Inference and search on graph-structured spaces

Charley M. Wu
Harvard University

Eric Schulz
Max Planck Institute for Biological Cybernetics

Samuel J Gershman
Harvard University and Center for Brains, Minds and Machines

How do people learn functions on structured spaces? And how do they use this knowledge to guide their search for rewards in situations where the number of options is large? We study human behavior on structures with graph-correlated values and propose a Bayesian model of function learning to describe and predict their behavior. Across two experiments, one assessing function learning and one assessing the search for rewards, we find that our model captures human predictions and sampling behavior better than several alternatives, generates human-like learning curves, and also captures participants' confidence judgements. Our results extend past models of human function learning to more complex, graph-structured domains.

Keywords: Function learning, Generalization, Inference, Graphs, Exploration-Exploitation

Introduction

On September 15th, 1835, Charles Darwin and the crew of the HMS Beagle arrived in the Galapagos Islands. As part of a five-year journey to study plants and animals along the coast of South America, Darwin collected specimens of Galapagos finches, which would become an important keystone for his theory of evolution. Back in England, Darwin began to study the geographical distribution of the birds, particularly the relationship between their features and their habitat. He noticed that while finches on nearby islands had similar beaks (e.g., the vegetarian tree finches and the large insectivorous tree finches with their broad and stout beaks), finches on more distant islands were more dissimilar (e.g., the cactus ground finch with its long and spike-like beak). From these observations, Darwin concluded that these finches all originally derived from the same finch and then gradually adapted to the conditions of the Islands. Since nearby islands had similar conditions, finches on these islands had more similar beaks.

Darwin's historical insight is an example of function learning, where a function represents a mapping from some input space to some output space. In Darwin's case, the hypothesis was a function mapping a bird's habitat to the characteristics of its beak (e.g., size). Function learning has traditionally been studied with continuous input spaces, but functions can also be defined over discrete input spaces such as graphs. While the geography of habitats can sometimes be described by a Cartesian coordinate system (latitude and longitude), the Galapagos is structured as a chain of islands, where the Euclidean distance within an island can be larger than the distance between islands. Since finches from the same island tend to be similar, the relevant metric for function learning may be topological rather than Euclidean distance, where the chain of islands can be described as a graph.

Function learning on graph-structured inputs spaces is not restricted to scientific epiphanies; it also applies ubiquitously to daily life. For example, the spread of disease, ideas, and cultural products from interpersonal contact can be understood as functions defined over social graphs. We can learn to predict which of our friends will like a piece of music after observing the music preferences of other friends in our social network. Similarly, as many parents of toddlers know, the appearance of a sickness in daycare is highly predictive of who will get sick next. Beyond social graphs, the flow of individuals in a transportation network and the distribution of food resources in patchy environments can likewise be described using graph-structured functions.

Despite the ubiquity of graph-structured functions, most studies of function learning (as we review below) have examined only continuous input spaces. In addition, reinforcement learning in discrete state spaces can also be interpreted as a form of graph-structured function learning, but relatively

Charley M. Wu, Harvard University; Eric Schulz, Max Planck Institute for Biological Cybernetics; Samuel J Gershman, Harvard University and the Center for Brains, Minds and Machines. This work was supported by the Dean's Fund for Competitive Research at Harvard University, the Center for Brains, Minds, and Machines (CBMM), funded by NSF STC award CCF-1231216, the Office of Naval Research (N00014-17-1-2984), and the Alfred P. Sloan Foundation.

Correspondence concerning this article should be addressed to Charley M. Wu, E-mail: charleywu@fas.harvard.edu

little work has examined patterns of generalization beyond very simple graph structures (e.g., Gershman & Niv, 2015; Wimmer, Daw, & Shohamy, 2012).

In this paper, we investigate how people learn graph-structured functions and use this knowledge to guide the search for rewards. In Experiment 1, we study how people infer the values of nodes on complex graphs (corresponding to the number of passengers on a virtual subway map), where values are correlated by the connectivity structure, such that connected nodes have similar values. In Experiment 2, we study how people search for rewards on complex graphs, tantamount to a 64-armed bandit problem, where each arm of the bandit corresponds to a node on a graph and rewards are similarly correlated based on connectivity.

Our results indicate that people learn and search for rewards consistent with a Bayesian model of function learning (Gaussian Process regression), which outperformed various alternative models in predicting inferences, sampling decisions, and uncertainty judgements. This model builds on past studies using Gaussian Process regression to describe human function learning on continuous spaces (Lucas, Griffiths, Williams, & Kalish, 2015; Schulz, Tenenbaum, Duvenaud, Speekenbrink, & Gershman, 2017), but using a prior over functions designed for discrete spaces (Kondor & Lafferty, 2002). Not only do we find strong empirical evidence for our model, but it also provides deep theoretical connections to past research on human function learning, sample-efficient exploration, and classic theories of generalization and learning.

Function learning in continuous spaces

Research on human function learning was originally pioneered by Carroll (1963), who studied how participants learned to predict the length of a line (output) based on the horizontal position of a "V" shaped marking (input). Unknown to participants, the relationship between the inputs and outputs were governed by either a positive linear, a quadratic, or a random function. Carroll's (1963) study was motivated by the goal of showing that people could extrapolate functions to generate novel predictions about outcomes that had never before been observed. In contrast to classical theories of generalization (Shepard, Hovland, & Jenkins, 1961), Carroll's work provided evidence for a mechanism of generalization that went beyond merely predicting the same outcome as that of the most similar previous experiences. Aside from showing that function learning was an important feature of human inference, Carroll (1963) also discovered that some functions, such as linear ones, were easier to learn than others, such as nonlinear ones. Subsequent studies of human function learning built on Carroll's initial insight and further investigated which types of functions were more difficult to learn (Brehmer, 1974; Busemeyer, Byun, DeLosh, & McDaniel, 1997; Koh & Meyer, 1991), finding that linear functions with positive slopes are the most learnable, and that both nonlinear functions and linear functions with negative slopes are more difficult to learn.

