

1 Exploration and generalization in vast spaces

2 Charley M. Wu^{1,*,+}, Eric Schulz^{2,+}, Maarten Speekenbrink³, Jonathan D. Nelson^{1,4}, and
3 Björn Meder^{1,5}

4 ¹Center for Adaptive Rationality, Max Planck Institute for Human Development, Berlin, Germany

5 ²Department of Psychology, Harvard University, Cambridge, Massachusetts, USA

6 ³Department of Experimental Psychology, University College London, London, UK

7 ⁴School of Psychology, University of Surrey, Guildford, UK

8 ⁵MPRG iSearch, Max Planck Institute for Human Development, Berlin, Germany

9 *cwu@mpib-berlin.mpg.de

10 +these authors contributed equally to this work

11 ABSTRACT

Foraging for food, developing new medicines, and learning complex games are search problems with vast numbers of possible actions. Yet, under real-world time or resource constraints, optimal solutions are generally unobtainable. How do humans generalize and learn which actions to take when not all options can be explored? We present two behavioral experiments in which the spatial correlation of rewards provides traction for generalization, yet a limited search horizon allows for exploration of only a small fraction of all available options. We competitively test 27 different probabilistic and heuristic models for making out-of-sample predictions of individual's search decisions. Our results show that a Gaussian Process function learning model, combined with an optimistic Upper Confidence Bound sampling strategy, robustly captures how humans use generalization to guide search behavior. Taken together, these two form a model of exploration and generalization that leads to reproducible and psychologically meaningful parameter estimates, providing novel insights into the nature of human search in vast spaces. We find a systematic—yet sometimes beneficial—tendency towards undergeneralization, as well as strong evidence for the separate phenomena of directed and undirected exploration. Our modeling results and parameter estimates are recoverable, and can be used to simulate human-like performance, bridging a critical gap between human and machine learning.

13 Introduction

14 From engineering proteins for medical treatment¹ to mastering the game of Go², many complex tasks
15 can be described as search problems³. Frequently, these tasks come with a vast space of possible actions,
16 each corresponding to some reward that can only be observed through experience. In such problems,
17 one must learn to balance the dual goals of exploring unknown options, whilst also exploiting existing
18 knowledge for immediate returns. This frames the *exploration-exploitation dilemma*, typically studied
19 using the multi-armed bandit framework^{*4–6}, with the assumption that each option has its own reward
20 distribution to be learned independently. Yet under real-world constraints of limited time or resources, it is
21 not enough to know *when* to explore, but also *where*.

22 Human learners are able to quickly adapt to unfamiliar environments, where the same situation is
23 rarely encountered twice^{7,8}. This highlights an intriguing gap between human and machine learning,
24 where traditional approaches to reinforcement learning typically learn about the distribution of rewards for
25 each state independently⁹. Such an approach falls short in more realistic scenarios where it is impossible

*The multi-armed bandit is a metaphor for a row of slot machines in a casino, where each slot machine has an independent payoff distribution. Solutions to the problem propose different policies for how to learn about which arms are better to play (exploration), while also playing known high-value arms to maximize reward (exploitation).

26 to try all possible actions in all possible states^{10,11}. How could an intelligent agent, biological or machine,
27 learn which actions to take when not all options can be explored?

28 In computer science, one method for dealing with vast state spaces is to use *function learning* as a
29 mechanism to generalize prior experience to unobserved states¹². The function learning approach relates
30 different state-action pairs to each other by approximating a global value function over all states and
31 actions, including ones not experienced yet⁸. This allows for generalization to vast and potentially infinite
32 state spaces, based on a small number of observations. Additionally, function learning scales to problems
33 with complex sequential dynamics and has been used in tandem with restricted search methods, such as
34 Monte Carlo sampling, for navigating intractably large search trees^{2,13}. While restricted search methods
35 have been proposed as models of human reinforcement learning in planning tasks^{14,15}, here we focus on
36 situations in which a rich model of environmental structure supports learning and generalization¹⁶.

37 Function learning has been successfully utilized for adaptive generalization in various machine learning
38 applications^{17,18}, although relatively little is known about how humans generalize *in vivo* (e.g., in a search
39 task). Building on previous work exploring inductive biases in pure function learning contexts^{19,20}, and
40 human behavior in univariate function optimization²¹, we present the first comprehensive research on how
41 people utilize generalization to effectively learn and search for rewards in large state spaces. Across two
42 studies using uni- and bivariate versions of a multi-armed bandit, we compare 27 different models in their
43 ability to predict individual human behavior.

44 In both experiments, the vast majority of individual subjects are best captured by a model combining
45 function learning using Gaussian Process (\mathcal{GP}) regression, with an optimistic Upper Confidence Bound
46 (UCB) sampling strategy that directly balances expectations of reward with the reduction of uncertainty.
47 Importantly, we recover meaningful and robust estimates of the nature of human generalization, showing
48 the limits of traditional models of associative learning²² in tasks where the environmental structure
49 supports learning and inference. Interestingly, the most predictive model of the behavioral data is also
50 currently the only known Bayesian optimization algorithm with competitive performance guarantees²³.
51 This result has rich theoretical implications for reinforcement learning and the study of intelligent human
52 behavior.

53 The main contributions of this paper are threefold.

- 54 1. We introduce the *spatially correlated multi-armed bandit* paradigm as a method for studying the
55 extent to which people use generalization to guide search, in a far more complex problem space
56 than traditionally used to study human behavior.
- 57 2. We find that a Bayesian model of function learning robustly captures how humans generalize and
58 learn about the structure of the environment, where an observed tendency towards undergeneraliza-
59 tion is shown to sometimes be beneficial.
- 60 3. We show that participants solve the exploration-exploitation dilemma by optimistically inflating
61 expectations of reward by the underlying uncertainty, with recoverable evidence for the sepa-
62 rate phenomena of directed exploration (towards reducing uncertainty) and random, undirected
63 exploration.

64 Results

65 A useful inductive bias in many real world search tasks is to assume spatial correlation between rewards
66 (i.e., clumpiness of resource distribution;²⁴). This is equivalent to assuming that actions will yield similar
67 outcomes in nearby locations. We present human data and modeling results from two experiments

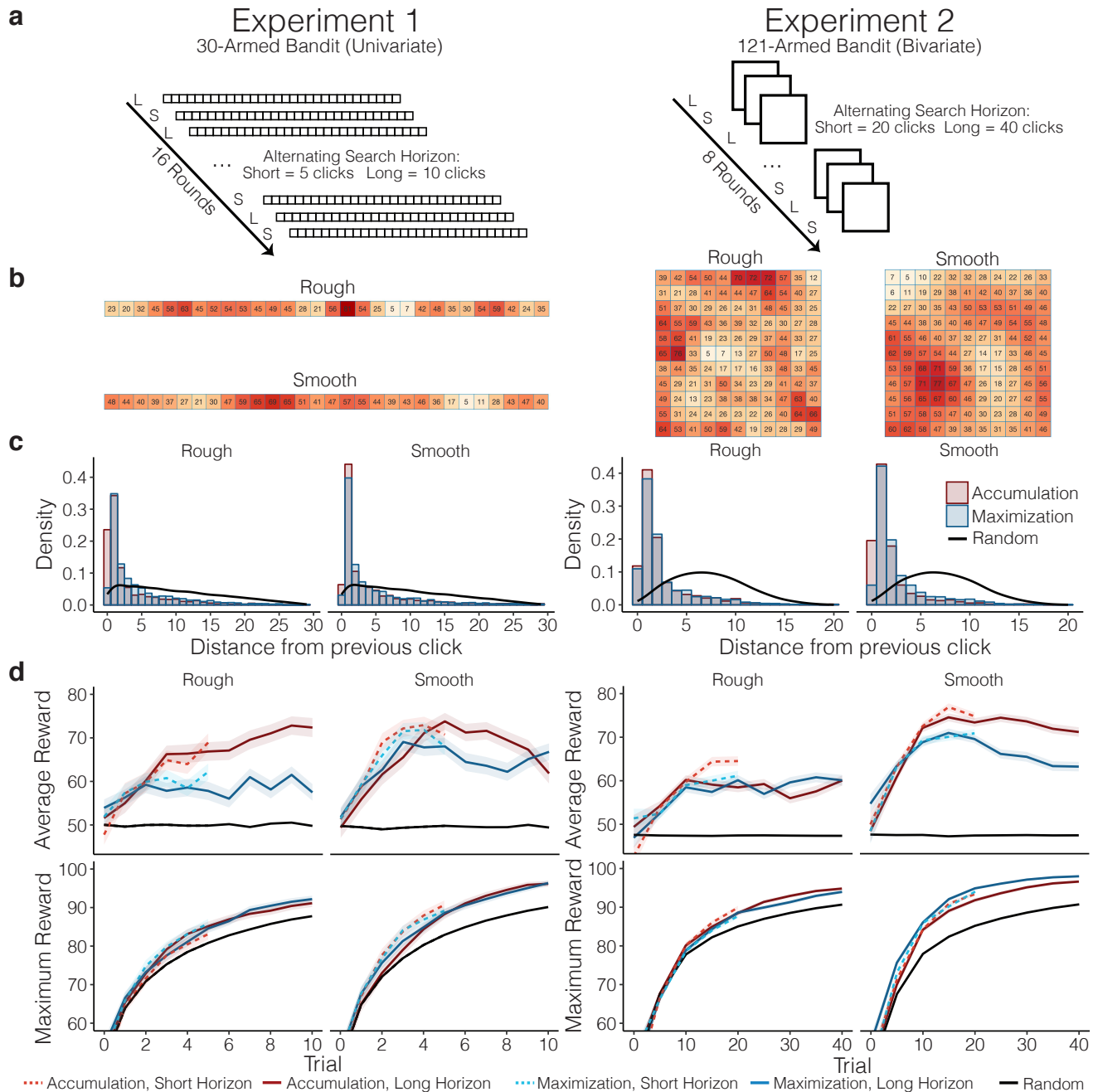


Figure 1. Procedure and behavioral results. Both experiments used a 2×2 between-subject design, manipulating the type of environment (Rough or Smooth) and the payoff condition (Accumulation or Maximization). **a**) Experiment 1 used a 1D array of 30 possible options, while Experiment 2 used a 2D array (11×11) with 121 options. Experiments took place over 16 (Experiment 1) or 8 (Experiment 2) rounds, with a new environment sampled without replacement for each round. Search horizons alternated between rounds, with the horizon length of the first trial counter-balanced between subjects. **b**) Examples of fully revealed search environments, where tiles were initially blank at the beginning of each round, except for a single randomly revealed tile. Rough and Smooth environments differed in the extent of spatial correlations (see Methods). **c**) Locality of sampling behavior compared to a random baseline simulated over 10,000 rounds (black line), where distance is measured using Manhattan distance. **d**) Average reward earned (Accumulation goal) and maximum reward revealed (Maximization goal), where colored lines indicate the assigned payoff condition and shaded regions show the standard error of the mean. Short horizon trials are indicated by lighter colors and dashed lines, while black lines are a comparison to a random baseline simulated over 10,000 rounds.

68 using spatially correlated multi-armed bandits on univariate (Experiment 1) and bivariate (Experiment 2)
69 environments (Fig. 1). The spatial correlation of rewards provides a context to each arm of the bandit,
70 which can be learned and used to generalize to yet unobserved contexts, thereby guiding search decisions.
71 Additionally, since recent work has connected both spatial and conceptual representations to a common
72 neural substrate²⁵, our results in a spatial domain provide potential pathways to other search domains,
73 such as contextual^{26,27} or semantic search^{28,29}.

74 Experiment 1

75 Participants searched for rewards on a 1×30 grid world, where each tile represented a reward-generating
76 arm of the bandit (Fig. 1a). The mean rewards of each tile were spatially correlated, with stronger
77 correlations in *Smooth* than in *Rough* environments (between subjects; Fig. 1b). Participants were either
78 assigned the goal of accumulating the largest average reward (*Accumulation condition*), thereby balancing
79 exploration-exploitation, or of finding the best overall tile (*Maximization condition*), an exploration goal
80 directed towards finding the global maximum. Additionally, the search horizons alternated between rounds
81 (*Short* = 5 vs. *Long* = 10), with the order counter-balanced between subjects. We hypothesized that,
82 if search behavior is guided by function learning, participants would perform better and learn faster in
83 smooth environments, in which stronger spatial correlations reveal more information about nearby tiles³⁰.

