Exploration and generalization in vast spaces

Charley M. Wu[1,2,*,+], **Eric Schulz**[3,+], **Maarten Speekenbrink**[3], **Jonathan D. Nelson**[1,4], **and Björn Meder**[1,5]

[1]Center for Adaptive Behavior & Cognition, Max Planck Institute for Human Development; Lentzeallee 94, 14195 Berlin, Germany
[2]Center for Adaptive Rationality, Max Planck Institute for Human Development; Lentzeallee 94, 14195 Berlin, Germany
[3]Department of Experimental Psychology, University College London; 26 Bedford Way, London WC1H 0AP, United Kingdom
[4]School of Psychology, University of Surrey, 388 Stag Hill, Guildford GU2 7XH, UK
[5]MPRG iSearch, Max Planck Institute for Human Development; Lentzeallee 94, 14195 Berlin, Germany
[*]cwu@mpib-berlin.mpg.de
[+]these authors contributed equally to this work

ABSTRACT

Foraging for food, developing new medicines, and learning complex games are search problems with vast numbers of possible actions. Under time or resource constraints, optimal solutions are generally unobtainable. How do humans generalize and learn which actions to take when not all outcomes can be explored? We present two behavioural experiments and competitively test 27 models for predicting individual search decisions. We find that a Bayesian function learning model, combined with an optimistic sampling strategy, robustly captures how humans use generalization to guide search behaviour. 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. Importantly, our modelling results and parameter estimates are recoverable, and can be used to simulate human-like performance, bridging a critical gap between human and machine learning.

Introduction

From engineering proteins for medical treatment[1] to mastering a game like Go[2], many complex tasks can be described as search problems[3]. Frequently, these tasks come with a vast space of possible actions, each corresponding to some reward that can only be observed through experience. In such problems, one must learn to balance the dual goals of exploring unknown options, while also exploiting existing knowledge for immediate returns. This frames the *exploration-exploitation dilemma*, typically studied using the multi-armed bandit framework[*4,5], with the assumption that each option has its own reward distribution to be learned independently. Yet under real-world constraints of limited time or resources, it is not enough to know *when* to explore, but also *where*. How could an intelligent agent, biological or machine, learn which actions to take when not all outcomes can be explored?

There is an intriguing gap between human and machine learning, since humans are able to quickly learn and adapt to unfamiliar environments, where the same situation is rarely encountered twice[6,7]. This contrasts with traditional approaches to reinforcement learning, which learn about the distribution of rewards for each state independently[8]. Such an approach falls short in more realistic scenarios where it is

[*]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).

impossible to observe the outcomes of all possible states and actions[9,10]. How could an agent efficiently learn and make intelligent decisions about where to explore in problems with a vast space of possible actions?

In computer science, one method for dealing with vast state spaces is to use *function learning* as a mechanism to generalize prior experience to unobserved states[11]. The function learning approach relates different state-action contexts to each other by approximating a global value function over all contexts, including unobserved outcomes[7]. This allows for generalization to vast and potentially infinite state spaces, based on a small number of observations. Additionally, function learning scales to problems with complex sequential dynamics and has been used in tandem with *restricted search* methods such as Monte Carlo sampling for navigating intractably large search trees[2,12]. While restricted search methods have been proposed as models of human reinforcement learning in planning tasks[13,14], here we focus on situations in which a rich model of environmental structure supports learning and generalization[15].

Function learning has been successfully utilized for adaptive generalization in various machine learning applications[16,17]. However, relatively little is known about how humans generalize *in vivo* (e.g., in a search task). Building on previous work exploring inductive biases in pure function learning contexts[18,19], and human behaviour in univariate function optimization[20], we present the first definitive research on how people utilize generalization to effectively learn and search for rewards in large state spaces. Across two studies using uni- and bivariate versions of a multi-armed bandit, we compare 27 different models in their ability to predict individual human behaviour.

In both experiments, the vast majority of individual subjects are best captured by a model combining function learning using Gaussian Process ($\mathcal{GP}$) regression, with an optimistic Upper Confidence Bound (UCB) sampling strategy that directly balances rewards and uncertainty. Importantly, we recover meaningful and robust estimates of the nature of human generalization, showing the limits of traditional models of associative learning[21] in tasks where the environmental structure supports learning and inference. Interestingly, the most predictive model of the behavioural data is also the only Bayesian optimization algorithm with competitive guarantees[22]. This result has rich theoretical implications for reinforcement learning.

The main contributions of this paper are threefold:

1. We introduce a novel paradigm, *the spatially correlated multi-armed bandit*, which allows us to study the extent to which people use generalization to guide search through vast problem spaces.

2. We find that a Bayesian model of function learning robustly captures how humans generalize and learn about the structure of the environment.

3. Participants solve the exploration-exploitation dilemma by optimistically inflating expectations of reward by the underlying uncertainty, with recoverable evidence for the separate phenomena of directed and undirected exploration.

Results

A useful inductive bias in many real world search tasks is to assume a spatial correlation between rewards (i.e., clumpiness of resource distribution)[23]. We present human data and modelling results from two experiments using spatially correlated multi-armed bandits on univariate (Experiment 1) and bivariate (Experiment 2) environments (Fig. 1). The spatial correlation of rewards provides a context to each arm of the bandit, which can be learned and used to generalize to yet unobserved contexts, thereby guiding search

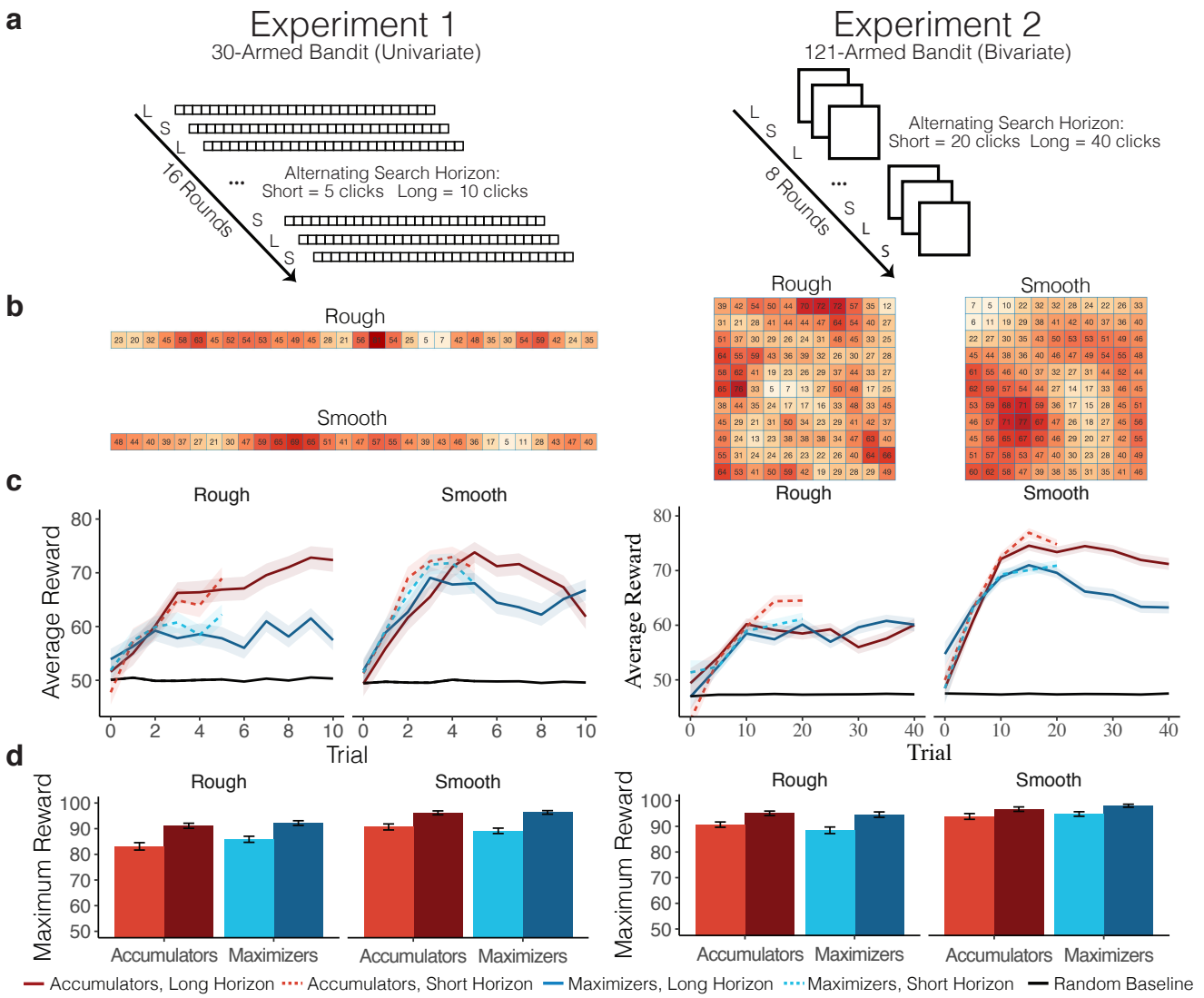

Figure 1. Procedure and behavioural results. Both experiments used a 2×2 between-subject design, manipulating the type of environment (Rough or Smooth) and the payoff condition (Accumulators or Maximizers). **a**, Experiment 1 used a 1D array of 30 possible options, while Experiment 2 used a 2D array (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 horizon order counter-balanced between subjects. **b**, Examples of fully revealed search environments, whereas tiles were initially blank at the beginning of each round, except for a single randomly revealed tile. Subjects were assigned to one of two different classes of environments, differing in the extent of spatial correlations (smoothness of the environment, see Methods). We describe behavioural results according to the average reward earned (**c**, Accumulator goal) and in terms of the maximum reward revealed up until the end of each round (**d**, Maximizer goal), where colours indicate the assigned payoff condition. Short horizon trials are indicated by lighter colours and/or dashed lines. Shaded regions (**c**) and error bars (**d**) show standard error of the mean. Black lines represent a random baseline simulated over 10,000 rounds.