A problem with many of these early studies was the inflexibility of their models. Likely inspired by timely advances in statistical methods of least-square estimation, they assumed that participants used a specific parametric model, for example linear regression, and then learned by optimizing the parameters to explain the data. Yet the parametric classes of function used in these studies were insufficiently flexible to account for human function learning. Instead of only adapting a specific class of functions to a particular set of observations, people seem to adapt the model itself when encountering novel data. Brehmer (1976) tried to explain some of these effects with a sequential hypothesis testing model of functional rule learning, according to which participants adapt the complexity of their model by performing sequential hypothesis tests and pivoting between parametric forms if necessary. However, this model still required a predetermined set of parametric rules that could be compared, such that it is not able to explain the ability to learn almost any function given enough data. Thus, these earlier, *rule-based* models of human function learning could not easily explain the full range of human function learning abilities; more flexible models were needed.

To overcome the weaknesses of rule-based models of human function learning, several researcher proposed a novel class of *similarity-based* models of function learning. These models operated under the assumptions that similar input points will produce similar outputs and used neural networks to model behavior (McClelland, Rumelhart, Group, et al., 1986). These models could not only theoretically learn nearly any function, they were also able to capture the effect that linear functions are easier to learn than non-linear functions.

An important distinction in the literature on function learning (and machine learning more generally) is between interpolation (i.e., predictions for points nested between training examples) and extrapolation (i.e., predictions outside the convex hull of training inputs). Whereas similarity-based models can explain order-of-difficulty effects in interpolation tasks, they have trouble explaining how people extrapolate. Specifically, people tend to make linear predictions with a positive slope and an intercept of zero when extrapolating functions (Kwantes & Neal, 2006). This linearity bias holds true even when the underlying function is non-linear; for example, when trained on a quadratic function, average predictions fall between the true function and straight lines fit to the closest training points (Kalish, Lewandowsky, & Kruschke, 2004).

Since traditional similarity-based models of function learning could not easily explain these extrapolation patterns, the class of function learning models had to be extended

even further. This led to the development of so-called *hybrid* models of function learning, which contain an associative learning process that acts on explicitly-represented rules. One such hybrid model is the Extrapolation-Association Model (DeLosh, Busemeyer, & McDaniel, 1997), which uses similarity-based interpolation, but extrapolates using a simple linear rule. The model effectively captured the human bias towards linearity, and could predict human extrapolations for a variety of functions, but without accounting for non-linear extrapolation (Bott & Heit, 2004).

More recently, another class of function learning models was developed based on the principles of Bayesian inference. These models use a prior over functions, modeled by a Gaussian Process (GP Rasmussen & Williams, 2006), and then infer a posterior given the observed data points. Importantly, GP regression is a non-parametric model (Gershman & Blei, 2012; Schulz, Speekenbrink, & Krause, 2018), meaning that it adapts its complexity to the encountered data. Griffiths, Lucas, Williams, and Kalish (2009) and Lucas et al. (2015) were the first to show that GP regression provides a rational model of human function learning, and that it replicates most of the observed empirical phenomena of human function learning. Importantly, GP regression performs posterior inference in a way that can be understood as both similarity-based (because the kernel provides a similarity metric between data points) and rule-based (because the kernel can be expressed as a linear weighted sum), providing a further unification of rule-based and similarity-based theories (Lucas et al., 2015).

Using function learning to guide search

Learning a function is not only useful for making explicit generalizations about novel situations, but can also be used to guide adaptive behavior by leveraging functional structure to predict unobserved rewards in the environment. For example, in reinforcement learning tasks where options had inversely correlated rewards (Wimmer et al., 2012) or with rewards structured as a linear function (i.e., linearly increasing rewards from option 1 to option N; Schulz et al., 2019), participants were able to rapidly learn this structure and leverage it to facilitate better performance, even without having been explicitly told about the underlying structure.

In tasks with a large number of options, it becomes important to be able to learn efficiently, for instance by using features of the task to predict rewards (Farashahi, Rowe, Aslami, Lee, & Soltani, 2017; Radulescu, Niv, & Ballard, 2019). One approach is to learn an implicit value function mapping features onto rewards (Schulz, Konstantinidis, & Speekenbrink, 2017), which can be used to guide efficient exploration even in infinitely large problem spaces. Previous work has successfully used a GP model of function learning to predict human search behavior in a variety of both spatially and conceptually correlated reward environments (Schulz, Wu, Huys, Krause, & Speekenbrink, 2018; Wu, Schulz, Garvert, Meder, & Schuck, 2020; Wu, Schulz, Speekenbrink, Nelson, & Meder, 2018), where the number of options vastly outnumbered the sampling horizon.

In transitioning from a pure function learning paradigm to a reward learning paradigm, the demands of the task change from pure information maximization to a balance between exploration and exploitation (Cohen, McClure, & Yu, 2007; Mehlhorn et al., 2015; Schulz & Gershman, 2019). Typically studied in multi-armed bandit tasks, the exploration-exploitation dilemma requires an agent to trade off between sampling novel options to acquire potentially useful information about the structure of rewards (exploration) with sampling options known to have high-value payoffs (exploitation). Not enough exploration, and the agent could get stuck in a local optima, while not enough exploration and the agent never reaps the rewards they have discovered.

Since optimal solutions in such tasks are intractable for all but the most simplistic scenarios (Whittle, 1980), a variety of heuristic algorithms are commonly used. One such algorithm is upper confidence bound sampling, which adds an "uncertainty bonus" to each option's value (Auer, 2002). Since this corresponds to a weighted sum of the expected reward and its uncertainty, this algorithm explicitly encodes the trade-off between exploration and exploitation. Although earlier studies produced mixed evidence for an uncertainty bonus in human decision making (Daw, O'doherty, Dayan, Seymour, & Dolan, 2006; Payzan-LeNestour & Bossaerts, 2011), many recent studies have shown that humans do engage in uncertainty-guided exploration (Gershman, 2018a, 2019; Knox, Otto, Stone, & Love, 2012; Speekenbrink & Konstantinidis, 2015; Wilson, Geana, White, Ludvig, & Cohen, 2014; Wu, Schulz, Speekenbrink, et al., 2018).

