Abstract representations emerge naturally in neural networks trained to perform multiple tasks

W. Jeffrey Johnston[1,2,*] and Stefano Fusi[1,2,*]

[1]Center for Theoretical Neuroscience
[2]Mortimer B. Zuckerman Mind, Brain, and Behavior Institute
Columbia University, New York, NY, USA

[*]Corresponding authors: wjeffreyjohnston@gmail.com and sf2237@columbia.edu

October 21, 2021

Abstract

Humans and other animals demonstrate a remarkable ability to generalize knowledge across distinct contexts and objects during natural behavior. We posit that this ability depends on the geometry of the neural population representations of these objects and contexts. Specifically, abstract, or disentangled, neural representations – in which neural population activity is a linear function of the variables important for making a decision – are known to allow for this kind of generalization. Further, recent neurophysiological studies have shown that the brain has sufficiently abstract representations of some sensory and cognitive variables to enable generalization across distinct contexts. However, it is unknown how these abstract representations emerge. Here, using feedforward neural networks, we demonstrate a simple mechanism by which these abstract representations can be produced: The learning of multiple distinct classification tasks. We demonstrate that, despite heterogeneity in the task structure, abstract representations that enable reliable generalization can be produced from a variety of different inputs – including standard nonlinearly mixed inputs, inputs that mimic putative representations from early sensory areas, and even simple image inputs from a standard machine learning data set. Thus, we conclude that abstract representations of sensory and cognitive variables emerge from the multiple behaviors that animals exhibit in the natural world, and may be pervasive in high-level brain regions. We make several specific predictions about which variables will be represented abstractly as well as show how these representations can be detected.

1 Introduction

The ability to generalize existing knowledge to novel stimuli or situations is essential to complex, rapid, and accurate behavior. As an example, when shopping for produce, humans make many different decisions about whether or not different pieces of produce are ripe – and, consequently, whether to purchase them. The knowledge we use in the store is often learned from experience with that fruit at home – thus, generalizing across distinct contexts. Further, the knowledge that we apply to a fruit that we buy for the first time might be derived from similar fruits – generalizing, for instance, from an apple to a pear. The determinations themselves are often multi-dimensional and multi-sensory: both firmness and appearance are important for deciding whether an avocado is the right level of ripeness. Yet, at the end of this complex process, we make a binary decision about each piece of fruit: we add it to our cart, or do not – and get feedback later about whether that was the right decision or not. This produce shopping example is not unique. Humans and other animals exhibit an impressive ability to generalize across contexts and between different objects in many situations. However, the neural underpinnings of this ability to generalize are not fully understood.

We hypothesize that this ability to generalize is tied to the geometry of neural representations. In particular, neural representations of sensory and cognitive variables are often highly nonlinear and have high embedding dimension[26, 9, 30]. While the nonlinearity of the representations allows flexible learning of new behaviors[9] and provides metabolically efficient and reliable representations[14], high-dimensional representations often do not permit generalization of knowledge across related contexts or stimuli[3]. Alternatively, for a representation with low embedding dimension, a classifier that learns to discriminate between a single pair of stimuli based on one latent variable may generalize to discriminate between other pairs of stimuli that differ in other latent variables. Recent experimental work has shown that low-dimensional, near-linear representations that could support this ability to generalize exist at the apex of the primate ventral visual stream, for faces in inferotemporal cortex[5, 12, 27]. Further, experimental work in the hippocampus and prefrontal cortex has shown that abstract representations exist for the sensory and cognitive features related to a complex cognitive task, and that these representations could support generalization[3]. We refer to low-dimensional, linear representations of task-relevant sensory and cognitive variables – like in these examples and others[7, 28] – as abstract representations.

In the machine learning literature, abstract representations are often referred to as factorized[2] or disentangled[2, 13, 4, 12] representations of interpretable stimulus features. Deep learning has been used to produce abstract representations primarily in the form of unsupervised generative models[16, 6, 13] (but see [18]). In this context, abstract representations are desirable because they allow potentially novel examples of existing stimulus classes to be produced by linear interpolation in the abstract representation space (for example, starting at a known exemplar and changing its orientation by moving linearly along

a dimension in the abstract representation space that is known to correspond to orientation)[13]. However, the machine learning models shown to have good performance at producing abstract representations often sacrifice reconstruction performance and can be brittle with respect to hyperparameter choices[13, 18] (but see [4]). Importantly for the application of these results from machine learning to phenomena observed in neuroscience, behaving animals need not perfectly reconstruct the sensory world – as would be analogous to the generative autoencoder case – and often behavior is driven by different forms of external feedback. Instead, behaving animals need to be able to perform a variety of distinct behaviors, or tasks, often applied to similar sensory stimuli – and these task yield some supervisory information (as in the produce shopping metaphor above).

While experimental work on animals performing more than a couple of distinct behavioral tasks remains nearly nonexistent[34], modeling work using recurrent neural networks has shown that the networks often develop representations that can be reused across distinct, but related tasks[35] – though the abstractness of these reusable representations was not measured. Thus, the behavioral constraint of multi-tasking may encourage the learning of abstract representations of stimulus features that are relevant to multiple tasks. To investigate this hypothesis, we train feedforward neural network models to perform multiple distinct tasks on a common stimulus space. In many cases, the representations developed in neural networks trained with backpropagation have been shown to closely mirror the structure of representations that exist in the brain[25].

Here, we ask how abstract representations – like those observed in higher brain regions[5, 3] – can be constructed from the nonlinear and high-dimensional representations observed in early sensory areas[33, 23, 20, 17, 29, 30]. To study this, we begin by constructing high-dimensional and nonlinear representations of latent variables, designed to be similar to representations observed in the brain (see *Non-abstract input generation* in *Methods*). Then, using a feedforward neural network model, we test our hypothesis that requiring multiple distinct tasks to be performed on these latent variables will induce abstract representations.

First, we show analytically that a multi-task setup will produce at least moderately abstract representations. Then, we introduce the multi-tasking model and show that the representations it produces are even more abstract than guaranteed by the theory. These abstract representations are surprisingly robust to heterogeneity and context-dependence in the tasks performed by the multi-tasking model. In these manipulations, we attempt to mirror the complex structure of real behavior: In particular, some tasks are only performed on some stimuli (e.g., firmness is more informative about an avocado than an orange) and, in each separate instance, we are only performing one of potentially many different behaviors (e.g., selecting fruit to eat or to use in a cake). We also explore the case in which only a fraction of tasks are closely related to the latent variables, and the remaining larger fraction are not. The multi-tasking model produces abstract representations in all of these cases. Throughout, we contrast the abstractness of representations produced by the multi-tasking model with

those produced by the β-variational autoencoder (βVAE), which is the current state of the art for producing abstract representations in machine learning. Finally, we demonstrate two applications of the multi-tasking model: First, we show that it reliably produces abstract representations from receptive field-like inputs, which are both highly nonlinear and non-abstract, as well as similar to the format of representations reported in the brain[33, 23, 20, 17, 29]; second, we show that the multi-tasking model produces abstract representations in a generative, machine learning context, which can be used to generate example images with particular features. Finally, we use this framework to make several predictions for how neural representations in the brain will be shaped by behavioral context. Overall, our work shows that abstract representations – similar to those observed in the brain[5, 3] – reliably emerge from learning to multi-task in multi-dimensional environments. We show that this multi-tasking constraint produces abstract representations even from the highly nonlinear representations thought to be developed in early sensory areas, where other methods fail to recover abstract structure. This indicates that abstract representations may be a consequence of – as well as a boon to[32] – complex behavior.

2 Results

2.1 Abstract representations allow knowledge to be generalized across contexts

Information in the world can often be generalized across contexts. For instance, the texture of a berry with a particular shape is often similar whether that berry is red or blue (fig. 1a, top); further, berries that are red may taste more similar to each other, despite differences in shape, than they do to berries that are blue (fig. 1a, bottom). Learning and taking advantage of this structure in the sensory world is important for animals that need to quickly react to novel stimuli that may be related to previously experienced stimuli.

We refer to neural representations as abstract if they reproduce the latent structure that is present in the sensory world in their geometry. From the example above, we can view shape and color as two continuous latent variables that describe different kinds of berries. An abstract or disentangled representation of these latent variables is a linear representation of them in neural population activity, and would have a low-dimensional rectangular structure in neural population space (fig. 1b, left); a non-abstract representation of these latent variables would have a non-rectangular, higher-dimensional distorted structure, such as one created by neurons that each respond only to particular conjunctions of color and shape (fig. 1b, right). The abstract representation has the desirable quality that, if we learned a neural readout that classifies blue berries from red berries using berries with only one shape (e.g., the two bottom berries in fig. 1b, left), then we would not need to modify this classifier to apply it to berries of a different shape (e.g., the two top berries in fig. 1b, left); this property does not hold for the non-abstract representation (compare the two

berries to the left and to the right in fig. 1b, right). This ability to generalize to novel classes of stimuli is highly desirable for behavior and for machine learning systems, and it relies on this correspondence between the latent structure of the sensory world and its representation that we refer to as abstraction.

To quantify the abstractness of a representation, we use two distinct metrics that are both related to metrics used to quantify abstraction in neural representations recorded experimentally[3]. The first tests how well a classifier that is trained on one half of the stimulus space generalizes to the left out half of stimulus space (fig. 1c, top). High classifier generalization performance has been observed for sensory and cognitive features in neural data recorded from the hippocampus and prefrontal cortex[3]. The second tests how well a linear regression model that is trained on one half of the stimulus space generalized to the left out half of stimulus space (fig. 1c, bottom; this metric is similar to several metrics used in the machine learning literature[13, 15]). The classifier generalization metric requires that the coarse structure of the representations be abstract, but is less sensitive to small deviations. The regression generalization metric is much stricter, and is sensitive to even small deviations from a representation that follows the underlying latent variable structure. In some cases, we also compare these metrics of out-of-distribution generalization to standard cross-validated performance on the whole latent variable space. Intuitively, the standard cross-validated performance of both metrics serves as a best case for their out-of-distribution generalization performance (i.e., the case where what is learned from only half the representation space is just as informative about the global representation structure as what would be learned from the whole representation space). In a perfectly abstract representation, the standard and out-of-distribution generalization performances would be equal to each other. Just as similar metrics were used to quantify the abstractness of neural representations recorded experimentally, we use the classification and regression generalization performance to quantify the abstractness of the representations developed by our feedforward neural network model.

