WILLIAM DORRELL - PHD UPGRADE REPORT - MAY 2023

ABSTRACT

By all accounts, no upgrade report has ever been read by anyone other than its authors.

As such, my report has been structured to try and make this exercise productive in a few ways:

- by using the committee as a sounding board for future ideas
- by thinking through my plans for my own benefit
- by asking for help with some large research-direction type questions that have arisen. Section one lists these questions in brief. If you read only one thing, read that!

In the following sections I then outline my research and plans in more typical fashion.

1 Questions to Answer in Brief

1.1 Can I just read for a couple of years?

In my PhD I've had a lot of fun, and done some hopefully helpful things. But I want to be more than the resident physicist who enjoys putting their tools in the service of neuroscience; I want to "go native": to be the neuroscientist trying to understand how the brain works, getting involved in debates, and occasionally using theory to answer them, at other times trying persuading experimentalists to do what I think are the relevant experiments.

I don't think doing another project, taking another course, or attending a conference, is the way to progress towards this goal. I think I should sit and read - ask the questions that seem most relevant, follow the literature, talk to people, and see what questions I end up with. Is that unrealistic? Does everyone have this pipe dream?

1.2 What should I do with a proto-paper on cerebellar connectivity?

One area I've studied is using the inductive bias of classification circuits as a way to interpret otherwise confusing features of biological circuits (William Dorrell, Yuffa, and P. Latham 2023) (e.g. "that nonlinearity is there because it makes these pertinent classifications easier to learn"). This work has led to a normative interpretation of some connectomic findings in cerebellar-like circuits. I have a small paper about this insight. Our approach is technically novel and better than those used in the literature, but I don't think it adds much to the existing knowledge. To make it a meaningful contribution I would have to do something interesting, I have some ideas but I'm not sure if they are worth pursuing. I would appreciate advice about whether to invest in this.

To learn more about this question read section 3.2, perhaps also section 3.3, or if you have time, all of section 3.

1.3 How far to travel down the grid cell road?

I, like about half the theorist in neuroscience, have produced a theory about grid cells. I am genuinely very proud of it, it is the only one that goes from a normative objective to multiple modules of hexagonal-ish cells - I really think it is currently the best theory on the market, and it means I have predictions coming out of my ears!

But that doesn't mean anyone does, or even should, care. At a cursory level our work seems incremental, and I have consistently underestimated how much effort to invest in outlining the relative merits of our work over other theories.

As such, I feel torn. Tim says pushing grid cell stuff isn't useful, and I kind of agree - compared to the work on prefrontal cortex that is (maybe?) helping to develop new ideas for how these areas work, the grid cell stuff is potentially just rehashing a well-understood system. On the other hand, it would be great to outline the relative merits of the different theories, and test predictions; wouldn't it be wild if precise predictions were borne out?! What should I do?!

To learn more about this question read section 4.1, perhaps also section 4.3, or if you have time, all of section 4.

1.4 What is the most interesting framing of future Disentangling work?

We have some biological constraints (low and positive firing) that encourage a representation that is a linear function of the data to disentangle data generating factors under a particular set of assumptions, where disentangling means each neuron responds to one generating factor. While constrained by linearity, the analysis makes some interesting predictions about modularisation in the brain, and the data assumptions are kind of fun - range independence and orthogonal-enough data generating factors. These ideas also make predictions about the modularising of activity in RNNs, or anywhere where you have an idea of what has to be encoded but not how to align it with the neuron basis. Finally they've also been pretty reliably empirically extended to the nonlinear ML setting.

James (Whittington) would like to write another paper about biological disentangling; quite how and what new things therefore need developing are an open questions. Something along the lines of, if these activity patterns have to be represented in a set of neurons, this is how they should modularise. Additional developments could include:

- A simple paper outlining the linear settings. Analyse the phase transitions that appear.
- Extend Andrew's theory to predict responses all the way to the neural level, can you add regularisation?
- Charge at something nonlinear?

I don't know whether this is worth doing - are there novel points that are useful to make?

To learn more about this question read section 5.1 and 5.3.

1.5 Around what theme should I write a "Neuroscience Paper"?

In my PhD I have only published in ML conferences. This forces a particular framing of the ideas, but really I think the ideas are most useful for neuroscientists. So I should write a neuro paper, there are three obvious options:

- A neuroscience grid cell paper, perhaps with many more predictions worked out, more precise comparisons to data, maybe even some experiments?
- An actionable codes paper, including grid cells, music box prefrontal cortex representations, and perhaps a third idea. See later for some suggestions.
- Something about disentangled representations via biology.

Should I be concerned about this at all? If so should I write one of these? Which? Or should I just carry on doing things, then only write something up at the end of my PhD?

To learn more about this question read section 4.1 and 4.3 for the grid cell ideas, the intro to section 4, 4.2 and 4.3 for the broader actionable ideas, and the intro to section 5 and section 5.3 for disentangling things.

1.6 How much to dive into neural networks?

I'm very interested in mechanistic interpretabilty, a field that is effectively neuroscience for neural networks. They try to understand how a neural network works at a level where you could hand-code its weights based on interpretable rules and recreate network function without training on data. They've made some significant progress in the last few years (Cammarata et al. 2020; Nanda et al. 2023; Olsson et al. 2022), including by providing insights that have powered improved model design (Dao et al. 2022). As the best models of visual cortex become deep neural networks, it suggests a new route for progress in neuroscience: dive into neural networks and use their complete observability to predict things in neuroscience, while being vigilant for the relevant differences between artificial and biological neural networks. In fact this happened! Someone found high-low frequency contrast cells in an artificial neural network (Schubert et al. 2021), and then they were found in the brain (Zhiwei Ding et al. 2023)!

I'm tempted to transition into this area of work more, should I??

Acknowledgements

This work took a village! Most bluntly, section 5.1 is largely James Whittington's work, section 5.2 is largely Kyle Hsu's, the data in section 4.2 is Mohamady El-Gaby's, and section 2 is a massive team project led by Emmett Thompson. Beyond that even the sections I coded myself etc. were only possible because of James, Tim, and Peter's advising; it's basically all their ideas - I just make nice plots. Finally, Maria Yuffa bravely agreed to be mentored by me, and she helped with all the stuff in section 3.1. Thanks all for making my PhD a good experience! (so far...)

Contents

2 PPSleep: Replay Detection with Point Process Models

TL;DR: Adapted a sequence-learning point process model to find replay; used it to show evidence for the replay of motor sequences in dorsolateral striatum. This has implications for the structure of replay (independent of hippocampus, and composed from distinct elements), that ongoing experiments are probing further.

Replay - internally generated patterns of neural activity - seems to play an important role in the brain: hypothesised functions include consolidation, planning in an internal model, or composing together elements of the world (Kurth-Nelson et al. 2023). Most of the evidence for this comes from the hippocampus, and its role in episodic memory (memory of episodes, like your memory of your commute this morning) - open questions include: how generalisable are these findings to other brain regions, for example to procedural memory (e.g. being able to ride a bike)? How co-ordinated are replay events across the brain? And what mechanisms are used to trigger replays with specific content at each timepoint?

Emmett Thompson in the Stephenson-Jones lab sought to answer some of these questions in the context of motor sequences in the dorsolateral striatum. He (heroically) trained mice to perform a stereotyped motor sequence, like typing your password. The task comprises poking your nose into a sequence of five ports, figure 1A. Lesions confirmed that dorsolateral striatum was required for both learning and performance of this task, and plasticity blocking experiments confirmed that offline synapse change in dorsolateral striatum was required for consolidating improvements acquired during the day, figures 1B - D. Further, additional lesion experiments demonstrated that this entire behaviour was hippocampus independent. He then recorded neural activity in motor cortex and dorsolateral striatum as mice performed the task, and as they slept before and after each session, figures 1E - F. The Sleepuence-Squad (or cluster cloggers) - Rodrigo, Clementine, Tom George, and I - were tasked with trying to find structure in this neural data, and preferably evidence for replay of that structure during sleep.

To do this we adapted a beautiful point process algorithm for sequence detection, PPSeq (A. Williams et al. 2020). PPSeq is an unsupervised model that looks for repeated instances of ordered activation of neurons (i.e. neuron 4 then neuron 3 then neuron 17 activated at roughly similar offsets from one another many times in the data). Given the number of sequences, it finds a setting of parameters that fits the most of the data. We ran it on data from expert mice performing the task and it found patterns of neural activity corresponding to different sections of the behaviour, figures 1G - I.

We then started to search for replay. We changed the algorithm (PPSeq -> PPSleep) so that you could run the model on one dataset, learn some sequences and fix their parameters, then run the model on another set of data - effectively searching not (only) for sequences that explain as much of the data as possible, but also for instances of reactivation of the fixed learnt sequences.

One huge technical advantage of this method over traditional approaches in the replay detection field was its unsupervised nature. In the hippocampus replay is usually associated with "sharp-wave ripples" - largescale, easily-detected, electrical fluctuations. This makes replay detection easy: look for a sharp-wave ripple, then try to decode the position of the animal during the ripple, and call it replay if the decoded path looks consistent with reality in some way (for example position moves continuously through the environment). These techniques don't work without the sharp-wave ripple, as the decoding algorithms tend to be too slow to run on the full dataset.

PPSleep on the other hand requires no such supervision! Perfectly suited to other brain regions where convenient biomarkers such as sharp-wave ripples are not known.

We ran PPSleep on sleeping data and found evidence for replay, figure 1K. This was confirmed by running a decoder around the timepoints PPSleep identified as putative replay events, figures 1L - M. We used a decoder from the hippocampal literature (Ólafsdóttir, Carpenter, and Caswell Barry 2016) that performs a bayesian mapping from neural firing rates to a position by creating an analogous position variable corresponding to progress in the task. This approach agreed with PPSleep, decoding continuous movement through the task variable.

This is an exciting start. A masters student, Benjamin Waked, is now taking the data analysis side of the project further, running our model on more mice, testing our conclusions, and improving the decoding baseline comparison - and we have been helping to mentor him. Further, new analysis and experiments are continuing apace. We hope to find a biomarker of for the different types of replay are happening, and to understand the role of different types of replay in consolidation of procedural memory.

TO DO: Soon I should submit a pull request to the PPSeq repo to integrate PPSleep, allowing others to go looking for replay with this method - especially since we have already had multiple requests to do this just from inside the building.

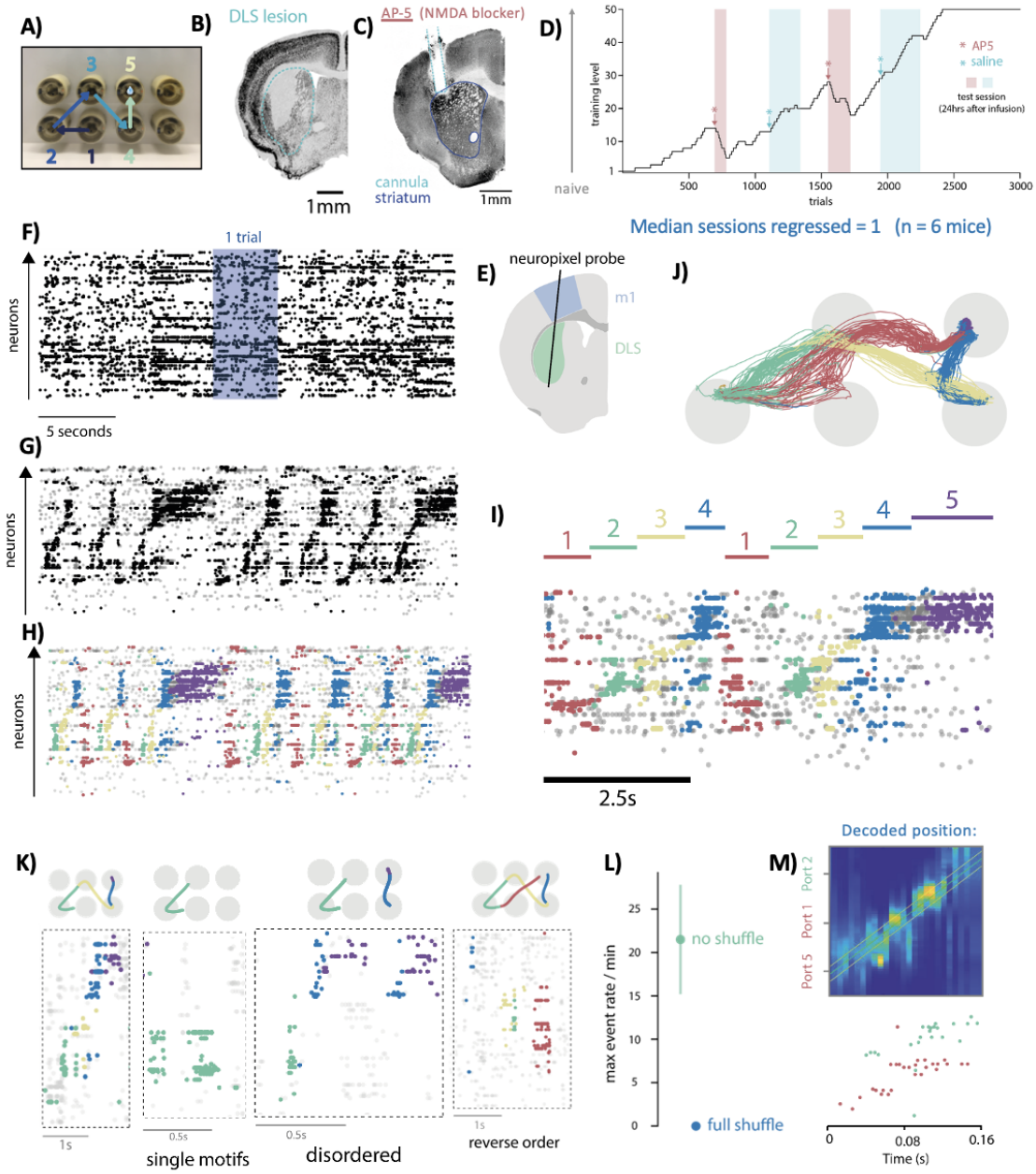

Figure 1: **A** Mice must poke in a sequence of five ports. **B** Lesions to dorsolateral striatum (DLS) removed ability to perform the task **C** Blocking plasticity in the DLS using an NMDA blocker led to **D** drops in the ability of the mouse. In fact, injecting NMDA blockers after a session tended to cause the mouse to regress to its ability the day before the injection, i.e. it blocked the consolidation of that day's memories. **E - F** Neuropixels measurements in DLS and M1. **G** Reordering the neurons using the structure found by PPSeq highlights the repeating patterns of neural activity as the mouse performs the task. **H** Colouring the spikes by sequence membership highlights the five types of sequences learnt that progress one to the other through the task. **J - I** These sequences map onto interpretable parts of the movement trajectory, with over representation of the longer lick periods at the end of each trial. **K** Running PPSleep on sleeping data found recalled sequences either of the whole sequence, single motifs, disordered motifs, or the sequence in reverse order. **L** We checked these found sequences were not spurious algorithmic fumes by shuffling the neurons, as expected this caused the discovered event rate to drop to 0. **M** Next we ran a Bayesian decoder on the PPSleep events, and found that the decoded position through sequence and PPSeq events matched well.

3 Interpreting Biological Circuits Using their Inductive Bias

Content Warning: if you are Peter L and cannot abide terms that are not precisely defined (let's ignore the fact that this applies to most useful terminology...) you can replace the phrase inductive bias with "generalisation error as a function of true label generating function" in what follows.

I find a learner's inductive bias a very neat way to understand its operation. To quickly explain the concept, lets stick with supervised learners trained on some dataset. For any dataset, there are infinitely many ways to fit the training data perfectly; some will generalise well, others won't. A particular learner will settle on a solution, and which solution is settles on is influenced by all the details that define the learner. Therefore one route to understanding why the learner looks like this and not like that (for example, why did it choose this nonlinearity not that one) is through the way this change influences the way it generalises - also known as its inductive bias.

The difficulty is that mapping between learner structure and inductive bias/generalisation error is difficult. People have managed it for kernel regression, and that's about it. Therefore we take two approaches to using this concept to understand biological circuits.

- We find circuits that look like kernel regression, and use this similarity to understand these circuits inductive bias. This is section 3.2.

  I am unsure what to do with this work, this is question 1.2, help appreciated!
- We design a more flexible tool that uses neural networks etc. to map from basically any learner to generalisation error. This is section 3.1.

3.1 Meta-Learning the Inductive Bias of Simple Neural Circuits

TL;DR: The structure of a supervised learner determines which labellings of data it finds easy to learn (i.e. reach generalisation error x quickly) and which it finds hard. This can be an interesting explanatory window - "perhaps that circuit feature is like this and not like this to allow these pertinent tasks to be learnt easily". We design a neural network approach that meta-learns functions that a given learner finds easy-to-generalise, flexible permitting this link between structure and function.

The text here is from William Dorrell, Yuffa, and P. Latham 2023

3.1.1 Introduction

Generalising to unseen data is a fundamental problem for animals and machines: you receive a set of noisy training data, say an assignment of valence to the activity of a sensory neuron, and must fill in the gaps to predict valence from activity, Fig. 1A. This is hard since, without prior assumptions, it is completely underconstrained. Many explanations or hypotheses perfectly fit any dataset (D. Hume 1748), but different choices will lead to wildly different outcomes. Further, the training data is likely noisy; how you choose to sift the signal from the noise can heavily influence generalisation, Fig. 1B.

Generalising requires prior assumptions about likely explanations of the data. For example, prior belief that small changes in activity lead to correspondingly small changes in valence would bias you towards smoother explanations, breaking the tie between options 1 and 2 in Fig. 1A. It is a learner's inductive bias that chooses certain, otherwise similarly well-fitting, explanations over others.

The inductive bias of a learning algorithm, such as a neural network, can be a powerful route to understanding in both Machine Learning and Neuroscience. Classically, the success of convolutional neural networks can be attributed to their explicit inductive bias towards translation-invariant classifications (LeCun et al. 1998), and these ideas have since been very successfully extended to networks with a range of structural biases (Bronstein et al. 2021). Further, many network features have been linked to implicit regularisation of the network, such as the stochasticity of SGD (Mandt, Hoffman, and Blei 2017), parameter initialisation (Glorot and Bengio 2010), early stopping (Hardt, Recht, and Y. Singer 2016), or low rank biases of gradient descent Gunasekar et al. 2017).

In neuroscience, the inductive bias has been used to assign normative roles to representational or structural choices via their effect on generalisation. For example, the non-linearity in neural network models of the cerebellum has been shown to have a strong effect on the network's ability to generalise functions with different frequency content (M. Xie et al. 2022). Experimentally, these network properties vary across the cerebellum, hence this work suggests that each part of the cerebellum may be tuned to tasks with particular smoothness properties. This is exemplary of a spate of recent papers applying similar techniques to visual representations (Bordelon, Canatar, and Pehlevan 2020; Pandey et al. 2021), mechanosensory representations (Pandey et al. 2021), and olfaction (K. D. Harris 2019).

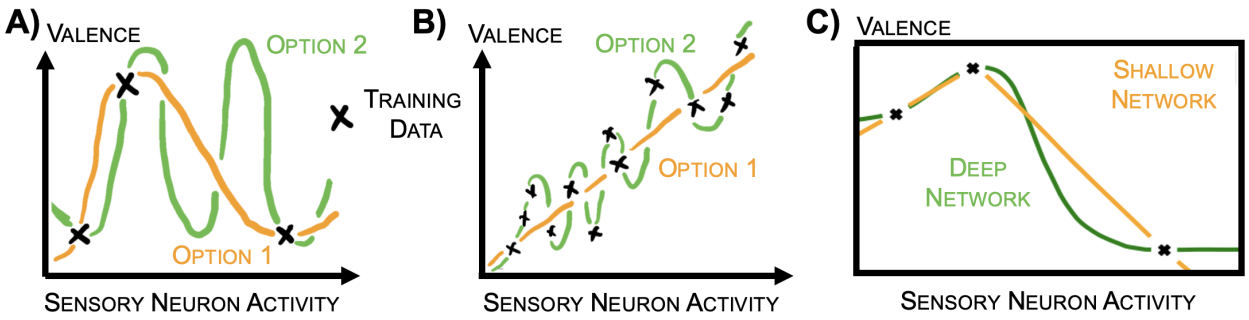

Figure 2: **Generalisation Requires Prior Assumptions. A:** The same dataset is perfectly fit by many functions. **B:** Different assumptions about signal quality lead to different fittings. **C:** Training a 2 (shallow) or 8 (deep) layer ReLU network on the same dataset leads to different generalisations.

Despite the potential of using inductive bias to understand neural circuits, the approach is limited, since mapping from learning algorithms to their inductive bias is highly non-trivial. Numerous circuit features (learning rules, architecture, non-linearities, etc.) influence generalisation. For example, training two simple ReLU networks of different depth to classify three data points leads to different generalisations for non-obvious reasons, Fig. 1C. In a few cases analytic bridges have been constructed that map learning algorithms to their inductive bias. In particular, the study of kernel regression, an algorithm that maps data points to a feature space in which linear regression to labels is then performed (Bordelon, Canatar, and Pehlevan 2020; Simon, Dickens, and DeWeese 2021; Sollich 1998), has been influential: all the cited examples of understanding in neuroscience via inductive bias have used this bridge. However, it severely limits the approach: most biological circuits cannot be well approximated as performing a fixed feature map then linearly regressing to labels!

In this project, we developed a flexible neural network approach that is able to meta-learn the inductive bias of essentially any differentiable supervised learning algorithm. It follows a meta-learning framework (Vanschoren 2019): an outer neural network (the meta-learner) assigns labels to a dataset, this labelled dataset is then used in the inner optimisation to train the inner neural network (the learner). The meta-learner is then trained on a meta-loss which measures the generalisation error of the learner to unseen data. Through gradient descent on the meta-loss, the meta-learner meta-learns to label data in a way that the learner finds easy to generalise. These easy-to-generalise functions form a description of the inductive bias, since easy-to-generalise functions are those that learners generalise appropriately from few training points. In sum, networks are inductively biased towards easy-to-generalise functions.

In the following sections we describe our scheme, and validate it by comparing to the known inductive biases of linear and kernel regression. We then extend it in several ways. First, networks are inductively biased towards areas of function space, not single functions. Therefore we learn a set of orthogonal functions that a learner finds easy to generalise, providing a richer characterisation of the inductive bias. Second, we introduce a framework that asks how a given design choice (architecture, learning rule, non-linearity) effects the inductive bias. To do that, we assemble two networks that differ only by the design choice in question, then we meta-learn a function that one network finds much easier to generalise than the other. This can be used to explain why a particular circuit feature is present. We again validate both schemes against linear and kernel regression. Finally we show our tool's flexibility in a series of more adventurous examples: we validate it on a challenging differentiable learner (a spiking neural network); we show it works in high-dimensions by meta-learning MNIST labels; and we highlight its explanatory power for neuroscience by using it to normatively explain patterns in recent connectomic data via their inductive bias.

3.1.2 A Neural Network to Meta-Learn Inductive Biases

Our main contribution is a meta-learning framework for extracting the inductive bias of differentiable learning algorithms, Fig. 2A, that we describe in this section. In the outer-loop a neural network, the meta-learner, assigns labels to inputs sampled from some distribution, hence creating the real-world function that our circuit of interest will try to learn. The inner-loop learning algorithm, the learner, is the circuit whose inductive bias we want to extract; for example, a biological sensory processing circuit that assigns valences to inputs. When provided with a training dataset of inputs and labels the learner adjusts its parameters according to its internal learning rules. Then the generalisation error of the trained learner is measured on a held-out test set, and this is used as the meta-loss to train the meta-learner. This process repeats, retraining the learner at every iteration and iteratively developing the meta-learner's weights, until the

meta-learner is labelling the data in a way that the learner finds easy to generalise after training on a few datapoints (we used around 30). Thus, the meta-learner has extracted a function towards which the learner is inductively biased.

As just outlined, the meta-learner will find the easiest-to-generalise function, usually the one that assigns all inputs the same label. To avoid this trivial function, we introduce a term in the meta-loss that forces the distribution of labels to take a particular (non-constant) form. Specifically, it measures the Sinkhorn divergence between the meta-learner's label distribution and a uniform distribution from -1 to 1 (other divergences also work, Appendix B of William Dorrell, Yuffa, and P. Latham 2023). The full pseudocode is in Algorithm 1.

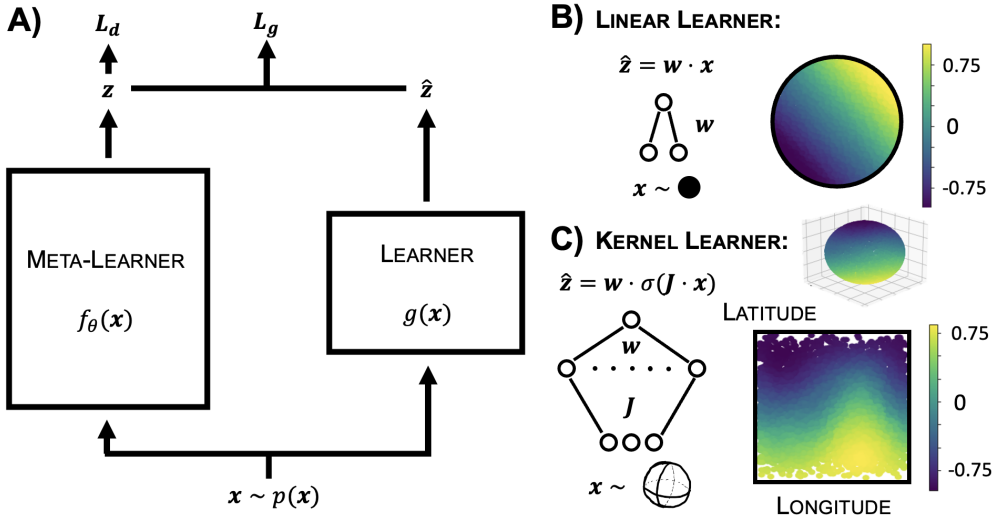

Figure 3: **Meta-Learning the Inductive Bias. A:** The meta-learner labels a dataset which is used to train the learner. Gradient descent is performed on a loss made of the learner's generalisation error on unseen data ($\mathcal{L}_g$), and the Sinkhorn divergence between the meta-learner's label distribution and a target distribution ($\mathcal{L}_d$), here choosen to be uniform from -1 to 1. **B:** The meta-learner learns a linearly separable labelling of data sampled from a circle for a ridge regression learner. **C:** For a kernel regression learner and data sampled from the surface of a sphere, the meta-learner's labelling is very close to the predicted spherical harmonic (99% of norm within first order harmonics).

Algorithm 1: Pseudocode for Meta-Learning the Learner's Inductive Bias

Initialise meta-learner: $f_\theta(\boldsymbol{x})$
while *Step count* $<$ *Total* **do**
  Generate dataset from input distribution: $\boldsymbol{x} \sim p(\boldsymbol{x})$
  Label using metalearner: $z = f_\theta(\boldsymbol{x})$
  Split inputs and labels into test and train datasets: $\mathcal{D}_{Tr}$ & $\mathcal{D}_{Te}$
  Train leaner using $\mathcal{D}_{Tr}$ giving trained learner network: $g(\boldsymbol{x})$
  Label $\mathcal{D}_{Te}$ using trained learner: $\hat{z} = g(\boldsymbol{x})$
  Compute the generalisation error of the leaner: $\mathcal{L}_g = \sum_i (z_i - \hat{z}_i)^2$
  Compute the Sinkhorn Divergence of metalearner's labels from uniform $[-1, 1]$: $\mathcal{L}_d$
  Take θ gradient step on meta-loss: $\mathcal{L} = \mathcal{L}_g + \mathcal{L}_d$
end

Our meta-learner must fit a function that the learner can generalise. To enable the meta-learner to learn all functions the learner might plausibly generalise well, its function class could usefully be a superset of the learner's. Therefore, we choose the meta-learner's architecture to be a slightly larger version of the learner's (though, beyond this, our findings appear robust, Appendix D of William Dorrell, Yuffa, and P. Latham 2023).

We validate our scheme by meta-learning sensible functions for linear and kernel learners, whose inductive biases are known. First, for ridge regression on data sampled from a 2D circle the meta-learner assigns linearly separable labels, Fig. 2B; exactly the labels linear circuits easily generalise.

Next, we meta-learn kernel ridge regression's inductive bias. Kernel regression involves projecting the input data through a fixed mapping to a feature space (e.g. the last hidden layer of a fixed neural network) and performing linear regression from feature space to labels. (Bordelon, Canatar, and Pehlevan 2020 show that the inductive bias of kernel regression can be understood through the kernel eigenfunctions ($\{v_i(\boldsymbol{x})\}$ with eigenvalue $\{\lambda_i\}$). These are defined on input distribution $p(\boldsymbol{x})$ via a kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ that measures the similarity of two inputs in feature space:

$$\int k(\boldsymbol{x}, \boldsymbol{x}')v_i(\boldsymbol{x}')dp(\boldsymbol{x}') = \lambda_i v_i(\boldsymbol{x}). \tag{1}$$

The algorithm is inductively biased towards higher eigenvalue eigenfunctions; i.e., fewer training points are needed to reach a given generalisation error when fitting high vs. low eigenvalue eigenfunctions. General functions can be understood by projecting onto the eigenbasis. Hence our meta-learner, in searching for kernel regression's easiest-to-generalise non-constant function, should choose the highest eigenvalue eigenfunction.

To test this, we meta-learn the inductive bias of a two-layer neural network with fixed first layer weights. We sample data uniformly from the sphere and randomly connect a large hidden layer of ReLU neurons to the three input neurons. The elements of this random weight matrix are drawn *iid* from a standard normal, and the learning algorithm performs ridge regression on the hidden layer activities. Previous work has analytically derived the kernel for this network, and computed its eigenfunctions (Cho and Saul 2009; Mairal and Vert 2018, which are spherical harmonics. The higher the frequency of the spherical harmonic the lower its eigenvalue. Matching this, our network meta-learns one of the set of lowest frequency spherical harmonics, Fig. 2C.

3.1.3 Meta-Learning Areas of Function Space

Having validated our tool on some simple test cases, we now extend it to find a richer characterisation of the inductive bias. A given learning algorithm is inductively biased towards areas of function space, not just one particular function. To gain access to this larger space, we learn a series of meta-learners. The first of these is exactly as described above, then we iteratively introduce additional meta-learners. To ensure each meta-learner learns a new aspect of the inductive bias we add a term to the meta-loss that penalises the square of the dot product between the current meta-learner's labelling and that of all the previously trained meta-learners, Fig. 3A. On a dataset $\{\boldsymbol{x}_n\}$:

$$\mathcal{L}_{\text{Orthog}} = \sum_i \left(\sum_n f_{\theta_i}(\boldsymbol{x}_n) f_{\theta'}(\boldsymbol{x}_n) \right)^2 \tag{2}$$

for each previous meta-learners $f_{\theta_i}(\boldsymbol{x})$ and the current meta-learner $f_{\theta'}(\boldsymbol{x})$. From the learner's perspective nothing has changed, at each meta-step it simply learns to fit the meta-learner that is currently being trained. But each additional meta-learner must discover an easy-to-generalise function that is orthogonal to all previous meta-learners.

We again validate this scheme on linear and kernel regression. For linear regression of 2D data the meta-learners learn two orthogonal linear labellings, then a third orthogonal function that the learner struggles to generalise, as expected, Fig. 3B. For the kernel regression network we described previously theory predicts that the meta-learners should learn the eigenfunctions in decreasing order of their eigenvalue. We find this to a good approximation, Fig. 3C, learning approximations to the three first order spherical harmonics, and then three approximations to second order harmonics.

For linear classifiers (e.g. linear and kernel regression) the full set of orthogonal functions explains the entire inductive bias, since the average generalisation error of any target function is the sum of the generalisation errors of each the kernel eigenfunctions weighted by projection of the target function onto that eigenfunction. This will not in general be true, since for a non-linear classifier the generalisation error on a function $f(\boldsymbol{x}) = f_a(\boldsymbol{x}) + f_b(\boldsymbol{x})$ does not equal the generalisation error on $f_a(\boldsymbol{x})$ plus that on $f_b(\boldsymbol{x})$. Nonetheless, we expect the set of orthogonal functions will still be a helpful guide to a network's inductive bias, even for non-linear classifiers.

3.1.4 Finding the Effect of Design Choices on the Inductive Bias

Our work is motivated by the desire to understand how design choices in learning algorithms - such as architecture, learning rule, and non-linearities - lead to downstream generalisation effects, particularly in biological networks. One additional setting which we have found useful is to compare two networks with some architectural difference between them, and learn functions that one of the networks finds much easier to generalise than the other. In this way, we can build intuition for the impact of design features on the inductive bias. To illustrate this we again create a meta-learner that labels data, but this time the labels are used to train two learners. We then train the meta-learner so that one learner (the chosen student) is much better at generalising than the other (neglected student). This is done by minimising the generalisation errors of the chosen student minus the neglected student, Fig. 4A. Validating this approach on well understood algorithms, we show that it can find functions that a kernel regression algorithm is able to learn better than linear regression, Fig. 4B, i.e. a non-linearly separable function.

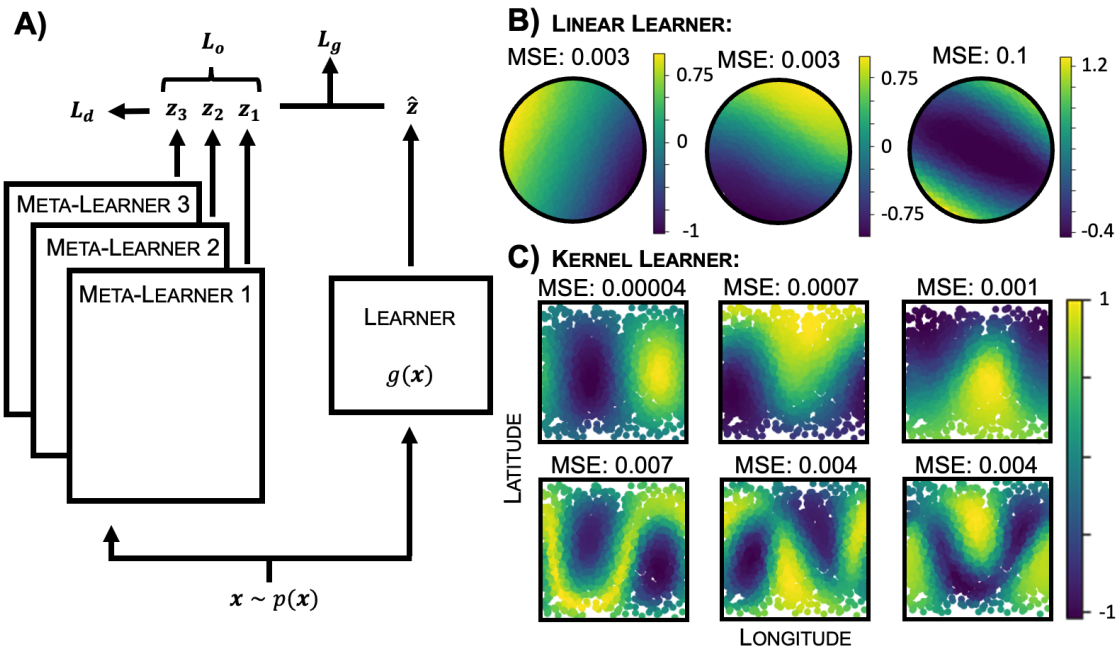

Figure 4: **Meta-Learning Many Functions. A:** We learn many meta-learners, each of which has to label orthogonally to all previous meta-learners. **B:** For a linear learner the meta-learners learn two orthogonal linear functions and an orthogonal but hard to learn third function. **C:** For a kernel learner we learn 6 meta-learners, the first 3 approximate well first order spherical harmonics (96% norm overlap), and the next 3 second order spherical harmonics (91% norm overlap), as predicted.

This illustrates some of the games that can be played in this setting. For example, you could play a co-operative game, in which you meta-learn a function that a set of learners all find easy to generalise, and each learner could have different connectivity matrices to match the distribution in real animals, ensuring the tool does not over-fit to some specific details. However as the losses become more complex training becomes harder, for example this adversarial setting between chosen and neglected student is hard to make robust if the two learners are relatively similar.

3.1.5 Meta-Learning Applied to More Complex Learning Algorithms

So far we have developed and tested a suite of tools for extracting the inductive bias of learning algorithms. We now apply our tools to networks whose inductive bias cannot be understood analytically. Specifically: we show our method works on a challenging differentiable learner, a spiking neural network; we validate our method on a high-dimensional MNIST example; and we illustrate how our tool can give normative explanations for biological circuit features, by meta-learning the impact of connectivity structures on the generalisation of a model of the fly mushroom body. Our tool is flexible: by taking gradients through the training procedure we can meta-learn inductive biases for networks trained using PyTorch. Code to produce our figures can be found at https://github.com/WilburDoz/Meta_Learning_Inductive_Bias, including a basic ReLU network (Appendix A of William Dorrell, Yuffa, and P. Latham 2023) which should be easily adapted to networks of interest.

3.1.5.1 Spiking Neural Network

The brain, unlike artificial neural networks, computes using spikes. 'How?' is an open question. A recent exciting advance in this area is the surrogate gradient method, which permits gradient based training of spiking neural networks by smoothing the discontinuous gradient (Neftci, Mostafa, and Friedemann Zenke 2019; Friedemann Zenke and Vogels 2021. We make use of this development to meta-learn the inductive bias of a spiking network, providing a challenging test case for our method.

We study a modification of a model developed for a recent tutorial (Goodman et al. 2022; Freidemann Zenke 2019, which is trained to assign a label to an incoming spike train. The network is a model of an interaural phase difference

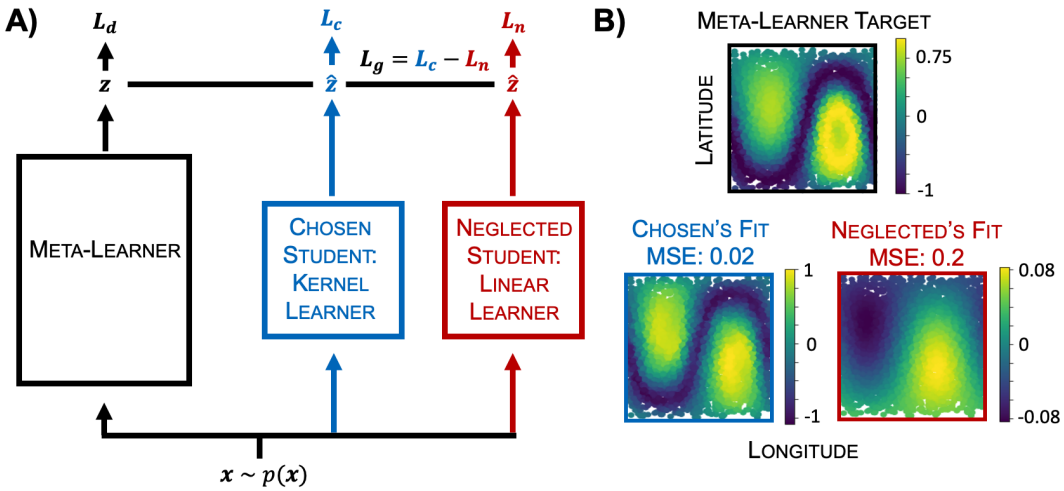

Figure 5: **Meta-learning design choice impact. A:** Labellings are learnt such that a chosen student generalises much better than a neglected one. **B:** The meta-learner finds a non-linear labelling for which kernel regression generalises an order of magnitude better than linear regression.

detection circuit. The input spike train is parameterised by a phase difference, $\Delta\phi$, that generates two sets of spike trains, one in each ear, Fig. 6A. These spikes are processed through a hidden layer of linear-integrate-and-fire neurons (LIF), before reaching a classification layer. A real-valued valence is assigned by summing the output neuron's activity over the trial. The meta-learning framework is as before: the meta-learner assigns valences to input phase differences, these labels are used to train the spiking network by surrogate gradient descent, then the meta-learner is trained to minimise the learner's generalisation error and a distribution loss. Our method works well, finding a simple smoothness prior, Fig. 6B.

