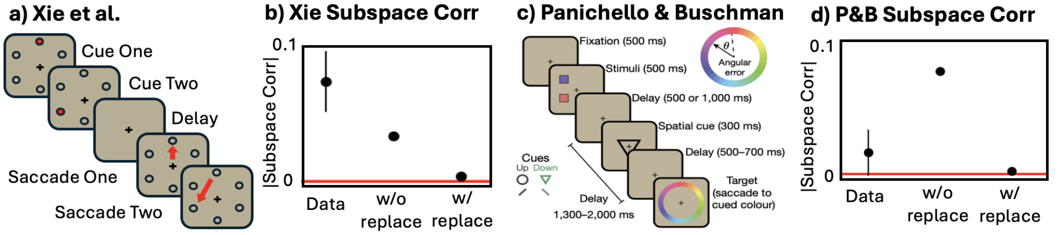


Figure 17: a) Xie et al. (2022) task. b) We estimate the non-orthogonality of subspaces in data, and networks trained on sequences sampled with and without replacement or with correlations. Sampling without replacement, as in the experiment, gives the best fit. c) Panichello and Buschman (2021) task, replicated. d) Subspace correlations for data and models. Sampling with replacement is necessary to fit the data.

In more detail, Xie et al. (2022) train monkeys on a sequential working memory task in which the subject observes a sequence of N dots positioned on a screen then, after a delay, must report that same sequence via saccading to the dot locations in order (Figure 17a). They find that the neural representation in the delay period decomposes into N subspaces (one for each item) that are significantly non-orthogonal. (To calculate the alignment we reanalysed the data using a measure of alignment inspired by Panichello and Buschman (2021), as we find other measures to be biased Appendix L). On the other hand, in a related task, Panichello and Buschman (2021) (P&B) find orthogonal memory. They present monkeys with two coloured squares, one below the other, then, after a delay, present a cue that tells the monkey to recall the top or bottom colour via a saccade to the appropriate point on a colour wheel (Figure 17c). P&B find that, during the delay after cue presentation, the colours are encoded in two subspaces that are orthogonal to one another (Figure 17d).

Why does one experiment result in orthogonality but the other not? Our theory says that differences in range (in)dependence can lead to modular or mixed (orthogonal or not in this case) codes. Crucially, across trials, P&B sample the two colours *independently*, whereas Xie et al. (2022) sample the dots *without replacement*. The latter sampling scheme compromises orthogonality since there are corners (indeed an entire diagonal) missing from the source support.

We model this in biologically-constrained RNNs for each task. For the Xie et al. (2022) task we present two inputs sequentially to a linear RNN with nonnegative hidden state and ask the networks to recall the inputs, in order, after a delay. Interestingly, our results depend on the particular choice of item encoding (one-hot or 2D), in ways that can be understood using our theory (Appendix K.1). To compare to the prefrontal data, we extract the item encoding from the prefrontal data during the delay period for *single-element* sequences. For P&B we assume the encoding is 2D.

Our simulated models recapitulate the major observations of monkey prefrontal representations. Sequences sampled without replacement in the Xie et al. (2022) task lead to aligned encoding subspaces like data (Figure 17b), while colours sampled independently lead to orthogonal encodings in the P&B task as in data (Figure 17d). Conversely, as a prediction for future primate experiments, swapping the sampling scheme swaps the prediction.

That said, there remain some puzzling discrepancies to explore. Panichello & Buschmann find that about 20% of colour-tuned neurons are tuned to the both colours, despite the orthogonal encoding (Panichello and Buschman, 2021), whereas our theory would predict all neurons would respond to one colour. Further, despite a qualitative match (i.e., orthogonal versus non-orthogonal), we do not obtain the exact angles between subspaces in Xie et al. using our RNNs (Figure 17b). We hope to explore these discrepancies further in future work.

K.1 MODELLING DETAILS AND EXTENDED RESULTS

Each RNN is trained on many sequences. For each sequence the RNN sees two cue encodings, sampled from a set of six $\{\boldsymbol{e}_i\}_{i=1}^{6}$. During the first two timesteps the model is shown the two cues, then recieves no further input. Then there is a delay timestep. Finally in the last two timesteps the model must use an affine output to recreate the two encodings. The recurrent dynamics are simple linear systems:

$$\boldsymbol{g}_{t+1} = \boldsymbol{W}_{\text{rec}}\boldsymbol{g}_t + \boldsymbol{W}_{\text{in}}\boldsymbol{e}_{t,i_t} + \boldsymbol{b} \qquad \boldsymbol{e}_{t,i_{t-3}} = \boldsymbol{W}_{\text{out}}\boldsymbol{g}_t + \boldsymbol{b}_{\text{out}} \tag{161}$$

$\boldsymbol{g}_0 = 0$. We train the networks using gradient descent to perfectly solve the task while ensuring the representation is positive. Further, we penalise the L2 norm of the activity and the weights. We used a single set of hyperparameters to generate all data that can be found in the supplementary code.