2.2 Understanding the learning dynamics that produce abstract representations

First, we develop a model to construct non-abstract representations of known D-dimensional latent variables (here, the latent variables are given a standard normal distribution – though the results are similar for a uniform distribution, see *A multiverse analysis of the multi-tasking model and βVAE* in *Supplement*). Later, we will use these non-abstract representations as input to our models that seek to learn abstract representations. To produce these non-abstract representations, we train a feedforward neural network with an autoencoder to satisfy two objectives: First, to maximize the dimensionality of activity in the representation layer; second, to reconstruct the original stimulus using only its representation. That is, we want a high dimensional representation that still preserves all of the information about the input. This transformation produces a distorted and, to some degree, tangled representation of the latent variables

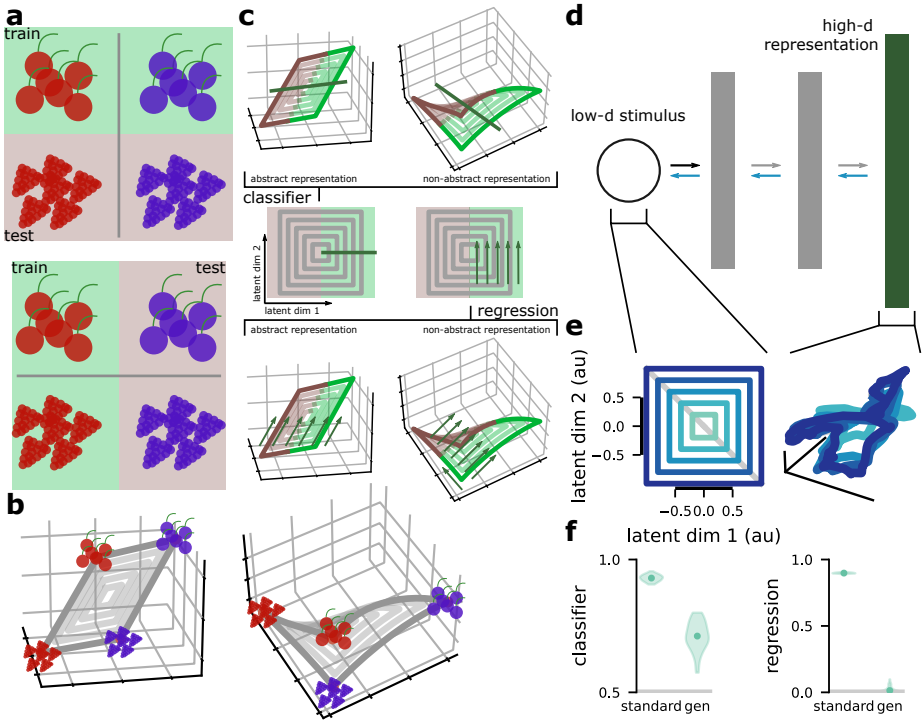

Figure 1: The abstraction metrics and input representations. **a** Illustration of the classification tasks. (top) A classification learned between red and blue berries of one shape should generalize to other shapes. (bottom) A classification between red berries of two different shapes should generalize to blue berries of different shapes. **b** Examples of linear, abstract (left) and nonlinear, non-abstract (right) representations of the four example berries. **c** Illustration of our two abstraction metrics. For a $D = 2$-dimensional latent variable (middle), we split the latent variable distribution into two regions: one used for training (green, left) and one used for testing (red, right). In the classifier generalization metric, we train a linear classifier to perform a binary classification of the latent variables using samples from the green region and test that classifier on samples from the red region (left). The abstract representation (top left) has good classifier generalization performance; the non-abstract representation (top right) has poor classifier generalization performance. The regression generalization metric is similar, but uses a linear regression model (right). The abstract representation (bottom left) has good regression generalization performance; the non-abstract representation (bottom right) has poor regression generalization performance. **d** We use a feedforward network to produce high-dimensional representations from our D-dimensional latent variables. The network is trained to maximize the dimensionality of the representation layer (green), while also retaining the ability to reconstruct the original latent variables through an autoencoder (blue lines). **e** We visualize the structure of the latent variable representation by plotting the first three principal components (left: original structure; right: representation structure). **f** We use both the classifier (left) and regression (right) generalization metrics to quantify that level of abstraction. In each plot, the left point is for the metric trained and tested on the whole space, the right point is trained on one half of the space and tested on the other half. The grey line is chance.

(fig. 1d) and, for a 5-dimensional latent variable, produces a representation layer with a dimensionality of approximately 200 (see *Participation ratio-maximized*

representations in *Methods* for more detail). We visualize this transformation by constructing concentric squares in the latent variable space (fig. 1e, left) and then visualizing the representation of these squares produced by the network (fig. 1e, right). If the concentric square structure is intact in the representation, then the representation is abstract; otherwise, the representation is non-abstract. The distorted representation of the latent variables produced here significantly decreases abstraction, as measured by both classifier (fig. 1f, left) and regression (fig. 1f, right) generalization metrics.

To recover abstract structure from non-abstract representations, we focus on what we refer to as the multi-tasking model. The multi-tasking model is a multilayer feedforward neural network model that is trained to perform a number of different binary classification tasks. These tasks can be viewed as analogous to the tasks that animals perform, as described above. For instance, if an animal eats a berry, the animal later receives information about whether that berry was edible or poisonous. If we assume that the edibility of a berry is represented by one of our D latent variables, then, in the multi-tasking model, this classification task corresponds to the model being trained to produce one output when the latent variable is positive and another output when the latent variable is negative. Importantly, the model (just like an animal) only has access to the sign of the latent variable, not its precise value. In the full model, each classification task does not correspond to a single latent variable, as the number of tasks P will often exceed the number of latent variables D. Instead, the tasks are chosen to be random hyperplanes in the latent variable space. To begin, each classification task hyperplane is chosen to divide the latent variable space into two balanced halves (as in fig. 1a), though this is later relaxed (see fig. 3a, b). At first, the model is trained to perform all P classification tasks simultaneously on each stimulus (fig. 2a, right), though this too is later relaxed (fig. 3a, b). In all of our analyses, we focus on the representations of the stimuli that are developed in the layer just prior to the task output layer. This layer is referred to as the representation layer.

To understand these representations, we consider three distinct solutions to the simultaneous performance of P classification tasks as formalized here. First, the representation could split along P separate dimensions of population activity, where each dimension corresponds to one of the P distinct tasks (fig. 2b, left). Second, the representation could consist only of an approximately D-dimensional sphere (or circle, in two dimensions), which exploits the correlation structure in the P different tasks (that is, when $P > D$, the outcomes from some pairs of tasks are necessarily correlated with each other; fig. 2b, middle). This second type of representation would have high classifier generalization performance but low regression generalization performance: That is, it is partially abstract in that it would recover the angular structure of the latent variables (as necessary for the P classification tasks), but not their magnitude (as this information is not necessary to solve the P tasks). Third, a fully abstract representation of the latent variables could be recovered. That is, the representation could recover both the angular structure of the latent variables, as in the second possibility, and their magnitude (fig. 2b, right). This would occur only if the

multi-tasking model does not discard information about the stimuli that is not necessary for satisfying the tasks, but which is also not explicitly trained to discard. Surprisingly, as we will see, this third form of representation is most common in our trained networks, even for more disordered tasks than we have described so far.

Next, we show analytically that only the second two forms of representation are likely to be developed in a machine learning system. In particular, we show that the feedback onto the representation layer increases the strength of the representation of an approximately D-dimensional component of the activity. If, instead, the multi-tasking model was to learn P relatively independent response dimensions, as in the first possibility, then this feedback component would have to be roughly P-dimensional. In particular, the dynamics of the activity in the representation layer $r(x)$ across training have the following form for simple backpropagation:

$$r(x)^{s+1} = r(x)^s - \mu \, \mathbb{E}_X \left[\frac{\partial L}{\partial r(x)} \right] \tag{1}$$

$$= r(x)^s + \mu \, \mathbb{E}_X \, W^T \operatorname{sign}(Ax) - \mu \, \mathbb{E}_X \, W^T W r(x) \tag{2}$$

where L is the MSE loss function, μ is the learning rate, A is the $P \times D$ matrix of random classification tasks, W are the weights between the representation layer and output layer, and $\mathbb{E}_X$ is the expectation over x. Thus, $r(x)$ will become dominated by a linear transform of $\operatorname{sign}(Ax)$ as training progresses. So, we show that,

$$\dim \left(\mathbb{E}_X \, \operatorname{sign}(Ax) \operatorname{sign}(Ax)^T \right) \approx \dim \left(\mathbb{E}_X \, Axx^T A^T \right) \tag{3}$$

$$= \min(P, D) \tag{4}$$

and the approximation becomes closer as D becomes larger (see *The dimensionality of representations in the multi-tasking model* in *Methods* for details). This means that, given sufficiently long training, the representation layer will be dominated by a $\min(P, D)$-dimensional representation of the latent variables. Given that this representation must also be able to satisfy the P tasks, it will be either of type two or three described above – that is, it will have at least good classifier generalization performance and may even have good regression generalization performance.

2.3 Learning multiple classification tasks leads to abstract representations

We show that feedforward multilayer neural networks, when trained to perform $P \geq D$ classification tasks, develop abstract representations of the D latent variables (as schematized in fig. 2b, right) rather than develop either nonlinear, task-specialized representations or magnitude-collapsed representations (as schematized in fig. 2b, left and middle).

