PNAS
www.pnas.org

Supplementary Information for
Comparing continual task learning in minds and machines

Timo Flesch, Jan Balaguer, Ronald Dekker, Hamed Nili and Christopher Summerfield

Email: timo.flesch@psy.ox.ac.uk

This PDF file includes:

Supplementary text
Figs. S1 to S8
Tables S1 to S3

www.pnas.org/cgi/doi/10.1073/pnas.1800755115

Supplementary Methods

1) Experiments 1,2

1.1) Participants. The study was approved by the University of Oxford Central University Research Ethics Committee (approval R50750/RE001). Participants were adults who provided consent prior to taking part in the experiment in accordance with local ethics guidelines. We recruited a large cohort (n = 768) of participants via the online platform provided by Amazon Mechanical Turk with restrictions to the US (Exp.1) and India (Exp.2). We set a criterion of >55% accuracy at test for inclusion in the analysis, and continued to recruit participants until we reached at least n = 40 in each training group; attrition rates were high but comparable with previous reports (1), presumably because many participants were unused to conducting online tasks that involved uninstructed rule discovery. In total, we included 352 male and 231 female participants (Table S1), with a mean age for included participants of 33.33 (range 19-55); ages did not differ reliably between groups (Table S2).

1.2) Stimuli. Trees were generated with a custom fractal tree generator, written in Python 2.7 with the PIL package. The two feature dimensions were each varied parametrically in five discrete steps, spanning a two-dimensional space of leafiness x branchiness. We manipulated branchiness in terms of the branching angle, branching probability and the total number of recursive calls of the generator. Leafiness was defined as total number of leaves per branch, relative to branch length. Furthermore, positioning of individual leaves, leaf size, rotation and individual branch colour tones were varied probabilistically. For human studies, we created independent training and test datasets with 8 unique trees for each level of branchiness and leafiness (5 x 5 levels). Thus, the same trees were shown for both tasks, to rule out low-level differences in saliency as influence on accuracy. Furthermore, the use of different trees for training and test ensured that participants could not rote learn the mappings between individual trees and responses. Each tree was shown once per task during training, and each test tree was unique and novel. The data set was normed to equate difficulty along the two dimensions in a pilot study not reported here.

1.3) Task and Procedure. Exp.1 and 2 were written in JavaScript and run in forced-fullscreen mode. We tested software compatibility with a variety of operating systems and web browsers before launching the task. All experiments consisted of a training phase (400 trials) and a test phase (200 trials). Self-paced breaks were allowed after 200 and 400 trials. We equated the number of presentations of each stimulus (leafy x branchy level). For Exp. 2, a similarity rating task was added before and after the main task (details below). At the beginning of the experiments, the participants received written instructions. They were asked to imagine that they owned two gardens, one in the north and one in the south. The goal was to learn via trial and error which type of trees grow best in each garden. Our instructions thus avoided alerting participants to the task-relevant dimensions (leafiness and branchiness).

In the training block, each trial began with the presentation of a centrally presented garden image (task context) for 500ms. Subsequently this garden image was blurred to direct the attention towards the overlaid tree stimulus that was presented for up to 2000ms. Responses were allowed for 4500ms post stimulus onset. As a reminder, the response assignments (e.g. keys for “plant” and “reject”) were displayed above the tree stimulus through the trial (counterbalanced). Feedback was displayed for 1500ms. Feedback on “plant” trials during training depended on the distance to the relevant boundary (±50 points for levels 1 and 5, ±25 points for levels 2 and 4 and 0 points for level 3). Reward was zero on reject trials. On plant trials, feedback was also accompanied by an animation (during which the garden was unblurred) that showed the tree grow or shrink proportionally to the magnitude of the received reward. These timings were identical in test blocks, with the exception that no feedback was provided.

In Exp.1a and Exp.2a, the correct decision depended on leafiness for one garden and branchiness for the other ("cardinal" boundary). We fully counterbalanced the rule, so that equal number of participants received positive rewards for planting trees that were leafy/non-leafy and branchy/non-branchy. In other words, across participants, the assignment of rewards was flipped for either one or two dimensions (e.g. more branchy trees might yield positive rewards for participant 1, but less branchy trees for participant 2). In Exp.1b and Exp. 2b, the decision boundary was rotated by 45 degrees, to align with the diagonal axes of the branch-x-leaf stimulus space. Thus, instead of learning the mapping from single feature dimensions onto rewards, participants had to learn the mapping of cardinal feature *combinations* onto rewards ("diagonal" boundary). Once again, the rules were counterbalanced across participants.

The test phase consisted of 200 trials in which the two contexts (gardens) were interleaved randomly over trials (100 instances of each). In Exp.1, we trained participants in 4 conditions, each involving 200 north and 200 south gardens in different order. In the interleaved condition, gardens were randomly interleaved over trials. In the B2, B20, and B200 conditions, gardens remained constant over 2, 20 or 200 trials (the first garden was selected at random). In Exp.2, we included only the B200 and interleaved training conditions.

1.4) Arena task (Exp. 2). In Exp.2a and 2b we additionally asked participants to rate the similarity among pools of 25 trees before and after the main experiment (2). We presented participants with a grey circular arena covering most of the screen area. This was populated by 25 trees (one per level of leafiness/branchiness) that were positioned in a random but non-overlapping configuration. Participants were asked to move the trees around by mouse drag-and-drop, arranging the trees so that more similar exemplars were close together and more dissimilar exemplars were further apart. The choice of a two-dimensional arena may have biased the participants towards reporting a two-dimensional dissimilarity structure, or, in other words, to report only variation along the two dimensions they found most salient (3). As the trees varied systematically along two dimensions (branchiness and leafiness), and a perfect description of the ground-truth dissimilarity structure could be provided in 2D, we refrained from using more sophisticated sub-sampling methods that allow participants to express higher dimensional dissimilarity structures (4). The opacity of the selected tree was changed during the dragging process for usability purposes. This procedure was self-paced. Once all trees had been arranged and the participant was satisfied with the outcome, the next trial could be initiated by clicking on a designated button. Each trial involved a unique tree set. Participants performed 5 trials in total, taking an average of ~10 minutes. The main task was identical to Exp.1a and 1b, with the exception that we omitted the B20 and B2 conditions.

1.5) Statistical Tests. Accuracies were compared with ANOVAs and t-tests. Other measures were compared with nonparametric tests (Kruskal Wallis, Rank Sum and Sign Rank), as violations of Gaussianity were observed. We calculated Cohen's d and z/sqrt(N) as measures of effect size for parametric and nonparametric post-hoc tests respectively.

1.6) Accuracy and Psychometrics. To display learning curves, we plotted training accuracy in bins of 50 trials, and test accuracy averaged over the entire 200 trials. For psychometric curves, for each participant we calculated p(plant) during the test phase for each of the 5 levels of the relevant and irrelevant dimensions and fit a logistic function to the data.

