Modularity in Biologically Inspired Representations Depends on Task Variable Range Independence

Anonymous Authors[1]

Abstract

Artificial and biological neurons sometimes modularise into disjoint groups each encoding a single meaningful variable; at other times they entangle the representation of many variables. Understanding why and when this happens would both help machine learning practitioners build interpretable representations and increase our understanding of neural wetware. In this work, we study optimal neural representations under the biologically inspired constraints of nonnegativity and energy efficiency. We develop a theory of the necessary and sufficient conditions on task structure that induce neural modularisation of task-relevant variables in both linear and partially nonlinear settings. Our theory shows that modularisation is governed not by statistical independence of underlying variables as previously thought, but rather by the independence of the ranges of these variables. We corroborate our theoretical predictions in a variety of empirical studies training feedforward and recurrent neural networks on supervised and unsupervised tasks. Furthermore, we apply these ideas to neuroscience data, providing an explanation of why prefrontal working memory representations sometimes encode different memories in orthogonal subspaces, and sometimes don't, depending on task structure. Lastly, we suggest a suite of surprising settings in which neurons might be or appear mixed selective without requiring complex nonlinear readouts, as in traditional theories. In summary, our theory prescribes precise conditions on when neural activities modularise, providing tools for inducing and elucidating modular representations in machines and brains.

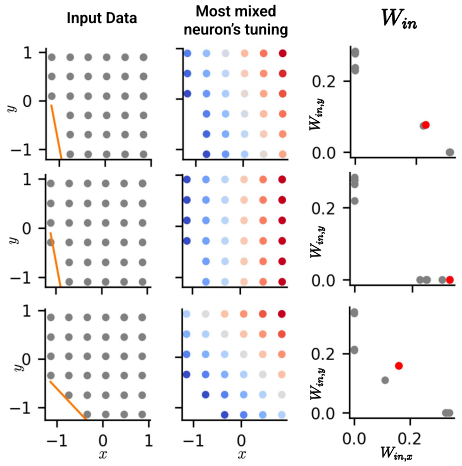

Figure 1: Simulations of our theory (Theorem 2.1) for three cases (rows) of linearly autoencoding two source variables under biologically inspired constraints (Eq. 1, Eq. 2). We remove particular datapoints from a corner of a uniform grid dataset (left) and use our inequality condition (Eq. 3) to predict whether an optimal representation should modularise. The middle row's dataset satisfies the condition: graphically, the dataset has support outside of a critical line. This results in modularisation: each neuron's activity only varies with one source variable (middle: one example neuron), and the neurons' input weight vectors are all one-hot (right). In contrast, the top and bottom rows' datasets violate the condition. This results in mixing, as shown by the most mixed neuron's activity (middle) and the existence of dense weight vectors (right). The precision of our theory is illustrated in how the addition of a single datapoint outside the line causes the optimal representation to shift from being mixed to being modularised (top vs. middle).

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 & 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, though not all, 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;

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Tye et al., 2024). The modules vs. mixtures duality has recently been reprised in the machine learning community. Both the mechanistic interpretability and disentangled representation learning 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 showing clear cases of mixed tuning. This brings us to the main research question considered in this work: Why are neurons, artificial and biological, sometimes modular and sometimes mixed selective?

This question has been explored in machine learning from two perspectives. 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).

In neuroscience, theories prompted by the recognition of neural mixed selectivity have argued that these 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). Finally, linear representations of statistically independent variables have been shown to modularise when constrained to exhibit the biologically inspired properties of nonnegativity and energy efficiency (Whittington et al., 2023b). However, an understanding of when modularisation may occur when variables are dependent is lacking.

In this work, we show that modularisation of representations trained under biologically inspired constraints is governed by more than just statistical independence. In particular, it is the independence of the *ranges* of the variables that facilitates modularisation, similar to the independent support property that has been investigated in the disentanglement literature (Roth et al., 2023). The shift from statistical to range independence enables an understanding of when (linear) recurrent neural network (RNN) representations of dynamic variables modularise, in the same way we can understand when representations of static variables modularise. Empirically, these results generalise to the nonlinear setting: we show that our range (in)dependence conditions predict modularisation in nonlinear feedforward networks on supervised and autoencoding tasks as well as nonlinear RNNs on supervised and distillation tasks. Furthermore, we generalise our theory slightly to nonlinear-encoding/linear-decoding representations. Finally, we apply our theory to neuroscience data. First, we provide an explanation of why working memory representation in prefrontal cortex might sometimes represent memories in orthogonal subspaces, and sometimes not. Second, we highlight a variety of settings in which neurons can be mixed selective without any need for flexible nonlinear categorisation, thus offering a different spin on the debate over the particular computational benefits of modular vs. mixed selective neurons.

In summary, our work contributes to the growing understanding of neural modularisation by highlighting some subtle determinants of modularisation and explaining puzzling representational observations from both neural networks and the brain in a cohesive normative framework.

2. Precise Constraints Governing the Modularisation of Linear Autoencoders

We begin by studying a linear autoencoding setting in which we can fully characterise the modularity of biologically inspired representations. We study nonnegative activities to match the nonnegativity of both biological neural activities and the ReLU activation function commonly used in machine learning. Similarly, we use an L2 energy regularisation on weight and activities to match biological energy constraints (Harris et al., 2012) and the simplicity bias of weight regularisation (Krogh & Hertz, 1991). We then obtain our governing equation of whether modularity is optimal, which will later guide our nonlinear experiments and neuroscience applications.

Theorem 2.1. *Let $\boldsymbol{s} \in \mathbb{R}^{d_s}$ be a vector of d_s scalar source variables (sources) which are linearly encoded and decoded from a nonnegative representation $\boldsymbol{z} \in \mathbb{R}^{d_z}$ using input and output weights and biases $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{d_z \times d_s}, \boldsymbol{b}_{\text{in}} \in \mathbb{R}^{d_z}, \boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d_s \times d_z}$, and $\boldsymbol{b}_{\text{out}} \in \mathbb{R}^{d_s}$, with $d_z \geq d_s$:*

$$\boldsymbol{z}^{[i]} = \boldsymbol{W}_{\text{in}}\boldsymbol{s}^{[i]} + \boldsymbol{b}_{\text{in}}, \;\; \boldsymbol{s}^{[i]} = \boldsymbol{W}_{\text{out}}\boldsymbol{z}^{[i]} + \boldsymbol{b}_{\text{out}}, \;\; \boldsymbol{z}^{[i]} \geq 0 \quad (1)$$

for all samples indexed by i. Consider the minimization of the activity and weight energy subject to the above constraints:

$$\left\langle ||\boldsymbol{z}^{[i]}||_2^2 \right\rangle_i + \lambda \left(||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2 \right). \quad (2)$$

At the minima of the constrained optimization, each column 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 $\boldsymbol{w} \in \mathbb{R}^{d_s}$:

$$\min_i[\boldsymbol{w}^\top \boldsymbol{s}^{[i]}] < \sqrt{\sum_{j=1}^{d_s}\left(w_j \min_i \bar{s}_j^{[i]}\right)^2 - \sum_{j,j'\neq j}^{d_s} w_j w_{j'} \left\langle \bar{s}_j^{[i]} \bar{s}_{j'}^{[i]} \right\rangle_i}, \quad (3)$$

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

Proof. We outline a sketch proof and defer a full treatment to App. A. Intuitively, modularisation is driven by the interaction of nonnegativity and the activity loss $\langle ||\boldsymbol{z}^{[i]}||_2^2 \rangle_i$. We can disregard the weight loss since it is minimised by modularising (App. A.2.1). Now consider two modular neurons, each encoding a single source, $z_1(s_1^{[i]})$ and $z_2(s_2^{[i]})$. Since neural firing is nonnegative and we want to minimise it, these encodings must be chosen so that $\min_i z_1(s_1^{[i]}) = 0$ (else shift downwards and improve the objective). Under these constraints, a simple summing of encodings into a putative mixed neuron will always be worse than modularising: $\langle (z_1(s_1^{[i]}) + |v| z_2(s_2^{[i]}))^2 \rangle_i \geq \langle (z_1(s_1^{[i]}))^2 \rangle_i + v^2 \langle (z_2(s_2^{[i]}))^2 \rangle_i$[1]. However, it is possible to merge and save for particular source co-range properties: when $\min_i[z_1(s_1^{[i]}) + |v| z_2(s_2^{[i]})] > \min_i[z_1(s_1^{[i]})] + \min_i[z_2(s_2^{[i]})] = 0$ (e.g. whenever $z_1(s_1^{[i]})$ is low, $z_2(s_2^{[i]})$ is high). In this case, the activity of the mixed neuron can be shifted down while preserving nonnegativity, and gaining efficiency. The tradeoff between this energy saving and the inherent cost of mixing determines whether the representation modularises. The above inequalities make this argument precise. □

Our theory prescribes a set of inequalities that determine whether the representation, $\boldsymbol{z}$, is modular (see App. A for details). If a single inequality is broken, the representation is mixed; else, the representation is modular: each neuron's activity is a function of a single source. These inequalities depend on two properties of the sources. First, the pairwise correlations. Second, the source co-range properties, i.e., does knowing the value of s_1 constrain the minima or maxima of possible values of s_2? Remarkably, the inequality conditions do not depend on the hyperparameter, λ. To help interpret these results, we consider a particularly clean specialisation.

Theorem 2.2. *In the same setting as Theorem 2.1, if $\left|\min_i \bar{s}_j^{[i]}\right| = \max_i \bar{s}_j^{[i]}$ $\forall j \in [d_s]$, i.e. each source is range-symmetric, then the optimal representation modularises if all sources are pairwise extreme-point independent, i.e.*

$$\min_i \left[s_j^{[i]} \middle| s_{j'}^{[i]} \in \left\{ \max_{i'} s_{j'}^{[i']}, \min_{i'} s_{j'}^{[i']} \right\} \right] = \min_i s_j^{[i]} \quad (4)$$

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

In other words, if the joint distribution has non-zero support on all extremal corners, the representation will modularise. Conversely, if this property does not hold, there exists a correlation value that will induce mixing (App. A). Extreme-point independence is an extreme setting; in general there might be some corner missing in the joint distribution. These missing-cornered sources correspond to those in the sketch proof for which mixing can increase the minima, leading to lower energy costs while preserving nonnegativity. Chop off a large enough corner, and the representation mixes. Theorem 2.1 tells us precisely how large each corner type must be to cause mixing.

Theory validation in linear autoencoders. We test our theory on linear autoencoders. Specifically we use our inequalities to design datasets that are near the modular-mixed boundary, and show the theory correctly predicts which modularise. Indeed we are able to create modular and mixed datasets that differ by just a single data point (Fig. 1).

Theoretical predictions. From our theory we extract qualitative trends to empirically test in more complex settings. (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 particular, predictable, corner slices 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 of 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, and compare a limited set of nonlinear-encoding/linear-decoding representations to theoretical predictions, finding promising agreement.

Metrics for representational modularity and inter-source statistical dependence. To quantify the modularity of a 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,

[1] We chose $|v|$ to ensure $z_1(s_1^{[i]}) + |v| z_y(y) > 0$

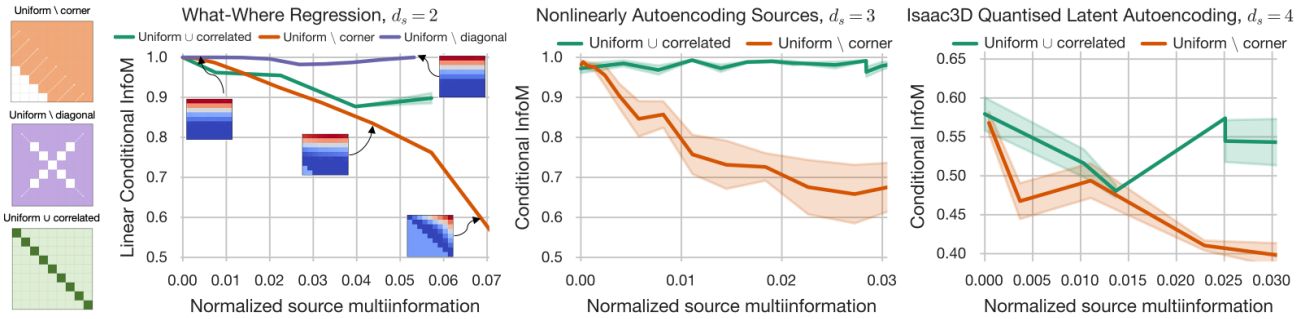

Figure 2: Impact of data distribution on modularisation in nonlinear feedforward networks processing low-dimensional and image data. Across three tasks, uniform \ corner significantly reduces modularity (orange), while uniform ∪ correlated (green) and uniform \ diagonal (purple) preserve it. The visual representations of dropout methods are shown on the left, and each task's appropriate conditional InfoM metric (App. E) is plotted against normalised source multiinformation. Shaded regions denote standard error of the mean.

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 possibility, in cases where there is significant multiinformation in the sources, that a neuron is only informative about a source because it is encoding a different source. We report normalized source multiinformation as a measure of inter-source statistical dependence. We defer further exposition of metrics to App. E.

What-where task. Inspired by the modularisation of what and where into the ventral and dorsal visual streams, we ask nonlinear networks with a single hidden layer to report where and which pixel is on in a simple binary image, producing two outputs, each an integer between one and nine (though one-hot labels, or more complex shapes also work, App. F). We L2 regularise activity and weight energies and satisfy nonnegativity using a ReLU activation function (details: App. F). If what and where are independent from one another, for example both uniformly distributed, then under our biological constraints (but not without them, App. F) 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, Fig 2: cutting corners from the co-range 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 their range independence introduces less mixing when compared to the corner cutting that induces the same amount of mutual information between what and where. Visual depictions of the source distributions and results are presented in Fig. 2, left.

One striking feature of the modular representations is that every neuron is tuned to linearly encode a half-plane (Fig. 2). We build a nonlinear version of our theory and study the most energy-efficient linearly-decodable nonlinear tuning curve. Remarkably, we find that this is a bias-free ReLU (App. D), matching the empirical findings. We return to this result and its neuroscience implications in Section 6.

Nonlinear autoencoding of sources. Next, we study a nonlinear autoencoder trained to autoencode multidimensional source variables. Again we find that under biological constraints (but not without G), independent source variables modularise, corner cutting induces mixing, but equivalently entangling sources while preserving range independence does not induce mixing (Fig. 2, middle). Finally, as in the what-where networks, we find latent neurons that each encode half a source, suggesting the theoretical results generalise somewhat to regularised nonlinear decoders.

Disentangled representation learning of images. Finally, 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 quantized, introducing a privileged basis with low channel capacity. In our biological 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). We find the same qualitative patterns emerge: corner cutting is a more important determinant of mixing than range-independent correlations (Fig. 2).

4. Modularisation in Recurrent Neural Networks

We now turn to RNNs. We study an analogous nonnegative linear RNN subject to activity and weight regularisation. Linear dynamical systems can only autonomously implement exponentially growing or decaying frequencies, so we consider under what conditions our biologically inspired

constraints lead to modularisation of dynamical modules, such as oscillations at different frequencies.