71 decisions. Additionally, since recent work has connected both spatial and conceptual representations to

a common neural substrate[24], our results in a spatial domain provide potential pathways to other search domains, such as contextual[25,26] or semantic search[27,28].

Experiment 1

Participants searched for rewards on a 1×30 grid world, where each tile was a reward-generating arm of the bandit (Fig. 1a). The mean rewards of each arm were spatially correlated, with stronger correlations in smooth than in rough environments (between subjects; Fig. 1b). Participants were either assigned the goal of accumulating the largest average reward (*Accumulators*), thereby balancing exploration-exploitation, or of finding the best overall tile (*Maximizers*), an exploration goal directed towards finding the global maximum. We hypothesized that, if search behaviour is guided by function learning, participants would perform better and learn faster in smooth environments, since stronger spatial correlations reveal more information about nearby tiles[29].

Participants in smooth environments obtained higher average rewards than participants in rough environments ($t(79) = 3.58$, $p < .001$, $d = 0.8$), consistent with the hypothesis that spatial patterns in the environment can be learned and used to guide search. The learning curves in Figure 1c show that longer search horizons (solid lines) do not always lead to higher average reward ($t(80) = 0.60$, $p > .5$, $d = 0.07$). We analysed both average reward and the maximum reward obtained for each subject, irrespective of their payoff condition (Maximizer or Accumulator). Interestingly, while Accumulators performed better than Maximizers on the average reward criterion ($t(79) = 2.89$, $p < .01$, $d = 0.65$), they performed equally well when trying to find the highest overall reward ($t(79) = -0.43$, $p > .6$, $d = 0.09$; Fig. 1d). Thus, a strategy balancing exploration and exploitation, at least for human learners, may find the global maximum *en passant*.

Experiment 2

Experiment 2 had the same design as Experiment 1, but used a 11×11 grid representing an underlying bivariate reward function (Fig. 1 right). Participants obtained higher rewards in smooth environments than in rough environments ($t(78) = 6.55$, $p < .001$, $d = 1.47$), as in Experiment 1, but with a larger effect size. As in Experiment 1, Accumulators were as good as Maximizers at discovering the highest rewards ($t(78) = 0.01$, $p > .5$, $d = 0.002$). In Experiment 2, however, Accumulators did not perform substantially better than Maximizers in terms of average reward ($t(78) = -1.31$, $p > .1$, $d = 0.29$). Again, short search horizons led to the same level of performance as longer horizons, ($t(79) = -0.96$, $p > .34$, $d = 0.11$), suggesting that for people, frugal search can be quite efficient. We also present full results for learning over rounds and trials in the SI.

Modelling Generalization and Search

We competitively tested a diverse set of 27 different models in their ability to predict each subject's trial-by-trial choices (Fig. S1, Table S1). These models include different combinations of models of learning and sampling strategies, along with simple heuristics, which make predictions without maintaining a model of the world. By far the most successful models used *Gaussian Process* ($\mathcal{GP}$) regression[30,31] as a method for learning an underlying value function relating all state-action contexts to each other, and *Upper Confidence Bound* (UCB) sampling[32] for predicting where to sample next. This maps onto the distinction between belief and sampling models, central to theories in statistics[33], psychology[34], and philosophy of science[35].

$\mathcal{GP}$s provide an expressive model for human function learning, and in contrast to neural network function approximators[36], yield psychologically interpretable parameter estimates. $\mathcal{GP}$ function learning

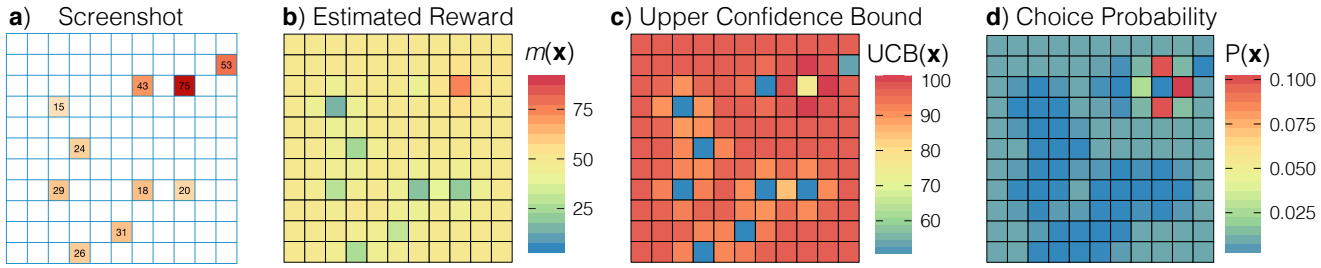

Figure 2. Overview of $\mathcal{GP}$-UCB specified using median participant parameter estimates (see Table S1). **a**, Screenshot of Experiment 2. Participants were allowed to select any tile until the search horizon was exhausted. **b**, Estimated reward as predicted by the $\mathcal{GP}$ function learning engine, based on the sampled points in Panel **a**. (Not shown, the estimated uncertainty). **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., lower temperature values lead to more precise predictions).

can guide search by making normally distributed predictions about the expected mean $m(\mathbf{x})$ and the underlying uncertainty $s(\mathbf{x})$ (estimated as a standard deviation) for each option $\mathbf{x}$ in the global state space (see Fig. 2b), conditioned on a finite number of previous observations of rewards $\mathbf{y}_T = [y_1, y_2, \ldots, y_T]^\top$ at inputs $\mathbf{X}_T = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$. Similarities between points are modeled by a *Radial Basis Function* (RBF) kernel:

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

where λ governs how quickly correlations between points $\mathbf{x}$ and $\mathbf{x}'$ (e.g., two tiles on the grid) decay towards zero as their distance increases. We use λ as a free parameter, which can be interpreted psychologically as the extent to which people generalize spatially. This is similar to Shepard's gradient of generalization[37], which also models generalization as an exponentially decreasing function of distance.

Given estimates about expected rewards $m(\mathbf{x})$ and the underlying uncertainty $s(\mathbf{x})$ (expressed as a standard deviation), UCB sampling makes predictions about where participants search next (Fig. 2c) using a sum,

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

where β is a free parameter governing how much the reduction of uncertainty is weighted relative to expectations of reward. This trade-off between exploiting known high-value rewards and exploring to reduce uncertainty[38] can be interpreted as optimistically inflating expected rewards by their attached uncertainty, and can be decomposed into two separate components that only sample points based on high expected reward (Pure Exploitation) or high uncertainty (Pure Exploration).

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

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

Figure 2 shows how the $\mathcal{GP}$-UCB Function Learning 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

a *Mean Tracker*. The Mean Tracker is a type of Kalman Filter without temporal dynamics[5], and is a more traditional type of associative learning model that learns the distribution of rewards for each state independently. Like the $\mathcal{GP}$-UCB Function Learning Model, the Mean Tracker also generates normally distributed predictions $m(\mathbf{x})$ and $s(\mathbf{x})$, which we combine with the same set of sampling strategies to make probabilistic predictions about search.