A key component for performing uncertainty-guided exploration is being able to estimate the uncertainty of one's predictions. Since GP regression is a Bayesian model of function learning, uncertainty is quantified by the posterior distribution. In contrast, a model that makes only point estimates of expected reward does not have access to uncertainty-guided exploration. Instead, less efficient random exploration strategies must be used (e.g., softmax exploration). A combined model of GP regression with upper confidence sampling has proved to be an effective model in a wide number of contexts, describing how people explore different food options based on real world data (Schulz et al., 2019), predicting whether or not to people will try out novel options (Stojic, Schulz, Analytis, & Speekenbrink, 2018), and explaining developmental differences between how children and adults search for rewards (Schulz, Wu, Ruggeri, & Meder, 2018).

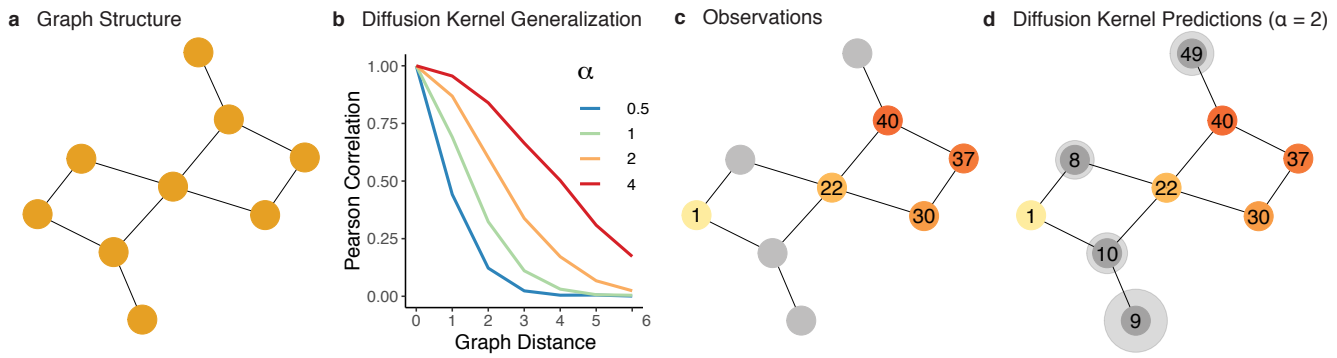

Figure 1. Function learning in graph-structured input spaces. **a**) An example of a graph structure, where nodes represent states and edges indicate the transition structure. **b**) A graph kernel models the covariance between function values at different nodes on the graph, and can be intuitively understood as a similarity metric between nodes. Here, we use a diffusion kernel that assumes function values diffuse across the graph according to a random walk. The correlation between function values at two nodes (normalized covariance) decays monotonically as a function of graph distance. The diffusion parameter α governs the rate of decay. **c**) Given some observations on the graph (colored nodes), we can use the diffusion kernel combined with the GP framework to make probabilistic predictions (**d**) about expected function values. The posterior mean is represented by numbers in the grey nodes, and the posterior uncertainty (variance) is represented by the size of the halo.

Function learning in graph-structured spaces

In the current work, we examine whether principles of function learning can be used to model human inference and search for rewards in graph-structured spaces (see Figure 1a). Studying these environments expands the scope of classical function learning models considerably, addressing an important gap in our understanding of function learning: the real world contains many graph-structured functions, as the examples in our introduction illustrated.

In what follows, we will first introduce the GP regression framework, and then specialize it to the problem of function learning on graphs. The key mathematical tool that we employ is the diffusion kernel (Kondor & Lafferty, 2002), which offers one of the simplest ways to define priors over functions on graphs. We will show how the diffusion kernel naturally connects to past models of human function learning. We will then put this model to an empirical test, presenting two experiments studying how people make inferences and search for rewards on graph structures. In Experiment 1, participants were shown a series of artificially generated subway maps and asked to predict the number of passengers at unobserved stations. In Experiment 2, participants played a graph-structured multi-arm bandit task, where arms correspond to nodes in the graph, and the payoffs are correlated via the connectivity structure.

Gaussian Process regression

A GP (Rasmussen & Williams, 2006) defines a distribution over functions $f : \mathcal{S} \rightarrow \mathbb{R}^n$ that map the input space $\mathcal{S}$ to real-valued scalar outputs (e.g., rewards):

$$f \sim \mathcal{GP}\left(m(s), k(s, s')\right), \tag{1}$$

where $m(s) = \mathbb{E}[f(s)]$ is a mean function specifying the expected output for s, and $k(s, s') = \mathbb{E}[(f(s) - m(s))(f(s') - m(s'))]$ is the kernel function defining the covariance between outputs for a given input pair. We follow the convention of setting the mean function to zero, such that the GP prior is fully defined by the kernel. In the next section, we will define a kernel specialized for graph-structured functions.

We model a scenario in which an observer measures $y = f(s) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is noise added to the output value. Given a data set of N input-output pairs, $\mathcal{D} = \{\mathbf{s}, \mathbf{y}\}$, the posterior predictive distribution $p(f(s_*)|\mathcal{D})$ for any target state s_* is a Gaussian distribution with mean m_* and variance v_*:

$$m_* = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \tag{2}$$

$$v_* = k(s_*, s_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_* \tag{3}$$

where $\mathbf{K}$ is the $N \times N$ covariance matrix evaluated at each pair of observed inputs, and $\mathbf{k}_* = [k(s_1, s_*), \ldots, k(s_N, s_*)]$ is the covariance between each observed input and the target input s_*. Thus, for any node in the graph, we can make Bayesian predictions about the expected output and our uncertainty (Fig. 1d).

As pointed out by Lucas et al. (2015), we can draw a connection between GP regression and similarity-based models of function learning. In particular, the posterior predictive mean can be expressed as:

$$m_* = \sum_{n=1}^{N} w_n k(s_n, s_*), \tag{4}$$

where each s_n is a previously observed input, and the weights are given by $\mathbf{w} = \left[\mathbf{K} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y}$. Intuitively, this means that GP regression is equivalent to a linearly-weighted sum of similarities between the target input and the observed input (Schulz, Speekenbrink, & Krause, 2018).

The diffusion kernel

We now introduce a kernel function that is specialized for graph-structured input spaces. A graph $G = (\mathcal{S}, \mathcal{E})$ consists of nodes $s \in \mathcal{S}$ and edges $e \in \mathcal{E}$ (Fig. 1a). As a concrete example, in our subway task, nodes correspond to stations and edges correspond to subway lines. For now, we assume that all edges are undirected, so that probabilistic dependencies between any two adjacent nodes are symmetric.