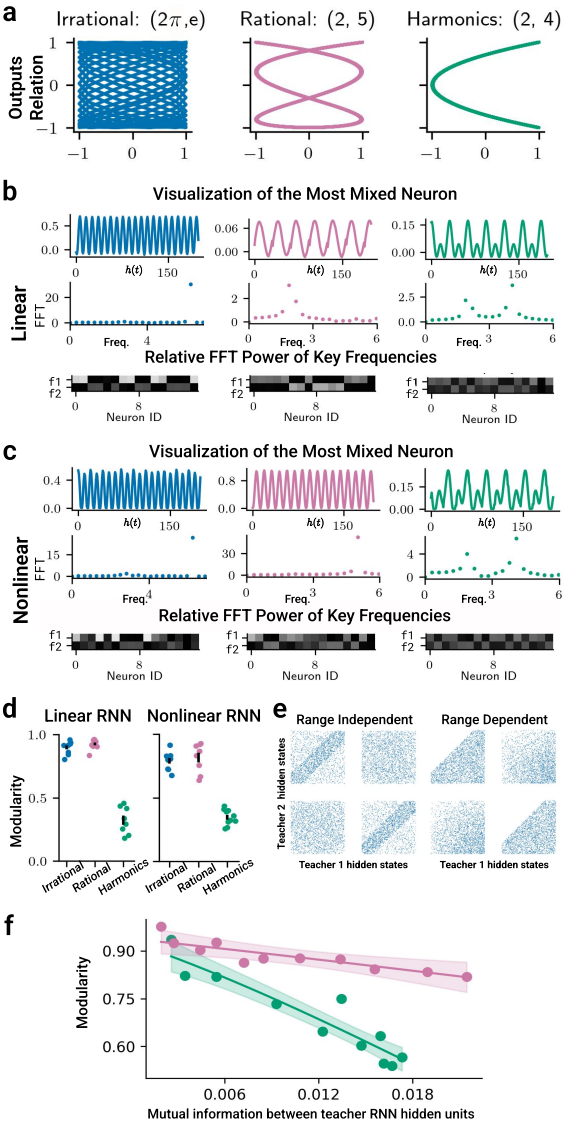

Figure 3: a) Frequencies pairs and their co-ranges. b) The most mixed neuron, its Fourier spectrum, and the population's Fourier spectrum in the two task-related frequencies show clear modularisation in irrational and rational, but not harmonic cases. c) Same as (b), but with *nonlinear* RNNs trained on frequency mixing tasks. d) Modularity score for RNNs trained on 10 frequency pairs. (e) Plot of joint distribution of hidden state activity for two 2-dimensional teacher RNNs. f) Trends of modularity scores of student network in range dependent and independent cases qualitatively agree with our theoretical results.

We develop an analogous version of the theory that tells us that the same inequalities can be used to predict the modularisation of dynamical motifs by just replacing the sources in the feedforward case with the appropriate dynamical variables, such as $\cos(\omega t)$ (App. B). This results in some surprising predictions. Consider a nonnegative, energy efficient, linear RNN, trained to produce two frequencies:

$$\boldsymbol{z}_{t+1} = \boldsymbol{W}_{\text{rec}}\boldsymbol{z}_t + \boldsymbol{b}_{\text{rec}}, \quad \boldsymbol{W}_{\text{out}}\boldsymbol{z}_t + \boldsymbol{b}_{\text{out}} = \begin{bmatrix}\cos(\omega_1 t)\\ \cos(\omega_2 t)\end{bmatrix}. \quad (5)$$

When should the network modularise one frequency from the other? In order to produce these frequencies, the network will oscillate at both frequencies:

$$\boldsymbol{z}_t = \boldsymbol{b}_z + \sum_{i=1}^{2} \left(\boldsymbol{a}_i \cos(\omega_i t) + \boldsymbol{b}_i \sin(\omega_i t)\right). \quad (6)$$

If ω_1 is an irrational multiple of ω_2, then after enough time $\cos(\omega_1 t)$ and $\cos(\omega_2 t)$ will be extreme point independent (Fig. 3a, left), and the theory predicts their representation should modularise. Conversely, if ω_1 is a harmonic of ω_2 or vice versa (Fig. 3a, right) then the co-range is missing enough of a corner to break at least one inequality, and the representation mixes. Most surprisingly, if ω_1 is a rational but non-harmonic multiple of ω_2 ($\omega_1 = \frac{n}{m}\omega_2$ for integers $n > 1$ and $m > 1$), the variables are range dependent, but no sufficiently large corner is missing (Fig. 3a, middle), so their representation should modularise (App. A.6.2)! Each of these results is confirmed in linear network simulations (Fig. 3b; App. H). We now show these results generalise to nonlinear RNNs in two steps.

Nonlinear frequency RNNs. We train nonlinear ReLU RNNs with biologically inspired constraints to perform a frequency mixing task. 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:

$$\boldsymbol{z}_{t+1} = \text{ReLU}\left(\boldsymbol{W}_{\text{rec}}\boldsymbol{z}_t + \boldsymbol{W}_{\text{in}}\begin{bmatrix}P_{\omega_1}(t)\\ P_{\omega_2}(t)\end{bmatrix} + \boldsymbol{b}_{\text{rec}}\right) \quad (7)$$

$$\boldsymbol{W}_{\text{out}}\boldsymbol{z}_t + \boldsymbol{b}_{\text{out}} = \begin{bmatrix}\cos([\omega_1 - \omega_2]t)\\ \cos([\omega_1 + \omega_2]t)\end{bmatrix}. \quad (8)$$

Exactly the same range-dependence properties, but applied to the frequencies $\omega_1 - \omega_2$ and $\omega_1 + \omega_2$, determine whether or not the network modularises (Fig. 3c): 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 theory predictions of when RNNs modularise, but in settings more realistic than pure frequencies, we generate training data trajectories from two randomly initialised teacher RNNs with tanh activation functions, and then train a student RNN (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 App. H.2 for details). Using carefully chosen inputs at each timestep, we are able to precisely control the distribution of teacher RNN hidden state activity (i.e., the source distribution). For example we can increase correlations/statistical non-independence of the hidden states, while also either maintaining or breaking range independence (Fig. 3e). This allows us to characterise the settings in which the student RNN learns a modular representation (Fig. 3f). Indeed, we observe that when range independence is maintained, the student learns a modular representation regardless of the statistical dependence of the teacher RNNs. Conversely, the student RNN does not modularise when the teacher RNNs become increasingly range dependent. Both features are exactly as our theory predicts and suggest that our results apply in settings more general than the ones in which we have proven them.

5. Modular Prefrontal Working Memory

We now apply our results to neuroscience to explain a puzzling difference in monkey prefrontal neurons in two seemingly similar working memory tasks. In both tasks (Xie et al., 2022; Panichello & 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 & Plaut, 2006; Piwek et al., 2023; Whittington et al., 2023a), 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 & 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).

In more detail, Xie et al. (2022) train monkeys on a sequential working memory task where the animal must observe a sequence of N dots positioned on a screen, then after a delay report that same sequence via saccading to the dot locations in order (Fig. 4a). They find that the neural representation in the delay period decomposes into N subspaces (one for each item) that are significantly non-orthogonal. On the other hand, Panichello & Buschman (2021) (P&B) find orthogonal memory encodings in different tasks. 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 (Fig. 4c). P&B find that, during the delay after cue presentation, the colours are encoded in two subspaces that are orthogonal to one another (Fig. 4d).

Before answering the puzzle of why these two highly related working memory tasks are encoded differently in monkey prefrontal cortex, we first verify that the subspaces

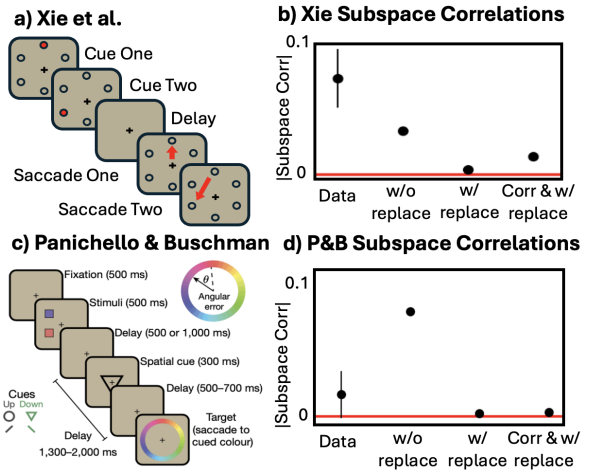

Figure 4: a) Xie et al. (2022) task. b) We estimate the non-orthogonality of subspaces in data, and networks trained on sequences sampled with and without replacement or with correlations (Apps. I and J). Sampling without replacement, as in the experiment, gives the best fit. c) Panichello & Buschman (2021) task, replicated. d) Subspace correlations for data and models. Sampling with replacement is necessary to fit the data.

of Xie et al. (2022) are truly non-orthogonal (as the original analysis uses a biased estimate of subspace alignment – see App. I) with a measure of subspace correlation originally used by P&B. Briefly, this finds matched pairs of vectors from each subspace, such as the vector encoding a green cue in the upper and lower subspaces, and calculates the average correlation across neurons over these pairs. When this is zero as in P&B (Fig. 4d, Data), the subspaces are orthogonal; whereas in (Xie et al., 2022) it is non-zero (Fig. 4b, Data), signalling non-orthogonality.

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*. This latter sampling leads to range dependence since there is a corner missing from the co-range of the two memories. This range dependence results in alignment.

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 (App. J). To compare to the prefrontal data, we extract the item encoding from the prefrontal data during the delay period for *single-element* sequences. Without access to P&B's neural data, we assume the encoding is 2D.

Our simulated models recapitulate the major observations

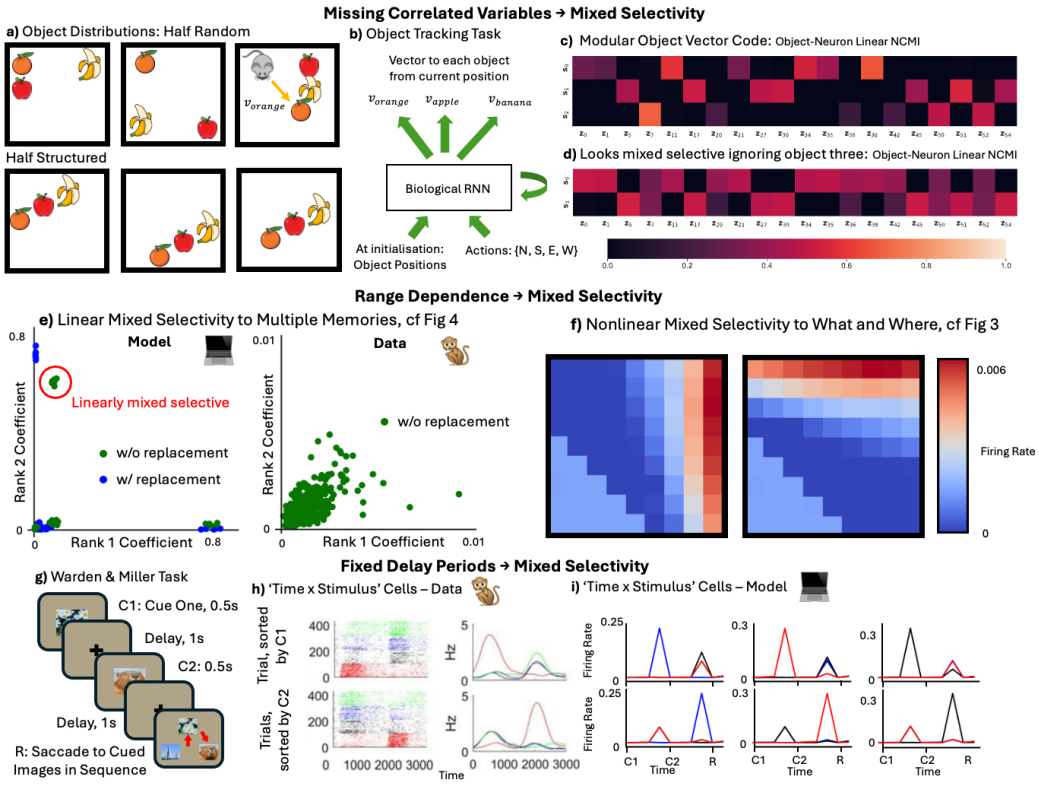

Figure 5: a-b) We train linear RNNs to report displacement to three objects as an agent moves within many rooms. In each room, object instances are either completely random (a, top) or clustered in a random line (a, bottom); object positions are therefore range independent but correlated. c) The neurons 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 are purely encoding the disregarded object appear mixed selective due to the statistical dependencies between objects. e) We extract the tuning of neurons to different memory encodings from both RNNs (left) and neural data (right) (Xie et al., 2022). In both the models and data, many neurons are linearly mixed selective. f) In nonlinear, but linearly decoded, representations from the what-where task, many neurons are nonlinearly mixed selective, with ReLU tuning curves predicted by theory D. g) Warden & Miller (2010) train monkeys to see, then report a sequence of two images. Mikkelsen et al. (2023) reanalyse the data and find h) neurons tuned to both time and stimulus identity. i) Our linear RNN model recapitulates this.

of monkey prefrontal representations. Sequences sampled without replacement in the Xie et al. (2022) task lead to aligned encoding subspaces like data (Fig. 4b), while colours sampled independently lead to orthogonal encodings in the P&B task as in data (Fig. 4d). Conversely, as a prediction for future primate experiments, swapping the sampling scheme swaps the prediction (Fig. 4b, d). Finally, in each case, changing the dataset to induce correlations between memories while preserving range independence leads to weaker effects for the same amount of induced cross-memory mutual information (App. J).

6. The Mixed Origins of Mixed Selectivity

Why neurons might be mixed selective is a matter of debate in neuroscience (Barack & Krakauer, 2021; Tye et al., 2024). Theories of *nonlinear* mixed selectivity have argued that, analogously to a nonlinear kernel, such schemes permit linear readouts to decode nonlinear task functions of the sources. 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 including nonlinear functions of the sources.

Missing variables. First, a purportedly mixed selective neuron could actually be a modular encoding, of a variable unknown to the experimenter. For example, entorhinal cortex is home to many modular spatial cells, such as grid cells (Hafting et al., 2005) and object vector cells (Høydal et al., 2019). However many entorhinal neurons are also seemingly mixed selective to combinations of spatial variables, such as speed, heading direction, or position (Hardcastle et al., 2017). An alternate explanation is that neurons 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 Fig. 5a, b. A mouse must keep track of three objects as it moves in an environment (Fig. 5a); we model this as a linear biological RNN that must report the displacement of all objects from itself (Fig. 5b). Object positions are range independent but correlated, so the RNN representation modularises such that each neuron only encodes one object (Fig. 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 (Fig. 5d).

Range-dependent variables. Second, our theory gives a precise set of inequalities for (biologically constrained) modularisation of scalar sources. Breaking any one inequality means the optimal representation is linearly mixed selective (Figs. 1 and 3 to 5). These range (in)dependent ideas also qualitatively predict modularisation of nonlinear networks across a range of settings (Figs. 2 and 3) as well as nonlinear representations with linear readouts (Fig. 2). Fig. 5e, f re-illustrates these results. Poignantly, we derived that the optimal nonlinear representation that linearly encodes a set of variables is nonlinearly mixed (Section 3 and App. D). This suggests that nonlinear mixed selectivity might exist simply to save energy, rather than the typical rationale of permitting flexible linear readouts of arbitrary categorisations (Barack & Krakauer, 2021; Tye et al., 2024).

