Generalization guides human exploration in vast decision-spaces

Charley M. Wu, Eric Schulz, Maarten Speekenbrink, Jonathan D. Nelson, & Björn Meder

Supplementary Methods

Full Model Comparison

We report the full model comparison of 27 models, of which 12 (i.e., four learning models and three sampling strategies) are included in the main text. We use different *Models of Learning* (i.e., Function Learning and Option Learning), which combined with a *Sampling Strategy* can make predictions about where a participant will search, given the history of previous observations. We also include comparisons to *Simple Heuristic Strategies*[1], which make predictions about search decisions without maintaining a representation of the world (i.e., without a learning model). Table S3 shows the predictive accuracy, the number of participants best described, the protected probability of exceedance and the median parameter estimates of each model. Figure S1 shows a more detailed assessment of predictive accuracy and model performance, with participants separated by payoff condition and environment type.

Models of Learning

Function Learning. The Function Learning Model adaptively learns an underlying function mapping spatial locations onto rewards. We use Gaussian Process (GP) regression as a Bayesian method of function learning[2]. A GP is defined as a collection of points, any subset of which is multivariate Gaussian. Let $f : \mathcal{X} \rightarrow \mathbb{R}^n$ denote a function over input space $\mathcal{X}$ that maps to real-valued scalar outputs. This function can be modelled as a random draw from a GP:

$$f \sim \mathcal{GP}(m,k), \tag{1}$$

where m is a mean function specifying the expected output of the function given input $\mathbf{x}$, and k is a kernel (or covariance) function specifying the covariance between outputs.

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{2}$$

$$k(\mathbf{x},\mathbf{x}') = \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))\right] \tag{3}$$

Here, we fix the prior mean to the median value of payoffs, $m(\mathbf{x}) = 50$ and use the kernel function to encode an inductive bias about the expected spatial correlations between rewards (see Radial Basis Function kernel). Conditional on observed data $\mathcal{D}_t = \{\mathbf{x}_j, y_j\}_{j=1}^t$, where $y_j \sim \mathcal{N}(f(\mathbf{x}_j), \sigma^2)$ is drawn from the underlying function with added noise $\sigma^2 = 1$, we can calculate the posterior predictive distribution for a new input $\mathbf{x}_*$ as a Gaussian with mean $m_t(\mathbf{x}_*)$ and variance $v_t(\mathbf{x}_*)$ given by:

$$\mathbb{E}[f(\mathbf{x}_*)|\mathcal{D}_t] = m_t(\mathbf{x}_*) = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}_t \tag{4}$$

$$\mathbb{V}[f(\mathbf{x}_*)|\mathcal{D}_t] = v_t(\mathbf{x}_*) = k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_\star^\top (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_*, \tag{5}$$

where $\mathbf{y} = [y_1, \ldots, y_t]^\top$, $\mathbf{K}$ is the $t \times t$ covariance matrix evaluated at each pair of observed inputs, and $\mathbf{k}_* = [k(\mathbf{x}_1, \mathbf{x}_*), \ldots, k(\mathbf{x}_t, \mathbf{x}_*)]$ is the covariance between each observed input and the new input $\mathbf{x}_*$.

We use the Radial Basis Function (RBF) kernel as a component of the GP function learning algorithm, which specifies the correlation between inputs.

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{\lambda}\right) \tag{6}$$

This kernel defines a universal function learning engine based on the principles of Bayesian regression and can model any stationary function. Note, sometimes the RBF kernel is specified as $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2l^2}\right)$ whereas we use $\lambda = 2l^2$ as a more psychologically interpretable formulation. Intuitively, the RBF kernel models the correlation between points as an exponentially decreasing function of their distance. Here, λ modifies the rate of correlation decay, with larger λ-values corresponding to slower decays, stronger spatial correlations, and smoother functions. As $\lambda \to +\infty$, the RBF kernel assumes functions approaching linearity, whereas as $\lambda \to 0$, there ceases to be any spatial correlation, with the implication that learning happens independently for each input without generalization (similar to traditional models of associative learning). We treat λ as a free parameter, and use cross-validated estimates to make inferences about the extent to which participants generalize.

Option Learning. The Option Learning Model uses a Bayesian Mean Tracker, which is a type of associative learning model that assumes the average reward associated with each option is constant over time (i.e., no temporal dynamics, as opposed to the assumptions of a Kalman filter or Temporal Difference Learning)[3], as is the case in our experimental search tasks. In contrast to the Function Learning model, the Option Learning model learns the rewards of each option separately, by computing an independent posterior distribution for the mean μ_j for each option j. We implement a version that assumes rewards are normally distributed (as in the GP Function Learning Model), with a known variance but unknown mean, where the prior distribution of the mean is again a normal distribution. This implies that the posterior distribution for each mean is also a normal distribution:

$$p(\mu_{j,t}|\mathcal{D}_{t-1}) = \mathcal{N}(m_{j,t}, v_{j,t}) \tag{7}$$

For a given option j, the posterior mean $m_{j,t}$ and variance $v_{j,t}$ are only updated when it has been selected at trial t:

$$m_{j,t} = m_{j,t-1} + \delta_{j,t} G_{j,t} \left[y_t - m_{j,t-1}\right] \tag{8}$$

$$v_{j,t} = \left[1 - \delta_{j,t} G_{j,t}\right] v_{j,t-1} \tag{9}$$

where $\delta_{j,t} = 1$ if option j was chosen on trial t, and 0 otherwise. Additionally, y_t is the observed reward at trial t, and $G_{j,t}$ is defined as:

$$G_{j,t} = \frac{v_{j,t-1}}{v_{j,t-1} + \theta_\varepsilon^2} \tag{10}$$

where θ_ε^2 is the error variance, which is estimated as a free parameter. Intuitively, the estimated mean of the chosen option $m_{j,t}$ is updated based on the difference between the observed value y_t and the

prior expected mean $m_{j,t-1}$, multiplied by $G_{j,t}$. At the same time, the estimated variance $v_{j,t}$ is reduced by a factor of $1-G_{j,t}$, which is in the range $[0,1]$. The error variance (θ_ε^2) can be interpreted as an inverse sensitivity, where smaller values result in more substantial updates to the mean $m_{j,t}$, and larger reductions of uncertainty $v_{j,t}$. We set the prior mean to the median value of payoffs $m_{j,0}=50$ and the prior variance $v_{j,0}=500$.

Sampling Strategies

Given the normally distributed posteriors of the expected rewards, which have mean $m_t(\mathbf{x})$ and the estimated uncertainty (estimated here as a standard deviation) $s_t(\mathbf{x})=\sqrt{v_t(\mathbf{x})}$, for each search option $\mathbf{x}$ (for the Option Learning model, we let $m_t(\mathbf{x})=m_{j,t}$ and $v_t(\mathbf{x})=v_{j,t}$, where j is the index of the option characterized by $\mathbf{x}$), we assess different sampling strategies that (with a softmax choice rule) make probabilistic predictions about where participants search next at time $t+1$.

Upper Confidence Bound Sampling. Given the posterior predictive mean $m_t(\mathbf{x})$ and the estimated uncertainty $s_t(\mathbf{x})$, we calculate the upper confidence bound (UCB) using a simple weighted sum

$$\mathrm{UCB}(\mathbf{x})=m_t(\mathbf{x})+\beta s_t(\mathbf{x}), \tag{11}$$

where the exploration factor β determines how much reduction of uncertainty is valued (relative to exploiting known high-value options) and is estimated as a free parameter.

Pure Exploitation and Pure Exploration. Upper Confidence Bound sampling can be decomposed into a Pure Exploitation component, which only samples options with high expected rewards, and a Pure Exploration component, which only samples options with high uncertainty.

$$\mathrm{PureExploit}(\mathbf{x})=m_t(\mathbf{x}) \tag{12}$$

$$\mathrm{PureExplore}(\mathbf{x})=s_t(\mathbf{x}) \tag{13}$$

Expected Improvement. At any point in time t, the best observed outcome can be described as $\mathbf{x}^+=\arg\max_{\mathbf{x}_i\in\mathbf{x}_{1:t}} m_t(\mathbf{x}_i)$. Expected Improvement (EXI) evaluates each option by *how much* (in the expectation) it promises to be better than the best observed outcome $\mathbf{x}^+$:

$$\mathrm{EXI}(\mathbf{x})=\begin{cases}\Phi(Z)(m_t(\mathbf{x})-m_t(\mathbf{x}^+))+s_t(\mathbf{x})\phi(Z), & \text{if } s_t(\mathbf{x})>0\\ 0, & \text{if } s_t(\mathbf{x})=0\end{cases} \tag{14}$$

where $\Phi(\cdot)$ is the normal CDF, $\phi(\cdot)$ is the normal PDF, and $Z=(m_t(\mathbf{x})-m_t(\mathbf{x}^+))/s_t(\mathbf{x})$.

Probability of Improvement. The Probability of Improvement (POI) strategy evaluates an option based on *how likely* it will be better than the best outcome ($\mathbf{x}^+$) observed so far:

$$\begin{aligned}\mathrm{POI}(\mathbf{x}) &= P\left(f(\mathbf{x})\geq f(\mathbf{x}^+)\right)\\ &= \Phi\left(\frac{m_t(\mathbf{x})-m_t(\mathbf{x}^+)}{s_t(\mathbf{x})}\right)\end{aligned} \tag{15}$$

Probability of Maximum Utility. The Probability of Maximum Utility (PMU) samples each option according to the probability that it results in the highest reward of all options in a particular context[3]. It is a form of probability matching and can be implemented by sampling from each option's predictive distributions, and then choosing each option proportional to the number of times it has the highest sampled payoff.

$$\text{PMU}(\mathbf{x}) = P\left(f(\mathbf{x}_j) > f(\mathbf{x}_{i\neq j})\right) \tag{16}$$

We implement this sampling strategy by Monte Carlo sampling from the posterior predictive distribution of a learning model for each option, and evaluating how often a given option turns out to be the maximum over 1,000 generated samples.

Simple Heuristic Strategies

We also compare various simple heuristic strategies that make predictions about search behaviour without learning about the distribution of rewards.

Win-Stay Lose-Sample. We consider a form of a win-stay lose-sample (WSLS) heuristic[4], where a *win* is defined as finding a payoff with a higher or equal value than the previously best observed outcome. When the decision-maker "wins", we assume that any tile with a Manhattan distance ≤ 1 is chosen (i.e., a repeat or any of the four cardinal neighbours) with equal probability. *Losing* is defined as the failure to improve, and results in sampling any unrevealed tile with equal probability.