84 Looking first at sampling behavior, the distance between sequential choices was more localized than
85 chance ($t(160) = 31.2, p < .001, d = 1.92$; Fig. 1c)[†], as has also been observed in semantic search²⁸ and
86 causal learning³¹ domains. Participants in the Accumulation condition sampled more locally than those in
87 the Maximization condition ($t(79) = 3.33, p = .001, d = 0.75$), corresponding to the increased demand
88 to exploit known or near known rewards. Comparing performance in different environments, the learning
89 curves in Fig. 1d show that participants in Smooth environments obtained higher average rewards than
90 participants in Rough environments ($t(79) = 3.58, p < .001, d = 0.8$), consistent with the hypothesis that
91 spatial patterns in the environment can be learned and used to guide search. Additionally, longer search
92 horizons (solid vs. dashed lines) did not lead to higher average reward ($t(80) = 0.60, p = .549, d = 0.07$).
93 We analyzed both average reward and the maximum reward obtained for each subject, irrespective of their
94 payoff condition (Maximization or Accumulation). Participants in the Accumulation condition performed
95 better than participants in the Maximization condition on the average reward criterion ($t(79) = 2.89,$
96 $p = .005, d = 0.65$), yet remarkably, they performed equally well in terms of finding the largest overall
97 reward ($t(79) = -0.73, p = .467, d = 0.16$). Thus, a strategy balancing exploration and exploitation, at
98 least for human learners, may achieve the global optimization goal *en passant*.

99 Experiment 2

100 Experiment 2 had the same design as Experiment 1, but used a 11×11 grid representing an underlying
101 bivariate reward function (Fig. 1 right). We replicated the main results of Experiment 1, showing
102 that participants sampled more locally than a random baseline ($t(158) = 42.7, p < .001, d = 4.47$; Fig.
103 1c), participants in the Accumulation condition sampled more locally than those in the Maximization
104 condition ($t(78) = 2.75, p = .007, d = 0.61$), and overall, participants obtained higher rewards in Smooth
105 environments than in Rough environments ($t(78) = 6.55, p < .001, d = 1.47$; Fig. 1d). For both
106 locality (compared to a random baseline) and the difference between environments, the effect size was
107 larger in Experiment 2 than in Experiment 1. We also replicated the result that participants in the
108 Accumulation condition were as good as participants in the Maximization condition at discovering the
109 highest rewards ($t(78) = -0.62, p = .534, d = 0.14$), yet in Experiment 2, the Accumulation condition did
110 not lead to substantially better performance than the Maximization condition in terms of average reward

[†]All reported *t*-tests are two-sided.

111 ($t(78) = -1.31, p = .192, d = 0.29$). Again, short search horizons led to the same level of performance
112 as longer horizons, ($t(79) = -0.96, p = .341, d = 0.11$), suggesting that for participants, frugal search
113 can be quite efficient. For full results on learning over rounds and trials see Fig. S3.

114 Modeling Generalization and Search

115 We competitively tested a diverse set of 27 different models in their ability to predict each subject's trial-
116 by-trial choices (for full results see Fig. S1 and Table S1). These models include different combinations of
117 *models of learning* and *sampling strategies*, which map onto the distinction between belief and sampling
118 models, central to theories in statistics³², psychology³³, and philosophy of science³⁴. Models of learning
119 form inductive beliefs about the value of possible actions based on previous observations, while sampling
120 strategies transform these beliefs into probabilistic predictions about where a participant will sample next.
121 We also consider *simple heuristics*, which make predictions about search behavior without maintaining a
122 model of the world (see SI). By far the best predictive models used *Gaussian Process* (\mathcal{GP}) regression^{35,36}
123 as a method for learning an underlying value function relating all state-action contexts to each other, and
124 *Upper Confidence Bound* (UCB) sampling³⁷.

125 Function learning provides an explanation of how individuals generalize from previous experience to
126 untested actions, by adaptively learning an underlying function mapping actions onto rewards. We use
127 \mathcal{GP} regression as an expressive model of human function learning, which in contrast to neural network
128 function approximators³⁸ can produce psychologically interpretable parameter estimates about the extent
129 to which generalization occurs (i.e., the strength of generalization as a function of spatial distance). \mathcal{GP}
130 function learning can guide search by making normally distributed predictions about the expected mean
131 $m(\mathbf{x})$ and the underlying uncertainty $s(\mathbf{x})$ (estimated here as a standard deviation; see Methods for details)
132 for each option \mathbf{x} in the global state space (see Fig. 2b), conditioned on a finite number of previous
133 observations of rewards $\mathbf{y}_T = [y_1, y_2, \dots, y_T]^\top$ at inputs $\mathbf{X}_T = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$. Similarities between options
134 are modeled by a *Radial Basis Function* (RBF) kernel:

$$135 \quad k_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\lambda}\right), \quad (1)$$

136

137 where λ governs how quickly correlations between points \mathbf{x} and \mathbf{x}' (e.g., two tiles on the grid) decay towards
138 zero as their distance increases. We use λ as a free parameter, which can be interpreted psychologically
139 as the extent to which people generalize spatially. Since the \mathcal{GP} prior is completely defined by the RBF
140 kernel, the underlying mechanisms are similar to Shepard's universal gradient of generalization³⁹, which
141 also models generalization as an exponentially decreasing function of distance.

142 Given estimates about expected rewards $m(\mathbf{x})$ and the underlying uncertainty $s(\mathbf{x})$, UCB sampling
143 generates a value for each action x , allowing us to make predictions about where participants will search
144 next (Fig. 2c) using a weighted sum:

$$145 \quad \text{UCB}(\mathbf{x}) = m(\mathbf{x}) + \beta s(\mathbf{x}), \quad (2)$$

146

147 where β is a free parameter governing how much the reduction of uncertainty is weighted relative to
148 expectations of reward. This trade-off between exploiting known high-value rewards and exploring to
149 reduce uncertainty⁴⁰ can be interpreted as optimistically inflating expected rewards by their attached
150 uncertainty, and can be decomposed into two separate components that only sample points based on high
151 expected reward (Pure Exploitation) or high uncertainty (Pure Exploration).

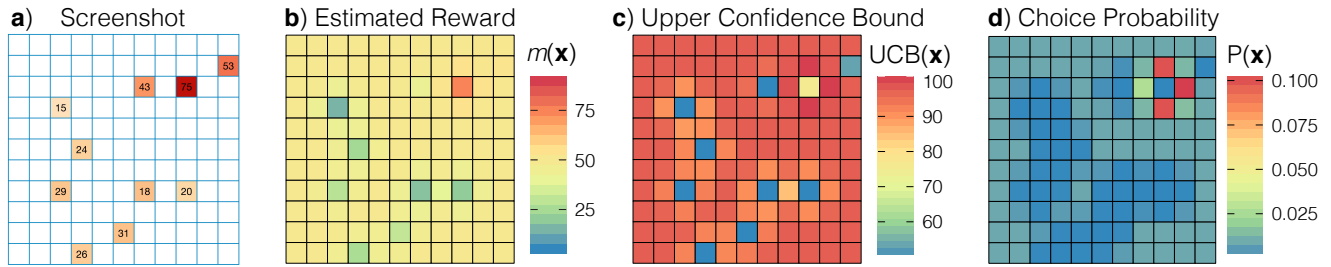


Figure 2. Overview of the Function Learning-UCB model specified using median participant parameter estimates (see Table S1). **a)** Screenshot of Experiment 2. Participants were allowed to select any tile until the search horizon was exhausted. **b)** Estimated reward (not shown, the estimated uncertainty) as predicted by the \mathcal{GP} Function Learning model, based on the points sampled in Panel **a)**. **c)** Upper confidence bound of predicted rewards. **d)** Choice probabilities after a softmax choice rule. $P(\mathbf{x}) = \exp(\text{UCB}(\mathbf{x})/\tau) / \sum_{j=1}^N \exp(\text{UCB}(\mathbf{x}_j)/\tau)$, where τ is the temperature parameter (i.e., higher temperature values lead to more random predictions).

$$\text{PureExploit}(\mathbf{x}) = m(\mathbf{x}) \quad (3)$$

$$\text{PureExplore}(\mathbf{x}) = s(\mathbf{x}) \quad (4)$$

Figure 2 shows how the Function Learning-UCB model makes inferences about the search space, and uses UCB sampling (with a softmax choice rule) to make probabilistic predictions about where the participant will sample next. We refer to this model as the *Function Learning model* and contrast it with an *Option Learning model*. The Option Learning model uses a Bayesian Mean Tracker (BMT) to learn about the distribution of rewards for each option independently (see Methods). The Option learning model is a type of traditional associative learning model, and can be understood as a variant of a Kalman filter without temporal dynamics⁵. Like the Function Learning model, the Option Learning model also generates normally distributed predictions $m(\mathbf{x})$ and $s(\mathbf{x})$, which we combine with the same set of sampling strategies and the same softmax choice rule to make probabilistic predictions about search. We use the softmax temperature parameter (τ) to estimate the amount of undirected exploration (i.e., higher temperatures correspond to more random sampling), in contrast to the β parameter of UCB, which estimates the level of exploration directed towards reducing uncertainty.

Modeling results

Experiment 1

Instead of learning rewards for each state independently, as assumed by the Option Learning model, participants were better described by the Function Learning model ($t(80) = 14.01$, $p < .001$, $d = 1.56$; comparing cross-validated predictive accuracies, both using UCB sampling), providing evidence against the assumption of state independence. Furthermore, by decomposing the UCB sampling algorithm into Pure Exploitation or Pure Exploration components, we show that both expectations of reward and estimates of uncertainty are necessary components for the Function Learning model to predict human search behavior, with Pure Exploitation ($t(80) = 8.85$, $p < .001$, $d = 0.98$) and Pure Exploration ($t(80) = 16.63$, $p < .001$, $d = 1.85$) variants of the model performing worse at predicting human behavior than the combined UCB algorithm.

Because of the observed tendency to sample locally (Fig. 1c), we created a localized variant of both Option Learning and Function Learning models (indicated by an asterisk*; Fig. 3a), giving larger weight to options closer to the previous selected option, without introducing additional free parameters (see

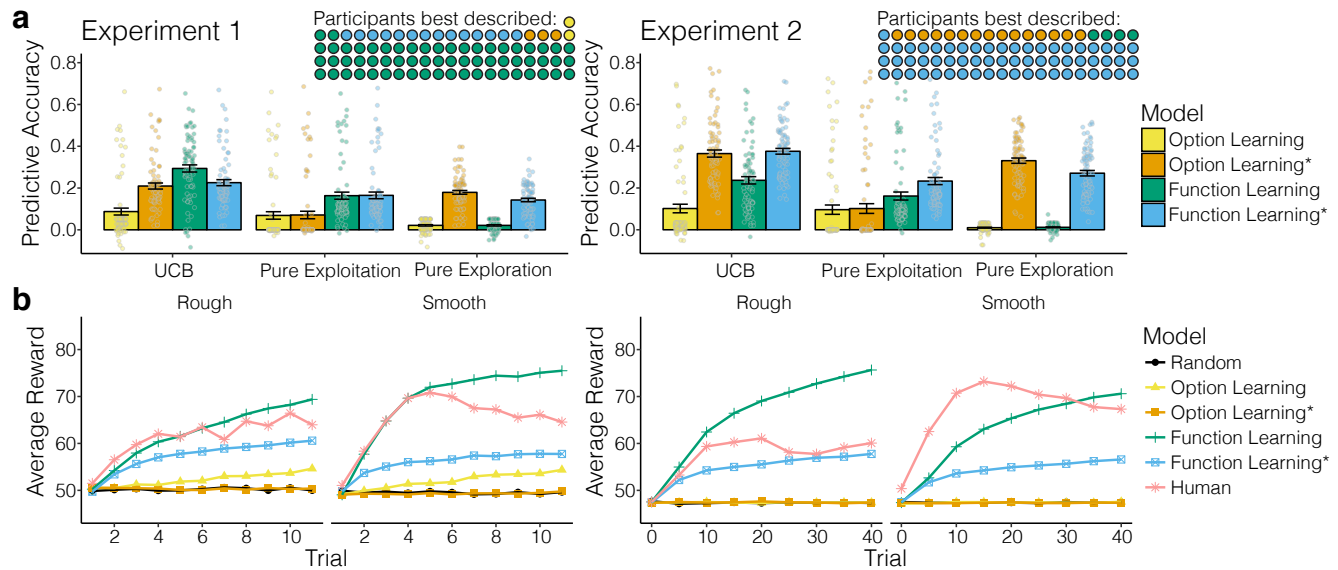


Figure 3. Modeling results. **a)** Cross-validated predictive accuracy of each model, with bars indicating the group mean (\pm SEM). Each individual participant is shown as a single dot, with the number of participants best described shown as an icon array (inset; aggregated by sampling strategies). Asterisks (*) indicate a localized variant of the Option Learning or Function Learning models, where predictions are weighted by the inverse distance from the previous choice (see Methods). **b)** Averaged learning curves of participants and models (combined with UCB-sampling only) simulated over 10,000 replications using sampled participant parameter estimates. Learning curves (and parameter estimates) are separated by environment, aggregated over payoff conditions and search horizons.