1.7) Behavioural RSA. For RSA (5, 6), we calculated p(plant) as a function of every level of leafiness x branchiness, and then computed an RDM expressing the dissimilarity (in accuracy) between each pair of leaf x branch level. We compared these to model-predicted RDMs that were generated to match a theoretically perfect observer (model 1) or an observer who learned the best possible single linear boundary through the 2D space and applied it to both tasks (model 2). To deal with partial correlation among these predictor matrices, they were orthogonalised using a recursive Gram-Schmidt approach. We computed Kendall's Tau correlations between predicted and observed RDMs and compared these using a nonparametric approach.

1.8) Psychophysical Model. We built a full psychophysical model of the task. In a first step, we assumed that each stimulus was initially represented by a decision variable that was proportional to its distance to a boundary in leaf x branch space. The boundary had signed angle φ with respect to the correct boundary. This value was converted to a choice probability via a logistic function with slope s, bias b and lapse parameter ε (**Fig. S3**). We first ran a parameter recovery study to verify that the parameters were identifiable. We compared two variants of this model. An unconstrained model, which had two boundaries and two sigmoids, one for each task (eight parameters) and a constrained model with only one boundary and one sigmoidal transducer (four parameters) The two models were fit to the human data via maximum likelihood estimation. We compared the best-fitting parameters across groups using nonparametric statistics. Model comparisons were conducted with random effects Bayesian model selection, for which we approximated the log model evidences with the subject-specific negative BIC scores. The model comparison itself was conducted using the VBA (7) to obtain exceedance probabilities (EPs, the probability that one model is more frequent than the alternatives) and estimated model frequencies (the relative percentage of subjects explained with each model). We report protected exceedance probabilities, which correct the original Eps for the possibility that differences are due to chance (8) and compared them with nonparametric statistics.

1.9) Task Lapse Analysis (Exp 1,2). To test whether the observed intrusion along the irrelevant task dimension was rather driven by confusion of rules (with perfectly learned boundaries) than biases in the boundary estimation, we performed two additional analyses. First, we generated synthetic data from the 2-boundary model using the best-fitting parameters estimated at single subject level, and repeated the intrusion analysis on this data by fitting sigmoidal curves to the choice probabilities as a function of the reward a participant had received if the rules were flipped (e.g. he/she had categorised a task A trial using the task B rule). Next, we developed a variant of the psychophysical model for which the boundaries were fixed to their optimal values, but introduced a "task lapse" parameter that modelled the probability to apply the wrong rule to a given context. We performed Bayesian model comparison between this model and the unconstrained 2-boundary model, to investigate whether the data might be best described by task lapses or boundary estimation errors. Results are illustrated in Fig S4.

1.10) Influence of Priors. In Exp.2, for each participant and condition we calculated average pairwise Euclidean distances between trees for each pair of dissimilarity ratings, yielding an RDM of the same size as that described above (25 x 25). This yielded three trial-specific RDMs per subjects. We averaged normalised RDMs within subjects across trials to obtain a reliable per-subject estimate of the reported dissimilarities. For visualisation, we then compressed this RDM to two dimension using multidimensional scaling. To compute the "grid" prior, for each participant we constructed a model RDM which exhibited perfect grid-like encoding of the two feature dimensions and correlated it with the empirical single-subject RDMs obtained from the arena task the participants engaged in prior to the main experiment. (Kendall tau_a rank correlation). We then repeated the analyses of test accuracy and the RDM analyses described above separately for participants with high and low griddiness prior, as indicated by a median split. We also conducted ANCOVA analysis on the Kendall's Tau correlations with grid prior as a covariate of interest.

2) Experiments 3

All models were implemented in python 3.5 with tensorflow 1.0 and trained in GPU mode on an Nvidia Tesla K80 GPU. A detailed description of all models and analysis procedures follows below.

2.1) Exp 3: CNN Simulations of Trees task *without* priors

In this experiment, we replicated the trees study with a convolutional neural network (CNN), which learned solely from pixels and a reward signal. The goal was to compare human performance and choice biases with the effects of blocked vs interleaved learning on its hidden layer representations and test performance, using the same task. The single-trial inputs were RGB images of gardens with trees. A key-feature of CNNs are convolutional layers, which consist of a set of filters, or kernels, that are convolved with the input to create an output of filter-specific features. As a CNN is a fully-parametrised model, these filters are not hand-crafted but learned solely by optimising the loss function (see below) via the backpropagation procedure.

2.1.1) Stimuli. We generated 50000 training and 10000 test tree images using exactly the same settings as for the behavioural experiments. The tree images were downscaled to 96x96x3 and saved as .png RGB images.

2.1.2) Network Architecture. The network architecture is illustrated in Fig. S6a.

2.1.3) Training. Training was conducted online with one sample per "trial". Each training trial consisted of an RGB image, composed of a task-specific image of a garden and a superimposed tree, which served as input. Convolutions with filters and downsampling operations processed the image until it was finally passed through a sigmoidal transducer function to generate a choice probability. We used the trial-specific reward, multiplied by -1 times the choice probability, as custom loss function. Thus, by learning convolutional filters and network weights to minimise the trial-wise loss (negative reward), the network maximised its trial-specific reward. The agent was trained via stochastic gradient descent (SGD) with the Adam optimiser and a learning rate of 1e-4. The learning rate was determined by a via hyperparameter optimisation at a previous piloting stage.

2.1.4) Task Design and Procedure. We compared the effects of blocked and interleaved training. In the blocked curriculum, the agent was first trained on 10000 trials from one task, and then on 10000 trials from the second task. In the interleaved group, the agent was trained on 20000 randomly shuffled trials from both tasks. In both cases, we evaluated the performance on 10000 previously unseen test trials per task. The training data was sampled from the set of 50000 trees, keeping the number of trees per combination of leafiness and branchiness consistent across tasks and runs. The test phases occurred after half of the training trials were fed through the network, and at the end of the training phase. During these evaluation phases, we recorded the activations in the convolutional, fully-connected and output layers on each trial.
We counterbalanced task order and reward assignments in the same way as for the behavioural experiments. In total, 40 independent runs with random weight initialisations per combination of boundary (cardinal or diagonal) and curriculum (blocked or interleaved) were collected. In contrast to human learners, deep learning agents require a considerable higher number of training examples, to avoid rote-learning and thus over-fitting to the idiosyncrasies of the training set.

2.1.5) Performance Analysis. To evaluate performance, we binned the training phase into bins of 100 trials, and computed the percentage of correct choices for each bin. This was carried out on independent runs, before we computed the group means for each training curriculum. For the test phase, we calculated the percentage of correct decision across all trials, for both tasks individually and across tasks, before averaging across runs. In both cases, we excluded category boundary trials, where no correct decision was possible.