Sequential processing. Finally, multiple works have highlighted nonlinear mixed selectivity to stimulus feature and time within task (Parthasarathy et al., 2017; Dang et al., 2021; 2022). We show that simple linear sequential computations can produce this seemingly nonlinear temporal mixed selectivity. For example, Warden & Miller (2010) train monkeys to view sequences of two images separated fixed delays (Fig. 5g). To be rewarded, monkeys must report the stimuli, e.g., by saccading to the images in the correct sequence. Recently, Mikkelsen et al. (2023) reanalysed this data and found that many neurons are nonlinearly mixed selective to stimulus and time (Fig. 5h). However, we train a biological *linear* RNN to recall two sequentially presented one-hot cues separated by delays, and find similar neurons (Fig. 5i and App. J). This form of nonlinear mixed selectivity arises not from downstream readout pressures, but simply as the optimal energy-efficient code during fixed-length delays. Indeed, interpreting these as nonlinear relies on one viewing time as a scalar linearly increasing variable; if instead time is represented one-hot, both monkey and model neurons could be interpreted as linearly mixed selective.

7. Discussion

We have given precise constraints that cause linear biologically-constrained networks to modularise. These constraints are highly dependent on peculiar co-range properties of variables, and providing insight into modularisation patterns in both nonlinear networks and the brain.

Limitations. There remain many aspects of our theory that need improvement. Extending the theory to predict the modularisation of multidimensional sources, to non-orthogonal decoding, to more nonlinear settings, to other norms (App. C), or to understand a more granular notion of modularity (rather than perfectly modular or not) are attractive directions. Further, they might help us understand one puzzling aspect of the neural data. Despite correctly predicting population orthogonalisation patterns (Section 5), the data does not always match our single neural predictions. In particular, Panichello & Buschman (2021) find that about 20% of colour-tuned neurons are tuned to both colours, despite the orthogonal encoding, unlike in our theory.

Mechanistic interpretability. Insight into circuit behaviour, artificial or biological, has come from both single neuron (Hafting et al., 2005; Goh et al., 2021) and population (or feature) coding properties (Mante et al., 2013; Elhage et al., 2022). Instead, this work places precise constraints on when we should expect the two levels to be identical (i.e. modularity), and highlights subtle cases where they are not. These cases are complementary to "superposition", which occurs when there are fewer neurons than features (Elhage et al., 2022). Extending our analysis to this setting could be informative. Finally, our work incrementally contributes to the growing picture of correspondences between artificial and biological intelligence: in our framework, the two show surprisingly similar phenomenology.

Mixed selectivity. In neuroscience, our findings add nuance to the ongoing debate over mixed vs. modular coding. We point to a variety of settings in which mixed selectivity arises under energy constraints, without reference to a flexible linear readout of nonlinear classifications, as is classically argued. We do not think this theory is categorically wrong: neurons in cerebellum or mushroom body are likely nonlinearly mixed selective to allow kernel regression-like classification. To cite one cortical example, Bernardi et al. (2020) find nonlinearly mixed selective coding of independent variables, a phenomenon our theory would never predict, and flexible linear readouts currently seem the most likely explanation. Yet, our theory suggests a suite of alternative explanations that seem more plausible in other settings, especially in prefrontal cortex (Fig. 4), and including in the dataset for which these nonlinear classification theories were originally developed (Warden & Miller, 2010).

A. Optimal Representations in Positive Linear Autoencoders

A.1. Problem Statement

To begin, we study our simplest setting: a positive linear autoencoder that has to represent two bounded, scalar, mean-zero, sources, $x(t)$ and $y(t)$. These are encoded in a representation $\boldsymbol{g}(t) \in \mathbb{R}^N$, 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:

$$\boldsymbol{g}(t) = \boldsymbol{W}_{\text{in}} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \boldsymbol{b}_{\text{in}} \qquad \boldsymbol{W}_{\text{out}} \cdot \boldsymbol{g}(t) + \boldsymbol{b}_{\text{out}} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \tag{9}$$

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: $\boldsymbol{g}(t) \geq \boldsymbol{0}$.

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

$$\mathcal{L} = \frac{1}{T} \sum_t ||\boldsymbol{g}(t)||^2 + \lambda(||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2) = \langle ||\boldsymbol{g}(t)||^2 \rangle_t + \lambda(||\boldsymbol{W}_{\text{out}}||_F^2 + ||\boldsymbol{W}_{\text{in}}||_F^2) \tag{10}$$

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 & 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 & Hertz, 1991).

Our question can now be simply posed. What properties of the sources $x(t)$ and $y(t)$ ensure that they are optimally represented in disjoint sets of neurons? Equivalently, when does the representation modularise? The arguments of Whittington et al. (2023b) 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 is neccessary and sufficent 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 ($|\min_t x(t)| = \max_t x(t)$) and extreme point independent, meaning $\min_t(x(t) + y(t)) = \min_t x(t) + \min_t y(t)$, they should modularise.

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

$$\boldsymbol{g}(t) = \boldsymbol{u}x(t) + \boldsymbol{v}y(t) + \boldsymbol{b} \tag{11}$$

We will show that, for 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.

A.2. Conditions for Modularisation

For now, to save unnecessary complexity we will make the following simplifying assumptions, that will be relaxed later:

1. The sources are linearly uncorrelated, $\langle x(t)y(t) \rangle_t = \frac{1}{T} \sum_{t=1}^T x(t)y(t) = 0$
2. The sources are range-symmetric around zero, i.e. $|\min_t x(t)| = \max_t x(t) = -b_x$ and $|\min_t y(t)| = \max_t y(t) = -b_y$

We will consider the two losses in turn.

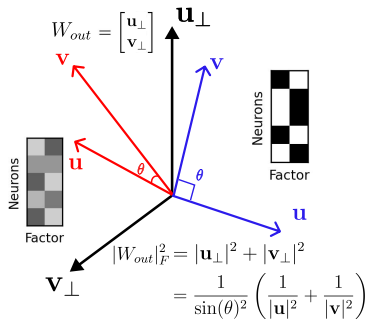

Figure 6: Schematics showing that modular solution minimizes readout weight W_{out} independent from encoding size. Smaller θ between $\mathbf{u}$ and $\mathbf{v}$ requires more energy to tease out the representation, which makes orthogonal solution being optimal.

A.2.1. Weight Loss

First, for a given linear representation of the form in equation 11, the minimum squared norm readout matrix has the following form:

$$\boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{v}_\perp^T \\ \boldsymbol{u}_\perp^T \end{bmatrix} \tag{12}$$

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:

$$||\boldsymbol{W}_{\text{out}}||_F^2 = \text{Tr}[\boldsymbol{W}_{\text{out}}\boldsymbol{W}_{\text{out}}^T] = ||\boldsymbol{u}_\perp||^2 + ||\boldsymbol{v}_\perp^2 \tag{13}$$

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 (see figure):

$$||\boldsymbol{W}_{\text{out}}||_F^2 = ||\boldsymbol{v}_\perp||^2 + ||\boldsymbol{u}_\perp||^2 = \frac{1}{\sin^2(\theta)}\left(\frac{1}{||\boldsymbol{u}||^2} + \frac{1}{||\boldsymbol{v}||^2}\right) \tag{14}$$

This has two interpretable properties. The larger the encoding the smaller the weight cost, and 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, 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:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}' \\ \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{v}' \end{bmatrix} \tag{15}$$

The min-norm input weight loss is simply:

$$||\boldsymbol{W}_{\text{in}}||_F^2 = ||\begin{bmatrix} \boldsymbol{u} & \boldsymbol{v} \end{bmatrix}||_F^2 = \text{Tr}[\boldsymbol{W}_{\text{in}}^T\boldsymbol{W}_{\text{in}}] = ||\boldsymbol{u}||^2 + ||\boldsymbol{v}^2|| \tag{16}$$

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.

A.2.2. 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:

$$g_i(t) = u_i x(t) + v_i y(t) + \Delta_i \tag{17}$$

Where, $\Delta_i = -\min_t[u_i x(t) + v_i y(t)]$ is the minimal bias required to make the representation non-negative. We will find the conditions under which, depending on the mixing coefficients (u_i and v_i), you decrease the loss by forming the modular solution:

$$\boldsymbol{g}_i(t) = \begin{bmatrix} u_i x(t) \\ v_i y(t) \end{bmatrix} + \begin{bmatrix} |u_i| b_x \\ |v_i| b_y \end{bmatrix} \tag{18}$$

Where $b_x = -\min_t x(t)$ is the bias required for a modular encoding of $x(t)$. If, for a given $x(t)$ and $y(t)$, it is true that modularising decreases the loss then the modular representation is optimal. If there are conditions when modularising increases the loss, then you can always usefully demodularise the representation to decrease the loss, and the optimal solution is not perfectly modular.

Let's analyse the activity loss of these two representations, for the modular representation (eq: 18):

$$\mathcal{L}_G^M = u_i^2(\langle x^2(t)\rangle_t + b_x^2)) + v_i^2(\langle y^2(t)\rangle + b_y^2) \tag{19}$$

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

$$\mathcal{L}_G^D = u_i^2\langle x^2(t)\rangle_t + v_i^2\langle y^2(t)\rangle + \Delta_i^2 \tag{20}$$

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

$$\Delta_i < \sqrt{u_i^2 b_x^2 + v_i^2 b_y^2} \tag{21}$$

Let's get some intuition for what this is saying for a simple setting when $v_i \approx 0$. Then the modular solution is better when, to second order in v_i, $\Delta_i < |u_i| b_x + \frac{v_i^2 b_y^2}{|u_i| b_x}$. This means that mixing a small amount of $y(t)$ into the representation of $x(t)$ is preferable when doing so does not increase the bias required to ensure that neuron stays positive, relative to what was required when it was representing x along. This is saying that $u_i x(t)$ has the same minima as $u_i x(t) + v_i y(t)$. For small enough v_i, an equivalent phrasing of this condition is that at the times at which $u_i x(t)$ takes its minimal value, $y(t)$ is non-negative. We can conceptualise this by plotting the allowed range of $x(t)$ and $y(t)$, it corresponds to a small slither of range missing from the shared range of $x(t)$ and $y(t)$ whose placement is determined by the sign of u_i.

That was for one particular pair of u_i and v_i, but the modular solution has to be better than all ratios in order to be optimal. 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 21, which in turns implies constraints on the allowed joint range of $x(t)$ and $y(t)$, one for each set of mixing coefficients.

Thus, the final argument for when modularity is preferred by the activity loss is as follows. If, at any corner (since, by assumption 1 - A.2, they are all symmetric), the allowed joint range of $x(t)$ and $y(t)$ is missing any of these slices a mixed solution will be preferred. If, however, for none of these lines, the required data is missing, the modular solution will be optimal.

This was a single neuron study of the activity loss, how does it interact with the weight loss? Since, for a fixed $||\boldsymbol{v}||$ and $||\boldsymbol{u}||$, the modular solution is one of the optimal solutions, we can never increase the loss by modularising as we have done from stepping from equation 17 to equation 18. Therefore, if making this change decreases the activity loss, it also decreases the whole loss.

Further, we could increase the sizes of $\boldsymbol{v}$ and $\boldsymbol{u}$, which would increase the activity loss at the cost of an increase in the weight loss, but it will not change this argument for when things modularise. Depending on the hyperparameter, λ, which trades-off the importance of the weight and activity losses in equation 10, these lengths will settle on different optimal, but regardless, the conditions under which modularity is optimal won't change. In general, if you only slightly break the condition for modularising being preferred you will only be slightly non-modular, and how 'non-modular' you become will depend on the hyperparameter λ. But whether the perfectly modular solution is optimal is, beautifully, independent of λ.

A.3. Relaxing Assumptions

We now show how the modularisation behaviour of variables that break the two assumptions listed at the top of A.2 can be similarly understood.

A.3.1. Correlated Variables

Correlated variables are easy to deal with, they simply introduce an extra term into the difference in activity losses, equation 20 minus equation 19:

$$\mathcal{L}_G^M - \mathcal{L}_G^D = u_i^2 b_x^2 + v_i^2 b_y^2 - 2u_i v_i \langle x(t)y(t)\rangle_t - \Delta_i^2 \tag{22}$$

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

$$\Delta_i < \sqrt{u_i^2 b_x^2 + v_i^2 b_y^2 - 2v_i u_i \langle x(t)y(t)\rangle_t} \tag{23}$$

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. 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 $x(t)$ and $y(t)$ 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 $x(t)$ negatively and $y(t)$ positively, i.e. anti-aligning them. If these two directions, one from the correlations and one from the range-depedent 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.

Of interest is that for these bounded variables the covariances, $\langle x(t)y(t)\rangle_t$, 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:

$$-b_y b_x \leq \langle x(t)y(t)\rangle_t \leq b_x b_y \tag{24}$$

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

$$\Delta_i > |u_i| b_x + |v_i| b_y \tag{25}$$

This bound is the development of equation 23 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. We will say that if, across time, when $x(t)$ takes its maximum (or minimum) value, $y(t)$ takes both its maximum and minimum values at different points in time, 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:

$$\Delta_i = -\min_t \Big[u_i x(t) + v_i y(t) \Big] = -|u_i| \min_t x(t) - |v_i| \min_t y(t) = |u_i| b_x + |v_i| b_y \tag{26}$$

Hence, we find that even the loosest inequality possible, eqaution 25, is never satisfied. Further, since the variables are extreme point independent their covariance never achieves the equalities in equation 24, so inequality 25 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.

A.3.2. Range Asymmetric Variables

Thus far we have assumed $|\min_t x(t)| = \max_t x(t)$, 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 ($g_+(t) = |u|x(t) + b$) or negatively ($g_-(t) = -|u|x(t) + b$), the losses will differ:

$$\langle g_+^2(t)\rangle_t = |u|^2 \big(\langle x(t)^2\rangle_t + (\min_t x(t))^2 \big) \qquad \langle g_-^2(t)\rangle_t = |u|^2 \big(\langle x(t)^2\rangle_t + (\max_t x(t))^2 \big) \tag{27}$$

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. 23:

$$\Delta_i < \sqrt{u_i^2 b_x^2 + v_i^2 b_y^2 - 2 v_i u_i \langle x(t) y(t) \rangle_t} \tag{28}$$

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.

One quick point is that, for range assymetric variables, the covariance is unbounded, and so it can be large enough to demodularise any pair of variables. The highest covariance you can reach for a pair of bounded, mean-zero, variables is for both variables to spend most of their time either both at the top of their range, $b_{\max}$, with some probability p, or at the bottom, $b_{\min}$, with some probability $1-p$. Because the variables are mean-zero $b_{\max} = -b_{\min}\frac{1-p}{p}$. Now the covariance is $b_{\min}^2 \frac{1-p}{p}$, which can grow to any value, and demodularise anything.

A.3.3. Remaining Assumptions

There are two remaining assumptions about the variables, that they are mean-zero and that they are bounded. The first of these is not really an assumption, 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. The boundedness constraint is 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.

A.4. Multivariate Representations

Now we go beyond two variables and consider when many sources modularise. First we generalise the arguments of section A.2.1 to multiple variables, then we study the activity loss.