181 Methods). While the Option Learning*-UCB model with localization was better than the standard Option
 182 Learning-UCB model ($t(80) = 16.13, p < .001, d = 1.79$), the standard Function Learning-UCB model
 183 still outperformed its localized variant ($t(80) = -5.05, p < .001, d = 0.56$).

184 Overall, 56 out of 81 participants were best described by the Function Learning-UCB model, with an
 185 additional 10 participants best described by the Function Learning*-UCB model with localization. Figure
 186 3b shows simulated learning curves of each model in comparison to human performance, where models
 187 were specified using parameters sampled from participant estimates (10,000 samples with replacement).
 188 Whereas both versions of the the Option Learning-UCB model barely beat the performance of a completely
 189 random baseline, both standard and localized versions of the Function Learning-UCB model behave
 190 sensibly and improve performance over time. This suggests there exists some overlap between the
 191 elements of human intelligence responsible for successful performance in our task, and the elements of
 192 participant behavior captured by the Function Learning-UCB model.

193 Looking more closely at the parameter estimates of the Function Learning-UCB model (Fig. 4),
 194 we find that people tend to underestimate the extent of spatial correlations, with estimated λ values
 195 significantly lower than the ground truth ($\lambda_{Smooth} = 2$ and $\lambda_{Rough} = 1$) for both Smooth (mean estimate:
 196 $\hat{\lambda} = 0.82, t(41) = -17.60, p < .001, d = 2.71$) and Rough environments ($\hat{\lambda} = 0.78, t(38) = -3.89,$
 197 $p < .001, d = 0.62$), which can be interpreted as a tendency towards undergeneralization^{8,41}.

198 To illustrate, an estimate of $\lambda = 0.8$ corresponds to generalization of the extent that the rewards of
 199 two neighboring options are expected to be correlated by $r = 0.42$, and that this correlation decays to zero
 200 if options are further than three tiles away from each other. Additionally, we found that the estimated
 201 exploration bonus of UCB sampling ($\hat{\beta} = 0.47, t(80) = 12.78, p < .001,$
 202 $d = 1.42$; compared to the lower estimation bound), reflecting that participants valued the exploration of

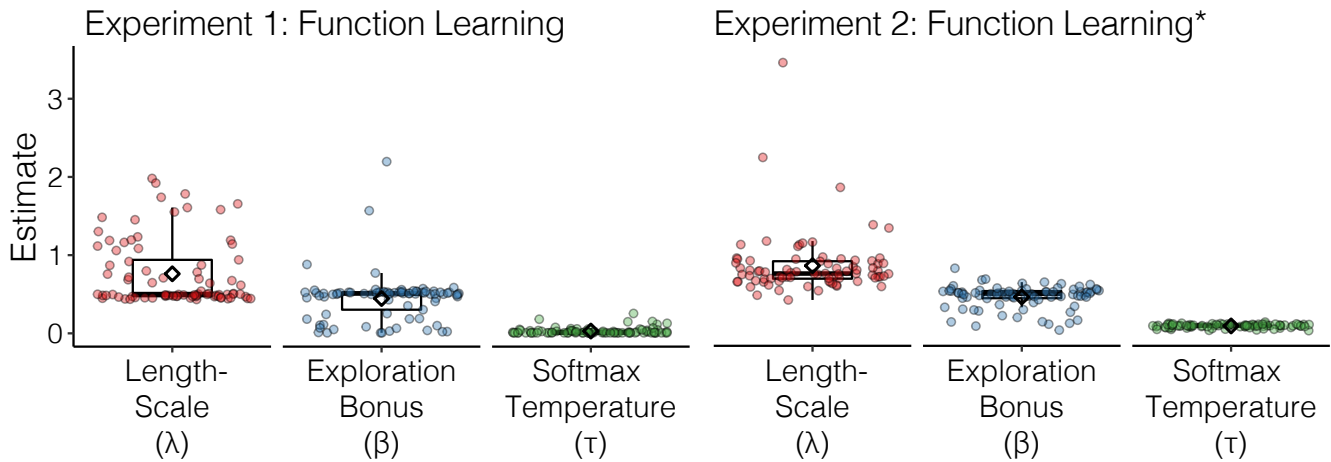


Figure 4. Parameter estimates of the best predicting model for each experiment. Each colored dot is the median estimate of a participant, with box and whisker plots indicating the upper and lower quartiles (box limits) and 1.5x IQR (whiskers), the median (horizontal line), and mean (diamond). λ is the length-scale of the RBF kernel reflecting the extent to which people generalize, β is the exploration bonus of the UCB sampling strategy, and τ is the temperature of the softmax choice rule.

203 uncertain options, along with exploiting high expectations of reward. More specifically, an exploration
204 bonus of $\beta = 0.47$ suggests that participants would prefer the hypothetical option x_1 predicted to have
205 mean reward $m(x_1) = 60$ and standard deviation $s(x_1) = 10$, over option x_2 predicted to have mean reward
206 $m(x_2) = 64$ and standard deviation $s(x_2) = 1$. This is because sampling x_1 is expected to reduce a large
207 amount of uncertainty, even though x_2 is predicted to have a slightly higher mean reward ($UCB(x_1) = 64.7$
208 vs. $UCB(x_2) = 64.47$). Lastly, we found relatively low estimates of the softmax temperature parameter
209 (mean estimate: $\hat{\tau} = 0.01$), suggesting that the search behavior of participants corresponded closely to
210 selecting the very best option, once they had taken into account both the exploitation and exploration
211 components of the available actions.

212 Experiment 2

213 In a more complex bivariate environment (Fig. 3a), the Function Learning-UCB model again made
214 better predictions than the Option Learning-UCB model ($t(79) = 9.99$, $p < .001$, $d = 1.12$), which was
215 also the case when comparing localized Function Learning*-UCB to localized Option Learning*-UCB
216 ($t(79) = 2.05$, $p = .044$, $d = 0.23$). In the two-dimensional search environment of Experiment 2, adding
217 localization improved predictions for both Option Learning-UCB ($t(79) = 19.92$, $p < .001$, $d = 2.23$) and
218 Function Learning-UCB ($t(79) = 10.47$, $p < .001$, $d = 1.17$), in line with the stronger tendency towards
219 localized sampling compared to Experiment 1 (see Fig. 1c). 61 out of 80 participants were best predicted
220 by the localized Function Learning*-UCB model, whereas only 12 participants were best described by the
221 localized Option Learning*-UCB model. Again, both components of the UCB strategy—the expected
222 reward ($t(79) = -6.44$, $p < .001$, $d = 0.72$) and the attached uncertainty ($t(79) = -14.32$, $p < .001$,
223 $d = 1.60$)—were necessary to predict choices well.

224 As in Experiment 1, simulated learning curves of the Option Learning-UCB models performed poorly
225 and were indistinguishable from a random sampling strategy, whereas both variants of the Function
226 Learning-UCB model achieved performance favorably comparable to that of human participants (Fig. 3b).
227 Median parameter estimates per participant from the Function Learning*-UCB model (Fig. 4) showed
228 that participants again underestimated the strength of the underlying spatial correlation in both Smooth

229 ($\hat{\lambda} = 0.92$, $t(42) = -14.62$, $p < .001$, $d = 2.22$; comparison to $\lambda_{Smooth} = 2$) and Rough environments
230 ($\hat{\lambda} = 0.78$, $t(36) = -5.31$, $p < .001$, $d = 0.87$; comparison to $\lambda_{Rough} = 1$), suggesting a robust tendency
231 to undergeneralize. The estimated exploration bonus β was again greater than 0 ($\hat{\beta} = 0.45$, $t(79) = 27.02$,
232 $p < .001$, $d = 3.02$, compared to the lower estimation bound). Both λ and β estimates were similar to
233 Experiment 1, while the estimated softmax temperature parameter τ was slightly larger than in Experiment
234 1 ($\hat{\tau} = 0.09$; see Table S1). Experiment 2 therefore replicated the main findings of Experiment 1.
235 Taken together, these results provide strong evidence that human search behavior is best explained by a
236 combination of function learning paired with an optimistic trade-off between exploration and exploitation.

237 The Adaptive Nature of Undergeneralization

238 In both experiments, we observed a robust tendency to undergeneralize about the spatial correlations of
239 the environment. Therefore, we ran three simulations which revealed that undergeneralization largely
240 leads to better performance than overgeneralization—and remarkably—is sometimes even better than an
241 exact match between the extent of one’s generalization and the underlying structure of the environment.
242 Our simulations consisted of generating search environments by sampling from a \mathcal{GP} prior specified
243 using a *teacher* length-scale (λ_0), and then simulating search in this environment using the Function
244 Learning-UCB model, where the \mathcal{GP} of the function learning component was specified by a *student*
245 length-scale (λ_1).

246 The first simulation assessed mismatch in the univariate setting of Experiment 1 (Fig. 5a), using the
247 median participant estimates of both the soft-max temperature parameter $\tau = 0.01$ and the exploration
248 parameter $\beta = 0.50$ and simulating 100 replications for every combination between $\lambda_0 = \{0.1, 0.2, \dots, 3\}$
249 and $\lambda_1 = \{0.1, 0.2, \dots, 3\}$. This simulation showed that it can be beneficial to undergeneralize (Fig. 5a,
250 area below the dotted line), in particular during the first five trials. Repeating the same simulations for
251 the bivariate setting of Experiment 2 (using the median participant estimates $\tau = 0.02$ and $\beta = 0.47$), we
252 found that undergeneralization can also be beneficial in a more complex two-dimensional environment
253 (Fig. 5b), at least in the early phases of learning.

254 To assess whether undergeneralization could be adaptive for Bayesian optimization algorithms in
255 a more general setting, we used a set-up regularly used by the machine learning community⁴² and ran
256 a final simulation (Fig. 5c) with continuous bivariate inputs in the range $x, y = [0, 1]$ and using every
257 combination between $\lambda_0 = \{0.1, 0.2, \dots, 1\}$ and $\lambda_1 = \{0.1, 0.2, \dots, 1\}$. Since the interpretation of λ is
258 always relative to the input range, a length-scale of $\lambda = 1$ along the unit input range would be equivalent
259 to $\lambda = 10$ in the $x, y = [0, 10]$ input range of Experiment 2. Thus, this third simulation represents a
260 broad set of potential mismatch alignments, while the use of continuous inputs extends the scope of
261 the task to an infinite state space. As before, we found that undergeneralization largely leads to better
262 performance than overgeneralization. This effect is more pronounced over time t , whereby a mismatch
263 in the direction of undergeneralization recovers over time (higher scores for larger values of t). This
264 is not the case for a mismatch in the direction of overgeneralization, which continues to produce low
265 scores, even at $t = 40$. Estimating the best possible alignment between λ_0 and λ_1 to produce the highest
266 score revealed that underestimating λ_0 by an average of about 0.21 produces the best scores over all
267 scenarios. These simulation results show that the systematically lower estimates of λ captured by our
268 models do not necessarily suggest a flaw or bias in human behavior—but instead—can sometimes lead to
269 better performance. Undergeneralization, as it turns out, might not be a bug but rather a feature of human
270 behavior.