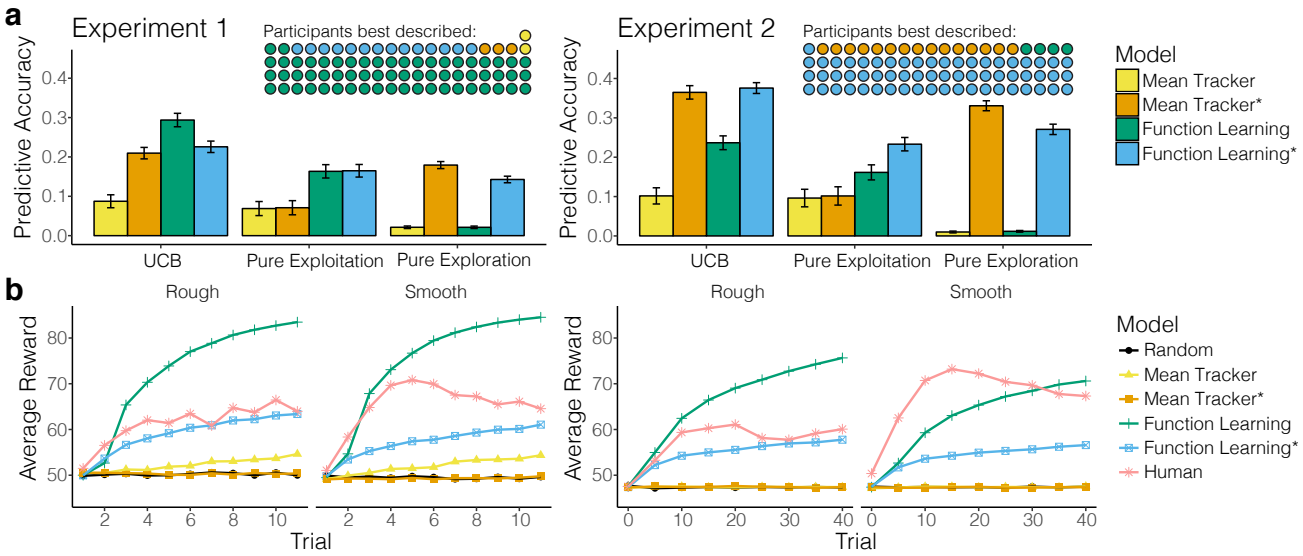

Figure 3. Modelling results. **a**, Cross-validated predictive accuracy of each model (bars), with the number of participants best described shown as an icon array (inset; aggregated by sampling strategies). Asterisks (*) indicate a localized variant of the Mean Tracker or Function Learning models, where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

Modelling results

Experiment 1

Instead of learning rewards for each state independently, as assumed in the Mean Tracker and traditional associative learning models, 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 high expectations of reward and the reduction of uncertainty are necessary components for the Function Learning Model to predict human search behaviour, with Pure Exploitation ($t(80) = 8.85$, $p < .001$, $d = 0.98$) and Pure Exploration ($t(80) = 16.63$, $p < .001$, $d = 1.85$) performing worse at predicting human behaviour than the combined UCB algorithm.

The distance between sequential samples was more localized than chance ($t(160) = 31.2$, $p < .0001$, $d = 1.92$; see Fig. S5), as has also been observed in semantic search[27] and causal learning[39] domains. Thus, we created a localized variant of both Mean Tracker and Function Learning models (indicated by an asterisk *; Fig. 3a), giving larger weight to options closer to the previous choice (see Methods). While localization improved predictive accuracy for Mean Tracker predictions ($t(80) = 16.13$, $p < .001$,

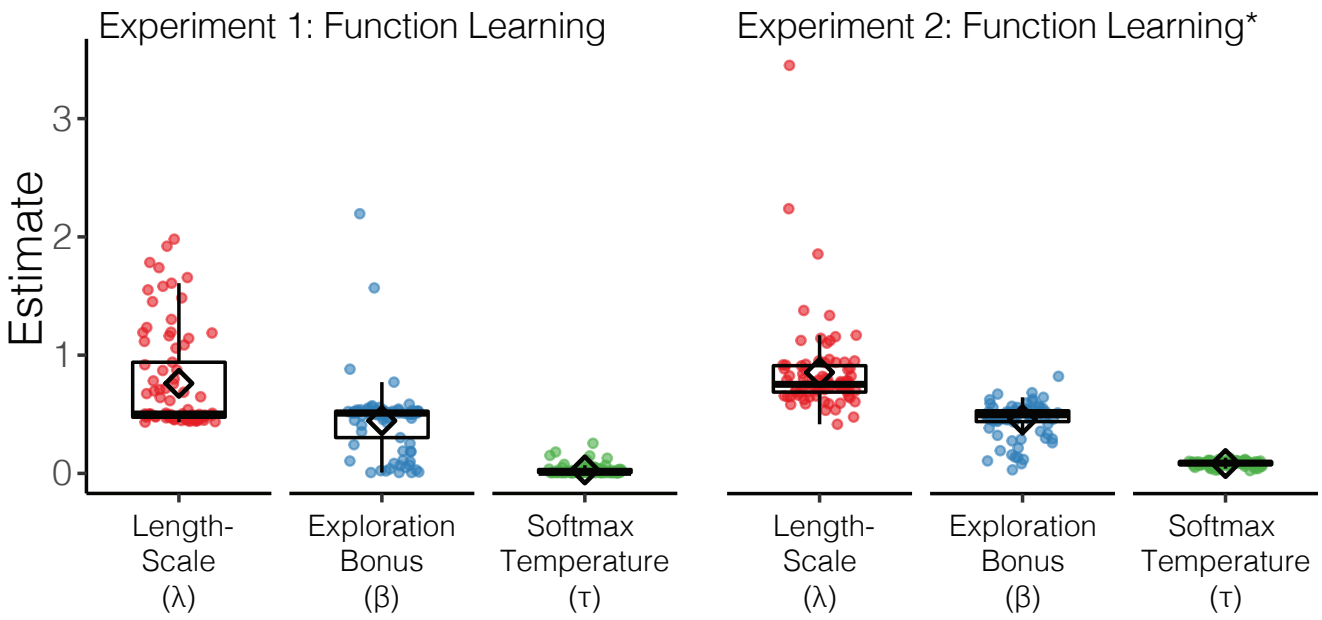

Figure 4. Parameter estimates of the best predicting model for each experiment. Each coloured point is the median estimate of a participant, with boxplots showing the median (line), mean (diamond), and interquartile range. λ is the length-scale of the RBF kernel reflecting the extent to which people generalize, β is the exploration bonus of the UCB sampling strategy, and τ is the temperature of the softmax choice rule.

$d = 1.79$; UCB), it led to worse performance for the $\mathcal{GP}$-UCB Function Learning Model ($t(80) = -5.05$, $p < .001$, $d = 0.56$).

Overall, 56 out of 81 participants were best described by the Function Learning Model combining $\mathcal{GP}$ learning with UCB sampling (66 of 81 participants including the localized variant), again showing that generalization is important for predicting search behaviour. Figure 3b shows simulated learning curves of each model in comparison to human performance, where models were specified using parameters sampled from participant estimates. Whereas the Mean Tracker models achieve performance close to random, the Function Learning Model (and its localized variant) behaves sensibly, by utilizing the structure of the environment, and achieves similar or better performance than humans.

Looking more closely at the parameter estimates of the $\mathcal{GP}$-UCB Function Learning Model (Fig. 4), we find that people tend to underestimate the extent of spatial correlations, with estimated λ values significantly lower than the ground truth ($\lambda_{Smooth} = 2$ and $\lambda_{Rough} = 1$) for both Smooth (mean estimate: $\hat{\lambda} = 0.82$, $t(41) = -17.60$, $p < .001$, $d = 2.71$) and Rough environments ($\hat{\lambda} = 0.78$, $t(38) = -3.89$, $p < .001$, $d = 0.62$), which could be interpreted as a tendency to avoid overgeneralization[7,40]. Remarkably, simulations suggest that a tendency towards undergeneralization can benefit search performance (Fig. S4). The exploration bonus of UCB sampling (β) was robustly estimated above 0 ($\hat{\beta} = 0.47$, $t(80) = 12.78$, $p < .001$, $d = 1.42$, compared to the lower estimation bound), indicating participants valued the exploration of uncertain options, along with exploiting high expectations of reward. Additionally, we found very low estimates of the softmax temperature (τ), corresponding to more precise model predictions ($\hat{\tau} = 0.01$). The model comparison and parameter estimates were highly robust and recoverable (Figs. S2-S3).

Experiment 2

In a more complex bivariate environment, the Function Learning Model again predicted participants' choices more accurately than the Mean Tracker Model, for both its standard ($t(79) = 9.99$, $p < .001$, $d = 1.12$; UCB) and localized variants ($t(79) = 2.05$, $p < .05$, $d = 0.23$; UCB, Fig. 3a). In Experiment 2's two-dimensional search environment, adding localization improved predictions for both Mean Tracker ($t(79) = 19.92$, $p < .001$, $d = 2.23$; UCB) and the Function Learning Model ($t(79) = 10.47$, $p < .001$, $d = 1.17$; UCB), in line with the stronger tendency towards localized sampling behaviour in Experiment 2 (see Fig. S5). 61 out of 80 participants were best predicted by the Function Learning* Model with UCB sampling, whereas only 12 participants were best described by the Mean Tracker* Model with UCB. Again, both components of the UCB strategy—the expected reward ($t(79) = -6.44$, $p < .001$, $d = 0.72$) and the attached uncertainty ($t(79) = -14.32$, $p < .001$, $d = 1.60$)—were necessary to predict choices.