A.4.1. Weight Loss

First, our modularisation argument relies on the fact that the weight loss is minimised for a modular solution. We showed this for a representation of two variables (section A.2.1), now we'll show the same is true for a multivariate representation:

$$\boldsymbol{g} = \sum_{i=1}^{M} \boldsymbol{u}_i x_i(t) + \boldsymbol{b} \tag{29}$$

With an affine transformation:

$$\boldsymbol{W}_{\text{out}} \cdot \boldsymbol{g}_t + \boldsymbol{b}_{\text{out}} = \begin{bmatrix} x_1(t) \\ \vdots \\ x_M(t) \end{bmatrix} \tag{30}$$

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

$$\boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{U}_1^T \\ \vdots \\ \boldsymbol{U}_M^T \end{bmatrix} \tag{31}$$

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

$$|\boldsymbol{W}_{\text{out}}|_F^2 = \text{Tr}[\boldsymbol{W}_{\text{out}} \boldsymbol{W}_{\text{out}}^T] = \sum_{m=1}^{M} |\boldsymbol{U}_m|^2 \tag{32}$$

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

$$\boldsymbol{U}_m = \frac{1}{|\boldsymbol{u}_m|^2}\boldsymbol{u}_m + \boldsymbol{u}_{m,\perp} \tag{33}$$

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

$$|\boldsymbol{W}_{\text{out}}|_F^2 = \sum_{m=1}^{M}\frac{1}{|\boldsymbol{u}_m|^2} + |\boldsymbol{u}_{m,\perp}|^2 \geq \sum_{m=1}^{M}\frac{1}{|\boldsymbol{u}_m|^2} \tag{34}$$

And this lower bound is achieved whenever the $\{\boldsymbol{u}_m\}_{m=1}^M$ vectors are orthogonal to one another, since then $\boldsymbol{u}_{m,\perp} = 0$. Therefore, again, 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. The input weight loss is, as before, dependent only on the encoding sizes and not on their alignment. This means we can again just study the activity loss' behaviour. The representation will modularise when the activity loss is minimised by modularising.

A.4.2. Complete Multivariable Modularisation

We can find generalised conditions for multivariate representations to entirely modularise. Let's say we have M uncorrelated, mean-zero, symmetric, variables $\{x_i(t)\}_{m=1}^M$. we compare two representations:

$$g_i(t) = \sum_{j=1}^{M} u_{ij}x_j(t) + \Delta_i \tag{35}$$

And:

$$\boldsymbol{g}_i(t) = \begin{bmatrix} u_{i1}x_1(t) \\ \vdots \\ u_{iM}x_M(t) \end{bmatrix} + \begin{bmatrix} |u_{i1}|b_1 \\ \vdots \\ |u_{iM}|b_M \end{bmatrix} \tag{36}$$

The activity loss difference between these two representations is:

$$\mathcal{L}_G^D - \mathcal{L}_G^M = \Delta_i^2 - \sum_j u_{ij}^2 b_j^2 \tag{37}$$

So the solution completely modularises if:

$$\Delta_i \geq \sqrt{\sum_j u_{ij}^2 b_j^2} \tag{38}$$

Or with correlations:

$$\Delta_i \geq \sqrt{\sum_j u_{ij}^2 b_j^2 - \sum_{j,k\neq j} u_{ij}u_{ik}\langle x_j(t)x_k(t)\rangle_t} \tag{39}$$

First, this preserves the property that a set of extreme point independent, range-symmetric, variables should always modularise, regardless of the correlation. Second, however, rather than creating one set of range conditions parameterised by a mixing ratio, as in 21, we get a family of conditions parameterised by $M-1$ mixing ratios.

A.5. Non-Veridical Inputs

One remaining concern might be that we provide our networks with direct access to the veridical sources, i.e. each dimension of the input vector is a source variable. The structure of our arguments makes it clear how the theory can easily be extended to data that is an orthogonal mixture of a set of sources, $\boldsymbol{x} = \boldsymbol{O}\boldsymbol{y}$, for some orthogonal matrix $\boldsymbol{O}$, meaning access to the veridical sources in the input is not the important determinant. On the other hand, extending to non-orthogonal encodings or decodings is difficult, as it means modularising no longer minimises the weight energy. This introduces a competition between activity and weights mediated by λ. We leave exploration of this tradeoff to future work.

A.6. Studying Particular Data Examples

A.6.1. One Hot Codes

Imagine you have two one-hot codes of dimension D, $\boldsymbol{x}$ and $\boldsymbol{y}$. If the two categories are sampled with replacement then for any linear projection of the one-hot codes, extreme point independence is satisfied, and the codes will modularise.

However, imagine a pair of sampled-without-replacement one-hot codes, and consider a single component of each code. Due to the sampled without replacement property they can never both be 'on', in fact, across the dataset, the joint probability distribution is:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{cases} \begin{bmatrix} D-1 \\ -1 \end{bmatrix} & \text{with probability } \frac{1}{D} \\ \begin{bmatrix} -1 \\ D-1 \end{bmatrix} & \text{with probability } \frac{1}{D} \\ \begin{bmatrix} -1 \\ -1 \end{bmatrix} & \text{with probability } \frac{D-2}{D} \end{cases} \tag{40}$$

Then we can consider two codes, either modularised:

$$\boldsymbol{g}_i = \begin{bmatrix} x_0 + 1 \\ y_0 + 1 \end{bmatrix} \qquad \langle |\boldsymbol{g}_i|^2 \rangle = 2(\langle x_0^2 \rangle + 1) \tag{41}$$

Or a mixed code that exploits the missing top-right corner of the range:

$$g_i = -x_0 - y_0 + 1 \qquad \langle g_i^2 \rangle = 2\langle x_0^2 \rangle + 1 + 2\langle x_0 y_0 \rangle \tag{42}$$

And since $\langle x_0 y_0 \rangle = -1$, $\langle g_i^2 \rangle \leq \langle |\boldsymbol{g}_i|^2 \rangle$, the code should mix.

Remodularising Correlations One might wonder if there was a way to change the correlations of these variables to remodularise a sampled-without-replacement code. Unfortunately there is not, even in the most general such code the correlations are fixed:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{cases} \begin{bmatrix} b_1 \\ -1 \end{bmatrix} & \text{with probability } p_1 \\ \begin{bmatrix} -1 \\ b_2 \end{bmatrix} & \text{with probability } p_2 \\ \begin{bmatrix} -1 \\ -1 \end{bmatrix} & \text{with probability } 1 - p_1 - p_2 \end{cases} \tag{43}$$

To keep the mean zero property $b_i = \frac{1-p_i}{p_i}$, and no matter the values of p_1 and p_2, $\langle x_0 y_0 \rangle = -1$.

A.6.2. Frequencies

Another relevant setting is representations of linear combinations of frequencies:

$$\boldsymbol{g}(t) = \boldsymbol{b}_0 + \sum_{d=1}^{D} \boldsymbol{a}_d \cos(\omega_d t) + \boldsymbol{b}_d \sin(\omega_d t) \tag{44}$$

Let's focus for now on a simple case of just two frequencies:

$$\boldsymbol{g}(t) = \boldsymbol{b} + \boldsymbol{a}_1 \cos(\omega_1 t) + \boldsymbol{a}_2 \cos(\omega_2 t) + \boldsymbol{b}_1 \sin(\omega_1 t) + \boldsymbol{b}_2 \sin(\omega_2 t) \tag{45}$$

And let's rescale time to remove one of the frequencies:

$$\boldsymbol{g}(t) = \boldsymbol{b} + \boldsymbol{a}_1 \cos(t) + \boldsymbol{a}_2 \cos(\omega t) + \boldsymbol{b}_1 \sin(t) + \boldsymbol{b}_2 \sin(\omega t) \tag{46}$$

When should these frequencies mix their representation to reduce the activity loss?

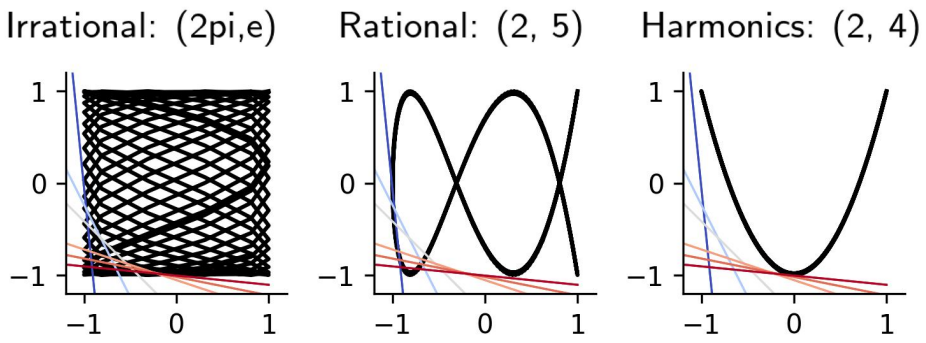

Figure 7: Schematics showing irrational and rational ratio case the data are range dependent beyond the modular-mixed boundary while in de-modularising harmonics case, the two periodic waves are range independent.

Irrational Frequencies Modularise If ω is irrational then, by Kronecker's theorem, you can find a value of t for which $\cos(t)$ and $\cos(\omega t)$ take any pair of values (and the same for sine). This makes the two frequencies, among other things, extreme point independent, and therefore they modularise.

Even Integer Multiples Mix If ω is an even integer then we can consider the following mixed neuron:

$$g_i(t) = \cos(t) + \delta \cos(nt) + \Delta \tag{47}$$

To modularise, for all δ:

$$\Delta > \sqrt{1+\delta^2} = 1 + \frac{\delta^2}{2} + \mathcal{O}(\delta^4) \tag{48}$$

$\cos(t)$ takes its minimal value of -1 at $t = m\pi$, at these values $\cos(nt) = 0$. If δ is small enough this is the only behaviour that matters, so $\Delta = 1$, which is smaller than the critical value, and 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:

$$g_i(t) = \cos(t) + \delta \sin(nt) + \Delta \tag{49}$$

Then the same argument goes through. Though notice that this required us to have both sine and cosine of every frequency to make this argument.

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

$$g_i(t) = \cos(t) + \delta \cos(\frac{p}{q}t) + \Delta \tag{50}$$

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 $t = q\pi$:

$$g_i(t) = \cos(q\pi) + \delta \cos(p\pi) + \Delta = -1 - \delta + \Delta > 0 \tag{51}$$

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

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:

$$g_i(t) = -1 + \delta \cos(p\pi) + \Delta = -1 - |\delta| + \Delta > 0 \tag{52}$$

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 that(Soudry & Speiser, 2015):

$$kp = q + 1 \mod 2q \tag{53}$$

Therefore take $t = k\pi$:

$$g_i(t) = \cos(k\pi) + \delta\cos(\pi(\frac{kp}{q})) + \Delta = -1 + \delta\cos(\pi(\frac{q+1+2nq}{q})) + \Delta \tag{54}$$

For some integer n. Developing:

$$g_i(t) = -1 - \delta\cos(\frac{\pi}{q}) + \Delta \tag{55}$$

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

$$-1 - \delta\cos(\frac{\pi}{q}) + \sqrt{1+\delta^2} \geq 0 \tag{56}$$

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

$$\delta \geq \frac{2\cos(\frac{\pi}{q})}{1-\cos^2(\frac{\pi}{q})}. \tag{57}$$

We can apply a similar argumnet with $t' = k\pi = \frac{qt}{p}$ to get:

$$-\cos(\frac{\pi}{p}) - \delta + \Delta \geq 0 \tag{58}$$

This leads us to:

$$\delta \leq \frac{1-\cos^2(\frac{\pi}{p})}{2\cos(\frac{\pi}{p})} \tag{59}$$

For any integer p, q $cos(\frac{\pi}{p})$ and $cos(\frac{\pi}{q})$ is is [0,1] and $p \neq q$ since they have different oddity. Thus the inequalities eqn. 56 and 58 cannot be held at the same time and thus such δ does not exist and we can conclude that there is no Δ below the critical value with nonnegativity constraint.

For the last case, where $\delta < 0$ and both $p, q < 0$, we have not finished the proof here. Yet, we empirically show on Fig 7 that in rational multiple frequencies, there is no modular-mixed boundary possible to break the modularisation.

B. 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. (2023b)) 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:

$$\boldsymbol{W}_{\text{out}}\boldsymbol{g}(t) + \boldsymbol{b}_{\text{out}} = \begin{bmatrix}\cos(\omega_1 t)\\ \cos(\omega_2 t)\end{bmatrix} \tag{60}$$

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

$$\boldsymbol{W}_{\text{rec}}\boldsymbol{g}(t) + \boldsymbol{b} = \boldsymbol{g}(t + \Delta t) \tag{61}$$

We will study this two frequency setting and ask when the representation learns to modularise the two frequencies. These results can be easily generalised to multiple variables as in section A.4. Again, our representation must be non-negative, $\boldsymbol{g}(t) \geq \boldsymbol{0}$, and we minimise the following energy loss:

$$\mathcal{L} = \langle||\boldsymbol{g}(t)||^2\rangle + \lambda_W||\boldsymbol{W}_{\text{rec}}||_F^2 + \lambda_R||\boldsymbol{W}_{\text{out}}||_F^2 \tag{62}$$

The optimal linear representation will contain three parts, two are as before: the frequencies that have to be readout, a positive offset to make the representation non-negative. However, additionally recurrence forces us to have some extra components. In order for the linear system to autonomously generate the frequencies, both the sine and cosine of any given frequency must be included, i.e. the optimal representation takes the following form:

$$\boldsymbol{g}(t) = \boldsymbol{a}_1 \cos(\omega_1 t) + \boldsymbol{b}_1 \sin(\omega_1 t) + \boldsymbol{a}_2 \cos(\omega_2 t) + \boldsymbol{b}_2 \sin(\omega_2 t) + \boldsymbol{b}_0 \tag{63}$$

Our argument will again rely on the fact that the weight losses are minimal when the representation is modular, so that we can just study the activity loss to predict modularisation. We studied the activity loss for linear combinations of frequencies in section A.6.2. These results generalise to the recurrent setting if all the weight losses are minimised when the solution is modular. We therefore proceed to show that this is true.

B.1. Readout Loss

The readout loss is relatively easy, again create some capitalised pseudoinverse vectors $\{\boldsymbol{A}_i, \boldsymbol{B}_i\}_{i=1}^2$ 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:

$$\boldsymbol{W}_{\text{out}} = \begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \end{bmatrix} \tag{64}$$

And the readout loss is:

$$|\boldsymbol{W}_{\text{out}}|_F^2 = \text{Tr}[\boldsymbol{W}_{\text{out}} \boldsymbol{W}_{\text{out}}^T] = |\boldsymbol{A}_1|^2 + |\boldsymbol{A}_2|^2 \tag{65}$$

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, and from each of the $\boldsymbol{b}_i$ vectors), which 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.