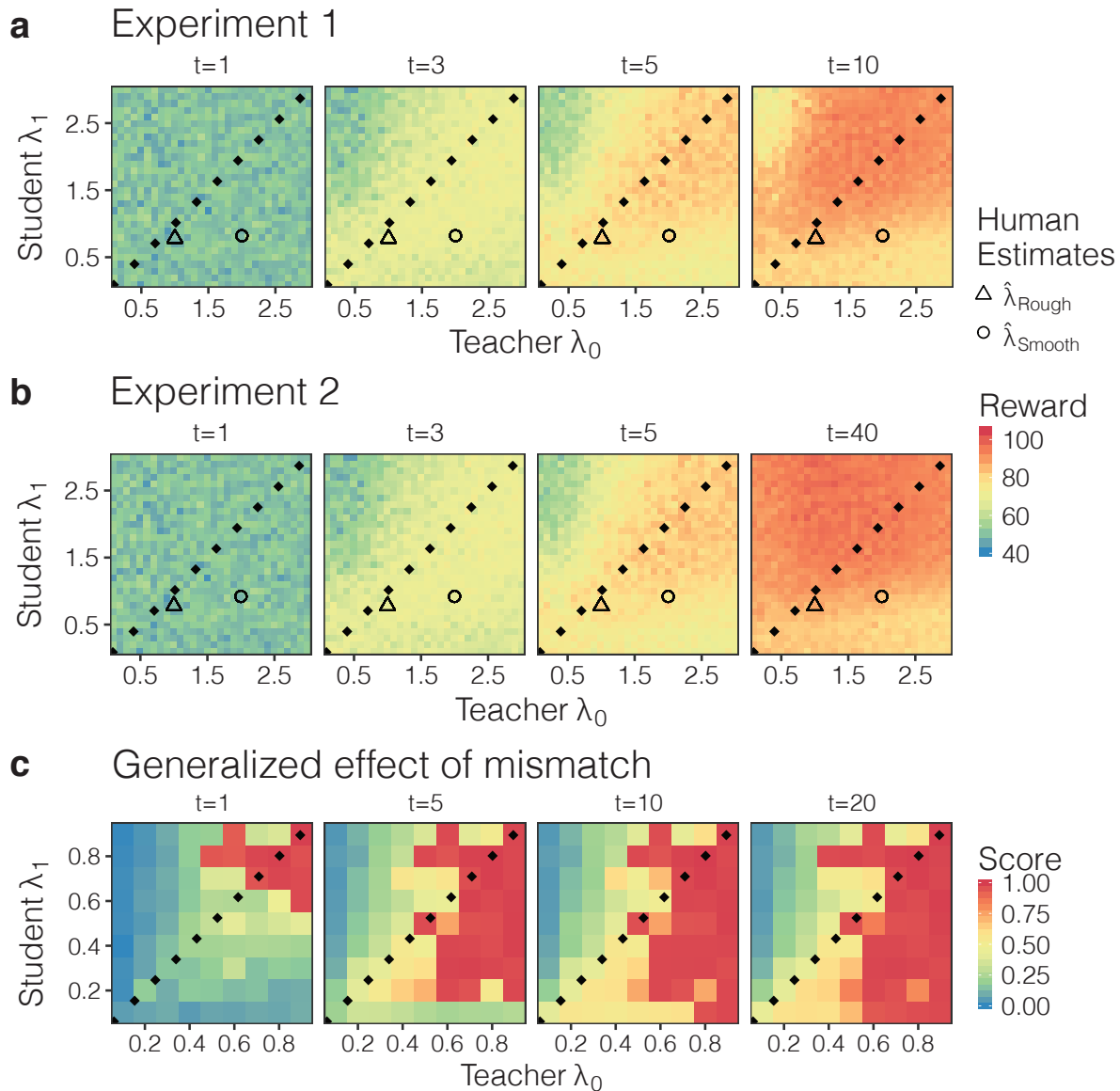


Figure 5. Mismatched length-scale (λ) simulation results. The teacher length-scale λ_0 is on the x-axis and the student length-scale λ_1 is on the y-axis. The teacher λ_0 values were used to generate environments, while the student λ_1 values were used to parameterize the Function Learning-UCB model 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. **a)** Effect of mismatch at trial numbers $t = \{1, 3, 5, 10\}$ in the univariate Experiment 1 setting, where the median participant parameters of $\tau = 0.01$ and $\beta = 0.50$ were used to specify the Function-Learning-UCB model. 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 participant λ estimates from Smooth and Rough conditions. **b)** Effect of mismatch at trial numbers $t = \{1, 3, 5, 40\}$ in the bivariate Experiment 2 setting, where the median participant parameters of $\tau = 0.02$ and $\beta = 0.47$ were used to specify the Function-Learning-UCB model. Again, each tile of the heat-map indicates the median reward obtained, with triangles and circles indicating participant λ estimates from Smooth and Rough conditions. **c)** Generalized effect of mismatch using continuous bivariate inputs in the unit range $x, y = [0, 1]$. Here the teacher λ_0 and student λ_1 range is also between $[0, 1]$. Because the value of λ is always relative to the input range, a length-scale of $\lambda = 1$ along the unit input range is equivalent to a $\lambda = 10$ in the $x, y = [0, 10]$ input range of Experiment 2. Here we report the score as a standardized measure of performance, such that 0 shows the lowest possible and 1 the highest possible log unit-score.

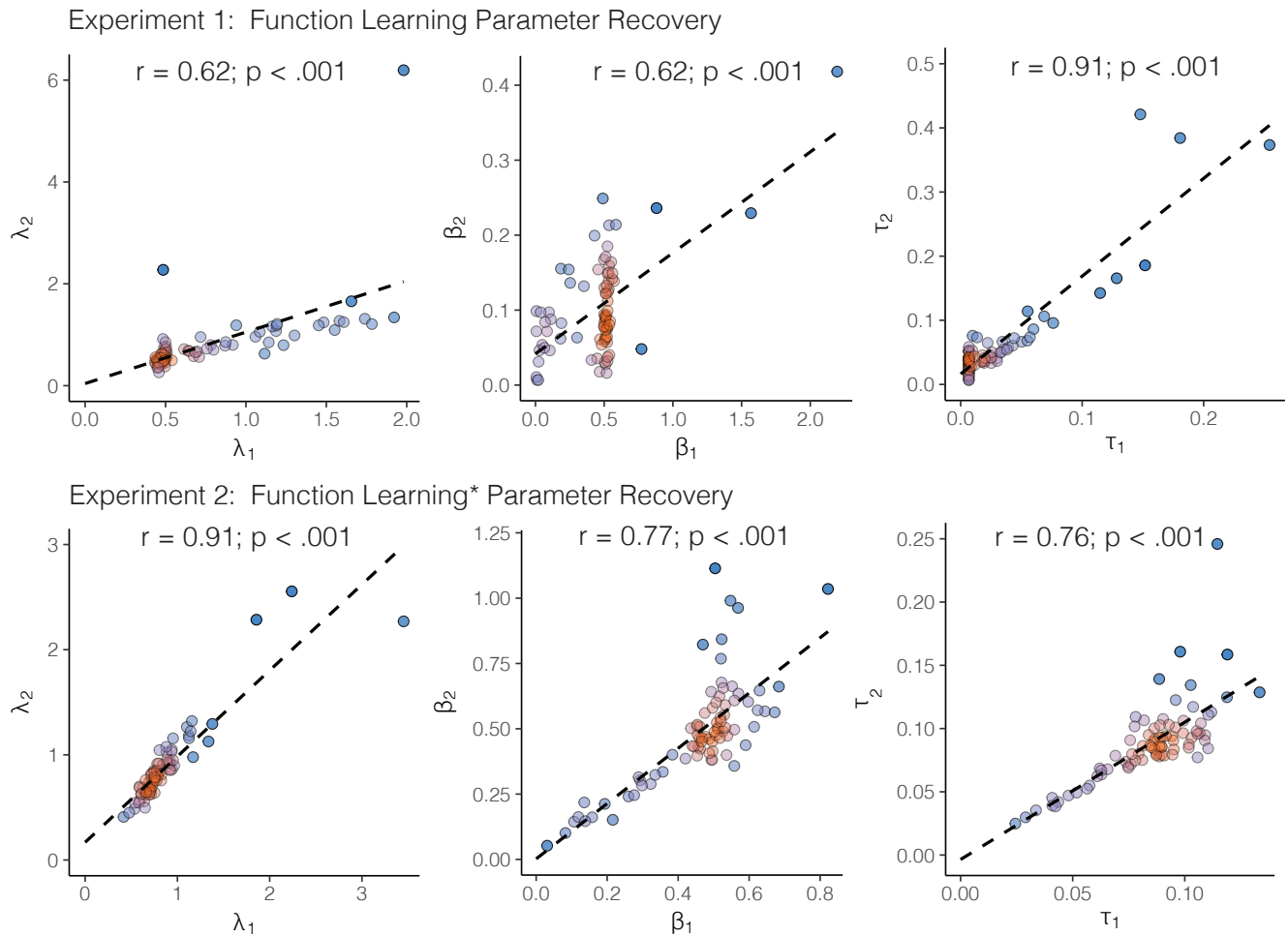


Figure 6. Parameter recovery. The generating parameter estimate is on the x-axis and the recovered parameter estimate is on the y-axis. The generating parameter estimates are from the cross-validated participant parameter estimates, which were used to simulate data. Recovered parameter estimates are 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 ($k_{Exp1} = 16; k_{Exp2} = 8$), we show the median estimate per (simulated) participant. The dashed line shows a linear regression on the data, while the Pearson correlation and p-value is shown above the plot. For readability, colors represent the bivariate kernel density estimate, with red indicating higher density.

271 Robustness and Recovery

272 To assess the validity of our modeling results, we conducted both model and parameter recovery simulations
 273 (see SI for full details). We performed model recovery by using each participant's parameter estimates
 274 to specify a *generating model* in order to simulate data, upon which we once again used the same
 275 cross-validation procedure to predict choices using a *recovering model*. In all cases, the best predictive
 276 accuracy occurred when the recovering model matched the generating model (Fig. S2), suggesting
 277 robustness to Type I errors and the unlikelihood of overfitting (i.e., the Function Learning model was not
 278 the best predictor for data generated by the Option Learning model). Additionally, the range of predictive
 279 accuracies (obtained for matching generating and recovering models) were not perfect, but rather highly
 280 similar to the predictive accuracy obtained by the Function Learning model on the participant data.

281 Parameter recovery was performed to ensure that each parameter in the Function Learning-UCB
 282 model robustly captured separate and distinct phenomena. Fig. 6 shows the results of the parameter
 283 recovery, where the generating parameter estimate (i.e., estimated from participant data) is on the x-axis,

284 and the recovered estimate is on the y-axis (i.e., estimated from generated data). In all cases, generating
285 and recovered parameter estimates were highly correlated. It is noteworthy that we found distinct and
286 recoverable estimates for β (exploration bonus) and τ (softmax temperature), because we provide evidence
287 for the existence of a *directed* exploration bonus¹¹ as a separate phenomena from random, undirected
288 exploration⁴³.

289 Discussion

290 How do people learn and adaptively make good decisions in situations where the number of possible
291 actions is vast and not all possibilities can be explored? Whereas Gaussian Process (\mathcal{GP}) function
292 learning combined with a UCB sampling algorithm has been successfully applied to search problems in
293 ecology⁴⁴, robotics⁴⁵, and biology⁴⁶, there has been little psychological research on how humans learn
294 and solve problems in environments with a rich set of possible actions. Here, we have found that Function
295 Learning, which we operationalized using \mathcal{GP} regression, provides a mechanism for generalization. The
296 ability to generalize guides participants' search behavior by forming inductive beliefs about unexplored
297 options. Combined with Upper Confidence Bound (UCB) sampling, this model navigates the exploration-
298 exploitation dilemma by optimistically inflating expectations of reward by the estimated uncertainty.

299 We have presented the first study to apply cognitive modeling to predict individual decisions in a
300 complex search task with spatially correlated outcomes. Our comparison of 27 models yielded robust
301 and recoverable results (Fig. S2) and parameter estimates (Fig. 6). The spatial correlation of rewards
302 made it possible for participants to generalize to unseen rewards by adaptively learning an underlying
303 value function based on spatial context. Our results show that participants capitalized on spatial context
304 in all task variants, and performed best in environments with the strongest spatial correlations. Even
305 though our current implementation only grazes the surface of the types of complex tasks people are able to
306 solve—and indeed could be extended in future studies using temporal dynamics or depleting resources—it
307 is nonetheless richer in both the set-up and modeling framework than traditional multi-armed bandit
308 problems used for studying human behavior.