Local Search. Local search predicts that search decisions have a tendency to stay local to the previous choice. We use inverse Manhattan distance (IMD) to quantify locality:

$$\text{IMD}(\mathbf{x}, \mathbf{x}') = \frac{1}{\sum_{i=1}^{n} |x_i - x_i'|} \tag{17}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are vectors in $\mathbb{R}^n$. For the special case where $\mathbf{x} = \mathbf{x}'$, we set $\text{IMD}(\mathbf{x}, \mathbf{x}') = 1$.

Localization of Models

With the exception of the *Local Search* model, all other models include a localized variant, which introduced a locality bias by weighting the predicted value of each option $q(\mathbf{x})$ by the inverse Manhattan distance (IMD) to the previously revealed tile. This is equivalent to a multiplicative combination with the Local Search model, similar to a "stickiness parameter"[5,6], although we implement it here without the introduction of any additional free parameters. Localized models are indicated with an asterisk (e.g., Function Learning*).

Model Comparison

We use maximum likelihood estimation (MLE) for parameter estimation, and cross-validation to measure out-of-sample predictive accuracy as well as the probability of exceedance to estimate a model's posterior probability to be the underlying predictive model of our task, given the pool of all models in our comparison. A softmax choice rule transforms each model's valuations into a probability distribution over options:

$$p(\mathbf{x}) = \frac{\exp(q(\mathbf{x})/\tau)}{\sum_{j=1}^{N} \exp(q(\mathbf{x}_j)/\tau)}, \tag{18}$$

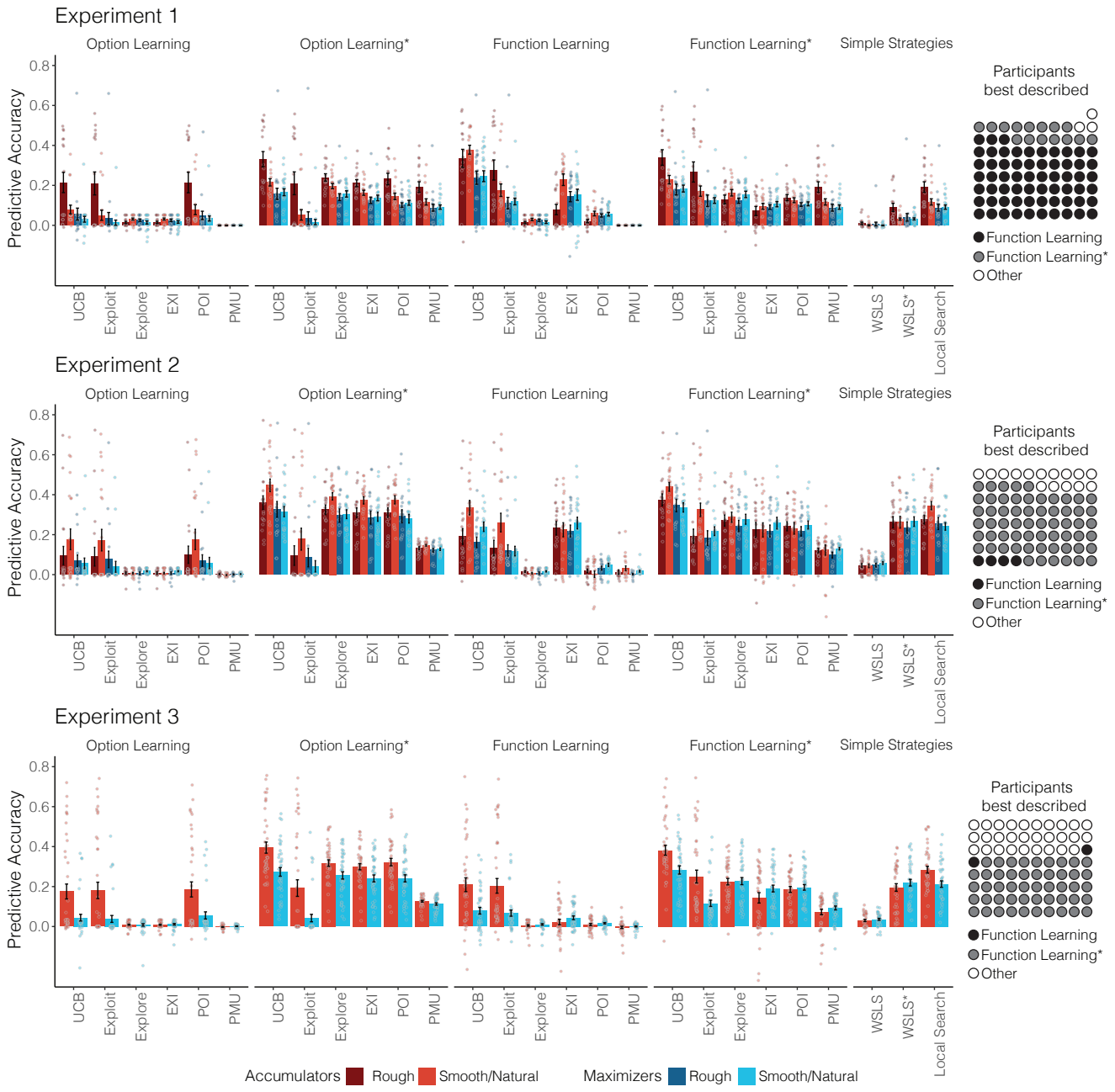

Figure S1. Full model comparison of all 27 models. The learning model is indicated above (or lack of in the case of simple heuristic strategies), and sampling strategy are along the x-axis. Bars indicate predictive accuracy (group mean) along with standard error, and are separated by payoff condition (colour) and environment type (darkness), with individual participants overlaid as dots. Icon arrays (right) show the number participants best described (out of the full 27 models) and are aggregated over payoff conditions, environment types, and sampling strategy. Table S3 provides more detail about the number of participants best described by each model as well as the protected probability of exceedance.

where $q(\mathbf{x})$ is the predicted value of each option $\mathbf{x}$ for a given model (e.g., $q(\mathbf{x}) = \text{UCB}(\mathbf{x})$ for the UCB model), and τ is the temperature parameter. Lower values of τ indicate more concentrated probability distributions, corresponding to more precise predictions. All models include τ as a free parameter. Additionally, Function Learning models estimate λ (length-scale), Option Learning models estimate θ_ε^2 (error variance), and Upper Confidence Bound sampling models estimate β (exploration bonus).

Cross Validation. We fit all models—per participant—using cross-validated MLE, with either a Differential Evolution algorithm[7] or a grid search if the model contained only a single parameter. Parameter estimates are constrained to positive values in the range $[\exp(-5), \exp(5)]$. Cross-validation is performed by first separating participant data according to horizon length, which alternated between rounds (within subjects). For each participant, half of the rounds corresponded to a short horizon and the other half corresponded to a long horizon. Within all rounds of each horizon length, we use leave-one-out cross-validation to iteratively form a training set by leaving out a single round, computing a MLE on the training set, and then generating out-of-sample predictions on the remaining round. This is repeated for all combinations of training set and test set, and for both short and long horizon sets. The cross-validation procedure yielded one set of parameter estimates per round, per participant, and out-of-sample predictions for 120 choices in Experiment 1 and 240 choices in Experiments 2 and 3 (per participant).

Predictive Accuracy. Prediction error (computed as log loss) is summed up over all rounds, and is reported as *predictive accuracy*, using a pseudo-R^2 measure that compares the total log loss prediction error for each model to that of a random model:

$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}, \tag{19}$$

where $\log \mathcal{L}(\mathcal{M}_{\text{rand}})$ is the log loss of a random model (i.e., picking options with equal probability) and $\log \mathcal{L}(\mathcal{M}_k)$ is the log loss of model k's out-of-sample prediction error. Intuitively, $R^2 = 0$ corresponds to prediction accuracy equivalent to chance, while $R^2 = 1$ corresponds to theoretical perfect prediction accuracy, since $\log \mathcal{L}(\mathcal{M}_k) / \log \mathcal{L}(\mathcal{M}_{\text{rand}}) \to 0$ when $\log \mathcal{L}(\mathcal{M}_k) \ll \log \mathcal{L}(\mathcal{M}_{\text{rand}})$. R^2 can also be below zero when the model predictions are worse than random chance.

Simulated learning curves

We use participants' cross-validated parameter estimates to specify a given model and then simulate performance. At each trial, model predictions correspond to a probabilistic distribution over options, which was then sampled and used to generate the observation for the next trial. In order to correspond with the manipulations of horizon length, payoff condition, and environment type, each simulation was performed at the participant level, producing data resembling a virtual participant for each replication. Iterating over each round, we selected the same environment as seen by the participant and then simulated data using the cross-validated parameters that were estimated using that round as the left-out round. Thus, just as model comparison was performed out-of-sample, the generated data was also out-of-sample, based on parameters that were estimated on a different set of rounds than the one being simulated. We performed 100 replications for each participant in each experiment, which were then aggregated to produce the learning curves in Figure 3b.

Model Recovery

We present model recovery results that assess whether or not our predictive model comparison procedure allows us to correctly identify the true underlying model. To assess this, we generated data based on each individual participant's parameter estimates (see above). We generated data using the Option Learning and the Function Learning Model for Experiment 1 and the Option Learning* Model and the Function Learning* Model for Experiments 2 and 3. In all cases, we used the UCB sampling strategy in conjunction with the specified learning model. We then utilized the same cross-validation method as before in order to determine if we could successfully identify which model generated the underlying data. Figure S2 shows the cross-validated predictive performance (half boxplot with each data point representing a single simulated participant) for the simulated data, along with the number of simulated participants best described (inset icon array).

Experiment 1

In the simulation for Experiment 1, our predictive model comparison procedure shows that the Option Learning Model is a better predictor for data generated from the same underlying model, whereas the Function Learning model is only marginally better at predicting data generated from the same underlying model. This suggests that our main model comparison results are robust to Type I errors, and provides evidence that the better predictive accuracy of the Function Learning model for participant data is unlikely due to overfitting.

When the Function Learning Model has generated the underlying data, the same Function Learning Model achieves a predictive accuracy of $R^2 = .4$ and describes 41 out of 81 simulated participants best, whereas the Option Learning model achieves a predictive accuracy of $R^2 = .39$ and describes 40 participants best. Furthermore, the protected probability of exceedance for the Function Learning Model is $\text{pxp} = 0.51$. This makes our finding of the Function Learning Model as the best predictive model even stronger as, technically, the Option Learning Model could mimic parts of the Function Learning behaviour.