First, we visualize how the representations developed by our model compare to the abstract latent variables. In particular, we again generate concentric

squares in latent variable space and show their idealized abstract representation (fig. 2d, left) alongside the representations actually developed by the model (fig. 2d, right). For only a single task, the representations in the model collapse along a single dimension, which corresponds to performance of that task (fig. 2d, top). While this representation is not abstract, it does mirror distortions in sensory representations that are often observed when animals are overtrained on single tasks[8, 31]. However, when we include a second task in the training procedure, abstract representations begin to emerge (fig. 2d, middle). In particular, the representation layer is dominated by a two-dimensional abstract representation of a linear combination of two of the latent variables. From our theory – and confirmed by these simulations – we know that when $P < D$, then the dimensionality of this dominating component in the representation layer will be approximately P. Next, we demonstrate that this abstract structure becomes more complete as the number of tasks included in training is increased. For $P = 8$ and $D = 5$, the visualization suggests that the representation has become fairly abstract (fig. 2d, bottom).

Next, we quantify how the level of abstraction developed in the representation layer depends on the number of classification tasks used to train the model (fig. 2e). For each number of classification tasks, we train 10 multi-tasking models to characterize how the metrics depend on random initial conditions. As the number of classification tasks P exceeds the dimensionality of the latent variables D, both the classification and regression generalization metrics saturate to near their maximum possible values (classifier generalization metric: exceeds 90 % correct with 8 tasks; regression generalization metric: exceeds $r^2 = .8$ with 9 tasks; fig. 2e, right of the grey line). Saturation of the classifier generalization metric indicates that the broad organization of the latent variables is perfectly preserved (but detailed information may have been lost); while saturation of the regression generalization metric indicates that even the magnitude information that the multi-tasking model did not receive supervised information about is preserved and represented in a fully abstract format. Importantly, both the training and testing set split and the classification boundary for the classifier generalization metric are randomly selected – they are not the same as classification tasks used in training.

2.4 Abstract representations emerge even when the classification tasks are heterogeneous

Next, we test how robust this finding is to changes to the classification tasks themselves. In particular, we show that our finding holds for three manipulations to the task structure. First, we show that unbalanced tasks (e.g., a more or less stringent criteria for judging the ripeness of a fruit – so either many more of the fruit are considered ripe than spoilt or vice versa; fig. 3a, top left; see *Unbalanced task partitions* in *Methods* for more details) have a negligible effect on the emergence of abstract representations (classifier generalization metric: exceeds 90 % correct with 9 tasks, regression generalization metric: exceeds $r^2 = .8$ with 9 tasks; fig. 3b). Second, we show that contextual tasks (e.g.,

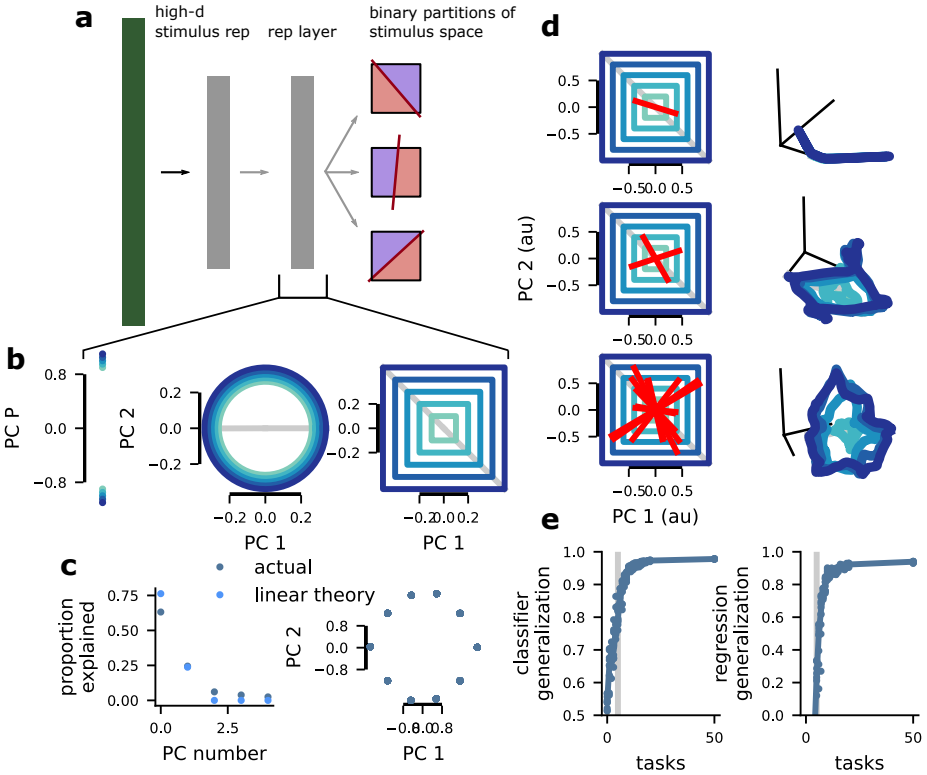

Figure 2: The emergence of abstraction from classification task learning. **a** Schematic of the multi-tasking model. It receives an entangled stimulus representation (as shown in fig. 1e, left) and learns to perform P binary classifications of the latent variables. We study the representations that this induces in the layer prior to the output: the representation layer. **b** Different possible solutions the network could learn. (left) The network could learn a dimension for each classification task and develop binary representations along each of those P dimensions. (middle) The network could learn a surface that matches the dimensionality D of the latent variables, but discards information about magnitude; this representation would have high classifier- but low regression-generalization performance. (right) The network could learn a fully abstract, approximately D-dimensional representation of the latent variables. **c** The dimensionality of the representation layer will be approximately D-dimensional (left), as predicted by eq. (4). (right) The first two principal components of the required output of the network; this structure is consistent with both the middle and right network solutions from **b**. **d** Examples of multi-tasking model representations for different numbers of classification tasks. We show a schematic of an idealized fully abstract representation (left) alongside the representation developed by the network (right). (top) When the model learns one task (left, red line), the representation (right) collapses into one dimension. (middle) When the model learns two tasks (left, red lines), it recovers more of the stimulus structure (right). (bottom) When the model learns more tasks than stimulus dimensions (here, stimulus dimension is five and eight tasks are learned), the model can produce a highly abstract representation of the original stimuli. **e** The classifier (left) and regression (right) metrics applied to model representations with different numbers of tasks.

determining the ripeness of different fruits that occupy only a fraction of latent variable space; fig. 3a, top right; see *Contextual task partitions* in *Methods*

for more details) produce a moderate increase in the number of tasks required to learn abstract representations (classifier generalization metric: exceeds 90 % correct with 14 tasks, regression generalization metric: exceeds $r^2 = .8$ with 14 tasks; fig. 3b). Third, we show that using training examples with information from only a single task (e.g., getting only a single data point on each trip to the store; fig. 3a, bottom, see *Partial information task partitions* in *Methods* for more details) also only moderately increase the number of tasks necessary to produce abstract representations (classifier generalization metric: exceeds 90 % correct with 11 tasks, regression generalization metric: exceeds $r^2 = .8$ with 14 tasks; fig. 3b).

Together, these results indicate the the multi-tasking model reliably produces abstract representations even given substantial heterogeneity in the amount of information per stimulus example and the form of that information relative to the latent variables. Further, these results are also robust to variation in architecture: Changing the width, depth, and several other parameters of the multi-tasking model have only minor effects on classification and regression generalization performance (see *A multiverse analysis of the multi-tasking model and βVAE* in *Supplement*).

2.5 The tasks learned by the model shape the representations

Our result show that the multi-tasking model robustly recovers the latent variables present in nonlinearly mixed stimulus representations by learning to perform classification tasks that are related to those latent variables (that is, the classification boundary for each task is defined by a vector on those latent variables). We demonstrate the specificity of this finding in two ways: First, we train the multi-tasking model on stimulus representations of $D = 5$ latent variables using classification tasks that rely on only $D_{\text{trained}} = 3$ of those latent variables (fig. 3c). As expected, the multi-tasking model learns an abstract representation of the trained latent variables (fig. 3d, blue line) but do not develop an abstract representation of the remaining, untrained latent variables (fig. 3d, grey line). Second, we construct classification tasks that are not aligned with the latent variables at all. In particular, we construct grid classification tasks, in which the latent variable space is divided into grid chambers, where each chamber has a roughly equal probability of being sampled (fig. 3e, red lines). Then, we randomly assign the each of the chambers to one of two categories (fig. 3e, coloring; see *Grid classification tasks* in *Methods* for more details). In this case, there is nothing in the design of the multi-tasking model that privileges a representation of the original latent variables, since they are no longer useful for learning to perform the multiple grid classification tasks that it must learn during training. In this case, the multi-tasking model does not recover a representation of the original latent variables (fig. 3f). We argue that the multi-tasking model also follows what would be expected in the natural world: latent variables are learned as a way to solve multiple related tasks and to generalize knowledge from one task to another, rather than for their own sake.

To make this intuition about the grid tasks more explicit, we show that – in contrast to the latent variable-aligned tasks that we have been using so far – the outcomes from a particular grid task are likely to be only weakly correlated with the outcomes from a different, randomly chosen grid task (fig. 3g). Thus, rather than having a D-dimensional structure even for $P \gg D$ tasks, the grid tasks will have a roughly P-dimensional structure for P tasks. As expected, the multi-tasking model fails to learn a strongly abstract representation of the original latent variables, and the representation becomes less abstract as the grid tasks become higher dimensional (i.e., when the grid has more chambers; fig. 3f, middle and right, blue and purple lines).