3.1.5.2 A High-Dimensional MNIST Example

Next, we test out method on a high-dimensional input dataset. Thus far, to visualise our results, we have only considered low dimensional input data. We demonstrate that our method continues to work in high-dimensions by applying it to a dataset made of the 0 and 1 MNIST digits (LeCun 1998. We meta-learn a labelling of this dataset that a simple convolutional neural network finds easy to generalise. Our meta-learner's architecture is also a convolutional neural network whose outputs are bounded between 0 and 1, and the meta-learner must learn an easy-to-generalise labelling with high variance. We find that the meta-learner consistently rediscovers the MNIST digits within the dataset, separating each digit into its own class, figure 7. We return to the important question of understanding high-dimensional inductive biases in the discussion.

3.1.5.3 Interpreting Connectivity Patterns through their Induced Inductive Bias

A large maturing source of neuroscience data is connectomics (a list of which neurons connect to one another). However, there is currently a dearth of methods for interpreting this data (Litwin-Kumar and Turaga 2019). In this section, we show our tool can be used to give normative roles to connectomic patterns through their induced inductive bias. We study a model of the fly mushroom body, a beautiful circuit that fruit flies use to assign valence to odours (Aso et al. 2014; Hige 2018), for which connectomic data has recently become available (Z. Zheng, Lauritzen, et al. 2018; Z. Zheng, F. Li, et al. 2022).

Odorants trigger a subset of the fly's olfactory receptors. These activations are represented in a small glomerular population (input neurons), projected to a large layer of Kenyon cells (hidden neurons), then onwards to output neurons that signal various dimensions of the odour's valence, Fig. 8A. An error signal is provided if the fly misclassifies a good odour as bad, or vice versa, allowing the fly to update its weights and learn appropriate responses. Classically, the input-to-hidden connectivity was assumed random; i.e., each hidden neuron connects to a few randomly selected input neurons. However, connectomic data has shown that hidden neurons preferentially connect to some inputs, and there are input groupings - if a hidden neuron connects to one member of a group it likely connects to many, Fig 8D

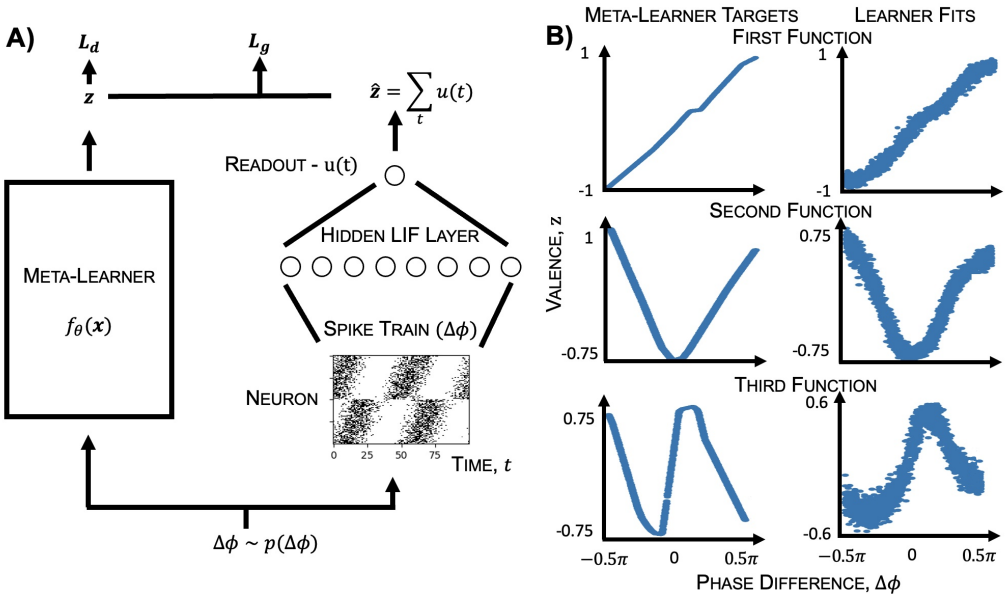

Figure 6: **Meta-learning through a Spiking Network. A:** Labellings are learnt that the spiking network, with weights trained via surrogate gradient descent, finds easy to generalise. Phase differences, $\Delta\phi$, are sampled uniformly and used to generate spike train by sampling from a poisson process with the following rates: for half the neurons $r_n = \frac{r_{\max}}{2}(1 + \sin(t + \theta_n))^2$, where n is a neuron index and θ_n are uniformly sampled offsets; for the other half we add a phase shift: $r_n = \frac{r_{\max}}{2}(1 + \sin(t + \theta_n + \Delta\phi))^2$. These populations represent sensory neurons in the two ears, and $\Delta\phi$ is the interaural phase difference. This activity feeds into a population of linear-integrate-and-fire neurons, then onwards to a readout linear-integrate neuron. The valence assigned is the sum of the readout's activity over time. **B:** We learn three orthogonal meta-learners (as in section 3.1.3) and find the spiking network finds it easiest to learn low frequency functions. Left: the meta-learner's target function. Right: the spiking network's labelling. As can be seen, the spiking network captures the main behaviour, but increasingly poorly at higher frequencies.

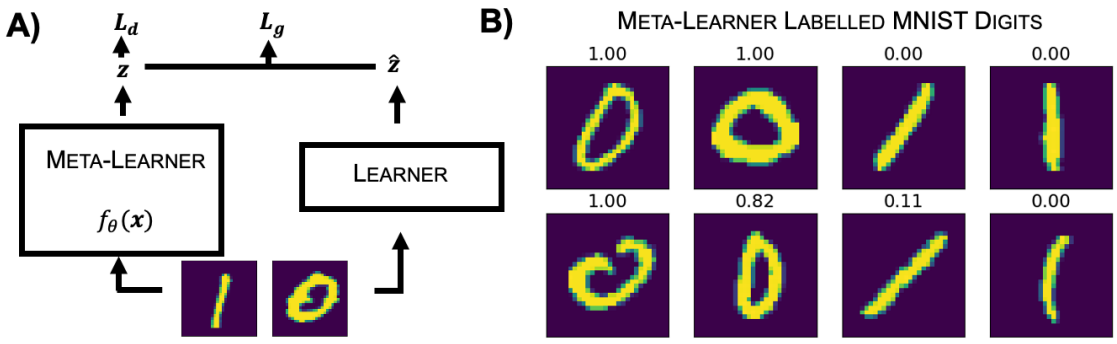

Figure 7: **Meta-Learning on MNIST A:** A meta-learner receives MNIST 0s and 1s, and assigns labels, bounded between 0 and 1, that have high variance and can be easily generalised by the learner. **B:** 99% of digits are assigned a label, shown in title, consistent with MNIST class.

(Z. Zheng, Lauritzen, et al. 2018; Z. Zheng, F. Li, et al. 2022). (Zavitz et al. 2021) tested networks with this connectivity on a battery of tasks and found that, compared to random, (1) they were better at identifying odours that activated over-connected inputs, and (2) they generalised assigned valence across a group (i.e. if you assign high valence to the activation of one neuron, you do the same for other neurons in the same group).

We used our tool to verify and develop these findings by examining the effect of different connectivity patterns on the inductive bias of a sparsely-connected model of the mushroom body, Fig. 8A. As a baseline, fully connected networks are biased towards smooth functions, appendix A of William Dorrell, Yuffa, and P. Latham 2023, the simplest being those that assign valence based on one direction in the input space: high at one end, low at the other, like in Fig. 3B - C.

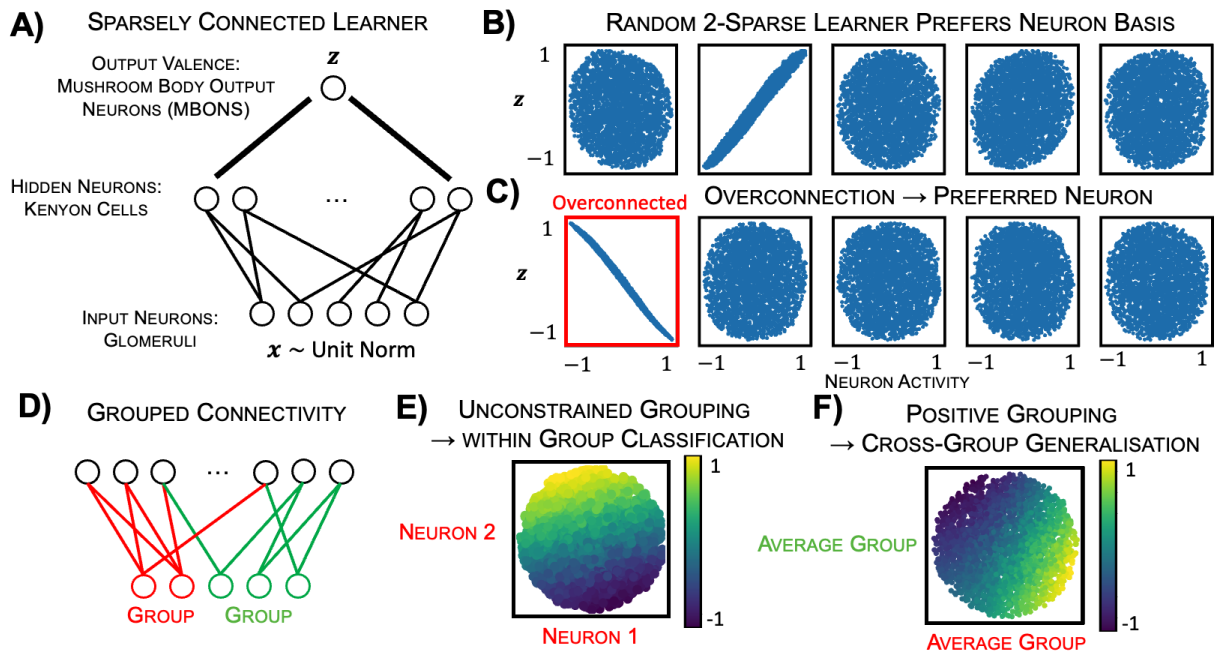

Figure 8: **Understanding Connectivity via Inductive Bias. A:** We model the fly mushroom body as a ReLU network with one large hidden layer. Each hidden neuron is connected to two of the five input neurons. **B:** The meta-learner finds the labelling the learner generalises most easily. We show this labelling projected against each of the input neuron activities. As can be seen, the labelling depends on only one neuron's activity, second from left. **C:** In the overconnected setting each hidden neuron still connects to two inputs, but there is a strong bias towards connecting to the first, highlighted neuron. As a result, the meta-learner settles on a labelling that depends only on this neuron's activity. **D:** We explore the impacts of group connectivity, in which the input neurons are divided into two groups, and hidden neurons tend to be connected to two neurons from the same group. **E:** We train the meta-learner, and find that it's labelling depends only on neurons within the same group. The plot shows the projection of the datapoints into a subspace defined by the two neurons in the red group. The labelling depends linearly on position within this subspace. **F:** However, if the input-hidden connections are constrained to be positive, the meta-learner's labelling depends only on the average activity within each group, i.e. if one member of a group increases the output, so do all members; hence, the function generalises across group members.

However, which direction is unimportant; they're all equally easy to learn. Sparsity breaks this degeneracy, aligning the easiest to learn functions with the input neuron basis, figure 8B. As such, sparse connectivity, which is ubiquitous in neuroscience, ensures the fly is best at assigning labels based on the activity of small collections of neurons. Next, we introduced the observed connectomic structure. Biasing the connectivity, so some inputs have more connections than others, broke the degeneracy amongst neuron axes. The networks were, as expected, best at generalising functions that depended on the activity of overconnected inputs, figure 8C, matching (Zavitz et al. 2021). Finally we introduce connectivity groups, figure 8D. Without additional changes this does little, the neuron basis is still preferred and, unlike (Zavitz et al. 2021), generalisation across inputs is not observed, figure 8E. Only when we additionally constrain the input-to-hidden connections to be excitatory (i.e. positive) do we see that the circuit becomes inductively biased towards functions that generalise across groups of inputs, figure 8F. In retrospect this can be understood intuitively: positive weights and grouped connectivity ensure that a hidden neuron that is activated by one input will also be activated by other group members, encouraging generalisation. This effect is removed by permitting negative weights, which let members of the same group excite or inhibit the same hidden neuron.

Thus, we verify and extend the findings of (Zavitz et al. 2021) without needing to presuppose a battery of tasks. This avoids a potential flaw in the approach of (Zavitz et al. 2021): you may reach an incorrect conclusion simply because your battery of tasks in not comprehensive enough! (though in this case we agree wholeheartedly with the conclusions of (Zavitz et al. 2021) Our method avoids this problem by meta-learning the appropriate tasks. In doing so we highlight how our tool can be used to gain insight into the role of circuit design choices, in particular, the importance of the neuron basis for sparsely connected networks.

3.1.6 Discussion & Conclusions

We presented a meta-learning approach to extract the inductive bias of differentiable supervised learning algorithms, which we hope will be useful in normatively interpreting the role of features of biological networks. This approach required few assumptions beyond those that make the inductive bias an interesting way to conceptualise a circuit in the first place. We required, first, that the input data distribution was specified. Second, the circuit must be interpretable as performing supervised learning. And, third, you must specify the way the circuit learns, and be able to take gradients through this learning process. We will discuss each of these requirements and ways they could be relaxed; regardless, it is heartening that any circuit satisfying these will, in principle, suffice. The analytic bridge between kernel regression and its inductive bias (Bordelon, Canatar, and Pehlevan 2020; Simon, Dickens, and DeWeese 2021) has already found multiple uses in biology in just a few years (Bordelon and Pehlevan 2021; K. D. Harris 2019; Pandey et al. 2021; M. Xie et al. 2022), despite its stringent assumptions. We hope that relaxing those assumptions will offer a route to allow these ideas to be applied more broadly.

The first requirement is access to an input distribution, which is often lacking. This can be avoided by using real neural data as the input. Or, if neural data is limited, generative modelling could be used to fit the neural data distribution and new samples drawn from that distribution. Finally, one could imagine a single meta-learner that creates not only the label, but also the data. That is, the meta-learner could generate the entire dataset by transforming a noise sample into an input-output pair. This would have to be carefully regularised to avoid trivial input distributions, but could in principle learn the input statistics that particular networks are tuned to process.

Next, we could relax our second assumption, that the learner performs supervised learning. This is a standard assumption in theoretical neuroscience (Hiratani and P. E. Latham 2022; Schaffer et al. 2018; Sorscher, Ganguli, and Sompolinsky 2022), and is often reasonable. Some circuits contain explicit supervision or error signals, like the fly mushroom body or the cerebellum (Shadmehr 2020), and generally brain areas that make predictions (i.e., all internal models), can use their prediction errors as a learning signal. Alternatively, some circuits are well modelled as one area providing a supervisory signal for another, as in classic systems consolidation (McClelland, McNaughton, and O'Reilly 1995), or receiving supervision from a past version of themselves through replay (Ven, Siegelmann, and Tolias 2020). Nevertheless, much biological learning seems to be unsupervised. Our framework could be extended to these settings by assuming an unsupervised objective and meta-learning the dataset on which an observed circuit performs well. For example, the unsupervised objective might be to produce a lower-dimensional representation of the data with the same dot-product similarity structure as the inputs. If the learner was doing PCA, then the meta-learner would learn data that lay in a linear subspace. It would be interesting to try this for different dimensionality reduction algorithms, or to see which bits of input structure biological unsupervised networks are tuned to process.

Finally, in a slightly kooky way, we could avoid even specifying the learning algorithm by interfacing with an animal directly! Animals' inductive biases are objects of interest in their own right, but can also give insight into the underlying neural processing. These biases could be studied by replacing the inner learner with a real animal that is trained on a labelled dataset from the meta-learner, then tested on new datapoints. Since we cannot compute gradients through the computations of a living animal, the meta-learner could be optimised using black-box optimisation procedures that rely only on meta-loss evaluations, like the Nelder-Mead method (S. Singer and Nelder 2009).

Despite our optimism for this approach, there remain challenges. Most fundamentally, sets of functions that a learner easily generalises are still hard to interpret. We have shown how our tool can provide insight for low-dimensional inputs (Fig. 2 - 5), by comparing to ground truth labels (Fig. 6), or by projecting the learnt functions onto an appropriate basis (Fig. 7). However, to make the concept of inductive bias more powerful, more tools are needed to interpret the resulting functions.

To conclude, the inductive bias is a promising angle from which to understand learning algorithms. Analytic bridges between circuit design and inductive bias have already 'explained' the presence of aspects of the circuit through their effect on the network's generalisation properties in both artificial (Bahri et al. 2021; Canatar, Bordelon, and Pehlevan 2021) and biological (Bordelon and Pehlevan 2021; K. D. Harris 2019; Pandey et al. 2021; M. Xie et al. 2022) networks. However, these techniques require very constraining assumptions. We have dramatically loosened these assumptions and shown our tools utility in, among other things, interpreting connectomic data. We believe it will prove useful on other datasets and problems.

3.2 Connectivity in Cerebellar-like Networks as an Inductive Bias

TL;DR: So you're telling me that if my neural network looks like it is performing kernel regression (nonlinear processing followed by linear regression) you can extract its inductive bias? Well, it turns out some of my favourite circuits look a bit like that! Can we therefore understand some of the recent findings in these circuits through their effect on the inductive bias?

Yes! We will look at connectomic data, but I suspect this could also be useful applied to temporal processing, or perhaps even context dependent shifting.

However, while this work was satisfying, I don't think it is worth publishing, because the link between cerebellar circuits and kernel regression is not novel (M. Xie et al. 2022), nor is the interpretation of the connectomic data as changing which functions are easier to learn (Zavitz et al. 2021; Z. Zheng, F. Li, et al. 2022).

ABSTRACT

Cerebellum-like networks, in which the activity of input neurons is projected to a much higher-dimensional space before classification, are a recurring neurobiological motif, present in the cerebellum, dentate gyrus, olfactory system, and electrosensory system of the electric fish. Their relatively simple design presents a promising test-case for understanding principles of biological learning, and they have long been posited to perform pattern separation. Previously, the projection to high dimensions in these networks has been modelled as random; however, electron-microscopy studies have discovered interesting hints of structure in one of these networks, the fly mushroom body. To explain this non-random connectivity, recent theoretical work tested models of this circuit with and without the observed connectivity on a set of tasks and found that the circuit with structure had improved performance relative to the random baseline on some, presumably naturally pertinent, tasks. Here, we present an alternative method to derive the same conclusions by building a simplified kernel regression model of the system and using results from Machine Learning theory to examine the inductive bias of the circuit. While requiring more modelling assumptions, our approach removes the need to guess tasks at which the network might perform well, instead telling you which, among all tasks, the network is better at learning. Hence, we find that the structure in the projection weights shapes the network's inductive bias, making some functions easier to learn and others harder. We hope this work provides a complementary approach to understand the functional implications of future connectomic findings about cerebellum-like networks.

3.2.1 Introduction

The 'crystalline' nature of cerebellar cortex has attracted theoretical work for over 50 years, since it was posited to perform pattern separation (Kawato, Ohmae, et al. 2021). The basic network structure comprises an expansion then contraction in the number of neurons: a small number of input mossy fibres connect to a vastly larger population of granule cells, which themselves converge onto a smaller number of purkinjie cells. This circuit motif has since been found repeatedly, in the dentate gyrus, the mushroom body of the fly, and the electrosensory organ of the electric fish. Classic theory has explained the expansion and contraction in terms of pattern separation: the purkinjie cell input weights undergo synaptic plasticity in order to recognise previously experienced patterns, and the structure of the granule cell layer's sparse representation allows maximal distinction between otherwise similar patterns. To first order, this pattern separation machine remains a good model of the operation of all these cerebellar-type networks.

However, as experimental techniques have improved we have gained access to more detailed information about these circuits, leading to updates of our theory. For example, in the cerebellum findings that the granule cell layer is often not so sparse have updated our view of the kinds of things the cerebellar network is designed to learn (M. Xie et al. 2022), from pattern clustering to smooth input-output mapping. One major shift in recent neuroscience is the availability of connectomic data, telling us which neurons are connected. Recently connectomic data for both the fly mushroom body (Z. Zheng, F. Li, et al. 2022) and a section of cerebellum (Nguyen et al. 2023) have become available, and this has led to another update of the underlying pattern separation theory. Previous models and theories have generally assumed that the connectivity between input and expansion layer is random. The connectomic data, however, suggests that there are hints of structure. Hidden layer neurons do not connect randomly to inputs; instead they connect preferentially to some inputs, and there are correlations in the connectivity, i.e. if a hidden neuron is connected to input A it is very likely to be connected to input B. Further, in the mushroom body at least, the connectivity structure appears independently of neural activity (Hayashi et al. 2022), suggesting it is genetically hard-wired. We are then prompted to ask, why is the nervous system investing effort to establish this connectivity? What functional role does it play in the pattern separating circuit?

Recent modelling work has tackled exactly this question. Zavitz et al. 2021 built models of the fly mushroom body with and without the observed structured connectivity. They then trained and tested each of the networks on a battery of tasks, and found that the networks with the observed connectivity performed better at some, presumably naturally pertinent, tasks, and worse at others. In particular, over connecting to some input glomeruli allows easy identification of monodours that only activate that glomerulus. Conversely, correlations in connectivity made the network worse at distinguishing odours that activated subsets of correlatedely connected groups of glomruli, suggesting the fly is trying to generalise across the group. Odours the fly cares about presumably have a tendency to activate single over-connected glomeruli or whole groups of correlatedly connected glomeruli, and this connectivity pattern permits their easy identification.

In this work we present an alternative route to reach broadly the same conclusions about the functional effect of structured connectivity. Our approach will make use of the correspondence between the fly mushroom body circuit (as in M. Xie et al. 2022), which seeks to assign valences to odours, and an algorithm called kernel regression, which assigns labels to inputs. In kernel regression inputs are passed through a fixed non-linear mapping to a feature space, from which a linear readout to labels is learnt using training examples. Similarly, the mushroom body seems to pass incoming odour signals through a fixed transformation to the kenyon cell population, from which readout neurons are trained using a supervised error signal. Thus, kernel regression is a reasonable model for the fly mushroom body and other cerebellum-like circuits. Recent advances in machine learning theory have characterised the generalisation properties of kernel regression; i.e. how well the algorithm is able to capture input-label mappings from a few training examples. We will apply these theoretical advances to a cerebellum-like model and show the effect of different connectivity patterns on the generalisation properties of the network. Hence, we assign a normative role to the structured connectivity: allowing the fly to learn some, presumably important, functions (assignments of valence to input neuron activities) faster than if the connectivity had been random, and other functions, presumably naturally unimportant ones, slower. Our work is part of a trend that has used kernel regression to understand various biological systems (Bordelon and Pehlevan 2021; K. D. Harris 2019; Pandey et al. 2021; M. Xie et al. 2022), that we hope will continue.

Relative to the work of Zavitz et al. 2021 our model is very toy for developing insight. However, we think it is valuable for a few reasons. First, it links the findings of Zavitz et al. 2021 to kernel generalisation properties, verifying them and providing a basis for understanding them. Second, it covers a potential flaw of the approach of Zavitz et al. 2021 (a comment on approach, not content!). Building models and testing how well they learn certain functions requires guessing a set of functions in the first place. This requires insight into the system and behaviour of the animal which is not always forthcoming. Our approach, on the other hand, tells you how the network learns all possible functions, ensuring that you are not missing a key effect because you didn't test the network on that function. Finally, we hope that as more experimental findings about these circuits come to light, connectomic or otherwise, future work will be able to use whichever approach is most suited to answering the question at hand, and that in many cases this will be kernel regression.

3.2.2 Cerebellar-like Networks as Kernel Regression

We begin by outlining our simplified model cerebellar-like networks, its relation to kernel regression, and the way in which machine learning theory can tell us its inductive bias or generalisation properties. We focus on describing fly olfactory system, though are arguments can be equally applied to other cerebellar-like networks.

3.2.2.1 The Fly Olfactory System, and Our Model

Odours arrive at the periphery of the fly olfactory system and activate a panel of olfactory receptor neurons (ORNs) through interactions with their receptor proteins. There are many different types of receptor proteins, but each ORN has only one (figure 9A, the different coloured receptor neurons). Each type of ORN converges onto a single glomerulus, of which there are about 50 in total. In the glomerulus the olfactory receptor neurons synapse with projection neurons (PNs) that carry the information onward towards both the lateral horn, another site of more hard-coded olfactory processing, and the mushroom body. In the mushroom body they meet the 2000 kenyon cells (KCs), which forms a large sparse representation of the odours that will serve as the basis for classification. Each KC receives input from roughly 7 projection neurons and a large inhibitory neuron synpases onto all the KCs and ensures that only 5-10% of neurons are active at any one time. Finally, mushroom body output neurons (MBONs) connect to the entire KC population and are thought to represent an odour classification of sorts, signalling various dimensions of the valence of a given odour. Most of these synapses are not thought to change on short learning timescales, except the important site of learning, the KC-MBON connections. These learning dynamics are governed by dopaminergic neurons, that are thought to send an error signal that drives learning in the KC-MBON connections. Hence, this system forms a neat odour classification device, schematised in figure 9A.

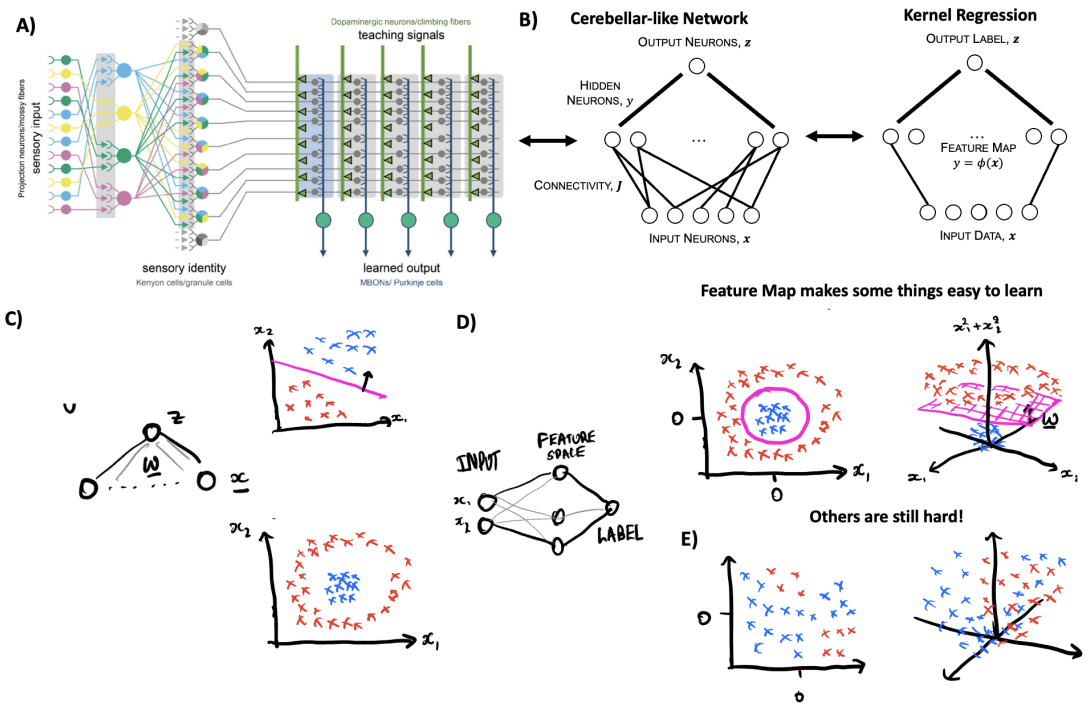

Figure 9: **A** Schematic of the fly mushroom body circuit. **B** Simplified model of this and other cerebellar-type networks; looks very similar to kernel regression! **C** Why not a linear readout? Very few classifications you can do! So **D** Why the non-linearity: the feature map makes many functions linearly learnable that weren't previously. But still leaves open the question of which feature map; each one makes some functions easier to learn than others.

In this work we are interested in the effect of the PN-KC connectivity, therefore we build a model of the mushroom body that abstracts away as many of the other details, figure 9B. We begin with PN activities, $\boldsymbol{x}$, that are sampled uniformly from the surface of the sphere. For all plots and numerical calculations we will use three PNs (any more and plotting becomes hard, and the numerics scale badly with dimension), but fortunately the analytics generalise trivially to higher dimensions. Our input assumption also means we are ignoring the interesting processing of the olfactory inputs before reaching the PNs (such as Wanner and Friedrich 2020), as it would obscure the main effects of the connectivity. Similarly, though the PN activity is not really uniformly distributed on the surface of a sphere, it makes calculation of the kernel analytically tractable, and allows us to get more insight into the effect of connectivity. We then connect the PNs to the much larger KC population through a connectivity matrix, $\boldsymbol{J}$, each element of which encodes the connection strength of one PN-KC connection. A non-linearity, ϕ, is applied to produce the final KC activity - $\boldsymbol{y} = \phi(\boldsymbol{J} \cdot \boldsymbol{x})$. We will get some analytic insight using a simple ReLU nonlinearity, but will show the same conclusions hold using a more biologically plausible sparsification so that only 5% of neurons are active at one time, as observed (Lin et al. 2014) [data not included here]. Finally, a readout weight, $\boldsymbol{w}$, assigns valences: $\hat{z} = \boldsymbol{w} \cdot \boldsymbol{y}$ and is learnt using a training set of 'true' input-label pairs: $\mathcal{D} = \{\boldsymbol{x}_i, z_i\}$, gathered during the recent experience of the fly. Hence, this network captures many of the features we are interested in: it has the expand-compress pattern of cerebellar-type networks, it matches the kernel regression template of processing inputs towards some feature space from which linear classification is performed, and it allows us to interrogate the role of the connectivity matrix, $\boldsymbol{J}$.

3.2.2.2 What role does the Connectivity play?

How can we understand the effect of different connectivity matrices on this network? To begin, we consider why include an extra layer of processing at all? Why not readout odour valence from the PN population directly? (Figure 1C) After all, simple biologically plausible learning rules can find you the optimal $\boldsymbol{w}$ relatively easily.

The answer comes due to the limitations on what this circuit is able to learn. A linear readout is very constrained, only able to assign valence in a way that depends on the position of the datapoint along one direction of activity space, the $\boldsymbol{w}$

direction, figure 9C. Many datasets of interest have valences that vary non-linearly, and hence cannot be classfied by our simplified circuit, figure 9C.

One way to solve this problem is to project the data into a feature space, then to perform the same simple linear classification, but from the feature space. For example, the non-linear classification problem in figure 9C can be solved by including extra axes in a higher dimensional feature space, from which linear data labels are assigned, figure 9D. The mapping from PNs to KCs can be thought of in the same way, as a feature map, permitting much more complex functions (assignments of valence to input) to be learnt by a simple linear readout which can be easily and biologically plasuibly trained.

However, we now face a different choice: which feature map to use? Different feature maps will make different non-linear classifications easier or harder to learn, figure 9D. Making your feature space very high-dimensional, by having many KCs, ensures that you have access to many different features of the data, hedging your bets about which functions you will need to learn in the future. However, while this may make it possible to learn many functions, not all of them can be learnt so easily.

The kernel regression algorithm, like all classification algorithms, carries with it an inductive bias; a preference for learning certain functions over others. During learning, the network has been given a few labelled examples that it fits, learning to assign the same valence to the same inputs. However, with finite examples there are infinitely many ways to generalise from the examples to unseen data, figure 1E, and choosing amongst them without prior assumptions is impossible (D. Hume 1748). Somehow, the kernel regression algorithm chooses one of these solutions. How it does so is called its inductive bias, and determines how good the regression algorithm is at learning to fit target functions given few training datapoints. If you need prohibitively large numbers of training points to persuade the network to fit the functions you care about, because it chooses other functions that fit the data equally well but aren't what is needed, then its of no use.

Recent machine learning theory work has characterised the inductive bias of kernel regression algorithms (Bordelon, Canatar, and Pehlevan 2020; Sollich 1998), and it is this characterisation that we shall use to understand the normative role of the observed structured connectivity. This is done using the kernel function - $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{y}(\boldsymbol{x}) \cdot \boldsymbol{y}(\boldsymbol{x}')$ - which measures the similarity of two datapoints in the feature space. This is the key quantity: if two inputs are represented similarly in the feature space then kernel regression finds it easy to assign them similar valences, but difficult to assign them different valences. It turns out that the key constructs are the kernels eigenfunctions, $v_i(\boldsymbol{x})$, and their eigenvalues, λ_i, defined analogously to eigenvectors:

$$\int k(\boldsymbol{x}, \boldsymbol{x}') v_i(\boldsymbol{x}') dp(\boldsymbol{x}') = \lambda_i v_i(\boldsymbol{x}) \tag{3}$$

The cited theory tells us that the higher the eigenvalue of a given eigenfunction, the fewer training datapoints are needed to learn the function to a given level of accuracy. Intuitively, eigenfunctions that assign the same label to points represented similarly in feature space will have high eigenvalues. Hence, the network is inductively biased towards eigenfunctions with high eigenvalue. Further, a general function, some assignment of valence to a set of neural activities, can, thanks to the linearity of the problem, be projected onto the eigenbasis and each component will be learnt separately, with a speed (in terms of training points) set by the corresponding eigenvalue.

This gives us a way to answer our initial question. The connectivity matrix somehow changes the structure of the representation in the kenyon cell layer. This can be summarised through changes to the kernel function, which measures representational similarity in the kenyon cell layer. These changes to the kernel cause corresponding changes to the eigenstructure, changing the inductive bias of the network as a whole. We can therefore understand the functional role of different connectivity structures by looking at their effect on the network's inductive bias.

3.2.3 Structured Connectivity and its Effects on Generalisation

In this section we study how a variety of connectivity schemes, modelling the observed connectivity structure in the fly, influence the inductive bias of the network. Generally calculating the kernel and its eigenstructure analytically is challenging, so in pursuit of some minimal analytics from which we can draw intuitive conclusions we choose a simple connectivity model. Fortunately, however, numerically evaluating the kernel and its eigenfunctions is always a possibility once you have specified the network, so we are able to verify our claims for more realistic connectivity schemes relatively easily [data not shown here]. Call the set of input weights to the ith KC $\boldsymbol{J}_i$, figure 10A. Our model specifies the connectivity matrix through specifying a probability distribution on $\boldsymbol{J}_i$, from which each KC samples its weights. We assume there are infinitely many KCs (the infinitely wide nose assumption), which is not a bad assumption for a network in which there are many more KCs than PNs or MBONs, and calculate the resulting kernel. Our distribution on $\boldsymbol{J}_i$ is a multivariate gaussian:

$$\boldsymbol{J}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}) \tag{4}$$

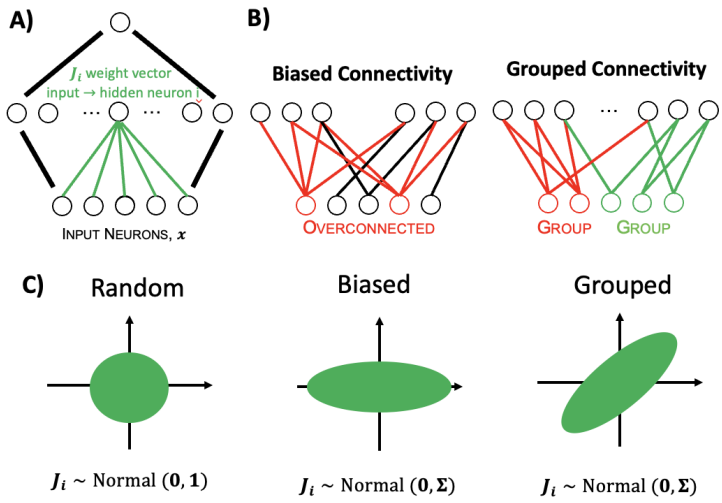

Figure 10: **A** To recapitulate the observed connectivity patterns in a simple way we study. **B** The observed connectomic deviations from random: biases - some input neurons tend to be more connected than others; grouping - input neurons come in groups such that if you are connected to one you tend to be connected to all. **C** We recapitulate some of these with a simple gaussian model.

This minimal model allows us to study the effects of the observed connectivity structure through the covariance matrix, $\mathbf{\Sigma}$. Random connectivity, $\boldsymbol{J}_R$, corresponds to $\mathbf{\Sigma} = \mathbf{1}$. It is observed that KCs connect to some PNs with higher probability than others, we model that by stretching the covariance along one PN axis relative to the others, figure 10. Similarly, there are correlations in the connectivity, there are groups of PNs such that if a KC is connected to one of them it is likely to be connected to all of the others. We model this by introducing correlations into $\mathbf{\Sigma}$, such that if the KC is strongly connected to one PN in the pair it is likely to also be to the other, figure 10. Given all these assumptions a simple extension of previous work (Cho and Saul 2009) allows us to derive the kernel for each connectivity scheme (Pandey et al. 2021). We will use this to understand each connectivity's effect on the inductive bias.

3.2.3.1 Inductive Bias of Randomly Connected Network

For a random connectivity matrix, $\boldsymbol{J}_R$, the kernel depends only on the angle between inputs, θ, in the input space:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \frac{|\boldsymbol{x}||\boldsymbol{x}'|}{\pi}[\sin(\theta) + (\pi - \theta)\cos(\theta)]. \tag{5}$$

The eigenfunctions are spherical harmonics, the lower the frequency the higher the eigenvalue. Therefore with $\boldsymbol{J}_R$ the network is inductively biased towards smoother functions: it requires fewer training points to learn a function that assigns similar labels to similar input, figure 11.

3.2.3.2 Effect of Biased Connectivity

We now consider including the bias, i.e. connecting to some PNs more strongly than others, figure 10. The kernel formula, eq. 5, doesn't change, we simply have to change the definition of θ and $\boldsymbol{x}$. Rather than the datapoints and the angle between them, first you stretch the sphere by the square root of the covariance matrix, then measure the angle between the datapoints, figure 12. This means that variations in the activity of the PN that is more strongly connected naturally have a larger effect on the structure of KC representation. This intuitively effects the eigenstructure; eigenfunctions with variation along the overconnected PN direction are easier to learn, and hence have higher eigenvalue, figure 12.

This leads us to our first intuitive conclusion. Networks that overconnect to some inputs find it easier to learn functions that depend upon variation of that input relative to the others. Therefore monodours that perhaps only activate one PN can still be well identified if they stimulate an overconnected PN.