309 Through multiple analyses, including trial-by-trial predictive cross-validation and simulated behavior
310 using participants' parameter estimates, we competitively assessed which models best predicted human
311 behavior. The vast majority of participants were best described by the Function Learning-UCB model or its
312 localized variant. Parameter estimates from the best-fitting Function Learning-UCB models suggest there
313 was a systematic tendency to undergeneralize the extent of spatial correlations, which we can sometimes
314 be a beneficial bias for search (Fig. 5).

315 Whereas previous research on exploration bonuses has had mixed results^{5,11,43}, we find robustly
316 recoverable parameter estimates for the separate phenomena of directed exploration encoded in β and
317 the random, undirected exploration encoded in the softmax temperature parameter τ , in the Function
318 Learning-UCB model. Even though UCB sampling is both *optimistic* (always treating uncertainty as
319 positive) and *myopic* (only planning the next timestep), it is nonetheless the only algorithm with currently
320 known performance guarantees in a bandit setting (i.e., sublinear regret, or in other words, monotonically
321 increasing average reward)²³. This shows a remarkable concurrence between intuitive human strategies
322 and the state of the art in machine learning.

323 Limitations and extensions

324 One potential limitation was that we failed to find reliable and interpretable differences in model parameters
325 across the different experimental manipulations (i.e., smoothness, payoff condition, and horizon length),
326 although behavioral differences were evident. In the future, starker contrasts in environment structure

327 or available search horizon may translate into more detectable differences from a modeling perspective.
328 Additionally, the goal of balancing exploration-exploitation (Accumulation condition) or the goal of
329 global optimization (Maximization condition) was induced through the manipulation of only a few lines
330 of text. While this may have been sufficient for observing behavioral differences, it may have been
331 inadequate to produce reliable differences in terms of generalization or exploration (either directed or
332 undirected), which were the (non-exhaustive) aspects of human behavior captured by our models. Indeed,
333 the practical difference between these two goals is even murky in the Bayesian optimization literature,
334 which from a purely computational perspective, often proposes abandoning the strict goal of finding the
335 global optimum⁴⁷, in favor of an approximate measure of performance, such as cumulative regret²³, which
336 more closely aligns to our Accumulation payoff condition.

337 The Function Learning-UCB model also offers many opportunities for theory integration. The Option
338 Learning model as specified here can be reformulated as special case of a \mathcal{GP} regression model⁴⁸. When
339 the length-scale of the RBF kernel approaches zero ($\lambda \rightarrow 0$), the Function Learning Model effectively
340 assumes state independence, as in the Option Learning Model. Thus, there may be a continuum of
341 reinforcement learning models, ranging from the traditional assumption of state *independence* to the
342 opposite extreme, of complete state *interdependence*. Moreover, \mathcal{GP} s also have equivalencies to Bayesian
343 Neural Networks⁴⁹, suggesting a further link to distributed function learning models⁵⁰. Indeed, one
344 explanation for the impressive performance of Deep Reinforcement Learning¹³ is that neural networks are
345 specifically a powerful type of function approximator⁵¹.

346 Lastly, recent findings have connected both spatial and conceptual representations to a common neural
347 substrate in the hippocampus²⁵, suggesting a potential avenue for applying the same Function Learning-
348 UCB model for modeling human behavior in domains such as contextual^{26,27} or semantic search^{28,29}.
349 One hypothesis for this common role of the hippocampus is that it performs predictive coding of future state
350 transitions⁵², also known as “successor representation”²². In our task, where there are no restrictions on
351 state transitions (i.e., each state is reachable from any prior state), it may be the case that the RBF kernel
352 driving our \mathcal{GP} Function Learning model performs the same role as the transition matrix of a successor
353 representation model, where state transitions are learned via a random walk policy.

354 Conclusions

355 In summary, we have introduced a new paradigm for studying how people use generalization to guide the
356 active search for rewards, found a systematic—yet sometimes beneficial—tendency to undergeneralize,
357 and uncovered strong evidence for the separate phenomena of directed exploration (towards reducing
358 uncertainty) and random, undirected exploration. Ultimately, our results help to advance our understanding
359 of adaptive behavior in complex and uncertain environments.

360 Methods

361 Participants

362 81 participants were recruited from Amazon Mechanical Turk for Experiment 1 (25 Female; mean \pm
363 SD age 33 ± 11), and 80 for Experiment 2 (25 Female; mean \pm SD age 32 ± 9). In both experiments,
364 participants were paid a participation fee of \$0.50 and a performance contingent bonus of up to \$1.50.
365 Participants earned on average $\$1.14 \pm 0.13$ and spent 8 ± 4 minutes on the task in Experiment 1,
366 while participants earned on average $\$1.64 \pm 0.20$ and spent 8 ± 4 minutes on the task in Experiment
367 2. Participants were only allowed to participate in one of the experiments, and were required to have a
368 95% HIT approval rate and 1000 previously completed HITs. The Ethics Committee of the Max Planck
369 Institute for Human Development approved the methodology and all participants consented to participation

370 through an online consent form at the beginning of the survey.

371 **Design**

372 Both experiments used a 2×2 between-subjects design, where participants were randomly assigned to
373 one of two different payoff structures (*Accumulation condition* vs. *Maximization condition*) and one of
374 two different classes of environments (*Smooth* vs. *Rough*). Each grid world represented a (either uni- or
375 bivariate) function, with each observation including normally distributed noise, $\varepsilon \sim \mathcal{N}(0, 1)$. The task
376 was presented over either 16 rounds (Exp. 1) or 8 rounds (Exp. 2) on different grid worlds drawn from the
377 same class of environments. Participants had either a short or long search horizon (Exp. 1: [5, 10]; Exp. 2:
378 [20, 40]) to sample tiles on the grid, including repeat clicks. The search horizon alternated between rounds
379 (within subject), with initial horizon length counterbalanced between subjects.

380 **Materials and procedure**

381 Participants observed four fully revealed example environments and had to correctly complete three
382 comprehension questions, prior to starting the task. Example environments were drawn from the same
383 class of environments assigned to the participant (Smooth or Rough). At the beginning of each round, one
384 random tile was revealed and participants could click any of the tiles in the grid until the search horizon
385 was exhausted, including re-clicking previously revealed tiles. Clicking an unrevealed tile displayed the
386 numerical value of the reward along with a corresponding color aid, where darker colors indicated higher
387 point values. Per round, observations were scaled to a randomly drawn maximum value in the range of 65
388 to 85, so that the value of the global optima could not be easily guessed (e.g., a value of 100). Re-clicked
389 tiles could show some variations in the observed value due to noise. For repeat clicks, the most recent
390 observation was displayed numerically, while hovering over the tile would display the entire history of
391 observation. The color of the tile corresponded to the mean of all previous observations.

392 **Payoff conditions.** We compared performance under two different payoff conditions, requiring either a
393 balance between exploration and exploitation (*Accumulation condition*) or corresponding to consistently
394 making exploration decisions (*Maximization condition*). In each payoff condition, participants received a
395 performance contingent bonus of up to \$1.50. *Accumulation condition* were given a bonus based on the
396 average value of all clicks as a fraction of the global optima, $\frac{1}{T} \sum (\frac{y_t}{y^*})$, where y^* is the global optimum,
397 whereas participants in the *Maximization condition* were rewarded using the ratio of the highest observed
398 reward to the global optimum, $(\frac{\max y_t}{y^*})^4$, taken to the power of 4 to exaggerate differences in the upper
399 range of performance and for between-group parity in expected earnings across payoff conditions. Both
400 conditions were equally weighted across all rounds and used noisy but unscaled observations to assign a
401 bonus of up to \$1.50. Subjects were informed in dollars about the bonus earned at the end of each round.

402 **Smoothness of the environment.** We used two classes of environments, corresponding to different levels
403 of smoothness. All environments were sampled from a \mathcal{GP} prior with a RBF kernel, where the length-scale
404 parameter (λ) determines the rate at which the correlations of rewards decay over distance. *Rough*
405 environments used $\lambda_{Rough} = 1$ and *Smooth* environments used $\lambda_{Smooth} = 2$, with 40 environments (Exp. 1)
406 and 20 environments (Exp. 2) generated for each class (Smooth and Rough). Both example environments
407 and task environments were drawn without replacement from the assigned class of environments, where
408 smoothness can be understood as the extent of spatial correlations.

409 **Search horizons.** We chose two horizon lengths (Short=5 or 20 and Long=10 or 40) that were fewer
410 than the total number of tiles on the grid (30 or 121), and varied them within subject (alternating between
411 rounds and counterbalanced). Horizon length was approximately equivalent between Experiments 1 and 2
412 as a fraction of the total number of options (short $\approx \frac{1}{6}$; long $\approx \frac{1}{3}$).

413 Models of Learning

414 We use different *Models of Learning* (i.e., Function Learning and Option Learning), which combined
 415 with a *Sampling Strategy* can make predictions about where a participant will search, given the history of
 416 previous observations.

417 Function Learning

418 The Function Learning Model adaptively learns an underlying function mapping spatial locations onto
 419 rewards. We use Gaussian Process (\mathcal{GP}) regression as a Bayesian method of function learning³⁶. A \mathcal{GP} is
 420 defined as a collection of points, any subset of which is multivariate Gaussian. Let $f : \mathcal{X} \rightarrow \mathbb{R}^n$ denote a
 421 function over input space \mathcal{X} that maps to real-valued scalar outputs. This function can be modeled as a
 422 random draw from a \mathcal{GP} :

$$423 \quad f \sim \mathcal{GP}(m, k), \quad (5)$$

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

$$427 \quad m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \quad (6)$$

$$428 \quad k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \quad (7)$$

430 Here, we fix the prior mean to the median value of payoffs, $m(\mathbf{x}) = 50$ and use the kernel function
 431 to encode an inductive bias about the expected spatial correlations between rewards (see Radial Basis
 432 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
 433 the underlying function with added noise $\sigma^2 = 1$, we can calculate the posterior predictive distribution for
 434 a new input \mathbf{x}_* as a Gaussian with mean $m_t(\mathbf{x}_*)$ and variance $v_t(\mathbf{x}_*)$ given by:

$$435 \quad \mathbb{E}[f(\mathbf{x}_*)|\mathcal{D}_t] = m_t(\mathbf{x}_*) = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t \quad (8)$$

$$436 \quad \mathbb{V}[f(\mathbf{x}_*)|\mathcal{D}_t] = v_t(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*, \quad (9)$$

438 where $\mathbf{y} = [y_1, \dots, y_t]^\top$, \mathbf{K} is the $t \times t$ covariance matrix evaluated at each pair of observed inputs, and
 439 $\mathbf{k}_* = [k(\mathbf{x}_1, \mathbf{x}_*), \dots, k(\mathbf{x}_t, \mathbf{x}_*)]$ is the covariance between each observed input and the new input \mathbf{x}_* .

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

$$442 \quad k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\lambda}\right) \quad (10)$$

444 This kernel defines a universal function learning engine based on the principles of Bayesian regression and
 445 can model any stationary function[‡]. Intuitively, the RBF kernel models the correlation between points as
 446 an exponentially decreasing function of their distance. Here, λ modifies the rate of correlation decay, with
 447 larger λ -values corresponding to slower decays, stronger spatial correlations, and smoother functions. As
 448 $\lambda \rightarrow \infty$, the RBF kernel assumes functions approaching linearity, whereas as $\lambda \rightarrow 0$, there ceases to be any
 449 spatial correlation, with the implication that learning happens independently for each discrete input without
 450 generalization (similar to traditional models of associative learning). We treat λ as a hyper-parameter, and
 451 use cross-validated estimates to make inferences about the extent to which participants generalize.

[‡]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.