Next, we examine the representations learned by the multi-tasking model when it must perform a mixture of latent variable-aligned and grid classification tasks. This situation is also chosen to mimic the natural world, as a set of latent variables may be relevant to some behaviors (the latent variable-aligned classification tasks), but an animal may need to perform additional behaviors on the same set of stimuli that do not follow the latent variable structure (the grid classification tasks). Here, we train the multi-tasking model to perform a fixed number of latent variable-aligned tasks, which are sufficient to develop abstract structure in isolation (here, 15 tasks). However, at the same time, the model is also being trained to perform various numbers of grid tasks. While increasing the number of grid tasks does moderately decrease the abstractness of the developed representation (fig. 3h, middle and right), the multi-tasking model retains strongly abstract representations even while performing more than 45 grid tasks – 3 times as many as the number of latent variable-aligned tasks.

Intuitively, this occurs because the latent variable-aligned tasks are highly correlated with each other and follow the structure of the D-dimensional latent variable space, while each of the grid tasks has low correlation with any other grid task (fig. 3b). Thus, a shared representation structure is developed to solve all the latent variable-aligned tasks essentially at once (corresponding to significant fraction of the variance in the target function), while a smaller nonlinear component is added on to solve each of the grid tasks relatively independently. Interestingly, the combination of abstract structure with nonlinear distortion developed by the multi-tasking model here has also been observed in the brain and other kinds of feedforward neural networks (though learning tasks analogous to our grid tasks was not necessary for it to emerge)[3]. We believe that this compromise between strict abstractness to allow for generalization and nonlinear distortion to allow for flexible learning of random tasks[26, 9] is fundamental to the neural code.

2.6 Abstract structure can be learned from early sensory-like representations.

While abstract representations have been widely observed in the brain, many representations have also been shown to be highly nonlinear[33, 23, 20, 17, 29, 30]. Thus, any mechanism for producing abstract representations, must be able to produce them from highly nonlinear and non-abstract representations. In

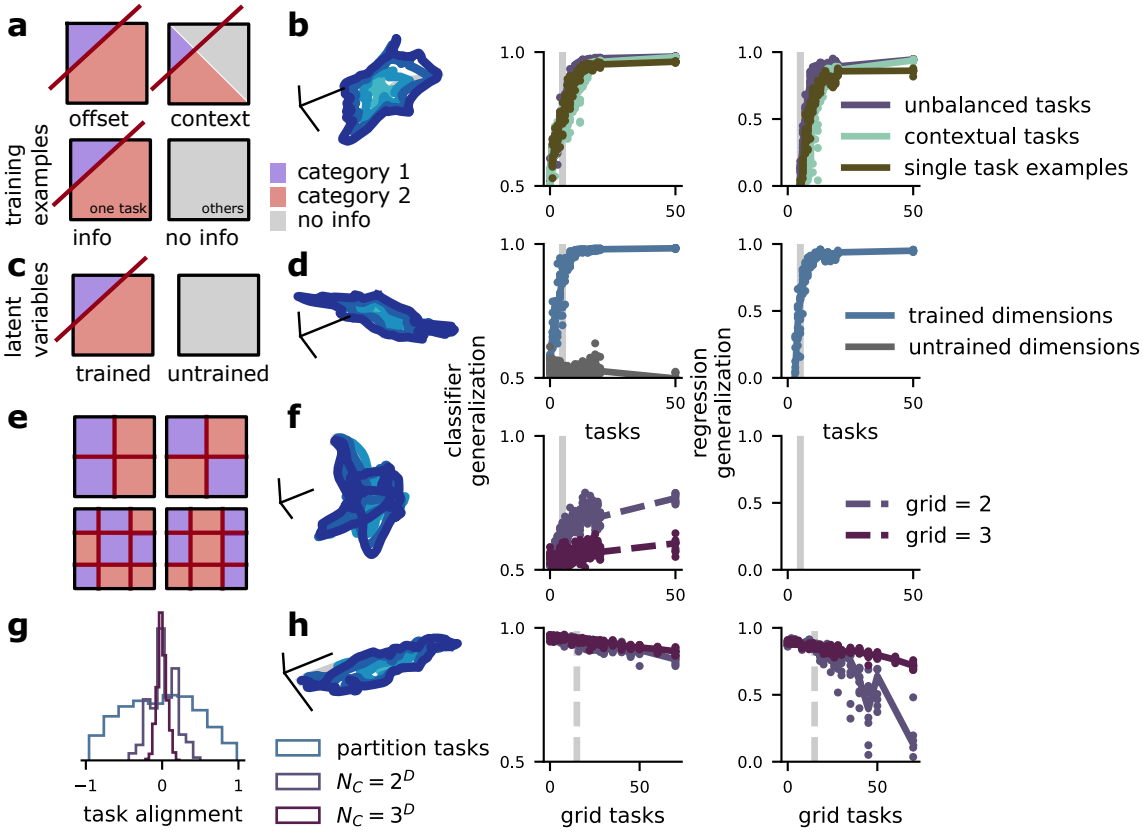

Figure 3: Abstract representations emerge for heterogeneous tasks, and in spite of high-dimensional grid tasks. **a** Schematics of different task manipulations. **b** (left) Visualization of the representations developed for contextual tasks $P = 25$. (middle) Classifier generalization performance. (right) Regression generalization performance. **c** Schematic showing the training scheme: A subset of latent variables are involved in tasks (left), the rest of the latent variables are not (right). **d** (left) Visualization of the trained latent variable representations. (middle) Classifier generalization performance for the trained and untrained latent variable dimensions. (right) Regression generalization performance for the trained and untrained latent variable dimensions. **e** Schematic of the new grid tasks. They are defined by n, the number of regions along each dimension (top: $n = 2$; bottom: $n = 3$), and the dimensionality of the latent variables, D. There are n^D total grid chambers, which are randomly assigned to category 1 (red) or category 2 (blue). Some grid tasks are aligned with the latent variables by chance (as in top left), but this fraction is small for even moderate D. **f** A multi-tasking model trained only on grid tasks. (left) Visualization of the representation. (middle) Classifier generalization performance. (right) Regression generalization performance. **g** The alignment (cosine similarity) between between randomly chosen tasks for latent variable aligned classification tasks, $n = 2$ and $D = 5$ grid tasks, and $n = 3$ and $D = 5$ grid tasks. **h** As **c**, but for a multi-tasking model trained with $P = 15$ latent variable aligned classification tasks and a variable number of grid tasks.

particular, neural receptive fields in early sensory areas have often been shown to have a unimodal and roughly Gaussian form, often for multiple stimulus

features[33, 23, 20, 17, 29]. Here, we construct a representation of a $D = 2$ latent variable using Gaussian receptive fields (fig. 4a, left) which induces a highly curved geometry in population space (fig. 4a, right). Then, we test whether or not the multi-tasking model can recover abstract representations from this highly nonlinear format. While this format is lower dimensional than the dimensionality-maximized input used previously, it is constructed to have no global structure (i.e., each neuron responds only to a local region of latent variable space). As a consequence of this receptive field-like format, almost any binary classification of the input space can be implemented with high accuracy, but the classifier generalization performance is near chance (fig. 4b, left). This is a consequence of the lack of global structure in the representation. The regression metric follows this same pattern: While the standard performance of a linear regression is relatively high, the regression generalization performance is at chance (fig. 4b, right).

Now, we train the multi-tasking model with these receptive field-like representations as input. We visualize the representations developed after training, as before, and show that an imperfect abstract structure is developed for a moderate number of classification tasks (fig. 4c, left). We compare this learned representation to the representation learned by a completely unsupervised model known as the β-variational autoencoder (βVAE), given the same input and architecture (see *Comparing the multi-tasking model with the unsupervised β VAE* in *Supplement* for more details). The βVAE does not develop strongly abstract representations for this input – and appears to retain much of the curved structure present in the input (fig. 4c, right).

We quantify this result using our classification and regression generalization metrics. As anticipated by the visualization, the classification metric saturates performance when supplied with 8 classification tasks (fig. 4d, left, blue line). The βVAE did not reach above 90 % classifier generalization performance for any value of β that we tested (fig. 4d, left, orange line). In contrast, neither model saturates performance for the regression generalization metric (fig. 4d, right). Because the multi-tasking model receives binary supervisory input and the βVAE does not receive any supervisory input at all, it is not particularly surprising that the multi-tasking model develops more abstract representations. However, we believe the contrast is still informative, as it indicates that abstract representations are unlikely to emerge by chance or without explicit training on tasks that are at least coarsely related to the latent variables of interest (and see [18]).

To understand why the multi-tasking model fails to produce high regression generalization performance, we inspect the regression residuals. This reveals that some of the input receptive field structure is still present in the representations learned by the multi-tasking model, and that this remaining structure disrupts the regression generalization performance (not shown). This can be understood by our theory: While we show that the dynamics of the representation layer increase the strength of a low-dimensional, abstract component of the stimulus representation, there is no component of the dynamics that explicitly reduces the strength of remaining high-dimensional components of the

representation. This is a problem for receptive field-like inputs in particular, as they already permit almost all classification tasks to be implemented with high accuracy without any training at all (fig. 4b, left), and thus the dynamics do not unfold for long enough to sufficiently increase the relative strength of this low-dimensional, abstract component of the representation so that it overpowers all remaining high-dimensional components. This phenomenon is likely compounded by the low dimensionality of the latent variables ($D = 2$), which show noisier learning of abstract representations for our other input type as well (see *The dependence of learned abstract representations on latent variable dimensionality* in *Supplement* and *A multiverse analysis of the multi-tasking model and βVAE* in *Supplement*). Thus, representing $D > 2$ latent variables in a particular brain region may allow the brain to more reliably learn abstract representations – though the mechanisms underlying this phenomenon are, to our knowledge, not well understood.