As in Experiment 1, simulated learning curves of the Mean Tracker models performed close to random, whereas both variants of the Function Learning Model achieved performance similar to participants (Fig. 3b). Median parameter estimates per participant from the Function Learning* Model (Fig. 4) showed that participants again underestimated the strength of the underlying spatial correlation in both Smooth ($\hat{\lambda} = 0.92$, $t(42) = -14.62$, $p < .001$, $d = 2.22$; comparison to $\lambda_{Smooth} = 2$) and Rough environments ($\hat{\lambda} = 0.78$, $t(36) = -5.31$, $p < .001$, $d = 0.87$; comparison to $\lambda_{Rough} = 1$), suggesting a robust tendency to undergeneralize. The estimated exploration bonus β was again greater than 0 ($\hat{\beta} = 0.45$, $t(79) = 27.02$, $p < .001$, $d = 3.02$, compared to the lower estimation bound), while the estimated softmax temperature parameter τ was slightly larger than in Experiment 1 ($\hat{\tau} = 0.09$; see Table S1), likely due to localization. As in Experiment 1, the model comparison and parameter estimates were highly robust and recoverable (Figs. S2-S3).

Experiment 2 therefore replicated the main findings of Experiment 1. Taken together, these results provide strong evidence that human search behaviour is best explained by a combination of function learning paired with an optimistic trade-off between exploration and exploitation.

Discussion

How should one behave in situations where the number of possible actions is vast and not all possibilities can be explored? Function learning provides a possible mechanism for generalization, by relating different options using spatial context. Gaussian Processes ($\mathcal{GP}$) combined with Upper Confidence Bound (UCB) sampling have been successfully applied to problems in ecology[41], robotics[42], and biology[43], but there has been little psychological research on human behaviour in such tasks.

We have presented the first study to apply cognitive modelling to predict individual decisions in such a complex search task. Our rigorous comparison of 27 models yielded robust and recoverable model comparisons (Fig. S2) and parameter estimates (Fig. S3). The spatial correlation of rewards made it possible to generalize to unseen rewards by learning an approximate underlying value function based on spatial context. Results show that participants capitalized on spatial context in all task variants, and performed best in environments with the strongest spatial correlations.

Through multiple analyses, including trial-by-trial predictive cross-validation and simulated behaviour using participant parameter estimates, we competitively studied which models best predicted human behaviour. The vast majority of participants were best described by the $\mathcal{GP}$-UCB Function Learning Model or its localized variant. Parameter estimates from the best-fitting $\mathcal{GP}$-UCB models suggest there was a systematic tendency to undergeneralize the extent of spatial correlations, which can be a beneficial bias for search (Fig. S4).

Whereas previous research on exploration bonuses has had mixed results[5,10,44], we find robustly

recoverable parameter estimates for the separate phenomena of directed exploration encoded in β and the random, undirected exploration encoded in the softmax temperature parameter τ, in the $\mathcal{GP}$-UCB Function Learning Model. Even though UCB sampling is both *optimistic* (always treating uncertainty as positive) and *myopic* (only planning the next timestep), it is nonetheless the only algorithm with known performance guarantees in a bandit setting (i.e.,sublinear regret, or in other words, monotonically increasing average reward)[22]. This suggests a remarkable concurrence between intuitive human strategies and the state of the art in machine learning.

The $\mathcal{GP}$-UCB Function Learning Model also offers many opportunities for theory integration. The Mean Tracker models as specified here can be reformulated as special case of a $\mathcal{GP}$ regression model[45] and hence implemented in our Function Learning Model. In addition, when the length-scale of the RBF kernel approaches zero ($\lambda \to 0$), the Function Learning Model effectively assumes state-independence, as in the Mean Tracker Model. Thus, there may be a continuum of reinforcement learning models, ranging from the traditional assumption of state independence to the opposite extreme, of complete state interdependence. Moreover, a $\mathcal{GP}$ is also equivalent to a Bayesian Neural Network with infinite nodes[46], suggesting a further link to distributed function learning models[47]. Indeed, one explanation for the impressive performance of Deep Reinforcement Learning[12] is that neural networks are specifically a powerful type of function approximator[48].

Lastly, recent findings have connected both spatial and conceptual representations to a common neural substrate[24], suggesting a potential avenue for applying the same $\mathcal{GP}$-UCB Function Learning Model for modelling human behaviour in domains such as contextual[25,26] or semantic search[27,28]. Marrying powerful yet interpretable function learning techniques with methods commonly applied in the reinforcement learning literature can further advance understanding of adaptive behaviour in complex and uncertain environments.

Methods

Participants

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

Design

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

Materials and procedure

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

Payoff conditions

We compared performance under two different payoff conditions, requiring either a balance between exploration and exploitation (*Accumulators*) or corresponding to consistently making exploration decisions (*Maximizers*). In each payoff condition, participants received a performance contingent bonus of up to \$1.50. *Accumulators* were given a bonus based on the 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, whereas *Maximizers* were rewarded using the ratio of the highest observed reward to the global optimum, $(\frac{\max y_t}{y^*})^4$, taken to the power of 4 to exaggerate differences in the upper range of performance and for between-group parity in expected earnings across payoff conditions. Both conditions were equally weighted across all rounds and used noisy but unscaled observations to assign a bonus of up to \$1.50. Subjects were informed in dollars about the bonus earned at the end of each round.

Smoothness of the environment

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

Search horizons

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

Models of Learning

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

Function Learning

The Function Learning Model adaptively learns an underlying function mapping spatial locations onto rewards. We use Gaussian Process ($\mathcal{GP}$) regression as a Bayesian method of function learning[31]. A $\mathcal{GP}$ is defined as a collection of points, any subset of which is multivariate Gaussian. Let $f : \mathcal{X} \rightarrow \mathbb{R}^n$ denote a function over input space $\mathcal{X}$ that maps to real-valued scalar outputs. This function can be modelled as a random draw from a $\mathcal{GP}$:

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

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

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

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

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

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

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

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

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

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

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

Mean Tracker

The Mean Tracker is a type of traditional associative learning model, which assumes the average reward associated with each option is constant over time (i.e., no temporal dynamics, as opposed to the assumptions

[†]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.

of a Kalman Filter or Temporal Difference Learning)[5], as is the case in our experimental search tasks. In contrast to the Function Learning Model, the Mean Tracker learns the rewards of each option independently, by computing an independent posterior distribution for the mean μ_j for each option j. We implement a version that assumes rewards are normally distributed (as in the $\mathcal{GP}$ Function Learning Model), with a known variance but unknown mean, where the prior distribution of the mean is again a normal distribution. This implies that the posterior distribution for each mean is also a normal distribution:

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

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

$$m_{j,t} = m_{j,t-1} + \delta_{j,t} G_{j,t} \left[y_t - m_{j,t-1} \right] \tag{12}$$

$$v_{j,t} = \left[1 - \delta_{j,t} G_{j,t} \right] v_{j,t-1} \tag{13}$$

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

$$G_{j,t} = \frac{v_{j,t-1}}{v_{j,t-1} + \theta_\varepsilon^2} \tag{14}$$

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

Sampling Strategies

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

Upper Confidence Bound Sampling

Given the posterior predictive mean $m_t(\mathbf{x})$ and its attached standard deviation $s_t(\mathbf{x}) = \sqrt{v_t(\mathbf{x})}$, we calculate the upper confidence bound using a simple sum

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

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

Pure Exploitation and Pure Exploration

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

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

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

Localization of Models

To penalize search options by the distance from the previous choice, we weighted each option by the 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 transformation. For the special case where $\mathbf{x} = \mathbf{x}'$, we set $\text{IMD}(\mathbf{x},\mathbf{x}') = 1$. Localized models are indicated by an asterix (*).

Model Comparison

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

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

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

Cross Validation

We fit all models—per participant—using cross-validated MLE, with either a Differential Evolution algorithm[49] or a grid search if the model contained only a single parameter. Parameter estimates are constrained to positive values in the range $[\exp(-5), \exp(5)]$.

Cross-validation is performed by first separating participant data according to horizon length, which alternated between rounds within subject. For each participant, half of the rounds corresponded to a short horizon and the other half corresponded to a long horizon. Within all rounds of each horizon length, we use leave-one-out cross-validation to iteratively form a training set by leaving out a single round, computing a MLE on the training set, and then generating out of sample predictions on the remaining round. This is repeated for all combinations of training set and test set, and for both short and long horizon sets. The cross-validation procedure yielded one set of parameter estimates per round, per participant, and out-of-sample predictions for 120 choices in Experiment 1 and 240 choices in Experiment 2 (per participant). In total, cross-validated model comparisons for both experiments required approximately 50,000 hours of computation, or about 3 days distributed across a 716 CPU cluster.