When the Option Learning Model generates data using participant parameter estimates, the same Option Learning Model achieves an average predictive accuracy of $R^2 = .1$ and describes 71 out of 81 simulated participants best. On the same generated data, the Function Learning Model achieves an average predictive accuracy of $R^2 = .08$ and only describes 10 out of 81 simulated participants best. The protected probability of exceedance for the Option Learning Model is $\text{pxp} = 0.99$. If the counterfactual had occurred, namely that if data generated by the Option Learning Model had been best predicted by the Function Learning Model, we would need to be sceptical about our modelling results on the basis that the wrong model could describe data better than the true generating model. However, here we see that the Function Learning Model does not make better predictions than the true model for data generated by the Option Learning Model.

Experiment 2

In the simulations for Experiment 2, we used the localized version of each type of learning model for both generation and recovery, since in both cases, localization improved model accuracy in predicting the human participants (Table S3). Here, we find very clear recoverability in all cases, with the recovering model best predicting the vast majority of simulated participants when it is also the generating model (Fig. S2).

When the Function Learning* Model generates the underlying data, the same Function Learning* Model achieves a predictive accuracy of $R^2 = .34$ and describes 77 out of 80 simulated participants best, whereas the Option Learning* Model describes only 3 out of 80 simulated participants best, with

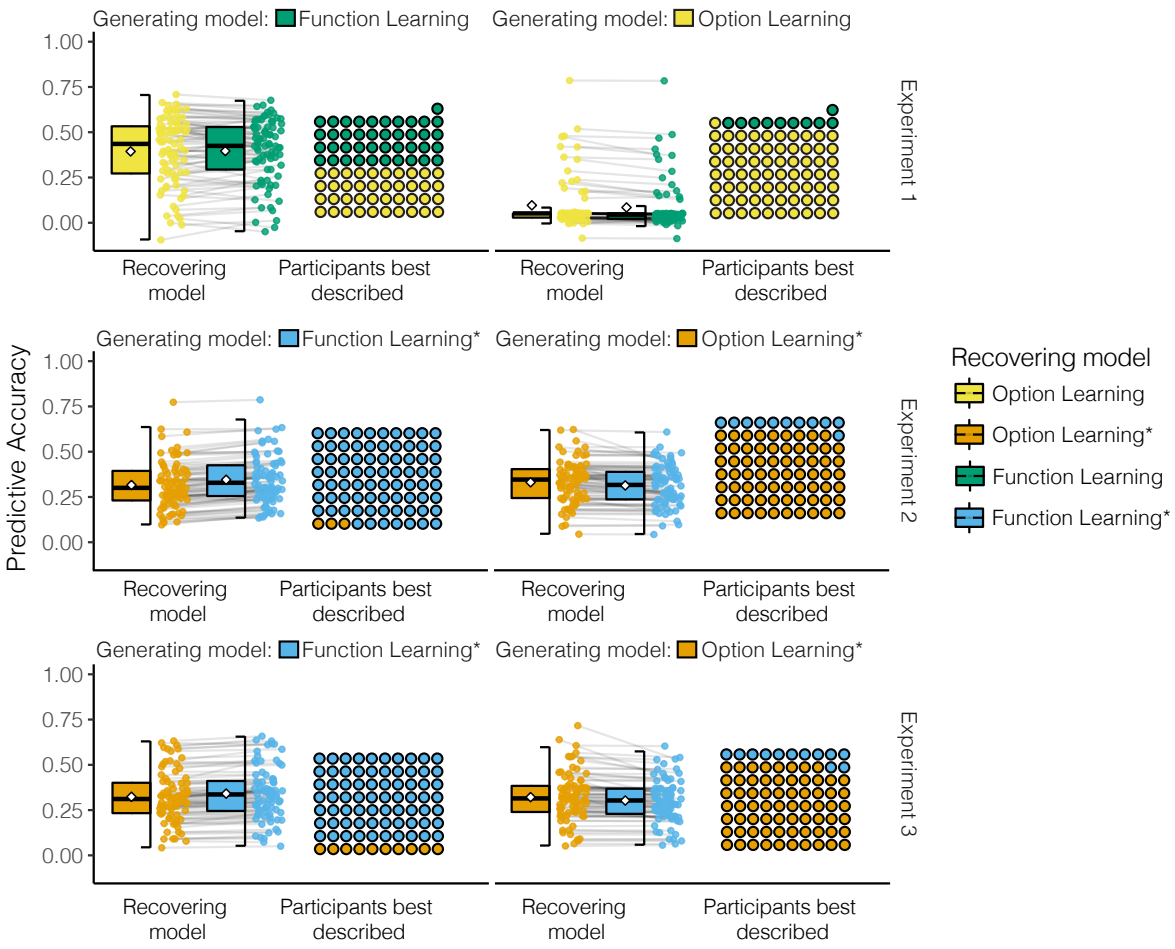

Figure S2. Model recovery results. Data was generated by the specified generating model (left and right columns) using individual participant parameter estimates. The recovery process used the same cross-validation method used in the model comparison. We report the predictive accuracy of each candidate recovery model (colours). Boxplots show the median (line), mean (diamond), interquartile range (box), and 1.5x IQR (whiskers). Each individual (simulated) participant is represented as a dot, with lines connecting each simulated participant. Icon arrays show the number of simulated participants best described. For both generating and recovery models, we used UCB sampling. Table S3 reports the median values of the cross-validated parameter estimates used to specify each generating model.

a average predictive accuracy of $R^2 = .32$. The protected probability of exceedance for the Function Learning* model is pxp $= 1$.

When the Option Learning* Model generates the data, the same Option Learning* Model achieves a predictive accuracy of $R^2 = .33$ and predicts 69 out of 80 simulated participants best, whereas the Function Learning* Model predicts only 11 simulated participants best, with an average predictive accuracy of $R^2 = .31$. The protected probability of exceedance for the Option Learning* model is pxp $= 1$. Again, we find evidence that the models are indeed discriminable, and that the Function Learning* Model does not overfit data generated by the wrong model.

Experiment 3

We again find in all cases the best recovery model is the same as the generating model. When the Function Learning* Model generates data, the matched recovery with the same Function Learning* Model best predicts 70 out of 80 participants, with an average predictive accuracy of $R^2 = .34$. The Option Learning* Model best predicts the remaining 10 participants, with an average predictive accuracy of $R^2 = .32$. The protected probability of exceedance for the Function Learning* model is pxp $= 1$.

When the Option Learning* Model generates the data, the same Option Learning* Model best predicts 68 out of 80 participants with an average predictive accuracy of $R^2 = .32$, whereas the Function Learning* Model only best predicts 12 out of 80 participants with an average predictive accuracy of $R^2 = .3$. The protected probability of exceedance for the Option Learning* model is pxp $= 1$.

In all simulations, the model that generates the underlying data is also the best performing model, as assessed by predictive accuracy, the number of simulated participants predicted best, and the protected probability of exceedance. Thus, we can confidently say that our cross-validation procedure distinguishes between these model classes. Moreover, in the cases where the Function Learning or Function Learning* Model generated the underlying data, the predictive accuracy of the same model is not perfect (i.e., $R^2 = 1$), but rather close to the predictive accuracies we found for participant data (Table S3).

High temperature recovery

We also assessed how much each model's recovery can be affected by the underlying randomness of the softmax choice function. For every recovery simulation, we selected the 10 simulations with the highest underlying softmax temperature parameter τ (ranges: $\tau^{10}_{Exp1} = [0.09, 0.42]$, $\tau^{10}_{Exp2} = [0.11, 0.25]$, $\tau^{10}_{Exp3} = [0.21, 9.7]$) and again calculated the probability of exceedance for the true underlying model. The results of this analysis led to a probability of exceedance for the Function Learning Model in Experiment 1 of pxp $= .81$, for the Function Learning* Model in Experiment 2 of pxp $= 0.99$, for the Function Learning* Model in Experiment 3 of pxp $= 0.93$, for the Option Learning Model in Experiment 1 of pxp $= 0.97$, for the Option Learning* Model in Experiment 2 of pxp $= 0.99$, and for the Option Learning Model in Experiment 3 of pxp $= 0.98$. Thus, the models seem to be well-recoverable even in scenarios with high levels of random noise in the generated responses.

Parameter Recovery

Another important question is whether or not the reported parameter estimates of the two Function Learning models are reliable and robust. We address this question by assessing the recoverability of the three parameters of the Function Learning model, the length-scale λ, the exploration factor β, and the temperature parameter τ of the softmax choice rule. We use the results from the model recovery simulation described above, and correlate the empirically estimated parameters used to generate data (i.e., the estimates based on participants' data), with the parameter estimates of the recovering model (i.e., the MLE from the cross-validation procedure on the simulated data). We assess whether the recovered parameter estimates are similar to the parameters that were used to generated the underlying data. We present parameter recovery results for the Function Learning Model for Experiment 1 and the Function Learning* Model for Experiments 2 and 3, in all cases using the UCB sampling strategy. We report the results in Figure S3, with the generating parameter estimate on the x-axis and the recovered parameter estimate on the y-axis. We report rank-correlation using Kendall's tau (r_τ), which should not be confused with the temperature parameter τ of the softmax function. Additionally, we calculate the Bayes Factor (BF_τ) to quantify the evidence for the presence of a positive correlation using non-informative, shifted, and scaled beta-priors as recommended by[8].