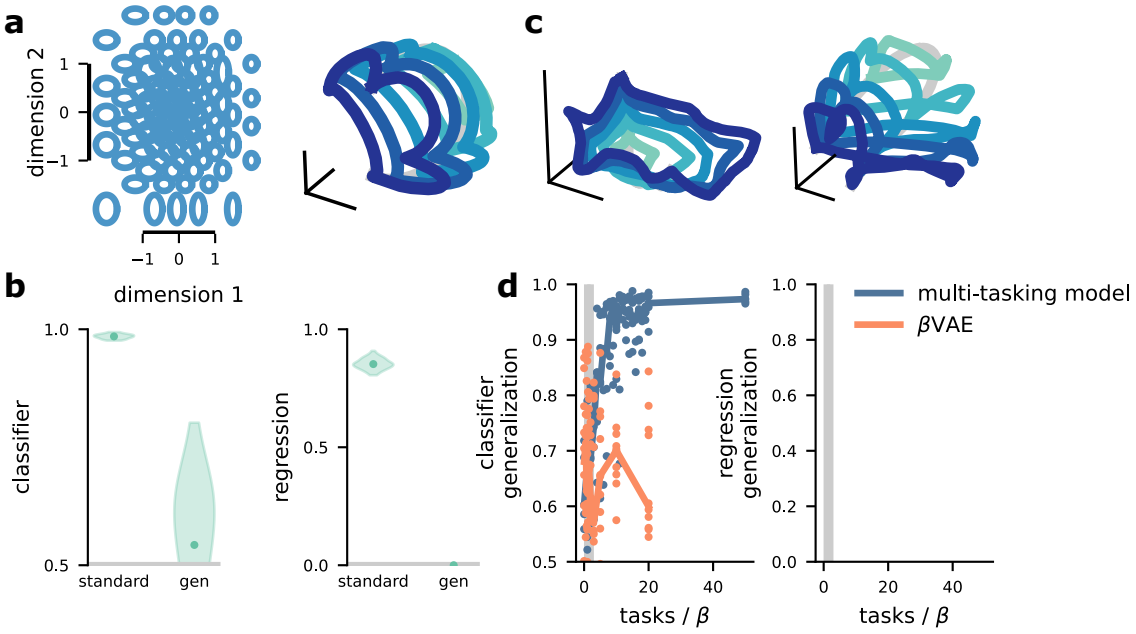

Figure 4: Abstract representations can be recovered even from highly nonlinear stimulus representations. **a** (left) Schematic of the receptive field inputs. They are arranged with density and RF width related to the probability density of the Gaussian inputs. (right) Low-dimensional projection of the RF representation, illustrating its high curvature. **b** Performance of the classifier- (left) and regression-metrics (right) on the RF inputs. **c** (left) Visualization of the dominant low-dimensional structure learned by the multi-tasking model. (right) Visualization of the dominant low-dimensional structure learned by the matched βVAE with $\beta =$. **d** Quantification of how the abstraction of the representations learned depends on the number of classification tasksfor the multi-tasking model and β for the βVAE.

2.7 The multi-tasking model can be used as an abstract, generative model

While learning abstract representations of multiple stimulus features has many advantages for neural systems, one specific application of deep learning models that are trained to produce abstract representations is their ability to be used in a generative context to produce image examples with chosen latent variable values. That is, if the representation layer of an autoencoder is abstract, once the dimensions that correspond to different features are learned, then they can be combined to generate novel examples due to their linearity. The βVAE is designed to represent each feature with the activity of a single unit in the representation layer. This is directly interpretable, but can be brittle in practice (see fig. 4and [18]). In the multi-tasking model, the representation layer has an abstract representation of the stimulus features, but we do not attempt to represent these features with the activity of single units. Instead, we learn linear transformations that reliably recover single dimensions – for example, by learning a linear regression for a feature of interest. Because the representations are abstract, once this transformation is learned for each feature, it can be viewed as equivalent to the βVAE.

Here, we compare the multi-tasking model to the βVAE in a generative context using a standard machine learning shape dataset[19]. Importantly, the multi-tasking model is supplied with categorical information that is related to the latent variables, as it is throughout the paper, so this comparison does not put the βVAE and multi-tasking model on equal footing; the βVAE is designed to develop abstract representations in a fully unsupervised setting. Further, we also modify the multi-tasking model, as described before, to add an autoencoder. Now, the multi-tasking model is trained to both satisfy the P classification tasks as well as reconstruct the original image sample to test its generative properties.

The dataset that we are using has five features that each take on several values: shape, scale, rotation, x-position, and y-position. The shape feature takes on three discrete values (heart, oval, square) while the other features all take on several continuously arranged values (see fig. 5a). First, we visualize the representation of x- and y-position developed by both models (fig. 5b, left: multi-tasking model, right: βVAE). In both cases, the representations appear to be relatively abstract. Next, we apply our metric to both the multi-tasking model and βVAE to quantify the abstractness of the representations developed by the two models for all the features (fig. 5c). While neither model saturates performance for either metric, the multi-tasking model performs better than the βVAE on both abstraction metrics. Both models produce representations that are more abstract than the input images.

Next, we test whether these abstract representations allow flexible generation of stimuli, as well as generalization in representation space. To do this, we selected one shape to be left out as a test shape, and then used the representations corresponding to the other two shapes to learn a linear regression that decodes shape scale. We then used this linear regression to generate images of one of the trained shapes at different scales (fig. 5e,f, top row). Both models

retained a reasonable degree of shape structure as well as produced an increase in scale, moving from left to right in the images shown. Next, we attempted to apply the learned representation of scale to the left out shape. Again, both models produce shapes with an increase in scale (fig. 5e,f, bottom row). However, while the multi-tasking model produces images with the left out shape, the βVAE does not represent a differentiated shape at all. This issue with the βVAE has been reported before: To achieve a high level of abstraction in the representations, the βVAE often sacrifices precision in its reconstruction of the target image[13] (but see [4]). In contrast, the multi-tasking model produces abstract representations while still preserving its ability to reconstruct different shapes from this dataset.

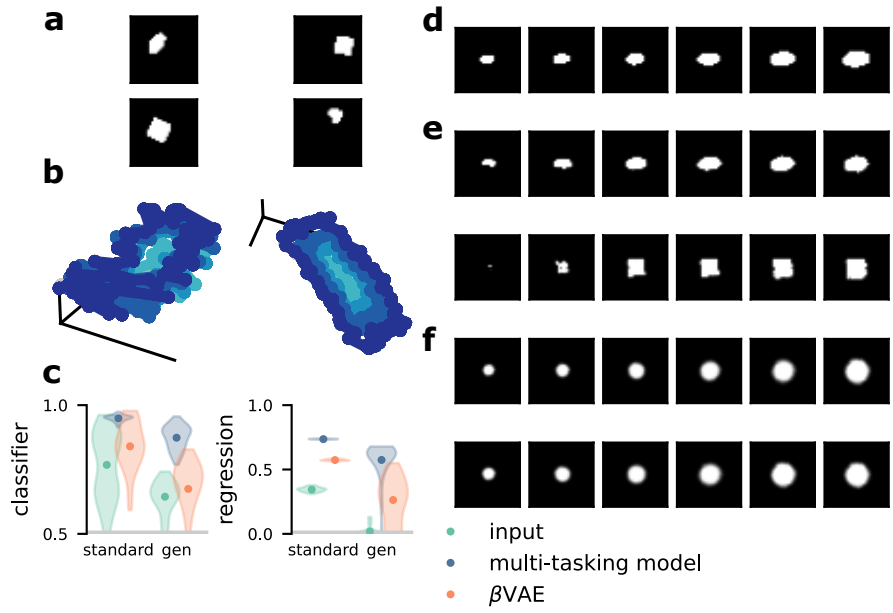

Figure 5: The multi-tasking model can be used for image generation. **a** Example images from the 2D shapes dataset. **b** Visualization of the image representation manifold for x- and y-position from the representation layers of the multi-tasking model (left) and the βVAE (right). **c** Quantification of the abstractness of the original dataset (left points), the multi-tasking model (middle points, $P = 50$), and the βVAE (right points, $\beta = 1$) according to both our classifier- (left) and regression-generalization (right) metrics. In each plot, performance when training and testing on the whole stimulus set is on the left, and training and testing on separate halves is on the right; chance for both is shown by the grey line. **d** An example traversal of the scale dimension from the image set. **e** Image reconstruction for the multi-tasking model with a shape that was present in the training set (top) and that was held out from the training set (bottom). **f** The same as **d** but for the βVAE.

3 Discussion

We demonstrate that requiring a feedforward neural network to multi-task reliably produces abstract representations. Our results center on artificial neural networks; however, we argue that abstract representations in biological neural systems could be produced through the same mechanism – as behaving organisms often need to multi-task in the same way as we have modeled here. We show that the learning of these abstract representations is remarkably reliable. They are learned even for heterogeneous classification tasks, stimuli with partial information, in spite of being required to learn additional non-latent variable aligned tasks, and for local receptive field-like stimulus representations. Further, multi-tasking more reliably produces abstract representations than the current state-of-the-art for producing abstract representations in the machine learning literature, though those models are trained in a fully unsupervised setting (that is, without the classification task information used by our model). Finally, we show that our multi-tasking model can be used in a generative context to produce samples from an image dataset with known latent features. Overall, this work provides insight into how abstract neural representations may emerge: Through the multiple constraints and complexity induced by naturalistic behavior.

We train our models to perform different binary classifications of latent variables as a proxy for different behaviors. This is, of course, a highly simplified approach. While feedforward binary classification most closely matches rapid objection recognition or, for example, go or no-go decisions, it does not provide an accurate model of behaviors that unfold over longer timescales. While most the experimental work that shows abstract representations in the brain[5, 3, 12, 28, 21] and other models that produce abstract representations in machine learning systems[16, 6, 13] have taken a static view of neural activity, network dynamics could play a role in establishing and sustaining abstract representations. While some work has shown that training recurrent neural networks to perform multiple dynamic tasks leads to shared representations of common task features, the abstractness of these representations has not been quantified[35]. Future work will probe to what degree our findings here generalize to networks trained to perform dynamic tasks.