2.1.6) RSA. We constructed Representational Dissimilarity Matrices (RDMs) from the individual layers, based on their responses to test data stimuli. The procedure was almost identical to our RSA on the behavioural data. However, instead of having just one value (e.g. choice probability) per stimulus, we obtained now response matrices of size 50xn with n being the size of the (flattened) layer activations. The dissimilarity between activity patterns for pairs of stimuli was calculated using the correlation distance measure (1-correlation), yielding one 50x50 RDM per

layer. Once again, we correlated these RDMs with model RDMs (Fig. S7), capturing pixel dissimilarity, factorised task encodings, catastrophic interference (e.g. encoding of task 1 as if it was task 2) or a shared linear boundary. The binary model RDMs were hand-crafted. For the pixel dissimilarity RDM, we computed the pairwise Euclidean distances between vectorised RGB images of gardens with superimposed trees, for all 25x2 combinations of branchiness, leafiness and context.

3) Experiment 4a and 4b

Exp. 4 Summary. Exp. 4 was aimed at replicating the effect of perceptual priors on continual task performance that we had observed in our human participants (Exp2). The experiment consisted of two distinct phases. In Exp4a, we implemented a recently developed variant of a variational autoencoder, known as a β-VAE, to learn disentangled representations of the main features of the tree dataset in the absence of any supervised learning objective (9). We discovered that with the right parametrisation, the β-VAE could learn two disentangled factors that closely resembled variation along the axes of "branchiness" and "leafiness". Importantly, the network was only trained on trees images (without contextual cues) and was therefore not able to separate contexts a priori. Next, in Exp4b, we trained a feed-forward neural network on the trees categorisation task, as we did previously in Exp3. However, to mimic the perceptual priors exhibited by human participants, we replaced the convolutional layers of the network with the trained encoder network of the β-VAE. The network still received images of trees superimposed onto gardens as inputs (and was trained either on a blocked or interleaved curriculum) but would now have access to a lower-dimensional, latent representations that separated trees along the learned dimensions leafiness and branchiness. Even though still present, the network suffered less from catastrophic interference, indicating that the pretraining intervention, e.g. augmenting the network with similar perceptual priors as our human participants had, facilitated task separation under blocked training curricula. Details of these two simulations are presented below.

3.1) Exp 4a: Unsupervised Learning of Disentangled Representations

Aim of this experiment was to simulate the unsupervised learning of the stimulus space and emergence of disentangled visual concepts (of branchinesss and leafiness). We attempted to show that the most efficient compression of the data would correspond to the structure of the trees space.

3.1.2) Stimuli. The training data set consisted of 50000 trees stimuli, with equal numbers of examples per combination of branchiness and leafiness. Another set of 10000 trees stimuli, equally balanced, was generated and served as test set. All tree stimuli were scaled to 96x96x3 pixels. We normalised the data separately for each colour channel by subtracting the mean and dividing by the standard deviation.

3.1.3) Training Procedure. The network architecture is illustrated in Fig. S6b. The network was trained with minibatches of 128 training stimuli and training was stopped after 20 epochs on the entire training data, as we noted that the network would otherwise overfit to the training data. We evaluated the network's performance after each epoch with a full pass of the test data. Furthermore, after each training epoch, a latent space traverse (see below) was performed and the encoder layer outputs for two full 5x5 sets of tree stimuli were stored for later Representational Similarity Analysis. Training was repeated for different values of beta, ranging from 1 to 100. We found a beta of 50 to yield the best disentanglement (see below).

3.1.4) Latent Space Traversal. For the latent space traversal, we fed linearly spaced values for the mean parameters, ranging from -2 to +2 along each of the two latent dimensions into the trained network and generated the tree via a full pass through the decoder. For each of these value pairs, we placed the generated image at the corresponding (x,y)-position in latent space, to obtain

a visualisation of the traverse. In doing so, we could qualitatively assess the degree to which each of the two latent dimensions captures unique variation along the relevant feature dimensions.

3.1.5) Grid Score RSA. Furthermore, we constructed 25x25 RDMs out of the activations recorded from the encoder layer, using the same methodology as previously described. These RDMs could then be correlated with a conceptual model RDM which exhibited a structure one would obtain if the stimuli were represented in a perfectly linear grid, according to their dissimilarity in branchiness and leafiness. Calculating these RDM model correlations (Kendall tau_a) yielded one "grid score" per beta parameter. This allowed us to identify the value for beta which resulted in the most grid-like representations in the last encoder layer and therefore find a close-to-optimal value for this hyperparameter.

3.2) Exp 4b: CNN simulation of trees task *with* priors. The previous experiment identified a parametrisation for the beta-VAE which allowed us to model our participant's "awareness" of the stimulus space structure. That is, the model learned a compressed representation which arranged the stimuli according to their variation in leafiness and branchiness. As Exp 2 revealed that participants with a high gridiness score, e.g. with a strong awareness of the stimulus space, performed better in the blocked curriculum, compared to subjects with a lower "prior", we could now ask if providing a neural network with such representations would boost its continual task performance. We replaced the convolutional layers of the agent with the trained encoder of the beta-VAE from Exp 4a and froze its weights, which means that a gradient update would not alter values of weights in these layers (Fig. S6c). With this approach, we simulated having a fixed feature extractor that disentangles branchiness and leafiness, and task learning (and thus weight adaptation) occurred on this new "intermediate" representation instead of the raw RGB images.

3.2.1) Training, Procedure and Analysis. The training parameters and task design were identical to the ones described for Exp. 3. Once again, training happened online with one trial per gradient update, and composite images consisting of a trial-specific garden and tree were used as inputs to the network. We collected the same number of runs per group and recorded the hidden layer activity on independent test data. Likewise, the same analyses were performed, namely calculation of learning curves, test performance and RSA on the layer-wise activity patterns.

References

1. Carvalho PF, Goldstone RL (2014) Putting category learning in order: Category structure and temporal arrangement affect the benefit of interleaved over blocked study. *Mem Cogn* 42(3):481–495.
2. Hout MC, Goldinger SD, Ferguson RW (2013) The versatility of SpAM: A fast, efficient, spatial method of data collection for multidimensional scaling. *J Exp Psychol Gen* 142(1):256–281.
3. Verheyen S, Voorspoels W, Vanpaemel W, Storms G (2016) Caveats for the spatial arrangement method: Comment on hout, goldinger, and ferguson (2013). *J Exp Psychol Gen* 145(3):376–382.
4. Kriegeskorte N, Mur M (2012) Inverse MDS: Inferring Dissimilarity Structure from Multiple Item Arrangements. *Front Psychol* 3:245.
5. Kriegeskorte N (2008) Representational similarity analysis – connecting the branches of systems neuroscience. *Front Syst Neurosci*. doi:10.3389/neuro.06.004.2008.
6. Kriegeskorte N, Kievit RA (2013) Representational geometry: Integrating cognition, computation, and the brain. *Trends Cogn Sci* 17(8):401–412.
7. Daunizeau J, Adam V, Rigoux L (2014) VBA: A Probabilistic Treatment of Nonlinear Models for Neurobiological and Behavioural Data. *PLoS Comput Biol* 10(1):e1003441.
8. Rigoux L, Stephan KE, Friston KJ, Daunizeau J (2014) Bayesian model selection for group studies - Revisited. *Neuroimage* 84:971–985.
9. Higgins I, et al. (2016) Early Visual Concept Learning with Unsupervised Deep Learning. Available at: http://arxiv.org/abs/1606.05579 [Accessed December 17, 2017].