452 Option Learning

453 The Option Learning model uses a Bayesian Mean Tracker (BMT), which is a type of traditional associative
 454 learning model that assumes the average reward associated with each option is constant over time (i.e., no
 455 temporal dynamics, as opposed to the assumptions of a Kalman filter or Temporal Difference Learning)⁵,
 456 as is the case in our experimental search tasks. In contrast to the Function Learning model, the Option
 457 Learning model learns the rewards of each option independently, by computing an independent posterior
 458 distribution for the mean μ_j for each option j . We implement a version that assumes rewards are normally
 459 distributed (as in the \mathcal{GP} Function Learning Model), with a known variance but unknown mean, where the
 460 prior distribution of the mean is again a normal distribution. This implies that the posterior distribution for
 461 each mean is also a normal distribution:

$$462 \quad p(\mu_{j,t} | \mathcal{D}_{t-1}) = \mathcal{N}(m_{j,t}, v_{j,t}) \quad (11)$$

463
 464 For a given option j , the posterior mean $m_{j,t}$ and variance $v_{j,t}$ are only updated when it has been selected
 465 at trial t :

$$466 \quad m_{j,t} = m_{j,t-1} + \delta_{j,t} G_{j,t} [y_t - m_{j,t-1}] \quad (12)$$

$$467 \quad v_{j,t} = [1 - \delta_{j,t} G_{j,t}] v_{j,t-1} \quad (13)$$

468
 469 where $\delta_{j,t} = 1$ if option j was chosen on trial t , and 0 otherwise. Additionally, y_t is the observed reward at
 470 trial t , and $G_{j,t}$ is defined as:

$$471 \quad G_{j,t} = \frac{v_{j,t-1}}{v_{j,t-1} + \theta_{\epsilon}^2} \quad (14)$$

472
 473 where θ_{ϵ}^2 is the error variance, which is estimated as a free parameter. Intuitively, the estimated mean
 474 of the chosen option $m_{j,t}$ is updated based on the difference between the observed value y_t and the prior
 475 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
 476 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
 477 sensitivity, where smaller values result in more substantial updates to the mean $m_{j,t}$, and larger reductions
 478 of uncertainty $v_{j,t}$. We set the prior mean to the median value of payoffs $m_{j,0} = 50$ and the prior variance
 479 $v_{j,0} = 500$.

480 Sampling Strategies

481
 482 Given the normally distributed posteriors of the expected rewards, which have mean $m_t(\mathbf{x})$ and variance
 483 $v_t(\mathbf{x})$, 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
 484 j is the index of the option characterized by \mathbf{x}), we assess different sampling strategies that (with a softmax
 485 choice rule) make probabilistic predictions about where participants search next at time $t + 1$.

486 **Upper Confidence Bound Sampling.** Given the posterior predictive mean $m_t(\mathbf{x})$ and the estimated
 487 uncertainty (estimated here as a standard deviation) $s_t(\mathbf{x}) = \sqrt{v_t(\mathbf{x})}$, we calculate the upper confidence
 488 bound using a simple sum

$$489 \quad \text{UCB}(\mathbf{x}) = m_t(\mathbf{x}) + \beta s_t(\mathbf{x}), \quad (15)$$

490
 491 where the exploration factor β determines how much reduction of uncertainty is valued (relative to
 492 exploiting known high-value options) and is estimated as a free parameter.

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

$$496 \text{PureExploit}(\mathbf{x}) = m_t(\mathbf{x}) \quad (16)$$

$$497 \text{PureExplore}(\mathbf{x}) = s_t(\mathbf{x}) \quad (17)$$

499 Localization of Models

500 To penalize search options by the distance from the previous choice, we weighted each option by the
501 inverse Manhattan distance (IMD) to the last revealed tile $\text{IMD}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n |x_i - x'_i|$, prior to the softmax
502 transformation. For the special case where $\mathbf{x} = \mathbf{x}'$, we set $\text{IMD}(\mathbf{x}, \mathbf{x}') = 1$. Localized models are indicated
503 by an asterix (*).

504 Model Comparison

505 We use maximum likelihood estimation (MLE) for parameter estimation, and cross-validation to measure
506 out-of-sample predictive accuracy. A softmax choice rule transforms each model's prediction into a
507 probability distribution over options:

$$508 p(\mathbf{x}) = \frac{\exp(q(\mathbf{x})/\tau)}{\sum_{j=1}^N \exp(q(\mathbf{x}_j)/\tau)}, \quad (18)$$

510 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
511 model), and τ is the temperature parameter. Lower values of τ indicate more concentrated probability
512 distributions, corresponding to more precise predictions. All models include τ as a free parameter.
513 Additionally, Function Learning models estimate λ (length-scale), Option Learning models estimate θ_ϵ^2
514 (error variance), and Upper Confidence Bound sampling models estimate β (exploration bonus).

515 **Cross Validation.** We fit all models—per participant—using cross-validated MLE, with either a Differ-
516 ential Evolution algorithm⁵³ or a grid search if the model contained only a single parameter. Parameter
517 estimates are constrained to positive values in the range $[\exp(-5), \exp(5)]$. Cross-validation is performed
518 by first separating participant data according to horizon length, which alternated between rounds within
519 subject. For each participant, half of the rounds corresponded to a short horizon and the other half corre-
520 sponded to a long horizon. Within all rounds of each horizon length, we use leave-one-out cross-validation
521 to iteratively form a training set by leaving out a single round, computing a MLE on the training set, and
522 then generating out of sample predictions on the remaining round. This is repeated for all combinations
523 of training set and test set, and for both short and long horizon sets. The cross-validation procedure
524 yielded one set of parameter estimates per round, per participant, and out-of-sample predictions for 120
525 choices in Experiment 1 and 240 choices in Experiment 2 (per participant). In total, cross-validated model
526 comparisons for both experiments required approximately 50,000 hours of computation, or about 3 days
527 distributed across a 716 CPU cluster.

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

$$531 R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}, \quad (19)$$

532

533 where $\log \mathcal{L}(\mathcal{M}_{\text{rand}})$ is the log loss of a random model (i.e., picking options with equal probability) and
534 $\log \mathcal{L}(\mathcal{M}_k)$ is the log loss of model k 's out-of-sample prediction error. Intuitively, $R^2 = 0$ corresponds
535 to prediction accuracy equivalent to chance, while $R^2 = 1$ corresponds to theoretical perfect prediction
536 accuracy, since $\log \mathcal{L}(\mathcal{M}_k) / \log \mathcal{L}(\mathcal{M}_{\text{rand}}) \rightarrow 0$ when $\log \mathcal{L}(\mathcal{M}_k) \ll \log \mathcal{L}(\mathcal{M}_{\text{rand}})$.

537 Mismatched generalization

538 We assessed the effect of mismatched λ -estimates on the performance of the Function Learning-UCB
539 Model. A mismatch is defined as estimating a different level of spatial correlations (captured by the
540 per participant λ -estimates) than the ground truth in the environment ($\lambda_{\text{Smooth}} = 2$, and $\lambda_{\text{Rough}} = 1$ for
541 both experiments). In both experiments, we found that participant λ -estimates were systematically
542 lower than the true value (Fig. 3), which can be interpreted as a tendency to undergeneralize about the
543 spatial correlation of rewards in the world. In order to test how this tendency to undergeneralize (i.e.,
544 underestimate λ) influences task performance, we present results of 3 simulations (Fig. 5) using different
545 λ values in a *teacher* kernel (x-axis) and a *student* kernel (y-axis).

546 Both teacher and student kernels were always RBF kernels, where the teacher kernel was parameterized
547 with a length-scale λ_0 and the student kernel with a length-scale λ_1 . For situations in which $\lambda_0 \neq \lambda_1$, the
548 assumptions of the student can be seen as mismatched with the environment. The student *overgeneralizes*
549 when $\lambda_1 > \lambda_0$ (Fig. 5 above the dotted line), and *undergeneralizes* when $\lambda_1 < \lambda_0$ (Fig. 5 below the dotted
550 line), as was captured by in our behavioral data. For the two empirical simulations, we simulate every
551 possible combination between $\lambda_0 = \{0.1, 0.2, \dots, 3\}$ and $\lambda_1 = \{0.1, 0.2, \dots, 3\}$, leading to 900 different
552 combinations of student-teacher scenarios. For each of these combinations, we sample a target function
553 from a \mathcal{GP} parameterized by λ_0 and then use the Function Learning-UCB Model parameterized by λ_1 to
554 search for rewards using the median parameter estimates from within the matching experiment.

555 For the generalized Bayesian optimization simulation, we simulate every possible combination between
556 $\lambda_0 = \{0.1, 0.2, \dots, 1\}$ and $\lambda_1 = \{0.1, 0.2, \dots, 1\}$, leading to 100 different combinations of student-teacher
557 scenarios. For each of these combinations, we sample a continuous bivariate target function from a \mathcal{GP}
558 parameterized by λ_0 and then use the Function Learning-UCB Model parameterized by λ_1 to search for
559 rewards. The exploration parameter β was set to 0.5 to resemble participant behavior (Table S1). The input
560 space was continuous between 0 and 1, i.e. any number between 0 and 1 could be chosen and GP-UCB
561 was optimized (sometimes called the inner-optimization loop) per step using NLOPT⁵⁴ for non-linear
562 optimization.

563 Figure 5 shows the results of each simulation. For Experiment 1 and Experiment 2 simulations (Fig.
564 5a-b), the color of each tile shows the median reward obtained at the indicated trial number, for each of the
565 100 replications using the specified teacher-student scenario. For the generalized Bayesian optimization
566 simulation, 5c), we report score as a standardized measure of performance using the log-unit scores,
567 normalized to a range of $[0, 1]$.

568 Supporting Information

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

576 **S2 Fig. Model recovery results.** Data was generated by the specified generating model using individual
577 participant parameter estimates. The recovery process used the same cross-validation method used in the
578 model comparison. We report the predictive accuracy of each candidate recovery model. Bars show the
579 group mean with standard error, with each individual (simulated) participant overlaid as a dot. Icon arrays
580 show the number of simulated participants best described. For both generating and recovery models, we
581 used UCB sampling. Table S1 reports the median values of the cross-validated parameter estimates used
582 to specify each generating model.

583 **S3 Fig. Learning over trials and rounds.** Average correlational effect size of trial and round on score
584 per participant as assessed by a standardized linear regression. Participants are ordered by effect size
585 in decreasing order. Dashed lines indicate no effect. Red lines indicate average effect size. Whereas
586 participants consistently improve over trials, there is no effect over rounds.

587 **S1 Table. Modeling results.** Full description of predictive accuracy, number of participants best
588 described, and parameter estimates for all 27 models.

589 Acknowledgments

590 We thank Peter Todd, Tim Pleskac, Neil Bramley, Henrik Singmann, and Mehdi Moussaïd for helpful
591 feedback. This work was supported by the International Max Planck Research School on Adapting
592 Behavior in a Fundamentally Uncertain World (CMW), and DFG grants ME 3717/2-2 to BM and NE
593 1713/1-2 to JDN as part of the New Frameworks of Rationality (SPP 1516) priority program. ES was
594 supported by the Harvard Data Science Initiative. Code and data available at <https://github.com/charleywu/gridsearch>
595

References

1. Romero, P. A., Krause, A. & Arnold, F. H. Navigating the protein fitness landscape with Gaussian Processes. *Proc. Natl. Acad. Sci.* **110**, E193–E201 (2013).
2. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nat.* **529**, 484–489 (2016).
3. Hills, T. T. *et al.* Exploration versus exploitation in space, mind, and society. *Trends Cogn. Sci.* **19**, 46–54 (2015).
4. Steyvers, M., Lee, M. D. & Wagenmakers, E.-J. A Bayesian analysis of human decision-making on bandit problems. *J. Math. Psychol.* **53**, 168–179 (2009).
5. Speekenbrink, M. & Konstantinidis, E. Uncertainty and exploration in a restless bandit problem. *Top. Cogn. Sci.* **7**, 351–367 (2015).
6. Palminteri, S., Lefebvre, G., Kilford, E. J. & Blakemore, S.-J. Confirmation bias in human reinforcement learning: Evidence from counterfactual feedback processing. *PLoS computational biology* **13**, e1005684 (2017).
7. Lee, S. W., Shimojo, S. & O’Doherty, J. P. Neural computations underlying arbitration between model-based and model-free learning. *Neuron* **81**, 687–699 (2014).
8. Gershman, S. J. & Daw, N. D. Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annu. Rev. Psychol.* **68**, 101–128 (2017).
9. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction*, vol. 1 (MIT press Cambridge, 1998).

10. Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. Building machines that learn and think like people. *Behav. Brain Sci.* **40** (2017). URL <http://arxiv.org/abs/1604.00289>. DOI <https://doi.org/10.1017/S0140525X16001837>. 1604.00289.
11. 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. *J. Exp. Psychol. Gen.* **143**, 2074 (2014).
12. Tesauro, G. Practical issues in temporal difference learning. *Mach. learning* **8**, 257–277 (1992).
13. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nat.* **518**, 529–533 (2015).
14. Huys, Q. J. *et al.* Interplay of approximate planning strategies. *Proc. Natl. Acad. Sci.* **112**, 3098–3103 (2015).
15. Solway, A. & Botvinick, M. M. Evidence integration in model-based tree search. *Proc. Natl. Acad. Sci.* **112**, 11708–11713 (2015).
16. Guez, A., Silver, D. & Dayan, P. Scalable and efficient Bayes-adaptive reinforcement learning based on monte-carlo tree search. *J. Artif. Intell. Res.* **48**, 841–883 (2013).
17. Rasmussen, C. E. & Kuss, M. Gaussian processes in reinforcement learning. In *Advances in Neural Information Processing Systems*, vol. 4, 1 (2003).
18. Sutton, R. S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Adv. Neural Inf. Process. Syst.* 1038–1044 (1996).
19. Lucas, C. G., Griffiths, T. L., Williams, J. J. & Kalish, M. L. A rational model of function learning. *Psychon. Bull. & Rev.* **22**, 1193–1215 (2015).
20. Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M. & Gershman, S. J. Compositional inductive biases in function learning. *Cogn. Psychol.* **99**, 44 – 79 (2017).
21. Borji, A. & Itti, L. Bayesian optimization explains human active search. In *Advances in Neural Information Processing Systems*, 55–63 (2013).
22. Dayan, P. & Niv, Y. Reinforcement learning: The good, the bad and the ugly. *Curr. opinion neurobiology* **18**, 185–196 (2008).
23. Srinivas, N., Krause, A., Kakade, S. & Seeger, M. W. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, 1015–1022 (2010).
24. Wilke, A. *et al.* A game of hide and seek: Expectations of clumpy resources influence hiding and searching patterns. *PloS one* **10**, e0130976 (2015).
25. Constantinescu, A. O., O’Reilly, J. X. & Behrens, T. E. Organizing conceptual knowledge in humans with a gridlike code. *Sci.* **352**, 1464–1468 (2016).
26. Stojic, H., Analytis, P. P. & Speekenbrink, M. Human behavior in contextual multi-armed bandit problems. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, 2290–2295 (Cognitive Science Society, 2015).
27. Schulz, E., Konstantinidis, E. & Speekenbrink, M. Putting bandits into context: How function learning supports decision making. *J. Exp. Psychol. Learn. Mem. Cogn.* (2017). DOI <http://dx.doi.org/10.1037/xlm0000463>.
28. Hills, T. T., Jones, M. N. & Todd, P. M. Optimal foraging in semantic memory. *Psychol. review* **119**, 431 (2012).

29. Abbott, J. T., Austerweil, J. L. & Griffiths, T. L. Random walks on semantic networks can resemble optimal foraging. *Psychol. Rev.* **122**, 558–569 (2015).
30. Schulz, E., Tenenbaum, J. B., Reshef, D. N., Speekenbrink, M. & Gershman, S. Assessing the perceived predictability of functions. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, 2116–2121 (Cognitive Science Society, 2015).
31. Bramley, N. R., Dayan, P., Griffiths, T. L. & Lagnado, D. A. Formalizing neurath’s ship: Approximate algorithms for online causal learning. *Psychol. Rev.* **124**, 301 (2017).
32. Lindley, D. V. On a measure of the information provided by an experiment. *The Annals Math. Stat.* 986–1005 (1956).
33. Nelson, J. D. Finding useful questions: On bayesian diagnosticity, probability, impact, and information gain. *Psychol. Rev.* **112** (2005).
34. Crupi, V. & Tentori, K. State of the field: Measuring information and confirmation. *Stud. Hist. Philos. Sci. Part A* **47**, 81–90 (2014).
35. Rasmussen, C. & Williams, C. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning (MIT Press, 2006).
36. Schulz, E., Speekenbrink, M. & Krause, A. A tutorial on gaussian process regression with a focus on exploration-exploitation scenarios. *bioRxiv* (2017). URL <https://www.biorxiv.org/content/early/2017/02/21/095190>. DOI 10.1101/095190. <https://www.biorxiv.org/content/early/2017/02/21/095190.full.pdf>.
37. Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.* **3**, 397–422 (2002).
38. Le Roux, N. & Bengio, Y. Deep belief networks are compact universal approximators. *Neural computation* **22**, 2192–2207 (2010).
39. Shepard, R. N. Toward a universal law of generalization for psychological science. *Sci.* **237**, 1317–1323 (1987).
40. Kaufmann, E., Cappé, O. & Garivier, A. On Bayesian upper confidence bounds for bandit problems. In *Artificial Intelligence and Statistics*, 592–600 (2012).
41. Schulz, E., Speekenbrink, M., Hernández-Lobato, J. M., Ghahramani, Z. & Gershman, S. J. Quantifying mismatch in Bayesian optimization. In *NIPS Workshop on Bayesian Optimization: Black-box Optimization and beyond* (2016).
42. Metzen, J. H. Minimum regret search for single-and multi-task optimization. *arXiv preprint arXiv:1602.01064* (2016).
43. Daw, N. D., O’doherly, J. P., Dayan, P., Seymour, B. & Dolan, R. J. Cortical substrates for exploratory decisions in humans. *Nat.* **441**, 876–879 (2006).
44. Gotovos, A., Casati, N., Hitz, G. & Krause, A. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1344–1350 (2013).
45. Cully, A., Clune, J., Tarapore, D. & Mouret, J.-B. Robots that can adapt like animals. *Nat.* **521**, 503–507 (2015).
46. Sui, Y., Gotovos, A., Burdick, J. & Krause, A. Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning*, 997–1005 (2015).

47. Mockus, J. *Bayesian approach to global optimization: Theory and applications*, vol. 37 (Springer Science & Business Media, 2012).
48. Reece, S. & Roberts, S. An introduction to Gaussian processes for the Kalman filter expert. In *13th Conference on Information Fusion (FUSION)*, 1–9 (IEEE, 2010).
49. Neal, R. M. *Bayesian learning for neural networks* (Springer Science & Business Media, 1996).
50. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nat.* **521**, 436–444 (2015).
51. Schölkopf, B. Artificial intelligence: Learning to see and act. *Nat.* **518**, 486–487 (2015).
52. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nat. Neurosci.* **20**, 1643 EP – (2017). URL <http://dx.doi.org/10.1038/nn.4650>. Article.
53. Mullen, K., Ardia, D., Gil, D., Windover, D. & Cline, J. DEoptim: An R package for global optimization by differential evolution. *J. Stat. Softw.* **40**, 1–26 (2011). URL <http://www.jstatsoft.org/v40/i06/>.
54. Johnson, S. G. The nlopt nonlinear-optimization package (2014).
55. Bonawitz, E., Denison, S., Gopnik, A. & Griffiths, T. L. Win-stay, lose-sample: A simple sequential algorithm for approximating bayesian inference. *Cogn. Psychol.* **74**, 35–65 (2014).

Supporting information

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*, which make predictions about search decisions without maintaining a representation of the world (i.e., with no learning model). Table S1 shows the predictive accuracy, the number of participants best described, and the median parameter estimates of each model. Figure S1 shows a more detailed assessment of predictive accuracy, with participants separated by payoff condition and environment type.

Additional Sampling Strategies

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}^+ :

$$\text{EXI}(\mathbf{x}) = \begin{cases} \Phi(Z)(m_t(\mathbf{x}) - m_t(\mathbf{x}^+)) + s_t(\mathbf{x})\phi(Z), & \text{if } s_t(\mathbf{x}) > 0 \\ 0, & \text{if } s_t(\mathbf{x}) = 0 \end{cases} \quad (20)$$

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:

$$\begin{aligned} \text{POI}(\mathbf{x}) &= P(f(\mathbf{x}) \geq f(\mathbf{x}^+)) \\ &= \Phi\left(\frac{m_t(\mathbf{x}) - m_t(\mathbf{x}^+)}{s_t(\mathbf{x})}\right) \end{aligned} \quad (21)$$

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 context⁵. It is a form of probability matching and can be implemented by sampling from each option's predictive distribution once, and then choosing the option with the highest sampled pay-off.

$$\text{PMU}(\mathbf{x}) = P(f(\mathbf{x}_j) > f(\mathbf{x}_{i \neq j})) \quad (22)$$

We implement this acquisition function 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 behavior without learning about the distribution of rewards.

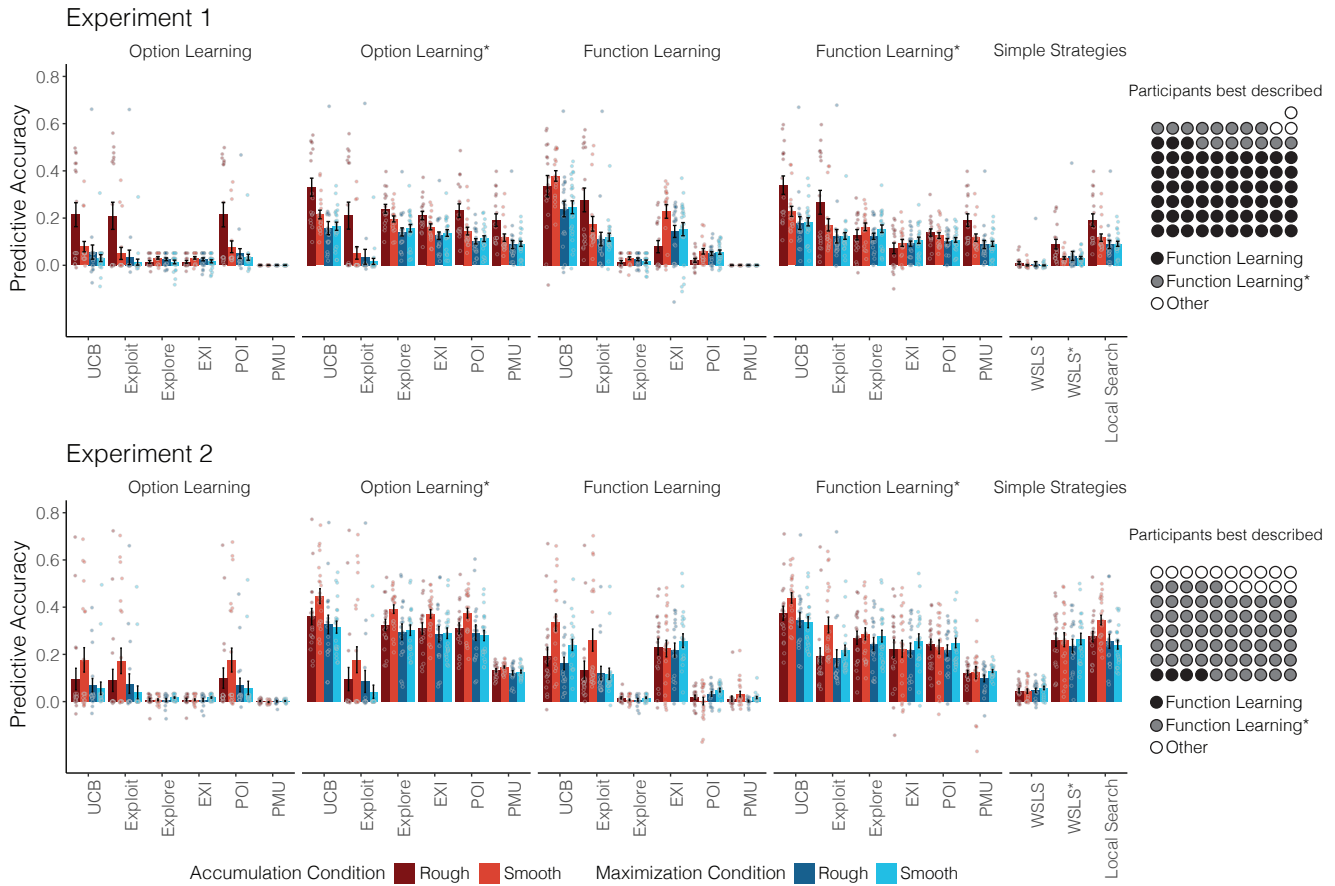


Figure S1. Full model comparison of all 27 models. The learning model is indicated above (or lack of in the case of simple heuristic strategies), and sampling strategy are along the x-axis. Bars indicate predictive accuracy (group mean) along with standard error, and are separated by payoff condition (color) 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 S1 provides more detail about the number of participants best described by each model.