B.2. Recurrent Loss without Bias

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

$$\boldsymbol{W}\boldsymbol{g}(t) = \boldsymbol{g}(t+1) \tag{66}$$

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

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{b}_1 & \boldsymbol{a}_2 & \boldsymbol{b}_2 & \boldsymbol{b}_0 \end{bmatrix} \tag{67}$$

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

$$\boldsymbol{X}^\dagger = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_1 & \boldsymbol{A}_2 & \boldsymbol{B}_2 & \boldsymbol{B}_0 \end{bmatrix} \tag{68}$$

And finally create an ideal rotation matrix:

$$\boldsymbol{R} = \begin{bmatrix} \cos(\omega_1 \Delta t) & -\sin(\omega_1 \Delta t) & 0 & 0 & 0 \\ \sin(\omega_1 \Delta t) & \cos(\omega_1 \Delta t) & 0 & 0 & 0 \\ 0 & 0 & \cos(\omega_2 \Delta t) & -\sin(\omega_2 \Delta t) & 0 \\ 0 & 0 & \sin(\omega_2 \Delta t) & \cos(\omega_2 \Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_1 & 0 & 0 \\ 0 & \boldsymbol{R}_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{69}$$

Now:

$$\boldsymbol{W} = \boldsymbol{X}\boldsymbol{R}\boldsymbol{X}^{\dagger,T} \tag{70}$$

We can then calculate the recurrent weight loss:

$$|\boldsymbol{W}|_F^2 = \text{Tr}[\boldsymbol{W}\boldsymbol{W}^T] = \text{Tr}[\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{R} \boldsymbol{X}^{\dagger,T} \boldsymbol{X}^\dagger \boldsymbol{R}^T] \tag{71}$$

$\boldsymbol{X}^T\boldsymbol{X}$ is a symmetric 5×5 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, $\boldsymbol{Y} = \boldsymbol{X}^T\boldsymbol{X}$:

$$|\boldsymbol{W}|_F^2 = \text{Tr}[\boldsymbol{Y}\boldsymbol{R}\boldsymbol{Y}^{-1}\boldsymbol{R}^T] \tag{72}$$

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, $e_1 \geq \ldots \geq e_n \geq 0$ and $f_1 \geq \ldots \geq f_n \geq 0$

$$\text{Tr}[\boldsymbol{E}\boldsymbol{F}] \geq \sum_{i=1}^{n} e_i f_{n-i+1} \tag{73}$$

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:

$$|\boldsymbol{W}|_F^2 = \text{Tr}[\boldsymbol{Y}\boldsymbol{R}\boldsymbol{Y}^{-1}\boldsymbol{R}^T] \geq \sum_{i=1}^{5} \frac{\lambda_i}{\lambda_i} = 5 \tag{74}$$

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:

$$|\boldsymbol{W}|_F^2 = \text{Tr}[\boldsymbol{Y}\boldsymbol{R}\boldsymbol{Y}^{-1}\boldsymbol{R}^T] = \text{Tr}[\boldsymbol{Y}\boldsymbol{Y}^{-1}\boldsymbol{R}^T\boldsymbol{R}] = \text{Tr}[\mathbb{I}_5] = 5 \tag{75}$$

B.3. Recurrent Loss with Bias

Now we return to the case of interest:

$$\boldsymbol{W}\boldsymbol{g}(t) + \boldsymbol{b} = \boldsymbol{g}(t+1) \tag{76}$$

Our energy loss, equation 62, 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 63) and constructing the following, smaller, min-norm weight matrix:

$$\boldsymbol{W} = \begin{bmatrix}\boldsymbol{a}_1 & \boldsymbol{b}_1 & \boldsymbol{a}_2 & \boldsymbol{b}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_1 & 0 \\ 0 & \boldsymbol{R}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_1^T \\ \boldsymbol{B}_1^T \\ \boldsymbol{A}_2^T \\ \boldsymbol{B}_2^T\end{bmatrix} = \hat{\boldsymbol{X}}\hat{\boldsymbol{R}}\hat{\boldsymbol{X}}^\dagger \tag{77}$$

Where $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ were defined in equation 69. 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 $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^2$, 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:

$$\boldsymbol{A}_1 = \hat{\boldsymbol{A}}_1 + \hat{\boldsymbol{A}_{1,\perp}} \tag{78}$$

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 $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^2$. 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 $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^2$, 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:

$$\hat{\boldsymbol{X}}^\dagger = \hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^\dagger \tag{79}$$

Then:

$$\hat{\boldsymbol{Y}}^\dagger = (\hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^\dagger)^T(\hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^\dagger) = \hat{\boldsymbol{X}}_0^{\dagger,T}\hat{\boldsymbol{X}}_0^\dagger + \hat{\boldsymbol{X}}_\perp^{\dagger,T}\hat{\boldsymbol{X}}_\perp^\dagger = \hat{\boldsymbol{Y}}^{-1} + \hat{\boldsymbol{Y}}_\perp^\dagger \tag{80}$$

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:

$$|\boldsymbol{W}|_F^2 = \text{Tr}[\hat{\boldsymbol{Y}}\hat{\boldsymbol{R}}\hat{\boldsymbol{Y}}^{-1}\hat{\boldsymbol{R}}^T] + [\hat{\boldsymbol{Y}}\hat{\boldsymbol{R}}\hat{\boldsymbol{Y}}_\perp^\dagger\hat{\boldsymbol{R}}^T] \tag{81}$$

Now, the first term is greater than or equal than 4, 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, $|\boldsymbol{W}|_F^2 \geq 4$, and orthogonal encodings achieve this bound, therefore it is an optimal solution according to the weight loss.

C. L1 Regularisation

There are other ways we could have decided to penalise the activity and weight energy. One particularly relevant one is the L1 norm. Biologically, the intuition for penalising firing rates is that each spike is costly so you should minimise the sum of the firing rates, i.e. the L1 norm of the activity vector. Trying this we find it interestingly cares only about the range properties. Consider a linear positive representation of two mean-zero bounded scalar variables

$$\boldsymbol{g}(t) = \boldsymbol{u}x(t) + \boldsymbol{v}y(t) + \boldsymbol{b} \tag{82}$$

Since the representation is positive:

$$\langle |\boldsymbol{g}(t)|_1 \rangle_t = \langle \boldsymbol{u}x(t) + \boldsymbol{v}y(t) + \boldsymbol{b} \rangle_t = |\boldsymbol{b}|_1 \tag{83}$$

Which is a nice simple answer. Let's consider a mixed tuned neuron:

$$g_i(t) = x(t) + vy(t) + b_x + vb_y - \Delta \tag{84}$$

And it's modularised counterpart, $\boldsymbol{g}_i(t)$. The difference in L1 activity energy is:

$$\mathcal{L}_G^D - \mathcal{L}_G^M = \langle |\boldsymbol{g}_i(t)|_1 \rangle_t - \langle |g_i(t)|_1 \rangle_t = -\Delta \tag{85}$$

So if the variables are extreme point independent, the two representations are equally costly, if not mixing will be preferred. This of course assumes we are using an L2 weight penalty. An L1 weight penalty might have other interesting effects.

D. Partially Nonlinear Theory

In this section we consider a nonlinear input/linear output setting, and show that we can still make some surprising amounts of progress.

D.1. Optimal Single-Neuron Univariate Linearly Decodable Representation is a ReLU

We can get some traction on this question by considering the optimal nonlinear encoding of a single variable. Let's say we want to linearly readout the variable, then the representation must have a linear component, and a second component that ensures positivity:

$$\boldsymbol{g}(t) = \boldsymbol{u}x(t) + \boldsymbol{f}(x(t)) \tag{86}$$

What should $\boldsymbol{f}$ be? Let's consider a single neuron:

$$g(x) = x + f(x) \tag{87}$$

We want to choose an f such that (i) $g \geq 0$ (ii) there is a strong encoding of x that can be linearly readout (iii) the firing energy is minimised. This is an inequality constrained functional optimisation problem:

$$\mathcal{L}_f = \langle g^2(x) \rangle_x + \mu(\langle f(x)x \rangle_x - \int \lambda(x)(x + f(x))dp(x) \tag{88}$$

The objective is the first term, the second term enforces orthogonality between the nonlinear term and x, so that the firing rates are incapable of just removing the linear encoding (otherwise the optimal f is $f = -x$, we return to this point later), and the last term embodies the set of positivity inequality constraints, so either $\lambda(x) = 0$ and $f(x) > -x$ or $f(x) = -x$ and $\lambda(x) > 0$. We take the derivative with respect to $f(x)$, and since, due to orthogonality, $\langle g^2(x) \rangle_x = \langle x^2 \rangle_x + \langle f^2(x) \rangle_x$:

$$\frac{\delta \mathcal{L}_f}{\delta f(x)} = 2f(x)p(x) + \mu x p(x) - \lambda(x)p(x) = 0 \tag{89}$$

When $p(x) = 0$ this equation is satisfied, and the behaviour of the functions f and λ is undefined. Let's focus on the values of x for which $p(x) \neq 0$. We can find the value of μ by multiplying this expression by x and integrating, this gives us:

$$\mu = \frac{\int \lambda(x) x dp(x)}{\langle x^2 \rangle_x} = \frac{\alpha}{\langle x^2 \rangle_x} \tag{90}$$

Then we get the following equation for $\lambda(x)$:

$$\lambda(x) = 2f(x)p(x) + \frac{\alpha x p(x)}{\langle x^2 \rangle_x} \tag{91}$$

Now, either $\lambda(x) = 0$, in which case:

$$f(x) = -\frac{\alpha x}{2\langle x^2 \rangle_x} \tag{92}$$

But remember, $f(x) > -x$, so this solution is only possible if $\alpha < 2\langle x^2 \rangle_x$. Or $f(x) = -x$, in which case:

$$\lambda(x) = \frac{(\alpha - 2\langle x^2 \rangle_x)x}{\langle x^2 \rangle_x} \tag{93}$$

But also $\lambda > 0$. Then we have two options, either $\alpha > 2\langle x^2 \rangle_x$, then, when $x > 0$, $\lambda(x) > 0$. This doesn't work however, since then 92 is never satisfied. This means that when $x < 0$ neither $\lambda(x)$, nor $f(x) + x$ is 0, breaking the assumptions.

The other solution must therefore hold, $\alpha < 2\langle x^2 \rangle_x$, hence when $x < 0$, $f(x) = -x$ and $\lambda(x)$ is as in 93. Then when $x > 0$, $\lambda(x) = 0$ and $f(x)$ is as in eqn. 92. We can find then find the value of α:

$$\alpha = \int \lambda(x) x dp(x) = \frac{\alpha}{\langle x^2 \rangle_x} \int_0^\infty x^2 dp(x) - 2\int_0^\infty x^2 dp(x) \tag{94}$$

Then:

$$\alpha = \frac{2\langle x^2 \rangle_x \int_0^\infty x^2 dp(x)}{\int_0^\infty x^2 dp(x) - \langle x^2 \rangle_x} = -2\langle x^2 \rangle_x \frac{\int_0^\infty x^2 dp(x)}{\int_{-\infty}^0 x^2 dp(x)} \tag{95}$$

And hence the optimal function to add is:

$$f(x) = \begin{cases} \frac{\int_0^\infty x^2 dp(x)}{\int_{-\infty}^0 x^2 dp(x)} x & x > 0 \\ -x & x < 0 \end{cases} \tag{96}$$

And the resulting optimal single unit tuning curve is a bias-free ReLU:

$$g(x) = \begin{cases} (\frac{\int_0^\infty x^2 dp(x)}{\int_{-\infty}^0 x^2 dp(x)} + 1)x & x > 0 \\ 0 & x < 0 \end{cases} \tag{97}$$

This is one neuron, in general across the population, you can encode $-x$ instead of x, and the whole population will cleanly encode x in a optimal energy-efficient decodable way.

A key part of this argument is that $f(x)$ should be orthogonal to x. This seems reasonable, if it is not then $f(x)$ has some linear component: $f(x) = \hat{f}(x) + f_x x$ with $\langle \hat{f}(x) x \rangle = 0$ and we can instead argue our theory is a prediction of $\hat{f}(x)$. Hence this neuron must have a constant offset, and it must make the representation energy-efficient, positive, without reducing the linearly decodable encoding of x, for which bias-free ReLUs are surprisingly optimal.

D.2. Optimal Bivariate Tuning Curve is still a Bias-Free ReLU

We can repeat this argument for functions of two variables:

$$g_{u,v}(x, y) = ux + vy + f_{u,v}(x, y) \tag{98}$$

Then the lagrangian is:

$$\langle |g_{u,v}(x, y)|^2 \rangle_{x,y} + \mu_X \langle f_{u,v}(x, y) x \rangle + \mu_Y \langle f_{u,v}(x, y) y \rangle - \langle \lambda(x, y)(ux + yv + f_{u,v}(x, y)) \rangle \tag{99}$$

Taking the functional derivative:

$$2f_{u,v}(x, y) + \mu_X x + \mu_Y y - \lambda(x, y) = 0 \tag{100}$$

Again, either $\lambda(x, y) = 0$, and:

$$2f_{u,v}(x, y) = -\mu_X x - \mu_Y y \geq -ux - vy \tag{101}$$

Or $2f_{u,v}(x, y) = -ux - vy$ and:

$$\lambda(x, y) = (\mu_X - 2u)x + (\mu_Y - 2v)y \tag{102}$$

We can see that these two regimes are separated by the hyperplane: $(\mu_X - 2u)x + (\mu_Y - 2v)y = 0$ (since μ_X and μ_Y are constants). One side of the hyperplane the neuron's activity is 0, the other it is some linear function of the data determined by the lagrange multipliers. Hence again we get:

$$g_{u,v}(x, y) = \begin{cases} 0 & (2u - \mu_X)x + (2v - \mu_Y)y < 0 \\ (u - \frac{\mu_X}{2})x + (v - \frac{\mu_Y}{2})y & \text{else} \end{cases} \tag{103}$$

Exactly the equation of a bias-free ReLU network.

D.3. When should nonlinear encodings modularise?

Let's now consider the difference between part of a modular representation of two range independent variables. As in the previous section, our optimal modular nonlinear encoding will include terms like:

$$\boldsymbol{g}(x, y) = \begin{bmatrix} \alpha x \\ \beta y \end{bmatrix} + \begin{bmatrix} |\alpha| f_x(x) \\ |\beta| f_y(y) \end{bmatrix} \tag{104}$$

Or its mixed equivalent:

$$g(x, y) = \alpha x + \beta y + f_{\alpha,\beta}(x, y) \tag{105}$$

Where we've used the results from the previous section to preclude the possibility that other mixed functions would be better to include. Then the cost difference of the two representations is:

$$\frac{1}{2}\Delta\mathcal{L}_G = \beta\alpha\langle xy\rangle_{x,y} + \langle f^2_{u,v}(x, y)\rangle_{x,y} - \beta^2\langle f^2_y(y)\rangle_y - \alpha^2\langle f^2_x(x)\rangle_x \tag{106}$$

In general, all these encodings could point in the opposite direction, so all four quadrants are separately explored. If either of the quandrants in which the encodings align (either $x > 0$ and $y > 0$, or $x < 0$ and $y < 0$) are negative it should mix. Similarly in the other two quadrants if the value if positive they should mix.

D.4. Empirical Evidence in What-Where Networks

These optimal single unit tuning curves can be seen in the 2D what-where setting. When these networks modularise, they do so by partitioning their response to each input dimension into two separate neurons, with each neuron responding to half the possible input values. This is also reflected in the hidden weights (see 8), whereby single units are differentially responsive to only half the set of shapes or positions.

This has an effect on the critical point at which corner-dropout starts encouraging the encodings to be mixed instead of modular; this is because it is how the encodings are affected by dropout that it is important. Considering the entire 9×9 input space, 5 data points need to be removed from the corner for mixed selectivity to be optimal; however, this number decreases to 3 data points when we move to the half-plane setting, suggesting that these networks should stop being modular at a lower dropout than previously expected.