Supplementary Figures

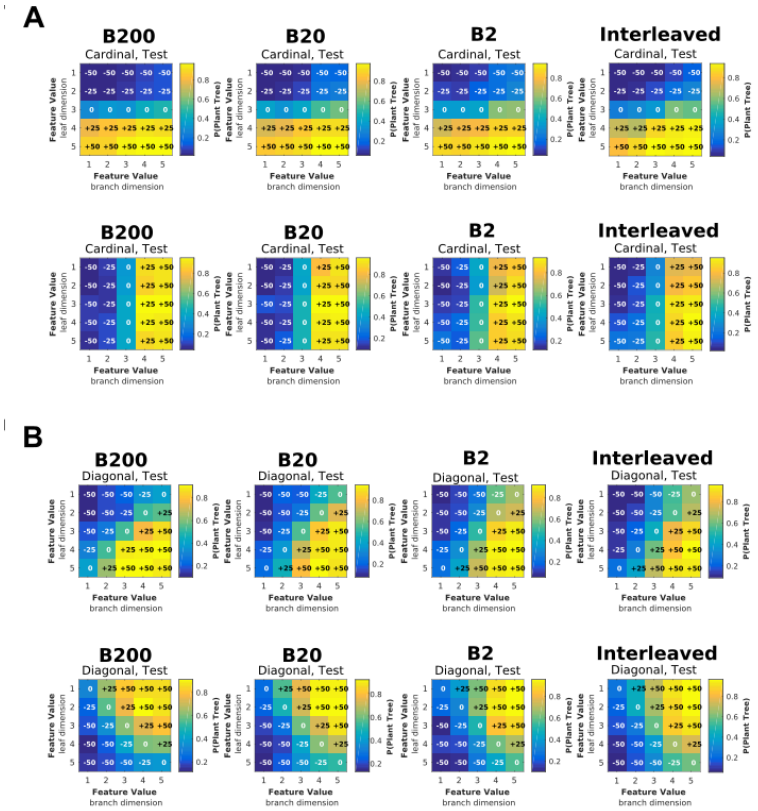

Fig. S1. Exp 1. Group-level choice probability matrices. (A) Exp. 1a (cardinal): Test-phase choice probability matrices for each task and group. One can nicely see a that patterns become more biased towards a combined boundary, the less the two tasks were temporally autocorrelated during training. Matrices were flipped and rotated on single subject level to account for counterbalancing of rewards signs across subjects. (B) Exp. 1b (diagonal): Test-phase choice probability matrices for each task and group. The patters resemble the ones described for Fig S1a. Whilst participants learned the boundaries very well under B200 training, interleaved or B2 training would lead to more mixed responses. In contrast to the cardinal boundary, there seem to be more random errors.

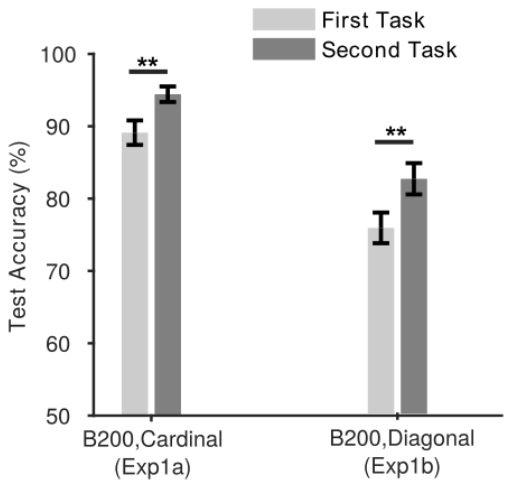

Fig. S2. Exp 1a,b. Test Performance of first vs second task, B200 training. Comparison of performance of first vs second task for cardinal boundaries (Exp 1a, left) and diagonal boundaries (Exp1b, right). In both cases, participants were slightly worse on the first task (Exp1a: First < Second, $Z = 3.09$, $p < 0.01$; Exp1b: First < Second, $Z = 2.65$, $p < 0.01$) indicating at least partial forgetting, which was, however, much less severe than for artificial neural networks. Forgetting was not more severe for Exp1b than for Exp 1a ($Z = 0.5$, $p = 0.62$).

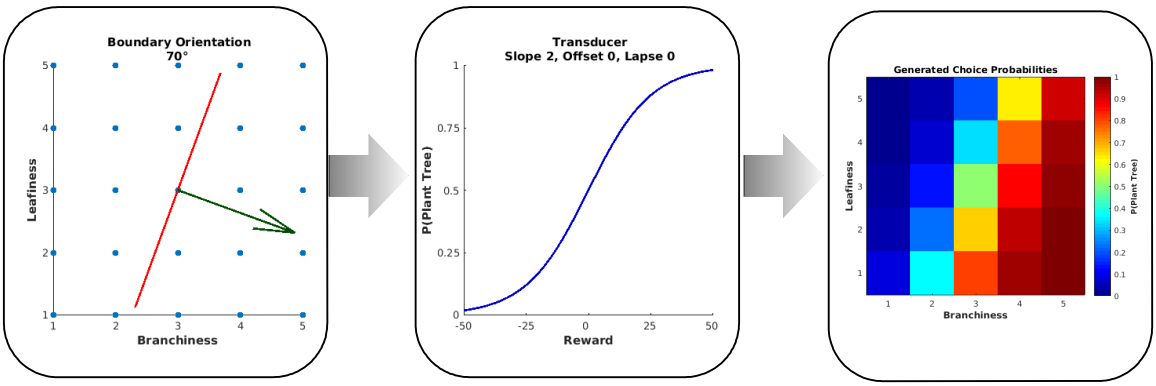

Fig. S3. Exp 1,2. Visualisation of Psychophysical Model. To estimate the factors that contributed to the observed group-level choice probability matrices, we fitted two variants of a psychophysical model to the data. Left: The orientation of the category boundary in trees space was dictated by a free parameter. Each coordinate pair was projected onto the line perpendicular to the bounadry (green arrow) to obtain an estimate of the distance to the category boundary. Middle: These scalar values were passed through a sigmoidal transducer with the parameters slope, offset and lapse (see methods for details) to generate a choice probability as function of the reward that was associated with planting the input tree. Right: The model output was a matrix with choice probabilities (averaged over trials) that was then fit to the single subject choices using a maximum likelihood procedure. We compared two variants with different flexibility: The 2-boundary ("unconstrained") model had eight parameters in total, one boundary and three sigmoid parameters per task, and could therefore estimate task-specific choice patterns. The 1-boundary model was more constrained, as it only had four free parameters – one boundary and one sigmoid with slope, offset and lapse – and allowed us to test the hypothesis that participants collapsed across tasks (i.e. ignored the contextual cue) and estimated a single rule to solve the task.