3.1 Other methods for quantifying abstractness

Our method of quantifying abstractness in both artificial and biological neural networks has an important difference from some previously used methods[12]. In particular, the βVAE attempts to isolate the representation of single latent variables to single units in the network[13]. Directly applied to neural data, this leads to the prediction that single neurons should represent single latent variables in abstract representations[12]. Our framework differs in that generalization performance depends on the geometry of the representations at the population level and it is unaffected by whether single neuron activity corresponds to a single latent variable, or to a linear mixture (i.e., a weighted sum)

of all the latent variables. Given the extensive linear and non-linear mixing observed already in the brain[24, 5, 26, 3], we believe that this flexibility is an advantage of our framework for detecting and quantifying the abstractness of neural representations. Further, we believe that searching for abstract representations using techniques that are invariant to linear mixing will reveal abstract representations where they may not have been detected previously – in particular, a representation can provide perfect generalization performance without having any neurons that encode only a single latent variable, and thus such a representation would not be characterized as abstract by many machine learning abstraction or disentanglement metrics.

3.2 Predictions

For experimental data, our findings predict that an animal trained to perform multiple distinct tasks on the same set of inputs will develop abstract representations of the latent variable dimensions that are used in the tasks. In particular, if the tasks only rely on three dimensions from a five dimensional input, then we expect strong abstract representations of those three dimensions (as in fig. 3a, b), but not of the other two. We expect all of the dimensions to still be represented in neural activity, however – we just do not expect them to be represented abstractly. Once this abstract representation is established through training on multiple tasks, if a new task is introduced that is aligned with these learned latent variables, we expect the animal to be able to learn and generalize that task more easily than a task that relies on either the other latent variables or is totally unaligned with the latent variables (as the grid tasks above). That is, we expect animals to be able to take advantage of the generalization properties provided by abstract representations that we have focused on throughout this work, as suggested by previous experimental work in humans[32].

A recent study in which human participants learned to perform two tasks while in a functional magnetic resonance (fMRI) scanner provides some evidence for our predictions[7]. The representations of a D dimensional dimensional stimulus with two task-relevant dimensions (one which was relevant in each of two contexts) were studied in both the fMRI imaging data and in neural networks that were trained to perform the two tasks (the setup in this work is similar to certain manipulations in our study, particularly to the partial information case shown in fig. 3a, b). They find that the representations developed by a neural network which develops rich representations (similar to abstract representations in our parlance) are more similar to the representations in the fMRI data than neural networks that develop high-dimensional, non-abstract representations. This provides evidence for our central prediction: That abstract representations emerge through learning to multi-task. However, the conditions explored in the human and neural network experiments in the study were more limited than those explored here. In particular, only two tasks were performed, the stimulus encoding was less nonlinear than in our studies, and the tasks were always chosen to be orthogonal. Thus, further work will be necessary to determine the limits of our finding in real brains.

Several additional predictions can be made from our results with the grid tasks, which showed that learning many random, relatively uncorrelated tasks both does not lead to the development of abstract representations alone, but also does not interfere with abstract representations that are learned from a subset of tasks that are aligned with the latent variables. First, if an animal is trained to perform a task analogous to the grid task, then we do not expect it to show abstract representations of the underlying latent variables – this would indicate that latent variables are not inferred when they do not support a specific behavior. Second, we predict that an animal trained to perform some tasks that are aligned to the latent variables as well as several (potentially more) non-aligned grid task analogues will still develop abstract representations. Both of these predictions can be tested directly through neurophysiological experiments as well as indirectly through behavioral experiments in humans (due to the putative behavioral consequences of abstract representations[32]).

3.3 Conclusions

Overall, our work indicates that abstract representations in the brain – which are thought to be important for generalizing knowledge across contexts – emerge naturally from learning to perform multiple categorizations of the same stimuli. This insight helps to explain previous observations of abstract representations in tasks designed with multiple contexts (such as [3]), as well as makes predictions of conditions in which abstract representations should appear more generally.

Acknowledgments: We thank Mattia Rigotti, Nicolas Masse, and Matthew Rosen for comments on an earlier version of this manuscript. This work was supported by the following grants: Simons Foundation 542983SPI, Neuronex NSF 1707398, and Gatsby Charitable Foundation GAT3708.

Author contributions: WJJ and SF conceived of the project and developed the simulations. WJJ performed the simulations and analytical calculation. WJJ analyzed the simulation results and made the figures. WJJ and SF wrote and edited the paper.

Competing interests: The authors declare no competing interests.

4 Methods

4.1 Code

All of our code for this project is written in Python, making extensive use of TensorFlow[1] and the broader python scientific computing environment (including numpy[11], scipy, matplotlib, and scikit-learn[22]).

The code is available at the following link: https://github.com/wj2/disentangled

4.2 Abstraction metrics

Both of our abstraction methods quantify how well a representation that is learned in one part of the latent variable space (e.g., a particular context) generalizes to another part of the latent variable space (e.g., a different context). To make this concrete, in both metrics, we train a decoding model on representations from only one – randomly chosen – half of the latent variable space and test that decoding model on representations from the non-overlapping half of the latent variable space.

4.2.1 The classifier generalization metric

First, we select a random balanced division of the latent variable space. One of these halves is used for training, the other is used for testing. Then, we select a second random balanced division of the latent variable space that is orthogonal to the first division. One of these halves is labeled category 1 and the other is labeled category 2. As described above, we train a linear classifier on this categorization using 500 training stimuli from the training half of the space, and test the classifier's performance on 500 stimuli from the testing half of the space. Thus, chance is set to .5 and perfect generalization performance is 1.

4.2.2 The regression generalization metric

As above, except we train a linear ridge regression model to read out all D latent variables using 500 sample stimulus representations from the training half of the space. We then test the regression model on 500 stimulus representations sampled from the testing half of the space. We quantify the performance of the linear regression with its r^2 value:

$$r^2 = 1 - \frac{\text{MSE}(X, \hat{X})}{\text{Var}(X)} \tag{5}$$

where X is the true value of the latent variables and $\hat{X}$ is the prediction from the linear regression. Because the MSE is unbounded, the r^2 value can be arbitrarily negative. However, chance performance is $r^2 = 0$, which would be the performance if the linear regression always predicted the mean of X, and $r^2 = 1$ indicates a perfect match between the true and predicted value.

4.3 Non-abstract input generation

In the main text, we use two methods for generating non-abstract inputs from a D-dimensional latent variables. We have also performed our analysis using several other methods, which we also describe here.

4.3.1 Participation ratio-maximized representations

We train a symmetric autoencoder (layers: 100, 200 units) to maximize the participation ratio[10] in its 500 unit representation layer. The participation

ratio is a measure of embedding dimensionality that is roughly equivalent to the number of principal components that it would take to capture 80 % of the total variance. The autoencoder ensures that information cannot be completely lost, while the participation ratio regularization ensures that the representation will have high-embedding dimension and, therefore, be non-abstract. The performance of our generalization metrics on this input representation is shown in fig. 1f.

4.3.2 Receptive field-style representations

This input transformation is constructed to be analogous to the receptive field representations observed in many early sensory areas[33, 23, 20, 17, 29]. In particular, where single units respond most strongly for a particular conjunction of the D latent variables, and their response falls off exponentially as distance from that center point increases.

In this case, we arrange receptive field centers to tile the probability distribution of the D-dimensional latent variables (i.e., for Gaussian latent variables there will be more receptive field centers clustered in the center, where the probability density is higher). The width of the receptive fields is inversely related to probability density (i.e., receptive fields are wider away from the center of the latent variable distribution). The receptive fields have a Gaussian shape. That is, for a receptive field i with center μ_i and width w_i where both of these parameters are vectors in the D-dimensional latent variable space. The response of a receptive field unit i is given by

$$\mathrm{RF}_i(x) = \exp\left(-\sum_j^D \frac{(x_j - \mu_{ij})^2}{w_{ij}}\right) \tag{6}$$

Receptive fields are particularly non-abstract, as shown by the performance of our generalization metrics directly on the receptive field representations, which is shown in fig. 4b.

4.4 The multi-tasking model

We primarily study the ability of the multi-tasking model to produce abstract representations according to our classification and regression generalization metrics. The multi-tasking model is a feedforward neural network. For figs. 2 and 3 it has the following parameters:

layer widths	250, 150, 100, 50
representation width	50
batch size	100
training examples	10000
epochs	200

For fig. 4, everything is kept the same except the number of layers is increased:

layer widths	250, 200, 100, 100, 50, 50

4.4.1 Full task partitions

In all cases, the models are trained to perform multiple tasks – specifically, binary classification tasks – on the latent variables. In the simplest case (i.e., fig. 2e), the task vector can be written as,

$$T(x) = \text{sign}\, Ax \tag{7}$$

where A is a $P \times D$ matrix with randomly chosen elements.

4.4.2 Unbalanced task partitions

For unbalanced partitions, the task vector has the following simple modification,

$$T(x) = \text{sign}\left[Ax + b\right] \tag{8}$$

where b is a P-length vector and $b_i \sim \mathcal{N}(0, \sigma_{\text{offset}})$. Notice that this decreases the average mutual information provided by each element of $T(x)$ about x.

4.4.3 Contextual task partitions

We chose this manipulation to match the contextual nature of natural behavior. As motivation, we only get information about how something tastes for the subset of stimuli that we can eat. Here, we formalize this kind of distinction by choosing P classification tasks that each only provide information during training in half of the latent variable space.

We can write each element i of the contextual task vector as follows,

$$T_i(x) = \begin{cases} \text{sign}\left[A_i x + b_i\right] & C_i x > 0 \\ \text{nan} & C_i x \leq 0 \end{cases} \tag{9}$$

where nan values are ignored during training and C is a $P \times D$ random matrix. Thus, each of the classification tasks influences training only within half of the latent variable space. This further reduces the average information provided about x by each individual partition.

4.4.4 Partial information task partitions

For contextual task partitions, the contextual information acts on particular tasks. For our partial information manipulation, we take a similar structure, but it instead acts on specific training examples. The intuitive motivation for this manipulation is to mirror another form of contextual behavior: At a given moment (i.e., sampled training example) an animal is only performing a subset of all possible tasks P. Thus, for a training example from that moment, only a subset of tasks should provide information for training.