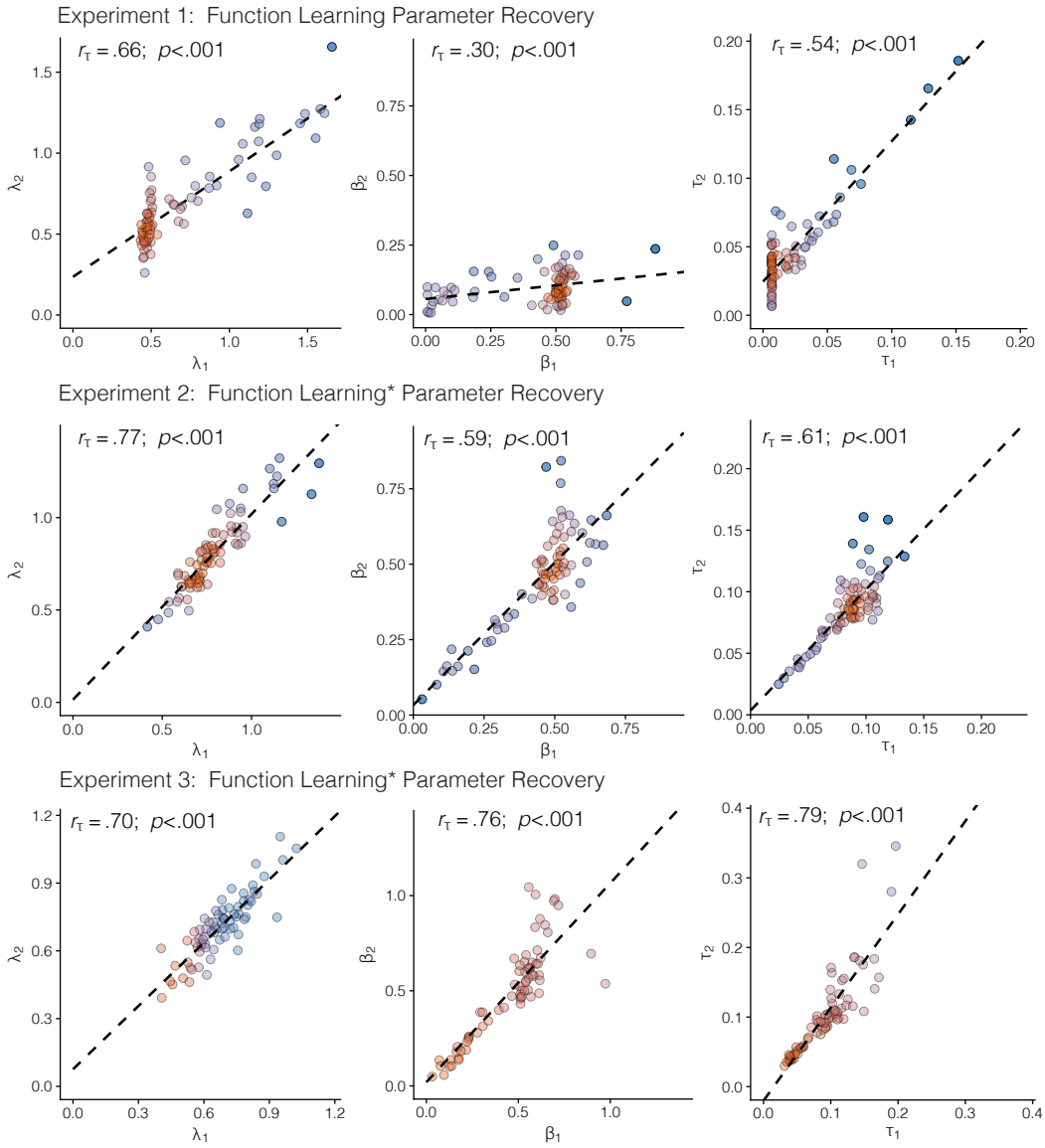

Figure S3. Parameter recovery. The generating parameter estimate is on the x-axis and the recovered parameter estimate is on the y-axis. The generating parameter estimates are from the cross-validated participant parameter estimates, which were used to simulate data. Recovered parameter estimates are the result of the cross-validated model comparison on the simulated data. While the cross-validation procedure yielded k estimates per participant, one for each round ($k_{Exp1} = 16$; $k_{Exp2} = k_{Exp3} = 8$), we show the median estimate per (simulated) participant. The dashed line shows a linear regression on the data, with the rank correlation (Kendall's tau) and p-value shown above. For readability, colours represent the bivariate kernel density estimate, with red indicating higher density. The axis limits are chosen based on 1.5× the IQR for the larger of the two values (generating or recovered parameter estimates). Thus, some outliers are omitted from these plots (2.3% in Exp. 1, 1.7% in Exp. 2, and 5.2% in Exp. 3) but all datapoints are used to calculate the rank correlations.

For Experiment 1, the rank-correlation between the generating and the recovered length-scale λ is $r_\tau = .66$, $p < .001$, $BF_\tau > 100$, the correlation between the generating and the recovered exploration factor β is $r_\tau = .30$, $p < .001$, $BF_\tau > 100$, and the correlation between the generating and the recovered softmax temperature parameter τ is $r_\tau = .54$, $p < .001$, $BF_\tau > 100$. For Experiment 2, the correlation between the generating and the recovered λ is $r_\tau = .77$, $p < .001$, $BF_\tau > 100$, for β the correlation is $r_\tau = .59$, $p < .001$, $BF_\tau > 100$, and for τ the correlation is $r =_\tau .61$, $p < .001$, $BF_\tau > 100$. For Experiment 3, the correlation between the generating and the recovered λ is $r_\tau = .70$, $p < .001$, $BF_\tau > 100$, for β the correlation is $r_\tau = .76$, $p < .001$, $BF_\tau > 100$, and for τ the correlation is $r = .79$, $p < .001$, $BF_\tau > 100$.

These results show that the rank-correlation between the generating and the recovered parameters is very high for all experiments and for all parameters. Thus, we have strong evidence to support the claim that the reported parameter estimates of the Function Learning Model (Table S3) are reliable, and therefore interpretable. Importantly, we find that estimates for β (exploration bonus) and τ (softmax temperature) are indeed separately identifiable, providing evidence for the existence of a *directed* exploration bonus[9], as a separate phenomena from noisy, undirected exploration[10] in our data.

Experimental conditions and model characteristics

To further assess how the experimental conditions influenced the model's behaviour, we performed Bayesian linear regressions of the experimental conditions onto the models' predictive accuracy and parameter estimates. To do so, we assumed a Gaussian prior on the coefficients, and an inverse Gamma prior on the conditional error variance, while inference was performed via Gibbs sampling. The results of these regressions are shown in Table S1. Whereas the smoothness of the underlying environments (in Experiments 1 and 2) had no effect on the model's predictive accuracy and almost no effect on parameter estimates (apart from a small effect on directed exploration in Experiment 1), participants in the Accumulation payoff condition showed decreased levels of directed exploration (as captured by β) in Experiment 1 and Experiment 3, and decreased levels of random exploration in Experiment 3. Thus, our model seems to capture meaningful differences between the two reward conditions in these two experiments.

Mismatched generalization

Generalized mismatch

A mismatch is defined as estimating a different level of spatial correlations (captured by the per participant λ-estimates) than the ground truth in the environment. In the main text (Fig. 4), we report a generalized Bayesian optimization simulation where we simulate every possible combination between $\lambda_0 = \{0.1, 0.2, \cdots, 1\}$ and $\lambda_1 = \{0.1, 0.2, \cdots, 1\}$, leading to 100 different combinations of student-teacher scenarios. For each of these combinations, we sample a continuous bivariate target function from a GP parameterized by λ_0 and then use the Function Learning-UCB Model parameterized by λ_1 to search for rewards. The exploration parameter β was set to 0.5 to resemble participant behaviour (Table S3). The input space was continuous between 0 and 1, i.e., any number between 0 and 1 could be chosen and GP-UCB was optimized (sometimes called the inner-optimization loop) per step using NLOPT[27] for non-linear optimization. It should be noted that instead of using a softmax choice rule, the optimization method uses an argmax rule, since the former is not defined for continuous input spaces. Additionally, since the interpretation of λ is always relative to the input range, a length-scale of $\lambda = 1$ along the unit input range would be equivalent to $\lambda = 10$ in the $x, y = [0, 10]$ input range of Experiments 2 and 3. Thus, this simulation represents a broad set of potential mismatch alignments,

Table S1. Bayesian linear regression of experimental conditions on model performance and parameter estimates.

	Predictive Accuracy R^2	Generalization λ	Exploration Bonus β	Temperature τ
Experiment 1				
Intercept	**0.23 (0.18, 0.28)**	**0.71 (0.59, 0.84)**	**0.40 (0.33, 0.47)**	**0.02 (0.01, 0.02)**
Smooth	0.02 (-0.03, 0.09)	-0.07 (-0.22, 0.09)	**0.09 (0.01, 0.18)**	0.00 (-0.01, 0.01)
Accumulator	**0.12 (0.05, 0.18)**	0.03 (-0.13, 0.18)	**-0.10 (-0.19, -0.02)**	0.00 (-0.01, 0.01)
Experiment 2				
Intercept	**0.33 (0.28, 0.37)**	**0.76 (0.69, 0.82)**	**0.50 (0.47, 0.53)**	**0.09 (0.08, 0.10)**
Smooth	0.03 (-0.02, 0.08)	0.04 (-0.03, 0.06)	0.01 (-0.03, 0.04)	0.00 (-0.01, 0.01)
Accumulator	**0.07 (0.01, 0.12)**	-0.01 (-0.08, 0.06)	0.00 (-0.04, 0.02)	-0.01 (0.00, 0.01)
Experiment 3				
Intercept	**0.28 (0.24, 0.33)**	**0.64 (0.60, 0.69)**	**0.56 (0.49, 0.63)**	**0.11 (0.10, 0.12)**
Accumulator	**0.10 (0.03, 0.16)**	0.06 (-0.01, 0.12)	**-0.15 (-0.24, -0.05)**	**-0.03 (-0.04, -0.01)**

Note: We use the Function Learning model for Experiment 1 and the localized Function Learning* model for Experiment 2 and Experiment 3. Columns indicate dependent variable, whereas rows shows independent variables' regression coefficients including 95% posterior credible sets in brackets. Boldface indicates estimates whose credible sets do not overlap with 0.

while the use of continuous inputs extends the scope of the task to an infinite state space.

Experiments 1 and 2

In both Experiments 1 and 2, we found that participant λ-estimates were systematically lower than the true value ($\lambda_{Rough} = 1$ and $\lambda_{Smooth} = 2$), which can be interpreted as a tendency to undergeneralize compared to the spatial correlation between rewards. In order to test how this tendency to undergeneralize (i.e., underestimate λ) influences task performance, we conducted two additional sets of simulations using the exact experimental design for Experiments 1 and 2 (Fig. S4a-b). These simulations used different combinations of λ values in a *teacher* kernel (x-axis) to generate environments and in a *student* kernel (y-axis), to simulate human search behaviour with the Function Learning Model.

Both teacher and student kernels were always RBF kernels, where the teacher kernel (used to generate environments) was parameterized with a length-scale λ_0 and the student kernel (used to simulate search behaviour) with a length-scale λ_1. For situations in which $\lambda_0 \neq \lambda_1$, the assumptions of the student can be seen as mismatched with the environment. The student *overgeneralizes* when $\lambda_1 > \lambda_0$ (Fig. S4a-b above the dotted line), and *undergeneralizes* when $\lambda_1 > \lambda_0$ (Fig. S4a-b below the dotted line), as was captured by our behavioural data. We simulated each possible combination of $\lambda_0 = \{0.1, 0.2, \cdots, 3\}$ and $\lambda_1 = \{0.1, 0.2, \cdots, 3\}$, leading to 900 different combinations of student-teacher scenarios. For each of these combinations, we sampled a target function from a GP parameterized by λ_0 and then used the Function Learning-UCB Model parameterized by λ_1 to search for rewards using the median parameter estimates for β and τ from the matching experiment (see Table S3).