Predictive Accuracy

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

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

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

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. 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).
7. 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).
8. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction*, vol. 1 (MIT press Cambridge, 1998).
9. Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. Building machines that learn and think like people. *Behav. Brain Sci.* 1–101 (2016).
10. 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).
11. Tesauro, G. Practical issues in temporal difference learning. *Mach. learning* **8**, 257–277 (1992).
12. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nat.* **518**, 529–533 (2015).
13. Huys, Q. J. *et al.* Interplay of approximate planning strategies. *Proc. Natl. Acad. Sci.* **112**, 3098–3103 (2015).
14. Solway, A. & Botvinick, M. M. Evidence integration in model-based tree search. *Proc. Natl. Acad. Sci.* **112**, 11708–11713 (2015).
15. 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).
16. Rasmussen, C. E. & Kuss, M. Gaussian processes in reinforcement learning. In *Advances in Neural Information Processing Systems*, vol. 4, 1 (2003).

17. Sutton, R. S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Adv. Neural Inf. Process. Syst.* 1038–1044 (1996).

18. 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).

19. Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M. & Gershman, S. J. Compositional inductive biases in function learning. *bioRxiv* (2016). DOI 10.1101/091298.

20. Borji, A. & Itti, L. Bayesian optimization explains human active search. In *Advances in Neural Information Processing Systems*, 55–63 (2013).

21. Dayan, P. & Niv, Y. Reinforcement learning: Tfhe good, the bad and the ugly. *Curr. opinion neurobiology* **18**, 185–196 (2008).

22. 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).

23. Wilke, A. *et al.* A game of hide and seek: Expectations of clumpy resources influence hiding and searching patterns. *PloS one* **10**, e0130976 (2015).

24. Constantinescu, A. O., O'Reilly, J. X. & Behrens, T. E. Organizing conceptual knowledge in humans with a gridlike code. *Sci.* **352**, 1464–1468 (2016).

25. 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).

26. Schulz, E., Konstantinidis, E. & Speekenbrink, M. Putting bandits into context: How function learning supports decision making. *J. Exp. Psychol. Learn. Mem. Cogn.* (in press).

27. Hills, T. T., Jones, M. N. & Todd, P. M. Optimal foraging in semantic memory. *Psychol. review* **119**, 431 (2012).

28. Abbott, J. T., Austerweil, J. L. & Griffiths, T. L. Random walks on semantic networks can resemble optimal foraging. *Psychol. Rev.* **122**, 558–569 (2015).

29. 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).

30. Rasmussen, C. & Williams, C. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning (MIT Press, 2006).

31. Schulz, E., Speekenbrink, M. & Krause, A. A tutorial on Gaussian process regression with a focus on exploration-exploitation scenarios. *bioRxiv* 095190 (2016).

32. Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.* **3**, 397–422 (2002).

33. Lindley, D. V. On a measure of the information provided by an experiment. *The Annals Math. Stat.* 986–1005 (1956).

34. Nelson, J. D. Finding useful questions: On bayesian diagnosticity, probability, impact, and information gain. *Psychol. Rev.* **112** (2005).

35. Crupi, V. & Tentori, K. State of the field: Measuring information and confirmation. *Stud. Hist. Philos. Sci. Part A* **47**, 81–90 (2014).

36. Le Roux, N. & Bengio, Y. Deep belief networks are compact universal approximators. *Neural computation* **22**, 2192–2207 (2010).

37. Shepard, R. N. Toward a universal law of generalization for psychological science. *Sci.* **237**, 1317–1323 (1987).

38. Kaufmann, E., Cappé, O. & Garivier, A. On Bayesian upper confidence bounds for bandit problems. In *Artificial Intelligence and Statistics*, 592–600 (2012).

39. Bramley, N. R., Dayan, P., Griffiths, T. L. & Lagnado, D. A. Formalizing neurath's ship: Approximate algorithms for online causal learning. *Psychol. review* **124**, 301 (2017).

40. 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).

41. 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).

42. Cully, A., Clune, J., Tarapore, D. & Mouret, J.-B. Robots that can adapt like animals. *Nat.* **521**, 503–507 (2015).

43. Sui, Y., Gotovos, A., Burdick, J. & Krause, A. Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning*, 997–1005 (2015).

44. Daw, N. D., O'doherty, J. P., Dayan, P., Seymour, B. & Dolan, R. J. Cortical substrates for exploratory decisions in humans. *Nat.* **441**, 876–879 (2006).

45. 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).

46. Neal, R. M. *Bayesian learning for neural networks* (Springer Science & Business Media, 1996).

47. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nat.* **521**, 436–444 (2015).

48. Schölkopf, B. Artificial intelligence: Learning to see and act. *Nat.* **518**, 486–487 (2015).

49. 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/.

50. Bonawitz, E., Denison, S., Gopnik, A. & Griffiths, T. L. Win-stay, lose-sample: A simple sequential algorithm for approximating bayesian inference. *Cogn. psychology* **74**, 35–65 (2014).

Acknowledgements

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

Author contributions statement

C.M.W. and E.S. designed the experiments, collected and analysed the data, and wrote the paper. M.S., J.D.N., and B.M. designed the experiments and wrote the paper.

Supplemental Materials: Exploration and generalization in vast spaces

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 Mean Tracking), 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 $\mathbf{x}^+$:

$$\mathrm{EXI}(\mathbf{x}) = \begin{cases} \Phi(Z)(m_t(\mathbf{x}) - m_t(\mathbf{x}^+)) + s_t(\mathbf{x})\phi(Z), & \text{if } s_t(\mathbf{x}) > 0 \\ 0, & \text{if } s_t(\mathbf{x}) = 0 \end{cases} \tag{S1}$$

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

Probability of Improvement

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

$$\begin{aligned} \mathrm{POI}(\mathbf{x}) &= P\left(f(\mathbf{x}) \geq f(\mathbf{x}^+)\right) \\ &= \Phi\left(\frac{m_t(\mathbf{x}) - m_t(\mathbf{x}^+)}{s_t(\mathbf{x})}\right) \end{aligned} \tag{S2}$$

Probability of Maximum Utility

The Probability of Maximum Utility (PMU) samples each option according to the probability that it results in the highest reward of all options in a particular context[5]. 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.

$$\mathrm{PMU}(\mathbf{x}) = P\left(f(\mathbf{x}_j) > f(\mathbf{x}_{i \neq j})\right) \tag{S3}$$

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.