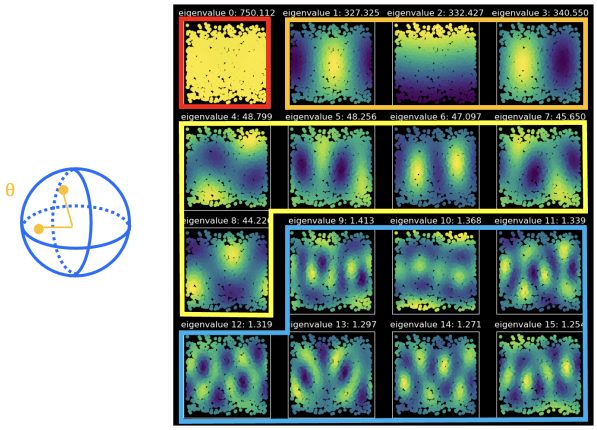

Figure 11: The important quantity for determining the inductive bias is the kernel, and this depends on the angle between input datapoints. The easiest functions to learn correspond to low frequency spherical harmonics, the higher the frequency the harder to learn. This is shown in the figure by the eigenfunctions, which are spherical harmonics, the higher the frequency the lower the eigenvalue. They are grouped by frequency, because there are 3 first order harmonics, 5 second, etc.

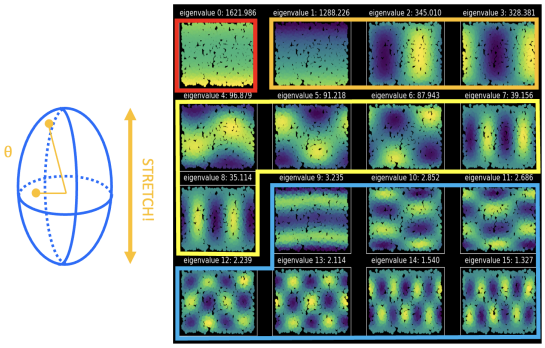

Figure 12: The similarity structure now changes to emphasise differences in how the overconnected neurons fire more. This causes perturbations to the underlying spherical haromonic structure, while roughly similar, now the linear combinations of spherical haromnics of a certain order that vary most along the overconnected neuron direction (the z axis) are easiest the learn; and those that vary along the x and y directions are hardest.

3.2.3.3 Effect of Correlated Connectivity

Correlations in the connectivity have a similar effect. We again keep the same kernel equation, again only the definitions of $\boldsymbol{x}$ and θ change, and again it corresponds to the angle between input datapoints after streching by the connectivity covariance, $\boldsymbol{\Sigma}$, figure 13. In figure 13 the x and y PN form a correlatedly connected group. Variations in the activity of all members of this group together, variation along the $x + y$ direction, cause large changes to the KC representation. Conversely, variations in which some members of the group are more active, others less, i.e. variation along the $x - y$ direction, cause small changes in the KC representation. The effect on the eigenstructure is again intuitive, figure 13. The network is inductively biased towards labellings that assign similar valences to datapoints that activate the correlatedly connected group of PNs to a similar degree, figure 13. Conversely, labellings that require differentiating between different inputs in which some members of the PN group are active and others inactive are much harder for the network to generalise.

Hence, we reach our second intuitive conclusion. Correlations in the connectivity encourage generalisation across the activity of members of the correlatedly connected group, making it easy to distinguish odours that tend to activate the whole group, and hard to make distinctions between different odours that activate some members of the group but not others.

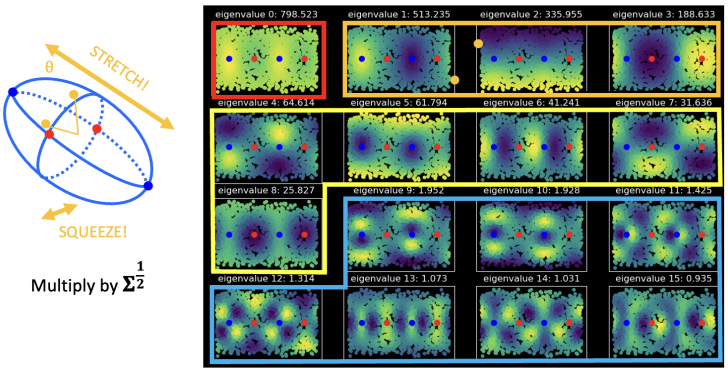

Figure 13: We perform the same mapping for the correlated connectivity, except now different patterns are easy and hard to learn. It is easy to spot variations that cause all members of a group to change in the same way, and harder to spot those that cause some members of the group to activate, others to inactivate. The blue and red dots on the ellipse are mapped onto the plots, and are supposed to guide your interpretation. For example in the first order spherical harmonics the function that is easiest to learn assigns high value to one blue point and low to the other. Conversely the hardest to learn function assigns it based on the red points.

3.2.3.4 Effect of Sparse Binary Connectivity

One final effect that is fun to explore is how sparsity in the connectivity matrix effects the inductive bias. We are not able to study this analytically, but we can run the numerics and, relatively intuitively, a sparse connectivity matrix leads the easy to learn function to align with the neuron axes, figure 14.

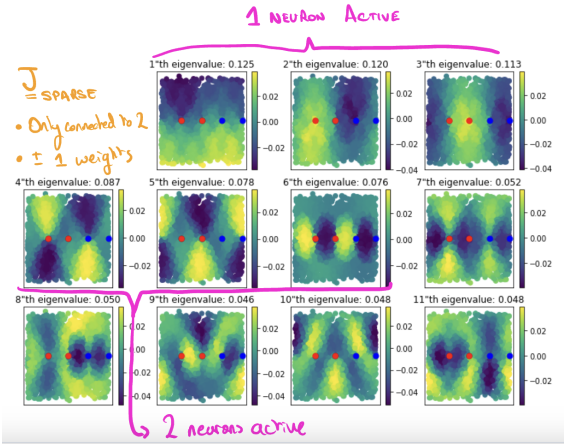

Figure 14: We change the connectivity matrix so that every hidden neuron connects to only two of the input neurons. What we find is that the easiest functions to learn correspond to 1 neuron being active, the next easiest to 2 neurons being active, then other functions. Nice!

3.2.4 Conclusion

In conclusion, we have provided a normative role for a peculiar connectomic structure: it changes the feature map making some functions easier to learn than others. Further, we have introduced some tools to study this circuit in other settings, that we hope will be useful broadly.

3.3 Future Directions in Inductive Bias World

I don't know if I want to take this work further. Both projects were fun, but the effort invested in sharing the biological work doesn't seem worth it, for both me and any readers. Some (semi-)interesting directions that could make it worthwhile:

- What is the role of temporal processing. Some very nice work found decorrelating dynamics in fish olfactory bulbs (Wanner and Friedrich 2020). There is work going on in the schaeffer lab at the Crick on the dynamics of odour processing in the mammalian olfactory bulb that could be interesting to analyse through this lens (Ackels et al. 2021). What is the functional role of the dynamic processing?

  This would be an alternative interpretation to things like sparse signal recovery (such as Hiratani and P. E. Latham 2020; Tootoonian and Lengyel 2014)
- Another related finding is the context dependent processing in the olfactory bulb, different task contexts lead to different bulb representations - and it can switch back and forth flexibly (Kay and Laurent 1999; Koldaeva, Schaefer, and Fukunaga 2019; Kudryavitskaya et al. 2021).

  Recently there's been evidence for context dependent processing in flies (based on Okray et al. 2022). This is great, because in flies you can really start to do mechanism, and use that as a constraint on the set of plausible ways context can affect your processing. Then you could imagine using those constraints to think about optimal ways to adapt your kernel to task demands, things like that...

  As a quick aside on the value of fly models, you can do ABSOLUTELY WILD experiments, like theorists wet dreams, where you LITERALLY CHANGE THE NUMBER OF KENYON CELLS, OR THE SPARSITY OF CONNECTIVITY, with GENETICS!!!! Ahmed et al. 2023 do this, it's wild, their conclusions are fun, but it would be great to test more precise theoretical questions with these methods.
- We could run these things on real data, to avoid the bad sphere approximation, though that seems tough...
- You could imagine setting up a constrained set of possible connectivities, where the constrained set somehow captures biological plausability, preferably by incorporating possible developmental programs. Then you could see how optimising this developmental set for different environmental challenges led to different connectivity matrices. This is inspired by a recent paper that studies the connectivity structures of different related species of flies (K. E. Ellis et al. 2023).

  This can go one of two ways, first it links structure to function. But you can also go the other way, use it as a way to detemine what a plausible set of developmentally-allowable connectivity matrices looks like, and therefore get access to the plausible space evolution is playing with.
- I would love to meta-learn the inductive bias of real animals. This requires gradient-free optimisation, which has recently become completely legit, and quick in jax (Lange et al. 2022). It would be fun to do a feasability study in silico, and then persaude some nutty experimentalist to give it a try.
- What about using the inductive bias to understand why dentate gyrus has ongoing neurogenesis?
- Perhaps we can invert this mapping, go from measured circuit to inferred task distribution, and use that to make statements about the shape of the input distribution that matters to the animal; or in general the problems that matter to the animal.

4 Actionable Representations

Science is all about building useful and accurate mental models of things in the world. Utility requires compression, we have to spot the commonalities between many disparate things in order that discoveries can be generalised and useful high-level rules extracted. In neuroscience I would argue that one of the most useful mental models has been the links between representations and computational principles. These are links from high-level normative principles to low-level empirically-verifiable neural recordings, permitting statements like "if the neural firing looks like this then these could be the computations". This seems like an important level of generality at which to attack the problem - while neural recordings are specific to this cell, at this time, in this brain structure, of this animal, of this species, ...; computational principles are general, across tasks, across individuals, across brain structures, across species, even between human and humachines. They allow summarising of vast datasets in a few principles, permitting precise predictions and flexible downsteam uses (such as new algorithms, creating BMI algorithms, or using one set of findings to test another Kandel 2007). The biggest part of my PhD work has been enlarging this link in one particular way.

Existing normative approaches tend to argue from principles that I roughly group as either functional or biological (well summarised by this PhD Thesis: Zhuo Wang 2016):

Functional goals are things like maximise the mutual information between the representation and some desired encoded variables, or ensure this collection of variables can be linearly readout from the population activity. They operationalise the word representation, and make it clear that to be a representation of x you must encode information about x, possibly in a way that ensures a noisy downstream behaviour can be executed using this encoding of x.

Biological goals are statements that try to enforce the fact that the underlying hardware (wetware) is a population of neurons. Like all claims of biological plausability, this can be enacted at numerous levels. A non-exhaustive list of possible biologically motivated representational constraints could include: the assumption that the code is a set of real numbers (i.e. firing rates), the assumption that the code is a set of spike times, energy constraints, positivity constraints, wiring minimisation constraints, networks that enact Dale's law, local learning rules.

Now, this is all well and good. But arguably is quite an impoverished view of representations. It is a feedforward view: you develop the representation of information through successive transformation from sensory periphery until some variable is informatively encoded or some behavioural readout is possible, and that's that. There seems to be a world of richness that is hard to capture in this framework. Most notably, representations are also the locus of computation; the brain is not just representing information to be studied by some homunculus, rather it is using its neurons to achieve computational goals and we should see fumes of this computation in the representation.

My work has tried to add a third criteria to predict the shape of neural responses, one that forces our representation to match what we might see if it were performing some useful computation. The key question is then of course, what computation?! And how to reflect that in representation??

We'll focus on one pertinent example that appears general enough to be useful: internal models. An internal model is some mapping between part of the world, and neural activity, such that the two follow similar rules. For example, you might have an internal model of your own body in which you can predict the consequences of motor sequences, allowing you to choose actions that will lift your mug to your face without pouring liquid embarassingly down your trousers. More broadly, possessing an internal model allows the execution of hypotheticals - what will happen if I do this? This pops up not just in motor control (Kawato and D. Wolpert 2007; D. M. Wolpert, Ghahramani, and Jordan 1995), but also in predictive coding models of sensory processing (Y. Huang and Rao 2011; Keller and Mrsic-Flogel 2018), as evidenced by the reward prediction error of the dopamine system (you had to have a hypothetical prediction to compare reality against) (Schultz, Dayan, and Montague 1997), or in almost any cognitive task, which we almost without fail conceptualise using rules and relations, that are just predictive relationships. As such, this seems a crucial operation in a world with regularity.

Now, you might reasonable argue what are we worrying about?! This, like every function, could be embedded in a feedforward network: input action and state, predict next state. All good and true, but empirically not how the brain does it (for example in the fly ring central complex S. S. Kim et al. 2017; Lyu, Abbott, and Maimon 2022; Mussells Pires, Abbott, and Maimon 2022), and I would strongly suspect that a feedforward implementation of an internal model would not match neural recordings. Rather than progressive transformations enacting this function, let's imagine there's some single population of neurons that performs this computation. What properties should it have?

To use this idea we have to make assumptions about the tools neural populations have at their disposal to computate. We need this to be the right mixture of analytically tractable and biologically plausible. Though I'd be keen to debate better choices, we've had success with matrices, i.e. state, action, & next state, s, a, s', and neural representation $\boldsymbol{g}$:

$$\boldsymbol{W}(a)\boldsymbol{g}(s) = \boldsymbol{g}(s') \tag{6}$$

Perhaps one way to argue that this choices strikes a balance between computational power and interpretability is that it has popped up all over machine learning and neuroscience! Here's my current list of how it parallels other models:

- It's been used to model relations, just as we will use it (Paccanaro and Geoffrey E Hinton 2001)
- It's been used as a model of grid cells (R. Gao, J. Xie, Wei, et al. 2021; Issa and Kechen Zhang 2012; J. C. Whittington, Muller, et al. 2020)
- It's like a bilinear collapsed along the action dimension, and bilinear models are pretty ubiquitous:
  - They pop up in machine learning as a model of how one variable influences another(Tenenbaum and Freeman 2000)
  - It's a model memory, binding two features together (Hiratani and Sompolinsky 2023)
  - It's in a proposal for transformer architectures (Sharkey 2023)
  - It's in a classic proposals for cerebellar networks (Vector Symbolic Architectures: e.g. Kleyko et al. 2022)
- It's like common ML techniques to create multiplicative interactions between two inputs (FiLM: Perez et al. 2018; AdaIN: Xun Huang and Belongie 2017; Mid-Vision Feedback: Maynord et al. 2023).
- It's like Koopman Theory, which says that all dynamics can be modelled with a linear dynamical system acting on a nonlinear function of the state (Brunton et al. 2021; Otto and Rowley 2021), except here there are different matrices at different points in time, which permits clever rule-based things to happen, and if it matches the system being modelled is likely much more efficient.
- Representations that embed the system into a linear dynamical representation have been used in RL to permit the use of control theory techniques like LQR (Banijamali et al. 2018; Watter et al. 2015; M. Zhang et al. 2019).
- If there is only one action then this update equation is just a linear RNN.
- With multiple action-dependent matrices it has links to the recent motor cortex models that focus on the role of thalamo-cortical loops (Logiaco, Abbott, and Escola 2021). In these thalamic neurons are recurrently connected to cortical neurons, but additionally each thalamic neuron is gated on or off. If you assume the thalamic dynamics are much faster than the cortical this induces low rank changes to the connectivity in the linear RNN that is used to model the whole system. (Which is additionally attractive as there's been significant analytic progress in understanding the operations of low rank nonlinear RNNs: Dubreuil et al. 2022; Mastrogiuseppe and Ostojic 2018) However, unlike this work, our changes to the connectivity are not low rank.
- It's a switching linear dynamical system, which have been successfully used to model neural data (Linderman et al. 2017).

In reality, RNN models, such as continuous attractor networks, have had a lot of success in modeling all sorts of cognitive phenomena, all the way down to connectivity (Burak and I. R. Fiete 2009; Cueva, Ardalan, et al. 2021; Mastrogiuseppe, Hiratani, and P. Latham 2023). For tractability, I've been studying this linear RNN-type thing. I'm very keen to think about ways to take steps towards biological plausability without losing tractability.

As such, this section is about an optimal neural implementation of an internal model - a particular computation. We use these ideas to understand and make predictions about grid cells in the entorhinal cortex, and music box representations in the prefrontal cortex, and I close this section with a discussion of the many ways I think these ideas could be extended.

4.1 Actionable Entorhinal Cortex: Grid Cells from Minimal Constraints

TL;DR: We propose a normative objective that asks a representation to represent position in a way that permits prediction of the consequenecs of your actions, subject to biological constriant. It produces the observed multiple modules of grid cells, in a completely understandable way, and makes many predictions. The text is taken from the main paper of Will Dorrell et al. 2023

ABSTRACT

To afford flexible behaviour, the brain must build internal representations that mirror the structure of variables in the external world. For example, 2D space obeys rules: the same set of actions combine in the same way everywhere (step north, then south, and you won't have moved, wherever you start). We suggest the brain must represent this consistent meaning of actions across space, as it allows you to find new short-cuts and navigate in unfamiliar settings. We term this representation an 'actionable representation'. We formulate actionable representations using group and representation theory, and show that, when combined with biological and functional constraints - non-negative firing, bounded neural activity, and precise coding - multiple modules of hexagonal grid cells are the optimal representation of 2D space. We support this claim with intuition, analytic justification, and simulations. Our analytic results normatively explain a set of surprising grid cell phenomena, and make testable predictions for future experiments. Lastly, we highlight the generality of our approach beyond just understanding 2D space. Our work characterises a new principle for understanding and designing flexible internal representations: they should be actionable, allowing animals and machines to predict the consequences of their actions, rather than just encode.

4.1.1 Introduction

Animals should build representations that afford flexible behaviours. However, different representation make some tasks easy and others hard; representing red versus white is good for understanding wines but less good for opening screw-top versus corked bottles. A central mystery in neuroscience is the relationship between tasks and their optimal representations. Resolving this requires understanding the representational principles that permit flexible behaviours such as zero-shot inference.

Here, we introduce **actionable representations**, a representation that permits flexible behaviours. Being actionable means encoding not only variables of interest, but also how the variable transforms. Actions cause many variables to transform in predictable ways. For example, actions in 2D space obey rules; north, east, south, and west, have a universal meaning, and combine in the same way everywhere. Embedding these rules into a representation of self-position permits deep inferences: having stepped north, then east, then south, an agent can infer that stepping west will lead home, having never taken that path - a zero-shot inference (Figure 4.1.1A).

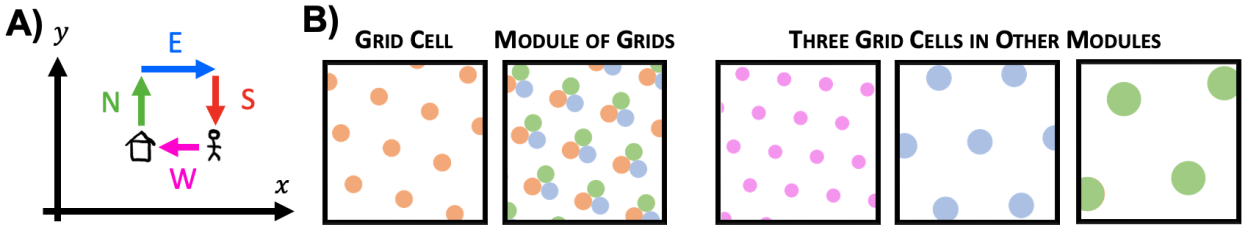

Figure 15: **A** 2D space is defined by rules, e.g. at all positions north $=-$south. **B** Left: Entorhinal grid cells are hexagonally tuned cells (orange). Different cells within a module are translated copies (orange/blue/green). Right: Different modules have different lattice scale (pink/blue/green).

Indeed biology represents 2D space in a structured manner. Grid cells in medial entorhinal cortex represent an abstracted 'cognitive map' of 2D space (Tolman 1948). These cells fire in a hexagonal lattice of positions (Hafting et al. 2005), (Figure 4.1.1B), and are organised in modules; cells within one module have receptive fields that are translated versions of one another, and different modules have firing lattices of different scales and orientations (Figure 4.1.1B), (H. Stensola et al. 2012).

Biological representations must be more than just actionable - they must be **functional**, encoding the world efficiently, and obey **biological** constraints. We formalise these three ideas - actionable, functional, and biological - and analyse the resulting optimal representations. We define *actionability* using group and representation theory, as the requirement that

each action has a corresponding matrix that linearly updates the representation; for example, the 'step north' matrix updates the representation to its value one step north. *Functionally*, we want different points in space to be represented maximally differently, allowing inputs to be distinguished from one another. *Biologically*, we ensure all neurons have non-negative and bounded activity. From this constrained optimisation problem we derive optimal representations that resemble multiple modules of grid cells.

Our problem formulation allows analytic explanations for grid cell phenomena, matches experimental findings, such as the alignment of grids cells to room geometry (T. Stensola et al. 2015), and predicts some underappreciated aspects, such as the relative angle between modules. In sum, we 1) propose actionable neural representations to support flexible behaviours; 2) formalise the actionable constraint with group and representation theory; 3) mix actionability with biological and functional constraints to create a constrained optimisation problem; 4) analyse this problem and show that in 2D the optimal representation is a good model of grid cells, thus offering a mathematical understanding of why grid cells look the way they do; 5) provide several neural predictions; 6) highlight the generality of this normative method beyond 2D space.

4.1.1.1 Related Work

Neuroscientists have long explained representations with normative principles like information maximisation (Attneave 1954; Barlow et al. 1961), sparse (Olshausen and Field 1996) or independent (Hyvärinen 2010) latent encodings, often mixed with biological constraints such as non-negativity (Sengupta et al. 2018), energy budgets (Niven, Anderson, and Laughlin 2007), or wiring minimisation (Hyvärinen, Hoyer, and Inki 2001). On the other hand, deep learning learns task optimised representations. A host of representation-learning principles have been considered (Bengio, Courville, and Vincent 2013); but our work is most related to geometric deep learning (Bronstein et al. 2021) which emphasises input transformations, and building neural networks which respect (equivariant) or ignore (invariant) them. This is similar in spirit but not in detail to our approach, since equivariant networks do not build representations in which all transformations of the input are implementable through matrices. Most related are Paccanaro and Geoffrey E Hinton 2001, who built representations in which relations (e.g. $\boldsymbol{x}$ is the father of $\boldsymbol{y}$) are enacted by a corresponding linear transform, exactly like our notion of actionable!

There is much previous theory on grid cells, which can be categorised as relating to our actionable, functional, and biological constraints. **Functional:** Many works argue that grid cells provide an efficient representation of position, that hexagons are optimal (Mathis, Herz, and M. Stemmler 2012; Mathis, Herz, and M. B. Stemmler 2012; Sreenivasan and I. Fiete 2011; Wei, Prentice, and Balasubramanian 2015) and make predictions for relative module lengthscales (Wei, Prentice, and Balasubramanian 2015). Since we use similar functional principles, we suspect that some of our novel results, such as grid-to-room alignment, could have been derived by these authors. However, in contrast to our work, these authors assume a grid-like tuning curve. Instead we give a normative explanation of why be grid-like at all, explaining features like the alignment of grid axes within a module, which are detrimental from a pure decoding view (M. Stemmler, Mathis, and Herz 2015). **Actionability:** Grid cells are thought to a basis for predicting future outcomes (Stachenfeld, M. M. Botvinick, and Gershman 2017; C. Yu, Behrens, and N. Burgess 2020, and have been classically understood as affording path-integration (integrating velocity to predict position) with units from both hand-tuned (Burak and I. R. Fiete 2009) and trained recurrent neural network resembling grid cells (Banino et al. 2018; Cueva and Wei 2018; Sorscher, G. Mel, et al. 2019). Recently, these recurrent network approaches have been questioned for their parameter dependence (Schaeffer, Khona, and I. Fiete 2022), or relying on decoding place cells with bespoke shapes that are not observed experimentally (Dordek et al. 2016; Sorscher, G. Mel, et al. 2019). Our mathematical formalisation of path-integration, combined with biological and functional constraints, provides clarity on this issue. Our approach is linear, in that actions update the representation linearly, which has previously been explored theoretically (Issa and Kechen Zhang 2012), and numerically, in two works that learnt grid cells (R. Gao, J. Xie, Wei, et al. 2021; J. C. Whittington, Muller, et al. 2020). Our work could be seen as extracting and simplifying the key ideas from these papers that make hexagonal grids optimal (see Appendix H of Will Dorrell et al. 2023), and extending them to multiple modules, something both papers had to hard code. **Biological:** Lastly, both theoretically (Sorscher, G. Mel, et al. 2019 and computationally (Dordek et al. 2016; J. C. Whittington, Warren, and Behrens 2021, non-negativity has played a key role in normative derivations of hexagonal grid cells, as it will here.

4.1.2 Actionable Neural Representations: An Objective

We seek a representation $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^N$ of 2D position $\boldsymbol{x} \in \mathbb{T}^2$, where N is the number of neurons. Our representation is built using three ideas: functional, biological, and actionable; whose combination will lead to multiple modules of grid cells, and which we'll now formalise.

Functional: To be useful, the representation must encode different positions differently. However, it is more important to distinguish positions 1km apart than 1mm, and frequently visited positions should be separated the most. To account for these, we ask our representation to minimise

$$\mathcal{L} = \iint e^{-\frac{\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|^2}{2\sigma^2}} \chi(\boldsymbol{x},\boldsymbol{x}')p(\boldsymbol{x})p(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}' \tag{7}$$

The red term measures the representational similarity of $\boldsymbol{x}$ and $\boldsymbol{x}'$; it is large if their representations are nearer than some distance σ in neural space and small otherwise. By integrating over all pairs $\boldsymbol{x}$ and $\boldsymbol{x}'$, $\mathcal{L}$ measures the total representational similarity, which we seek to minimise. The green term is the agent's position occupancy distribution, which ensures only visited points contribute to the loss, for now simply a Gaussian of lengthscale L. Finally, the blue term weights the importance of separating each pair, encouraging separation of points more distant than a lengthscale, l.

$$\chi(\boldsymbol{x},\boldsymbol{x}') = 1 - e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2l^2}} \qquad p(\boldsymbol{x}) = e^{-\frac{\|\boldsymbol{x}\|^2}{2L^2}} \tag{8}$$

Biological: Neurons have non-negative firing rates, so we constrain $\boldsymbol{g}_i(\boldsymbol{x}) \geq 0$. Further, neurons can't fire arbitrarily fast, and firing is energetically costly, so we constrain each neuron's response $g_n(\boldsymbol{x})$ via $\int g_n^2(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = 1$

Actionable: Our final constraint requires that the representation is actionable. This means each transformations of the input must have its own transformation in neural space, independent of position. For mathematical convenience we enact the neural transformation using a matrix. Labelling this matrix $\boldsymbol{T}(\Delta\boldsymbol{x}) \in \mathbb{R}^{N\times N}$, for transformation $\Delta\boldsymbol{x}$, this means that for all positions $\boldsymbol{x}$,

$$\boldsymbol{g}(\boldsymbol{x}+\Delta\boldsymbol{x}) = \boldsymbol{T}(\Delta\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x}) \tag{9}$$

For intuition into how this constrains the neural code $\boldsymbol{g}(\boldsymbol{x})$, we consider a simple example of two neurons representing an angle $\theta \in [0, 2\pi)$. Replacing $\boldsymbol{x}$ with θ in equation 9 we get the equivalent constraint: $\boldsymbol{g}(\theta+\Delta\theta) = \boldsymbol{T}(\Delta\theta)\boldsymbol{g}(\theta)$. Here the matrix $\boldsymbol{T}$ performs a rotation, and the solution (up to a linear transform) is for $\boldsymbol{T}$ to be the standard 2×2 rotation matrix, with frequency n.

$$\boldsymbol{g}(\theta+\Delta\theta) = \begin{pmatrix}\cos(n[\theta+\Delta\theta])\\ \sin(n[\theta+\Delta\theta])\end{pmatrix} = \begin{pmatrix}\cos(n\Delta\theta) & -\sin(n\Delta\theta)\\ \sin(n\Delta\theta) & \cos(n\Delta\theta)\end{pmatrix}\begin{pmatrix}\cos(n\theta)\\ \sin(n\theta)\end{pmatrix} = \boldsymbol{T}(\Delta\theta)\boldsymbol{g}(\theta) \tag{10}$$

Thus as θ rotates by 2π the neural representation traces out a circle an integer number, n, times. Thanks to the problem's linearity, extending to more neurons is easy. Adding two more neurons lets the population contain another sine and cosine at some frequency, just like the two neurons in equation 10. Extrapolating this we get our actionability constraint: the neural response must be constructed from some invertible linear mixing of the sines and cosines of $D < \frac{N}{2}$ frequencies,

$$\boldsymbol{g}(\theta) = \boldsymbol{a}_0 + \sum_{d=1}^{D} \boldsymbol{a}_d \sin(n_d\theta) + \boldsymbol{b}_d \cos(n_d\theta) \qquad \text{for integer } n_d \tag{11}$$

The vectors $\{\boldsymbol{a}_d, \boldsymbol{b}_d\}_{d=1}^{D} \in \mathbb{R}^N$ are coefficient vectors that mix together the sines and cosines, of which there are D. $\boldsymbol{a}_0$ is the coefficient vector for a frequency that cycles 0 times.

This argument comes from an area of maths called Representation Theory (a different meaning of representation!) that places constraints on the matrices $\boldsymbol{T}$ for variables whose transformations form a mathematical object called a group. This includes many of interest, such as position on a circle, torus, or sphere. These constraints on matrices can be translated into constraints on an actionable neural code just like we did for $\boldsymbol{g}(\theta)$ (see Appendix A of Will Dorrell et al. 2023). When generalising the above example to 2D space (a torus), we must consider a few things: First, the space is two-dimensional, so compared to our previous equation 11, the frequencies, denoted $\boldsymbol{k}_d$, are now two dimensional. Second, to approximate a finite region of flat 2D space, we consider a similarly sized region of a torus. As the radius of the torus grows this approximation becomes arbitrarily good (see Appendix A.4 of Will Dorrell et al. 2023 for discussion). Periodicity constrains the frequencies in equation 11 to be $\frac{n}{R}$ for integer n and ring radius R. As the loop (torus in 2D) becomes very large these permitted frequencies become arbitrarily close, so we drop the integer constraint,

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{a}_0 + \sum_{d=1}^{D} \boldsymbol{a}_d \sin(\boldsymbol{k}_d\cdot\boldsymbol{x}) + \boldsymbol{b}_d \cos(\boldsymbol{k}_d\cdot\boldsymbol{x}) \tag{12}$$

Our constrained optimisation problem is complete. Equation 12 specifies the set of actionable representations. Without additional constraints these codes are meaningless: random combinations of sines and cosines produce random neural responses (Figure 16A). We will choose from amongst the set of actionable codes by optimising the parameters $\boldsymbol{a}_0, \{\boldsymbol{a}_d, \boldsymbol{b}_d, \boldsymbol{k}_d\}_{d=1}^{D}$ to minimise $\mathcal{L}$, subject to biological (non-negative and bounded firing rates) constraints.

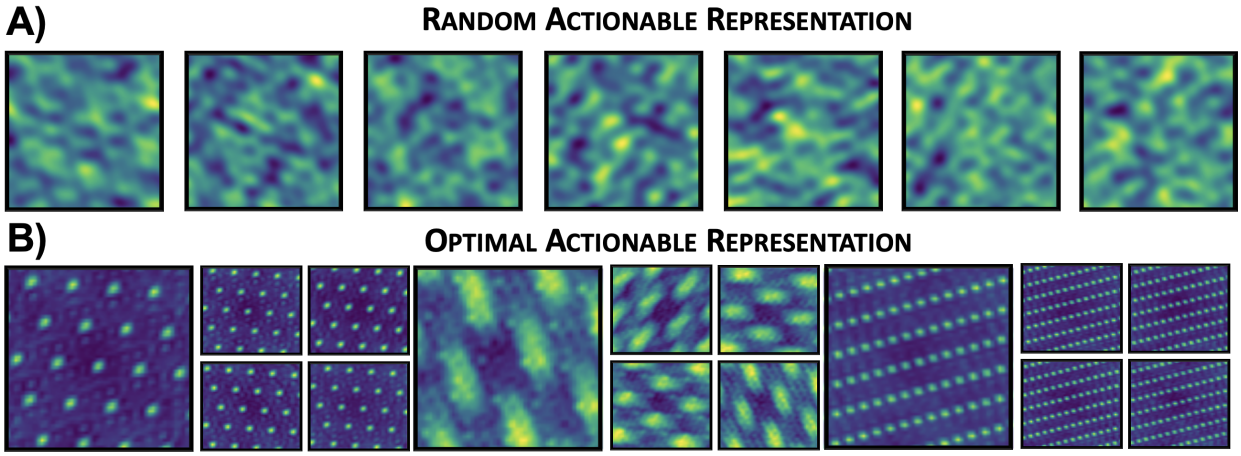

Figure 16: **A** Random actionable representations (equation 12) are meaningless combinations of sines and cosines. ($g_n(\boldsymbol{x})$ plotted for different neurons, n) **B** Optimising among actionable codes to achieve functional and biological constraints produces multiple modules of ∼hexagonal grid cells. (Figure 6 in Will Dorrell et al. 2023 shows that all the neurons in the population belong to one of these three modules)

4.1.3 Optimal Representations

Optimising over the set of actionable codes to minimise $\mathcal{L}$ with biological constraints gives multiple modules of grid cells (Figure 16B). This section will, using intuition and analytics, explain why.

4.1.3.1 Non-negativity Leads to a Module of Lattice Cells

To understand how non-negativity produces modules of lattice responses we will study the following simplified loss, which maximises the *Euclidean distance* between representations of angle, $\boldsymbol{g}(\theta)$,

$$\mathcal{L}_0 = -\frac{1}{4\pi^2}\iint_{-\pi}^{\pi} \|\boldsymbol{g}(\theta) - \boldsymbol{g}(\theta')\|^2 d\theta d\theta' \tag{13}$$

This is equivalent to the full loss (equation 7) for uniform $p(\theta)$, $\chi(\theta, \theta') = 1$, and σ very large. Make no mistake, this is a bad loss. For contrast, the full loss encouraged the representations of different positions to be separated by more than σ, enabling discrimination[1]. Therefore, sensibly, the representation is most rewarded for separating nearby (closer than σ) points. $\mathcal{L}_0$ does the opposite! It grows quadratically with separation, so $\boldsymbol{g}(\theta)$ is most rewarded for pushing apart already well-separated points, a terrible representational principle! Nonetheless, $\mathcal{L}_0$ will give us key insights.

Since actionability gives us a parameterised form of the representations (equation 11), we can compute the integrals to obtain the following constrained optimisation problem (details: Appendix C in Will Dorrell et al. 2023)

$$\min_{\substack{\boldsymbol{a}_0,\\ \{\boldsymbol{a}_d,\boldsymbol{b}_d,n_d\}_{d=1}^D}} \mathcal{L}_0 = -\sum_{d=1}^{D} \|\boldsymbol{a}_d\|^2 + \|\boldsymbol{b}_d\|^2 \quad \text{with} \quad \overbrace{\boldsymbol{g}(\theta) > 0,}^{\text{Non-negativity}} \quad \overbrace{\|\boldsymbol{a}_0\|^2 + \frac{1}{2}\sum_{d=1}^{D} \|\boldsymbol{a}_d\|^2 + \|\boldsymbol{b}_d\|^2 = N}^{\text{Bounded firing rates}} \tag{14}$$

Where N is the number of neurons. This is now something we can understand. First, reminding ourselves that the neural code, $\boldsymbol{g}(\theta)$, is made from a constant vector, $\boldsymbol{a}_0$, and $\theta-$dependent parts (equation 11; Figure 17A), we can see that $\mathcal{L}_0$ separates representations by encouraging the size of each varying part, $\|\boldsymbol{a}_d\|^2 + \|\boldsymbol{b}_d\|^2$, to be maximised. This effect is limited by the firing rate bound, $\|\boldsymbol{a}_0\|^2 - \frac{1}{2}\mathcal{L}_0 = N$. Thus, to minimise $\mathcal{L}_0$ we must minimise the constant vector, $\boldsymbol{a}_0$. This would be easy without non-negativity (when any code with $\|\boldsymbol{a}_0\| = 0$ is optimal), but no sum of sines and cosines can be non-negative for all θ without an offset. Thus the game is simple; choose frequencies and coefficients so the firing rates are non-negative, but using the smallest possible constant vector.

[1] σ could be interpreted as a noise level, or a minimum discriminable distance, then points should be far enough away for a downstream decoder to distinguish them.

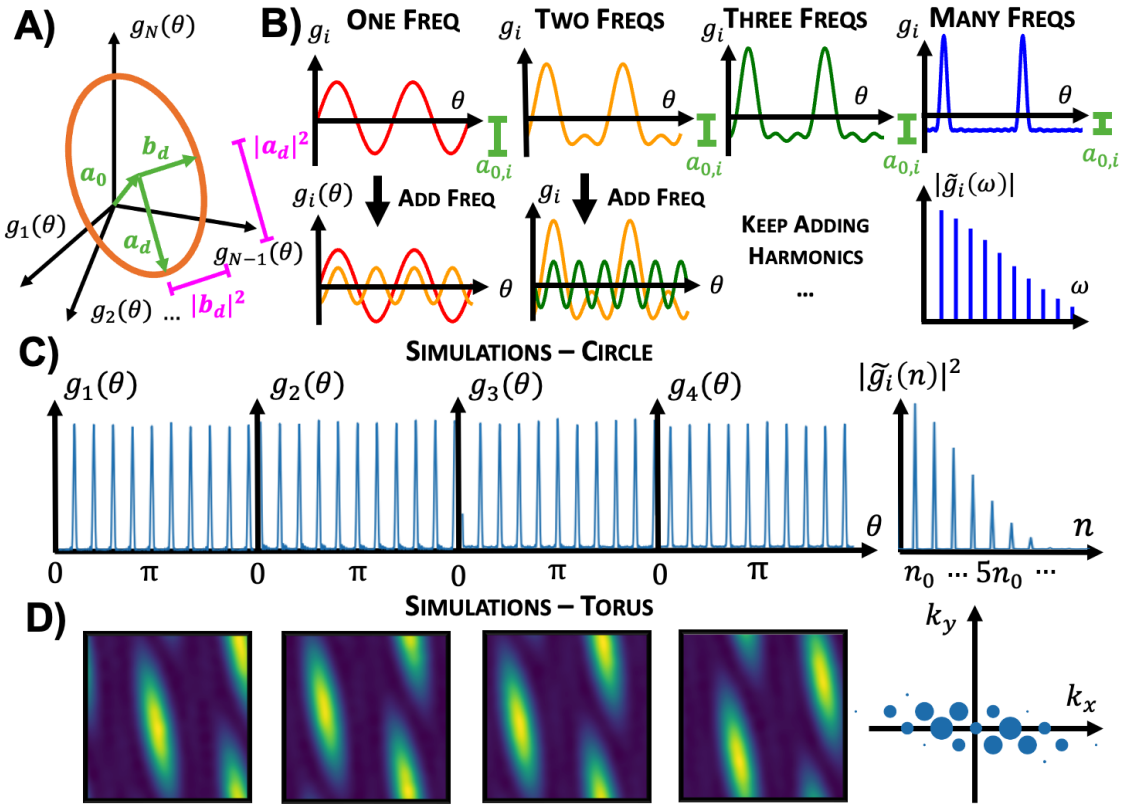

Figure 17: **A** The neural activity consists of a constant vector, $\boldsymbol{a}_0$, and $\theta-$dependent loops. **B** Progressively adding harmonic frequencies increases the code's minima, allowing the code to be made non-negative using the smallest possible $\boldsymbol{a}_0$. This give a grid-like tuning curve. Simulations results confirm this heuristic in **C** 1D and **D** 2D, right: frequency spectrum, 2D dot size is frequency power.