Mathematically, we can write this partial information structure as follows. For each training example x, the task vector is given by,

$$T_i(x) = \begin{cases} \text{sign}\left[A_i x + b_i\right] & p \geq M \\ \text{nan} & p < M \end{cases} \tag{10}$$

where p is a uniformly distributed random variable on $[0, 1]$, which is sampled uniquely for each training example x and M is a parameter also on $[0, 1]$ that sets the fraction of missing information. That is, $M = .9$ means that, for each training example, 90 % of tasks will not provide information.

While results are qualitatively similar for many values of M, in the main text we use a stricter version of this formalization: For each training sample, one task is randomly selected to provide information and the targets for all other tasks are set to nan.

4.4.5 Grid classification tasks

The grid tasks explicitly break the latent variable structure. Each dimension is broken into n parts with roughly equal probability of occurring (see schematic in fig. 3a). Thus, there are n^D unique grid compartments, each of which is a D-dimensional volume in latent variable space, and each compartment has roughly equal probability of being sampled. Then, to define classification tasks on this space, we randomly assign each compartment to one of the two categories – there is no enforced spatial dependence.

4.4.6 The dimensionality of representations in the multi-tasking model

First, we consider a deep network trained to perform P balanced classification tasks on a set of D latent variables $X \sim \mathcal{N}(0, I_D)$. We focus on the activity in the layer just prior to readout, which we refer to as the representation layer and denote as $r(x)$ for a particular $x \in X$. This representation layer is connected to the P output units by a linear transform W. In our full multi-tasking model, we then apply a sigmoid nonlinearity to the output layer. To simplify our calculation, we leave that out here. The network is trained to minimize error, according to a loss function which can be written as:

$$E = \frac{1}{2}\left[\text{sign}(Ax) - Wr(x)\right]^2 \tag{11}$$

where A is a $P \times D$ matrix of randomly selected partitions (and it is assumed to be full rank). To understand how $r(x)$ will change during training, we write the update rule for r (to be achieved indirectly by changing preceding weights),

$$r(x)^{s+1} = r(x)^s - \mu \frac{\partial E}{\partial r(x)} \tag{12}$$

$$= r(x)^s + \mu W^T \text{sign}(Ax) - \mu W^T W r(x) \tag{13}$$

Thus, we can see that, over training, $r(x)$ will be made to look more like a linear transform of the target function, $\text{sign}(Ax)$. Next, to link this to abstract representations, we first observe that Ax produces an abstract representation of the latent variables X. Then, we show that $\text{sign}(Ax)$ has approximately the same dimensionality as Ax. In particular, the covariance matrix $M = E_X\left[\text{sign}(Axx^TA^T)\right]$ has the elements,

$$M_{ij} = 1 - \frac{2}{\pi}\arccos A_i A_j \tag{14}$$

where A_i is the ith row of A. To find the dimensionality of $\text{sign}(Ax)$ we need to find the dimensionality of M. First, the distribution of dot products between random vectors is centered on 0 and the variance scales as $1/D$. Thus, we can Taylor expand the elements of the covariance matrix around $A_iA_j = 0$, which yields

$$M_{ij} \approx \frac{2}{\pi}A_i A_j \tag{15}$$

We identify this as a scalar multiplication of the covariance matrix for the linear, abstract target $E_X\left[Axx^TA^T\right]$. Further, we know that the rank of this matrix is $\min(P, D)$. So, this implies that the matrix M also has rank approximately $\min(P, D)$. Deviations from this approximation will produce additional non-zero eigenvalues, however they are expected to be small.

4.5 The βVAE

The βVAE is an autoencoder designed to produce abstract (or, as referred to in the machine learning literature, disentangled) representations of the latent variables underlying a particular dataset[13]. The βVAE is totally unsupervised, while the multi-tasking model receives the supervisory task signals. Abstract representations are encouraged through tuning of the hyperparameter β, which controls the strength of regularization in the representation layer, which penalizes the distribution of representation layer activity for being different from the standard normal distribution.

In fig. S.1, the βVAE is trained with the same parameters as given in section 4.4 – the layers are replicated in reverse for the backwards pass through the autoencoder. For fig. 4, the parameters are as described in section 4.4. In both cases, instead of fitting models across different numbers of partitions, we fit the models with different values chosen for β.

For fig. 5, parameters for the βVAE are as described in section 4.6. We also explored numerous other architectures for the βVAE in that figure, but never obtained qualitatively or quantitatively better results.

4.6 The generative multi-tasking model

To move our multi-tasking model into a generative context, we simply add a series of layers connected to the representation layer that are trained to reproduce the original stimulus. Our objective function then has two parts: The first

is to satisfy the training classification tasks and the second is to reconstruct the original input, as with a traditional autoencoder. The generative multi-tasking model was trained on the 2D shapes dataset (which were resized to be 32×32 images) with the following parameters,

layer widths	128x2x2, 128x2x2, 512, 256, 128, 128
representation width	50
batch size	30
training examples	100000
epochs	200

For the reconstruction part of the model, the given layer list is reversed.

5 Supplement

5.1 Comparing the multi-tasking model with the unsupervised βVAE

We compare the level of abstraction of the representations learned by the multi-tasking model to those learned by an auto-encoder that is designed to produce abstract representations. In particular, the β-variational autoencoder (βVAE) is the current state-of-the-art for unsupervised disentangling of latent variables[13] (and it has many variations[4, 15]). It is designed around a hyperparameter, β, that is thought to control the trade-off between the abstractness of the representations in the latent variables and reconstruction error for output from the auto-encoder. That is, increasing β is understood to increase the level of abstraction in the βVAE representation layer, while decreasing the quality of of reconstruction of the original input representation.

Using the same architecture as in our multi-tasking model, we trained βVAEs to disentangle the same set of latent variables as in our other experiments. Applying the same two metrics as to our other models, we found that the βVAE produces moderately abstract representations, as quantified by the classifier generalization metric – though classifier generalization performance does not saturate to the same level as for the multi-tasking model. The βVAE does not produce high regression generalization performance for any choice of β that we tested. Because the multi-tasking model receives binary supervisory input and the βVAE does not receive any supervisory input at all, it is not particularly surprising that the multi-tasking model develops more abstract representations. However, we believe the contrast is still informative, as it indicates that abstract representations are unlikely to emerge by chance or without explicit training on tasks that are at least coarsely related to the latent variables of interest (and see [18]). Further, this multi-tasking approach to producing abstract representations is less sensitive to changes in model and input parameters than the βVAE (see *A multiverse analysis of the multi-tasking model and βVAE* in *Supplement*). This further indicates the feasibility of the multi-tasking approach in conditions similar to those found in the brain.

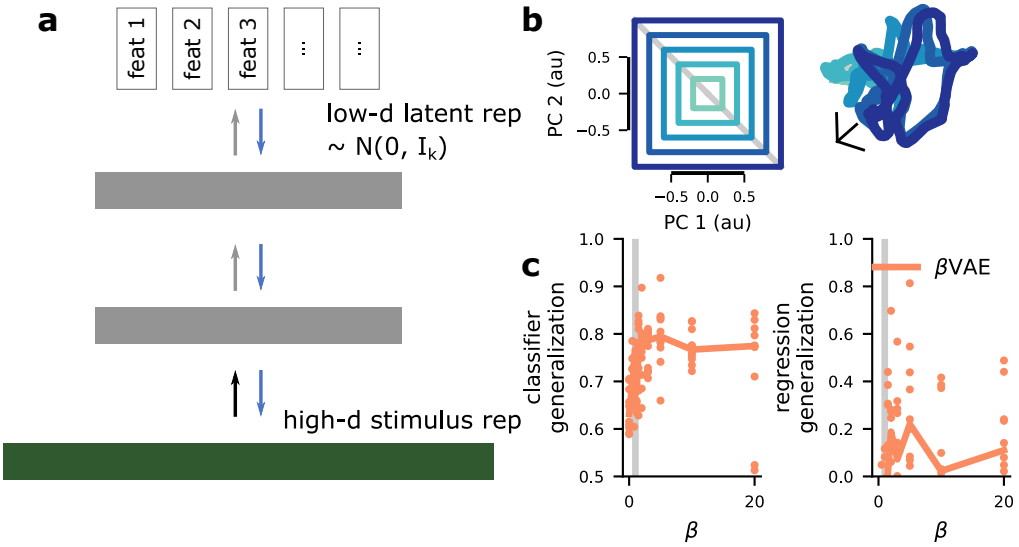

Figure S.1: The βVAEdoes not reliably produce abstract representations. **a** A schematic of the βVAE. It is an autoencoder regularized to produce a low-dimensional representation in its representation layer. **b** A purely auto-encoding approach with the βVAE is not supplied with any classification tasks (left), but does produce moderately abstract representations (right). **c** The βVAE produces high classifier generalization performance for a small range of βs (left), but does not provide high regression generalization performance for any choice of β that we tested (right).

5.2 The dependence of learned abstract representations on latent variable dimensionality

For both the multi-tasking model and βVAE, our simulations reveal that abstract representations are more readily and consistently produced for higher-dimensional latent variables (fig. S.2). We believe that this is due to a decrease in dimensionality expansion per latent dimension as D increases. In particular, for each of $D = 2, 3, 4, 5$, the participation ratio after expansions is approximately 200 and the participation ratio per latent variable dimension is approximately $\frac{200}{D}$. However, further work is necessary to confirm this intuition.