We played with three different memory encoding schemes, either a 6 dimensional one-hot code, 6 two-dimensional points drawn evenly from the unit circle (like the positions of the dots on the screen shown to the monkey), or the data driven encoding discussed in the Appendix L. Finally, we used two different sequence structures: sampling both sequence elements uniformly, or sampling them without replacement but otherwise uniformly, as in the experiments.

We analysed the resulting delay period activity using the techniques discussed in Appendix L. We found, first, that in all cases, the delay period activity for networks trained on sequences sampled with replacement comprised two orthogonal subspaces. Second, depending on the encoding scheme, some networks trained on sequences sampled without replacement were orthogonal, others aligned, and the degree of alignment depending on the hyperparameter weighting of the activity vs. weight loss. Third, in both sequence settings we also found that the subspace encoding the first memory was a small amount bigger than the second, matching data. We now talk through some of these results in more detail. First, of the three encoding schemes we find that the 2D data is orthogonalised when sampled without replacement (Fig 18), while the one-hot (not shown) and data driven codes do not (Figure 17B). It was this puzzling discrepancy between the behaviour of the one-hot and 2D codes that led us to the data-driven encoding. This behaviour can be somewhat understood by looking at the joint distribution of the encoded memories, figure 19. The required inequality constraint is not satisfied, while it is for 1-hot memories. The neural data inspired encoding presumably lies closer to the 1-hot end of the spectrum.

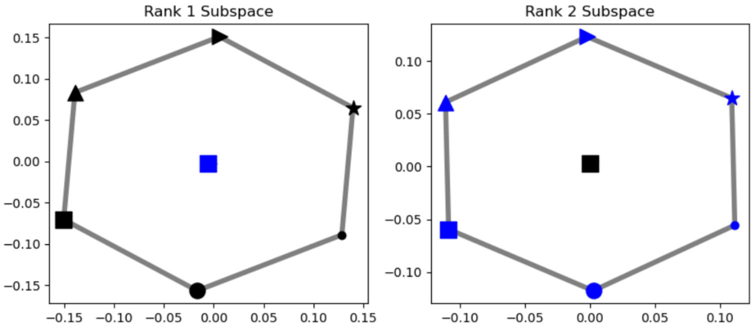


Figure 18: We estimate the two-dimensional subspaces encoding, the two subspaces are basically orthogonal, the circular mean of the 2 principal angles between the subspaces is $89.95°$

We return to the alignment of correlated sequences after discussing the similar results from the Panichello & Buschmann studies.

K.1.1 PANICHELLO & BUSCHMANN, MODEL, AND EXTENDED RESULTS

We study a very similar linear RNN setting, the loss, training protocol, and sequence structures are mostly identical to the previous section. We list the differences. First, we assume the memory encoding is the equivalent of the 2D one from the previous section.

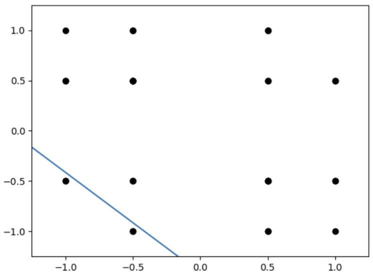


Figure 19: Left: We plot the joint distribution of one dimension of each of the memories against one another for the 2D coding. The x axis is cosine of the first sequence element, the y axis is the second. Sampling without replacement removes the two extreme corner points (but not the whole diagonal, as though this dimension of the code might be equal, the memory is 2D, and the other dimension can be different). The line marks one of the inequality constraints that can be broken to induce mixing. As can be seen, removing one data point is not sufficient to remove all data below this line, so the code modularises. Right: The converse is true for one-hot variables.

Second, network architecture: rather than providing inputs to the linear network, we let the network learn the input by allowing it to initialise itself differently depending on the pair of stimuli. This was an implementational choice that doesn't constrain the input to be a linear function of the encodings. Nonetheless, the network learns to initialise itself as a linear function of the two presented cues. We could equally have provided the two cues as inputs as in the previous section, it would likely have changed little.