One lattice cell. We now heuristically argue, and confirm in simulation, that the optimal solution for a single neuron is a lattice tuning curve (see Appendix F in Will Dorrell et al. 2023 for why this problem is non-convex). Starting with a single frequency component, e.g. $\sin(\theta)$, achieving non-negativity requires adding a constant offset, $\sin(\theta)+1$ (Figure 17B). However, we could also have just added another frequency. In particular adding harmonics of the base frequency (with appropriate phase shifts) pushes up the minima (Figure 17B). Extending this argument, we suggest non-negativity, for a single cell, can be achieved by including a grid of frequencies. This gives a lattice tuning curve (Figure 17B right).

Module of lattice cells. Achieving non-negativity for this cell used up many frequencies. But as discussed (Section 4.1.2), actionability only allows a limited number frequencies in the population ($< \frac{N}{2}$ since each frequency uses 2 neurons (sine and cosine)), thus how can we make lots of neurons non-negative with limited frequencies? Fortunately, we can do so by making all neuron's tuning curves translated versions of each other, as translated curves contain the same frequencies but with different phases. This is a module of lattice cells. We validate our arguments by numerically optimising the coefficients $\boldsymbol{a}_0, \{\boldsymbol{a}_d, \boldsymbol{b}_d\}_{d=1}^D$ and frequencies $\{n_d\}_{d=1}^D$ to minimise $\mathcal{L}_0$ subject to constraints, producing a module of lattices (Figure 17C; details in Appendix B of Will Dorrell et al. 2023). These arguments equally apply to representations of a periodic 2D space (a torus; Figure 17D).

Studying $\mathcal{L}_0$ has told us why lattice response curves are good. But surprisingly, all lattices are equally good, even at infinitely high frequency. Returning to the full loss will break this degeneracy.

4.1.3.2 Prioritising Important Pairs of Positions Produces Hexagonal Grid Cells

Now we return to the full loss and understand its impact in two steps, beginning with the reintroduction of χ and p, which break the lattice degeneracy, forming hexagonal grid cells.

$$\mathcal{L} = \iint_{-\infty}^{\infty} e^{-\frac{\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|^2}{2\sigma^2}} \chi(\boldsymbol{x}, \boldsymbol{x}') p(\boldsymbol{x}) p(\boldsymbol{x}') d\boldsymbol{x} d\boldsymbol{x}' \tag{15}$$

χ prefers low frequencies: recall that $\chi = 1 - e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2l^2}}$ ensures very distant inputs have different representations, while allowing similar inputs to have similar representations, up to a resolution, l. This encourages low frequencies ($\|\boldsymbol{k}_d\| < \frac{1}{l}$), which separate distant points but produce similar representations for pairs closer than l (Analytics: Appendix D of Will Dorrell et al. 2023). At this stage, for periodic 2D space, the lowest frequency lattices, place cells, are optimal (see Appendix F of Will Dorrell et al. 2023; Sengupta et al. 2018).

$p(\boldsymbol{x})$ prefers high frequencies: However, the occupancy distribution of the animal, $p(\boldsymbol{x})$, counters χ. On an infinite 2D plane animals must focus on representing a limited area, of lengthscale L. This encourages high frequencies ($\|\boldsymbol{k}_d\| > \frac{1}{L}$), whose response varies among the visited points (Analytics: Appendix D of Will Dorrell et al. 2023). More complex $p(x)$ induce more complex frequency biases, but, to first order, the effect is always a high frequency bias (Figure 19F-G, Appendix L of Will Dorrell et al. 2023).

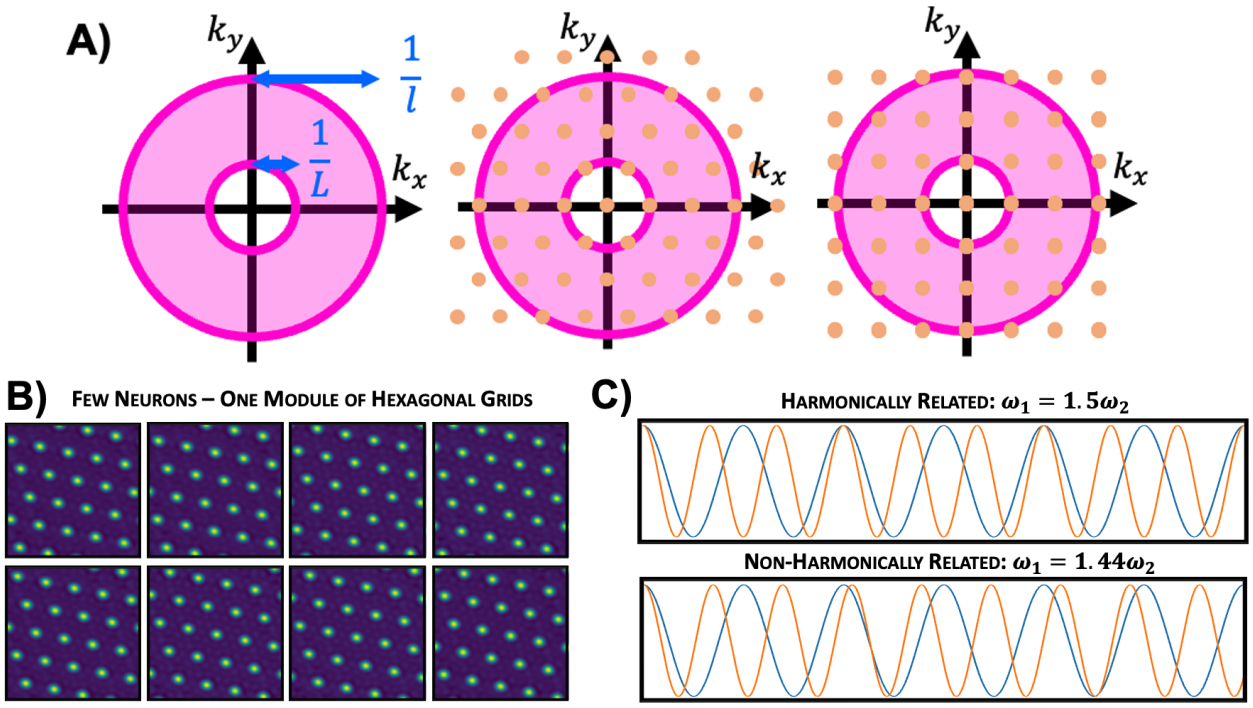

Figure 18: **A** χ and $p(\boldsymbol{x})$ induce a bias towards frequencies with magnitudes between $\frac{1}{L}$ and $\frac{1}{l}$. Since, of all lattices, hexagons fit the most frequencies within the annulus, they are preferred, and hexagonal frequency lattices lead to hexagonal grid cells. **B** Simulations confirm. **C** Harmonically related frequencies co-repeat more often than non-harmonic, meaning that, as a pair, harmonically related frequencies are worse at encoding, since they encode many points in the same way.

Combination → Hexagons: Satisfying non-negativity and functionality required a lattice of many frequencies, but now p and χ bias our frequency choice, preferring those beyond $\frac{1}{L}$ (to separate points the animal visits) but smaller than $\frac{1}{l}$ (to separate distant visited points). Thus to get as many of these preferred frequencies as possible, we want the lattice with the densest packing within a Goldilocks annulus in frequency space (Figure 18A). This is a hexagonal lattice in frequency space which leads to a hexagonal grid cell. Simulations with few neurons agree, giving a module of hexagonal grid cells (Figure 18B).

4.1.3.3 A Harmonic Tussle Produces Multiple Modules

Finally, we will study the neural lengthscale σ, and understand how it produces multiple modules.

$$\mathcal{L} = \iint_{-\infty}^{\infty} e^{-\frac{\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|^2}{2\sigma^2}} \chi(\boldsymbol{x}, \boldsymbol{x}') p(\boldsymbol{x}) p(\boldsymbol{x}') d\boldsymbol{x} d\boldsymbol{x}' \tag{16}$$

As discussed, $\mathcal{L}$ prioritises the separation of poorly distinguished points, those whose representations are closer than σ. This causes certain frequencies to be desired *in the overall population*, in particular those unrelated to existing frequencies by simple harmonic ratios, i.e. not $\omega_1 = \frac{3}{2}\omega_2$ (Figure 18C; see Appendix E of Will Dorrell et al. 2023 for a perturbative derivation of this effect). This is because pairs of harmonically related frequencies represent more positions identically than a non-harmonically related pair, so are worse for separation (similar to arguments made in Wei, Prentice, and Balasubramanian 2015).

This, however, sets up a 'harmonic tussle' between what the population wants - non-harmonically related frequencies for $\mathcal{L}$ - and what single neurons want - harmonically related frequency lattices for non-negativity (Section 4.1.3.1). Modules of grid cells resolve this tension: harmonic frequencies exist within modules to give non-negativity, and non-harmonically related modules allow for separation, explaining the earlier simulation results (Figure 16B; further details in Appendix E.4 of Will Dorrell et al. 2023).

This concludes our main result. We have shown three constraints on neural populations - actionable, functional, and biological - lead to multiple modules of hexagonal grid cells, and we have understood why. We posit this is the minimal set of requirements for grid cells (see Appendix I of Will Dorrell et al. 2023 for ablations simulations and discussion).

4.1.4 Predictions

Our theory makes testable predictions about the structure of optimal actionable codes for 2D space. We describe three here: tuning curve sharpness scales with the number of neurons in a module; the optimal angle between modules; and the optimal grid alignment to room geometry.

4.1.4.1 Lattice Size:Field Width Ratio scales with Number of Neurons in Module

In our framework the number of neurons controls the number of frequencies in the representation (equation 12). A neuron within a module only contains frequencies from that module's frequency lattice, since other modules have non-harmonically related frequencies. More neurons in a module, means more and higher frequencies in the lattice, which sharpen grid peaks (Figure 5A). We formalise this (Appendix J of Will Dorrell et al. 2023) and predict that the number of neurons within a module scales with the square of the lattice lengthscale, ν, to field width, μ, ratio, $N \propto \left(\frac{\nu}{\mu}\right)^2$. This matches the intuition that the sharper a module's peak, the more neurons you need to tile the entire space. In a rudimentary analysis, our predictions compare favourably to data from H. Stensola et al. 2012 assuming uniform sampling of grid cells across modules (Figure 5B). We are eager to test these claims quantitatively.

4.1.4.2 Modules are Optimally Oriented at Small Offsets ($\sim 4^\circ$)

In section 4.1.3.3 we saw how frequencies of different modules are maximally non-harmonically related in order to separate the representation of as many points as possible. To maximise non-harmonicity between two modules, the second module's frequency lattice can be both stretched *and* rotated relative to the first. 0 or 30° relative orientations are particularly bad coding choices as they align the high density axes of the two lattices (Figure 5C). The optimal angular offset of two modules, calculated via a frequency overlap metric (Appendix K of Will Dorrell et al. 2023), is small (Figure 5D); the value depends on the grid peak and lattice lengthscales, μ and ν, but varies between 3° and 8° degrees. Multiple modules should orient at a sequence of small angles (Appendix K of Will Dorrell et al. 2023). In a rudimentary analysis, our predictions compare favourably to the observations of H. Stensola et al. 2012 (Figure 5E).

4.1.4.3 Optimal Grids Morph to Room Geometry

In Section 4.1.3.2 (and Appendix D of Will Dorrell et al. 2023) we showed that $p(x)$, the animal's occupancy distribution, introduced a high frequency bias - grid cells must peak often enough to encode visited points. However, changing $p(x)$ changes the shape of this high frequency bias (Appendix L of Will Dorrell et al. 2023). In particular, we examine an animal's encoding of square, circular, or rectangular environments, Appendix L of Will Dorrell et al. 2023, with

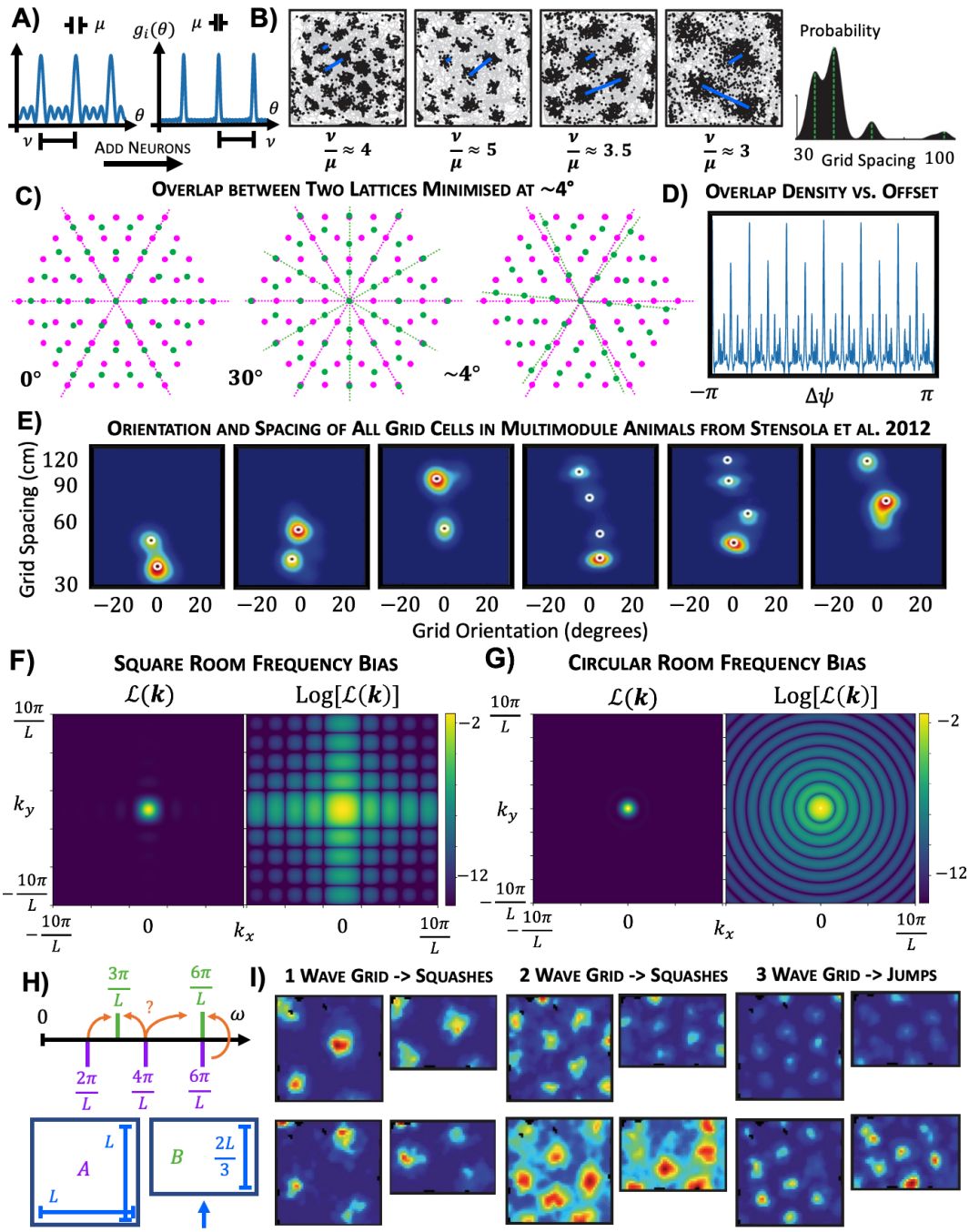

Figure 19: **A** Adding neurons to a module sharpens the grid peaks. **B** In data from H. Stensola et al. 2012 the most sharply peaked grids were recorded most often (2nd from left), and the broadest the least (rightmost). **C** Aligning the high-density axes of two lattices creates high overlap, while small offsets minimise it. **D** We quantified the overlap as a function of offset angle (Appendix J of Will Dorrell et al. 2023 and found the minima occured at $\sim 4^\circ$ from aligned (aligned = 6 maxima, 4° offsets are the 12 minima). **E** The orientation and spacing of all grids in animals with multiple modules recorded by H. Stensola et al. 2012. Many modules are misaligned by small offsets, matching our prediction. **F** Square or **G** circular rooms create complex frequency biases, $\mathcal{L}(\boldsymbol{k})$ is the loss for a one frequency code of fixed amplitude. **H** The optimal frequencies along one axis of a box occur at $\frac{2\pi n}{L}$ for integer n: Squishing a room makes the optimal frequencies expand. Grids should change to fit the optimal patterns in the recorded environment, unless they happen to be optimal for both, as $\frac{6\pi}{L}$ is. **I** Most grid cells scale with the room, but, when one side is squashed by a factor of 2/3, those at $\frac{6\pi}{L}$ are stable.

the assumption that $p(x)$ is uniform over that space. In each case the bias is coarsely towards high frequencies, but has additional intricacies: in square and rectangular rooms optimal frequencies lie on a lattice, with peaks at integer multiples of $\frac{2\pi}{L}$ along one of the cardinal axes, for room width/height L (Figure 5F); whereas in circular rooms optima are at the zeros of a Bessel function (Figure 5G). These ideas make several predictions. For example, grid modules in circular rooms should have lengthscales set by the optimal radii in Figure 5G, but they should still remain hexagonal since the Bessel function is circularly symmetric. However, the optimal representation in square rooms should not be perfectly hexagonal since $p(x)$ induces a bias inconsistent with a hexagonal lattice (this effect is negligible for high frequency grids). Intriguingly, shearing towards squarer lattices is observed in square rooms (T. Stensola et al. 2015, and it would be interesting to test its grid-size dependence.

Lastly, these effects make predictions about how grid cells morph when the environment geometry changes. A grid cell that is optimal in both geometries can remain the same, however sub-optimal grid cells should change. For example turning a square into a squashed square (i.e. a rectangle), stretches the optimal frequencies along the squashed dimension. Thus, some cells are optimal in both rooms and should stay stable, will others will change, presumably to nearby optimal frequencies (Figure 19H). Indeed H. Stensola et al. 2012 recorded the same grid cells in a square and rectangular environment (Figure 19I), and observed exactly these phenomena.

4.1.5 Discussion & Conclusions

We have proposed actionability as a fundamental representational principle to afford flexible behaviours. We have shown in simulation and with analytic justification that the optimal actionable representations of 2D space are, when constrained to be both biological and functional, multiple modules of hexagonal grid cells, thus offering a mathematical understanding of grid cells. We then used this theory to make three novel grid cell predictions that match data on early inspection.

While this is promising for our theory, there remain some grid cell phenomena that, as it stands, it will never predict. For example, grid cell peaks vary in intensity (Dunn et al. 2017, and grid lattices bend in trapezoidal environments (Krupic et al. 2015. These effects may be due to incorporation of sensory information or uncertainty - things we have not included - to better infer position. Including these may recapitulate these findings, similar to Ocko et al. 2018 and Kang, D. M. Wolpert, and Lengyel 2023.

Our theory is normative and abstracted from implementation. However, both modelling (Burak and I. R. Fiete 2009 and experimental (Gardner et al. 2022; S. S. Kim et al. 2017 work suggests that continuous attractor networks (CANs) implement path integrating circuits. Actionability and CANs imply seemingly different representation update equations; future work could usefully compare the two.

While we focused on understanding the optimal representations of 2D space and their relationship to grid cells, our theory is more general. Most simply, it can be applied to behaviour in other, non 2D, spaces. In fact many variables whose transformations form a group are relatively easily analysed. The brain represents many such variables, e.g. heading directions, (Finkelstein et al. 2015, object orientations, (Logothetis, Pauls, and Poggio 1995, the'4-loop task' of Sun et al. 2020 or 3-dimensional space (Ginosar et al. 2021; Grieves et al. 2021. Interestingly, our theory predicts 3D representations with regular order (Figure 19E in Appendix M of Will Dorrell et al. 2023), unlike those found in the brain (Ginosar et al. 2021; Grieves et al. 2021 suggesting the animal's 3D navigation is sub-optimal.

Further, the brain represents these differently-structured variables not one at a time, but simultaneously; at times mixing these variables into a common representation (Hardcastle et al. 2017, at others giving each variable its own set of neurons (e.g. grid cells, object-vector cells Høydal et al. 2019). Thus, one potential concern about our work is that it assumes a separate neural population represents each variable. However, in a companion paper, we show that our same biological and functional constraints encourage any neural representation to encode independent variables in separate sub-populations (J. C. R. Whittington, Will Dorrell, et al. 2023), to which our theory can then be cleanly applied.

But, most expansively, these principles express a view that representations must be more than just passive encodings of the world; they must embed the consequences of predictable actions, allowing planning and inferences in never-before-seen situations. We codified these ideas using Group and Representation theory, and demonstrated their utility in understanding grid cells. However, the underlying principle is broader than the crystalline nature of group structures: the world and your actions within it have endlessly repeating structures whose understanding permits creative analogising and flexible behaviours. A well-designed representation should reflect this.

4.2 Actionable Prefrontal Cortex: Normative Music Boxes

TL;DR: Recent work has shown a surprising 'programmable music box' representation: neurons can be configured to encode a sequence of behaviours, like a music box can be configured to play a sequence of tones. Mice use this representation to solve a simple, but very structured, repeating task. We use actionability to predict the cartoon version of these neural responses; there remain many exciting features to think about.

4.2.1 Experiment and Measurement

Mohamady El-Gaby in the Behrens lab has trained mice to perform something called the ABCD task. There are 9 food ports; during a particular trial four of these are rewarded in a sequence, figure 20 top right, so the mouse has to travel to port A then port B, then C, then D, then back to A again, to get a reward. After a period of time the trial changes, meaning the ABCD locations permute amongst the 9 possible locations. Mohamady then measured the prefrontal cortical representations as the mouse learns and performs this task.

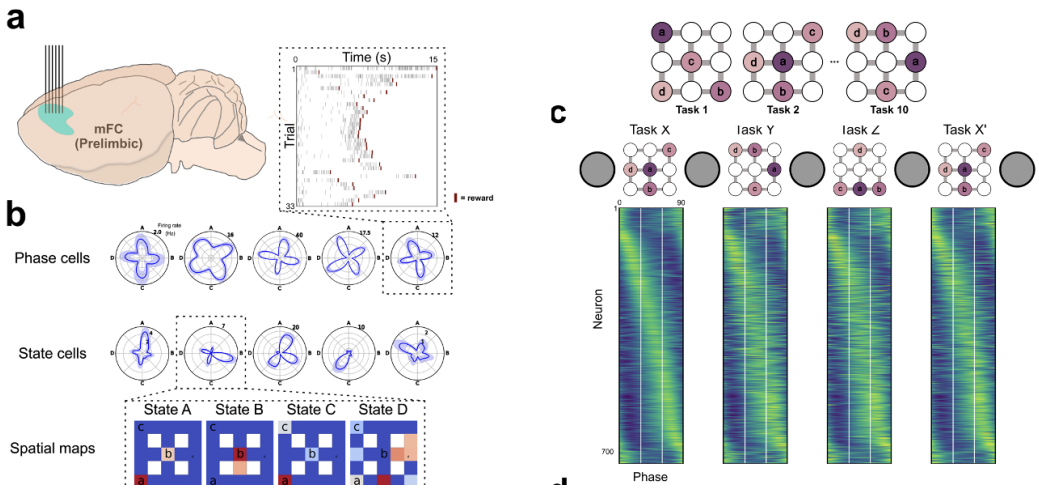

Figure 20: **Top right:** Schematic of task structure, there are 9 food locations, at any one time 4 of them are rewarded in a particular order, marked ABCD. After a time the task changes, meaning the four locations permute. **A:** Neural recordings frmo mFC. Single neuron trace showing that it responds just before the reward completely independently of time taken or distance moved. **B:** Examples of neural responses, shown in the top row, neurons that respond to 'phase' i.e. progress towards current goal location, independent of where in the task it is. Bottom row: 'state cells' that respond in one particular point in the ABCD loop. They have a consistent phase tuning, and a location tuning that often changes between tasks. **C:** Across the whole population you have neurons that tile the phase tuning space (shockingly 80% of neurons are robustly phase tuned!!). They are consistent across task.

The representation that he finds is wild, it implements two suprising things:

- First, the prefrontal cortical firing strongly correlates with scaled progress towards goals in a scaled way; meaning it doesn't encode time to goal, nor distance to goal, instead it encodes percentage progress from start to end, figure 20. How does it work this out? We will not make progress to understanding this here.
- Second, it implements a programmable music box that sequences the mouse's movement, figure 21. In this analogy we can consider a cartoon representation where the 9 notes that can be played are the 9 locations, and the programming chooses which 4 to be played in this task, and in what order. This is implemented in a simple set of ring attractors. You put activity into four locations on this cylinder, each horizontal location corresponds to a location, each location on the circle corresponds to a point in the sequence. The activity then rotates around these ring attractors at the same pace implementing the turning of the music box. When each activity bump reaches the top of the ring it 'plays' telling the mouse to go there. WOW!

We will present a normative, work-in-progress, theory, based on actionability, that predicts this programmable music box, giving it a normative interpretation. We will then suggest ways in which the theory can be extended to capture more of the findings, or ways in which the theory will never fit the data.

4.2.2 An Objective

We consider a representation of a sequence of positions, $\boldsymbol{g}(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector of locations, $\theta_i \in \{1, ..., 9\}$. The first element of the vector is the next location the mouse has to go to, and so on down the vector. Similar to the grid cells, our normative representational theory has three classes of ideas. **Functional:** The mouse must use this representation to

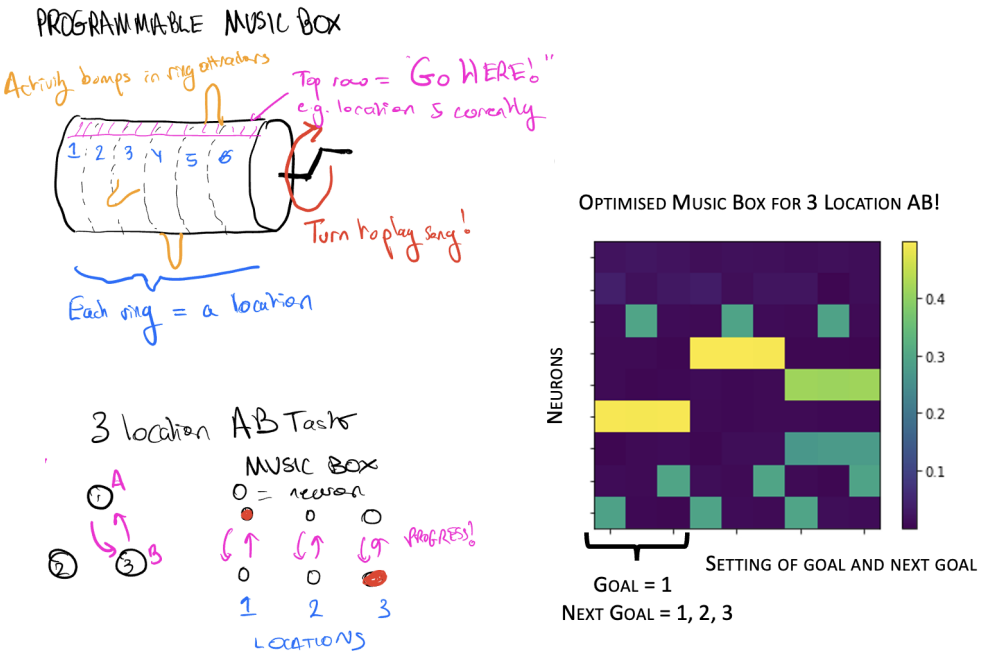

Figure 21: **Top left:** Schematic of the music box: a barrel of rings, each ring corresponds to a particular location. In any one task you put bumps of activity in four of these rings offset by 90 degrees. The bump at the top of the music box shouts: go to my ring's location! Then once you do that you rotate the rings all by 90 degrees, so a new location shouts "come here!". In this way you sequence a cognitive habit. **Bottom left:** For illustration purposes, consider a three location AB task. We can encode such a task in a 6 neuron music box, each pair of neurons is a particular location, the top row of three are the goal coding neurons. For example, goal = location 1, next goal = 3, can be encoded as shown in the red bumps. Then when you reach 1 you swap the top and bottom rows of activity. **Right:** Optimal representation under the proposed optimisation problem. Each row is a neuron, each column is a setting of goal and next goal. As you can see, each neuron codes for a particular location being either goal or next goal, like the music box!

know where to go, as such, we ask that it can linearly readout the desired location. We do this by asking for there to be a fixed matrix that maps the representation to a one-hot encoding, $\boldsymbol{o}$, of the next location to go to:

$$\boldsymbol{R}\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{o}(\theta_1) \tag{17}$$

Actionable: Independent of current position, the mouse must be able to progress its representation when one goal is reached and find out where to go next. We will do this progression with a matrix:

$$\boldsymbol{W}\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{g}([\boldsymbol{\theta}_{2:4}, \boldsymbol{\theta}_1]) \tag{18}$$

This links the representations within the same task. I'm not sure whether we need this, but for the moment I also introduce some ideas to link representations across different tasks. Assume the mouse wants to be able to embed these memories in the same way independent of where in the sequence they appear. We will say that this happens through pulsing in a set of activity that only depends on the position being encoded. Then it progresses the representation and embeds the next memory. Calling these memories $\boldsymbol{e}(\theta)$, we get the representation:

$$\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{e}(\theta_1) + \boldsymbol{W}\boldsymbol{e}(\theta_2) + \boldsymbol{W}^2\boldsymbol{e}(\theta_3) + \boldsymbol{W}^3\boldsymbol{e}(\theta_4) \tag{19}$$

I would love to try without this last constraint. For now, suffice to say that it works!

Biological: Comes in three flavours:

- The representation must be positive
- We want to minimise the energy usage, specifically the energy used in spiking. To do that we will try to achieve the other goals while minimising the average L2 norm of the activity.
- Synapses also contribute to energy usage, and we don't want them to be large, so we will penalise the L2 norm of all the weights. In this case that corresponds to the readout matrix $\boldsymbol{R}$, and the recurrent weights $\boldsymbol{W}$; we can either add them to the loss to be minimised or ascribe them a finite frobenius norm.

4.2.3 Optimal Code

We minimise the energy loss subject to the functional, actionable, and biological constraints, by optimising the parameters that define the memories $\boldsymbol{e}$. When we do that we get a music box representation! A simplified 3 location AB task is shown in figure 21 (along with a guide to understanding why that it is the music box prediction; then a 6 location ABC task is shown in figure 22 (it generalises to 9 location ABCD also, just not shown).

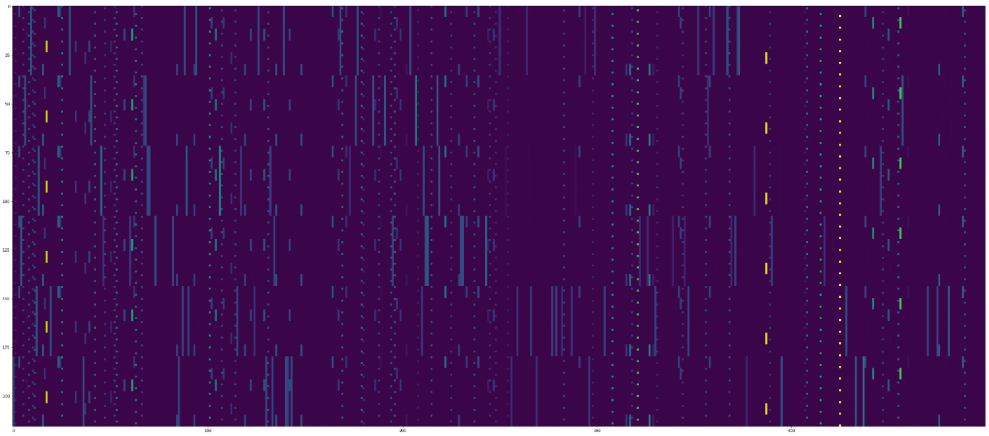

Figure 22: This matrix shows the optimised neural representation for a 6 location version of the ABCD task. Each column is a neuron, each row corresponds to a vector of angles $\boldsymbol{\theta}$. They are organised such that in the top sixth of all rows the next location to go to is 1, then in the next sixth it is 2, etc. Then the first sixth of the first sixth correspond to $\theta_2 = 1$, the next sixth of the first sixth $\theta_2 = 2$, etc. As you can see, neurons code very interpretable, each neuron fires whenever a particular location is in a particular slot of $\boldsymbol{\theta}$, just like the music box!

Wonderfully, we can really understand why this is the optimal solution, like we did for grids!

4.2.4 Discussion & Conclusion

Now, so far so good, but some things are obviously wrong. We miss some vital pieces of encoded information: space (the goal locations are just indexed, rather than being embedded in 2D space), phase (a HUGE part of the data), and the brain encodes the complete continuous path - how do you encode a continuous trajectory!?

But, these preliminary results are exciting. First it suggests that actionability can be applied around the brain to predict neural responses for regions performing computations. Second, it might help us think about other prefrontal cortical representations. Thus far, the theory hasn't provided much insight, it's just a clean encapsualtion of the ideas. It would be great if we could push the theory from predictions to post-dictions. I suspect that will require incorporating at least one additional element of structure. I have some ideas for how to do this, discussed in the next section.

Overall though, this work, towards which my contribution is truly minor, elucidates an exciting interpretation of prefrontal cortex. Not as just an elaborate memory (though you can see how the music box representation could be confused with a simple working memory model), nor just as value encoding (though you can see how the representation of goals can be correlated to a value representation). Rather, it adds neural evidence to the theory that prefrontal cortex is storing and computing with goals and how to achieve them, as long suggested (Le Merre, Ährlund-Richter, and Carlén 2021; Earl K Miller and Cohen 2001). This interpretation meshes well with other recent recordings in frontal areas. (Basu et al. 2021).

4.3 Some Actionable Actionability Ideas for Future Research

I break this into ideas related to (i) grid cells, (ii) prefrontal cortex, and (iii) broader ideas.

4.3.1 Broader Actionability Directions

Actionability Around the Brain

One big goal is show how this representational principle can be applied broadly to understand brain circuits doing computations. A third example (beyond grids and music boxes) would, I feel, go a long way to justifying this broader claim; because 3's the magic number. I've got a few ideas about where we could try that.

A) Shiftable Representations: The brain has to modulate its sensory processing based on context. There's some great examples of this, including mechanistic underpinnings, in auditory cortex (Bajo et al. 2010; King et al. 2011; Willmore and King 2023). I'm there's loads of similar findings in vision, but there's also good examples in olfaction (Koldaeva, Schaefer, and Fukunaga 2019; Kudryavitskaya et al. 2021), even flies seem to also do it (Okray et al. 2022 - though how flexibly is unclear). This could be thought of in the same way as an actionable code, context acts on the representation and modulates it.

This moves away from the internal model version of events a little, and stepping away from these rule based systems which look a lot like groups might limit how much analytics we can do. But I'd like to think about what a flexible adaptive representation should look like.

B) Other frontal cortical datasets: Unclear how much the theory is being useful here, but we could just apply the same data analysis ideas to existing frontal cortical datasets. For example, there's a dataset in the lab of monkeys navigating a large conceptual grid. Just looking for grid cells hasn't really worked, perhaps grid cells with phase anchors would?

C) Language of Thought and Generative Programs: There's a beautiful series of papers that use a programmatic language of thought to match neural representations (at the fMRI level). They pose that we represent things with the shortest programme we have that could create the objects (Al Roumi et al. 2020; Dehaene et al. 2022; K. Ellis et al. 2020; Goodfellow et al. 2014; Sablé-Meyer et al. 2022).

That's great! That's just an internal model where the rules are the commands of the program. I think, though I'd have to read to check this more, they claim the rules are encoded consistently, so you could imagine a neural representation being built in an actionable way. Perhaps we can mix these ideas to predict neural responses...

Learning Actionable Representations

I had a couple of idea to think about learning these representations:

A) Normative Actionable Learning Rules: Cengiz Pehlevan and colleagues have a series of paper where they optimise objectives using biolically plasuible learning rules (Pehlevan and Chklovskii 2015, 2019; Sengupta et al. 2018). It's really cool! A super clean mapping from learning objective to networks of bio-plausible neurons!

Is there any way to map the actionable objective to such a network? It doesn't seem impossible, they've done similar things and with grids when actionability just means fourier sparsity, it just boils down to applying a fourier transform matrix and then enforcing sparsity?

Promisingly for this direction, there are some hebbian rules that appear to generate grid cells (Widloski and I. R. Fiete 2014); though I've heard they only really work in 1D, I wonder why...?

B) Learning structured representation: More broadly, you could imagine using the actionability constraint as a learning objective - create a representation of your state and your action such that you can predict the consequences of your actions! Perhaps also do it in a generalisable way, like actionability, perhaps not.

This kind of idea has popped up quite a lot recently (Caselles-Dupré, Garcia Ortiz, and Filliat 2019; Hansen-Estruch et al. 2021; Quessard, Barrett, and Clements 2020; Saanum and Schulz 2022), in particular in the homomorphism autoencoder, that seeks to learn an abstraction of action that is literally a group representation in some representational space (Keurti et al. 2022). Perhaps it could be a good model of "what is an action?".

C) Deep Linear Networks for Tensors!: Perhaps we could study learning of this bilinear system (a bit like in Hiratani and Sompolinsky 2023), as a model of both learning what are the right abstractions to learn for internal models, but also Mohamady's music boxes as they learn? It doesn't seem like a crazy step beyond the learning dynamics that have been extracted from linear networks (A. Saxe, Sodhani, and Lewallen 2022; A. M. Saxe, McClelland, and Ganguli 2013).

Two Other Large Project Ideas

A) Automatic actionability: A masters student, Carina Hung, suggested setting up some automatic procedure to map an experiment setup to a constrained optimisation problem, and then to neural predicitions. I don't know how feasible this is, but would be cool if it worked!

B) More bio-plausible actionability: Actionability is a statement about how the brain uses modulatory signals to update its representation; it is therefore a statement about how real neurons do things. Our current version is retro-fitted to group representation theory - rather than to measured biological function. What are plausible but tractable models of neocortical recurrent circuits? Is actionability at all bio-plausible itself?

4.3.2 Primarily Music Box Related

Phase: The prefrontal cortex also encodes phase - progress towards current goal. In fact, that is the largest signal in the representaion. Why? And can we understand this in a normative representational framework?

To the first question: phase provides a scaffold of behaviour, sequences things over time. It is there before the mice understand the task. The most plausbile answer to why is phase there then, is that it is the fundamental code of progress towards goal. Only later does the the animal work out that it needs to chain four goals together, so it works with the structures it has at hand and sequences together four goal coding phase sequences.

To the second question: it becomes very difficult to see why an optimal representation, such as our actionability theory predicts, would choose the intriguing structure the brain has gone for. In the simple music box so far discussed all the rings are location anchored, meaning that one part of the ring is always active when the mouse is in a given location. In reality, however, rings are *phase-location* anchored - if the mouse reaches a location at a particular phase then one part of the ring is always active.