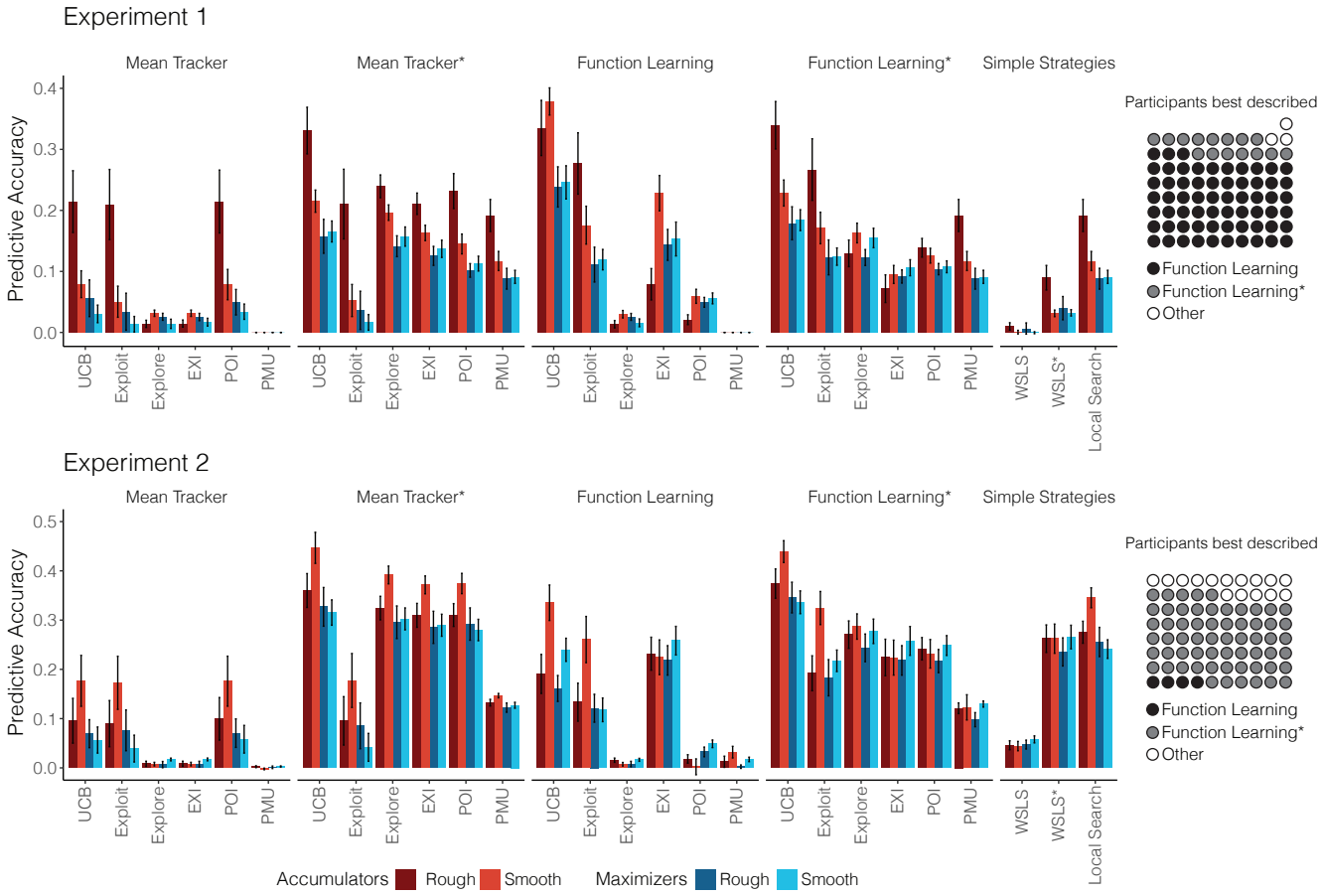

Figure S1. Full model comparison of all 27 models, with the learning model indicated above (or lack of in the case of simple heuristic strategies), and sampling strategy along the x-axis. Bars indicate predictive accuracy along with standard error, and are separate by payoff condition (colour) and environment type (darkness). Icon array (right) shows 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.

Simple Heuristic Strategies

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

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:

$$\mathrm{IMD}(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{n} |x_i - x_i'|, \tag{S4}$$

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

Win-Stay Lose-Sample

We also consider a form of a win-stay lose-sample (WSLS) heuristic[50], 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.

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*). See *Locality of sampling behaviour* for a behavioural analysis of locality.

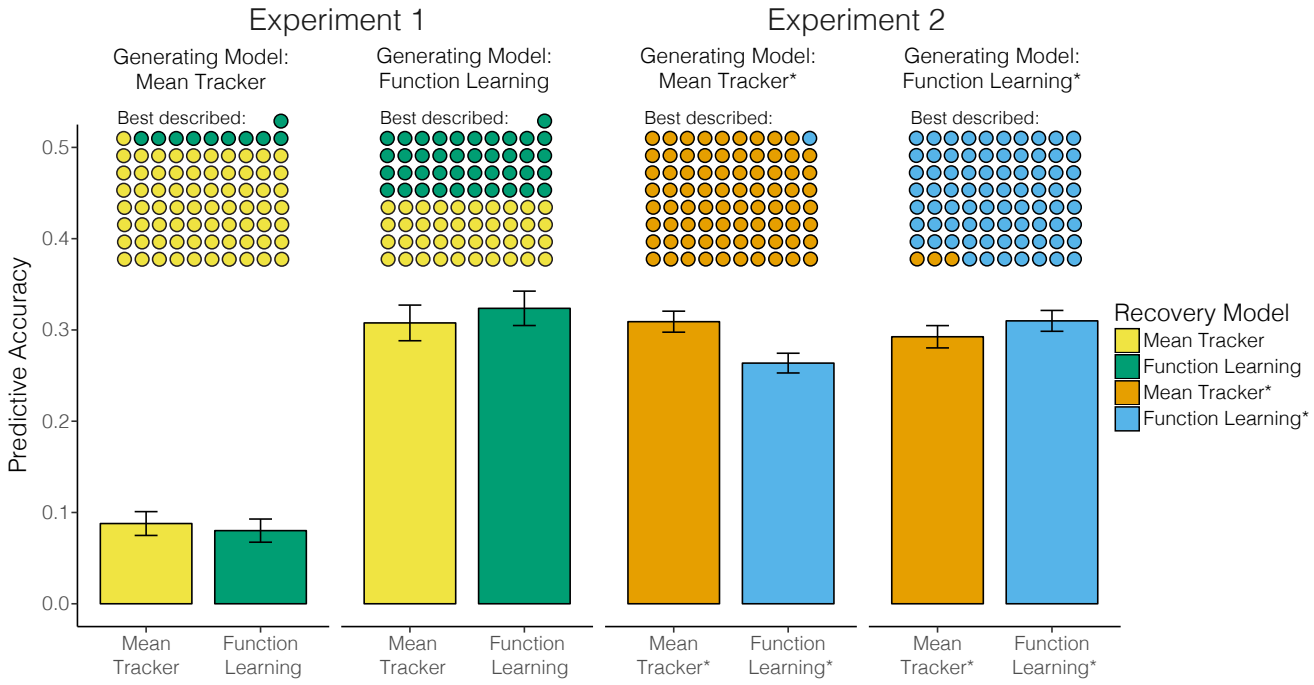

Figure S2. Model recovery results, where data was generated by the specified generating model using individual participant parameter estimates. The recovery process used the same cross-validation method used in the model comparison. We report the predictive accuracy of each candidate recovery model (bars including standard error) and the number of simulated participants best described (icon array). 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.

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 Mean Tracker and the Function Learning model for Experiment 1 and the Mean Tracker* 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 Mean Tracker 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 Mean Tracker model generates data using participant parameter estimates, the same Mean Tracker model achieves an average predictive accuracy of $R^2 = .1$ and describes 71 out 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 Mean Tracker 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 Mean Tracker model could mimic parts of the Function Learning behaviour.

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 Mean Tracker* model generated the data, the Mean Tracker* 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 Mean Tracker* 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.,

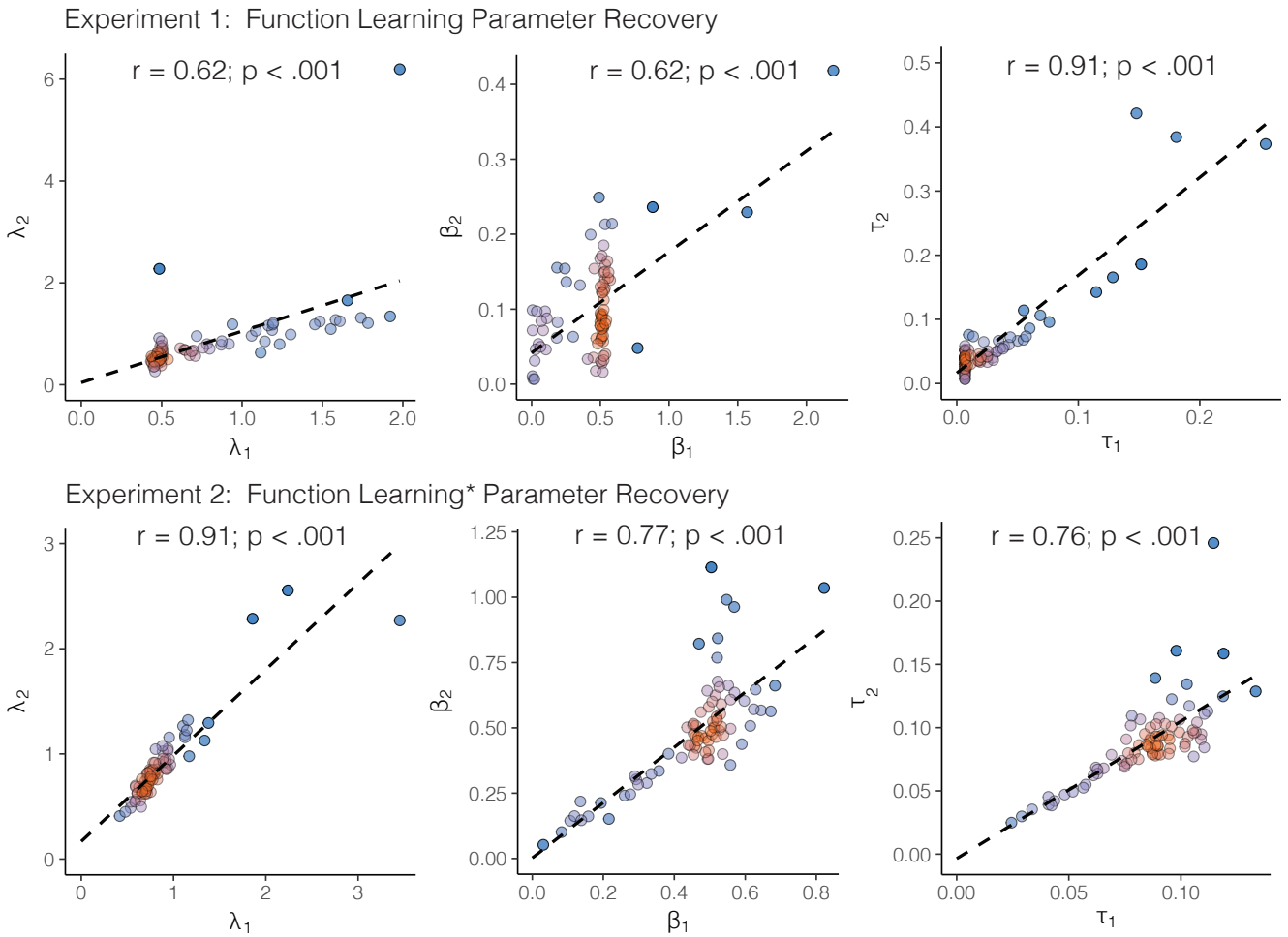

Figure S3. Parameter recovery results. The generating parameter estimate is on the x-axis and the recovered parameter estimate is on the y-axis. The generating parameter estimates are from the cross-validated participant parameter estimates, which were used to simulate data (see Model Recovery). Recovered parameter estimates are the result of the cross-validated model comparison (see 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, colours represent the bivariate kernel density estimate, with red indicating higher density.