Third, we implement the different cues by updating the initial representation with different matrices. If the animal is cued to attend to the top stimulus the initial activity, which we think of as the delay period activity, $\boldsymbol{g}_0(c_{\text{top}}, c_{\text{bottom}})$ for two colours c_{top} and c_{bottom}, is updated using one matrix, which then drives the readout:

$$\boldsymbol{W}_{\text{top}}\boldsymbol{g}_0(c_{\text{top}}, c_{\text{bottom}}) + \boldsymbol{b}_{\text{top}} = \boldsymbol{g}_1(c_1, c_2) \quad \boldsymbol{R}\boldsymbol{g}_1(c_{\text{top}}, c_{\text{bottom}}) + \boldsymbol{b}_{\text{out}} = \boldsymbol{e}_{c_{\text{top}}} \tag{162}$$

And the same for the bottom matrix. This is inspired by recent work that has shown these type of action dependent matrices are both mathematically tractable, and biologically plausible (Logiaco et al., 2021; Dorrell et al., 2023; Whittington et al., 2020; 2023b;a).

Fourth, Number of Samples: in the real experiment Panichello & Buschman present colours drawn from a continuous colour wheel. They then analyse it by binning the colour wheel into groups of four. We skip straight to the binned colours, and pretend the encoded cues that the network predicts are 4 points sampled evenly from the unit circle. Other than this change the sampling schemes are the same.

We extract the subspaces and their alignments as discussed in Appendix L. We make some final interesting (to us!) observations. First, the top and bottom colour subspaces are equally sized, unlike in Xie et al., likely because they are equally important, whereas in the Xie task one stimuli has to be recalled before the other, making it, temporally, more important. Second, since we only sample 4 'colours' from the circle, rather than the 6 in Xie et al., the prediction shown in figure 19 changes. With four data points even 2D encodings are missing the appropriate corner 20!

This matches the finding presented in Figure 17, that when sampled without replacement the subspaces align, since a sufficiently large corner is missing.

Finally, there remain discrepancies between the neural data and our theory and models. Most fundamentally, both theory and models predict that if the subspaces are orthogonal they should be encoded by disjoint groups of neurons, if they align, they should not. Yet, both Xie and Panichello and Buschman find neurons tuned to both cues. This is expected for Xie's data, but not for Panichello and Buschman's, where they find roughly 20% of colour tuned neurons are mixed selective. We do not yet have a good understanding of this discrepancy. More superificially, we are not able to fit the exact numerical value of the alignment between subspaces in Xie et al.'s data, and we remain unsure whether this could be done with different hyperparameters or a noise model. We leave this for further exploration in a more targeted venue.

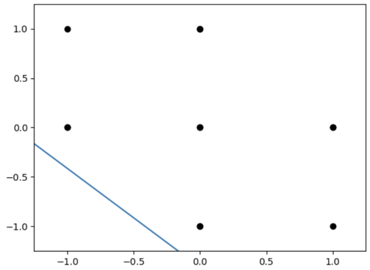


Figure 20: As in 19, we plot the joint distribution of one dimension of each of the memories against one another for the 2D coding. The x axis is cosine of the first sequence element, the y axis is the second. Since there are now only 4 settings of the angle we can see sufficient corner has been removed to predict mixing, as we see in simulations Figure 17.

L NEURAL DATA ANALYSIS

We were kindly given the data of Xie et al. We used it to generate Figure 17B. In this section we talk through our analysis techniques. First, however we discuss methods for measuring subspace alignment.

L.1 SUBSPACE ALIGNMENT METRICS

Previous works have calculated a single angle between subspaces, most rigorously the first principal angle (Xie et al., 2022), alternatively the unique angle between the two subspaces after projecting to a three dimensional PC space. We found that these angle based methods were biased for two reasons.

First, two N dimensional subspaces have N principal angles, and these are ordered, the first principal angle is smaller than the second, which is smaller than the third, etc. This ordering means that noise added to a representation will bias the estimates of the first principal angles down, and the last ones up. To show this we created data that lives in two planes oriented at 72 degrees. We then add guassian noise and find that this biases the estimate of the first principal angle down, and the second up, despite the fact that they are both 72, Figure 21.