One way to force a representation to do this is to ensure that at all points in time you can consistently readout both goal location and current phase from the population. This is not surprising, it effectively bakes phase is, but would be interesting to show nonetheless. One subtlety with this is that each time a physical step is made the progress update equation ($\boldsymbol{W}\boldsymbol{g} = \boldsymbol{g}'$) has to scale the size of its update , i.e. the turning of music box handle, to the length of the task.

Another possibility is that the mouse uses phase to solve this handle turning problem - in early trials of one task it is working out how fast to crank the handle of the music box for a given step size and phase facilitates this - in a way that isn't quite precise.

In my mind the most likely explanation, however, is that, as stated in the first paragraph, this structure reflects the algorithm the mouse uses to learn this structure. Further thought is required to work out what this algorithm is, why it is being used, and how to crystallise it into similarly precise notions to enable downstream predictions.

Space: Currently goal locations are represented by indices, rather than true position in 2D space. This is ludicrous - of course the mouse is not only able to solve the version of this task with 9 fixed locations, it empirically can generalise to new locations, and I'm sure it could deal with targets that moved.

We must therefore somehow embed the rules of space into this representaiton, so that all goals can be encoded, not just 9 special ones. One way to do this is to make the representation actionable in space:

$$\boldsymbol{T}(\Delta\theta_0)\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{g}(\boldsymbol{\theta} + \Delta\theta_0) \tag{20}$$

This works (the optimal representation is music boxes), but implies some strange things about how the mouse thinks about the task - it suggests that it thinks of moving the goal north the same, no matter where the goal is.

Another approach is to assume that the prefrontal cortex is inheriting its notion of space from some other brain area - i.e. the entorhinal cortex. This could be done by forcing the memory vectors, $\boldsymbol{e}$ to be grid representations of position.

Continuous things: The representation seems to be much closer to encoding continuous paths for the mouse to follow. This requires combining the two previous points: thinking about a representation that embeds both phase and continuous space. This sounds difficult, but also sounds like a real problem that theory can help with - so worth pursuing.

4.3.3 Primarily Grid Cell Related

There's some (really many, but here's two) experiments it would be cool to get run, and compare to:

- How do grid cells represent curvy 2D space? All approaches to this question (like getting mice to climb a slope - Hayman et al. 2015) have so far studied a space in which the rules of path integration are unchanged. I would LOVE to see how grid cells respond if the rules of path integration are changed - for example by recording them on a pyramid.

  I tried to persuade people to do this, but its been going pretty slowly recently...

My ideal prediction would be that each module of grids squashes space into 2D in a different way. This enable path integration, but also ensures that the mouse doesn't code areas of space badly.

- The theory predicts that grids should change coding in different rooms. In fact, it matches an experiment in which exactly this happened. It would be cool to get chatting to some experimentalists who would measure some grid cells in one room, we'd then be like "change the room like this and I think only this module will change", and then see what happens.

  If I'm serious about this I should write a proposal with some example experiments going on.

There are a number of specific predictions of aspects of the grid cell code that would be fun to work out, or study numerically:

- Predict the number of modules vs sigma, the separation lengthscale in neural space.
- How is does the optimal angle between modules change with module lengthscale? What is the optima of both of these things at the same time?
- What is the optimal shape of an individual grid bump? Spherical, or should it be slightly elliptical?
- What are the optimal frequencies in a hexagonal room? Does that match recordings of grid cells in those rooms?
- How do optimal module relationships vary in different rooms? Does this match the recordings people have made?
- How should the number of grid cells vary by module?

Then there's some particular experiments that we could try and match:

- Matthias Horan in the SWC is measuring entorhinal representations from animals in VR. They explore worlds with teleporters, how would our predictions change there? Are they at all reasonable?
- Misha Ahrens has recordings of a brainstem integrator of positional movements in larval zebrafish (E. Yang et al. 2022). It would be cool to see if we can match his findings. What is different in the case of the fish?
- 3D grid cells do not match our theory at all. Perhaps we they can be fit in the same framework if we introduce an error threshold to actionability that is much higher in the z than x and y directions? What does the functional form of the code look like in those cases?
- Krupic et al. 2015 and others show that grids bend in particular environments, can this be understood as the mouse just getting confused about its position?

Finally there's some pure theory improving steps that could be taken:

- Why do some works appear to get grids without positivty, despite obeying so many of our assumptions?! (namely: R. Gao, J. Xie, Wei, et al. 2021; R. Gao, J. Xie, Zhu, et al. 2018; J. C. Whittington, Muller, et al. 2020) Could it be due to the discretisation of space into a grid?
- Can we exactly solve the optimisation problem in any cases?
- Could we study the representation of curvy 2D space using fibre bundles? This has been started by a masters thesis (Lutz 2021), but would be good to use to make predictions.
- What is the complete representation theory of 2D space? Currently we have to use tori, and then take size to infinity; it would be cleaner to do without this step.
- Does uncertainty need to be encoded? If so how? (Kang, D. M. Wolpert, and Lengyel 2023) Would that fit the warping grids data while still being actionable?
- Could we fix the frequencies, solve the convex optimisation problem, then take gradients through the initial settings of the frequencies?

5 Disentangling and Modularising with Biological Constraints and Compositional Inductive Biases

The world around us is modular, and so is the way we think about it. Our brain is shockingly modularised, yet at other times we see motor information appearing all around the brain. Why? What principles drive the push to sometimes be very modular, at other times all mixed up?

As shown already in the grid cell work, the representational constraints we've been playing with cause modularity, specifically they said modules of grid cells were optimal. James extended this marvellously to show that the biological constraints cause disentangling - the separation of the encoding of different meaningful factors of variation in data - under an appropriate definition of meaningful. This matches recent findings that RNNs modularise their activity, different sets of neurons encode different functional subparts of a large set of tasks, but only when the neural network uses a ReLU - i.e. only with the biological constraint of positivity! As such, positivity, energy efficiency and functional performance seem to together be enough to make predictions about when things should and shouldn't modularise.

So, there are three things in this section. The first describes the finding that biological constraints disentangle independent factors in linear data; that this also seems to work in nonlinear settings; and that this can explain neural data showing that in some situations grid cells code only for space (disentangled), at other times they mix up space with other things. The second section describes a machine learning project that uses a compositional latent space to improve disentangling performance. The third describes our ongoing work to broaden our understanding of biological disentangling and use it to predict modularisation in aritifical and biological networks. We unify the finding of grid modules and disentangling; use this unification to predict modularisation in linear RNNs; predict phase transitions as variables become more or less compositional; and make plans for future work.

5.1 Disentangling with Biological Constraints, A Theory of Functional Cell Types

ABSTRACT

Neurons in the brain are often finely tuned for specific task variables. Moreover, such disentangled representations are highly sought after in machine learning. Here we mathematically prove that simple biological constraints on neurons, namely nonnegativity and energy efficiency in both activity and weights, promote such sought after disentangled representations by enforcing neurons to become selective for single factors of task variation. We demonstrate these constraints lead to disentanglement in a variety of tasks and architectures, including variational autoencoders. We also use this theory to explain why the brain partitions its cells into distinct cell types such as grid and object-vector cells, and also explain when the brain instead entangles representations in response to entangled task factors. Overall, this work provides a mathematical understanding of why single neurons in the brain often represent single human-interpretable factors, and steps towards an understanding task structure shapes the structure of brain representation.

5.1.1 Introduction

Understanding why and how neurons behave is now foundational for both machine learning and neuroscience. Such understanding can lead to better, more interpretable artificial neural networks, as well as provide insights into how biological networks mediate cognition. A key to both these pursuits lies in understanding how neurons can best structure their firing patterns to solve tasks.

Neuroscientists have some understanding of how task demands affect both early single neuron responses McIntosh et al. 2016; Ocko et al. 2018; Olshausen and Field 1996; Yamins et al. 2014 and population level measures such as dimensionality Stringer et al. 2019. However, there is little understanding of neural population structure in higher brain areas. As an example, we do not even understand why many different bespoke cellular responses exist for physical space, such as grid cells Hafting et al. 2005, object-vector cells Høydal et al. 2019, border vector cells Lever et al. 2009; Solstad et al. 2008, band cells Krupic et al. 2015, or many other cells Deshmukh and Knierim 2013; Gauthier and Tank 2018; O'Keefe and Dostrovsky 1971; Sarel et al. 2017. Each cell has a well defined, specific cellular response pattern to space, objects, *or* borders, as opposed to a mixed response to space, objects, *and* borders. Similarly, we don't understand why neurons in inferior temporal cortex are aligned to axes of data generative factors Bao et al. 2020; Chang and Tsao 2017; Higgins, Chang, et al. 2021, why visual cortical neurons are de-correlated Ecker et al. 2010, why neurons in parietal cortex are selective only for specific tasks Lee et al. 2022, why prefrontal neurons are apparently mixed-selective Rigotti et al. 2013, and why grid cells sometimes warp towards rewarded locations Boccara et al. 2019 and sometimes don't Butler, Hardcastle, and Giocomo 2019. In essence, why are some neural representations entangled and others not?

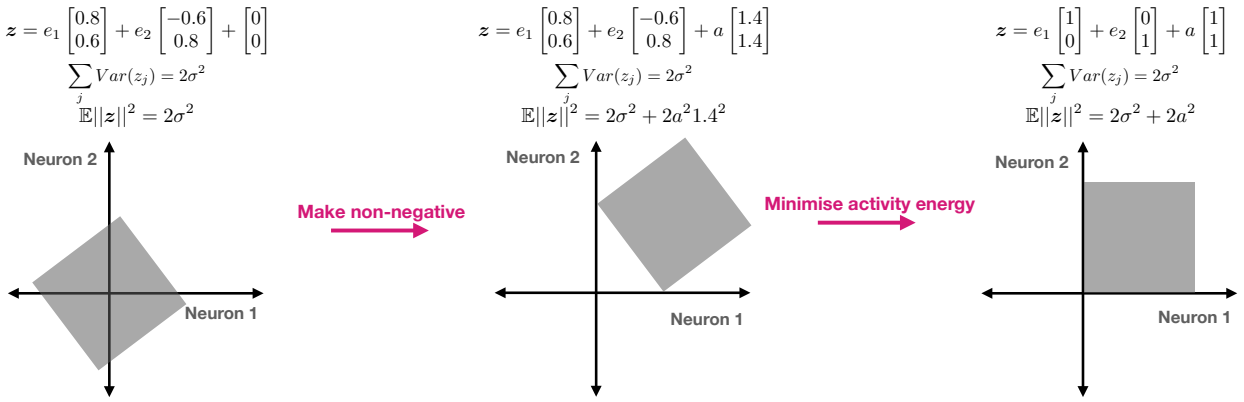

Figure 23: **Proof intuition.** Two uniformly distributed independent factors represented with two entangled neurons (left). The representation can be made nonnegative at the expense of activity energy (middle). Activity energy is minimised under a nonnegativity (and variance) constraint when the neurons are axis aligned to task factors (i.e. disentangled, right). Grey boxes denote uniform distributions over neural activity induced by uniform distributions over task factors. Note our proof does not require uniformity.

Machine learning has long endeavoured to build models that disentangle factors of variation Bengio, Courville, and Vincent 2013; Higgins, Matthey, et al. 2017; Geoffrey E. Hinton, Krizhevsky, and S. D. Wang 2011; Locatello et al. 2019. We define disentanglement as single neurons responding to single factors of variation (see Appendix of paper for further details). Such disentangled factors can facilitate compositional generalisation and reasoning Higgins, Pal, et al. 2017; Higgins, Sonnerat, et al. 2018; J. C. R. Whittington, Kabra, et al. 2021 (though some work has challenged the idea that disentangled representations generalise better Schott et al. 2022, as well as lead to more interpretable outcomes in which individual neurons represent meaningful quantities. Unfortunately, building models that disentangle is challenging Locatello et al. 2019.

In this work we 1) prove simple biological constraints of **nonnegativity** and **minimising activity energy** lead to factorised representations in linear networks; 2) empirically show these constraints lead to disentangled representations in both linear and nonlinear networks; 3) obtain competitive disentanglement scores on a standard disentanglement benchmark; 4) provide an understanding why neurons in the brain are characterised into specific cell types due to these same biological constraints; 5) empirically show these constraints lead to specific cell types; 6) suggest when and why neurons in the brain exhibit disentanglement versus mixed-selectivity.

Please see appendix of paper for a comprehensive discussion relating our work to existing literature.

5.1.2 Linear disentanglement with biological constraints

We first provide a theorem that suggests why the combined biological constraints of nonnegativity and energy efficiency lead to neural disentanglement (proofs of all theorems are in Appendix of paper):

Theorem 1. Let $\boldsymbol{e} \in \mathbb{R}^k$ be a random vector whose k independent components denote k task factors. We assume each independent task factor e_i is drawn from a distribution [2]that has mean 0, variance σ^2, and maximum and minimum values of $\min(e_i) = -a$ and $\max(e_i) = a$. Also let $\boldsymbol{z} \in \mathbb{R}^n$ be a linear neural representation of the task factors given by

$$\boldsymbol{z} = \boldsymbol{M}\boldsymbol{e} + \boldsymbol{b}_z,$$

where $\boldsymbol{M} \in \mathbb{R}^{n \times k}$ are mixing weights and $\boldsymbol{b}_z \in \mathbb{R}^n$ is a bias. We further assume two constraints: (1) the neural representation is *nonnegative* with $z_i \geq 0$ for all $i = 1, \ldots, n$, and (2) the neural population variance is a nonzero constant, $\sum_j Var(z_j) = C$, so that the neural representation retains some information about the task variables. Under these two constraints we show that in the space of all possible neural representations (parameterised by $\boldsymbol{M}$ and $\boldsymbol{b}_z$), the representations that achieve minimal activity energy $\mathbb{E}||\boldsymbol{z}||^2$ also exhibit disentanglement, by which we mean every neuron z_j is selective for at most one task parameter: i.e. $|M_{jk}||M_{jl}| = 0$ for $k \neq l$, a.k.a. each row of $\boldsymbol{M}$ has at most 1 non-zero entry. (Proof in Appendix of paper).

[2]We understand that task factors, or indeed neurons in the brain, are generally not i.i.d., but we have made this assumption for mathematical convenience.

Intuition. The intuition underlying the proof of the theorem is shown in Fig.23, where the key idea can be seen with two neurons encoding two factors. In particular, the bias must make every z_i nonnegative for all values of e_1 and e_2. But since e_1 and e_2 are independent, the minimum firing of neuron 1 for example obeys $\min(z_1) = \min(0.8e_1 - 0.6e_2) = \min(0.8e_1) + \min(-0.6e_2) = -a(0.8 + 0.6)$. Thus for neurons that mix factors, a larger bias term must be used to ensure nonnegativity, which leads to increased expected energy. Minimising this energy (subject to a constant variance) requires the smallest possible bias for each neuron, which occurs when each neuron is selective for a single task factor. We note that this is consistent with neurons having a baseline firing rate where now activity below baseline corresponds to negative/positive values in the distribution, and activity above baseline corresponds to positive/negative values in the distribution.

The above theorem, while simple, is restricted in two ways: (1) the independent task factors e_i are *directly* available to the network; (2) representational collapse (i.e. setting $\boldsymbol{z} = 0$) under energy minimisation is prevented solely by a variance constraint. We thus consider a more general setting where a neural circuit receives not the independent task factor vector $\boldsymbol{e}$, but instead receives the mixed combination $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{e}$, where $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{D} \in \mathbb{R}^{m \times k}$ and $m \geq n \geq k$. We further model the neural representation $\boldsymbol{z}$ as a linear generative model that can predict observed data $\boldsymbol{x}$ via $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}_x$. Thus prediction, not variance constraint, now prevents collapsing neural representations (proof in Appendix of paper). Furthermore we also prove the following:

Theorem 2. Let $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{e}$ be observed entangled data, where the independent task factor vector $\boldsymbol{e}$ obeys the same distributional assumptions as in Theorem 1. Let a neural representation $\boldsymbol{z}$ exactly predict observed data via $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}_x$ with zero error. Then for all such data generation models (with parameters $\boldsymbol{D}$) and all such neural representations (with parameters $\boldsymbol{W}$ and $\boldsymbol{b}_x$), as long as: (1) the columns of $\boldsymbol{D}$ are (scaled) orthonormal; (2) the norm of the read-out weights $||\boldsymbol{W}||_F^2$ is finite; (3) the neural representation is nonnegative (i.e. $\boldsymbol{z} > 0$), then out of all such neural representations, the minimum energy representations are also disentangled ones. By this we mean that each neuron z_i will be selective for at most one hidden task factor e_j.

We note that $\boldsymbol{D}$ having (scaled) orthonormal columns may seem like a strong constraint, but it holds approximately for any random matrix $\boldsymbol{D}$ with many observations (dimensionality of $\boldsymbol{x}$) and few independent task factors (dimensionality of $\boldsymbol{e}$) (proof in Appendix of paper). The appendix also discusses and provides intuition for when disentanglement occurs as $\boldsymbol{D}$ takes more general forms.

Strikingly, the essential content of Theorem 2 is that any linear, nonnegative, optimally energetically efficient, generative neural representation that accurately predicts entangled observations that are linear mixtures of hidden task factors, will possess single neurons that are selective for individual task factors, despite *never* having *direct* access to them. In terms of applications, this theorem could apply in supervised or self-supervised machine learning settings in any neural network layer $\boldsymbol{z}$ that is linearly read-out from, or in neuroscience settings where $\boldsymbol{z}$ reflects a neural population from which one attempts to linearly decode task factors. While Theorem 2 holds in a simple setting, we will show through simulations that its essential content, namely that nonnegativity and energy efficiency together promote disentanglement, holds in practice in much more complex multilayer neural networks.

5.1.3 Disentanglement in machines

We now present simulation results demonstrating that nonnegativity and energy efficiency (minimising either activity or weight energy) lead to single neuron selectivity for single task factors. We show this for supervised and unsupervised learning, both for linear and nonlinear tasks and networks (details of datasets, models, and simulations in Appendix of paper.

A measure for disentangled subspaces. While our theory describes when single neurons become selective for single independent task factors, it does not limit the number of neurons selective for any given factor. For example in Theorem 2, four copies of the same neuron in $\boldsymbol{z}$, each with half the activity, along with four copies of projecting weights each with half the values, predicts $\boldsymbol{x}$ just as well and has exactly the same energy in both $\boldsymbol{z}$ and $\boldsymbol{W}$ [3]. More interestingly, an underlying task factor may not be one-dimensional, e.g. spatial location, in which case the subspace that codes for this factor have at least the same dimension. This phenomena cannot be captured by many metrics of disentanglement (e.g. the popular mutual information gap; MIG; R. T. Chen et al. 2018) since they score highly if each factor is represented in just *one* neuron. Thus we define a new metric (mutual information ratio; MIR) that instead scores highly if each neuron only cares about one factor (see Appendix of paper for details).

Regularizers as constraints. We impose nonnegativity via a ReLU activation function, or softly via explicit regularization $\mathcal{L}_{\text{nonneg}} = \beta_{\text{nonneg}} \sum_i \max(-a_i, 0)$ where i indexes a neuron in the network, and β_{nonneg} determines the regularization strength. Similarly, we apply regularization to the activity energy and weight energy; $\mathcal{L}_{\text{activity}} = \beta_{\text{activity}} \sum_l ||\boldsymbol{a}_l||^2$

[3] Learning dynamics may favour fewer neurons per factor as there are fewer weights to align.

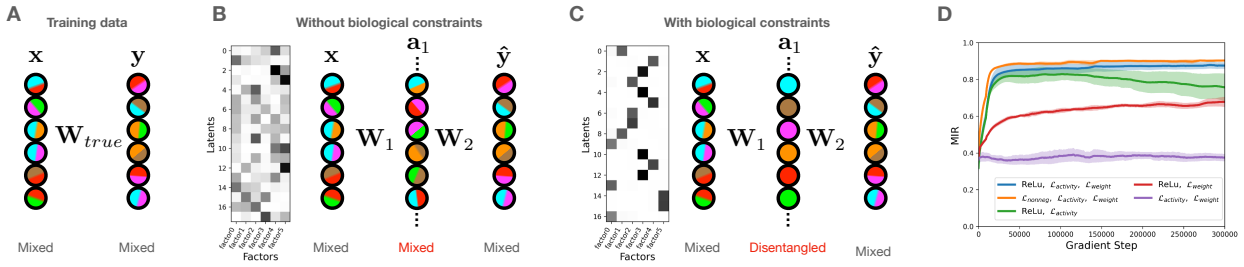

Figure 24: **Shallow linear networks disentangle.** We train 1-hidden layer linear networks on linear data. **A)** Cartoon schematic showing both input and output are entangled linear mixtures of factors (colours). Neurons colours schematically denote which of the factors it codes for. $\boldsymbol{W}_{true}$ is the true mapping between $\boldsymbol{x}$ and $\boldsymbol{y}$. **B)** A model without biological constraints learns entangled internal representations. Mutual information matrix (scale 0 to 0.6) shown on left, cartoon schematic on right. $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the learnable weights projecting to and from the hidden layer. **C)** A model with our constraints learns disentangled representations (MI matrix scale 0 to 2.25). **D)** Several model variants, in which only those with all our constraints learn disentangled representations (definition of metric MIR in Appendix of paper). Average and standard error shown for 5 random seeds.

and $\mathcal{L}_{\text{weight}} = \beta_{\text{weight}} \sum_l ||\boldsymbol{W}_l||^2$. The role of $\mathcal{L}_{\text{weight}}$ is to promote activity (variance) in the network, otherwise activity could be reduced via $\mathcal{L}_{\text{activity}}$, and such reduced activity could be compensated for with arbitrarily large weights. The total loss we optimise is

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{nonneg}} + \mathcal{L}_{\text{activity}} + \mathcal{L}_{\text{weight}}}_{\text{Biological constraints}} + \underbrace{\mathcal{L}_{\text{prediction}}}_{\text{Functional constraints}} .$$

Here 'functional constraints', are any prediction losses the network has i.e. error in predicting target labels in supervised learning, or reconstruction error in autoencoders.

Disentanglement in supervised shallow neural networks. First we consider a dataset $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\}$, where $\boldsymbol{x}$ is a orthogonal mixture of six i.i.d. random variables (hidden independent task factors; uniform distribution), and $\boldsymbol{y}$ is a linear transform of $\boldsymbol{x}$ (dimension 6; Fig. 24A). First we train shallow linear networks to read-out $\boldsymbol{y}$ from $\boldsymbol{x}$. Networks without biological constraints exhibit mixed internal representations (Fig. 24B). However, with our constraints, networks learn distinct sub-networks for each task factor (Fig. 24C). Removing any one of our constraints leads to entangled representations (Fig. 24D). Lastly we note sparsity constraints do not induce disentanglement (Appendix of paper). Thus the disentanglement effect of ReLUs is not from sparsity, but instead from nonnegativity.

Disentanglement in supervised deep neural networks. Training deep nonlinear (ReLU) networks on this data also leads to distinct sub-networks, with all layers learning disentangled representations (Fig. 25A). However with nonlinear data ($\boldsymbol{x} \leftarrow \boldsymbol{x}^3$, $\boldsymbol{y}$ remaining the same), the early layers are mixed-selective, whereas the later layers are disentangled (Fig. 25B-C). Understanding why the final hidden layer disentangles is easy, since it linearly projects to the target and so our theory directly applies. By extrapolating our theory, we conjecture that our biological constraints encourage any layer to be as linearly related to task factors and as disentangled as possible. However, early layers cannot be linear in hidden task factors since they are required to perform nonlinear computations on the nonlinear data, and thus only once activity becomes linearly related to independent task factors in later layers does disentanglement set in (as predicted by our linear theory).

Disentanglement in unsupervised neural networks. We now consider unsupervised learning, i.e. $\mathcal{D} = \{\boldsymbol{x}\}$, where $\boldsymbol{x}$ is a linear mixture of multiple independent task factors as in Theorem 2. Training 0-hidden layer autoencoders on this data, with our biological constraints, recovers the independent task factors in individual neural subspaces (Fig. 26A/B). Moreover, this only occurs when all constraints are present (Fig. 26A). Again, even though our theory applies to the linear setting, the same phenomena occur when training deep nonlinear autoencoders on nonlinear data; i.e. when $\boldsymbol{x}$ is a *nonlinear* mixture of multiple i.i.d. random variables, i.e. $\mathcal{D} = \{f(\boldsymbol{x})\}$ (Fig. 26C-D).

Disentanglement on a standard benchmark with VAEs. We now consider a standard disentanglement dataset (Fig. 27A; H. Kim and Mnih 2018. To be consistent with, and to compare to, the disentanglement literature we use a VAE and measure disentanglement with the familiar mutual-information gap (MIG) metric R. T. Chen et al. 2018. For nonnegativity we ask the mean of the posterior to be nonnegative (via a ReLU) [4], but we *do not* add a norm constraint as the VAE loss already includes one in its KL term between the Gaussian posterior and the Gaussian prior.

[4]Using a nonnegative posterior mean is odd when the prior is Gaussian, but it allows for easier comparison.

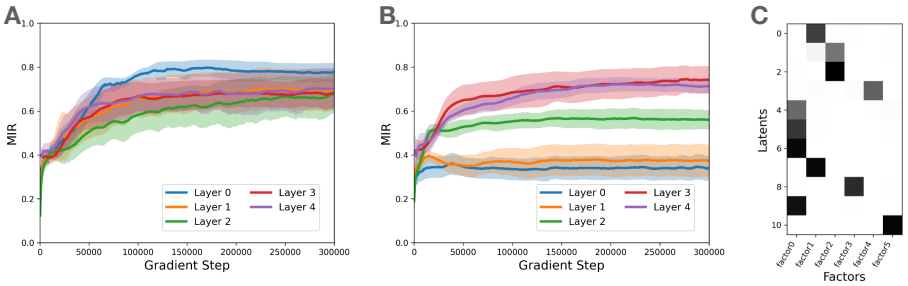

Figure 25: **Deep nonlinear networks disentangle.** We train 5-hidden layer nonlinear networks with our constraints on linear and nonlinear data. **A)** For *linear* data, all layers in the network learn a disentangled representation. **B)** For *nonlinear* data, only later layers learn a disentangled representations. **C)** Example mutual information matrix from the penultimate hidden layer.

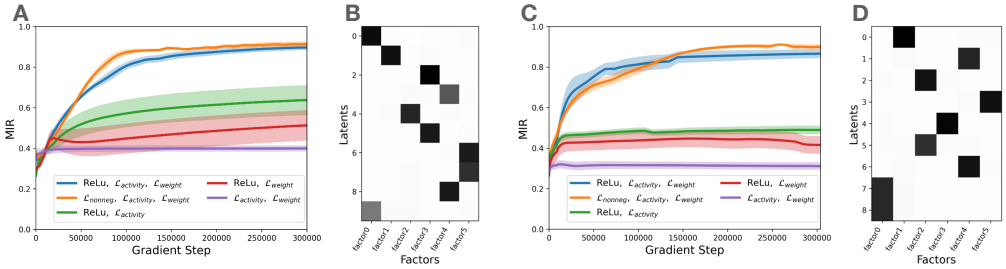

Figure 26: **Learning data generative factors with autoencoders. A)** Training linear autoencoders on linear data. Only models with our constraints learn disentangled representations. **B)** Example mutual information matrix from a high MIR model. **C)** Nonlinear autoencoders trained on nonlinear data. Only models with our constraints learn disentangled representations. **D)** Example mutual information matrix from a high MIR model. All learning curves show mean and standard error from 4 mean from 5 random seeds.

While state-of-the-art results are not our aim (instead we wish to elucidate that simple biological constraints lead to disentanglement), our disentanglement results (Fig. 27B) are competitive and often better than those in the literature (comparing to results of many models shown in Locatello et al. 2019) even though those models explicitly ask for a factorised aggregate posterior. The particular baseline model we show here is β-VAE (Fig. 27D). We see that (1) our constraints lead to disentanglement (Fig. 27B-C); (2) including a ReLU improves β-VAE disentanglement (as predicted by nonnegativity arguments above, Fig. 27D); and (3) our constraints give results in the Goldilocks region of high disentanglement and high reconstruction accuracy (Fig. 27E).

5.1.4 Disentanglement in brains: A theory of cell types

We next turn our attention to neuroscience, which is indeed the inspiration for our biological constraints. While we hope our general theory of neural representations will be useful for explaining representations across tasks and brain areas, for reasons stated below, we choose our first example from spatial processing in the hippocampal formation. We show our biological constraints lead to separate neural populations (modules) coding for separate task variables, but **only** when task variables correspond to independent factors of variation. Importantly, the modules consist of distinct functional cell types with similar firing properties, resembling grid Hafting et al. 2005 and object-vector cells Høydal et al. 2019 (GCs and OVCs).

We choose to focus on spatial representations for two reasons. Firstly, there is a significant puzzle about why neurons deep in the brain, synaptically far from the sensorimotor periphery, almost miraculously develop single cell representations for human-interpretable factors (e.g. GCs for location in space; Fig. 28A, and OVCs for relative location to objects Fig. 28B). Such observations are not easily accounted for by standard neural network accounts that argue that representations are unlikely to be human-interpretable Richards et al. 2019. Secondly, whilst these bespoke spatial representations are commonly observed to factorise into single cells, there are situations in which selectivity spans across multiple task variables Boccara et al. 2019; Hardcastle et al. 2017. For example, sometimes spatial firing patterns of GCs are warped by reward Boccara et al. 2019 and sometimes they are not Butler, Hardcastle, and Giocomo 2019. There is no theory for explaining why and when this happens.

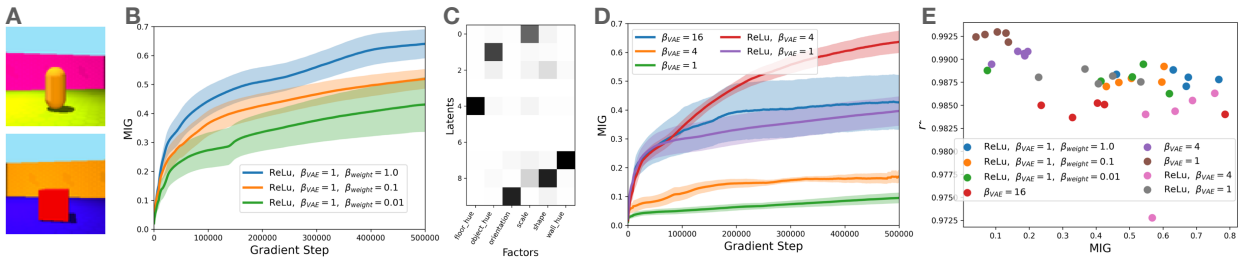

Figure 27: **Learning data generative factors with variational autoencoders. A)** We train on the Shapes3D dataset, with two example images shown. These images have 6 underlying factors. **B)** MIG scores are higher with higher weight regularization, and generally higher than any β-VAE (panel D). β_{weight} is the regularisation strength of the weight regularisation . **C)** Mutual information matrix for a high scoring model. **D)** β-VAE MIG scores. Adding a ReLU improves MIG scores. **E)** MIG score against R^2 shows models with our constraints lie in the Goldilocks region of high disentanglement and high reconstruction. All learning curves show mean and standard error from 5 random seeds. Results from an additional dataset are in the main paper

A factorised task for rodents. We consider a task in which rodents must know where they are in space, but must also approach one of multiple objects. If objects appear in different places in different contexts, the task is factorised into independent factors (Fig. 28C): 'Where am I in allocentric spatial coordinates?' and 'Where am I in object-centric coordinates?'. By contrast, if objects always appear in the same locations, the task is not factorised (as spatial location can predict object location). Formally, our task requires predicting spatial location, $\boldsymbol{x}$, whether an object is observed, $\boldsymbol{o}$, and the optimal action, $\boldsymbol{a}$. If objects move between tasks, then $p(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{o}) = p(\boldsymbol{x})p(\boldsymbol{a}, \boldsymbol{o})$, where $\boldsymbol{o}$ and $\boldsymbol{a}$ are not factored since optimal actions are dependent on objects (see Appendix of paper for details). Our theory says the representation will have two subspaces - one for allocentric location (for predicting $\boldsymbol{x}$) and one for location relative to objects (for predicting $\boldsymbol{o}$ and $\boldsymbol{a}$) - and that these sub-spaces should be represented in separate neural populations when biological constraints are present.

The representation in rodent brains indeed has two distinct modules of non-overlapping cell populations: (1) GCs Hafting et al. 2005 which represent allocentric space via hexagonal firing patterns (Fig. 28A); and (2) OVCs Høydal et al. 2019 which represent relative location to objects through firing fields at specific relative distances and orientations (Fig. 28B).

Model with additional structural constraint. Predicting allocentric spatial locations from egocentric self-motion cues is known as path integration Burak and I. R. Fiete 2009, and is believed to be a fundamental function of entorhinal cortex (where GCs and OVCs are found). GCs naturally emerge from training RNNs to path integrate under several additional biological constraints Banino et al. 2018; Cueva and Wei 2018; Sorscher, G. Mel, et al. 2019; Sorscher, G. C. Mel, et al. 2023. Hence to model this task (with locations *and* objects) we could train an RNN, $\boldsymbol{z}$, that predicts (1) what the spatial location, $\boldsymbol{x}$, will be and (2) whether we will encounter an object, after an action, $\boldsymbol{a}$, from the current location, and (3) what the expected action, $\boldsymbol{a}$, will be. However, here we adopt a far more general framework that does not limit future applications of our approach simply to sequential integration problems.

In particular, it was recently shown R. Gao, J. Xie, Wei, et al. 2021 that path integration constraints can be applied directly on the representation by adding a new constraint in the loss imposing

$$\boldsymbol{z}(\boldsymbol{x}) = f(\boldsymbol{W}_{\boldsymbol{a}}\boldsymbol{z}(\boldsymbol{x} - \boldsymbol{a})).$$

Here $f(\cdot)$ is an activation function and $\boldsymbol{W}_{\boldsymbol{a}}$ is a weight matrix that depends on the action $\boldsymbol{a}$. This surrogate constraint imposes potential path integration by ensuring that a motion $\boldsymbol{a}$ in space $\boldsymbol{x}$ imposes a lawful change in neural representation $\boldsymbol{z}$, thereby transforming the sequential path integration problem into the problem of directly estimating neural representations of space (see Appendix of paper for more details). Thus we minimise:

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{nonneg}} + \mathcal{L}_{\text{activity}} + \mathcal{L}_{\text{weight}}}_{\text{Biological constraints}} + \underbrace{\mathcal{L}_{\text{location}} + \mathcal{L}_{\text{actions}} + \mathcal{L}_{\text{objects}}}_{\text{Functional constraints}} + \underbrace{\mathcal{L}_{\text{path integration}}}_{\text{Structural constraints}}$$

These are the same biological constraints as above, but now the functional constraints involve predicting location, object, and action, and an additional structural constraint imposes equation 5.1.4. Interestingly, the structural constraint leads to a pattern forming optimisation dynamics (see Appendix of paper for mathematical details).

Modules of distinct cell types when tasks are factorised and representations are nonnegative. Just as our theory predicts, when training on tasks where objects and space are factorised (i.e. objects can be anywhere in space), under

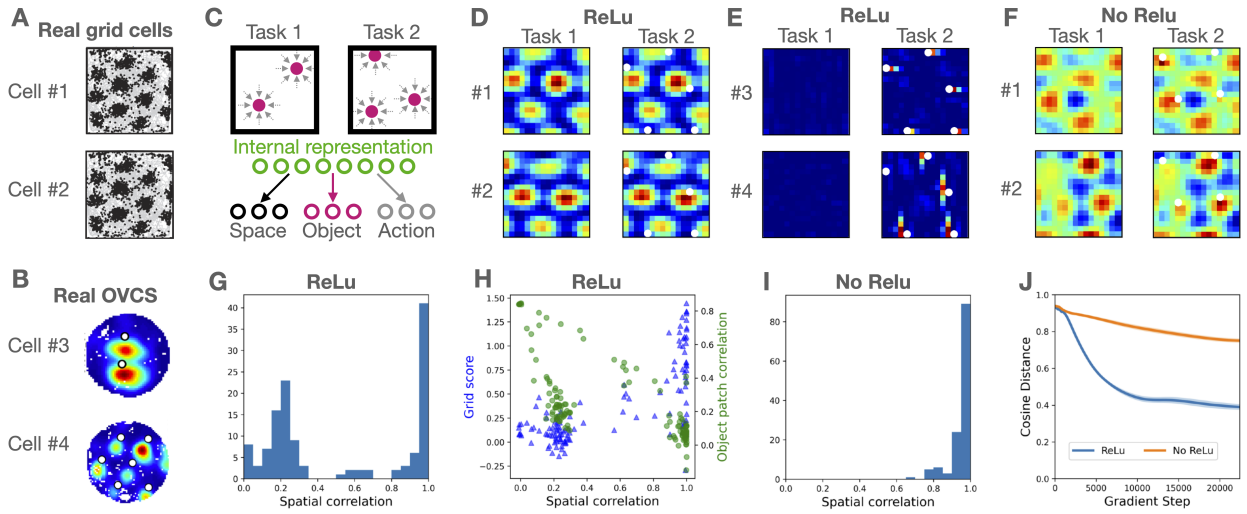

Figure 28: **Modules of distinct cell types form with nonnegativity and factorised tasks. A)** When rodents navigate environments with objects, GCs encode location in physical space with firing fields lying on a hexagonal lattice, while **B)** OVCs encode relative location to objects with firing fields at specific distances and orientations from (white) objects. These plots are ratemaps; the average firing of a given cell at every location. **C)** Top: To model these cells we use a task environment where objects move location in different contexts - space and objects are factorised. Bottom: We train a representation to predict 1) spatial location, 2) object location, and 3) correct action at every location. **D-E)** Model ratemaps when trained *with* a ReLU activation function. Task 1 contains no objects, while task 2 has several objects (white dots). We see two types of cell representation: **D)** GCs that do not change across tasks and **E)** OVCs that only appear in the presence of objects. **F)** Model ratemaps when trained *without* ReLu activation. All representations are multi-peaked but amorphous. Further cells representations shown in appendix of paper. **G)** To quantify modules in the ReLu model, we compute the distribution of individual cell spatial correlations across different tasks. **H)** The mode with high spatial correlation are cells that have high grid-score C. Barry et al. 2012 and are grid cells (GCs), the mode with low spatial correlation are cells that respond similarly around objects (OVCs). **I)** Only one mode of cells is seen without a ReLu activation. **J)** To quantify module-ness over many random seeds, we compute the cosine distance between the population's contribution to $\boldsymbol{x}$ and $\boldsymbol{o}$. This is done by taking the absolute value of weights projecting from $\boldsymbol{z}$ to $\boldsymbol{x}$ and $\boldsymbol{o}$, then summing over the space/object dimension (to obtain a vector the same dimension as $\boldsymbol{z}$), then computing the cosine distance. We see low cosine distance for the ReLu indicating different cells code for space vs objects - i.e. modules.

our biological constraints of nonnegativity and energy efficiency, distinct neural modules emerge, each selective for a single task factor (Fig. 28D-E). We see GC-like neurons that consistently represent space independent of object locations, and OVC-like neurons that recenter their representations around the moving objects or are inactive if no objects are present (further cells are shown in the appendix of the paper). Whereas without the nonnegativity constraint, all cells look qualitatively similar - a single module of multi-peaked amorphous cells (Fig. 28F). To quantify whether a population really has two distinct cell types (two modules) we analyse the consistency of a cell's representation by taking its average spatial correlation between many different object configurations (Fig. 28G-I). In the ReLu case, there are two modes (Fig. 28G-I) showing a double dissociation: the mode containing cells that don't change have high grid-score C. Barry et al. 2012 and do not have consistent activity around objects. These are GCs. Whereas the mode containing cells that change between tasks have low grid-score and respond consistently around objects (Fig. 28H). These cells respond to objects and are comparable to OVCs.