The diffusion kernel (DF; Kondor & Lafferty, 2002) defines a similarity metric $k(s, s')$ between any two nodes based on the matrix exponentiation of the graph Laplacian:

$$\mathbf{K} = \exp(-\alpha\mathbf{L}), \tag{5}$$

where the graph Laplacian $\mathbf{L}$ is defined by:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{6}$$

with adjacency matrix $\mathbf{A}$ and degree $\mathbf{D}$. Each element A_{ij} is 1 when nodes i and j are connected, and 0 otherwise, while the diagonals of D describe the number of connections of each node (off-diagonals are all 0)[1]. In practice, the matrix exponentiation in Eq. 5 can be computed by first decomposing $\mathbf{L}$ into its eigenvectors $\{u_i\}$ and eigenvalues $\{\lambda_i\}$, and then substituting matrix exponentiation with real exponentiation using $\mathbf{K} = \sum_i e^{\alpha\lambda_i}\mathbf{u}_i\mathbf{u}_i^\top$.

Intuitively, the diffusion kernel assumes that output values diffuse along the graph similar to a heat diffusion process. Thus, closely connected nodes will tend to have similar output values. The free parameter α models the rate of diffusion, where $\alpha \to 0$ assumes complete independence between nodes, and $\alpha \to \infty$ assumes all nodes are perfectly correlated.

Connecting spatial and structured generalization

The GP framework allows us to relate similarity-based generalization on graphs to theories of generalization in continuous domains. Consider the case of an infinitely fine lattice graph (i.e., a grid-like graph with equal connections for every node and with the number of nodes and connections approaching infinity). Following Kondor and Lafferty (2002) and using the diffusion kernel defined by Eq. 5, this limit can be expressed as

$$k(s, s') = \frac{1}{\sqrt{(4\pi\alpha)}}\exp\left(\frac{-|s - s'|^2}{4\alpha}\right), \tag{7}$$

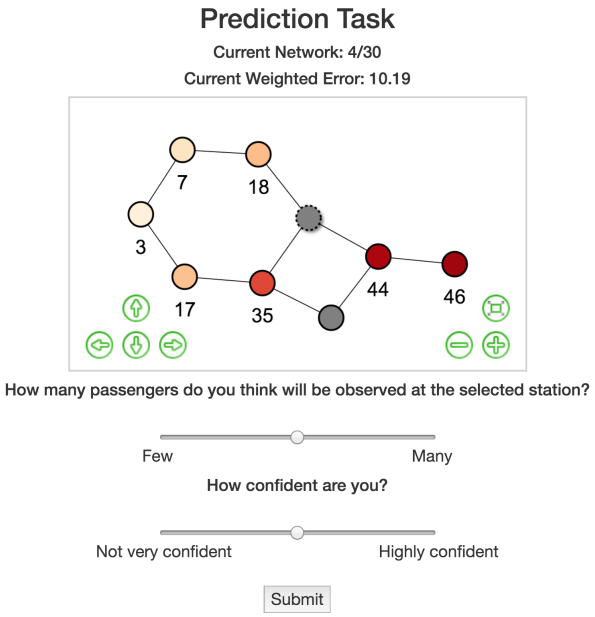

Figure 2. Screenshot from Experiment 1. Observed nodes (3, 5, or 7 depending on the information condition) are shown with a numerical value and a corresponding color aid (darker indicates larger values). The target node is indicated by the dashed line, and dynamically changed color/value as participants moved the top slider. Confidence judgments were used to compute a weighted error (i.e., more confident answers having a larger contribution), which was used to determine the performance-contingent bonus.

which is equivalent to the Radial Basis Function (RBF) kernel that has been used in past research on human function learning and search on regular grids (Wu, Schulz, Speekenbrink, et al., 2018). Thus, the RBF kernel can be understood as a special case of the diffusion kernel, offering a connection to theories of function learning over spatial and conceptual features (Wu, Schulz, Garvert, Meder, & Schuck, 2018; Wu et al., 2020).

Experiment 1: Subway prediction task

In our first experiment, participants were shown various graph structures described as subway maps (Fig. 2), and were asked to make predictions about unobserved nodes. For each prediction, participants also gave confidence judgments, which we use as an estimate of their (inverse) uncertainty. We used a GP parameterized with a diffusion kernel as a model of function learning in this task, which we compared

[1]The graph Laplacian can also describe graphs with weighted edges, where we substitute the weighted adjacency matrix $\mathbf{W}$ for $\mathbf{A}$ and the degree matrix describes the weighted degree of each node. All analyses apply in the weighted case.

to several alternative models.

Methods

Participants. We recruited 100 participants (M_{age} = 32.7; SD = 8.4; 28 female) on Amazon Mechanical Turk (requiring a 95% approval rate and 100 previously completed HITs) to perform 30 rounds of a graph prediction task. The experiment was approved by the Harvard Institutional Review Board (IRB15-2048).

Procedure. On each graph, numerical information was provided about the number of passengers at 3, 5, or 7 other stations (along with a color aid), from which participants were asked to predict the number of passengers at a target station and provide a confidence judgment (Likert scale from 1 to 11). The subway passenger cover story was used to provide intuitions about graph correlated functions, similar to our example from the introduction. Additionally, participants observed 10 fully revealed graphs to familiarize themselves with the task and completed a comprehension check before starting the task.

Participants were paid a base fee of $2.00 USD for participation with an additional performance contingent bonus of up to $3.00 USD. The bonus payment was based on the mean absolute judgement error weighted by confidence judgments: $R_{bonus} = \$3.00 \times (25 - \sum_i \tilde{c}_i \epsilon_i)/25$ where $\tilde{c}_i$ is the normalized confidence judgment $\tilde{c}_i = \frac{c_i}{\sum c_j}$ and ϵ_i is the absolute error for judgment i. On average, participants completed the task in 8.09 minutes (SD = 3.7) and earned $3.87 USD ($SD$ = $0.33).