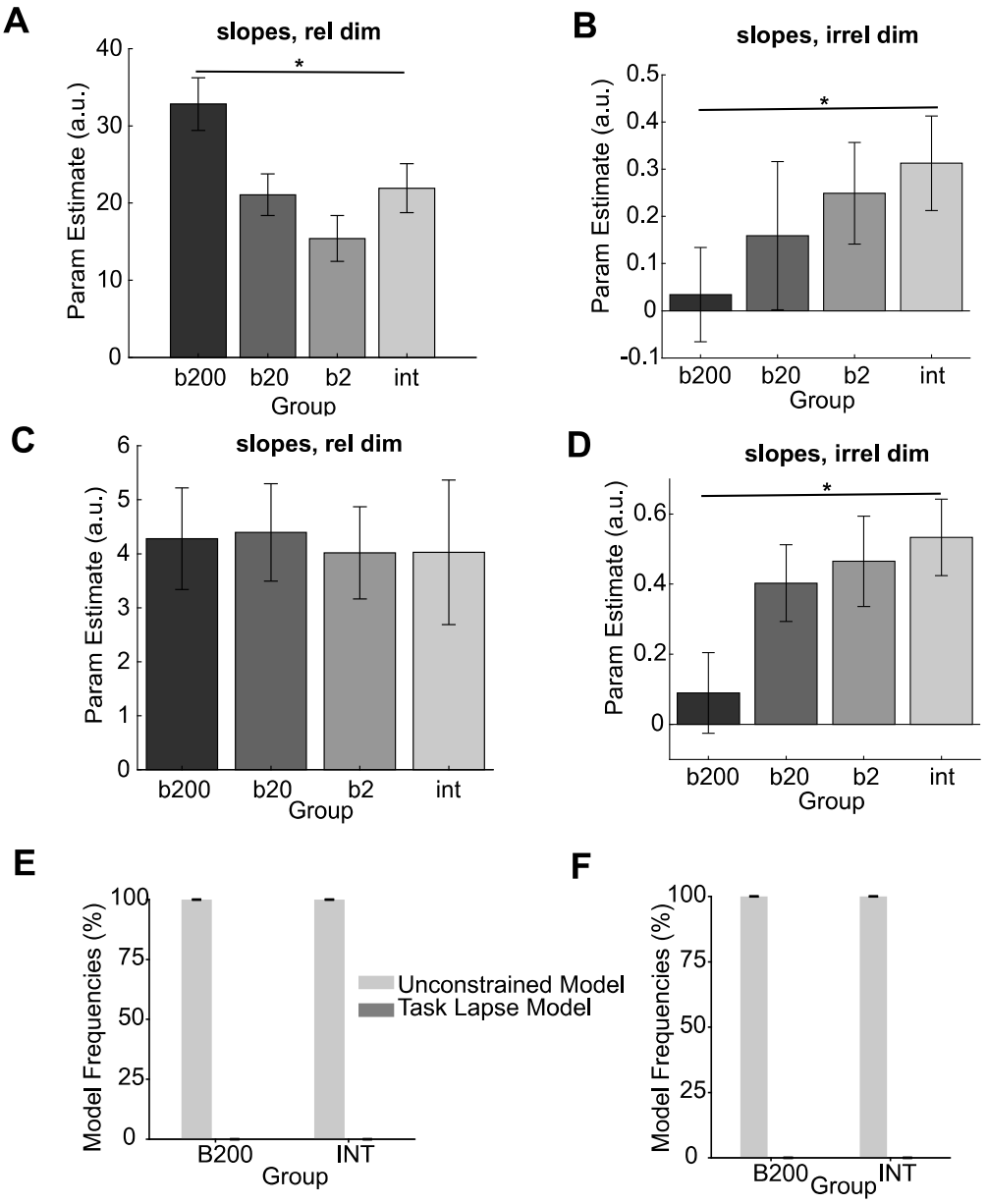

Fig S4. Task Lapse Analysis. (A-B) Exp 1a. Fits of sigmoids (relevant and irrelevant dimension) to synthetic choices sampled from the unconstrained model with best-fitting parameters. The results replicate the findings reported in Fig 2c, indicating that the intrusions along the irrelevant dimension are driven by a wrong estimate of the category boundary. (C-D) Exp 1b. Same analysis as for 1a, now with diagonal boundary. Once again, the stronger intrusion for interleaved compared to B200 suggests a boundary bias as underlying cause (E-F) Random Effects Bayesian model comparison of the unconstrained model with the task lapse model, on cardinal (E) and diagonal (F) boundaries. The Unconstrained model explains the data significantly better than the task lapse model, supporting our hypothesis that the source of error are driven by a bias in the boundary estimate rather than rule confusion (with perfectly learned boundaries).