Grid cell warping and mixed-selectivity when tasks are entangled. Experimental results show GCs sometimes warp their firing fields towards rewarded locations Boccara et al. 2019 and sometimes don't Butler, Hardcastle, and Giocomo 2019. Intriguingly, in the warping situation, the rodents exhibited stereotyped behaviour; sequentially running between rewards using the same spatial trajectories rather than freely exploring the space (i.e. behaviour is entangled with space). We now explain these neuroscience observations as a consequence of space becoming entangled with objects/rewards.

Modelling factorised versus entangled tasks (objects changing locations versus always staying in the same locations), produces very different GC behaviours. In the factorised case, grid fields are unrelated to objects (Fig. 29A), whereas in the entangled task grid fields warp to objects (Fig. 29B). We quantify this by measuring the average spatial correlation of patches around each object. Only when the task is entangled are fields consistently warped towards objects (Fig.

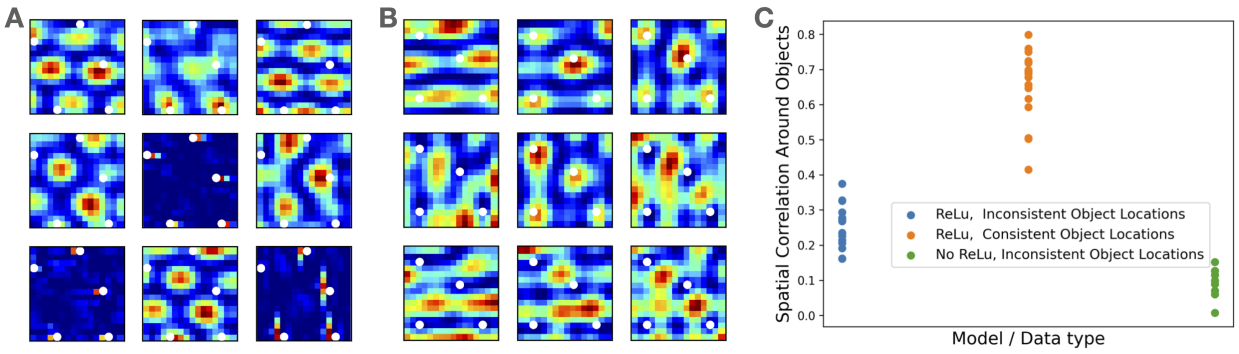

Figure 29: **Entangled tasks lead to entangled representations and grid cell warping.** We show a representative selection of cells from a model with **A)** a factorised task (as in Fig. 28) and **B)** an entangled task. The phases of the firing fields in the entangled task are locked to object location - they have warped their firing fields. This is not the case for the factorised task (asides from object specific cells). **C)** To quantify phase locking for each of the task/model variant, we compute the average spatial correlation of patches around objects. Only the entangled task shows high correlation, i.e. the cell representations have warped around objects. Each point is a model trained from a random seed.

29C). Thus we have an explanation for GC warping; they warp when space can no longer be disentangled from other factors (e.g. objects or behaviour) in behavioural tasks.

5.1.5 Discussion

We have proven that simple biological constraints like nonnegativity and energy efficiency lead to disentanglement, and empirically verified this in machine learning and neuroscience tasks, leading to a new understanding of functional cell types. We now consider some more neuroscience implications.

Representing categories. Our theory additionally says that representations of individual categories should be encoded in separate neural populations (disentangled; see Appendix of paper [5].). This provides a potential explanation for "grandmother cells" that selectively represent specific concepts or categories Quiroga et al. 2005, and have long puzzled proponents of distributed representations It further potentially explains situations where animals have been trained on multiple tasks, and different neurons are found to engage in each task Wael F Asaad, Rainer, and Earl K Miller 2000; Flesch et al. 2022; Lee et al. 2022; Rainer, Wael F. Asaad, and Earl K. Miller 1998; Roy et al. 2010. We note that while our theory says you need at least one neuron per concept that is read-out (true even without disentanglement), it does not say you need a neuron for every possible concept - only for ones that are explicitly read-out.

When to disentangle? Our theory speaks to situations in which brains or networks must generalise to new combinations of learnt factors. In this situation, if the input-output (or input-latent-output) transformations are linear, biological constraints will cause complete disentanglement of the networks. When the mappings are nonlinear, we show empirically that mixed-selectivity exists, but gradually de-mixes as layers approach the output (in supervised), or latent (in unsupervised), layers.

Optimising for low firing contrasts with previous ideas which instead optimise for linear read-out. This latter situation is akin to kernel regression, where mixed-selectivity through random expansion increases the dimensionality of neural representations, allowing simple linear read-outs in the high dimensional space to perform arbitrary nonlinear operations on the original low dimensional space.

Mixed-selectivity. Mixed-selectivity exists in the brain. For example, Kenyon cells in the Drosophila mushroom body increase the dimensionality of their inputs by an order of magnitude by close to random projections Aso et al. 2014. This may allow linear read-out to behaviour via simple dopamine gating. Similarly rodent hippocampal cells encode conjunctions of spatial and sensory variables to allow rapid formation of new memories Komorowski, Manns, and Eichenbaum 2009]. More recently it has been suggested that PFC neurons have this same property, for the same reason Rigotti et al. 2013. However, it is less clear that this is a general property of representations in associative cortex (including PFC), which can separate into different neuronal representations of different interpretable factors Bernardi et al. 2020; Hirokawa et al. 2019 or tasks Flesch et al. 2022; Lee et al. 2022.

[5].

[5]We note this phenomena could also be accounted for with a sparsity constraint.

One possibility is that in overtrained situations with only a relatively small number of categories or trial-types (where mixed-selectivity has been observed), the task can effectively be solved by categorising the current trial into one of a few previous experiences. By contrast in tasks where combinatorial generalisation is required, the factored solution may be preferred.

A program to understand how brain representations structure themselves. This work is one piece of the puzzle. It tells us when neural circuits systems should represent different factors in different neurons. It does not tell us, however, how each factor itself should be represented. For example it does not tell us why GCs and OVCs look the ways they do. We believe that the same principles of nonnegativity, minimising neural activity, and representing structure, will be essential components obtaining this more general understanding. Indeed in a companion paper, we use the same constraints, along with formalising structure/path-integration using group and representations theory, to mathematically understand why grid cells look like grid cells Will Dorrell et al. 2023. Similarly, our current understanding is limited to the optimal solution for factorised representations, but we anticipate similar ideas will be applicable to neural dynamics Driscoll, Shenoy, and Sussillo 2022.

5.1.6 Conclusion

We introduced constraints inspired by biological neurons - nonnegativity and energy efficiency (w.r.t. either activity or weights) - and proved these constraints lead to linear factorised codes being disentangled. We empirically verified this in simulation, and showed the same constraints lead to disentanglement with both nonlinear data and nonlinear networks. We even achieve competitive disentanglement scores on a baseline disentanglement task, even though this was not our specific aim. We showed these biological constraints explain why neuroscientists observe bespoke cell types, e.g. GCs Hafting et al. 2005, OVCs Høydal et al. 2019, border vector cells Lever et al. 2009; Solstad et al. 2008, since space, boundaries, and objects appear in a factorised form (i.e. occur in any independent combination), and so are optimally represented by different neural populations for each factor. These same principles explain why neurons in inferior temporal cortex are axis aligned to underlying factors of variation that generate the data they represent Bao et al. 2020; Chang and Tsao 2017; Higgins, Chang, et al. 2021, why visual cortex neurons are decorrelated Ecker et al. 2010, or why neurons in parietal cortex only selective for specific tasks Lee et al. 2022. Lastly, we also explained the confusing finding of grid fields warping towards rewards Boccara et al. 2019 as the space and rewards becoming entangled.

This work bridges the gap between single neuron and population responses, and offers an understanding of properties of neural representations in terms of task structure above and beyond just dimensionality. Additionally it demonstrates the utility of neurobiological considerations in designing machine learning algorithms. Overall, we hope this work demonstrates the promise of a unified research program that more deeply connects the neuroscience and machine learning communities to help in their combined quest to both understand and learn neural representations. Such a unified approach spanning brains and machines could help both sides, offering neuroscientists a deeper understanding of how cortical representations structure themselves, and offering machine learners novel ways to control and understand the representations their machines learn.

5.2 Disentangling using Quantized Latent Codebooks

TL;DR: My rotation work, led by Kyle Hsu. We propose an architecture with an inductive bias towards disentangling: the latent space is quantized and the quantization is compositional. This means the datapoint is mapped to image space, then along each axis it is mapped to the nearest latent encoding. This forces the latent space to be compositional, which matches (in an intuitive sense) the kinds of structure we think underlie factors of variation in the environment. Experiments agree that this latent space is good for disentangling. This is being submitted to Neurips this year.

ABSTRACT

In disentangled representation learning, a model is asked to tease apart a dataset's underlying sources of variation and represent them independently of one another. Since the model is provided with no ground truth information about these sources, inductive biases take a paramount role in shaping to what extent the sources are recovered and how they are represented. In this work, we propose latent quantization, a simple yet effective design choice tailored for disentangling. Latent quantization endows a model with an inductive bias towards compositionally representing the data, a key property of the true sources. We demonstrate the broad applicability of latent quantization by adding it to both basic data-reconstructing (vanilla autoencoder) and latent-reconstructing (InfoGAN) generative models. Surprisingly, this single change is sufficient to propel the modularity, explicitness, and compactness of the learned representations beyond that of strong methods from prior work. In particular, on a comprehensive suite of standard benchmarks, the quantized latent autoencoder consistently dominates in all three disentanglement properties while maintaining competitive data reconstruction. We also present infoMEC, new metrics for disentanglement that are cohesively grounded in information theory and fix well-established shortcomings in previously proposed metrics.

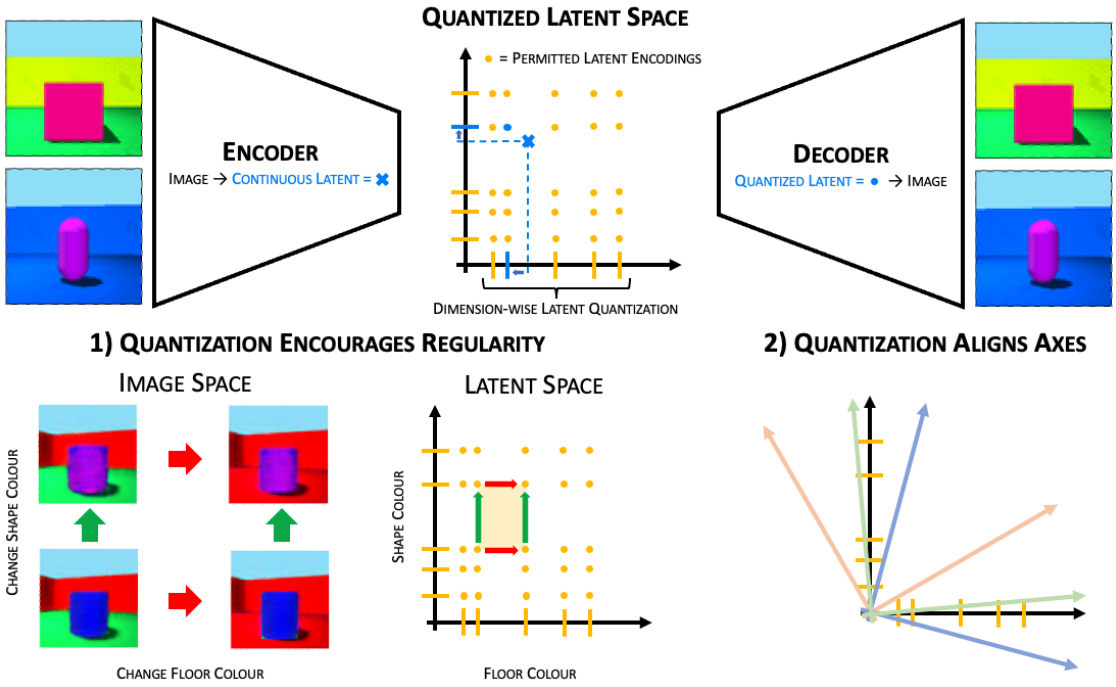

Figure 30: Overview of latent quantization and its inductive bias towards disentanglement.

5.2.1 Introduction

Our increasing reliance on black-box methods for processing high-dimensional data underscores the importance of developing techniques for learning human-interpretable representations. To name but a few possible benefits, such representations could foster more informed human decision-making Xiaowei Huang et al. 2020; Schmidt and Biessmann 2019, facilitate efficient model debugging and improvement Du, Liu, and Hu 2019; Lertvittayakumjorn and Toni 2021,

and streamline auditing and regulation Creager et al. 2019; Mittelstadt 2016. In this context, the problem of disentangled representation learning serves as a worthwhile scaffolding: loosely speaking, the goal is for a model to tease apart the underlying sources of variation in a dataset and represent them independently of one another.

Despite this motivation, disentangled representation learning has turned out to be difficult in several ways. In its full generality, i.e. with no ground truth information about the sources, the problem is theoretically impossible to solve without further assumptions Hyvärinen and Pajunen 1999; Locatello et al. 2019, but satisfyingly general assumptions have proven elusive. Even evaluating the extent to which a representation is disentangled is challenging, with multiple definitions and no universally accepted metrics. Previously proposed metrics have been found to be uncalibrated, sensitive to hyperparameters, or sample inefficient Carbonneau et al. 2022.

In this work, we propose quantizing the latent representation of a model into learnable discrete values, a simple yet effective design choice tailored for disentanglement. Crucially, we maintain a separate scalar "codebook" for every latent dimension, a key detail that distinguishes our scheme from the prototypical methods for discrete representation learning Van Den Oord, Vinyals, et al. 2017. We show that latent quantization introduces an inductive bias for the model to compositionally represent the data, especially when further reinforced by expressive network architectures and standard model regularization techniques. As a side benefit, models endowed with latent quantization sidestep many of the aforementioned issues that have hindered evaluation in previous works since the use of discrete quantities enables simple and well-understood distribution estimation techniques.

We demonstrate the broad applicability of latent quantization by adding it to both basic data-reconstructing (vanilla autoencoder) and latent-reconstructing (InfoGAN) generative models. Surprisingly, this single change is sufficient to propel the modularity, explicitness, and compactness of the learned representations beyond that of strong methods from prior work. In particular, on a comprehensive suite of four disentanglement datasets with image observations and ground-truth source evaluation C. Burgess and H. Kim 2018; Gondal et al. 2019; Nie 2019, the quantized latent autoencoder (QLAE, pronounced like Klay) consistently dominates in all three disentanglement properties while maintaining competitive data reconstruction. As an auxiliary contribution, we also derive and present infoMEC, new metrics for (in decreasing importance) modularity, explicitness, and compactness that are cohesively grounded in information theory and fix well-established shortcomings in previously proposed metrics. We provide a modular code repository implementing all methods considered in this work as well as infoMEC estimation in the supplementary material.

5.2.2 Preliminaries

Unlike most quantitatively-minded areas of computer science, disentangled representation learning has no universally accepted definition. Hence, we will devote some attention to motivating our understanding of the problem.

5.2.2.1 Nonlinear Independent Components Analysis

We begin by borrowing from the formalism of nonlinear independent components analysis (ICA), a related area of study that is much older and more theoretical in focus. Consider a standard nonlinear ICA data-generating model Hyvärinen and Pajunen 1999:

$$p(\mathbf{s}) = \prod_{i=1}^{n_s} p(\mathbf{s_i}), \;\; \mathbf{x} = g(\mathbf{s}) \tag{21}$$

where $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_{n_s})$ comprises the n_s (mutually) independent source variables; $\mathbf{x}$ is the observed data variable; and $g : \mathcal{S} \to \mathcal{X}$ is the nonlinear data-generating function. We denote realizations of the sources, data, and latents as $s \in \mathcal{S}$, $x \in \mathcal{X}$, and $z \in \mathcal{Z}$, respectively.

The nonlinear ICA problem is as follows: given a dataset $\mathcal{D}$ of samples from this model, recover the underlying sources up to some allowable transformations, i.e. find an approximate inverting function $\hat{g}^{-1} : \mathcal{X} \to \mathcal{Z}$ so that $\hat{g}^{-1} \circ g$ is a composition of a permutation and a dimension-wise invertible function. If successful, the latent variables $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_{n_s})$ then correspond to the unobserved source variables up to this allowable equivalence class.

As stated, this problem is *nonidentifiable* (or underspecified). Given $\mathcal{D}$, one may find a set of independent sources (and an associated nonlinear generator) that fits the data but is non-trivially different from the true generative sources Hyvärinen and Pajunen 1999. As such, recovering the true sources from the data is impossible since there simply isn't a unique ground truth. Much recent work in nonlinear ICA has focused on proposing additional problem assumptions so as to provably pare down the possibilities to a single solution, making the problem well-posed. These theoretical assumptions can then be transcribed into architectural choices in g and $\hat{g}^{-1}$ or regularization terms in the objective. Such approaches

have shown some promise in increasing our theoretical understanding of what disentanglement should entail; we elaborate on this in Section 5.2.6.

5.2.2.2 Disentangled Representation Learning

While uniqueness is conceptually appealing, defining sufficiently generalizable assumptions that accurately describe non-toy datasets has proven hard. The field of disentangled representation learning has taken a more pragmatic approach, focusing on empirically evaluating the recovery of a dataset's designated source set. To simulate scenarios in which the true number of sources is unknown, it is standard to select an upper bound $n_z \geq n_s$ on the number of sources we would like to recover and choose this to be the number of latents. To accommodate this change, the strict bipartite source-latent correspondence of nonlinear ICA has to be generalized to two complementary axes. **Modularity** measures the extent to which sources are separated into disjoint sets of latents, whereas **compactness** measures the extent to which latents only contain information about disjoint sets of sources. When $n_z = n_s$, perfect modularity and perfect compactness are equivalent, but when $n_z > n_s$, it is impossible to achieve this for both. Of the two, modularity should be prioritized Carbonneau et al. 2022; Ridgeway 2016 and indeed has been referred to as disentanglement itself Eastwood and C. K. Williams 2018.

We now need a granular measure of the extent to which a particular latent is an invertible function of a particular source. Mutual information is in many ways well-suited for this purpose since it is sensitive to arbitrary nonlinear dependencies between the variables. However, there are several subtle yet impactful issues with how mutual information metrics for modularity and compactness are commonly estimated Carbonneau et al. 2022. In Section 5.2.4, we propose InfoM and InfoC, variants that address these issues.

Finally, the literature sometimes considers an additional property of a disentangled representation. **Explicitness** complements modularity and compactness in measuring the extent to which the sources and latents are *simple* functions of each other. This has been defined and implemented in an ad hoc manner in prior works Eastwood and C. K. Williams 2018; Ridgeway and Mozer 2018. In Section 5.2.4.4, we conceptually ground explicitness as estimating the predictive linear information Yilun Xu et al. 2020 of the sources given by the latents, yielding the InfoE metric.

The disentangled representation learning problem statement considered in this work is as follows. Given a dataset of paired source-data samples from the nonlinear ICA model equation 21, learn an encoder $\hat{g}^{-1}$ without using the source labels such that the infoMEC as measured by the latents (encoded data) and source labels is high, with priority given to InfoM and InfoE over InfoC.

5.2.2.3 Autoencoding and InfoGAN as Data and Latent Reconstruction

We will apply our proposed latent quantization scheme to two complementary, foundational approaches for disentangled representation learning: vanilla autoencoders (AEs) and information maximizing generative adversarial networks (InfoGANs) X. Chen et al. 2016. Here, we provide a brief overview linking the two as reconstructing data and latents, respectively. Both approaches involve learning an encoder $\hat{g}^{-1} : \mathcal{X} \to \mathcal{Z}$ and decoder $\hat{g} : \mathcal{Z} \to \mathcal{X}$. An autoencoder takes a datapoint $x \in \mathcal{X}$ as input and produces a reconstruction $\hat{g} \circ \hat{g}^{-1}(x) \in \mathcal{X}$ that is optimized to match the input:

$$\mathcal{L}_{\text{reconstruct data}}(\hat{g}^{-1}, \hat{g}; \mathcal{D}) := \mathbb{E}_{x \sim \mathcal{D}} \left[-\log p(x \mid \hat{g} \circ \hat{g}^{-1}(x)) \right]. \tag{22}$$

An InfoGAN instead takes a latent code $z \in \mathcal{Z}$ as input. The decoder (aka generator) maps z to the data space, and from this the encoder produces a reconstruction of the latent:

$$\mathcal{L}_{\text{reconstruct latent}}(\hat{g}^{-1}, \hat{g}) := \mathbb{E}_{z \sim p(\mathbf{z})} \left[-\log p(z \mid \hat{g}^{-1} \circ \hat{g}(z)) \right]. \tag{23}$$

This is clearly insufficient for learning as the dataset $\mathcal{D}$ is not even used. InfoGAN in full can be interpreted as grounding latent reconstruction by additionally optimizing for the marginal distribution of generations $\hat{g}(\mathbf{z})$ to be indistinguishable from the empirical data distribution. A concrete measure of this latter property is provided by a binary classifier (discriminator) Goodfellow et al. 2014 or a value regressor (critic) Arjovsky, Chintala, and Bottou 2017 trained alongside but in opposition to the decoder. While InfoGAN was originally motivated as optimizing a variational bound on the mutual information between the latent and the generated data, we find interpreting it as latent autoencoding (plus data distribution matching) to be clarifying.

5.2.3 Latent Quantization

Regardless of what flavor of generative model we consider, a ubiquitous choice for the latent space $\mathcal{Z}$ is $\mathbb{R}^{n_z}$, the n_z-dimensional real space. The main technical contribution of this work is to instead propose for $\mathcal{Z}$ to be a learnable discrete

latent set $Z = V_1 \times \cdots \times V_{n_z}$, the Cartesian product of n_z distinct sets. Each of the component sets $V_j = \{v_{jk}\}_{k=1}^{n_v}$ consists of n_v reals. To encode a datapoint, it is first mapped through an encoder network $\hat{g}^{-1} : \mathcal{X} \to \mathbb{R}^{n_z}$, then quantized along each dimension:

$$z_j = \arg\min_k |\hat{g}^{-1}(x)_j - v_{jk}|, \;\; v_{jk} \in V_j, \;\; j = 1, \ldots, n_z. \tag{24}$$

This is equivalent to finding the latent $z \in Z$ closest to $\hat{g}^{-1}(x)$ in ℓ_1 norm, but the optimization in equation 24 is much easier to do in practice as it avoids explicitly representing all ${n_z}^{n_v}$ elements of Z. Instead, we represent Z by its constituent discrete values, concretely as a learnable two-dimensional array $\mathrm{V} \in \mathbb{R}^{n_z \times n_v}$. We adapt techniques from vector-quantized variational autoencoders (VQ-VAE) Van Den Oord, Vinyals, et al. 2017 to facilitate optimization of V and the flow of gradients through the nondifferentiable quantizing operation via straight-through estimation Bengio, Léonard, and Courville 2013. We present pseudocode for training a quantized latent autoencoder (QLAE), a vanilla autoencoder endowed with latent quantization, in Algorithm 2.

Algorithm 2: Pseudocode for optimizing a quantized latent autoencoder (QLAE). All experiments in the paper used fixed hyperparameter values.

Require: Input: dataset $\mathcal{D}$, batch size b, AdamW hyperparameters $(\alpha, \beta_1, \beta_2, \text{weight decay})$, loss weights $\lambda_{\text{reconstruct}}, \lambda_{\text{quantize}}, \lambda_{\text{commit}}$

initialize encoder $\hat{g}^{-1} : \mathcal{X} \to \mathbb{R}^{n_z}$, discrete value array $\mathrm{V} \in \mathbb{R}^{n_z \times n_v}$, decoder $\hat{g} : \mathbb{R}^{n_z} \to \mathcal{X}$
while $\theta := (\hat{g}^{-1}, \mathrm{V}, \hat{g})$ has not converged **do**
  for $i = 1, \ldots, b$ **do**
    $x \sim \mathcal{D}$
    $z_c \leftarrow \hat{g}^{-1}(x)$
    $z \leftarrow \arg\min_{v \in Z} ||z_c - v||_1, \;\; v_j \in V_j, \;\; \|_{j=1}^{n_z} V_j = \mathrm{V}$ {Implement via equation 24.}
    $\mathcal{L}_{\text{quantize}} \leftarrow \; ||\text{StopGradient}(z_c) - z_2^2||$
    $\mathcal{L}_{\text{commit}} \leftarrow ||z_c - \text{StopGradient}(z)||_2^2$
    $z \leftarrow z_c + \text{StopGradient}(z - z_c)$ {Straight-through estimator.}
    $\mathcal{L}_{\text{reconstruct}} \leftarrow \text{BinaryCrossEntropy}(\hat{g}(z), x)$
    $\mathcal{L}_{\text{QLAE}}^{(i)}(\hat{g}^{-1}, \mathrm{V}, \hat{g}) = \lambda_{\text{reconstruct}}\mathcal{L}_{\text{reconstruct}} + \lambda_{\text{quantize}}\mathcal{L}_{\text{quantize}} + \lambda_{\text{commit}}\mathcal{L}_{\text{commit}}$
  end for
  $\theta \leftarrow \text{AdamW}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{b} \mathcal{L}_{\text{QLAE}}^{(i)}, \theta, \alpha, \beta_1, \beta_2, \text{weight decay})$
end while

5.2.3.1 The Inductive Bias of Quantized Latent Representations

Why would latent quantization help with disentanglement? We remark that the resulting latent set exhibits the following property: every code (latent realization) z has $\binom{n_z}{d}(n_v - 1)$ d-neighbors z' such that $||z - z'||_0 = d$, where $||\cdot||_0$ is the ℓ_0 "norm" or Hamming distance. We conjecture that every code sharing $n_z - d$ *exact* values with its d-neighbors comprises an inductive bias towards the model assigning a consistent, disentangled meaning to every discrete value in every dimension. In comparison, for a continuous latent space such as $\mathbb{R}^{n_z}$, something analogous to the d-neighbors property would have to be *learned* as it would occur with vanishing probability at initialization.

To gain intuition for this inductive bias, contrast the aforementioned strategy with another valid scheme for autoencoding the data: treating the quantized latent set as an index set, i.e. fixing an arbitrary ordering $(z_l)_{l=1}^{{n_v}^{n_z}}$, then redefining the encoder as $\hat{g}^{-1}(x) = l$. But this is fantastically difficult to actually achieve, as it requires the encoder to classify between $|\mathcal{D}|$ classes, and, even worse, requires the decoder to memorize every single datapoint *separately* for decoding, as a datapoint's code would only indicate its identity and none of its content. Indeed, this terrible idea is essentially doing the reverse of learning embedding spaces for discrete data such as nodes in graph representation learning and tokens in natural language processing.

But consider what happens if a putative source $\mathbf{s}_i$ that manifests in the dataset with m values can be consistently identified by the encoder and interpreted by the decoder. Then all (or much) of the information about $\mathbf{s}_i$ in the $\approx |\mathcal{D}|/m$ datapoints with a particular realization s_{ik} can be represented by the same value v_{jk}, so the onerous memorization is eased by a factor of m. And why would these m values be stored along the same latent dimension, as opposed to being distributed in multiple? As realizations of the same variable, they are *mutually exclusive*, a property mirrored by each latent dimension's realizations V_j since equation 24 specifies a winner-takes-all optimization.

Another way to think about the inductive bias of latent quantization is that for every code z with entry $z_j = v_{jk}$, there are (up to) ${n_z}^{n_v-1}$ other codes z' that also have $z'_j = v_{jk}$. Assigning a consistent meaning to $z_j = v_{jk}$ would make it both easier for the encoder to correctly encode all the myriad datapoints who should have it in their latent, and easier for the decoder to apply the effect of this value when decoding all the myriad codes who have it. We can thus view latent quantization as helping the encoder and decoder conspire to autoencode most efficiently and correctly, which intuitively also describes the relationship between the ground-truth sources and data-generating function of datasets we care about.

A final important feature of the bias is that it makes the neuron basis the preferred basis, ensuring that when we go looking for modules we know where to look: along the neuron axes.

5.2.4 InfoMEC: Information-Theoretic Metrics for Disentanglement

Before we empirically test the conjectured inductive bias of latent quantization, we first need to ensure that we can properly measure disentanglement. The question of how to do so has been a consistent source of controversy. Here, we give a unifying presentation of how we measure modularity, explicitness, and compactness, all of which are heavily based on or inspired by prior work Carbonneau et al. 2022; R. T. Chen et al. 2018; Eastwood and C. K. Williams 2018; Z. Li et al. 2020; Ridgeway 2016; J. C. Whittington, Will Dorrell, et al. 2022. In particular, the idea of disentangling disentanglement into these three properties (or similar ones) was independently proposed by Eastwood and C. K. Williams 2018 and Ridgeway and Mozer 2018, and we use the terminology suggested by Carbonneau et al. 2022. We focus on providing a thorough accounting of the several key design decisions involved. We do not expect this to be the final word on disentanglement metrics, especially since our empirical study is in many ways idealized, so we hope this presentation enables others to clearly understand the decision-making underlying our proposed metrics, infoMEC, and propose further improvements.

5.2.4.1 Mutual Information Estimation

When the number of sources and latents are the same, the ICA problem asks for the latents to recover the sources up to a permutation and componentwise invertible transformation. A suitable quantity for measuring the latter is the mutual information between individual sources and latents:

$$I(\mathbf{s}_i; \mathbf{z}_j) := D_{\mathrm{KL}} p(\mathbf{s}_i, \mathbf{z}_j) p(\mathbf{s}_i) p(\mathbf{z}_j), \tag{25}$$

a symmetric measure of arbitrarily nonlinear dependence between the two variables. Estimating this is already nontrivial. If both variables are discrete, we can explicitly calculate the empirical joint distribution of $(\mathbf{s}_i, \mathbf{z}_j)$ and apply Eq. 25. If either variable is continuous, it is common to quantize the realizations via binning before running the above estimation Carbonneau et al. 2022; Locatello et al. 2019. Perhaps this has been considered acceptable since the variables are scalar and the curse of dimensionality does not apply. However, we have found that the outcome is nonetheless very sensitive to the binning strategy, with binning that is too coarse resulting in overestimation and vice-versa. This echoes concerns raised by others Carbonneau et al. 2022.

The mutual information estimation literature has long tackled this problem. It seems a waste to not use the fruits of their labor. For the continuous-continuous case, a de facto standard is the k-nearest neighbor based KSG estimator Kraskov, Stögbauer, and Grassberger 2004 which has seen widespread use due to its practical performance and been proven to enjoy desirable properties such as consistency W. Gao, Oh, and Viswanath 2018. The datasets for quantitative evaluation considered in this work have discrete sources, so for evaluating models with continuous latents we use a well-known variant of KSG introduced by Ross 2014. We remark that an additional benefit of latent quantization is that it enables reliable evaluation using the simpler discrete-discrete estimator.

5.2.4.2 Normalization

To facilitate aggregation and comparison, we would like to normalize estimated information to a fixed interval such as $[0, 1]$. To this end, we first state a basic identity:

$$I(\mathbf{s}_i; \mathbf{z}_j) = H(\mathbf{s}_i) - H(\mathbf{s}_i \mid \mathbf{z}_j). \tag{26}$$

For a discrete source, entropy is nonnegative, so $I(\mathbf{s}_i; \mathbf{z}_j)$ is bounded above by $H(\mathbf{s}_i)$. We define a normalized mutual information as

$$\mathrm{NMI}(\mathbf{s}_i, \mathbf{z}_j) := \frac{I(\mathbf{s}_i; \mathbf{z}_j)}{H(\mathbf{s}_i)}, \tag{27}$$

a choice first made by R. T. Chen et al. 2018 in the context of disentangled representation learning. While a plethora of other normalization schemes (based on some function of both marginal entropies) has been suggested for other contexts,

this one is particularly convenient for us. It is neatly interpreted as the proportion of entropy in a source reduced by conditioning on a latent, and thus provides a consistent scaling to $[0, 1]$ over the course of model optimization. Also, the asymmetry induced by only considering the marginal source entropy avoids the scale-dependent (and possibly negative) marginal differential entropy of a continuous latent, which is itself non-trivial to estimate Kraskov, Stögbauer, and Grassberger 2004.

5.2.4.3 Modularity and Compactness

We gather all $n_s \times n_z$ evaluations of $\mathrm{NMI}(\mathbf{s}_i, \mathbf{z}_j)$ into a 2-dimensional array $\mathrm{NMI} \in [0, 1]^{n_s \times n_z}$. We then remove columns of NMI corresponding to inactive latents, which we define heuristically as those whose range over the evaluation sample is less than $1/8$ of that of the latent with largest range. We state this pruning criterion for posterity, noting that previous works by and large omit their definition.

Recall that modularity is the extent to which sources are separated into disjoint sets of latents. Thus, a representation exhibits perfect modularity when every latent is informative of only one source, i.e. when every column of NMI is monomial (has only one nonzero element). For a granular measure of modularity, we can consider the *gap* between the two largest entries in the column R. T. Chen et al. 2018, or alternatively the *ratio* of the largest entry in the column to the column sum J. C. Whittington, Will Dorrell, et al. 2022. We prefer the ratio since the gap is agnostic to the smallest $n_s - 2$ values in the column, but how close these values are to 0 matters and should be reflected in the metric. Since the possible range of values for this ratio is $[1/n_s, 1]$, we re-normalize to $[0, 1]$. Finally, we define modularity as the average of this quantity over latents:

$$\text{InfoM} := \frac{\frac{1}{n_z}\sum_{j=1}^{n_z} \frac{\max_i \mathrm{NMI}_{ij}}{\sum_{i=1}^{n_s} \mathrm{NMI}_{ij}} - \frac{1}{n_s}}{1 - \frac{1}{n_s}}. \tag{28}$$

Compactness is the dual of modularity; recall that it is the extent to which latents only contain information about disjoint sets of sources. We therefore define it analogously to modularity, but considering columns of NMI instead of rows, etc.:

$$\text{InfoC} := \frac{\frac{1}{n_s}\sum_{i=1}^{n_s} \frac{\max_j \mathrm{NMI}_{ij}}{\sum_{j=1}^{n_z} \mathrm{NMI}_{ij}} - \frac{1}{n_z}}{1 - \frac{1}{n_z}}. \tag{29}$$

We advocate for this terminology since previously proposed names such as "mutual information gap" R. T. Chen et al. 2018 and "mutual information ratio" J. C. Whittington, Will Dorrell, et al. 2022 are ambiguous, and indeed the former of these works considered solely compactness and the latter solely modularity, with neither mentioning the distinction.

5.2.4.4 Explicitness

Because we measure modularity and compactness in terms of mutual information, the functional manifestation of the dependence between source and latent can be arbitrarily complex. The notion of explicitness complements this by measuring the dependence when the functional form is limited, e.g., linear Eastwood and C. K. Williams 2018; Ridgeway and Mozer 2018. Unfortunately, the explicitness metrics used in prior works have been heuristic and ad hoc. Instead, we propose to conceptually ground explicitness in the language of predictive $\mathcal{V}$-information, a generalization of mutual information that specifies an allowable function class, denoted $\mathcal{V}$, for the computation of the information. Our presentation of predictive $\mathcal{V}$-information is informal for the sake of exposition, but readers are encouraged to consult the original formulation Yilun Xu et al. 2020 for technical precision.

To calculate explicitness, we first estimate the predictive $\mathcal{V}$-information of each source $\mathbf{s}_i$ given by all of the (active) latents $\mathbf{z}$ *jointly* (in contrast to the pairwise mutual information estimation done for modularity and compactness):

$$I_{\mathcal{V}}(\mathbf{z} \to \mathbf{s}_i) = H_{\mathcal{V}}(\mathbf{s}_i \mid \emptyset) - H_{\mathcal{V}}(\mathbf{s}_i \mid \mathbf{z}). \tag{30}$$

This requires estimating the predictive conditional $\mathcal{V}$-entropy

$$H_{\mathcal{V}}(\mathbf{s}_i \mid \mathbf{z}) = \inf_{f \in \mathcal{V}} \mathbb{E}_{s \sim p(\mathbf{s}), z \sim p(\mathbf{z}|\mathbf{s})} \left[-\log p(s_i \mid f(z)) \right] \tag{31}$$

and the marginal $\mathcal{V}$-entropy of the source

$$H_{\mathcal{V}}(\mathbf{s}_i \mid \mathbf{z}) = \inf_{f \in \mathcal{V}} \mathbb{E}_{s \sim \mathbf{s}} \left[-\log p(s_i \mid f(\emptyset)) \right], \tag{32}$$

where $\emptyset$ is an uninformative constant. The first entropy estimation is essentially supervised learning of the i-th source from the latents via maximum likelihood, where the dataset comes from the data-generating process and the encoder.

Table 1: Summary of disentanglement results. infoMEC := (InfoM, InfoE, InfoC). Modularity is the key property, followed by (linear) explicitness, with compactness a distant third.

	infoMEC ↑			
model	Shapes3D	MPI3D	Falcor3D	Isaac3D
QLAE (ours)	$0.94, 1.00, 0.50$	$0.54, 0.62, 0.47$	$0.72, 0.74, 0.38$	$0.73, 0.93, 0.46$
β-TCVAE R. T. Chen et al. 2018	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$
β-VAE Higgins, Matthey, et al. 2017	$0.52, 1.00, 0.51$	$0.45, 0.71, 0.50$	$0.71, 0.73, 0.70$	$0.65, 0.81, 0.59$
BioAE J. C. Whittington, Will Dorrell, et al. 2022	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$
VQ-VAE Van Den Oord, Vinyals, et al. 2017	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$
AE	$0.33, 1.00, 0.20$	$0.33, 0.71, 0.22$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$
QLInfoWGAN-GP (ours)	$0.61, 0.45, 0.53$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$
InfoWGAN-GP X. Chen et al. 2016	$0.45, 0.24, 0.65$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$
WGAN-GP Gulrajani et al. 2017	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$	$0.00, 0.00, 0.00$

This corresponds intuitively to the notion of measuring the dependence of the source on the latents, with the allowable complexity of the dependence determined by $\mathcal{V}$. While this has the semblance of a machine learning problem, note that generalization is not considered; the estimation calls for the best *in-sample* negative log likelihood.

We choose $\mathcal{V}$ to be the space of linear models and so run logistic regression for classifying a discrete source and linear regression for regressing to a continuous source. We use no regularization for either algorithm. We compute the marginal $\mathcal{V}$-entropy $H_{\mathcal{V}}(\mathbf{s}_i \mid \emptyset)$ by the same procedure but with the inputs replaced with a constant (zero). We remark that this formulation of explicitness elegantly enables authors to pick other function classes $\mathcal{V}$ as appropriate for their needs.