E. Metrics for Representational Modularity and Inter-Source Statistical Dependence

Dunion et al. (2023) leverage conditional mutual information in a similar vein 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

$$I(\boldsymbol{z}_j; \boldsymbol{s}_i|\boldsymbol{s}_{-i}) = I(\boldsymbol{z}_j; \boldsymbol{s}) - I(\boldsymbol{z}_j; \boldsymbol{s}_{-i}), \tag{107}$$

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 normalize conditional mutual information by $H(\boldsymbol{s}_i|\boldsymbol{s}_{-i})$ to obtain a

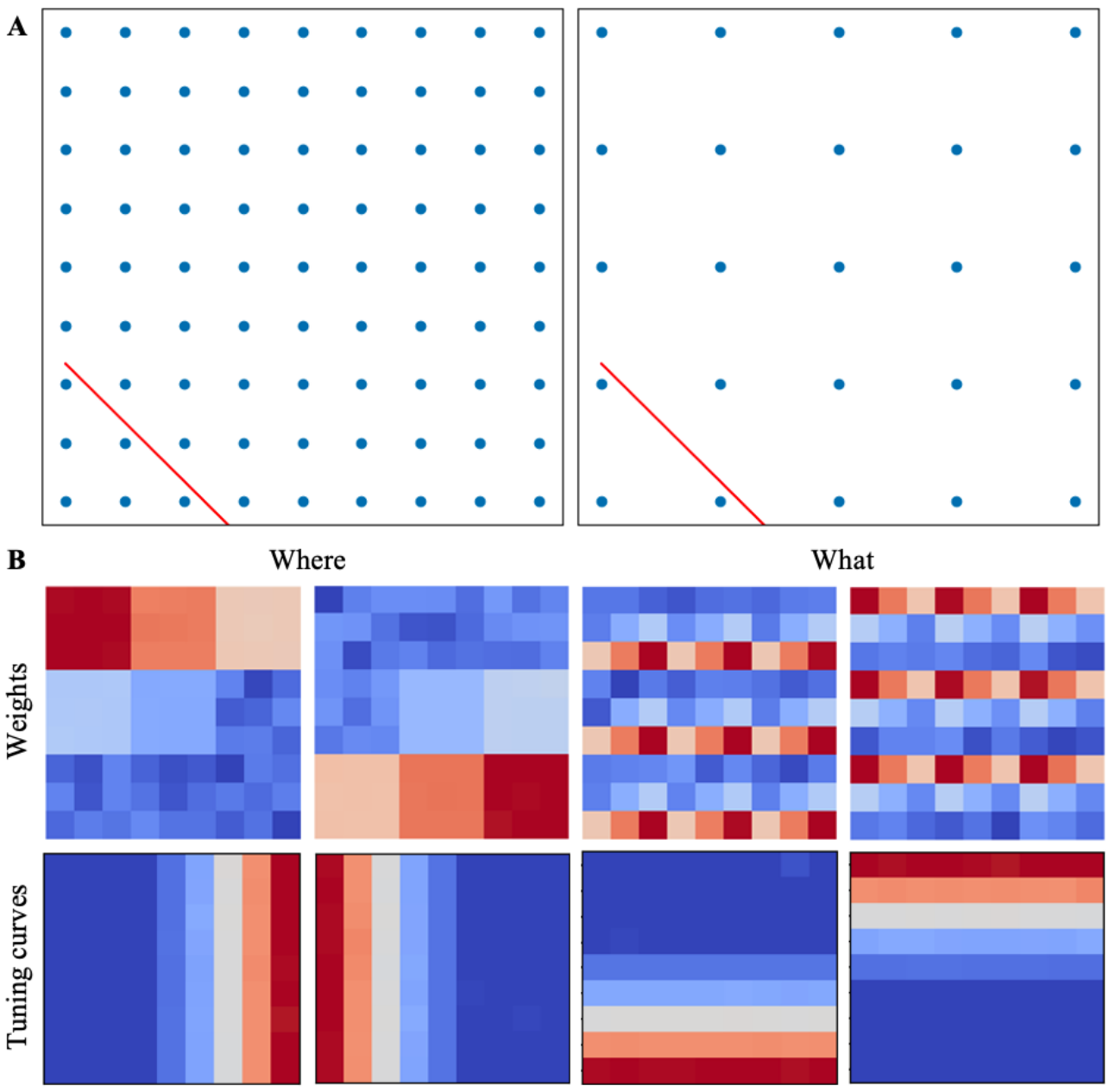

Figure 8: A. The change in dropout necessary for mixed selectivity, when considering activity responses in the linear case (left) versus the ReLU case (right). Note that this change corresponds with a discrete jump in the number of data points that must be removed from the corner (5 up from 3). B. Hidden weights and tuning curves of 4 units, showing optimal nonlinear encodings take the form of half-plane ReLUs

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 normalized average "max-over-sum" in a column as a measure of sparsity, following Hsu et al. (2023):

$$\mathrm{CInfoM}(\boldsymbol{s}, \boldsymbol{z}) := \left(\frac{1}{d_z} \sum_{j=1}^{d_z} \frac{\max_i \boldsymbol{C}_{ij}}{\sum_{i=1}^{d_s} \boldsymbol{C}_{ij}} - \frac{1}{d_s} \right) \Bigg/ \left(1 - \frac{1}{d_s} \right). \tag{108}$$

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 $\mathcal{V}$-information framework of Xu et al. (2020). We specify the function class $\mathcal{V}$ as linear and calculate

$$\begin{aligned} I_{\mathcal{V}}(\boldsymbol{z}_j \to \boldsymbol{s}_i | \boldsymbol{s}_{-i}) &= I_{\mathcal{V}}(\boldsymbol{z}_j \to \boldsymbol{s}) - I_{\mathcal{V}}(\boldsymbol{z}_j \to \boldsymbol{s}_{-i}) \\ &= H_{\mathcal{V}}(\boldsymbol{s}) - H_{\mathcal{V}}(\boldsymbol{s}|\boldsymbol{z}_j) - H_{\mathcal{V}}(\boldsymbol{s}_{-i}) + H_{\mathcal{V}}(\boldsymbol{s}_{-i}|\boldsymbol{z}_j), \end{aligned} \tag{109}$$

followed by a normalization by $H_{\mathcal{V}}(\boldsymbol{s}_i|\boldsymbol{s}_{-i})$. Each predictive conditional $\mathcal{V}$-entropy term is estimated via a standard maximum log-likelihood optimization over $\mathcal{V}$, 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 Eq. 108.

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

$$\mathrm{NI}(\boldsymbol{s}) = 1 - \frac{H(\boldsymbol{s})}{\sum_{i=1}^{d_s} H(\boldsymbol{s}_i)}. \tag{110}$$

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.

F. What-Where Task

F.1. Experimental Setup

F.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 (9) 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. Correlations later introduced duplicate a subset of these data points.

F.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 (9).The network architecture is formally defined as follows:

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

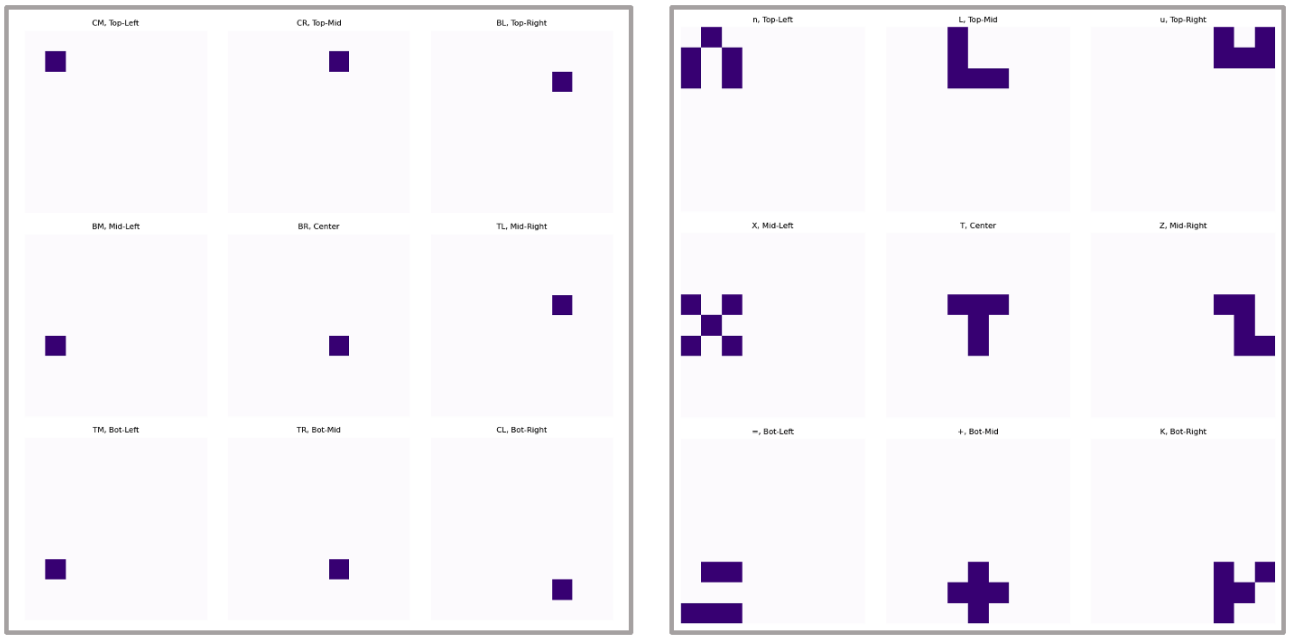

Figure 9: Different possible inputs to the network. Both one-hot (left) and letter-based (right) shapes can be outputted as either a concatenation of one-hots or as a 2D variable.

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.

F.1.3. Training Protocols

The network uses the Adam optimiser ((Kingma & 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:

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

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

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

Here, $\mathbf{W}$ and $\mathbf{h}$ represent the hidden weights and activations, respectively, and λ_R is the regularisation constant, 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.

F.2. Modularity of What & Where

F.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:

$$L_{reg} = \frac{\lambda_R}{N}\left(\sum_{i,j} W_{ij}^2 + \sum_i h_i^2\right)$$

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.

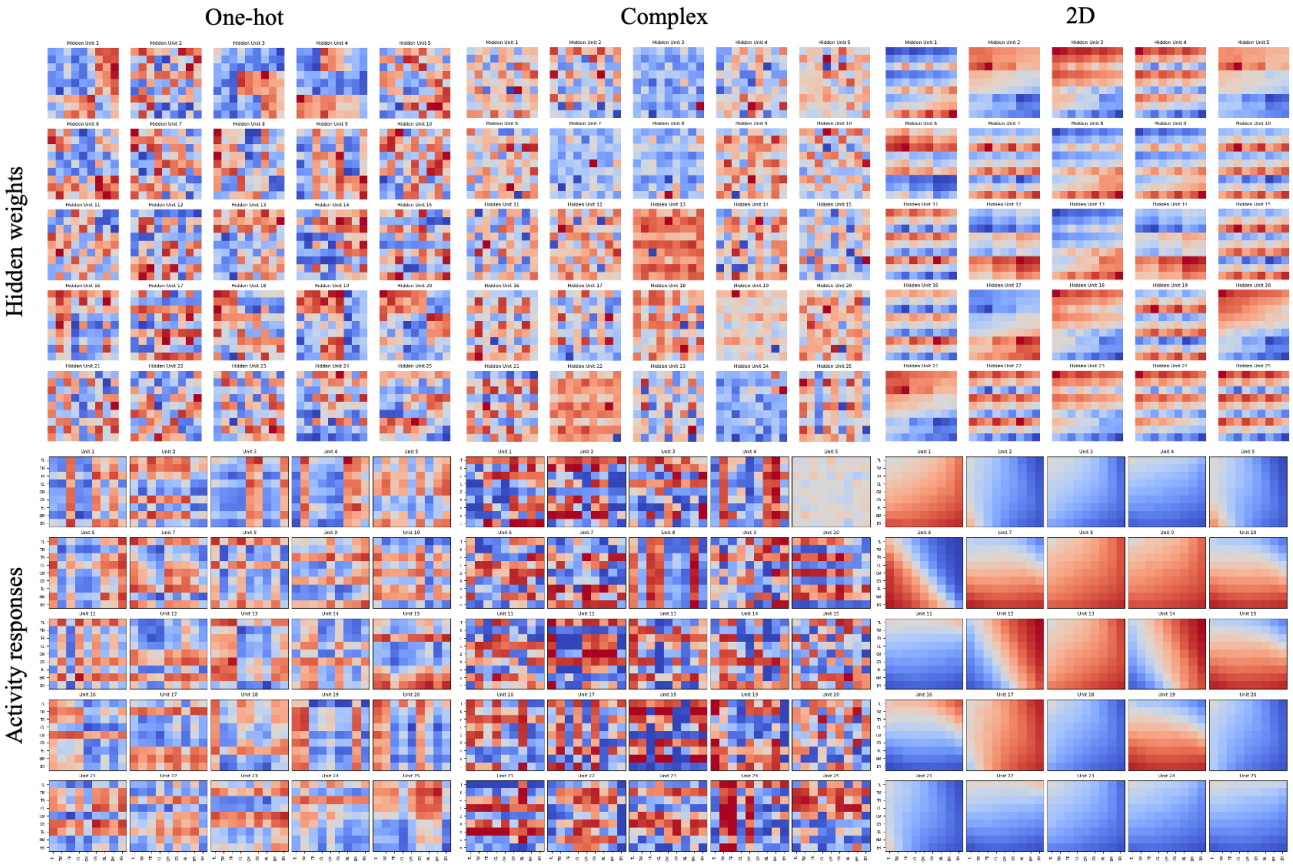

Figure 10: Hidden weights and activity responses of FF networks without biologically-inspired constraints. The one-hot and letter-like binary cases are shown, as well the 2D output setting with one-hot inputs.

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.

F.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

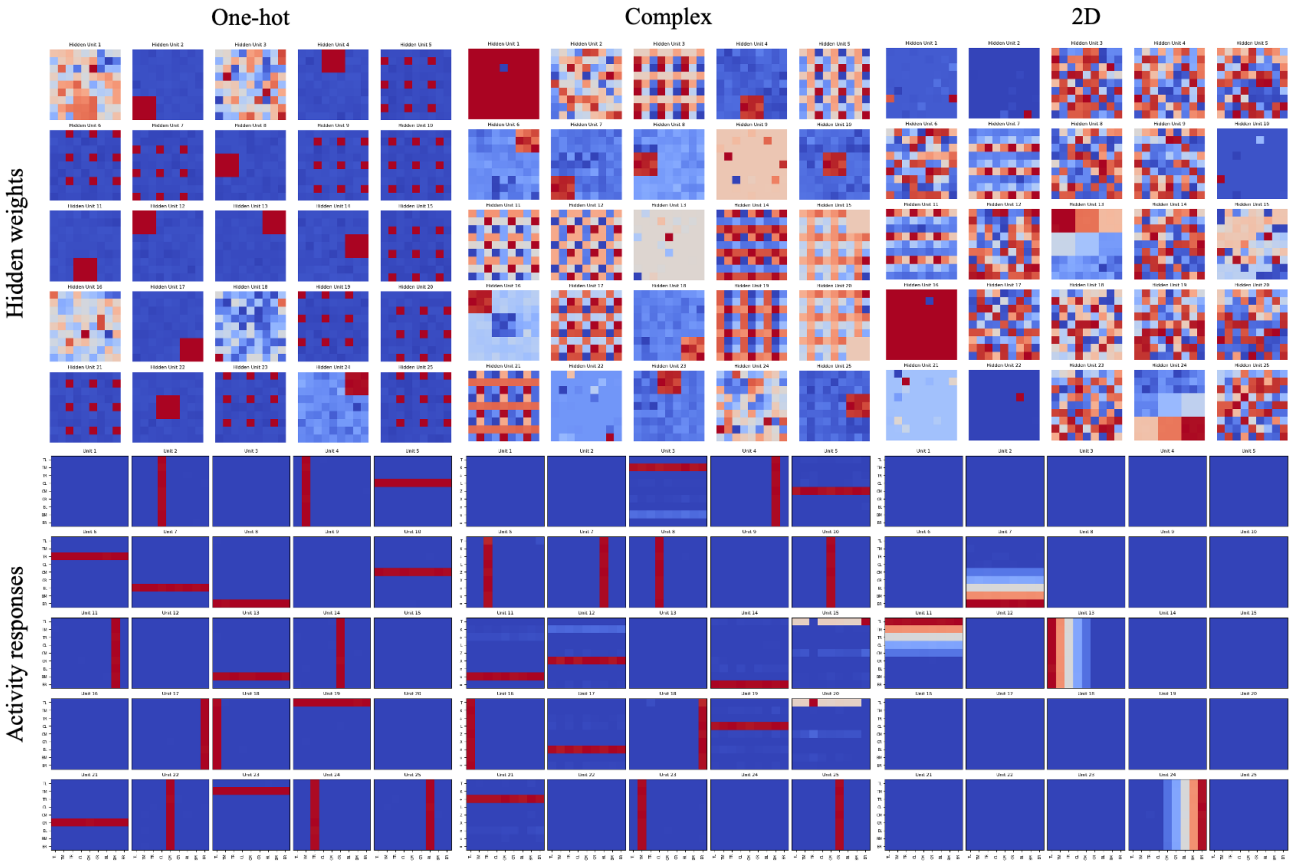

Figure 11: Weights and activity response of networks under biological constraints. The one-hot and letter-like binary cases are shown, as well the 2D output setting with one-hot inputs.