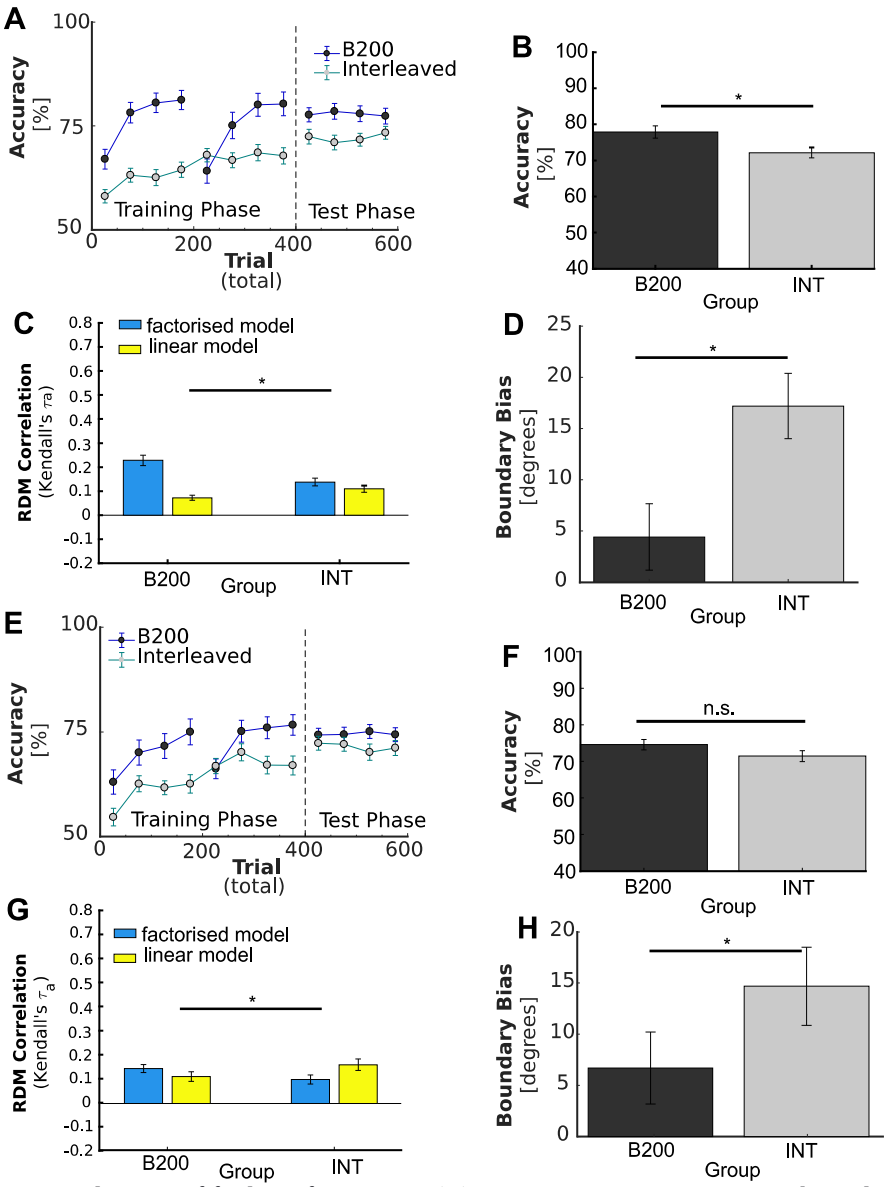

Fig. S5. Exp 2: Replication of findings from Exp1 (A) Exp. 2a, Learning curves. At the end of the training phase, both groups had reached a stable performance plateau. The B200 group learned fast and reached a higher training performance than the Interleaved group. This advantage continued during the test phase. (B) Exp. 2a. Mean test performance for B200 and Interleaved groups. Blocked training led to significantly higher test accuracy than interleaved training. (C) Exp2a. RDM Model correlations. There is stronger evidence for factorised representations after B200 than after interleaved training. Furthermore, interleaved training results in higher correlations with a linear model, than B200 training. (D) Exp. 2a, test phase decision boundary bias, estimated by unconstrained model. Interleaved training resulted in a significantly stronger difference between estimated decision boundary and true category boundary, than B200 training. (E) Exp. 2b, learning curves. While the B200 group learned faster and reached a higher terminal training performance than the interleaved group, the differences in the test phase were rather minute. (F) Exp. 2b, test phase mean accuracy. As for Exp. 1b, there was no significant difference in test performance between B200 and interleaved training on diagonal boundaries. (G) Exp. 2b, RDM model correlations. The factorised model explained the B200 group data – but not the interleaved group data – better than the linear model. (H) Exp 2b, decision boundary bias, unconstrained model. Participants in the interleaved group had a significantly higher bias in their estimated decision boundaries than participants in the B200 group.

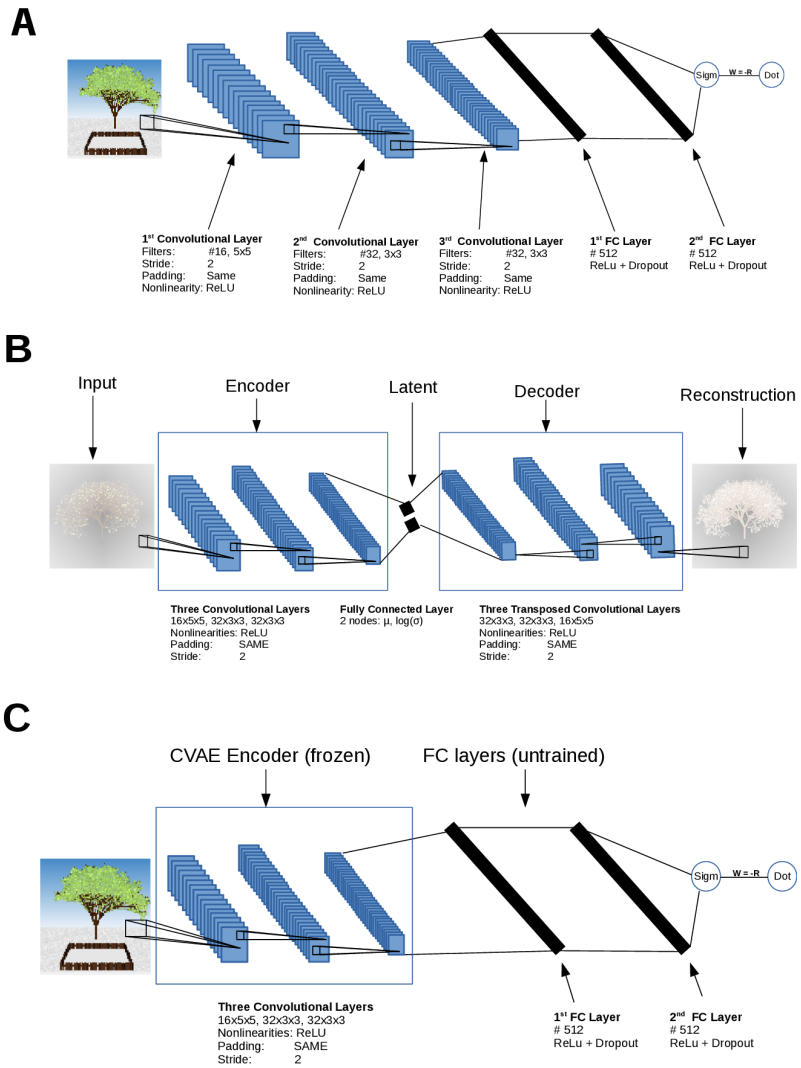

Fig. S6. Neural Network Architectures. (A) Exp 3. Model Architecture. The agent consisted of three convolutional layers (16x5x5,32x3x3,32x3x3, padding=SAME, nonlinearity=ReLU, stride=2) and two fully-connected layers (512x1,512x1, nonlinearity=ReLU, dropout=0.5), followed by a one-dimensional output unit (nonlinearity = Sigmoid). Weights were initialised with He-Initialisation, and biases set to 0.01. (B) Exp 4a. Model Architecture. The encoder consistent of three convolutional layers which were identical to the first three layers of the network depicted in Fig. S4a. We flattened the output of the encoder and fed it into the two-dimensional latent layer, which was separated into a node for the mean paramenter and a node for the logsima parameter of the latent distribution. These parameters were combined with a sample from a zero mean, isotropic variance gaussian distribution in the subsequent reparametrisation layer. The decoder consisted of three layers with transposed convolutions, arranged in reverse order of the encoder layers. All layers in the encoder and the first two decoder layers passed their outputs through ReLU nonlinearities. No nonlinearities were applied to the latent layer and the final output of the decoder was passed through a hyperbolic tangent, as the input image could have values below zero due to normalisation, and the optimisation aim of the VAE was to resemble the input in its output as close as possible. The weights were initialized using He-Initialization and all biases were initially set to 0.001. (C) We selected the encoder of the model with the highest gridiness score in Exp. 4a and used it to replace the convolutional layers of the agent from Exp. 3. Note that architecture is exactly the same, but weights now instead of being randomly initialised encode an efficient representation of the stimulus space before any training on the trees task has taken place. Next, we freezed the weights of the convolutional layers to ensure that the representation remains persistent and serves as feature extractor throughout training.