Figures S4a-b show the results of the Experiment 1 and Experiment 2 simulations, where the colour of each tile shows the median reward obtained at the indicated trial number, for each of the 100

replications using the specified teacher-student scenario. The first simulation assessed mismatch in the univariate setting of Experiment 1 (Fig. S4a), using the median participant estimates of both the softmax temperature parameter $\tau = 0.01$ and the exploration parameter $\beta = 0.50$ and simulating 100 replications for every combination between $\lambda_0 = \{0.1, 0.2, \cdots, 3\}$ and $\lambda_1 = \{0.1, 0.2, \cdots, 3\}$. This simulation showed that it can be beneficial to undergeneralize (Fig. S4a, area below the dotted line), in particular during the first five trials. Repeating the same simulations for the bivariate setting of Experiment 2 (using the median participant estimates $\tau = 0.02$ and $\beta = 0.47$), we found that undergeneralization can also be beneficial in a more complex two-dimensional environment (Fig. S4b), at least in the early phases of learning. In general, assumptions about the level of correlations in the environment (i.e., extent of generalization λ) only influence rewards in the short term, and can disappear over time once each option has been sufficiently sampled[11].

Experiment 3

Given the robust tendency to undergeneralize in Experiments 1 and 2 (where there was a true underlying level of spatial correlation), we ran one last simulation to examine how adaptive participant λ estimates were in the real-world datasets used in Experiment 3, compared to other possible λ values. Figure S4c shows the performance of different student λ values in the range $\{0.1, 0.2, \cdots, 3\}$ simulated over 10,000 replications sampled (with replacement) from the set of 20 natural environments. Red lines show performance in terms of average cumulative reward (Accumulation criterion) and blue lines show performance in terms of maximum reward (Maximization criterion). Vertical dashed lines indicate the interquartile range of participant λ estimates. As student λ values increase, performance by both metrics typically peaks within the range of human λ estimates, with performance largely staying constant or decreasing for larger levels of λ (with the exception of average reward at $t = 40$). Thus, we find that the extent of generalization observed in participants is generally adaptive to the real-world environments they encountered. It should also be noted that higher levels of generalization beyond what we observed in participant data have only marginal benefits, yet could potentially come with additional computational costs (depending on how it is implemented). Recall that a λ of 1 corresponds to assuming the correlation of rewards effectively decays to 0 for options with a distance greater than 3. If we assume a computational implementation where information about uncorrelated options is disregarded (e.g., in a sparse GP[12]), then the range of participant λ estimates could suggest a tendency towards lower complexity and memory requirements, while sacrificing only marginal benefits in terms of either average cumulative reward or maximum reward.

Natural Environments

The environments used in Experiment 3 were compiled from various agricultural datasets[13–26] (Table S2), where payoffs correspond to normalized crop yield (by weight), and the rows and columns of the 11x11 grid correspond to the rows and columns of a field. Because agricultural data is naturally discretized into a grid, we did not need to interpolate or transform the data in any way (so as not to introduce any additional assumptions), except for the normalization of payoffs in the range $[0, 100]$, where 0 corresponds to the lowest yield and 100 corresponds to the largest yield. Note that as in the other experiments, Gaussian noise was added to each observed payoff in the experiment.

In selecting datasets, we used three inclusion criteria. Firstly, the datasets needed to be at least as large as our 11x11 grid. If the dataset was larger, we randomly sampled a 11x11 subsection from the data. Secondly, to avoid datasets where payoffs were highly skewed (e.g., with the majority of payoffs around 0 or around 100), we only included datasets where the median payoff was in the range $[25, 75]$.

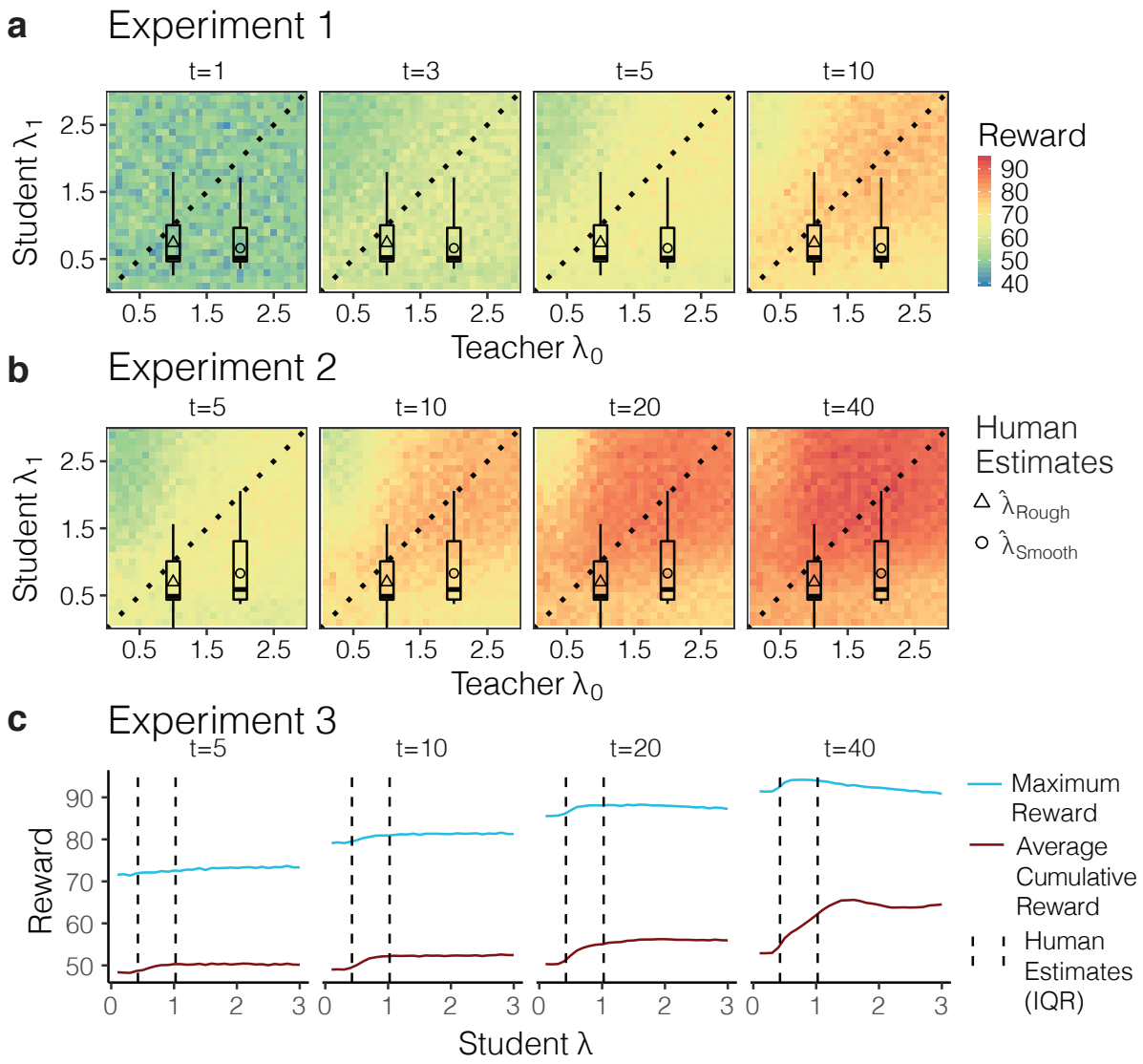

Figure S4. Mismatched length-scale (λ) simulation results. **a-b**) The teacher length-scale λ_0 is on the x-axis, the student length-scale λ_1 is on the y-axis, and each panel represents a different trial t. The teacher λ_0 values were used to generate environments, while the student λ_1 values were used to parameterize the Function Learning-UCB Model to simulate search performance. The dotted lines show where $\lambda_0 = \lambda_1$ and mark the difference between undergeneralization and overgeneralization, with points below the line indicating undergeneralization. Each tile of the heat-map indicates the median reward obtained for that particular λ_0-λ_1-combination, aggregated over 100 replications. Triangles and circles indicate mean participant λ estimates from Rough and Smooth conditions, with boxplots showing the interquartile range, the median (line), and 1.5x IQR (whiskers). **c**) Simulations with student λ values in the range $[0, 3]$ over 10,000 samples (sampled with replacement) from the set of 20 different natural environments. Red lines show average cumulative reward and blue lines show the maximum reward. Vertical dashed lines show the interquartile range of participant λ estimates.

Lastly, we required that the spatial autocorrelation of each environment (computed using Moran's I)

Table S2. Agricultural datasets used in Experiment 3

Dataset Name	Spatial Autocorrelation (Moran's I)	Crop	Source
batchelor.lemon.uniformity	0.053	Lemon	[14]
batchelor.navel1.uniformity	0.028	Navel Orange	[14]
batchelor.valencia.uniformity	0.098	Valencia Orange	[14]
draper.safflower.uniformity	0.075	Safflower	[15]
goulden.barley.uniformity	0.036	Barley	[16]
iyer.wheat.uniformity	0.047	Wheat	[17]
kalamkar.wheat.uniformity	0.004	Wheat (Yeoman II)	[18]
khin.rice.uniformity	0.011	Rice	[19]
kristensen.barley.uniformity	0.146	Barley	[20]
montgomery.wheat.uniformity	0.243	Wheat (Winter)	[21]
moore.polebean.uniformity	0.119	Blue Lake Pole Beans	[22]
moore.bushbean.uniformity	0.028	Bush Beans	[22]
moore.sweetcorn.uniformity	0.039	Sweet Corn	[22]
moore.carrots.uniformity	0.030	Carrots	[22]
moore.springcauliflower.uniformity	0.013	Spring Cauliflower	[22]
nonnecke.corn.uniformity	0.117	Sweet Corn	[23]
odland.soybean.uniformity	0.105	Soybean	[24]
odland.soyhay.uniformity	0.069	Soyhay	[24]
polson.safflower.uniformity	0.059	Safflower	[25]
stephens.sorghum.uniformity	0.043	Sorghum	[26]

be positive:

$$I = \frac{N}{W} \frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2} \tag{20}$$

where N is the total number of samples (i.e., each of the 121 sections of land in a 11x11 grid), x_i is the normalized yield (i.e., payoff) for option i, $\bar{x}$ is the mean payoff over all samples, and W is the spatial weights matrix where $w_{ij} = 1$ if i and j are the same or neighbouring samples and $w_{ij} = 0$ otherwise. Moran's I ranges between $[-1, 1]$ where intuitively $I = -1$ would resemble a checkerboard pattern (with black and white tiles reflecting the highest and lowest values in the payoff spectrum), indicating maximum difference between neighbouring samples. On the other hand, $I \to 1$ would reflect a linear step function, with maximally high payoffs on one side of the environment and maximally low payoffs on the other side. We included all environments where $I > 0$, indicating that there exists some level of positive spatial correlation that could be used by participants to guide search.