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:

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

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.

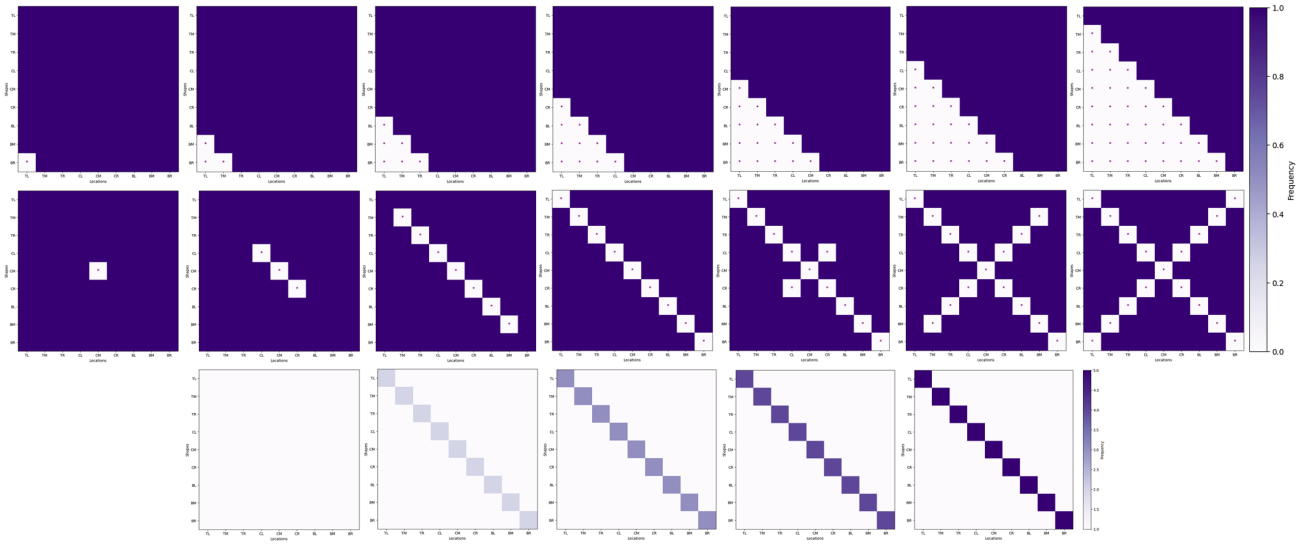

Figure 12: Increasing dropout (left to right) for both the corner-cutting (top) and diagonal (middle) cases, as well as correlation distributions (bottom). Note: asterisk denotes the absence of training data for this pair of input features.

G. Nonlinear Autoencoders

G.1. Autoencoding Sources

Three-dimensional source data is sampled from $[0,1]^3$ and discretized 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 $\lambda_{\text{reconstruct}} = 10$, $\lambda_{\text{activity energy}} = 0.1$, $\lambda_{\text{activity nonnegativity}} = 10$, and $\lambda_{\text{weight energy}} = 0.001$. Models are initialized from a He initialization scaled by 0.1 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.

G.2. Autoencoding Images

We subsample 4 out of the 9 sources in the Isaac3D dataset, fixing a single value for the other 5. We use an expressive convolutional encoder and decoder taken from the generative modeling literature. We use 8 latents, each quantised to take on 6 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.

H. Recurrent Neural Networks

H.1. Linear and Nonlinear Periodic Wave RNN Experiment Details

For linear RNN, we considered a task where RNN gets 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 2D cosine wave of the same frequencies, $y_k(t) = \cos(w_k t)$.

For nonlinear RNN, we designed a two frequency mixing task. Given two source frequencies, generating a cosine wave with a frequency of sum or difference of two requires nonlinearity; $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b), \cos(a-b) =$

$\cos(a)\cos(b) + \sin(a)\sin(b)$. In the task, 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 $\cos(w_2)$.

With the following recurrent neural network,

$$\mathbf{g}(t+1) = f(\mathbf{W}_{\text{rec}}\mathbf{g}(t) + \mathbf{W}_{\text{in}} + \mathbf{b}_{\text{rec}}) \tag{111}$$

$$\mathbf{y}(t) = \mathbf{R}\mathbf{g}(t) + \mathbf{b}_{\text{out}} \tag{112}$$

in linear RNN $f(\cdot)$ is identity and in nonlinear RNN we used $ReLU$ activation to enforce positivity condition.

For irrational output frequency ratio case, we used

$$w_1 = p\pi, w_2 = \sqrt{q}, p \sim \mathcal{U}(0.5, 4), q \ \sim \mathcal{U}(1, 10). \tag{113}$$

For rational case, we sampled

$$w_1, w_2 \in [1, 20] \cap \mathcal{Z}. \tag{114}$$

For harmonics case,

$$w_1 \in [1, 10] \cap \mathcal{Z}, w_2 = 2w_1. \tag{115}$$

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

To assess the modularity of the activity space in trained RNN, we perform Fast Fourier Transform(FFT) on each neuron's activity and measured the relative power of key frequency $w1, w2$ with respect to the sum of total power spectrum

$$C_{\text{neuron}_i, w_j} = \frac{|FFT(g_i; w_j)|}{\sum_f |FFT(g_i; f)|} \tag{116}$$

and used it as a proxy of mutual information for modularity metric introduced in eqn. 108.

H.2. Nonlinear Teacher-Student RNNs

Network details. For each teacher, the input dimension is 2 and the output dimension is 1. The two Teacher RNNs each have 2 hidden neurons, have orthogonal recurrent weights, identity input weights, and unit normed output weights. The Student RNN has input dimension 4 (two teacher inputs concatenated), output dimension 2 (two teacher outputs concatenated), and hidden dimension 64. It is initialised as per PyTorch default settings. Each 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 a different weight matrices etc).

Generating training data. We wish to have two teacher RNNs that generate 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 RNNs, we use the following procedure. 1) We sample two randomly initialised Teacher RNNs (orthogonal recurrent matrices), each with hidden dimension of two. 2) $\boldsymbol{h}_0$ is initialised as a vector of zeros. 3) With a batch size of N, we take a **single** step of each Teacher RNN (starting from 0 hidden state) assuming $\boldsymbol{i}_t = 0$. This produces a set of networks predictions, $\boldsymbol{p}_t = tanh(\boldsymbol{W}_{rec}\boldsymbol{h}_{t-1}$. 4) We then sample samples 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 each 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 each RNN being input driven, on step 4), we solve a linear sum assignment problem across all batches (N batches), so each 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 two Teacher RNNs. The learning objective is a prediction loss $|\boldsymbol{o}_t^{teachers} - \boldsymbol{o}_t^{student}|^2$ 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.

Modularity metric. The cInfoM metric for modularity is not easily applied in the recurrent setting. Instead, since we have full access to the weights in our Student RNN, and because the mapping from student hidden state to output is linear, we simply develop a modularity measure based off the output weights, $\boldsymbol{W}_{out}$. These weights have dimensions number of neurons (64) by number of outputs (2). We take the element-wise absolute value of these weights, $|\boldsymbol{W}_{out}|$, and then use the following metric (heavily following CInfoM)

$$\text{Modularity} = \frac{\frac{\sum_{n=1}^{N} \max_m |\boldsymbol{W}_{out}|_{nm}}{\sum_{nm} |\boldsymbol{W}_{out}|_{nm}} - \frac{1}{M}}{1 - \frac{1}{M}} \tag{117}$$

I. Neural Data Analysis

We were kindly given the data of (Xie et al., 2022) We used it to generate figures 4 B and E. In this section we talk through our analysis techniques. First, however we discuss methods for measuring subspace alignment.

I.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.

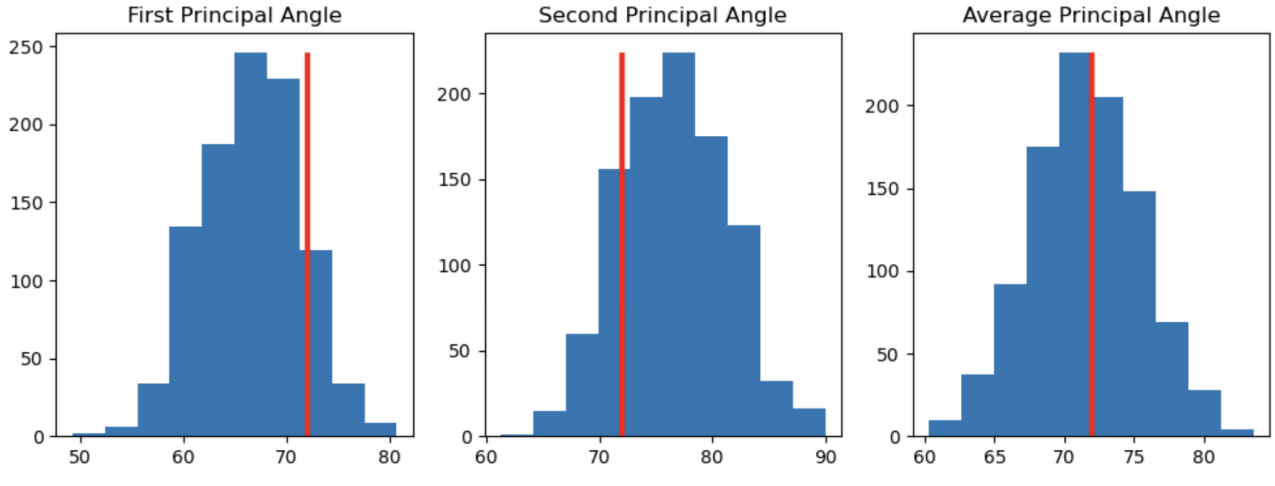

Figure 13: We create some fake data of two 2D subspaces that are aligned at 72°, i.e. both of the two principal angles are 72°. I then add a small amount of noise 1000 times and calculate the principal angles between the noisy vectors (all the vectors are length one, and I add a zero-mean gaussian noise 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 though, 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.

We counteract both these effects by using a single metric, whose bounds are nowhere near the ranges of interest. Specifically

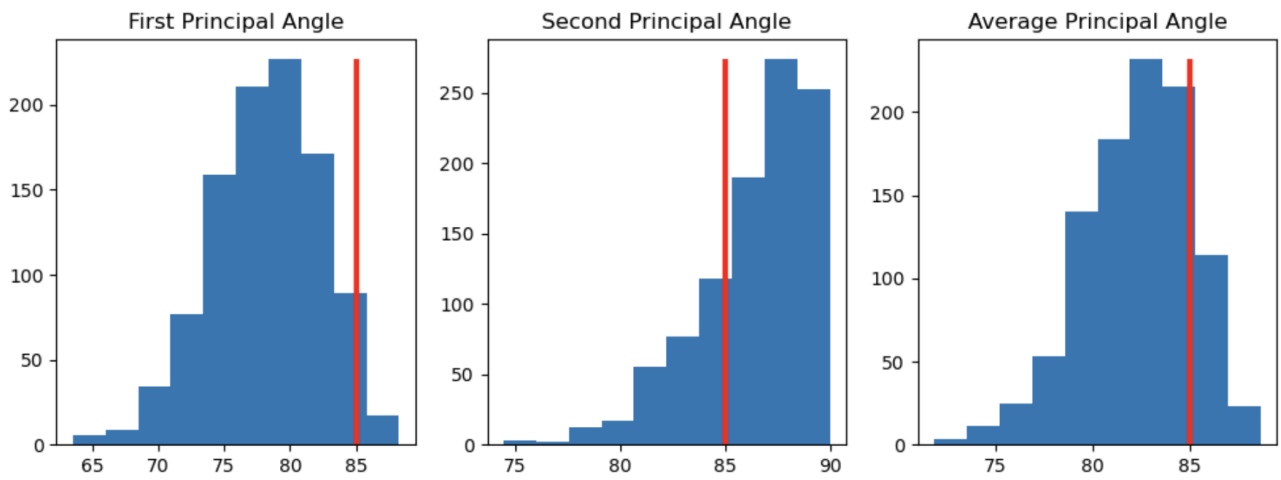

Figure 14: Same setting as figure 13, except now the true principal angle is $85°$. We can see that all estimators of the average principal angle are shifted downwards.

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:

$$\rho = \langle \text{corr}(\boldsymbol{g}(c,1) - \langle \boldsymbol{g}(c',1)\rangle_{c'}, \boldsymbol{g}(c,2) - \langle \boldsymbol{g}(c',2)\rangle_{c'}\rangle_c \tag{118}$$

Where the correlation is over neurons. This is 0 when subspaces are orthogonal, and avoids both pitfalls. Further, we don't have access to the data of Panichello & Buschman, so we are fortunate that they report this metric, as well as angles. We therefore move to extract the same metric from the data of Xie.

I.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., 2022) 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 & Buschman, 2021) and simulations (Botvinick & Plaut, 2006; Whittington et al., 2023a)) 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., 2022) 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:

$$\left\langle \boldsymbol{g}\left(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\right) - \boldsymbol{g}\left(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\right)\right\rangle_{\theta_2} \tag{119}$$

Equal to:

$$\frac{Q-2}{Q}\left\langle \boldsymbol{g}\left(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\right) - \boldsymbol{g}\left(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\right)\right\rangle_{\theta_2\neq\theta_1,\theta_1'} + \frac{1}{Q}\underbrace{\left(\boldsymbol{g}\left(\begin{bmatrix}\theta_1\\ \theta_1\end{bmatrix}\right) - \boldsymbol{g}\left(\begin{bmatrix}\theta_1'\\ \theta_1\end{bmatrix}\right) + \boldsymbol{g}\left(\begin{bmatrix}\theta_1\\ \theta_1'\end{bmatrix}\right) - \boldsymbol{g}\left(\begin{bmatrix}\theta_1'\\ \theta_1'\end{bmatrix}\right)\right)}_{\text{Error Term}} \tag{120}$$

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:

$$\left\langle \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\Big)\right\rangle_{\theta_2} = \boldsymbol{g}_1(\theta_1) - \boldsymbol{g}_1(\theta_1') \tag{121}$$

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!

$$\boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_1\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_1\end{bmatrix}\Big) + \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_1'\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_1'\end{bmatrix}\Big) = 2(\boldsymbol{g}_1(\theta_1) - \boldsymbol{g}_1(\theta_1')) \tag{122}$$

Hence, plugging these into equation 119:

$$\left\langle \boldsymbol{g}\Big(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\Big) - \boldsymbol{g}\Big(\begin{bmatrix}\theta_1'\\ \theta_2\end{bmatrix}\Big)\right\rangle_{\theta_2 \neq \theta_1, \theta_1'} = \boldsymbol{g}_1(\theta_1) - \boldsymbol{g}_1(\theta_1') \tag{123}$$

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 118, this result is reported in Fig 4E. We form error bars by subsampling by half the number of neurons 200 times and estimating the mean and variance over 200 simulations.