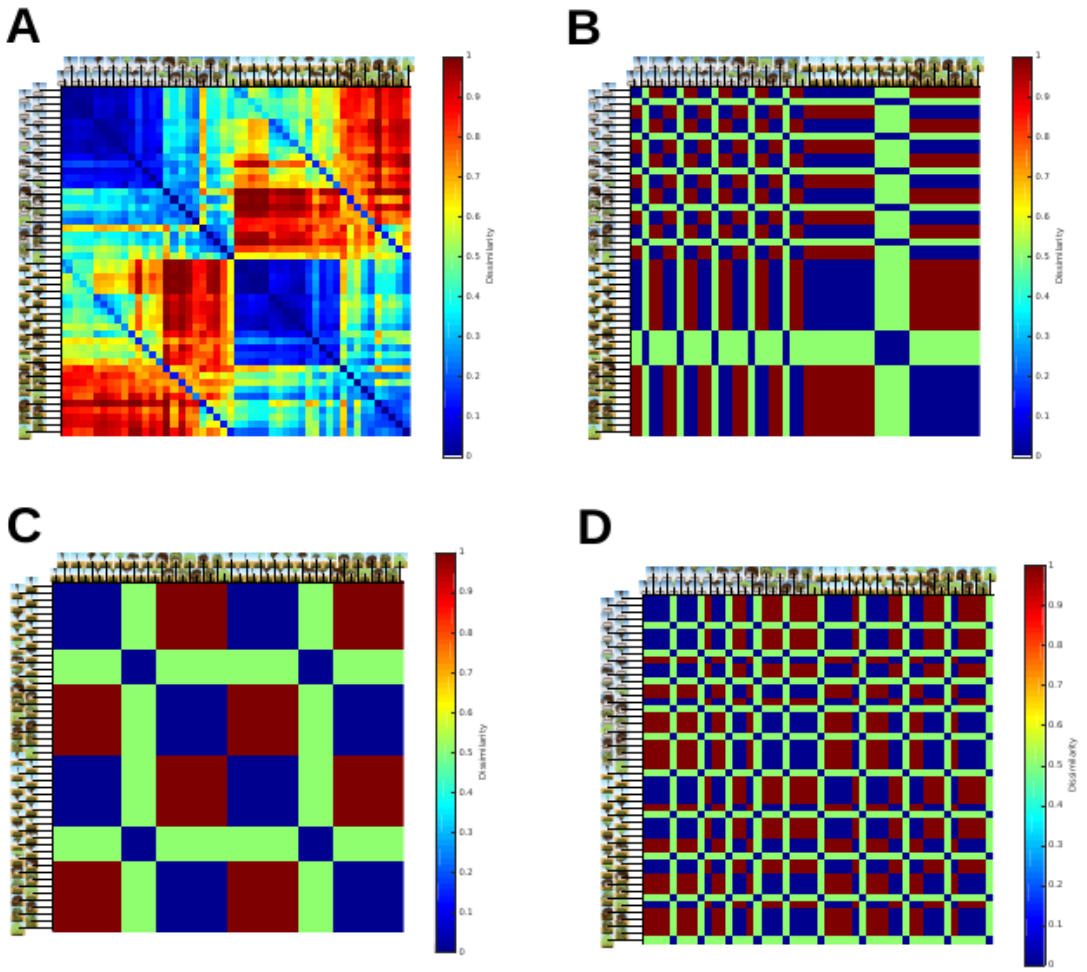

Fig S7. Exp3,4. Example of Model RDMs. (A) Pixel Space dissimilarity model. Here, we computed the dissimilarity between vectorised RGB images for all 50 possible combinations of leafiness,branchiness and task. (B) Factorised Model. Cardinal boundaries. This model RDM assumed that the network had learned perfect binary representations of both categories for each task (compare Fig 2d). (C) Interference Model RDM. Cardinal boundaries. If the network had been trained first on the north and subsequently on the south task, we expected catastrophic interference. This model RDM assumes that the first task is treated as if it was the second task. (D) Linear Boundary Model. Cardinal boundaries. In the behavioural experiments, we tested the hypothesis that participants learn only one single, *linear* boundary for both tasks, which is positioned in trees space such that reward is maximised in both gardens. This RDM illustrates how representational geometries would look like in such a model.