Although the structure of rewards in real-world data can sometimes be distributed differently and in particular more discretely (for example, imagine a bitmap or other structural patterns such as a checkerboard or a crop circle), we believe that our environment inclusion criteria allow us to appropriately model generalization using our pool of models, while at the same time extending the scope to more complex and challenging natural structures.

Additional Behavioural Analyses

Learning over trials and rounds

We assessed whether participants improved more strongly over trials or over rounds (Fig. S5). If they improved more over trials, this means that they are indeed finding better and better options, whereas if they are improving over rounds, this would also suggest some kind of meta-learning as they would get better at the task the more rounds they have performed previously. To test this, we fit a linear regression to every participant's outcome individually, either only with trials or only with rounds as the independent variable. Afterwards, we extract the mean standardized slopes for each participant including their standard errors. Notice that these estimates are based on a linear regression, whereas learning curves are probably non-linear. Thus, this method might underestimate the true underlying effect of learning over time.

Results (from one-sample t-tests with $\mu_0 = 0$) show that participants' scores improve significantly over trials for Experiment 1 ($t(80) = 5.57$, $p < .001$, $d = 0.6$, 95% CI $(0.2, 1.1)$, $BF > 100$), Experiment 2 ($t(79) = 2.78$, $p < .001$, $d = 0.31$, 95% CI $(-0.1, 0.8)$, $BF = 4.4$), and Experiment 3 ($t(79) = 5.91$, $p < .001$, $d = 0.7$, 95% CI $(0.2, 1.1)$, $BF > 100$). Over successive rounds, there was a negative influence on performance in Experiment 1 ($t(80) = -2.78$, $p = .007$, $d = -0.3$, 95% CI $(-0.7, 0.1)$, $BF = 4.3$), no difference in Experiment 2 ($t(79) = 0.21$, $p = .834$, $d = 0.02$, 95% CI $(-0.4, 0.5)$, $BF = 0.1$), and a minor positive influence in Experiment 3 ($t(79) = 2.16$, $p = .034$, $d = 0.2$, 95% CI $(-0.2, 0.7)$, $BF = 1.1$). Overall, participants robustly improved over trials in all experiments, with the largest effect sizes found in Experiments 1 and 3. There was no improvement over rounds in all of the experiments, suggesting that the four fully revealed example environments presented prior to the start of the task was sufficient for familiarizing participants with the task.

Individual Learning Curves

To better understand why the aggregated participant learning curves sometimes decrease in average reward over time, whereas the simulated model curves tend not to (Fig. 3b), we present individual participant learning curves in Figure S6. Here, we separate the behavioural data by horizon (colour), payoff condition (rows), and environment (columns), where each line represents a single participant. We report performance in terms of both average reward (top section: Accumulation goal) and maximum reward (bottom section: Maximization goal).

The individual learning curves reveal two main causes for the decrease in reward over time when aggregating over conditions and participants. Firstly, looking at the learning curves for participants assigned to the Accumulation condition (Fig. S6 top row), we see that roughly half of participants in the long search horizon (blue lines) show a decreasing trend at the midway point of the round. However, the other half of participants continue to gain increasingly higher rewards, more like the simulated learning curves of the Function Learning model in Figure 3b. This may be a by-product of the alternating search horizon manipulation, since the curves typically tend to decrease near the trial where a short horizon round would have ended, but also a tendency towards over-exploration that more closely resembles the Maximization goal.

Secondly, in aggregating over conditions and participants, the performance of the Accumulation and Maximization participants are averaged together. Whereas many Accumulation payoff condition participants display more positively increasing average reward, these data points are washed out by the Maximization payoff condition participants who tend to have flatter average reward curves in pursuit of the global optimization goal.

Lastly, one additional insight from the individual learning curves comes from the flat-lined maximum

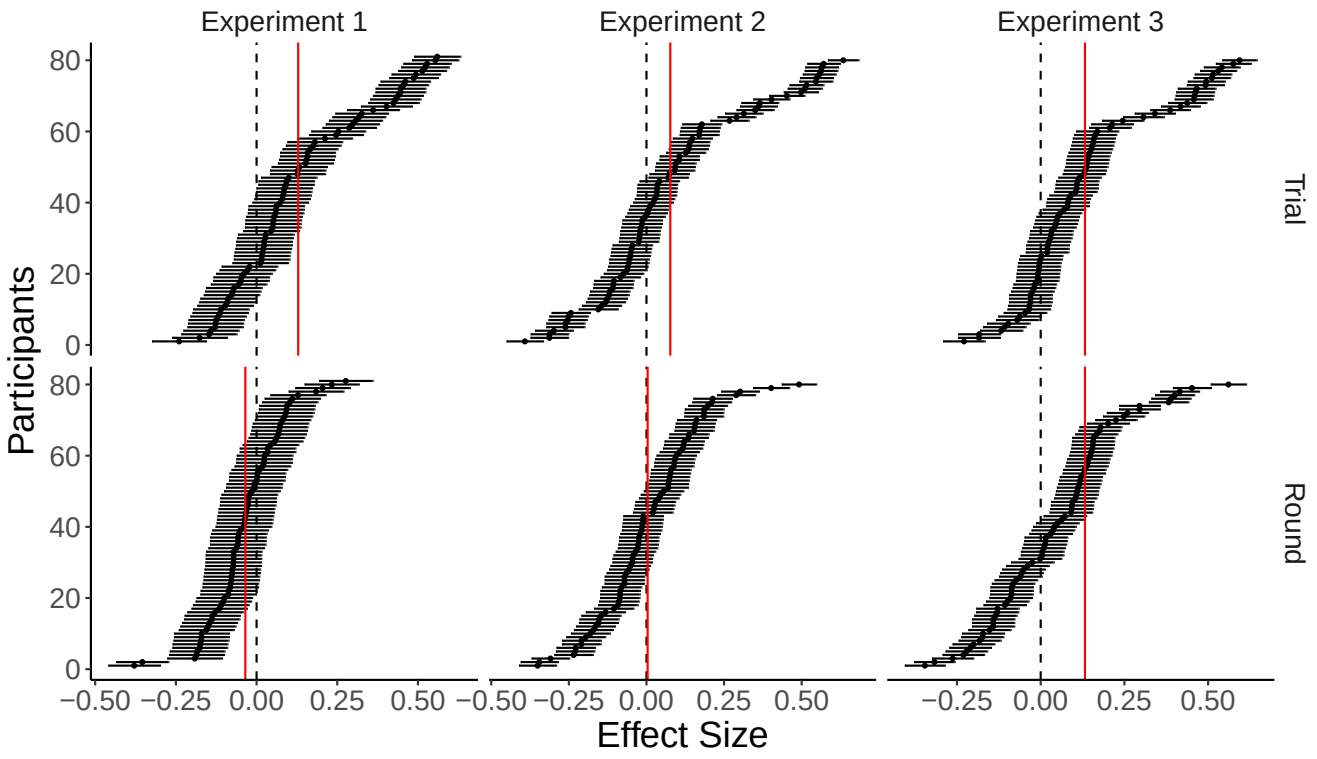

Figure S5. Learning over trials and rounds. Average correlational effect size of trial and round on score per participant as assessed by a standardized linear regression. Participants are ordered by effect size in decreasing order. Dashed lines indicate no effect. Red lines indicate average effect size.

reward lines (S6, bottom section). Found more often in Accumulation participants, these flat lines represent participants who have reached a satisfactory payoff and cease additional exploration in order to exploit it. This is yet another behavioural signature of the payoff manipulations.

Experiment Instructions

Figures S7-S9 provide screenshots from each experiment, showing the instructions provided to participants, separated by payoff condition. The top row of each figure shows the initial instructions, while the bottom row shows a set of summarized instructions provided alongside the task. Links to each of the experiments are also provided below.

- Experiment 1:
  `https://arc-vlab.mpib-berlin.mpg.de/wu/gridsearch1/experiment1.html`
- Experiment 2:
  `https://arc-vlab.mpib-berlin.mpg.de/wu/gridsearch2/experiment2.html`
- Experiment 3:
  `https://arc-vlab.mpib-berlin.mpg.de/wu/gridsearch3/experiment3.html`