To make the 2D subspace plots as in Figure 4B we perform PCA on the set of difference vectors and extract a 2D subspace. We then reconstruct from the difference vectors an estimate of $\boldsymbol{g}_s(\theta)$ for each cue, θ, and sequence element, s. We project each of these estimates into each of the subspaces. If the subspaces are orthogonal the projection is zero, and the other projection of points from one subspace to another should all fall at the origin, else the subspaces are somewhat aligned.

I.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. 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.

I.4. Subspace Sizes

Finally, we estimate the relative size of the two subspaces in the data by taking the average norm of our estimates of the vectors within each subspace. We replicated the finding that the encoding of the cue presented first is larger than that presented second, with a ratio of encoding sizes around 1.16

J. RNN Models of Neuroscience Tasks

We now talk through the biologically inspired linear RNNs used in sections 5 and 6.

J.1. Xie Task, Model, 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:

$$\boldsymbol{g}_{t+1} = \boldsymbol{W}_{\text{rec}}\boldsymbol{g}_t + \boldsymbol{W}_{\text{in}}\boldsymbol{e}_{t,i_t} + \boldsymbol{b} \qquad \boldsymbol{e}_{t,i_{t-3}} = \boldsymbol{W}_{\text{out}}\boldsymbol{g}_t + \boldsymbol{b}_{\text{out}} \tag{124}$$

$\boldsymbol{g}_0 = 0$. We train the networks using gradient descent to perfectly solve the task while ensuring the representation is positive. Further, we penalise the 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 I. We then used three different sequence structures: sampling both sequence elements uniformly; sampling them without replacement but otherwise uniformly, as in the experiments; or, as a test of the effect of correlations, sampling sequences in which the two dots are diametrically opposite one another twice as often as all other sequences.

We analysed the resulting delay period activity using the techniques discussed in Appendix I. 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, on networks trained on correlated data (the only ones we trained used the data-driven memory encoding), the subspaces were slightly aligned, but less so than via sampling without replacement, which are roughly equivalent transformations. Finally, in all 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 15), while the one-hot (not shown) and data driven codes (fig 4C, 4E) do not. 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

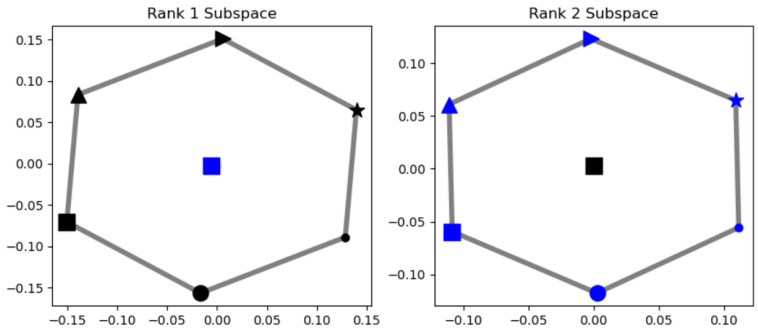

Figure 15: We estimate the two-dimensional subspaces encoding, the two subspaces are basically orthogonal, the circular mean of the 2 principal angles between the subspaces is 89.95°

by looking at the joint distribution of the encoded memories, figure 16. The required inequality constraint is not satisfed, while it is for 1-hot memories (Appendix A.6.1). The neural data inspired encoding presumably lies closer to the 1-hot end of the spectrum.

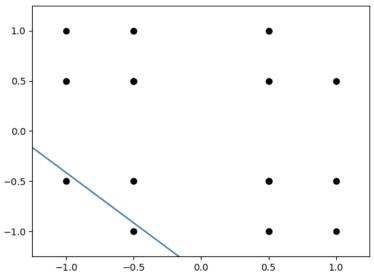

Figure 16: We plot the joint distribution of one dimension of each of the memories against one another for the 2D coding. The x axis is cosine of the first sequence element, the y axis is the second. Sampling without replacement removes the two extreme corner points (but not the whole diagonal, as though 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.

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

J.2. 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, without access to neural data we assume the memory encoding is the equivalent of the 2D one from the preivous section.

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.

Second, 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:

$$\boldsymbol{W}_{\text{top}}\boldsymbol{g}_0(c_{\text{top}}, c_{\text{bottom}}) + \boldsymbol{b}_{\text{top}} = \boldsymbol{g}_1(c_1, c_2) \quad \boldsymbol{R}\boldsymbol{g}_1(c_{\text{top}}, c_{\text{bottom}}) + \boldsymbol{b}_{\text{out}} = \boldsymbol{e}_{c_{\text{top}}} \tag{125}$$

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; 2023a;b).

Number of Samples: in the real experiment Panichello & Buschman present a colours drawn from a continous 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 are 4 points sampled evenly from the unit circle. Other than this change the sampling schemes are the same.

To generate figures 4I, J and K we follow the same process as described in Appendix I. To generate figure 4H we follow the original methods of Panichello and Buschman. We make some final interesting (to us!) observations. First, the top and bottom colour subspaces are equally sized, unlike in (Xie et al., 2022), 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., 2022), the prediction shown in figure 16 changes. With four data points even 2D encodings are missing the appropriate corner 17!

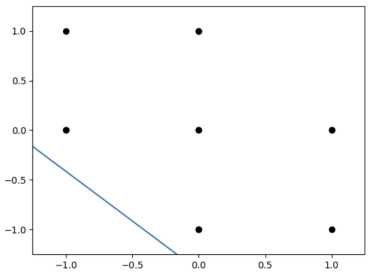

Figure 17: As in 16, we plot the joint distribution of one dimension of each of the memories against one another for the 2D coding. The x axis is cosine of the first sequence element, the y axis is the second. Since there are now only 4 settings of the angle we can see sufficient corner has been removed to predict mixing, as we see in simulations (Fig 4K and 4J).

This matches the finding presented in figures 4K and J, that when sampled without replacement the subspaces align, since a sufficiently large corner is missing.

Before wrapping up, one might wonder why correlations induce alignment in the Xie task but not the Panichello and Buschman one? Further, is this not the opposite of what the theory predicts? (Theorem 2.2?) An answer to the second question is that the theory only concerns the modularisation of scalar variables, and in these tasks we are considering the modularisation of multidimensional memories. It is likely that an update to theory would be able to precisely explain these effects. Promising evidence in that direction is that the lower dimensional the memory encoding (e.g. 2D memories) the more they match our theoretical predictions. The only break in our unidimensional source intuition comes from figure 4E, where introducing correlations for range independent data caused alignment. This simulation used memories that were each 5 dimensional, further from the original statement of the theory.

Finally, there remains one discrepancy between the neural data and our theory and models. 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.

J.3. Warden & Miller Task, and Model

In the task two images sampled without replacement from a set of four are shown to the animal with delay periods in between. The animal then has to report its memory of the images in a couple of different ways.

We use the same linear RNN as before, but with three differences. First, we include a delay period between the first and second cue, and the second cue and the report time. Second, to take into account the heterogenous nature of the report, we simply ask the network to output 2 4-dimensional one-hot vectors, corresponding to the images presented first and second. Third, we use a different weight update matrix during delay times and cue presentation times. So at times C1 and C2 (fig 5C, F, G) we use one recurrent weight and bias, during the delay period we use another. This was to enable such a simple linear system to solve the task in a minimally simple way.

To create the plots we perform the same analysis as (Mikkelsen et al., 2023), we average the neural firing rates at each timepoint according to which stimulus was presented first or second. We find many neurons that code for stimulus-time conjunctions, and fewer that code for only time, and they tended to fire around cue times, as shown in the Figure 4F. Note these averages do not take account of the correlations between the two images, introduced by sampling without replacement. As such, the tuning curves in figure 4F cannot be interpreted as evidence of mixed selectivity: in fact these neurons are modular, they respond only to a single stimulus, the ohter bumps are induced by the task structure. Since the sequences are sampled with replacement our networks have other neurons that are linearly mixed selective to the two cues, as you would expect from our theory and the preceeding discussion.

J.4. Missing Variables Task, and Model

Our final biological RNN model is inspired by the entorhinal literature and linear network models of it (Dorrell et al., 2023; Whittington et al., 2020; 2023a;b). In each trial we position 3 objects in a 3x3 periodic environment. The RNN receives one special input only at timepoint 0, a 27-dimensional 3-hot vector that tells it where each of the 3 objects are relative to its current position. 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:

$$\boldsymbol{g}_t = \boldsymbol{W}_{a_t}\boldsymbol{g}_{t-1} + \boldsymbol{b}_{a_t} \tag{126}$$

At each timestep it has to output, via an affine readout, 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 (fig 4J). However, pretend we did not know the third object existed. Perhaps it is a location the mouse cares about, but no experimenter could ever be expected to guess, such as its favourite corner, toilet spot, or scratching post - all good things to keep oriented in your mind. 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.

References

Aso, Y., Hattori, D., Yu, Y., Johnston, R. M., Iyer, N. A., Ngo, T.-T., Dionne, H., Abbott, L., Axel, R., Tanimoto, H., et al. The neuronal architecture of the mushroom body provides a logic for associative learning. *elife*, 3:e04577, 2014.

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

Barack, D. L. and Krakauer, J. W. Two views on the cognitive brain. *Nature Reviews Neuroscience*, 22(6):359–371, 2021.

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

Bernardi, S., Benna, M. K., Rigotti, M., Munuera, J., Fusi, S., and Salzman, C. D. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4):954–967, 2020.

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

Boyle, L. M., Posani, L., Irfan, S., Siegelbaum, S. A., and Fusi, S. Tuned geometries of hippocampal representations meet the computational demands of social memory. *Neuron*, 2024.

Dang, W., Jaffe, R. J., Qi, X.-L., and Constantinidis, C. Emergence of nonlinear mixed selectivity in prefrontal cortex after training. *Journal of Neuroscience*, 41(35):7420–7434, 2021.

Dang, W., Li, S., Pu, S., Qi, X.-L., and Constantinidis, C. More prominent nonlinear mixed selectivity in the dorsolateral prefrontal than posterior parietal cortex. *Eneuro*, 9(2), 2022.

Dorrell, W., Latham, P. E., Behrens, T. E., and Whittington, J. C. Actionable neural representations: Grid cells from minimal constraints. In *The Eleventh International Conference on Learning Representations*, 2023.

Driscoll, L., Shenoy, K., and Sussillo, D. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *bioRxiv*, pp. 2022–08, 2022.

Duncker, L., Driscoll, L., Shenoy, K. V., Sahani, M., and Sussillo, D. Organizing recurrent network dynamics by task-computation to enable continual learning. *Advances in neural information processing systems*, 33:14387–14397, 2020.

Dunion, M., McInroe, T., Luck, K. S., Hanna, J., and Albrecht, S. Conditional mutual information for disentangled representations in reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

El-Gaby, M., Harris, A. L., Whittington, J. C., Dorrell, W., Bhomick, A., Walton, M. W., Akam, T., and Behrens, T. E. A cellular basis for mapping behavioural structure. *bioRxiv*, pp. 2023–11, 2023.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., and Olah, C. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

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

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

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

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

Hsu, K., Dorrell, W., Whittington, J., Wu, J., and Finn, C. Disentanglement via latent quantization. *Advances in Neural Information Processing Systems*, 36, 2023.

Hsu, K., Hamid, J. I., Burns, K., Finn, C., and Wu, J. Tripod: Three complementary inductive biases for disentangled representation learning. In *International Conference on Machine Learning*, 2024.

Hubel, D. H. and Wiesel, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

Høydal, Ø. A., Skytøen, E. R., Andersson, S. O., Moser, M.-B., and Moser, E. I. 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.

Jarvis, D., Klein, R., Rosman, B., and Saxe, A. On the specialization of neural modules. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Klindt, D., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

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

Lanore, F., Cayco-Gajic, N. A., Gurnani, H., Coyle, D., and Silver, R. A. Cerebellar granule cell axons support high-dimensional representations. *Nature neuroscience*, 24(8):1142–1150, 2021.

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

Lee, J. H., Mannelli, S. S., and Saxe, A. Why do animals need shaping? a theory of task composition and curriculum learning. *arXiv preprint arXiv:2402.18361*, 2024.

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

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

Mikkelsen, C. A., Charczynski, S. J., Brincat, S. K., Warden, M. R., Miller, E. K., and Howard, M. W. Coding of time with non-linear mixed selectivity in prefrontal cortex ensembles. *bioRxiv*, pp. 2023–04, 2023.

Nayebi, A., Attinger, A., Campbell, M., Hardcastle, K., Low, I., Mallory, C. S., Mel, G., Sorscher, B., Williams, A. H., Ganguli, S., et al. Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. *Advances in Neural Information Processing Systems*, 34:12167–12179, 2021.

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

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

Parthasarathy, A., Herikstad, R., Bong, J. H., Medina, F. S., Libedinsky, C., and Yen, S.-C. Mixed selectivity morphs population codes in prefrontal cortex. *Nature neuroscience*, 20(12):1770–1779, 2017.

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

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

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

Roth, K., Ibrahim, M., Akata, Z., Vincent, P., and Bouchacourt, D. Disentanglement of correlated factors via hausdorff factorized support. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Saxe, A., Sodhani, S., and Lewallen, S. J. The neural race reduction: Dynamics of abstraction in gated networks. In *International Conference on Machine Learning*, pp. 19287–19309. PMLR, 2022.

Schug, S., Kobayashi, S., Akram, Y., Wolczyk, M., Proca, A. M., Von Oswald, J., Pascanu, R., Sacramento, J., and Steger, A. Discovering modular solutions that generalize compositionally. In *The Twelfth International Conference on Learning Representations*, 2024.

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

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

Soudry, D. and Speiser, D. 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.

Tye, K. M., Miller, E. K., Taschbach, F. H., Benna, M. K., Rigotti, M., and Fusi, S. Mixed selectivity: Cellular computations for complexity. *Neuron*, 2024.

Warden, M. R. and Miller, E. K. Task-dependent changes in short-term memory in the prefrontal cortex. *Journal of Neuroscience*, 30(47):15801–15810, 2010.

Whittington, J. C., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., and Behrens, T. E. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5): 1249–1263, 2020.

Whittington, J. C., Dorrell, W., Behrens, T. E., Ganguli, S., and El-Gaby, M. On prefrontal working memory and hippocampal episodic memory: Unifying memories stored in weights and activity slots. *bioRxiv*, pp. 2023–11, 2023a.

Whittington, J. C., Dorrell, W., Ganguli, S., and Behrens, T. Disentanglement with biological constraints: A theory of functional cell types. In *The Eleventh International Conference on Learning Representations*, 2023b.

Xie, Y., Hu, P., Li, J., Chen, J., Song, W., Wang, X.-J., Yang, T., Dehaene, S., Tang, S., Min, B., et al. Geometry of sequence working memory in macaque prefrontal cortex. *Science*, 375(6581):632–639, 2022.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., and Wang, X.-J. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

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