We propose a simple normalization analogous to the one done for modularity and compactness:

$$\mathrm{NMI}_{\mathcal{V}}(\mathbf{z} \to \mathbf{s}_i) := \frac{I_{\mathcal{V}}(\mathbf{z} \to \mathbf{s}_i)}{H_{\mathcal{V}}(\mathbf{s}_i \mid \emptyset)}, \tag{33}$$

which can be interpreted as the relative reduction in the $\mathcal{V}$-entropy of a source achieved by knowing the latents. For logistic regression, the negative log likelihood is computed as cross-entropy. For linear regression, we leverage Propositions 1.3 and 1.5 from Yilun Xu et al. 2020 to argue that $\mathrm{NMI}_{\mathcal{V}}(\mathbf{z} \to \mathbf{s}_i) = R^2$, the coefficient of determination for the nontrivial regression. Both cases result in a value in $[0, 1]$. We can now compute explicitness as

$$\text{InfoE} := \frac{1}{n_s} \sum_{i=1}^{n_s} \mathrm{NMI}_{\mathcal{V}}(\mathbf{z} \to \mathbf{s}_i). \tag{34}$$

We now have cohesively information-theoretic metrics for evaluating the modularity, explicitness, and compactness of a representation, each with range $[0, 1]$. We collectively refer to them as $infoMEC :=$ (InfoM, InfoE, InfoC) and provide our implementation for estimating them in the supplementary material.

5.2.5 Experiments

Ongoing experiments are testing a suite of methods on four datasets using our suite of metrics. The results are incomplete but look promising for our approach.

To our knowledge these are the strongest autoencoder variants that do not bring in additional complexity such as progressive training Karras, Laine, and Aila 2019; Z. Li et al. 2020 or make limiting assumptions about the dataset Leeb et al. 2023.

5.2.6 Related Work

Representation learning with discrete representations. Discrete representation can be a natural fit for many deep learning problems: the data being modelled is often discrete (such as language); reasoning seems more natural using discrete encodings; and discretization seems likely to make latents more interpretable. Their use was initially limited by optimization difficulties. VQ-VAE Van Den Oord, Vinyals, et al. 2017) broke through these problems by using straight-through gradient estimation and a well-designed loss function; we adapt both techniques for our purposes. VQ-VAE and variants have been successfully applied to videos Walker, Razavi, and Oord 2021; Yan et al. 2021, audio Baevski, Schneider, and Auli 2019; Borsos et al. 2022; Dhariwal et al. 2020; Tjandra, Sakti, and Nakamura 2020,

Table 2: Image reconstruction achieved by runs in Table 1 in peak signal-to-noise ratio (PSNR).

model	PSNR (dB) ↑ Shapes3D	MPI3D	Falcor3D	Isaac3D
QLAE (ours)	34	36	29	39
β-TCVAE R. T. Chen et al. 2018 β-VAE Higgins, Matthey, et al. 2017	35	38	29	38
BioAE J. C. Whittington, Will Dorrell, et al. 2022 AE	37	39	23	
QLInfoWGAN-GP (ours)	17			
InfoWGAN-GP X. Chen et al. 2016 WGAN-GP Gulrajani et al. 2017	14			

anomaly detection Marimont and Tarroni 2021, and are part of large text-to-image systems like DALL-E Ramesh et al. 2021, Parti J. Yu et al. 2022, and latent diffusion models Rombach et al. 2022.

Of particular relevance are works that learn multiple codebooks, or, equivalently, partition the representation so that certain dimensions can only access certain codes. A few papers, like us, use one scalar codebook per latent dimension, largely for efficiency in retrieval Ballé, Laparra, and Simoncelli 2016; Kaiser et al. 2018; Theis et al. 2017; Wu and Flierl 2019. Others project to multiple subspaces which each have their own codebook Lu et al. 2023. Most related to us in purpose is the work of Kobayashi et al. 2021 who use two separate codebooks to disentangle medical images into normal and abnormal features. Their work enforces disentangling using supervision—each training image comes with a mask labeling the abnormality, and the abnormal codebook is only used to predict this mask. Our approach, in contrast, uses per-dimension codebooks to disentangle all the sources of variation in a dataset without requiring the sources to be specified via supervision.

Nonlinear ICA and disentangled representation learning. There is a long history of trying to build interpretable representations that tease out the sources of variation in a dataset. This goes back to classic work on (linear) ICA Comon 1994; Hyvärinen and Oja 2000, and has been known in deep learning as disentangling Bengio 2013. Without assumptions, nonlinear ICA and its relatives in the disentangling literature are provably underspecified Hyvärinen and Pajunen 1999; Locatello et al. 2019. One class of methods uses weak or partial supervision, such as labeling a small number of sources in a dataset Nie et al. 2020, or showing pairs of datapoints in which only one or a few sources differ Shu et al. 2019. Of more relevance for us are approaches that assume additional source structure and disentangle by ensuring the representation respects this structure Xi and Bloem-Reddy 2023. Assumptions and methods include: VAEs with factorized priors R. T. Chen et al. 2018; Higgins, Matthey, et al. 2017, biologically inspired activity constraints J. C. Whittington, Will Dorrell, et al. 2022, sparse source variation over time Klindt et al. 2020; Sprekeler, Zito, and Wiskott 2014, (structurally) sparse source to pixel influence Moran et al. 2021; Y. Zheng, Ng, and Kun Zhang 2022, geometric assumptions on the source to image mapping Gresele et al. 2021; Horan, Richardson, and Weiss 2021; Sorrenson, Rother, and Köthe 2020; X. Yang et al. 2022, hierarchical decoding Leeb et al. 2023, sparse underlying causal graphs between sources Lachapelle and Lacoste-Julien 2022; Lachapelle, Rodriguez, et al. 2022, and piecewise linearity Kivva et al. 2022. Our method, latent quantization, is a continuation of this literature's ongoing search for generally useful inductive biases for disentangling.

5.2.7 Discussion, Conclusions, & Further Work

In sum, we've shown that a quantized latent space with a bias towards compositionality exerts a helpful tug towards disentangling.

There are many things that it would be fun to try in the future. It should be possible to be more precise about the way this latent representations encourages compositionality. Perhaps only certain types of compositionality can be disentangled by this kind of network. If so it would be very fun to make datasets with each of these ideas of compositionality in them, and see which methods were good at disentangling. My suspicion would be that we've been conflating types of composition that the network does not see as the same.

One major concern is that we might just have overfit to these noisless discrete benchmark datasets - especially given the exhaustive nature of these datasets (all settings of all datasets are included). We could probe this by using datasets that are far more continuous, though even now there are 5 times more settings of the sources than there are latents.

We make a couple of technical obserbations. First, for both VAE-based and GAN-based methods, going from the vanilla non-disentanglement method to the basic disentanglement method *increased* training stability. This was particularly

dramatic for the GAN. Second, we specify a fixed number of values per latent. We notice that models consistently prune down the number of values used. This is likely due to the weight regularization and quantization/commitment loss: reducing redundant coding simplifies the decoder's job, and having fewer targets for similar-looking features simplifies the encoder's job.

It would be interesting to see if this disentangling can work in interesting places that break previous classic assumptions. For example, can it still make progress when the sources are correlated? Or can it combinatorially generalise? i.e. can you provide data such that the marginal of each source has full support, but the full distribution does not, and generalise to unseen examples?

Finally, since this work is being done in a robotics lab, it would be fun if we could begin to use disentangled representations to solve other problems. In my mind there are three exciting targets. The first, is a good exploration policy, this is hard to come up with, but disentangled variables tend to encode for semantic things. Learning an exploration policy that causes sparse variations along these axes seems like a good way to generate interesting exploration data. Secondly, hierarchical RL struggles to choose the state space on which to choose high and low-level actions. This disentangling has a notion of hierarchy, through the layer at which each latent is disentangled. Using that to define a hierarchical RL agent would be a lot of fun. Finally, many distribution shifts can likely be understood as shifts in some disentangled variable, by separating out the factors perhaps downstream decoders become robust to such distribution shifts?

5.3 Modularising with Biological Constraints

So, it appears that getting a neural network to perform tasks while satisfying some biological constraints (positive firing rates, and low energy usage via penalising L2 activity and weight norm) leads to modularisation, at least sometimes. We've even used these ideas to understand modularity and mixed-selectivity in the brain. Our goals will be to elucidating exactly what properties tasks have to have in order for modularising to be optimal, and using these ideas to understand findings in artificial and biological networks.

We will make progress towards these goals in a couple of simple linear settings, and we will outline ways that we might hope to extend this in the future.

5.3.1 Linear Disentangling

The original paper, as in section 5.1, says the following. Let's say you have data, $\boldsymbol{x}$, that is an orthogonal mixture of a vector of independent bounded sources $\boldsymbol{Os}$ ($\boldsymbol{O}$ is an orthogonal matrix, $\boldsymbol{s}$ a vector of sources). Then you linearly create a neural representation, $\boldsymbol{g} = \boldsymbol{W}\boldsymbol{x}$. We seek the optimal representation such that:

- **Functional:** The representation encodes all the information such that you can reconstruct the data linearly, i.e. there's some matrix $\boldsymbol{B}$, and $\boldsymbol{Bg} = \boldsymbol{x}$.
- **Biological:** Both activity, $||\boldsymbol{g}||^2$, and weights norm $||\boldsymbol{W}||^2$, $||\boldsymbol{B}||^2$, are minimised, and the activities are all positive.

It turns out that in this case, as long as the number of neurons is larger than the number of sources each neuron responds to a single source! It demixes them! We will develop this in two ways. First, we will show that the set of conditions under which it will always disentangle is actually larger. Second, we will show that when these assumptions are broken there is a phase transition, for small deviations things still disentangle, at large deviations they don't.

5.3.1.1 Biology Disentangles with Weaker Assumptions than thought!

There is a proof in the main paper, but we will do a version of this proof that actually relaxes two of these assumptions. First we will require not that the data is a linear function of some random variables, but that it is a bounded function that can be rotated into a basis such that each subspace depends on only one source:

$$\boldsymbol{x}(\boldsymbol{s}) = \begin{bmatrix} \boldsymbol{x}(s_1) \\ \boldsymbol{x}(s_2) \end{bmatrix}$$

Second, we won't require that the sources are independent, rather we require something called range independence: knowing the value of one source tells you nothing about the range of possible values for the second source. This is much more flexible, it permits variables to be correlated! (a similar assumption to Roth et al. 2022). If the data is a linear function of the sources we can even relax this further, and say that if one source takes its maximum or minimum value, the other sources can still take both their maximum or minimum values. But since we're now in this non-linear world we must stick to range independence.

Now because the representation is a linear function of the data we know:

$$\boldsymbol{g}(\boldsymbol{s}) = \sum_i \boldsymbol{e}_i(c_i(s_1) + d_i(s_2) + b_i)$$

Where we've expressed the neural activity in the neuron basis. Due to the linearity of everything each basis vector is multiplied by coefficients that are either constant (bias) or depend on one of the two sources. We should choose the bias to be as small as possible to achieve positivity, any other solution wastes activity, therefore:

$$b_i = -\min_{s_1} c_i(s_1) - \min_{s_2} d_i(s_2)$$

Let's create a new variable that is always positive:

$$\Delta c_i(s_1) = c_i(s_1) - \min_{s_1} c_i(s_1)$$

Then:

$$\boldsymbol{g} = \sum_i \boldsymbol{e}_i(\Delta c_i(s_1) + \Delta d_i(s_2))$$

So the activity loss is:

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{s}}[\boldsymbol{z}^T\boldsymbol{z}] = \mathbb{E}_{\boldsymbol{s}}\left[\sum_i \Delta c_i^2(s_1) + \Delta d_i^2(s_2) + \Delta d_i(s_2)\Delta c_i(s_1)\right]$$

$$= \mathbb{E}_{s_1}\left[\Delta \boldsymbol{c}^T \Delta \boldsymbol{c}(s_1)\right] + \mathbb{E}_{s_2}\left[\Delta \boldsymbol{d}^T \Delta \boldsymbol{d}(s_2)\right] + \mathbb{E}_{s_1}\left[\Delta \boldsymbol{d}(s_2)\right]^T \mathbb{E}_{s_2}\left[\Delta \boldsymbol{c}(s_1)\right]$$

Now, my claim is that the value of the first two terms will depend on some, potentially subtle, tradeoff between activity and weight regularisation. The last term, on the other hand, measures the dot product of some positive only vectors. The only way for it to be zero is for the coefficients to be non-zero in disjoint elements of the vectors, so if we can show that there is way to minimise the first two while letting the dot product be zero, then we're good, it's disentangled!

If we have many neurons my argument goes as follows. Let's say you give me some $\boldsymbol{g}$ you say is optimal. I can take the s_1 dependent parts of the neural activity and put them in some separate neural population, the first two terms won't have changed, but the third term will have dropped to zero, and this can be acheived without changing the weight norm. Therefore the best you can do is put the representations in two different populations.

5.3.1.2 Breaking assumptions leads to a disentangling phase transition

What happens if these assumptions are not perfectly satisfied?

For example consider the linear data generating situation, $\boldsymbol{x} = \boldsymbol{O}\boldsymbol{s}$, what if $\boldsymbol{O}$ is not orthogonal? The way to think about this is consider each source's data generating vector, i.e. the columns of the matrix, $\boldsymbol{o}_i$. In the limit, if two of these vectors align, then it is crazy to ask the sources to be disentangled, they cause variations along the same directions - they're indistinguishable!

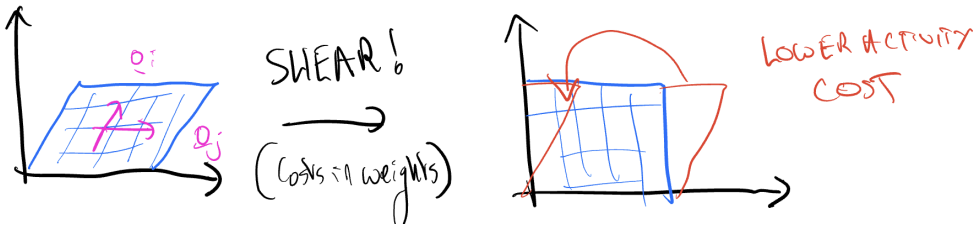

Figure 31: If the data is not orthogonal than the lowest weight norm embedding is just a rotation of some sort, that preserves the non-orthogonality. However, the activity is lower if you shear the data-generating factors into orthogonality. Below some threshold dot product the activity wins out and you disentangle!

But if they only partially align then we actually get a tradeoff. On the one hand, activity minimisation would say you should push the activity against the axes, and so reduce activity. But weight minimisation would say, don't screw with the data! This kind of shearing operation will cost you in weight norm. The tradeoff of these two will set a transition point, below this value of dot product of the two vectors disentangling will occur in the optimal solution, above, it won't, figure 31. The exact point will therefore depend on the weight to activity regularisation ratio. My guess, based on some intricate energy calculations (J. J. Harris, Jolivet, and Attwell 2012), that say synapses are twice as costly as neural activity, would be that that ratio should be 2:1, though units become very important here. Setting this value at 2:1, the threshold alignment dot product is around 0.55, below this sources are disentangled, above they're not.

We have a rough hold on the theory here, that should be enough to do some approximate predictions of the phase transition point. If we wanted we could also go hard core stats phys on it which could be fun.

There's a whole load of important parameters, for example distribution of the sources will also effect the position of the phase transition. Further, there is likely a phase transition in terms of range independence. If you break this slightly it will be fine, but too much and it will begin to entangle.

Anyway, there's a lot to explore about this behaviour, lay it all out and hope the ideas carry over to when and why nonlinear or biological networks disentangle and modularise.

5.3.2 Functional disentangling in linear RNNs

So far we've been dealing with feedforward linear representations of data with particular properties, and it's not immediately obvious how to relate this to the modularising we were finding in grid cells, and to the functional

modularisation that people observe in RNNs. In this section we will use the relaxed version of the assumptions to give another view on why we observed modularisation in our grid cells work. We will then use that to predict modularisation in linear RNNs in a simple setting. We will then suggest how this scheme can be extended to predict the modularisation of any network where we know the activity patterns that must be there, just not how their arranged relative to the neuron basis.

5.3.2.1 Modularisation of Grid Cells

Why did grid cells modularise?

Well, actionability told us the code was a sum of a small number of frequencies:

$$\boldsymbol{g} = \boldsymbol{a}_0 + \sum_d \boldsymbol{a}_d \cos(\boldsymbol{k}_d \boldsymbol{x}) + \boldsymbol{b}_d \sin(\boldsymbol{k}_d \boldsymbol{x}) \tag{35}$$

Let's twist our head and view this in a disentangling way. Each of these frequencies is a source variable and they're all obviously dependent on one another. However interestingly some sets of frequencies are range independent with respect to each other, others are not, figure 32. So, if your code is made up of multiple sets of frequencies that are each harmonics of one another, the non-harmonically related ones should modularise, the others shouldn't! Great.

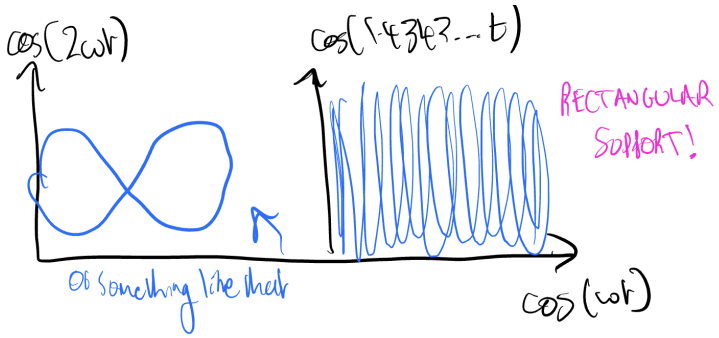

Figure 32: Two non-harmonically related frequencies have rectangular support, so should be modularised. Harmonically related ones shoudln't!

So we can understand both types of modularisation in the same framework (which has implications for what compositionality means that we'll discuss later). For now though, our expanded view of what a source is and what makes it disentangle can let us predict modularisation in other networks!

5.3.2.2 Modularisation of a linear RNN

As a simple first test case, imagine a linear RNN (a linear dynamical system) with no input that at the beginning of every trial is intialised in some state. Its job is to map linearly to a two dimensional output that is an orthogonal mixture of two signals that oscillate at particular frequencies. How should it do it? And should it modularise? (To answer that second question we need a definition of modularisation - for now we will say that it is modularised if the network falls into two sets of neurons, each defined by one frequency, if you chop out that set of neurons it will fail to output the corresponding frequency)

Well, maybe you can see why we chose this toy example. In order to solve this task the internal activity of network has to have loops that oscillate at each of the two required frequencies. If those frequencies are harmonics of one another then they shouldn't modularise, if they're not harmonically related (where the important measure of non-harmonicity is the linear, extremal, version of the range independence we discussed earlier) then they should fall into different popolations of neurons! I've run this and it works, figure 33.

5.3.3 Future Directions

So, it seems like there's a lot of interesting structure that we can use to understand why networks sometimes modularise and sometimes don't, as long as everything stays linear. An artificial setting where this could be valuable is shown in the recent studies on modularisation of RNNs (Driscoll, Shenoy, and Sussillo 2022; G. R. Yang et al. 2019); biologically

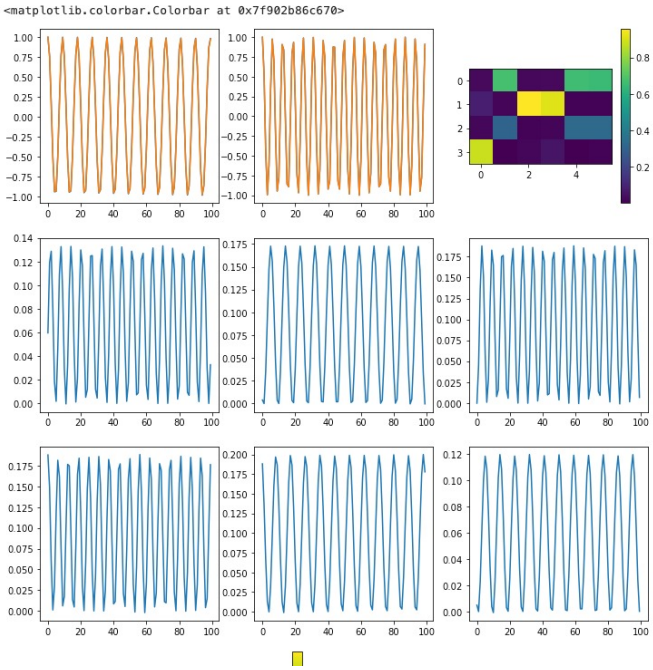

Figure 33: A linear RNN is trained to, from a fixed initial point, output two frequencies, shown in the top left two plots. The six neurons in the recurrent network do this, and with the biological constraints, modularise the way they do this if the freuqencies are non-harmonically related (as in this case). You can see three neurons for each frequency. On the top left you see the dot product of each neuron's activity onto cos (top two rows) and sine (bottom two) of each of the two frequencies, which clearly shows the modularisation.

there is some interesting data, though all the stuff I know was explained quite nicely by J. C. R. Whittington, Will Dorrell, et al. 2023. Nonetheless, as shown by the prefrontal cortex work, its a useful thing to have in mind to explore the presence of functional cell types; and, as always with science, some of the most interesting things happen when this is assumed knowledge and you can turn it on its head and use it to infer other things, like how the animal is thinking about a task from the way it is modularising.

Let's not get ahead ourselves though. On the road to matching data we need to develop the theory in a few ways. The first things to are thresh out the phase transitions in the simple system I described above, beyond that we have a few ideas.

This theory takes activity patterns and tells you how to arrange them in neural space, predicting fun modularity things along the way. This is obviously far from a complete theory, you need to understand activity patterns in the first place, either because the model is simple enough (linear data, or linear RNNs), or by other routes (grid cells).

But hope is not lost! Rather than developing a full theory of neural networks, we can piggyback on tthe progress of others! For example, consider a teacher-student framework, I imagine there are lots of settings where the known network patterns from the teacher end up being recapitulated in the network (due to the simplicity bias of the neural network, e.g. as shown in Cohen-Karlik et al. 2023). You could imagine asking an RNN to learn a function embedded in a teacher RNN. Since you know the activity patterns that have to go in, you can hope to predict how things will modularise! (Most interesting would be the, likely many, situations in which the teacher and student would disagree on their modularisation!)

Another case that could be amenable is in combination with Andrew's race reduction theory (A. Saxe, Sodhani, and Lewallen 2022; A. M. Saxe, McClelland, and Ganguli 2013). This predicts the similarity structure of modules of neurons. Perhaps we could combine this with activity regularisation etc. to get down to single neuron predictions, or at least lower level. And who knows, it might even predict additional modularisation.

Finally, I'd love to understand some nonlinear things - what kind of functional forms of modularity can nonlinear networks extract? There's two ways in which this might be tractable. The first, is to build on the gated linear network stuff, as done in the Saxe lab, or in the dendritic gated network work (Sezener et al. 2021; Veness et al. 2021). Perhaps in such a nice constrained form of non-linearity we can hope to make progress? The second option is to build on the

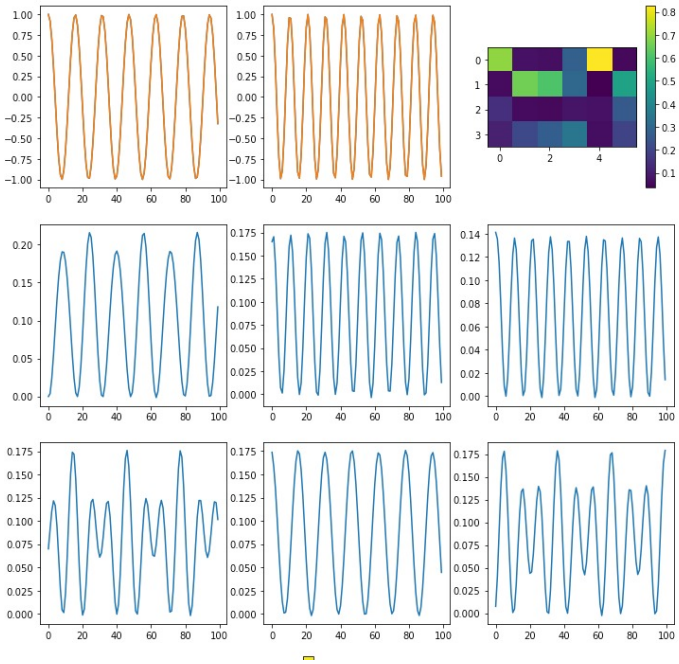

Figure 34: In this version, by contrast, the frequencies are harmonically related and the neurons don't perfectly modularise.

large body of work in the identifiability of nonlinear ICA world (these kinds of things: Hyvarinen, Khemakhem, and Morioka 2023; Hyvärinen, Khemakhem, and Monti 2023; Khemakhem et al. 2020; Kivva et al. 2022; Lachapelle and Lacoste-Julien 2022; Lachapelle, Rodriguez, et al. 2022; Y. Zheng, Ng, and Kun Zhang 2022).

One final pipe dream would be a combination of learning actionable codes and disentangling (which is attractively all linear). Can you look at data and unsupervisedly extract sets of variables like style and content by learning how they together impact an image? Bilinear models already seem good at this (Tenenbaum and Freeman 2000), and they might be the natural next step for linear analysis?

My last comment is on the super pleasing way these explorations make it clear what the definitions of the things were in the first place. It's super hard to define compositionality, task-similarity, factors of variation, or modularity. There are so many ways things can be compositional. But in these understandable settings we know what they mean! For a network of linear data and linear neurons minimising energy costs, factors are things that are range independent of one another, and influence the data in sufficiently orthogonal ways. For a linear RNN task-similarity can be defined in terms of range independence! And these normatively make reasonable sense, if things are not range independent then perhaps one is influencing the other, or there's some third underlying variable that is causing shifts in the other two. This suggests you shouldn't disentangle them because there's relationships between these variables! They should be linked in your representation as well! Formalising this line of thinking more seems like a fun and useful excercise.

5.3.4 Postscript on Disentangling for Interpretability

There's been some super interesting recent work on interpreting neural networks (Cammarata et al. 2020; Elhage et al. 2022; Goh et al. 2021; Olah, Cammarata, et al. 2020; Olah, Satyanarayan, et al. 2018). The relevant claim for our discussion is that, while many neurons in artificial neural networks are interpretable (perhaps disentangled?), many others are most parsimoniously understood as mixed-selective. But! The current hypothesis are that many neuron's responses can be understood as linear sums of hypothetical disentangled neuron responses - a hypothesis called superposition. To aid interpreting and understanding these networks we could usefully disentangle these responses, to understand what neurons are coding for!

I tried using the biological constraints to disentangle a toy model of superposition. In this model neurons are literally linear superpositions of features. The disentangling did a good job up to features that aligned with one another, predictably, which is kind of promising. I'm gonna chat to a few people who are into this kind of stuff and see if that is of interest.

6 Other Future Project Ideas

In this final section I list some other ideas I've been thinking about.

Dendrites as nonlinear processors for boring things: In the disentangling section we discussed how in nonlinear networks biological constraints appear to still help disentangling. While neurons in the latent layer become disentangled, those in earlier layers have to serve nonlinear processing roles. In fact, this seems to be the focus of the majority of neurons. However, given how costly firing is, and how much dendritic processing neurons can do, is there a way we might have got this the wrong way round? Could it be that the dendrites are for nonlinear processing, the neurons are for signalling of meaningful variables that you want to be communicated all around the brain. If this were true then the complexity of the dendritic true should be tuned to the computational complexity of the jump between the input and the output layers of complex inputs. I wonder how to test this?

Threshold as confidence, an idea from Trenton Bricken: Imagine a simple linear readout performing classification on some one-hidden-layer nonlinear representation of an input. This is a model of the fly mushroom body. In these models a large population of kenyon cells form the hidden layer, and are coupled to each other through an inhibitory interneuron. This interneuron enforces sparsity in the kenyon cells (roughly 5% of neurons are active at any one time) and can be thought of as a k-sparsifyer, applying a negative bias to all the neurons equal to the activity of the k-th most active neuron's input. Think of each kenyon cell as tiling some point in the input space. The denser the tiling the more neurons with the same activity when that input arrives, as such the smaller the activity of the kenyon cell after removing the k-th activity. In this way the density of the kenyon cells is reflected in the size of activity of the neurons. If the kenyon cells arrange their weights to reflect the distribution of inputs then this is a measure of confidence that scales with how often you see data. And since the activity scales the learning rate of the readout weights, this ensures you learn faster on more uncertain data - neat!

Bayesian Bees: Bees do this crazy waggle dance. Scouts reporting back with food come and do a figure of eight dance, the length of the waggle section of the figure of eight codes the distance from the hive to the food. The orientation of the dance relative to the honeycomb codes for the direction, figures 35 & 36. They use this coding not just for searching

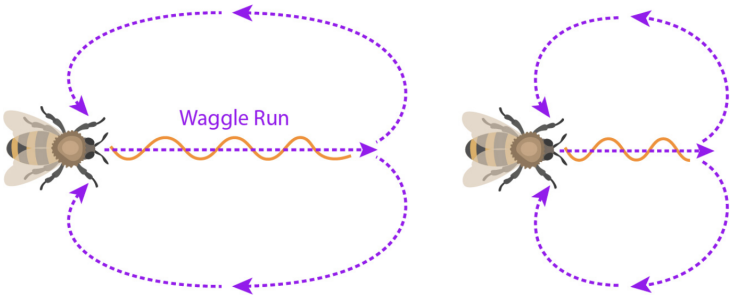

Figure 35: Distance coding by bee scouts

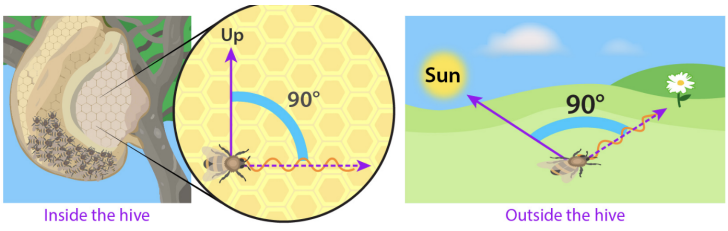

Figure 36: Orientation coding of bee scouts

for food, but also when choosing a new nest site. Scouts go out, evaluate the quality of a nest, then return and waggle its location to the others. Crucially, however, how long they perform the waggle dance is related to how good the nest site was. This leads to a compounding effect, if a site is good the dance is long, and many bees see it, go check out the site and come back and report. This gets independent verification of the original findings, and leads to a collective decision making effect, eventually all the bees decide that one site is the best, and the swarm moves! (Seeley 2011)

Interestingly, however, the same bee goes to one nest site multiple times, and when it returns it will waggle with lengths of time with a very high variance, including investing all that effort and then returning and not even waggling, despite the fact the last time it came back and waggled loads!

Perhaps, like synaptic failures (Aitchison et al. 2021), it is coding for confidence? You could test this by getting nest sites where evaluating how good it is either easy or hard (in the dark vs light, or they know the algorithm bees use to estimate volume of a cavity, so it could be sites that vary in how easy it is to estimate their volume), then see if the spread of bee waggle times goes up or down? I emailed Tom Seeley about this but unfortunately he seems to be ill.

Learning Lie group structure: Lie algebras are vector spaces that have to satisfy a particular equation, the Jacobi identity. The set of lie algebras for a particular dimensionality can be parameterised. Perhaps this parameterisation can then be optimised over to try and find the lie algebra that best fits a particular dataset?

Which version of the CAN is in the brain? There are at least two continuous attractor network models out there, though one of them has come out the clear winner in the case of flies. I think Charlotte Bocarra has some data that could distinguish which model is at work in the entorhinal cortex, and offered to host me to analyse them at one point (I think...?). Sounds like it could be fun?

References

Ackels, Tobias et al. (2021). "Fast odour dynamics are encoded in the olfactory system and guide behaviour". In: *Nature* 593.7860, pp. 558–563.

Ahmed, Maria et al. (2023). "Hacking brain development to test models of sensory coding". In: *bioRxiv*, pp. 2023–01.

Aitchison, Laurence et al. (2021). "Synaptic plasticity as Bayesian inference". In: *Nature neuroscience* 24.4, pp. 565–571.

Al Roumi, Fosca et al. (2020). "An abstract language of thought for spatial sequences in humans". In: *bioRxiv*.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning*.

Asaad, Wael F, Gregor Rainer, and Earl K Miller (2000). "Task-specific neural activity in the primate prefrontal cortex". In: *Journal of neurophysiology* 84.1, pp. 451–459.

Aso, Yoshinori et al. (2014). "The neuronal architecture of the mushroom body provides a logic for associative learning". In: *elife* 3, e04577.

Attneave, Fred (1954). "Some informational aspects of visual perception." In: *Psychological review* 61.3, p. 183.

Baevski, Alexei, Steffen Schneider, and Michael Auli (2019). "vq-wav2vec: Self-supervised learning of discrete speech representations". In: *arXiv preprint arXiv:1910.05453*.

Bahri, Yasaman et al. (2021). "Explaining neural scaling laws". In: *arXiv preprint arXiv:2102.06701*.

Bajo, Victoria M et al. (2010). "The descending corticocollicular pathway mediates learning-induced auditory plasticity". In: *Nature neuroscience* 13.2, pp. 253–260.

Ballé, Johannes, Valero Laparra, and Eero P Simoncelli (2016). "End-to-end optimized image compression". In: *arXiv preprint arXiv:1611.01704*.

Banijamali, Ershad et al. (2018). "Robust locally-linear controllable embedding". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1751–1759.

Banino, Andrea et al. (2018). "Vector-based navigation using grid-like representations in artificial agents". In: *Nature* 557.7705, pp. 429–433.

Bao, Pinglei et al. (July 2020). "A map of object space in primate inferotemporal cortex". In: *Nature* 583.7814, pp. 103–108. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2350-5. URL: https://doi.org/10.1038/s41586-020-2350-5.

Barlow, Horace B et al. (1961). "Possible principles underlying the transformation of sensory messages". In: *Sensory communication* 1.01.

Barry, C. et al. (2012). "Grid cell firing patterns signal environmental novelty by expansion". In: *Proceedings of the National Academy of Sciences* 109.43. ISBN: 1091-6490 (Electronic)$\backslash$n0027-8424 (Linking) _eprint: arXiv:1408.1149, pp. 17687–17692. ISSN: 0027-8424. DOI: 10.1073/pnas.1209918109. URL: http://www.pnas.org/cgi/doi/10.1073/pnas.1209918109.

Basu, Raunak et al. (2021). "The orbitofrontal cortex maps future navigational goals". In: *Nature* 599.7885, pp. 449–452.

Bengio, Yoshua (2013). "Deep learning of representations: Looking forward". In: *Statistical Language and Speech Processing: First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013. Proceedings 1*. Springer, pp. 1–37.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.

Bengio, Yoshua, Nicholas Léonard, and Aaron Courville (2013). "Estimating or propagating gradients through stochastic neurons for conditional computation". In: *arXiv preprint arXiv:1308.3432*.

Bernardi, Silvia et al. (Nov. 2020). "The Geometry of Abstraction in the Hippocampus and Prefrontal Cortex". In: *Cell* 183.4, 954–967.e21. ISSN: 00928674. DOI: 10.1016/j.cell.2020.09.031. URL: https://linkinghub.elsevier.com/retrieve/pii/S0092867420312289.

Boccara, Charlotte N et al. (2019). "The entorhinal cognitive map is attracted to goals". In: *Science* 363.6434, pp. 1443–1447.

Bordelon, Blake, Abdulkadir Canatar, and Cengiz Pehlevan (2020). "Spectrum dependent learning curves in kernel regression and wide neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 1024–1034.

Bordelon, Blake and Cengiz Pehlevan (2021). "Population codes enable learning from few examples by shaping inductive bias". In: *BioRxiv*.

Borsos, Zalán et al. (2022). "Audiolm: a language modeling approach to audio generation". In: *arXiv preprint arXiv:2209.03143*.

Bronstein, Michael M et al. (2021). "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges". In: *arXiv preprint arXiv:2104.13478*.

Brunton, Steven L et al. (2021). "Modern Koopman theory for dynamical systems". In: *arXiv preprint arXiv:2102.12086*.

Burak, Yoram and Ila R Fiete (2009). "Accurate path integration in continuous attractor network models of grid cells". In: *PLoS computational biology* 5.2, e1000291.

Burgess, Chris and Hyunjik Kim (2018). *3D Shapes Dataset*. https://github.com/deepmind/3dshapes-dataset/.
Butler, William N., Kiah Hardcastle, and Lisa M. Giocomo (Mar. 2019). "Remembered reward locations restructure entorhinal spatial maps". In: *Science* 363.6434, pp. 1447–1452. ISSN: 0036-8075. DOI: 10.1126/science.aav5297. URL: http://www.sciencemag.org/lookup/doi/10.1126/science.aav5297.
Cammarata, Nick et al. (2020). "Thread: circuits". In: *Distill* 5.3, e24.
Canatar, Abdulkadir, Blake Bordelon, and Cengiz Pehlevan (2021). "Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks". In: *Nature communications* 12.1, pp. 1–12.
Carbonneau, Marc-André et al. (2022). "Measuring disentanglement: A review of metrics". In: *IEEE Transactions on Neural Networks and Learning Systems*.
Caselles-Dupré, Hugo, Michael Garcia Ortiz, and David Filliat (2019). "Symmetry-based disentangled representation learning requires interaction with environments". In: *Advances in Neural Information Processing Systems* 32.
Chang, Le and Doris Y. Tsao (June 2017). "The Code for Facial Identity in the Primate Brain". In: *Cell* 169.6. ISBN: 0092-8674 Publisher: Elsevier Inc., 1013–1028.e14. ISSN: 10974172. DOI: 10.1016/j.cell.2017.05.011. URL: http://dx.doi.org/10.1016/j.cell.2017.05.011.
Chen, Ricky TQ et al. (2018). "Isolating sources of disentanglement in variational autoencoders". In: *Advances in neural information processing systems* 31.
Chen, Xi et al. (2016). "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in neural information processing systems* 29.
Cho, Youngmin and Lawrence Saul (2009). "Kernel methods for deep learning". In: *Advances in neural information processing systems* 22.
Cohen-Karlik, Edo et al. (2023). "Learning Low Dimensional State Spaces with Overparameterized Recurrent Neural Nets". In: *The Eleventh International Conference on Learning Representations*.
Comon, Pierre (1994). "Independent component analysis, a new concept?" In: *Signal processing* 36.3, pp. 287–314.
Creager, Elliot et al. (2019). "Flexibly fair representation learning by disentanglement". In: *International conference on machine learning*. PMLR, pp. 1436–1445.
Cueva, Christopher J, Adel Ardalan, et al. (2021). "Recurrent neural network models for working memory of continuous variables: activity manifolds, connectivity patterns, and dynamic codes". In: *arXiv preprint arXiv:2111.01275*.
Cueva, Christopher J and Xue-Xin Wei (2018). "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization". In: *arXiv preprint arXiv:1803.07770*.
Dao, Tri et al. (2022). "Hungry Hungry Hippos: Towards Language Modeling with State Space Models". In: *arXiv preprint arXiv:2212.14052*.
Dehaene, Stanislas et al. (2022). "Symbols and mental programs: a hypothesis about human singularity". In: *Trends in Cognitive Sciences*.
Deshmukh, Sachin S and James J Knierim (Apr. 2013). "Influence of local objects on hippocampal representations: Landmark vectors and memory." In: *Hippocampus* 23.4, pp. 253–67. ISSN: 1098-1063. DOI: 10.1002/hipo.22101. URL: http://www.ncbi.nlm.nih.gov/pubmed/23447419.
Dhariwal, Prafulla et al. (2020). "Jukebox: A generative model for music". In: *arXiv preprint arXiv:2005.00341*.
Ding, Zhiwei et al. (2023). "Bipartite invariance in mouse primary visual cortex". In: *bioRxiv*.
Dordek, Yedidyah et al. (2016). "Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis". In: *Elife* 5, e10094.
Dorrell, Will et al. (2023). "Actionable Neural Representations: Grid Cells from Minimal Constraints". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=xfqDe72zh41.
Dorrell, William, Maria Yuffa, and Peter Latham (2023). "Meta-Learning the Inductive Biases of Simple Neural Circuits". In: *ICML*.
Driscoll, Laura, Krishna Shenoy, and David Sussillo (2022). "Flexible multitask computation in recurrent networks utilizes shared dynamical motifs". In: *bioRxiv*, pp. 2022–08.
Du, Mengnan, Ninghao Liu, and Xia Hu (2019). "Techniques for interpretable machine learning". In: *Communications of the ACM* 63.1, pp. 68–77.
Dubreuil, Alexis et al. (2022). "The role of population structure in computations through neural dynamics". In: *Nature neuroscience* 25.6, pp. 783–794.
Dunn, Benjamin et al. (2017). "Grid cells show field-to-field variability and this explains the aperiodic response of inhibitory interneurons". In: *arXiv preprint arXiv:1701.04893*.
Eastwood, Cian and Christopher KI Williams (2018). "A framework for the quantitative evaluation of disentangled representations". In: *International Conference on Learning Representations*.
Ecker, Alexander S. et al. (Jan. 2010). "Decorrelated Neuronal Firing in Cortical Microcircuits". In: *Science* 327.5965. Publisher: American Association for the Advancement of Science, pp. 584–587. DOI: 10.1126/science.1179867. URL: https://www.science.org/doi/abs/10.1126/science.1179867 (visited on 08/18/2022).
Elhage, Nelson et al. (2022). "Toy Models of Superposition". In: *arXiv preprint arXiv:2209.10652*.