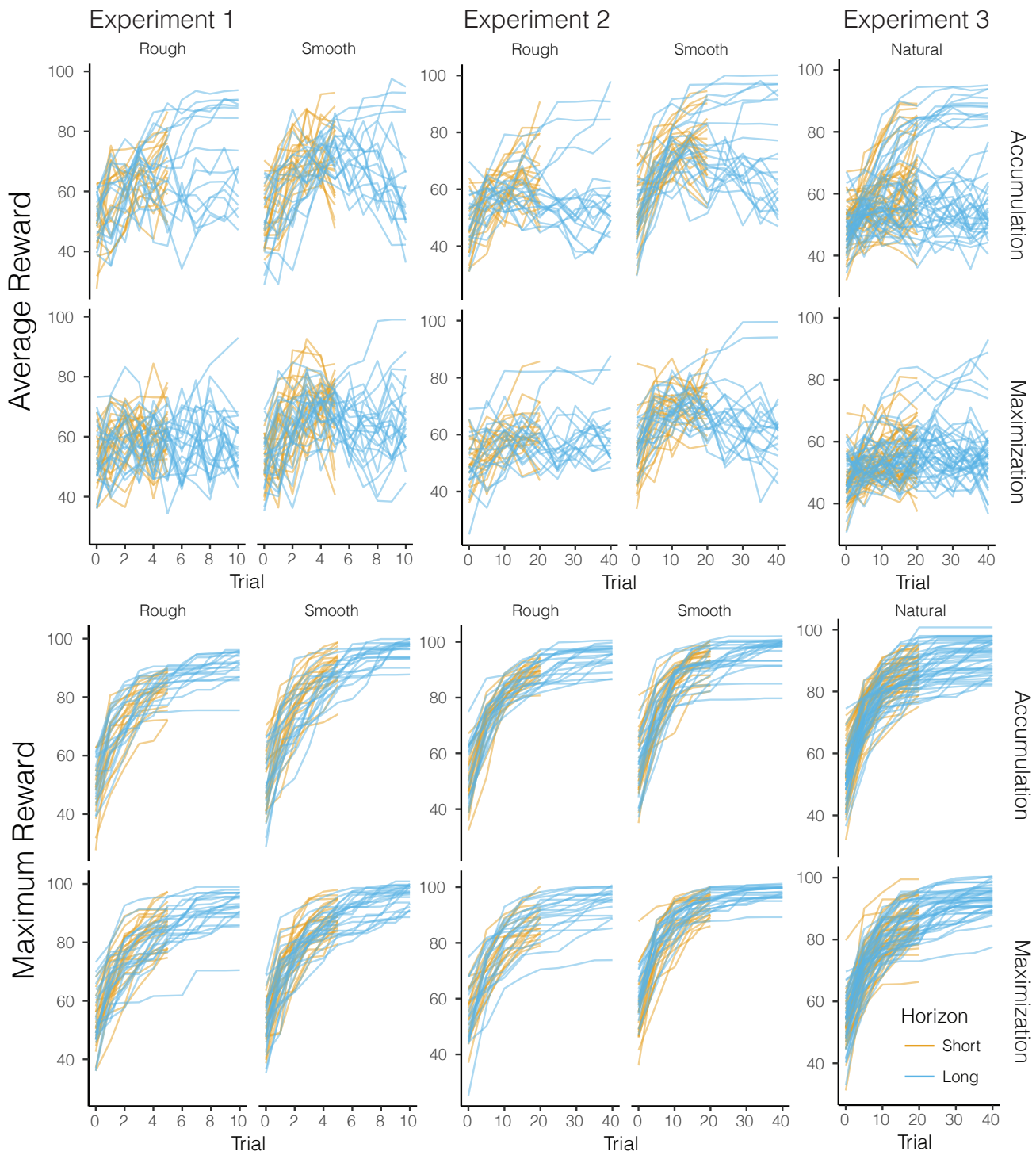

Figure S6. Individual participant learning curves. Each line represents a single participant, separated by search horizon (colour), by payoff condition (rows), and environment (columns). The top section shows performance in terms of average reward, while the bottom section shows performance in terms of maximum reward.

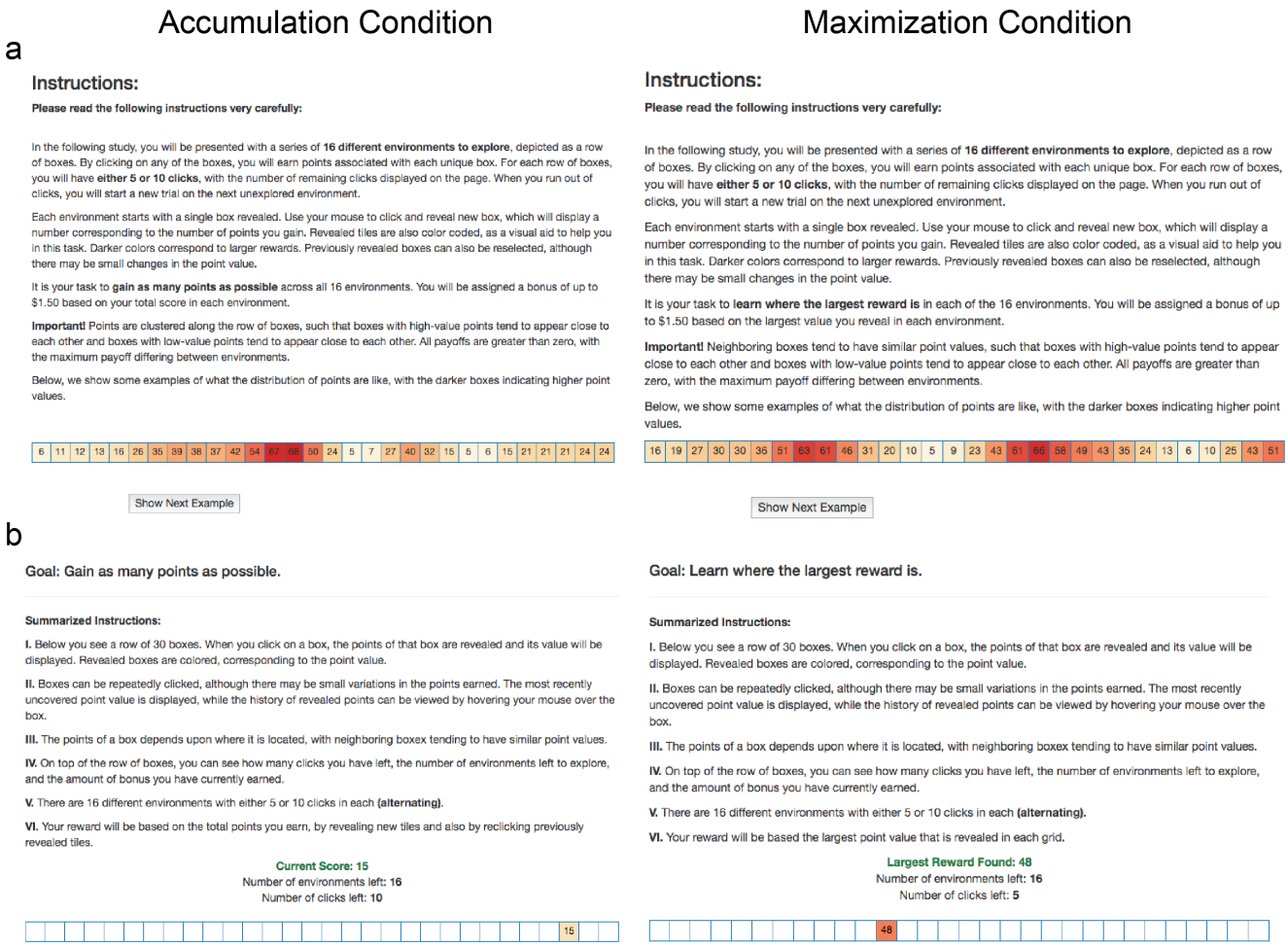

Figure S7. Screenshots from Experiment 1. Accumulation condition on the left and Maximization condition on the right. **a**) Initial instructions given to participants, followed by **b**) summarized instructions provided alongside the task.

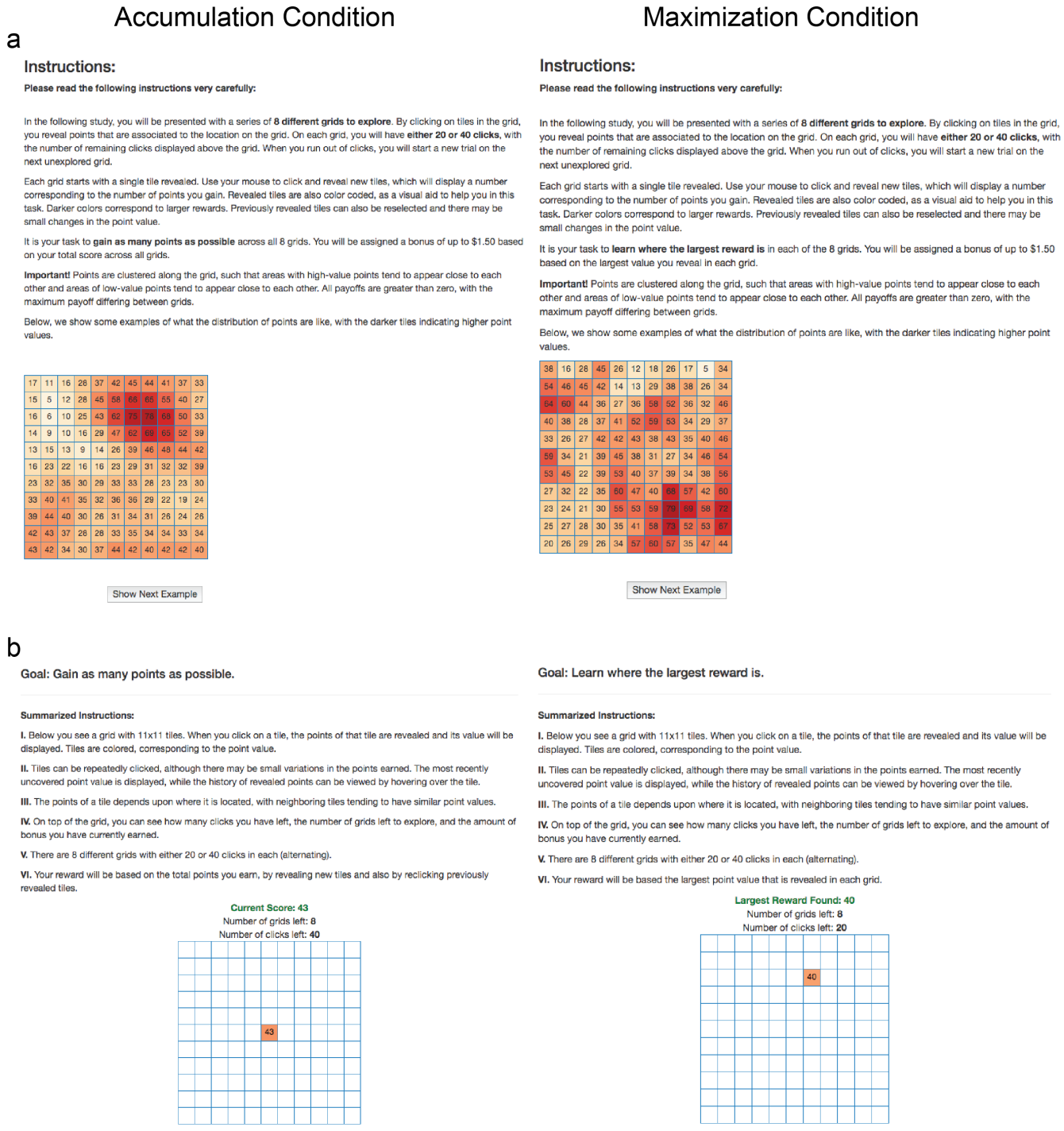

Figure S8. Screenshots from Experiment 2. Accumulation condition on the left and Maximization condition on the right. **a**) Initial instructions given to participants, followed by **b**) summarized instructions provided alongside the task.

Figure S9. Screenshots from Experiment 3. Accumulation condition on the left and Maximization condition on the right. **a**) Initial instructions given to participants, followed by **b**) summarized instructions provided alongside the task.