All participants observed the same set of 40 graphs that were sampled without replacement for the 10 fully revealed examples in the familiarization phase and for the 30 graphs in the prediction task. We generated the set of 40 graphs by iteratively building 3×3 lattice graphs (also known as mesh or grid graphs), and then randomly pruning 2 out of the 12 edges. In order to generate the functions (i.e., number of passengers), we sampled a single function from a GP prior over the graph, where the diffusion parameter was set to $\alpha = 2$.

Modeling. We compared the predictive performance of the GP with two heuristic models that use a nearest-neighbors averaging rule (see below). Models were fit to each individual participant by using leave-one-round-out cross-validation to iteratively compute the maximum likelihood estimates on a test set, and then make out-of-sample predictions on the held-out round. We repeated this procedure for all rounds and compared the predictive performance[2] over all held-out rounds.

The two heuristic strategies for function learning on graphs make predictions about the output values of a target state s_* based on a simple nearest neighbors averaging rule. The *k-Nearest Neighbors* (kNN) strategy averages the values of the k nearest nodes (including all nodes with same shortest path distance as the k-th nearest), while the *d-Nearest Neighbors* (dNN) strategy averages the values of all nodes within path distance d. Both kNN and dNN default to a prediction of 25 when the set of neighbors are empty (i.e., the median value in the experiment).

Both the dNN and kNN heuristics approximate the local structure of a graph with the intuition that nearby states have similar output values. While they sometimes make the same predictions as the GP model while having lower computational demands, they fail to capture the full connectivity structure of the graph. Thus, they are unable to learn directional trends (e.g., increasing function values from one end of the graph to the other) or asymmetric influences (e.g., a central hub exerting relatively larger influence on sparsely connected neighbors). Additionally, they only make point-estimate predictions, and thus do not capture a the underlying uncertainty of a prediction (which we use to model confidence judgments).

Results and discussion

All code and data necessary to replicate the analyses in this manuscript are publicly available at `https://github.com/charleywu/graphInference`. Figure 3 shows the behavioral and model-based results of the experiment. We applied Bayesian mixed-effects regression to estimate the effect of the number of observed nodes on participant prediction errors, with participants as a random effect (see Table A1 for details). Participants made systematically lower errors in their predictions as the number of observations increased ($b_{\text{numNodes}} = -0.60$, 95% HPD: $[-0.79, -0.41]$, $BF_{10} = 1.1 \times 10^7$; Table A1; Fig. 3a). Repeating the same analysis but using participant confidence judgments as the dependent variable, we found that confidence increased with the number of available observations ($b_{\text{numNodes}} = 0.23$, 95% HPD: $[0.17, 0.30]$, $BF_{10} = 4.7 \times 10^8$; Table A1; Fig. 3b). Finally, participants were also able to calibrate confidence judgments to the accuracy of their predictions, with higher confidence predictions having lower error ($b_{\text{confidence}} = -0.66$, 95% HPD: $[-0.83, -0.49]$, $BF_{10} = 4.0 \times 10^8$; Table A1; Fig. 3c). We found no effect of round number on prediction error ($b_{\text{round}} = 0.01$, 95% HPD: $[0.02, -0.03]$, $BF_{10} = 0.06$), suggesting that the familiarization phase and cover story were sufficient for providing intuitions about graph correlated structures.

Figure 3d shows the model comparison results. We evaluated the relative performance of models using the protected exceedence probability (pxp), as a Bayesian estimate

[2]The predictive performance is defined in terms of log likelihood of the out-of-sample predictions, which is a monotonic transformation of the mean squared prediction error by assuming a Gaussian probability density: $\log \mathcal{L} = \sum_i \frac{-(f(x_i)-x_i)^2}{2\sigma_\epsilon^2} - \log(\frac{1}{\sigma_\epsilon \sqrt{2\pi}})$, where each x_i are the participant judgments, each $f(x_i)$ are the out-of-sample model predictions, and we set $\sigma_\epsilon = 1$.