Ellis, Kaitlyn Elizabeth et al. (2023). "Evolution of connectivity architecture in the Drosophila mushroom body". In: *bioRxiv*, pp. 2023–02.
Ellis, Kevin et al. (2020). "Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning". In: *arXiv preprint arXiv:2006.08381*.
Finkelstein, Arseny et al. (2015). "Three-dimensional head-direction coding in the bat brain". In: *Nature* 517.7533, pp. 159–164.
Flesch, Timo et al. (2022). "Orthogonal representations for robust context-dependent task performance in brains and neural networks". In: *Neuron* 110.7. Publisher: The Authors, 1258–1270.e11. ISSN: 08966273. DOI: 10.1016/j.neuron.2022.01.005. URL: https://doi.org/10.1016/j.neuron.2022.01.005.
Gao, Ruiqi, Jianwen Xie, Xue-Xin Wei, et al. (2021). "On Path Integration of grid cells: isotropic metric, conformal embedding and group representation". In: *Advances in neural information processing systems* 34.
Gao, Ruiqi, Jianwen Xie, Song-Chun Zhu, et al. (2018). "Learning grid cells as vector representation of self-position coupled with matrix representation of self-motion". In: *arXiv preprint arXiv:1810.05597*.
Gao, Weihao, Sewoong Oh, and Pramod Viswanath (2018). "Demystifying fixed k-nearest neighbor information estimators". In: *IEEE Transactions on Information Theory* 64.8, pp. 5629–5661.
Gardner, Richard J et al. (2022). "Toroidal topology of population activity in grid cells". In: *Nature* 602.7895, pp. 123–128.
Gauthier, Jeffrey L. and David W. Tank (2018). "A Dedicated Population for Reward Coding in the Hippocampus". In: *Neuron* 99.1. Publisher: Elsevier Inc., 179–193.e7. ISSN: 10974199. DOI: 10.1016/j.neuron.2018.06.008. URL: https://doi.org/10.1016/j.neuron.2018.06.008.
Ginosar, Gily et al. (2021). "Locally ordered representation of 3D space in the entorhinal cortex". In: *Nature* 596.7872, pp. 404–409.
Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 249–256.
Goh, Gabriel et al. (2021). "Multimodal neurons in artificial neural networks". In: *Distill* 6.3, e30.
Gondal, Muhammad Waleed et al. (2019). "On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/d97d404b6119214e4a7018391195240a-Paper.pdf.
Goodfellow, Ian et al. (2014). "Generative adversarial networks". In: *Neural Information Processing Systems (NeurIPS)*.
Goodman, Dan et al. (2022). *Spiking Neural Network Models in Neuroscience - Cosyne Tutorial 2022*. https://zenodo.org/record/7044500#.Yy7crezML9s.
Gresele, Luigi et al. (2021). "Independent mechanism analysis, a new concept?" In: *Advances in neural information processing systems* 34, pp. 28233–28248.
Grieves, Roddy M et al. (2021). "Irregular distribution of grid cell firing fields in rats exploring a 3D volumetric space". In: *Nature neuroscience* 24.11, pp. 1567–1573.
Gulrajani, Ishaan et al. (2017). "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30.
Gunasekar, Suriya et al. (2017). "Implicit regularization in matrix factorization". In: *Advances in Neural Information Processing Systems* 30.
Hafting, Torkel et al. (2005). "Microstructure of a spatial map in the entorhinal cortex". In: *Nature* 436.7052, pp. 801–806.
Hansen-Estruch, Philippe et al. (2021). "Gem: Group enhanced model for learning dynamical control systems". In: *arXiv preprint arXiv:2104.02844*.
Hardcastle, Kiah et al. (2017). "A multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex". In: *Neuron* 94.2, pp. 375–387.
Hardt, Moritz, Ben Recht, and Yoram Singer (2016). "Train faster, generalize better: Stability of stochastic gradient descent". In: *International conference on machine learning*. PMLR, pp. 1225–1234.
Harris, Julia J, Renaud Jolivet, and David Attwell (2012). "Synaptic energy use and supply". In: *Neuron* 75.5, pp. 762–777.
Harris, Kameron Decker (2019). "Additive function approximation in the brain". In: *arXiv preprint arXiv:1909.02603*.
Hayashi, Tatsuya Tatz et al. (2022). "Mushroom body input connections form independently of sensory activity in Drosophila melanogaster". In: *Current Biology* 32.18, pp. 4000–4012.
Hayman, Robin MA et al. (2015). "Grid cells on steeply sloping terrain: evidence for planar rather than volumetric encoding". In: *Frontiers in psychology* 6, p. 925.
Hige, Toshihide (2018). "What can tiny mushrooms in fruit flies tell us about learning and memory?" In: *Neuroscience Research* 129, pp. 8–16.

Higgins, Irina, Le Chang, et al. (2021). "Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons". In: *Nature Communications* 12.1, pp. 1–14. ISSN: 20411723. DOI: 10.1038/s41467-021-26751-5.

Higgins, Irina, Loic Matthey, et al. (2017). "beta-vae: Learning basic visual concepts with a constrained variational framework". In: *International conference on learning representations*.

Higgins, Irina, Arka Pal, et al. (2017). "DARLA: Improving Zero-Shot Transfer in Reinforcement Learning". In: *arXiv preprint*. ISBN: 1748-2623 _eprint: 1707.08475. ISSN: 1938-7228. DOI: 10.3109/17482620903223036. URL: http://arxiv.org/abs/1707.08475.

Higgins, Irina, Nicolas Sonnerat, et al. (2018). "SCAN: Learning Hierarchical Compositional Visual Concepts". In: *arXiv preprint*. ISBN: 978-1-4503-1246-2 _eprint: 1707.03389, pp. 1–24. ISSN: 1346-9843. DOI: 10.1186/s12884-017-1520-4. URL: http://arxiv.org/abs/1707.03389.

Hinton, Geoffrey E., Alex Krizhevsky, and Sida D. Wang (2011). "Transforming Auto-Encoders". en. In: *International Conference on Artificial Neural Networks* 6791, pp. 44–51. DOI: 10.1007/978-3-642-21735-7_6. URL: http://link.springer.com/10.1007/978-3-642-21735-7_6 (visited on 08/21/2022).

Hiratani, Naoki and Peter E Latham (2020). "Rapid Bayesian learning in the mammalian olfactory system". In: *Nature communications* 11.1, p. 3845.

– (2022). "Developmental and evolutionary constraints on olfactory circuit selection". In: *Proceedings of the National Academy of Sciences* 119.11, e2100600119.

Hiratani, Naoki and Haim Sompolinsky (2023). "Optimal quadratic binding for relational reasoning in vector symbolic neural architectures". In: *Neural Computation* 35.2, pp. 105–155.

Hirokawa, Junya et al. (Dec. 2019). "Frontal cortex neuron types categorically encode single decision variables". en. In: *Nature* 576.7787. Number: 7787 Publisher: Nature Publishing Group, pp. 446–451. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1816-9. URL: https://www.nature.com/articles/s41586-019-1816-9 (visited on 07/21/2022).

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

Høydal, Øyvind Arne et al. (2019). "Object-vector coding in the medial entorhinal cortex". In: *Nature* 568.7752, pp. 400–404.

Huang, Xiaowei et al. (2020). "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability". In: *Computer Science Review* 37, p. 100270.

Huang, Xun and Serge Belongie (2017). "Arbitrary style transfer in real-time with adaptive instance normalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510.

Huang, Yanping and Rajesh PN Rao (2011). "Predictive coding". In: *Wiley Interdisciplinary Reviews: Cognitive Science* 2.5, pp. 580–593.

Hume, David (1748). *An enquiry concerning human understanding and other writings*.

Hyvarinen, Aapo, Ilyes Khemakhem, and Hiroshi Morioka (2023). "Nonlinear Independent Component Analysis for Principled Disentanglement in Unsupervised Deep Learning". In: *arXiv preprint arXiv:2303.16535*.

Hyvärinen, Aapo (2010). "Statistical models of natural images and cortical visual representation". In: *Topics in Cognitive Science* 2.2, pp. 251–264.

Hyvärinen, Aapo, Patrik O Hoyer, and Mika Inki (2001). "Topographic independent component analysis". In: *Neural computation* 13.7, pp. 1527–1558.

Hyvärinen, Aapo, Ilyes Khemakhem, and Ricardo Monti (2023). "Identifiability of latent-variable and structural-equation models: from linear to nonlinear". In: *arXiv preprint arXiv:2302.02672*.

Hyvärinen, Aapo and Erkki Oja (2000). "Independent component analysis: algorithms and applications". In: *Neural networks* 13.4-5, pp. 411–430.

Hyvärinen, Aapo and Petteri Pajunen (1999). "Nonlinear independent component analysis: Existence and uniqueness results". In: *Neural networks* 12.3, pp. 429–439.

Issa, John B and Kechen Zhang (2012). "Universal conditions for exact path integration in neural systems". In: *Proceedings of the National Academy of Sciences* 109.17, pp. 6716–6720.

Kaiser, Lukasz et al. (2018). "Fast decoding in sequence models using discrete latent variables". In: *International Conference on Machine Learning*. PMLR, pp. 2390–2399.

Kandel, Eric R (2007). *In search of memory: The emergence of a new science of mind*. WW Norton & Company.

Kang, Yul HR, Daniel M Wolpert, and Máté Lengyel (2023). "Spatial uncertainty and environmental geometry in navigation". In: *bioRxiv*, pp. 2023–01.

Karras, Tero, Samuli Laine, and Timo Aila (2019). "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410.

Kawato, Mitsuo, Shogo Ohmae, et al. (2021). "50 years since the Marr, Ito, and Albus models of the cerebellum". In: *Neuroscience* 462, pp. 151–174.

Kawato, Mitsuo and Daniel Wolpert (2007). "Internal models for motor control". In: *Novartis Foundation Symposium 218-Sensory Guidance of Movement: Sensory Guidance of Movement: Novartis Foundation Symposium 218*. Wiley Online Library, pp. 291–307.
Kay, Leslie M and Gilles Laurent (1999). "Odor-and context-dependent modulation of mitral cell activity in behaving rats". In: *Nature neuroscience* 2.11, pp. 1003–1009.
Keller, Georg B and Thomas D Mrsic-Flogel (2018). "Predictive processing: a canonical cortical computation". In: *Neuron* 100.2, pp. 424–435.
Keurti, Hamza et al. (2022). "Homomorphism Autoencoder–Learning Group Structured Representations from Observed Transitions". In: *arXiv preprint arXiv:2207.12067*.
Khemakhem, Ilyes et al. (2020). "Variational autoencoders and nonlinear ica: A unifying framework". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2207–2217.
Kim, Hyunjik and Andriy Mnih (July 2018). "Disentangling by Factorising". en. In: *Proceedings of the 35th International Conference on Machine Learning*, pp. 2649–2658. ISSN: 2640-3498. URL: https://proceedings.mlr.press/v80/kim18b.html (visited on 08/21/2022).
Kim, Sung Soo et al. (2017). "Ring attractor dynamics in the Drosophila central brain". In: *Science* 356.6340, pp. 849–853.
King, Andrew J et al. (2011). "Neural circuits underlying adaptation and learning in the perception of auditory space". In: *Neuroscience & Biobehavioral Reviews* 35.10, pp. 2129–2139.
Kivva, Bohdan et al. (2022). "Identifiability of deep generative models without auxiliary information". In: *Advances in Neural Information Processing Systems* 35, pp. 15687–15701.
Kleyko, Denis et al. (2022). "Vector symbolic architectures as a computing framework for emerging hardware". In: *Proceedings of the IEEE* 110.10, pp. 1538–1571.
Klindt, David et al. (2020). "Towards nonlinear disentanglement in natural data with temporal sparse coding". In: *arXiv preprint arXiv:2007.10930*.
Kobayashi, Kazuma et al. (2021). "Decomposing normal and abnormal features of medical images for content-based image retrieval of glioma imaging". In: *Medical image analysis* 74, p. 102227.
Koldaeva, Anzhelika, Andreas T Schaefer, and Izumi Fukunaga (2019). "Rapid task-dependent tuning of the mouse olfactory bulb". In: *Elife* 8, e43558.
Komorowski, Robert W., Joseph R. Manns, and Howard Eichenbaum (Aug. 2009). "Robust Conjunctive Item-Place Coding by Hippocampal Neurons Parallels Learning What Happens Where". In: *Journal of Neuroscience* 29.31. ISBN: 1529-2401 (Electronic), pp. 9918–9929. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.1378-09.2009. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.1378-09.2009.
Kraskov, Alexander, Harald Stögbauer, and Peter Grassberger (2004). "Estimating mutual information". In: *Physical review E* 69.6, p. 066138.
Krupic, Julija et al. (2015). "Grid cell symmetry is shaped by environmental geometry". In: *Nature* 518.7538, pp. 232–235.
Kudryavitskaya, Elena et al. (2021). "Flexible categorization in the mouse olfactory bulb". In: *Current Biology* 31.8, pp. 1616–1631.
Kurth-Nelson, Zeb et al. (2023). "Replay and compositional computation". In: *Neuron*.
Lachapelle, Sébastien and Simon Lacoste-Julien (2022). "Partial disentanglement via mechanism sparsity". In: *arXiv preprint arXiv:2207.07732*.
Lachapelle, Sébastien, Pau Rodriguez, et al. (2022). "Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA". In: *Conference on Causal Learning and Reasoning*. PMLR, pp. 428–484.
Lange, Robert Tjarko et al. (2022). "Discovering Evolution Strategies via Meta-Black-Box Optimization". In: *arXiv preprint arXiv:2211.11260*.
Le Merre, Pierre, Sofie Ährlund-Richter, and Marie Carlén (2021). "The mouse prefrontal cortex: Unity in diversity". In: *Neuron* 109.12, pp. 1925–1944.
LeCun, Yann (1998). "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/*.
LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.
Lee, Julie J. et al. (Aug. 2022). "Task specificity in mouse parietal cortex". en. In: *Neuron*. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2022.07.017. URL: https://www.sciencedirect.com/science/article/pii/S0896627322006626 (visited on 08/17/2022).
Leeb, Felix et al. (2023). "Structure by Architecture: Structured Representations without Regularization". In: *International Conference on Learning Representations*.
Lertvittayakumjorn, Piyawat and Francesca Toni (2021). "Explanation-based human debugging of nlp models: A survey". In: *Transactions of the Association for Computational Linguistics* 9, pp. 1508–1528.
Lever, Colin et al. (2009). "Boundary vector cells in the subiculum of the hippocampal formation". In: *Journal of Neuroscience* 29.31, pp. 9771–9777. ISSN: 02706474. DOI: 10.1523/JNEUROSCI.1319-09.2009.

Li, Zhiyuan et al. (2020). "Progressive learning and disentanglement of hierarchical representations". In: *arXiv preprint arXiv:2002.10549*.
Lin, Andrew C et al. (2014). "Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination". In: *Nature neuroscience* 17.4, pp. 559–568.
Linderman, Scott et al. (2017). "Bayesian learning and inference in recurrent switching linear dynamical systems". In: *Artificial Intelligence and Statistics*. PMLR, pp. 914–922.
Litwin-Kumar, Ashok and Srinivas C Turaga (2019). "Constraining computational models using electron microscopy wiring diagrams". In: *Current opinion in neurobiology* 58, pp. 94–100.
Locatello, Francesco et al. (2019). "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *international conference on machine learning*. PMLR, pp. 4114–4124.
Logiaco, Laureline, LF Abbott, and Sean Escola (2021). "Thalamic control of cortical dynamics in a model of flexible motor sequencing". In: *Cell reports* 35.9, p. 109090.
Logothetis, Nikos K, Jon Pauls, and Tomaso Poggio (1995). "Shape representation in the inferior temporal cortex of monkeys". In: *Current biology* 5.5, pp. 552–563.
Lu, Zepu et al. (2023). "Differentiable Optimized Product Quantization and Beyond". In: *Proceedings of the ACM Web Conference 2023*, pp. 3353–3363.
Lutz, Anthony Erich (2021). *Learning Grid Cells and Remapping in Curved Space: A Gauge Theoretic Perspective*. University of California, Los Angeles.
Lyu, Cheng, LF Abbott, and Gaby Maimon (2022). "Building an allocentric travelling direction signal via vector computation". In: *Nature* 601.7891, pp. 92–97.
Mairal, Julien and Jean-Philippe Vert (2018). "Machine learning with kernel methods". In: *Lecture Notes, January* 10.
Mandt, Stephan, Matthew D Hoffman, and David M Blei (2017). "Stochastic gradient descent as approximate bayesian inference". In: *arXiv preprint arXiv:1704.04289*.
Marimont, Sergio Naval and Giacomo Tarroni (2021). "Anomaly detection through latent space restoration using vector quantized variational autoencoders". In: *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 1764–1767.
Mastrogiuseppe, Francesca, Naoki Hiratani, and Peter Latham (2023). "Evolution of neural activity in circuits bridging sensory and abstract knowledge". In: *Elife* 12, e79908.
Mastrogiuseppe, Francesca and Srdjan Ostojic (2018). "Linking connectivity, dynamics, and computations in low-rank recurrent neural networks". In: *Neuron* 99.3, pp. 609–623.
Mathis, Alexander, Andreas VM Herz, and Martin Stemmler (2012). "Optimal population codes for space: grid cells outperform place cells". In: *Neural computation* 24.9, pp. 2280–2317.
Mathis, Alexander, Andreas VM Herz, and Martin B Stemmler (2012). "Resolution of nested neuronal representations can be exponential in the number of neurons". In: *Physical review letters* 109.1, p. 018103.
Maynord, Michael et al. (2023). "Mid-Vision Feedback". In: *The Eleventh International Conference on Learning Representations*.
McClelland, James L, Bruce L McNaughton, and Randall C O'Reilly (1995). "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory." In: *Psychological review* 102.3, p. 419.
McIntosh, Lane et al. (2016). "Deep learning models of the retinal response to natural scenes". In: *Advances in neural information processing systems* 29.
Miller, Earl K and Jonathan D Cohen (2001). "An integrative theory of prefrontal cortex function". In: *Annual review of neuroscience* 24.1, pp. 167–202.
Mittelstadt, Brent (2016). "Automation, algorithms, and politics| auditing for transparency in content personalization systems". In: *International Journal of Communication* 10, p. 12.
Moran, Gemma E et al. (2021). "Identifiable deep generative models via sparse decoding". In: *arXiv preprint arXiv:2110.10804*.
Mussells Pires, Peter, LF Abbott, and Gaby Maimon (2022). "Converting an allocentric goal into an egocentric steering signal". In: *bioRxiv*, pp. 2022–11.
Nanda, Neel et al. (2023). "Progress measures for grokking via mechanistic interpretability". In: *arXiv preprint arXiv:2301.05217*.
Neftci, Emre O, Hesham Mostafa, and Friedemann Zenke (2019). "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks". In: *IEEE Signal Processing Magazine* 36.6, pp. 51–63.
Nguyen, Tri M et al. (2023). "Structured cerebellar connectivity supports resilient pattern separation". In: *Nature* 613.7944, pp. 543–549.
Nie, Weili (2019). *High resolution disentanglement datasets*. https://github.com/NVlabs/High-res-disentanglement-datasets.

Nie, Weili et al. (2020). "Semi-supervised stylegan for disentanglement learning". In: *Proceedings of the 37th International Conference on Machine Learning*, pp. 7360–7369.
Niven, Jeremy E, John C Anderson, and Simon B Laughlin (2007). "Fly photoreceptors demonstrate energy-information trade-offs in neural coding". In: *PLoS biology* 5.4, e116.
O'Keefe, John and J. Dostrovsky (Nov. 1971). "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat". In: *Brain Research* 34.1. ISBN: 0006-8993 (Print)$\backslash$r0006-8993 (Linking), pp. 171–175. ISSN: 00068993. DOI: 10.1016/0006-8993(71)90358-1. URL: http://linkinghub.elsevier.com/retrieve/pii/0006899371903581.
Ocko, Samuel A et al. (2018). "Emergent elasticity in the neural code for space". In: *Proceedings of the National Academy of Sciences* 115.50, E11798–E11806.
Okray, Zeynep et al. (2022). "Multisensory learning binds modality-specific neurons into a cross-modal memory engram". In: *bioRxiv*, pp. 2022–07.
Ólafsdóttir, H Freyja, Francis Carpenter, and Caswell Barry (2016). "Coordinated grid and place cell replay during rest". In: *Nature neuroscience* 19.6, pp. 792–794.
Olah, Chris, Nick Cammarata, et al. (2020). "Zoom in: An introduction to circuits". In: *Distill* 5.3, e00024–001.
Olah, Chris, Arvind Satyanarayan, et al. (2018). "The building blocks of interpretability". In: *Distill* 3.3, e10.
Olshausen, Bruno A and David J Field (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583, pp. 607–609.
Olsson, Catherine et al. (2022). "In-context learning and induction heads". In: *arXiv preprint arXiv:2209.11895*.
Otto, Samuel E and Clarence W Rowley (2021). "Koopman operators for estimation and control of dynamical systems". In: *Annual Review of Control, Robotics, and Autonomous Systems* 4, pp. 59–87.
Paccanaro, Alberto and Geoffrey E Hinton (2001). "Learning hierarchical structures with linear relational embedding". In: *Advances in neural information processing systems* 14.
Pandey, Biraj et al. (2021). "Structured random receptive fields enable informative sensory encodings". In: *bioRxiv*.
Pehlevan, Cengiz and Dmitri B Chklovskii (2015). "Optimization theory of Hebbian/anti-Hebbian networks for PCA and whitening". In: *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, pp. 1458–1465.
– (2019). "Neuroscience-inspired online unsupervised learning algorithms: Artificial neural networks". In: *IEEE Signal Processing Magazine* 36.6, pp. 88–96.
Perez, Ethan et al. (2018). "Film: Visual reasoning with a general conditioning layer". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.
Quessard, Robin, Thomas D Barrett, and William R Clements (2020). "Learning group structure and disentangled representations of dynamical environments". In: *arXiv preprint arXiv:2002.06991*.
Quiroga, R. Quian et al. (June 2005). "Invariant visual representation by single neurons in the human brain". en. In: *Nature* 435.7045. Number: 7045 Publisher: Nature Publishing Group, pp. 1102–1107. ISSN: 1476-4687. DOI: 10.1038/nature03687. URL: https://www.nature.com/articles/nature03687 (visited on 09/07/2022).
Rainer, Gregor, Wael F. Asaad, and Earl K. Miller (June 1998). "Selective representation of relevant information by neurons in the primate prefrontal cortex". en. In: *Nature* 393.6685. Number: 6685 Publisher: Nature Publishing Group, pp. 577–579. ISSN: 1476-4687. DOI: 10.1038/31235. URL: https://www.nature.com/articles/31235 (visited on 09/10/2022).
Ramesh, Aditya et al. (2021). "Zero-shot text-to-image generation". In: *International Conference on Machine Learning*. PMLR, pp. 8821–8831.
Richards, Blake A. et al. (Nov. 2019). "A deep learning framework for neuroscience". en. In: *Nature Neuroscience* 22.11. Number: 11 Publisher: Nature Publishing Group, pp. 1761–1770. ISSN: 1546-1726. DOI: 10.1038/s41593-019-0520-2. URL: https://www.nature.com/articles/s41593-019-0520-2 (visited on 09/01/2022).
Ridgeway, Karl (2016). "A survey of inductive biases for factorial representation-learning". In: *arXiv preprint arXiv:1612.05299*.
Ridgeway, Karl and Michael C Mozer (2018). "Learning deep disentangled embeddings with the f-statistic loss". In: *Advances in neural information processing systems* 31.
Rigotti, Mattia et al. (2013). "The importance of mixed selectivity in complex cognitive tasks". In: *Nature* 497.7451. ISBN: doi:10.1038/nature12160 Publisher: Nature Publishing Group, pp. 1–6. ISSN: 1476-4687. DOI: 10.1038/nature12160. URL: http://dx.doi.org/10.1038/nature12160.
Rombach, Robin et al. (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695.
Ross, Brian C (2014). "Mutual information between discrete and continuous data sets". In: *PloS one* 9.2, e87357.
Roth, Karsten et al. (2022). "Disentanglement of Correlated Factors via Hausdorff Factorized Support". In: *arXiv preprint arXiv:2210.07347*.
Roy, Jefferson E. et al. (June 2010). "Prefrontal Cortex Activity during Flexible Categorization". en. In: *Journal of Neuroscience* 30.25. Publisher: Society for Neuroscience Section: Articles, pp. 8519–8528. ISSN: 0270-6474, 1529-

2401. DOI: 10.1523/JNEUROSCI.4837-09.2010. URL: https://www.jneurosci.org/content/30/25/8519 (visited on 09/10/2022).
Saanum, Tankred and Eric Schulz (2022). "Learning Parsimonious Dynamics for Generalization in Reinforcement Learning". In: *arXiv preprint arXiv:2209.14781*.
Sablé-Meyer, Mathias et al. (2022). "A language of thought for the mental representation of geometric shapes". In: *Cognitive Psychology* 139, p. 101527.
Sarel, Ayelet et al. (2017). "Vectorial representation of spatial goals in the hippocampus of bats". In: *Science* 355.6321, pp. 176–180. ISSN: 10959203. DOI: 10.1126/science.aak9589.
Saxe, Andrew, Shagun Sodhani, and Sam Jay Lewallen (2022). "The neural race reduction: dynamics of abstraction in gated networks". In: *International Conference on Machine Learning*. PMLR, pp. 19287–19309.
Saxe, Andrew M, James L McClelland, and Surya Ganguli (2013). "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks". In: *arXiv preprint arXiv:1312.6120*.
Schaeffer, Rylan, Mikail Khona, and Ila Fiete (2022). "No Free Lunch from Deep Learning in Neuroscience: A Case Study through Models of the Entorhinal-Hippocampal Circuit". In: *ICML 2022 Workshop AI4Science*.
Schaffer, Evan S et al. (2018). "Odor perception on the two sides of the brain: consistency despite randomness". In: *Neuron* 98.4, pp. 736–742.
Schmidt, Philipp and Felix Biessmann (2019). "Quantifying interpretability and trust in machine learning systems". In: *arXiv preprint arXiv:1901.08558*.
Schott, Lukas et al. (Feb. 2022). *Visual Representation Learning Does Not Generalize Strongly Within the Same Domain*. arXiv:2107.08221 [cs]. URL: http://arxiv.org/abs/2107.08221 (visited on 11/09/2022).
Schubert, Ludwig et al. (2021). "High-low frequency detectors". In: *Distill* 6.1, e00024–005.
Schultz, Wolfram, Peter Dayan, and P Read Montague (1997). "A neural substrate of prediction and reward". In: *Science* 275.5306, pp. 1593–1599.
Seeley, Thomas D (2011). *Honeybee democracy*. Princeton University Press.
Sengupta, Anirvan et al. (2018). "Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks". In: *Advances in neural information processing systems* 31.
Sezener, Eren et al. (2021). "A rapid and efficient learning rule for biological neural circuits". In: *BioRxiv*, pp. 2021–03.
Shadmehr, Reza (2020). "Population coding in the cerebellum: a machine learning perspective". In: *Journal of neurophysiology* 124.6, pp. 2022–2051.
Sharkey, Lee (2023). "A technical note on bilinear layers for interpretability". In: *arXiv preprint arXiv:2305.03452*.
Shu, Rui et al. (2019). "Weakly supervised disentanglement with guarantees". In: *arXiv preprint arXiv:1910.09772*.
Simon, James B, Madeline Dickens, and Michael R DeWeese (2021). "Neural tangent kernel eigenvalues accurately predict generalization". In: *arXiv preprint arXiv:2110.03922*.
Singer, Saša and John Nelder (2009). "Nelder-mead algorithm". In: *Scholarpedia* 4.7, p. 2928.
Sollich, Peter (1998). "Learning curves for Gaussian processes". In: *Advances in neural information processing systems* 11.
Solstad, Trygve et al. (Dec. 2008). "Representation of Geometric Borders in the Entorhinal Cortex". In: *Science* 322.5909. _eprint: NIHMS150003, pp. 1865–1868. ISSN: 0036-8075. DOI: 10.1126/science.1166466. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1166466.
Sorrenson, Peter, Carsten Rother, and Ullrich Köthe (2020). "Disentanglement by nonlinear ica with general incompressible-flow networks (gin)". In: *arXiv preprint arXiv:2001.04872*.
Sorscher, Ben, Surya Ganguli, and Haim Sompolinsky (2022). "Neural representational geometry underlies few-shot concept learning". In: *Proceedings of the National Academy of Sciences* 119.43, e2200800119.
Sorscher, Ben, Gabriel Mel, et al. (2019). "A unified theory for the origin of grid cells through the lens of pattern formation". In: *Advances in neural information processing systems* 32.
Sorscher, Ben, Gabriel C Mel, et al. (2023). "A unified theory for the computational and mechanistic origins of grid cells". In: *Neuron* 111.1, pp. 121–137.
Sprekeler, Henning, Tiziano Zito, and Laurenz Wiskott (2014). "An extension of slow feature analysis for nonlinear blind source separation". In: *The Journal of Machine Learning Research* 15.1, pp. 921–947.
Sreenivasan, Sameet and Ila Fiete (2011). "Grid cells generate an analog error-correcting code for singularly precise neural computation". In: *Nature neuroscience* 14.10, pp. 1330–1337.
Stachenfeld, Kimberly L, Matthew M Botvinick, and Samuel J Gershman (2017). "The hippocampus as a predictive map". In: *Nature neuroscience* 20.11, pp. 1643–1653.
Stemmler, Martin, Alexander Mathis, and Andreas VM Herz (2015). "Connecting multiple spatial scales to decode the population activity of grid cells". In: *Science Advances* 1.11, e1500816.
Stensola, Hanne et al. (2012). "The entorhinal grid map is discretized". In: *Nature* 492.7427, pp. 72–78.
Stensola, Tor et al. (2015). "Shearing-induced asymmetry in entorhinal grid cells". In: *Nature* 518.7538, pp. 207–212.

Stringer, Carsen et al. (2019). "High-dimensional geometry of population responses in visual cortex". In: *Nature*. Publisher: Springer US. ISSN: 14764687. DOI: 10.1038/s41586-019-1346-5. URL: http://dx.doi.org/10.1038/s41586-019-1346-5.

Sun, Chen et al. (2020). "Hippocampal neurons represent events as transferable units of experience". In: *Nature neuroscience* 23.5, pp. 651–663.

Tenenbaum, Joshua B and William T Freeman (2000). "Separating style and content with bilinear models". In: *Neural computation* 12.6, pp. 1247–1283.

Theis, Lucas et al. (2017). "Lossy image compression with compressive autoencoders". In: *arXiv preprint arXiv:1703.00395*.

Tjandra, Andros, Sakriani Sakti, and Satoshi Nakamura (2020). "Transformer vq-vae for unsupervised unit discovery and speech synthesis: Zerospeech 2020 challenge". In: *arXiv preprint arXiv:2005.11676*.

Tolman, Edward C (1948). "Cognitive maps in rats and men." In: *Psychological review* 55.4.

Tootoonian, Sina and Máté Lengyel (2014). "A dual algorithm for olfactory computation in the locust brain". In: *Advances in neural information processing systems* 27.

Van Den Oord, Aaron, Oriol Vinyals, et al. (2017). "Neural discrete representation learning". In: *Advances in neural information processing systems* 30.

Vanschoren, Joaquin (2019). "Meta-learning". In: *Automated machine learning*. Springer, Cham, pp. 35–61.

Ven, Gido M van de, Hava T Siegelmann, and Andreas S Tolias (2020). "Brain-inspired replay for continual learning with artificial neural networks". In: *Nature communications* 11.1, pp. 1–14.

Veness, Joel et al. (2021). "Gated linear networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 11, pp. 10015–10023.

Walker, Jacob, Ali Razavi, and Aäron van den Oord (2021). "Predicting video with vqvae". In: *arXiv preprint arXiv:2103.01950*.

Wang, Zhuo (2016). *Optimal neural codes for natural stimuli*. University of Pennsylvania.

Wanner, Adrian A and Rainer W Friedrich (2020). "Whitening of odor representations by the wiring diagram of the olfactory bulb". In: *Nature neuroscience* 23.3, pp. 433–442.

Watter, Manuel et al. (2015). "Embed to control: A locally linear latent dynamics model for control from raw images". In: *Advances in neural information processing systems* 28.

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

Whittington, James C. R., Will Dorrell, et al. (2023). "Disentanglement with Biological Constraints: A Theory of Functional Cell Types". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=9Z_GfhZnGH.

Whittington, James C. R., Rishabh Kabra, et al. (2021). "Constellation: Learning relational abstractions over objects for compositional imagination". In: *arXiv preprint*. _eprint: 2107.11153. URL: http://arxiv.org/abs/2107.11153.

Whittington, James CR, Will Dorrell, et al. (2022). "Disentangling with Biological Constraints: A Theory of Functional Cell Types". In: *arXiv preprint arXiv:2210.01768*.

Whittington, James CR, Timothy H Muller, et al. (2020). "The Tolman-Eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation". In: *Cell* 183.5, pp. 1249–1263.

Whittington, James CR, Joseph Warren, and Timothy EJ Behrens (2021). "Relating transformers to models and neural representations of the hippocampal formation". In: *arXiv preprint arXiv:2112.04035*.

Widloski, John and Ila R Fiete (2014). "A model of grid cell development through spatial exploration and spike time-dependent plasticity". In: *Neuron* 83.2, pp. 481–495.

Williams, Alex et al. (2020). "Point process models for sequence detection in high-dimensional neural spike trains". In: *Advances in neural information processing systems* 33, pp. 14350–14361.

Willmore, Ben DB and Andrew J King (2023). "Adaptation in auditory processing". In: *Physiological Reviews* 103.2, pp. 1025–1058.

Wolpert, Daniel M, Zoubin Ghahramani, and Michael I Jordan (1995). "An internal model for sensorimotor integration". In: *Science* 269.5232, pp. 1880–1882.

Wu, Hanwei and Markus Flierl (2019). "Learning product codebooks using vector-quantized autoencoders for image retrieval". In: *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, pp. 1–5.

Xi, Quanhan and Benjamin Bloem-Reddy (2023). "Indeterminacy in Generative Models: Characterization and Strong Identifiability". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 6912–6939.

Xie, Marjorie et al. (2022). "Task-dependent optimal representations for cerebellar learning". In: *bioRxiv*.

Xu, Yilun et al. (2020). "A theory of usable information under computational constraints". In: *arXiv preprint arXiv:2002.10689*.

Yamins, Daniel LK et al. (2014). "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the national academy of sciences* 111.23, pp. 8619–8624.

Yan, Wilson et al. (2021). “Videogpt: Video generation using vq-vae and transformers”. In: *arXiv preprint arXiv:2104.10157*.
Yang, En et al. (2022). “A brainstem integrator for self-location memory and positional homeostasis in zebrafish”. In: *Cell* 185.26, pp. 5011–5027.
Yang, Guangyu Robert et al. (2019). “Task representations in neural networks trained to perform many cognitive tasks”. In: *Nature neuroscience* 22.2, pp. 297–306.
Yang, Xiaojiang et al. (2022). “Nonlinear ica using volume-preserving transformations”. In: *International Conference on Learning Representations*.
Yu, Changmin, Timothy EJ Behrens, and Neil Burgess (2020). “Prediction and Generalisation over Directed Actions by Grid Cells”. In: *arXiv preprint arXiv:2006.03355*.
Yu, Jiahui et al. (2022). “Scaling autoregressive models for content-rich text-to-image generation”. In: *arXiv preprint arXiv:2206.10789*.
Zavitz, D et al. (2021). “Connectivity patterns that shape olfactory representation in a mushroom body network model”. In.
Zenke, Freidemann (2019). *SpyTorch*. https://zenodo.org/record/3724018#.Yy7coOzML9t.
Zenke, Friedemann and Tim P Vogels (2021). “The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks”. In: *Neural computation* 33.4, pp. 899–925.
Zhang, Marvin et al. (2019). “Solar: Deep structured representations for model-based reinforcement learning”. In: *International conference on machine learning*. PMLR, pp. 7444–7453.
Zheng, Yujia, Ignavier Ng, and Kun Zhang (2022). “On the Identifiability of Nonlinear ICA: Sparsity and Beyond”. In: *arXiv preprint arXiv:2206.07751*.
Zheng, Zhihao, J Scott Lauritzen, et al. (2018). “A complete electron microscopy volume of the brain of adult Drosophila melanogaster”. In: *Cell* 174.3, pp. 730–743.
Zheng, Zhihao, Feng Li, et al. (2022). “Structured sampling of olfactory input by the fly mushroom body”. In: *Current Biology* 32.15, pp. 3334–3349.