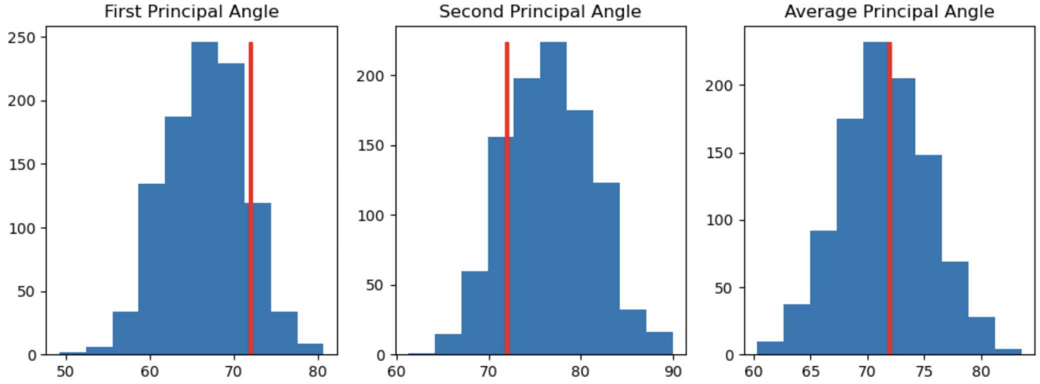


Figure 21: We create some fake data of two 2D subspaces that are aligned at 72°, i.e. both of the two principal angles are 72°. We then add a small amount of noise 1000 times and calculate the principal angles between the noisy vectors (all the vectors are length one, and I add a zero-mean gaussian noise matrix with variance 0.1). As you can see, the estimate of the first principal angle is biased down, the second up, but the average appears unbiased.

Second, without further thought, angles are bounded, since they are necessarily smaller than 90. If your true data is oriented close to 90 degrees that means that noise will more likely push your estimate away from 90 than towards it. To show this effect we create another noisy dataset at 85 degrees, and show the noise pushes all estimates, even the mean of the two angles, significantly below 85, Figure 22.

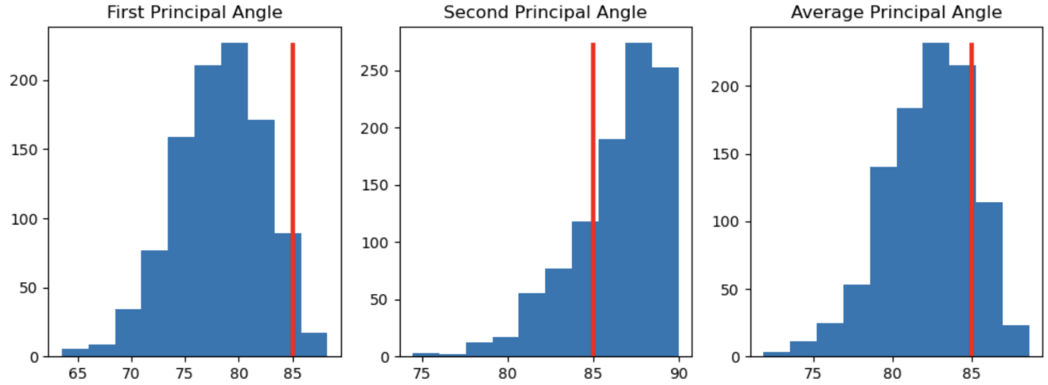


Figure 22: Same setting as figure 21, except now the true principal angle is 85°. We can see that all estimators of the average principal angle are shifted downwards.

We counteract both these effects by using a single metric, whose bounds are nowhere near the ranges of interest. Specifically we use the correlation metric from Panichello & Buschmann. Given a set of vectors from the two subspaces of interest: $\{\boldsymbol{g}(c,s)\}_{c=1,s=1}^{C,2}$, indexed by subspace s and condition c (for example, which cue was presented), the correlation is:

$$\rho = \langle \mathrm{corr}(\boldsymbol{g}(c,1) - \langle \boldsymbol{g}(c',1)\rangle_{c'}, \boldsymbol{g}(c,2) - \langle \boldsymbol{g}(c',2)\rangle_{c'}\rangle_c \tag{163}$$

Where the correlation is over neurons. This is 0 when subspaces are orthogonal, and avoids both pitfalls. Panichello & Buschman reported this metric on their data. We therefore move to extract the same metric from the data of Xie.