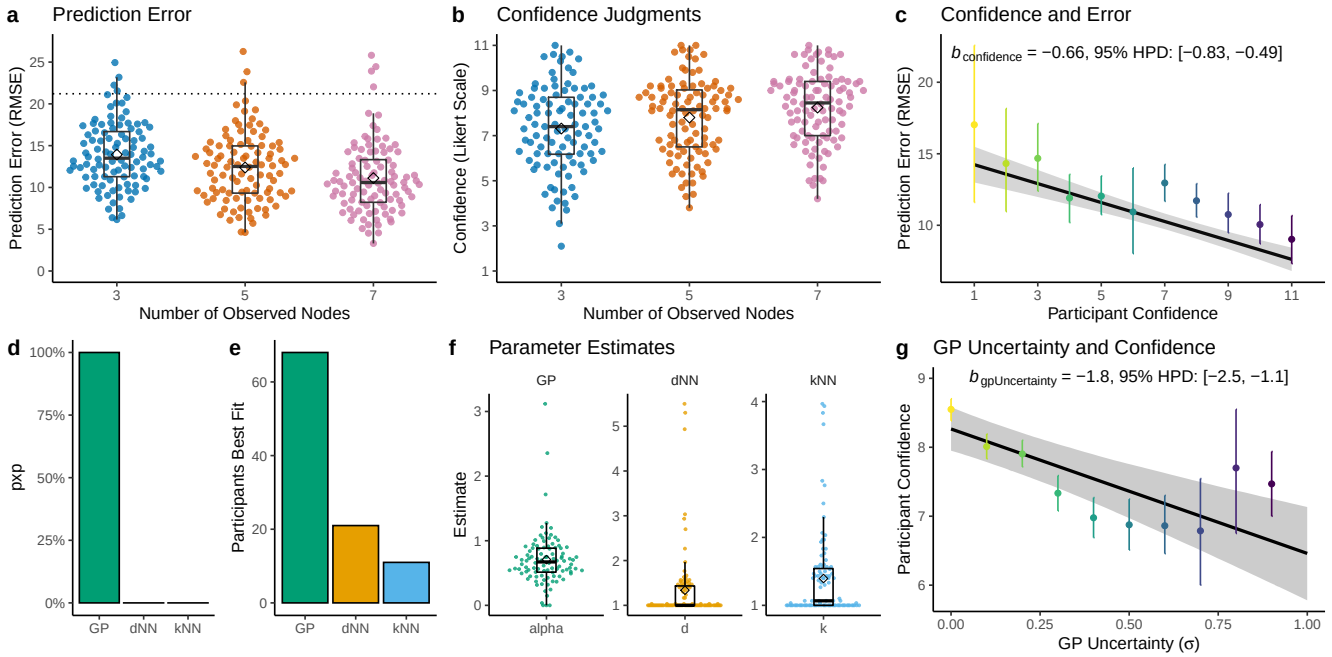

Figure 3. Experiment 1 results. **a-b**) Participant judgment errors and confidence estimates. Each dot is a single participant (averaged over each number of observed nodes), with Tukey box plots and diamonds indicating group means. The dotted line in **a**) is a random baseline. **c**) Judgment error and confidence. Each colored dot is the aggregate mean with error bars indicating the 95% CI. The black line is the group-level effect of a Bayesian mixed model (Table A1), indicating the posterior mean and 95% CI (ribbon). **d**) Hierarchical Bayesian model comparison between the GP with diffusion kernel, d-nearest neighbors (dNN), and k-nearest neighbors (kNN). The bars indicate the protected exceedence probability (pxp) as an estimate of the posterior probability of a given model being the most frequent in the population (corrected for chance). **e**) The number of participants best fit by each model. **f**) Parameter estimates, where each dot is the mean cross-validated estimate for each participant, with Tukey box plots and diamonds indicating group means. **g**) The inverse relationship between GP uncertainty estimates (σ) and participant confidence judgments (Likert scale), where the colored dot and error bars indicate (respectively) the aggregate mean ± 95% CI computed at 10 equally spaced intervals along the x-axis. The black line is the fixed-effect of a Bayesian mixed model (Table A1), with the ribbon indicating the 95% CI.

of the probability that a particular model is more frequent in the population than all the other models under consideration, corrected for chance (Rigoux, Stephan, Friston, & Daunizeau, 2014; Stephan, Penny, Daunizeau, Moran, & Friston, 2009). The GP with diffusion kernel was overwhelmingly the best model, with $pxp(GP) \approx 1$. Overall, 68 out of 100 participants were best predicted by the GP, 21 by the dNN, and 11 by the kNN (Fig. 3e; see Fig. B1 for additional comparisons between model predictions and participant judgments).

Figure 3f shows individual parameter estimates of each model. The estimated diffusion parameter α was substantially lower than the ground truth of $\alpha = 2$ ($t(99) = -31.3$, $p < .001$, $d = 3.1$, $BF_{10} = 4.4 \times 10^{29}$)[3], replicating previous findings that have shown undergeneralization to be a prominent feature of human behavior (Wu, Schulz, Speekenbrink, et al., 2018). Estimates for d and k were highly clustered around the lower limit of 1, suggesting that averaging over larger portions of the graph were not consistent with participant predictions.

Lastly, an advantage of the GP is that it produces Bayesian uncertainty estimates for each prediction. While the dNN and kNN models make no predictions about confidence, the GP's uncertainty estimates correspond to participant confidence judgments, which we validated using a Bayesian mixed model regressing the uncertainty estimates of the GP onto participant confidence judgments ($b_{gpUncertainty} = -1.8$, 95% HPD: $[-2.5, -1.1]$, $BF_{10} = 1.2 \times 10^5$; Table A1, Fig. 3g).

The results of this experiment demonstrate that a GP with a diffusion kernel can successfully model human function learning on graphs, in particular the empirical pattern of predictions and confidence ratings. Our model extends existing

[3]Previous results reported in Wu, Schulz, and Gershman (2019) suffered from numerical instability during matrix exponentiation when computing the diffusion kernel, and thus yielded different estimates.

theories of human function learning in continuous spaces, where the RBF kernel (commonly used in continuous domains) can be seen as a special limiting case of the diffusion kernel.

Experiment 2: Graph bandit

In our next experiment, we tested the suitability of the diffusion kernel as a model of search, using a multi-armed bandit task with structured rewards (see also Wu, Schulz, Speekenbrink, et al., 2018). In particular, extending our previous work on spatially and conceptually correlated multi-armed bandits (Wu et al., 2020; Wu, Schulz, Speekenbrink, et al., 2018), we constructed a task where rewards were defined by the connectivity structure of a graph (Fig. 4). In this task, participants searched for rewards by clicking nodes on a graph. As in Experiment 1, the output values (rewards) were generated by a function drawn from a GP with a diffusion kernel. This induced a graph-correlated reward structure, allowing for similarity-based generalization to aid in search, but where similarity was defined based on connectivity rather than perceptual features or Euclidean distances between options as in our previous work.

Methods

Participants. We recruited 100 participants on Amazon Mechanical Turk (requiring 95% approval rate and 100 previously completed HITs). Two participants were excluded because of missing data, making the total sample size $N = 98$ ($M_{age} = 34.3$; $SD = 8.7$; 32 female). Participants were paid \$2.00 for completing the task and earned an additional performance contingent bonus of up to \$3.00. Overall, the task took 7.2 ± 3.3 minutes and participants earned \$4.32 ± \$0.24 USD. The experiment was approved by the Harvard Institutional Review Board (IRB15-2048).

Procedure. Participants were instructed to earn as many points as possible by clicking on the nodes of a graph. Each node represented a reward generating arm of the bandit, where connected nodes yielded similar rewards, such that across the whole graph the expected rewards were defined by a graph-correlated structure (see Fig. 4a). Along with the instructions indicating the correlated structure of rewards, participants were shown four fully revealed graphs to familiarize them with the reward structure and had to correctly answer three comprehension questions before starting the task.

After completing the comprehension questions, participants performed a search task over 10 rounds, each corresponding to a different randomly sampled graph structure. In each task, participants were initially shown a single randomly revealed node, and had 25 clicks to either explore unrevealed nodes or to reclick previously observed nodes, where each observation included normally distributed noise $\epsilon \sim \mathcal{N}(0, 1)$. Each clicked node displayed the numerical value (most recent observation if selected multiple times) and a color aid, where darker colors corresponded to larger rewards (Fig. 4b). After finishing each round, participants were informed about their performance as a percentage of the best possible score (compared to selecting the global optimum each trial). The final performance bonus (up to \$3.00) was also calculated based on this percentage, averaged over all rounds.