Win-Stay Lose-Sample

We consider a form of a win-stay lose-sample (WSLS) heuristic⁵⁵, where a *win* is defined as finding a payoff with a higher or equal value than the previous best. 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:

$$\text{IMD}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n |x_i - x'_i|, \quad (23)$$

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

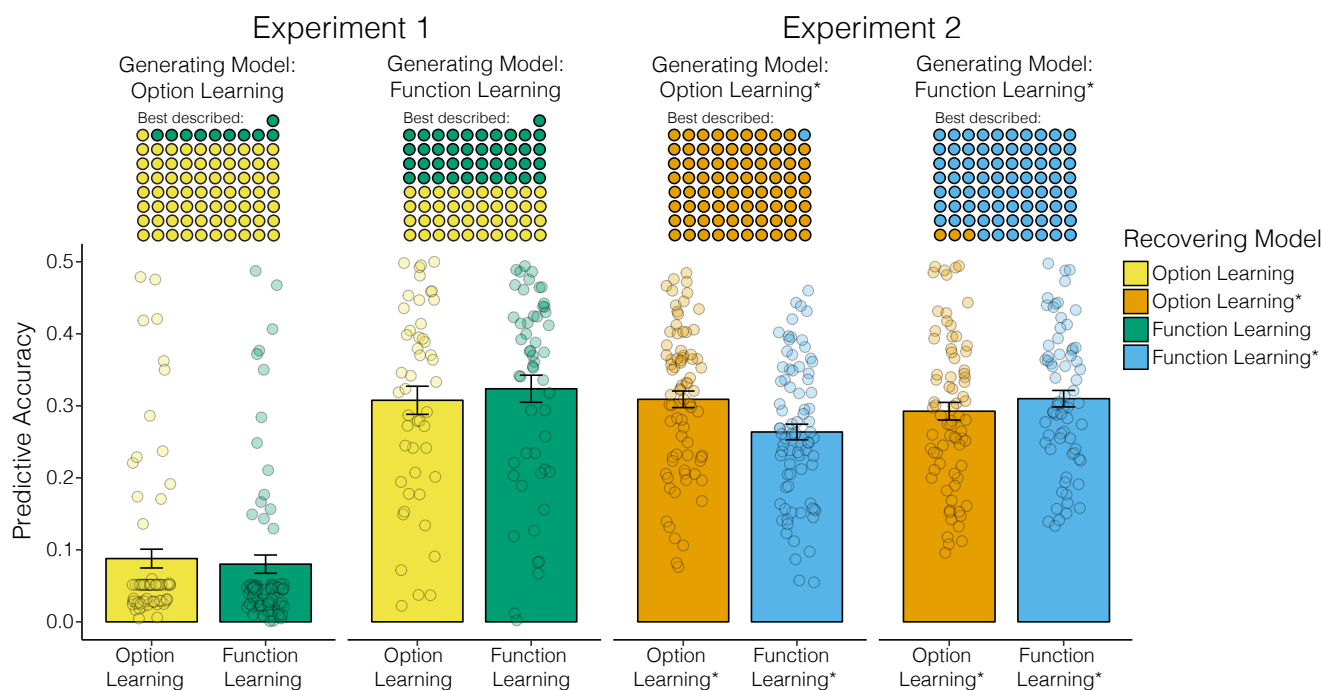


Figure S2. Model recovery results. Data was generated by the specified generating model using individual participant parameter estimates. The recovery process used the same cross-validation method used in the model comparison. We report the predictive accuracy of each candidate recovery model. Bars show the group mean with standard error, with each individual (simulated) participant overlaid as a dot. Icon arrays show the number of simulated participants best described. For both generating and recovery models, we used UCB sampling. Table S1 reports the median values of the cross-validated parameter estimates used to specify each generating model.

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, without the introduction of any additional free parameters. Localized models are indicated with an asterisk (e.g., Function Learning*).

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. More specifically, for each participant and round, we use the cross-validated parameter estimates to specify a given model, and then generate new data resembling participant data. We generate data using the Option Learning and the Function Learning model for Experiment 1 and the Option Learning* model and the Function Learning* model for Experiment 2. In all cases, we use the UCB sampling strategy in conjunction with the specified learning model. We then utilize the same cross-validation method as before in order to determine if we can successfully identify which model has generated the underlying data. Figure S2 shows the cross-validated predictive performance (bars) for the simulated data, along with the number of simulated participants best described (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 on participant data is unlikely due to overfitting.

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.

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. This makes our finding of the Function Learning as the best predictive model even stronger as—technically—the Option Learning model could mimic parts of the Function Learning behavior.

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 predictive accuracy of human participants (Table S1). 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 Option Learning* model generated the data, the Option Learning* model achieves a predictive accuracy of $R^2 = .32$ and predicts 79 out of 80 simulated participants best, whereas the Function Learning* model predicts only a lone simulated participant better, with an average predictive accuracy of $R^2 = .26$.

If the Function Learning* model generated 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 only describes 3 out of 80 simulated participants better, with a average predictive accuracy of $R^2 = .32$.

In all of the these simulations, the model that has generated the underlying data is also the best performing model, as assessed by its predictive accuracy and the number of simulated participants predicted best. Thus, we can confidently say that our cross-validation procedure distinguishes between the two assessed model classes. Moreover, in the cases where the Function Learning or Function Learning* model has 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 S1).

Parameter Recovery

Another important question is whether or not the reported parameter estimates of the two Function Learning models are reliable and recoverable. 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 generate the underlying data. We present parameter recovery results for the Function Learning model for Experiment 1 and the Function Learning* model for Experiment 2, both using the UCB sampling strategy. We report the results in Figure 6, with the generating parameter estimate on the x-axis and the recovered parameter estimate on the y-axis.

For Experiment 1, the correlation between the generating and the recovered length-scale λ is $r = .62$, $p < .001$, the correlation between the generating and the recovered exploration factor β is $r = 0.62$, $p < .001$, and the correlation between the generating and the recovered softmax temperature parameter τ is $r = 0.91$, $p < .001$. For Experiment 2, the correlation between the generating and the recovered λ is $r = 0.91$, $p < .001$, for β the correlation is $r = 0.77$, $p < .001$, and for τ the correlation is $r = 0.76$, $p < .001$.

These results show that the correlation between the generating and the recovered parameters is high for both 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 S1) are recoverable, reliable, and therefore interpretable. Importantly, we find that estimates for β (exploration bonus) and τ (softmax temperature) are indeed recoverable, providing evidence for the existence of a *directed* exploration bonus¹¹, as a separate phenomena from random, undirected exploration⁴³ in our behavioral data.

Further behavioral Analysis

Learning over trials and rounds

We assessed whether participants improved more strongly over trials or over rounds (Fig. S3). If they are 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[§]. Results (from one-sample t-tests with $\mu_0 = 0$) show that participants' scores improve significantly over trials for both Experiment 1 ($t(80) = 5.57$, $p < .001$, $d = 0.62$) and Experiment 2 ($t(79) = 2.78$, $p < .001$, $d = 0.31$). Over successive rounds, there was a negative influence on performance in Experiment 1 ($t(80) = -2.78$, $p = .007$, $d = 0.3$) and no difference in Experiment 2 ($t(79) = 0.21$, $p = 0.834$, $d = 0.02$).

[§]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

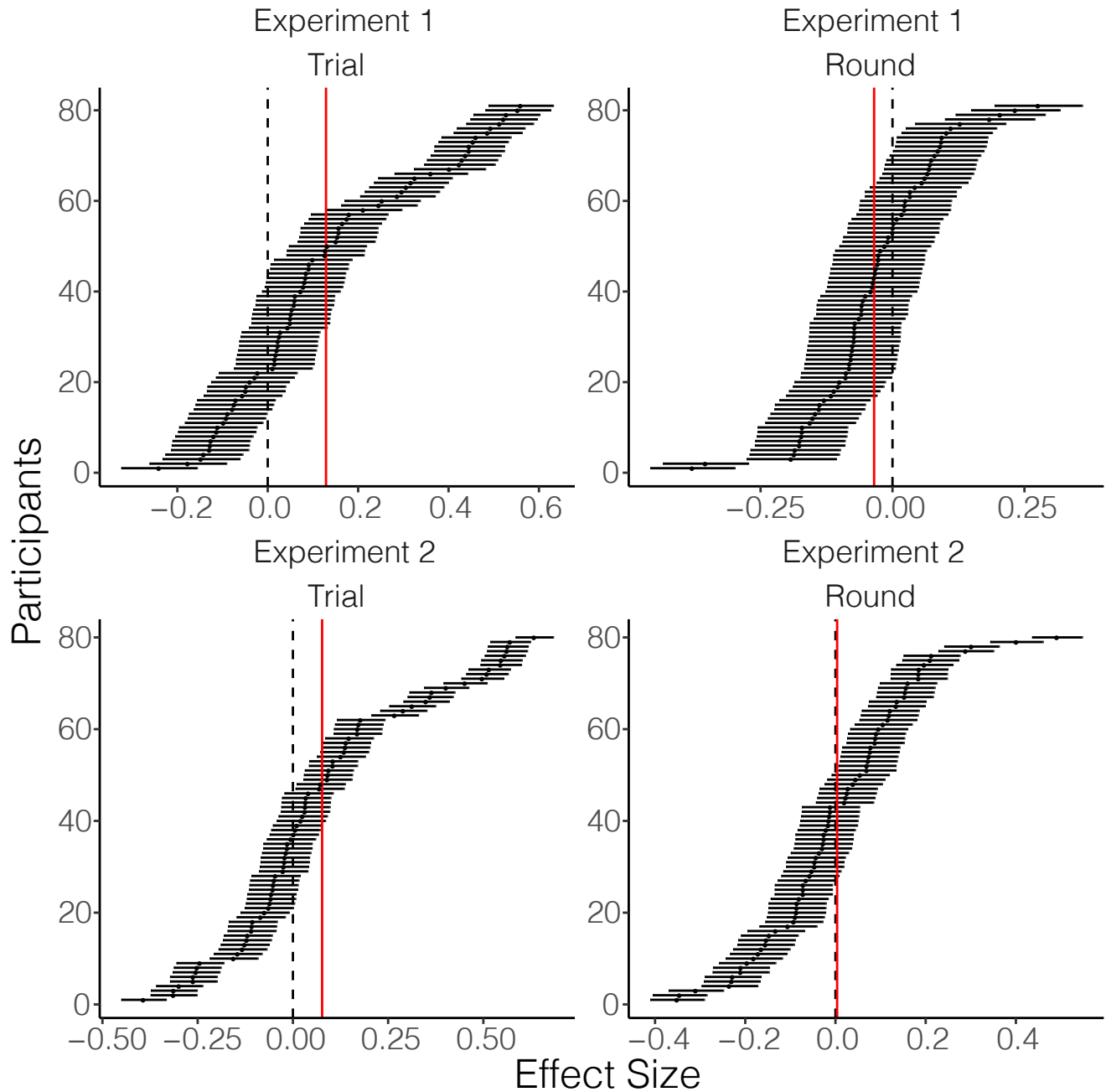


Figure S3. Learning over trials and rounds. Average correlational effect size of trial and round on score per participant as assessed by a standardized linear regression. Participants are ordered by effect size in decreasing order. Dashed lines indicate no effect. Red lines indicate average effect size. Whereas participants consistently improve over trials, there is no effect over rounds.

Table S1. Modeling Results

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

Note: Parameter estimates are the median over all participants. There were 81 participants in Experiment 1 and 80 participants in Experiment 2. The best performing model for each experiment is highlighted in boldface. Asterisks (*) indicate a localized variant of a model.