L.2 EXTRACTING CODING SUBSPACES FROM SAMPLED WITHOUT REPLCEMENT SEQUENCES

We analyse the firing rates that came pre-extracted from calcium imaging by Xie et al. We study the delay period representation of the two-sequence task. Call the neural representation of a two element sequence during the delay period $\boldsymbol{g}(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix})$. We make the assumption (supported by data (El-Gaby et al., 2023; Xie et al., 2022; Panichello and Buschman, 2021) and simulations (Botvinick and Plaut, 2006; Whittington et al., 2023b)) that the data decomposes into subspaces encoding each sequence element: $\boldsymbol{g}(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}) = \boldsymbol{g}_1(\theta_1) + \boldsymbol{g}_2(\theta_2) + \boldsymbol{c}$, where $\boldsymbol{g}_i(\theta)$ denotes the activity in subspace i when the i'th element of the sequence is θ. If the sequences are sampled with replacement the two sequence elements are uncorrelated and you can find $\boldsymbol{g}_1(\theta_1)$ up to a constant offset by just averaging over the other sequence element, $\boldsymbol{g}_1(\theta_1) = \langle \boldsymbol{g}(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix})\rangle_{\theta_2} = \boldsymbol{g}_1(\theta_1) + \langle \boldsymbol{g}_2(\theta)\rangle_\theta + \boldsymbol{c}$, where $\langle\rangle$ denotes averaging. Performing PCA on the set of $\{\boldsymbol{g}_1(\theta_i)\}_{i=1}^{Q}$ then gets you a perfect estimate of the subspace containing information about θ_1, as it removes the shared constant offset. This process does not work if the sequences are sampled without replacement, as you cannot perform the $\langle \boldsymbol{g}_2(\theta)\rangle_\theta$ average.

Xie et al. get around this problem by doing regularised linear regression. To remove any potential hyperparameter dependence we employ a novel difference scheme. Consider the following term:

$$\Big\langle \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\Big)\Big\rangle_{\theta_2} \tag{164}$$

Equal to:

$$\frac{Q-2}{Q}\Big\langle \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\Big)\Big\rangle_{\theta_2\neq\theta_1,\theta_1'} + \frac{1}{Q}\Big(\underbrace{\boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_1\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_1\end{bmatrix}\Big) + \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_1'\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_1'\end{bmatrix}\Big)}_{\text{Error Term}}\Big) \tag{165}$$

Again, the left hand side is what we want, the first term on the right is something we can get, and the leftover part is going to be ignored, our error term. Why have we chosen to leave this particular term out? Using the decomposition property of our representations we can see that the thing we want to estimate is:

$$\Big\langle \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\Big)\Big\rangle_{\theta_2} = \boldsymbol{g}_1(\theta_1) - \boldsymbol{g}_1(\theta_1') \tag{166}$$

Which makes sense, it is the difference in subspace 1 between the encoding of these two stimuli. But now we can analyse the error term and see that it is equal to the same thing!

$$\boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_1\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_1\end{bmatrix}\Big) + \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_1'\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_1'\end{bmatrix}\Big) = 2(\boldsymbol{g}_1(\theta_1) - \boldsymbol{g}_1(\theta_1')) \tag{167}$$

Hence, plugging these into equation 164:

$$\Big\langle \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\Big)\Big\rangle_{\theta_2\neq\theta_1,\theta_1'} = \boldsymbol{g}_1(\theta_1) - \boldsymbol{g}_1(\theta_1') \tag{168}$$

So we can see that this object is a good estimator of this quantity!

This is therefore the quantity we estimate. For all different stimuli pairs θ, θ', we build an estimate of the difference vectors. We then use these vectors to calculate the correlation alignment measure 163, this result is reported in Figure 17B. We form error bars by subsampling by half the number of neurons 200 times and estimating the mean and variance over 200 simulations.

L.3 EXTRACTING ENCODINGS FROM SINGLE-ELEMENT 'SEQUENECE' DATA

Finally, we need to choose how to encode the memories in the Xie task for our RNN models to use, see Appendix K.1. We take a data driven approach and estimate the monkey's internal representation of each cue by analysing the trials in which the only a single stimulus was presented. We average these trials to create six encoding vectors, then find the dot product similarity matrix of this data. Finally, we find a set of 5 dimensional encodings that exactly recreate this data dot product structure. These are the vectors we feed our RNN: they have the same similarity structure as the monkey's representation, without increasing training time by introducing a neuron-dimensional weight matrix.