In total, we generated 40 different graphs by building 8x8 lattice graphs and then randomly pruning 40% of the edges, with the constraint that the resulting graph be comprised of a single connected component. We then sampled a single reward function for each graph from a GP prior, parameterized by a diffusion kernel fit on the graph (with $\alpha = 2$). The layout for each graph was pre-generated using the Fruchterman-Reingold (1991) force-directed graph placement algorithm, such that a single canonical layout for each graph was observed by all participants. For each participant, we sampled (without replacement) from the same set of 40 pre-generated graphs to build the set of 4 fully revealed graphs shown in the instructions and the 10 graphs used in the main experiment.

Prior to beginning the very last round, participants were informed that it was a "bonus round". The goal of acquiring as many points as possible remained the same, but after 20 clicks, participants were shown a series of 10 unrevealed nodes and asked to make judgments about the expected reward and their confidence (Fig. 4c). After all 10 judgments were completed, participants were forced to choose one of the 10 options, and then the task was completed as normal. Behavioral and modeling results exclude the bonus round, except for the analyses of the judgment data.

Modeling. In order to understand how participants search for rewards, we used computational modeling to make predictions about choices in the bandit task and the judgments from the bonus round. Models were fit to the bandit data (omitting the bonus round) using leave-one-round-out cross validation, where we iteratively held out a single round as a test set, and computed the maximum likelihood estimate on the remaining rounds as the training set. We compared models using the summed out-of-sample prediction accuracy on the held-out rounds. Altogether, we compared four different models corresponding to different strategies for generalization and exploration (see below).

Each model computes a value for each option $q(s)$, which is then transformed into a probability distribution using a softmax choice rule:

$$P(s_i) = \frac{\exp(q(s_i)/\tau)}{\sum_j \exp(q(s_j)/\tau)}, \tag{8}$$

where the temperature parameter τ is a free parameter controlling the level of random exploration. In addition, all models also use a stickiness parameter ω that adds a bonus onto the value of the most recently chosen option. This is a common feature of reinforcement learning models (Christakou et al., 2013; Gershman, Pesaran, & Daw, 2009) and

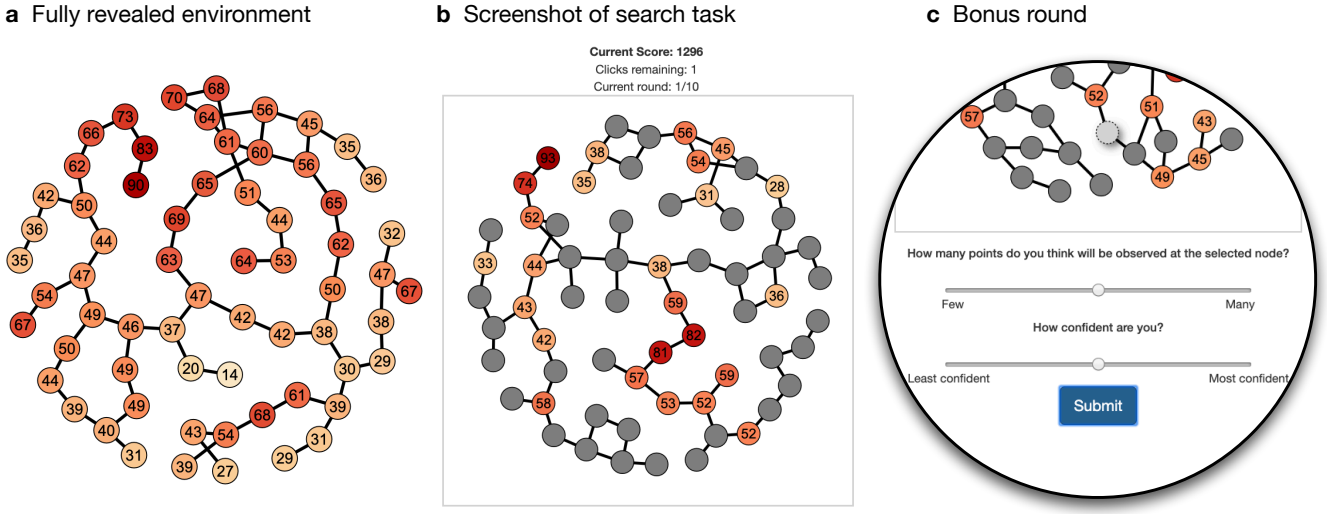

Figure 4. Experiment 2 screenshots. **a**) Four fully revealed environments were shown to participants prior to beginning the task. **b**) During the task participants were instructed to click nodes to earn as much reward as possible. Clicked nodes displayed the numeric value of the earned reward and a color guide (darker colors indicate higher rewards). **c**) Zoomed in screenshot of the bonus round, which activated after the 20th trial on the last round. 10 unclicked nodes were uniformly sampled and participants were sequentially asked to make judgments about expected rewards and report their confidence rating. The expected reward slider was mapped to the selected node, such that the color and numerical value dynamically changed as the slider was moved.

particularly in multi-armed bandit tasks (Schulz et al., 2019), which we include here to account for repeat clicks.

The GP model uses the diffusion kernel (Eq. 5) to make predictive generalizations about reward, where we fit α as a free parameter defining the extent to which generalizations diffuse along the graph structure. For each node s, the GP produces normally distributed predictions that can be summarized in terms of an expected value $m(s)$ (Eq. 2) and the underlying uncertainty $v(s)$ (Eq. 3). In order to model how participants balance between exploiting high value rewards and exploring highly uncertain options, we use upper confidence bound (UCB) sampling (Auer, 2002) to produce a valuation of each node:

$$q_{\text{UCB}}(s) = m(s) + \beta\sqrt{v(s)}, \tag{9}$$

where the exploration bonus β is a free parameter that governs the level of exploration directed towards highly uncertain options. Higher values of β correspond to more exploratory behavior, which is directed towards nodes with the highest estimated level of uncertainty.

The Bayesian mean tracker (BMT) is a prototypical reinforcement learning model that can be interpreted as a Bayesian variant of the traditional Rescorla-Wagner (1972) model (Gershman, 2015). The BMT assumes a normally distribute prior over rewards $\mathcal{N}(m_{j,0}, v_{j,0})$ computed independently for each option j, where we set the prior mean to the median value of payoffs $m_{j,0} = 50$ and the prior variance to $v_{j,0} = 500$.