Table S3. Modelling Results

Model	Experiment 1							Experiment 2							Experiment 3						
	Model Comparison			Parameter Estimates				Model Comparison			Parameter Estimates				Model Comparison			Parameter Estimates			
	R^2	Best Described	pxp	Length Scale λ	Exploration Bonus β	Error Variance $\sqrt{\theta_\varepsilon^2}$	Softmax Temperature τ	R^2	Best Described	pxp	Length Scale λ	Exploration Bonus β	Error Variance $\sqrt{\theta_\varepsilon^2}$	Softmax Temperature τ	R^2	Best Described	pxp	Length Scale λ	Exploration Bonus β	Error Variance $\sqrt{\theta_\varepsilon^2}$	Softmax Temperature τ
Option Learning																					
Upper Confidence Bound	0.09	0	0	–	3.51	0.94	0.03	0.1	0	0	–	0.97	1.96	0.02	0.11	0	0	–	0.85	2.08	0.03
Pure Exploitation	0.07	1	0	–	–	54.6	54.6	0.1	0	0	–	–	148.41	148.41	0.11	0	0	–	–	148.41	148.41
Pure Exploration	0.02	0	0	–	–	0.32	0.02	0.01	0	0	–	–	15.9	0.03	0.01	0	0	–	–	5.42	0.05
Expected Improvement	0.02	0	0	–	–	0.37	0.01	0.01	0	0	–	–	1.56	0.02	0.01	0	0	–	–	0.73	0.21
Probability of Improvement	0.09	0	0	–	–	0.01	0.15	0.1	0	0	–	–	0.01	0.11	0.12	0	0	–	–	0.02	0.11
Probability of Maximum Utility	0.00	0	0	–	–	0.69	0.69	0	0	0	–	–	0.54	0.01	0.00	0	0	–	–	0.65	0.01
Option Learning*																					
Upper Confidence Bound	0.21	1	0	–	44.7	0.01	28.07	0.36	12	0	–	44.08	0.07	15.79	0.33	21	0.05	–	42.25	0.01	16.25
Pure Exploitation	0.07	1	0	–	–	54.6	0.01	0.1	0	0	–	–	148.41	148.41	0.12	0	0	–	–	148.41	148.41
Pure Exploration	0.18	0	0	–	–	0.01	0.71	0.33	3	0	–	–	0.58	0.43	0.29	5	0	–	–	0.45	0.46
Expected Improvement	0.16	0	0	–	–	0.01	0.27	0.32	0	0	–	–	0.63	0.14	0.27	0	0	–	–	0.4	0.16
Probability of Improvement	0.14	0	0	–	–	0.01	0.19	0.32	0	0	–	–	0.01	0.09	0.28	0	0	–	–	0.01	0.1
Probability of Maximum Utility	0.12	0	0	–	–	0.67	0.46	0.13	0	0	–	–	0.36	0.01	0.00	0	0	–	–	0.54	0.01
Function Learning																					
Upper Confidence Bound	**0.29**	**48**	**1**	**0.5**	**0.51**	–	**0.01**	0.24	4	0	0.54	0.47	–	0.02	0.14	2	0	0.52	0.4	–	0.02
Pure Exploitation	0.16	6	0	1.94	–	–	0.15	0.16	0	0	1.55	–	–	0.11	0.14	0	0	1.16	–	–	0.13
Pure Exploration	0.02	0	0	0.11	–	–	0.03	0.01	0	0	0.17	–	–	0.55	0.01	0	0	0.17	–	–	0.55
Expected Improvement	0.15	9	0	0.56	–	–	0.01	0.23	0	0	0.67	–	–	0.05	0.03	0	0	0.49	–	–	0.01
Probability of Improvement	0.05	0	0	3.43	–	–	0.18	0.02	0	0	0.87	–	–	0.09	0.01	0	0	0.78	–	–	0.14
Probability of Maximum Utility	0.00	0	0	0.69	–	–	7.17	0.02	0	0	0.49	–	–	0.01	0.00	0	0	0.42	–	–	0.01
Function Learning*																					
Upper Confidence Bound	0.23	10	0	0.96	0.54	–	0.16	**0.38**	**60**	**1**	**0.76**	**0.49**	–	**0.09**	**0.33**	**47**	**0.95**	**0.67**	**0.52**	–	**0.1**
Pure Exploitation	0.16	1	0	7.13	–	–	0.12	0.23	0	0	14.4	–	–	0.06	0.18	0	0	10.87	–	–	0.06
Pure Exploration	0.14	3	0	0.08	–	–	0.32	0.27	0	0	0.17	–	–	.19	0.23	2	0	0.17	–	–	0.2
Expected Improvement	0.09	1	0	0.71	–	–	0.11	0.23	1	0	0.67	–	–	0.05	0.17	0	0	0.64	–	–	0.06
Probability of Improvement	0.12	0	0	7.14	–	–	0.2	0.24	0	0	0.84	–	–	0.09	0.19	0	0	0.72	–	–	0.1
Probability of Maximum Utility	0.12	0	0	0.67	–	–	0.46	0.12	0	0	0.46	–	–	0.01	0.08	0	0	0.27	–	–	0.01
Simple Heuristics																					
Win-Stay Lose-Sample	0.00	0	0	–	–	–	3.72	0.05	0	0	–	–	–	0.32	0.03	0	0	–	–	–	0.39
Win-Stay Lose-Sample*	0.05	0	0	–	–	–	0.73	0.26	0	0	–	–	–	0.22	0.21	1	0	–	–	–	0.24
Local Search	0.12	0	0	–	–	–	0.46	0.28	0	0	–	–	–	0.22	0.25	2	0	–	–	–	0.23

Note: R^2 indicates out-of-sample predictive accuracy, "Best Described" indicates the number of participants in each experiment that were best described by a model, and pxp is the protected probability of exceedance[28] using the model's out-of-sample log-evidence. Parameter estimates are the median over all participants. There were 81 participants in Experiment 1, 80 participants in Experiment 2, and 80 participants in Experiment 3. The best performing model for each experiment is highlighted in boldface. Asterisks (*) indicate a localized variant of a model.

Supplementary References

1. Gigerenzer, G. Todd, P., & ABC Research Group *Simple heuristics that make us smart* (Oxford University Press, 1999).
2. Schulz, E., Speekenbrink, M. & Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology* **85**, 1–16 (2018).
3. Speekenbrink, M. & Konstantinidis, E. Uncertainty and exploration in a restless bandit problem. *Topics in Cognitive Science* **7**, 351–367 (2015).
4. Bonawitz, E., Denison, S., Gopnik, A. & Griffiths, T. L. Win-stay, lose-sample: A simple sequential algorithm for approximating bayesian inference. *Cognitive Psychology* **74**, 35–65 (2014).
5. Christakou, A. *et al.* Disorder-specific functional abnormalities during sustained attention in youth with attention deficit hyperactivity disorder (adhd) and with autism. *Molecular psychiatry* **18**, 236–244 (2013).
6. Gershman, S. J., Pesaran, B. & Daw, N. D. Human reinforcement learning subdivides structured action spaces by learning effector-specific values. *Journal of Neuroscience* **29**, 13524–13531 (2009).
7. Mullen, K., Ardia, D., Gil, D., Windover, D. & Cline, J. DEoptim: An R package for global optimization by differential evolution. *Journal of Statistical Software* **40**, 1–26 (2011).
8. Wagenmakers, E. J., Verhagen, J. & Ly, A. How to quantify the evidence for the absence of a correlation. In *Behavior Research Methods*, 413–426 (2016).
9. Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A. & Cohen, J. D. Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General* **143**, 2074–2081 (2014).
10. Daw, N. D., O'doherty, J. P., Dayan, P., Seymour, B. & Dolan, R. J. Cortical substrates for exploratory decisions in humans. *Nature* **441**, 876–879 (2006).
11. Srivastava, V., Reverdy, P. & Leonard, N. E. Correlated multiarmed bandit problem: Bayesian algorithms and regret analysis. *arXiv preprint arXiv:1507.01160* (2015).
12. Herbrich, R., Lawrence, N. D. & Seeger, M. Fast sparse gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, 625–632 (2003).
13. Wright, K. *agridat: Agricultural Datasets* (2017). URL `https://CRAN.R-project.org/package=agridat`. R package version 1.13.
14. Batchelor, L. & Reed, H. Relation of the variability of yields of fruit trees to the accuracy of field trials. *Journal of Agricultural Research* **12**, 461–468 (1918).
15. Draper, A. D. *Optimum plot size and shape for safflower yield tests*. Ph.D. thesis, The University of Arizona. (1959).
16. Goulden, C. H. *Methods of statistical analysis* (John Wiley and Sons, Inc., 1939).
17. Krishna Iyer, P. V. Studies with wheat uniformity trial data. i. size and shape of experimental plots and the relative efficiency of different layouts. *The Indian Journal of Agricultural Science* **12**, 240–262 (1942).
18. Kalamkar, R. A study in sampling technique with wheat. *The Journal of Agricultural Science* **22**, 783–796 (1932).

19. Khin, S. *Investigation into the relative costs of rice experiments based on the efficiency of designs*. Ph.D. thesis, University of the West Indies (2016).

20. Kristensen, R. Anlaeg og opgoerelse af markforsoeg. *Tidsskrift for landbrugets planteavl* **31** (1925).

21. Montgomery, E. Variation in yield and methods of arranging plats to secure comparative results. In *Twenty-Fifth Annual Report of the Agricultural Experiment Station of Nebraska*, 164–180 (1912).

22. Moore, J. F. & Darroch, J. *Field plot technique with Blue Lake pole beans, bush beans, carrots, sweet corn, spring and fall cauliflower* (Washington Agricultural Experiment Stations, Institute of Agricultural Sciences, State College of Washington, 1956).

23. Nonnecke, I. The precision of field experiments with vegetable crops as influenced by plot and block size and shape: I. sweet corn. *Canadian Journal of Plant Science* **39**, 443–457 (1959).

24. Odland, T. & Garber, R. Size of plat and number of replications in field experiments with soybeans. *Journal of the American Society of Agronomy* (1928).

25. Polson, D. E. *Estimation of Optimum Size, Shape, and Replicate Number of Safflower Plots for Yield Trials*. Ph.D. thesis, Utah State University (1964).

26. Stephens, J. C. & Vinall, H. Experimental methods and the probable error in field experiments with sorghum. Tech. Rep. (1928).

27. Johnson, S. G. The nlopt nonlinear-optimization package (2014). URL `http://ab-initio.mit.edu/nlopt`.

28. Stephan, K. E., Penny, W. D., Daunizeau, J., Moran, R. J. & Friston, K. J. Bayesian model selection for group studies. *Neuroimage* **46**, 1004–1017 (2009).