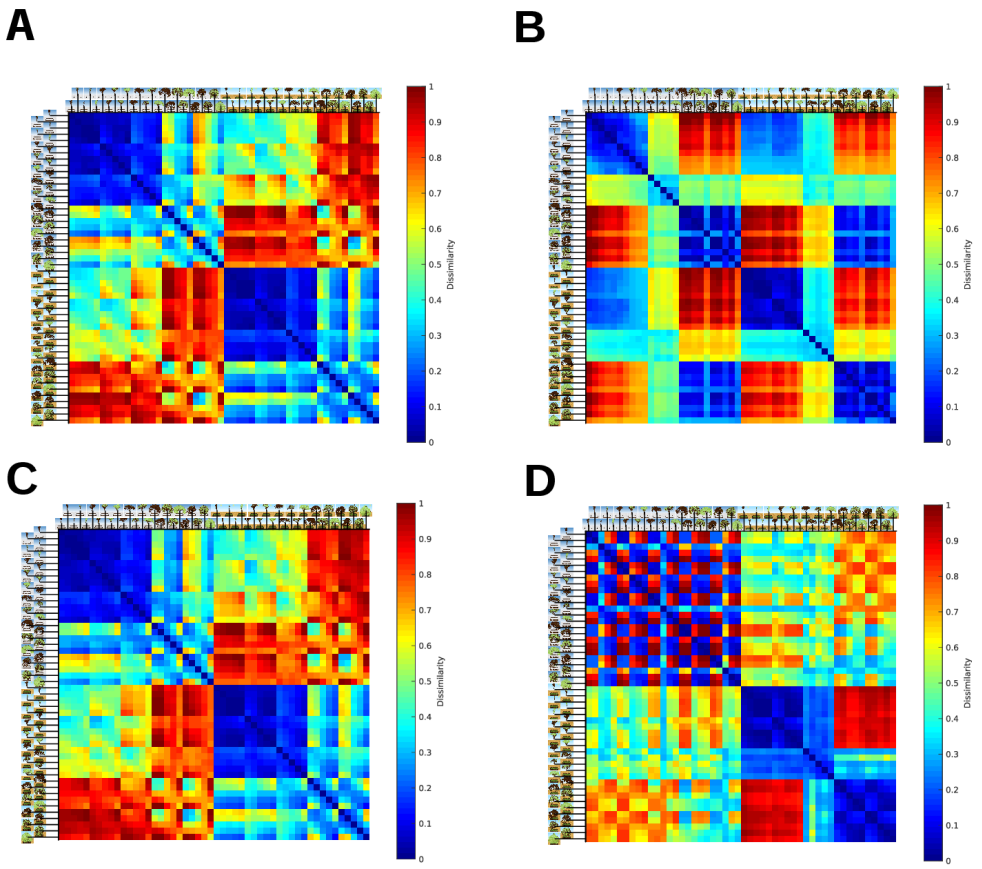

Fig S8. Exp 3a, Example of Layer RDMs. (A) Exp 3a, Blocked training. RDM constructed from the activity patterns in the first convolutional layer during the final test phase. (B) Exp 3a, Blocked training. RDM constructed from the test-phase activity patterns in the FC2 layer. As expected, the network responds to the first task (top left quadrant) as if it was the second task (bottom right quadrant). (C) Exp. 3b, interleaved training. RDM obtained from the first convolutional layer during the last test phase. As under Blocked training, mostly differences in pixel value patterns is encoded. (D) Exp 3b, interleaved training. RDM obtained from the FC2 layer. In contrast to Blocked training (Fig S6b), the second FC layer exhibits clear separation of representational geometries for the north and south task. In other words, interleaved training allowed the network to learn the boundaries for each of the two tasks, which is reflected in FC2-layer responses to branchiness in the south and leafiness in the north task.

Supplementary Tables

Table S1. Numbers of above-chance participants per experiment and group

total(m,f)	B200	B20	B2	Interleaved
Exp 1a	48(29,19)	41(18,23)	40(23,17)	47(30,17)
Exp 1b	42(19,23)	40(20,20)	40(22,18)	44(24,20)
Exp 2a	68(43,25)	-	-	70(51,19)
Exp 2b	55(40,15)	-	-	48(33,15)

Table S2. Mean age of participants per experiment and group

	B200	B20	B2	Interleaved
Exp 1a	34.38	34.22	35.85	35
Exp 1b	34	37.1	34.5	36.82
Exp 2a	31.15	-	-	30.71
Exp 2b	32.89	-	-	29.38

Mean age of participants per group and experiment.

Table S3 Overview of achieved statistical power for all reported post-hoc tests

This table illustrates the achieved power (1-beta) for each post-hoc test reported in the main text . Power was computed using the G*Power software package for Microsoft Windows.

EXPERIMENT	QUANTITY	COMPARISON	1-BETA
EXP 1A	**Test Accuracy**	B200 > Int	0.63
	(All trials)	B200 > B2	0.94
		Int > B2	0.56
	Test Accuracy	B200 > Int	0.53
	(Switch Trials)	B200 > B2	0.94
	Intrusions	B200 < Int	0.77
	(Irrelevant Dimension)		
	Task Factorisation	$B200_{diff} > Int_{diff}$	0.93
	(RSA)	$B200_{diff} > B2_{diff}$	0.94
	Task Factorisation	$B200_{diff} > Int_{diff}$	0.95
	(Model Selection)	$B200_{diff} > B2_{diff}$	0.91
	Angular Bias	B200 < Int	0.64
	(Unconstrained Model)		
	Slope, Relevant Dim.	B200 > Int	0.62
	(Unconstrained Model)	B200 > B2	0.93
EXP 1B	**Intrusions**	B200 < Int	0.89
	(Irrelevant Dim.)	B200 < B2	0.86
	Task Factorisation	$B200_{diff} > Int_{diff}$	0.91
	(RSA)	$B200_{diff} > B2_{diff}$	0.96
	Task Factorisation	$B200_{diff} > Int_{diff}$	0.88
	(Model Selection)	$B200_{diff} > B2_{diff}$	0.79
	Angular Bias	B200 < Int	0.82
	(Unconstrained Model)	B200 < B2	0.73
	Random Lapses	B200 > Int	0.22
EXP 2A	**Task Factorisation**	B200 > Int	0.91
	(RSA)		
	Task Factorisation	B200 > Int	0.98
	(Model Selection)		
	Angular Bias	B200 < Int	0.74
	(Unconstrained Model)		
	Grid Prior	$B200_{high} > B200_{low}$	0.87

	(Accuracy)	$Int_{high} > Int_{low}$	0.33
	Grid Prior	$B200_{high} > B200_{low}$	0.94
	(Task Factorisation)	$Int_{high} > Int_{low}$	0.40
EXP 2B	**Task Factorisation** (RSA)	B200 > Int	0.36
	Task Factorisation (Model Selection)	B200 > Int	0.54
	Angular Bias (Unconstrained Model)	B200 < Int	0.37
	Grid Prior (Accuracy)	$Pooled_{high} > Pooled_{low}$	0.48
	Grid Prior (Task Factorisation)	$Pooled_{high} > Pooled_{low}$	0.64
EXP 4B	**Retained Accuracy** (First Task, Cardinal)	VanillaCNN < PriorCNN	0.99
	Retained Accuracy (First Task, Diagonal)	VanillaCNN < PriorCNN	0.99
	Task Factorisation (RSA, Cardinal)	$VanillaCNN_{fc1} < PriorCNN_{fc1}$	0.99
		$VanillaCNN_{fc2} < PriorCNN_{fc2}$	0.99
		$VanillaCNN_{out} < PriorCNN_{out}$	0.99
	Task Factorisation (RSA, Diagonal)	$VanillaCNN_{fc1} < PriorCNN_{fc1}$	0.28
		$VanillaCNN_{fc2} < PriorCNN_{fc2}$	0.27
		$VanillaCNN_{out} < PriorCNN_{out}$	0.29
	Task Interference (RSA, Cardinal)	$VanillaCNN_{fc1} > PriorCNN_{fc1}$	0.99
		$VanillaCNN_{fc2} > PriorCNN_{fc2}$	0.99
		$VanillaCNN_{out} > PriorCNN_{out}$	0.99
	Task Interference (RSA, Diagonal)	$VanillaCNN_{fc1} > PriorCNN_{fc1}$	0.74
		$VanillaCNN_{fc2} > PriorCNN_{fc2}$	0.86
		$VanillaCNN_{out} > PriorCNN_{out}$	0.92