5.3 A multiverse analysis of the multi-tasking model and βVAE

While we have focused on manipulation of the number (and kind) of classification tasks provided to the multi-tasking model and to the value of β for the βVAE, both models depend on numerous other parameter choices, which were essentially arbitrary. The parameters were held constant across the two models, but these choices can still affect the results produced by both models in different ways. To explore the dependence of our results on these other parameter choices, we performed a multiverse analysis[]. That is, for many of the parame-

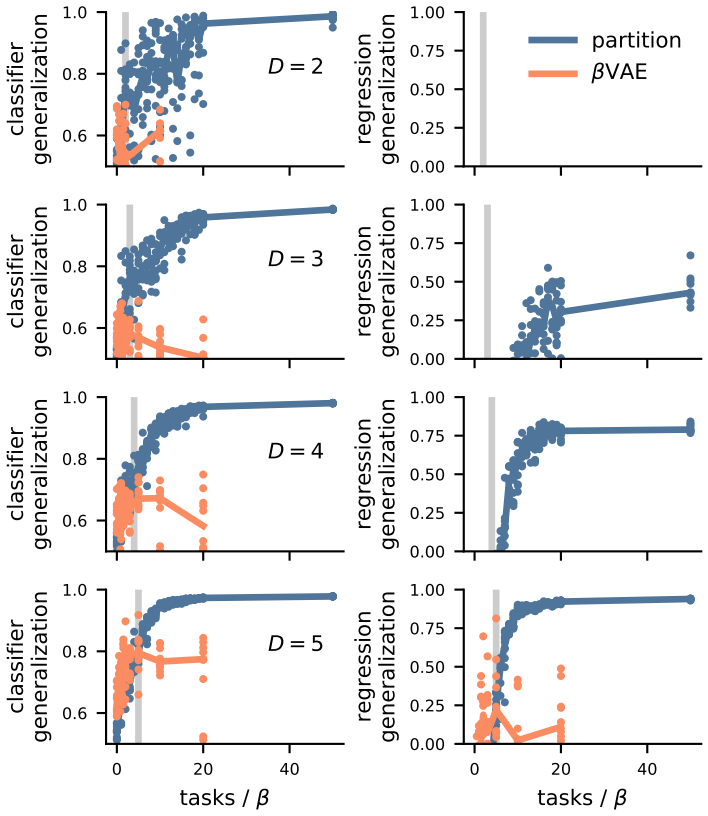

Figure S.2: Abstraction learning depends on latent variable dimensionality. (top to bottom) Increasing latent variable dimensionality D, from $D = 2$ to $D = 5$ (see left inset). (left) Classifier generalization performance as a function of the number of classification tasks for the multi-tasking model and β for the βVAE. (right) Regression generalization performance as a function of the number of classification tasks for the multi-tasking model and β for the βVAE.

ters of our models, we chose several other similarly reasonable parameter values, and the trained models with those parameters (e.g., using a tanh nonlinearity instead of the ReLU). In exploring this parameter space, we defined 7128 and 3369 distinct combinations of parameters for the multi-tasking model and βVAE respectively. Then, for each of these parameter combinations, we trained two models of the corresponding type and averaged their classification and regression generalization performance. The parameters varied for each were the same except for the choices of the number of partitions, values of β, and we included a version of the multi-tasking model with and without an autoencoder.

To analyze these results, we fit linear models with ridge regression to account for the classification and regression generalization performance from the different parameter choices. Using only the first order version of this model (that is,

without fitting interaction terms for the different parameters), the model has $r^2 = .62$ and $r^2 = .70$ for the multi-tasking model and βVAE, respectively. As expected, for the multi-tasking model, the number of classification tasks has by far the strongest effect on both classification and regression generalization performance (fig. S.3a,b), though minor effects on both are produced by almost all the other parameter choices – and the regression generalization metric is strongly affected by the dimensionality of the latent variables (fig. S.3b). Surprisingly, while choice of β does affect classification and regression generalization performance for the βVAE, the size of the effect is similar in size to the effects associated with many of the other parameters – and much smaller than the increase in both classification and regression generalization performance that is produced by using a tanh nonlinearity rather than a ReLU nonlinearity.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

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

[4] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[5] Le Chang and Doris Y Tsao. The code for facial identity in the primate brain. *Cell*, 169(6):1013–1028, 2017.

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016.

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

[8] David J Freedman and John A Assad. Experience-dependent representation of visual categories in parietal cortex. *Nature*, 443(7107):85, 2006.

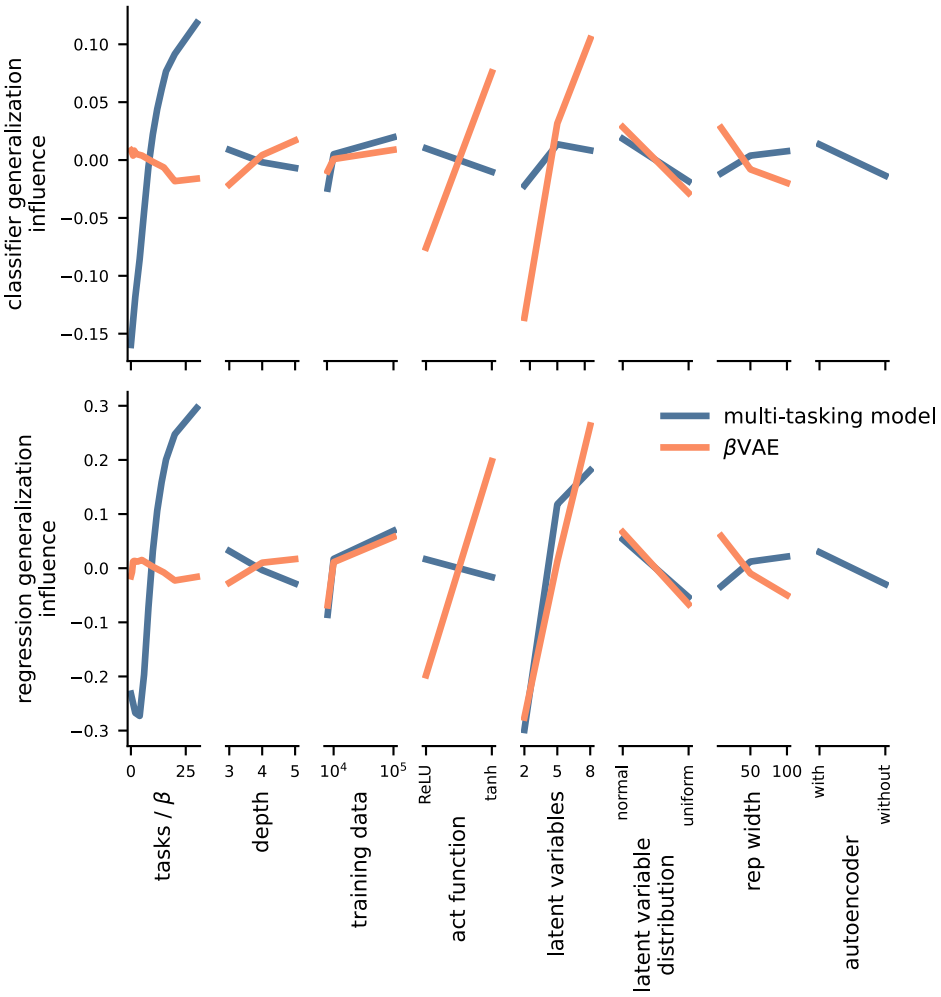

Figure S.3: A multiverse analysis of both the multi-tasking model and βVAE. (top) The effects of different parameter choices on classifier generalization performance. (bottom) The effects of different parameter choices on regression generalization performance.

[9] Stefano Fusi, Earl K. Miller, and Mattia Rigotti. Why neurons mix: High dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37:66–74, 2016.

[10] Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *BioRxiv*, page 214262, 2017.

[11] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus,

Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[12] Irina Higgins, Le Chang, Victoria Langston, Demis Hassabis, Christopher Summerfield, Doris Tsao, and Matthew Botvinick. Unsupervised deep learning identifies semantic disentanglement in single inferotemporal neurons. *arXiv preprint arXiv:2006.14304*, 2020.

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[14] W Jeffrey Johnston, Stephanie E Palmer, and David J Freedman. Nonlinear mixed selectivity supports reliable neural computation. *PLoS computational biology*, 16(2):e1007544, 2020.

[15] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[16] Tejas D Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B Tenenbaum. Deep convolutional inverse graphics network. *arXiv preprint arXiv:1503.03167*, 2015.

[17] Michael S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4), 2002.

[18] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[19] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset, 2017.

[20] Bruno A. Olshausen and David J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.

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

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Javier Perez-Orive, Ofer Mazor, Glenn C. Turner, Stijin Cassenaer, Rachael I. Wilson, and Gilles Laurent. Oscillations and Sparsening of Odor Representations in the Mushroom Body. *Science*, 297:359–365, 2002.

[24] David Raposo, Matthew T Kaufman, and Anne K Churchland. A category-free neural population supports evolving demands during decision-making. *Nature Neuroscience*, 17(12):1784–1792, 2014.

[25] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.

[26] Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):1–6, 2013.

[27] Liang She, Marcus K Benna, Yuelin Shi, Stefano Fusi, and Doris Y Tsao. The neural code for face memory. *bioRxiv*, 2021.

[28] Hannah Sheahan, Fabrice Luyckx, Stephanie Nelli, Clemens Teupe, and Christopher Summerfield. Neural state space alignment for magnitude generalization in humans and recurrent networks. *Neuron*, 109(7):1214–1226, 2021.

[29] Evan C. Smith and Michael S. Lewicki. Efficient auditory coding. *Nature*, 439:978–982, 2006.

[30] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, page 1, 2019.

[31] Sruthi K Swaminathan and David J Freedman. Preferential encoding of visual categories in parietal cortex compared with prefrontal cortex. *Nature neuroscience*, 15(2):315–320, 2012.

[32] Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? *arXiv preprint arXiv:1905.12506*, 2019.

[33] William E Vinje and Jack L Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–1276, 2000.

[34] Guangyu Robert Yang, Michael W Cole, and Kanaka Rajan. How to study the neural mechanisms of multiple tasks. *Current opinion in behavioral sciences*, 29:134–143, 2019.

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