the estimates based on participants' data), with the parameter estimates of the recovering model (i.e., the MLE from the cross-validation procedure on the simulated data). We assess whether the recovered parameter estimates are similar to the parameters that were used to generated the underlying data. We present parameter recovery results for the Function Learning model for Experiment 1 and the Function Learning* model for Experiment 2, both using the UCB sampling strategy. We report the results in Figure S3, with the generating parameter estimate on the x-axis and the recovered parameter estimate on the y-axis.

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

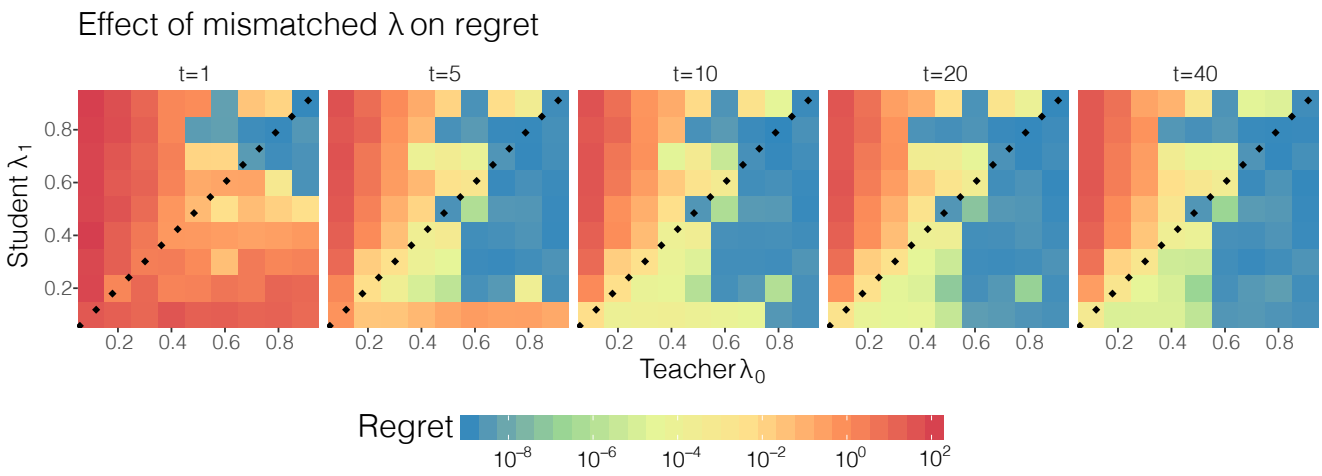

Figure S4. Mismatched length-scale simulation results. The generating teacher length-scale parameter is on the x-axis and the student length-scale parameter is on the y-axis. The teacher length-scale values were used to generating environments, while the student length-scale parameters were used to parameterize the $\mathcal{GP}$-UCB Function Learning Model to simulate human search performance. Each tile of the heat-map indicates the median regret of that particular λ_0-λ_1-combination, aggregated over 100 replications at trial numbers $t = \{1, 5, 10, 20, 40\}$. The dotted lines show where $\lambda_0 = \lambda_1$ and mark the difference between undergeneralization and overgeneralization, where points below the line show the regret produced by undergeneralizing student kernels. For readability, colours represent the log-regret of different λ_0-λ_1-combinations, with red indicating higher regret.

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 bonus[10], as a separate phenomena from random, undirected exploration[44] in our behavioural data.

Mismatched generalization

We assess the effect of mismatched λ-estimates on the performance of the $\mathcal{GP}$-UCB Function Learning Model. A mismatch is defined as estimating a different level of spatial correlations (captured by the per participant λ-estimates) than the ground truth in the environment ($\lambda_{Smooth} = 2$, and $\lambda_{Rough} = 1$ for both experiments). In both experiments, we found that participant λ-estimates were systematically lower than the true value (Fig. 4), which can be interpreted as a tendency to undergeneralize about the spatial correlation of rewards in the world. In order to test how this tendency to undergeneralize (i.e., underestimate λ) influences task performance, we present simulation results (Fig. S4) using different λ values in a teacher kernel (x-axis) and a student kernel (y-axis).

Both teacher and student kernels were RBF kernels, where the teacher kernel was parameterized 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 smoothness assumptions can be seen as misaligned. The student *overgeneralizes* when $\lambda_1 > \lambda_0$ (Fig. S4

above the dotted line), and *undergeneralizes* when $\lambda_1 > \lambda_0$ (Fig. S4 below the dotted line), as was captured by in our behavioural data.

We simulate every possible combination between $\lambda_0 = \{0.1, 0.2, \cdots, 1\}$ and $\lambda_1 = \{0.1, 0.2, \cdots, 1\}$, leading to 100 different combinations of student-teacher scenarios. For each of these combinations, we sample a bivariate target function from a $\mathcal{GP}$ parameterized by λ_0 and then use the $\mathcal{GP}$-UCB Function Learning Model parameterized by λ_1 to search for rewards. The exploration parameter β was set to 0.5 to resemble participant behaviour (Table S1).

Figure S4 shows the median regret for 100 replications for all 100 λ_0-λ_1-combination at trial $t = \{1, 5, 10, 20, 40\}$. Regret is defined as the difference between the reward obtained at trial t by sampling x_t, and the best possible reward that could have been obtained by sampling the global optimum $x_\star$, if the reward distributions of all options was fully known (i.e., with perfect knowledge).

$$R_t = f(x_\star) - f(x_t) \tag{S5}$$

The simulations revealed several interesting results. First of all, regret is generally lower (blue values) when the student undergeneralizes (below the dotted line) than when the student overgeneralizes (above the dotted line). This effect is more pronounced over time, whereby a mismatch in the direction of undergeneralization recovers over time (less regret for larger values of t). This is not the case for a mismatch in the direction of overgeneralization, which continues to produce high regret, even at $t = 40$. Thus, undergeneralization leads to better performance than overgeneralization.

Estimating the best possible alignment between λ_0 and λ_1 to produce the lowest regret revealed that underestimating λ_0 by an average of about 0.21 produces the best regret over all scenarios. These simulation results provide strong evidence that the systematically lower estimates of λ captured by our model comparison procedure do not necessarily suggest a flaw or bias in human behaviour—but instead—can sometimes lead to better performance. Undergeneralization, as it turns out, might not be a bug but rather a feature of human behaviour.

Further Behavioural Analysis

Locality of sampling behaviour

Figure S5 shows the locality of participants' sampling behaviour compared to a random baseline. Locality is assessed by the euclidean distance between two consecutively sampled points. This distance is compared between participants' sampling behaviour and a fully random sampler. Whereas participants sample more locally than a random sampler in both the univariate Experiment 1 ($t(160) = 31.2$, $p < .0001$) and the bivariate Experiment 2 ($t(158) = 42.7$, $p < .0001$), this locality effect is much stronger for the latter ($d = 4.47$ vs. $d = 1.92$), in line with our finding that localized models generate better predictions in Experiment 2.

Learning over trials and rounds

Next, we assess whether participants are more strongly improving over trials or over rounds (Fig. S6). If they are improving 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

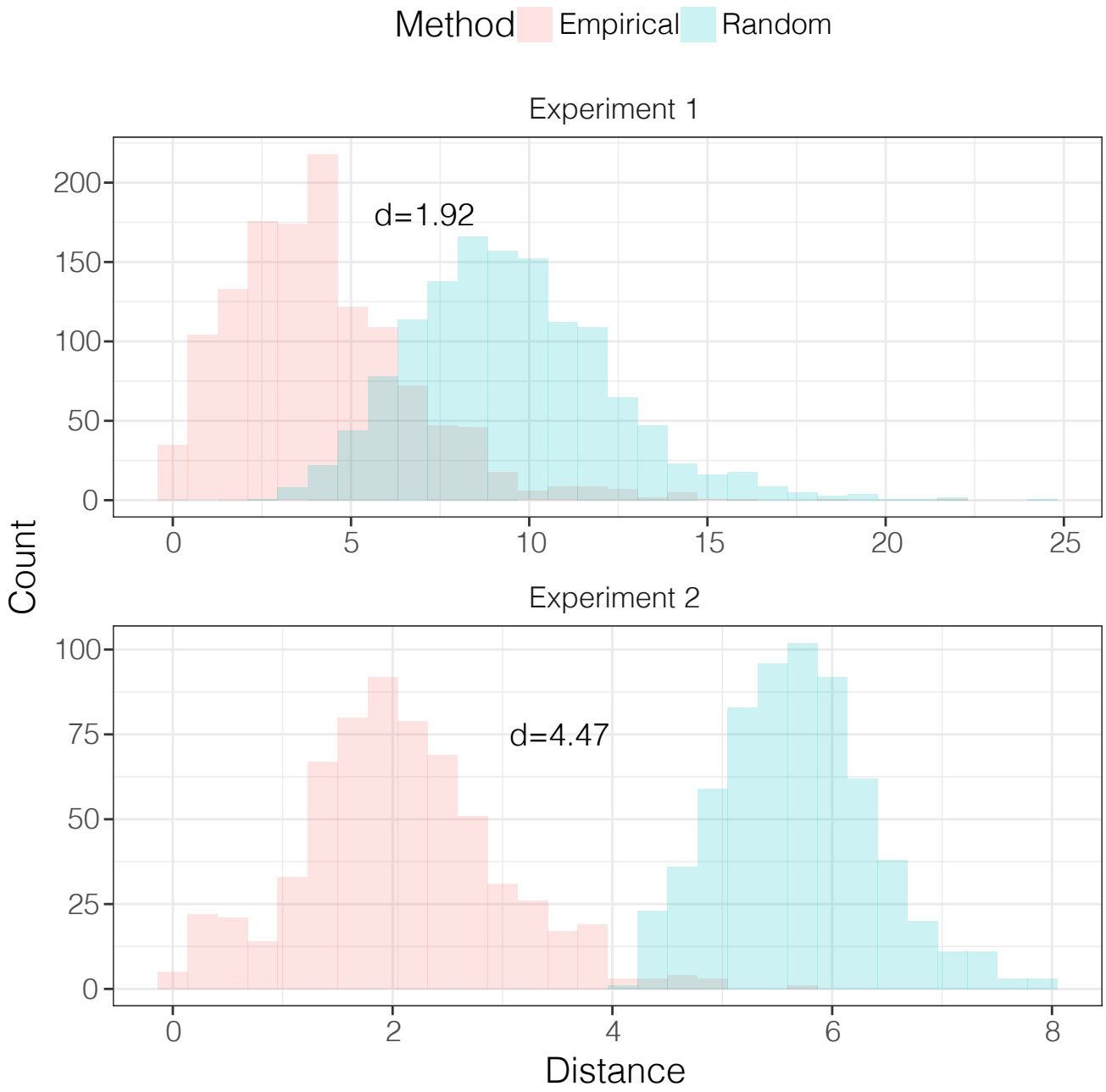

Figure S5. Locality of empirically observed and random sampling behaviour. Whereas participants consistently sample more locally than what would be expected by chance, this effect is much stronger for the bivariate Experiment 2 (d=4.47) than for the univariate Experiment 1 (d=1.92).

errors[‡]. Results (from one-sample t-tests with $\mu_0 = 0$) show that participants' scores improve significantly over trials for both Experiment 1 ($t(80) = 5.57$, $p < .0001$, $d = 0.62$) and Experiment 2 ($t(79) = 2.78$, $p < .001$, $d = 0.31$). There were no effects for rounds in either Experiment 1 ($t(80) = -2.7$, $p > .9$, $d = -0.3$) or Experiment 2 ($t(79) = 0.21$, $p > 0.4$, $d = 0.02$).

[‡]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

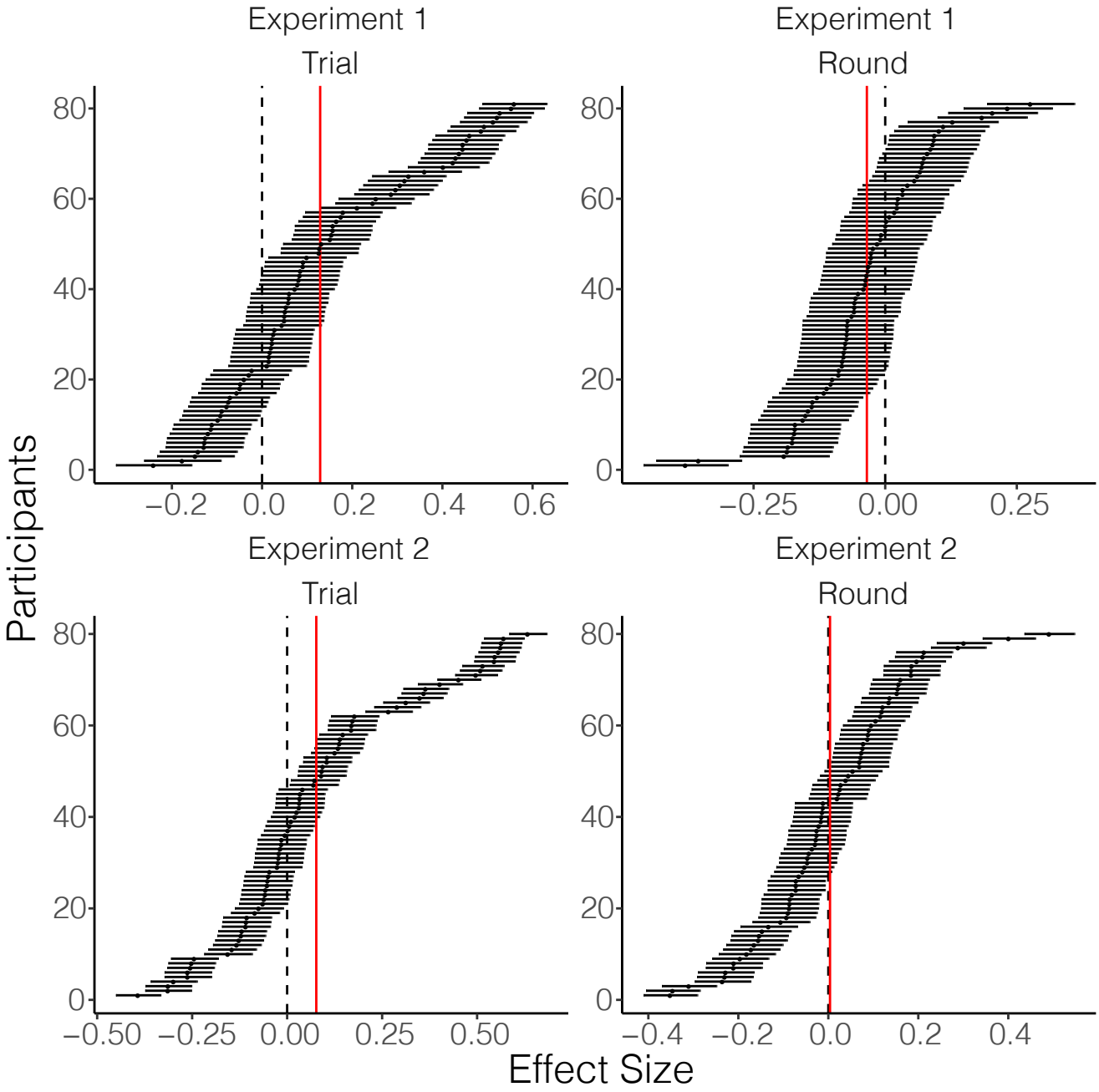

Figure S6. Average correlational effect size of trial and round on score per participant as assessed by a standardized linear regression. Participants are ordered by effect size in decreasing order. Dashed lines indicate no effect. Red lines indicate average effect size. Whereas participants consistently improve over trials, there is no effect over rounds.

Table S1. Modelling 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 ($\sqrt{\theta_\varepsilon^2}$)	Softmax Temperature (τ)	Predictive Accuracy	Participants Best Described	Length Scale (λ)	Exploration Bonus (β)	Error Variance ($\sqrt{\theta_\varepsilon^2}$)	Softmax Temperature (τ)
Mean Tracker												
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
Mean Tracker*												
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
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 mean over all participants. There were 81 participants in Experiment 1 and 80 participants in Experiment 2. We have highlighted the best performing model for each experiment in boldface.