Given some observations $\mathcal{D}_{t-1} = \{\mathbf{X}_{t-1}, \mathbf{y}_{t-1}\}$ of rewards $\mathbf{y}_{t-1}$ for inputs $\mathbf{X}_{t-1}$, the BMT learns the rewards of each option independently by computing independent posterior distributions for the mean μ_j for each option j

$$P(\mu_j|\mathcal{D}_{t-1}) = \mathcal{N}(m_{j,t-1}, v_{j,t-1}) \tag{10}$$

In each iteration, the posterior mean $m_{j,t}$ and variance $v_{j,t}$ for the option j chosen at time t are updated based on the prediction error:

$$m_{j,t} = m_{j,t-1} + \delta_{j,t} G_{j,t}\left[y_{j,t} - m_{j,t-1}\right] \tag{11}$$

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

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

$$G_{j,t} = \frac{v_{j,t-1}}{v_{j,t-1} + \theta^2_\epsilon} \tag{13}$$

where θ^2_ϵ is the error variance 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 $m_{j,t-1}$ (i.e., prediction error), scaled by the Kalman gain $G_{j,t}$. 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 $v_{j,t}$.

Like the GP, the BMT also used UCB sampling, along with stickiness and a softmax choice rule. Unlike the GP, the BMT does not generalize, and thus defaults to the prior mean and variance for any unobserved options. For this reason, we did not consider the BMT as a candidate model for Exp. 1. Nevertheless, it provided a sensible benchmark in the graph bandit task, since it is an optimal model for learning independent reward distributions through experience, and can support both directed and random exploration algorithms.

In addition to reinforcement learning models, we also considered both the dNN and kNN heuristics from Experiment 1. Predictions about expected reward were computed using the respective nearest neighbor averaging rule, where d and k were estimated as free parameters. For predictions where no observed nodes satisfied the averaging rule (i.e., all observations were too far away), we defaulted to an expected value of $m(s) = 50$ (median over all environments). In contrast to the GP and BMT models, these models make only point estimates about reward, and thus precluded UCB sampling. Instead, choice probabilities were calculated using only softmax choice rule on expected reward and with estimated stickiness weights.

Results and discussion

Participants performed well in the task, achieving higher rewards over successive trials ($r = .93$, $p < .001$, $BF_{10} = 4.5 \times 10^7$; Fig. 5a) and decisively outperforming a random baseline ($t(97) = 29.6$, $p < .001$, $d = 3.0$, $BF_{10} = 7.2 \times 10^{46}$). There was no influence of round number on performance ($r = .49$, $p = .182$, $BF = 1.1$), indicating that the fully revealed environments in the instructions (Fig. 4a) and comprehension questions were sufficient for conveying the goal of the task and the underlying covariance structure of rewards.

Participants adapted their search behavior as a function of reward value: higher rewards predicted a higher probability of making a repeat selection (Bayesian mixed model: Odds ratio = 1.13, 95% HPD: [1.12,1.14], $BF_{10} = 3.2 \times 10^{40}$; Table A2; Fig. 5b). We also found that higher rewards predicted shorter path distances to the subsequent selection ($b_{prevReward} = -0.11$, 95% HPD: $[-0.12, -0.10]$, $BF_{10} = 2.6 \times 10^{43}$; Fig. 3c). Thus, participants searched locally when finding high rewards, and explored further way upon finding poor rewards (see Appendix C for analyses on connectivity structure and sampling patterns). This provides early evidence that participants used generalization to guide their search for rewards, since they systematically adapted their search distance as a function of reward value, thereby avoiding regions with poor rewards and searching more locally in richer areas.

Overall, the GP was the most predictive model (Fig. 5d) with an estimated prevalence of $pxp(GP) = .86$, with the other models having $pxp(BMT) = .12$, $pxp(dNN) = .02$, and $pxp(kNN) = .001$. As a benchmark, we also fit a null model that made the same prediction for every node, which combined with stickiness and the softmax choice rule, was worse than all other models $pxp(sticky) < .001$. At the individual level, 34 out of 98 participants were best fit by the GP, 27 by the BMT, 20 by the dNN, and 17 by the dNN. Participants with higher performance on the bandit task were better predicted by the GP model ($r = -.83$, $p < .001$, $BF = 8.8 \times 10^{21}$) and also tended to be more diagnostic between the GP and BMT models (see Fig. D3), in favor of the GP.

We simulated the behavior of each model by sampling (10k samples with replacement) from the set of participant parameter estimates and computing the average learning curves (Fig. 5e). Although all models performed below the human curves, the GP achieved the closest levels of performance, with the BMT performing next best (see Appendix D for a detailed analysis of parameter estimates from each model).

To provide additional support for our modeling results, we also predicted participant judgments in the bonus round using participant parameter estimates from the bandit task. Since model parameters were estimated through cross-validation on all rounds except the bonus round, we used each participant's median parameter estimates (over rounds 1 to 9) to make out-of-task predictions about the bonus round judgments. The GP model best predicted the largest number of participants (Fig. 5f) and had the lowest prediction error on average (comparing RMSE), although there was no difference in comparison to the dNN ($t(97) = -0.1$, $p = .897$, $d = 0.01$, $BF = .11$), which had the second lowest prediction error. Looking more closely at the individual correspondence between participant judgments and model predictions (Fig. 5g), we fit separate Bayesian mixed effects regression for each model, predicting participant judgments based on model predictions (Table A3). Overall, the fits of these models for the GP, dNN, and kNN were highly similar.

While there was mixed evidence for which model best predicted judgments of expected reward, we next looked at how predictions of uncertainty corresponded to participant confidence ratings. Here, we interpreted confidence to be the inverse of uncertainty. In this analysis, we considered only the GP and BMT models, since no other models could generate uncertainty estimates. Figure 5h shows a comparison between the (per participant) rank-ordered confidence ratings and rank-ordered uncertainty estimates of the models. While the BMT estimated the same level of uncertainty for all unobserved nodes (making correlations undefined), we found that the GP uncertainty estimates corresponded well

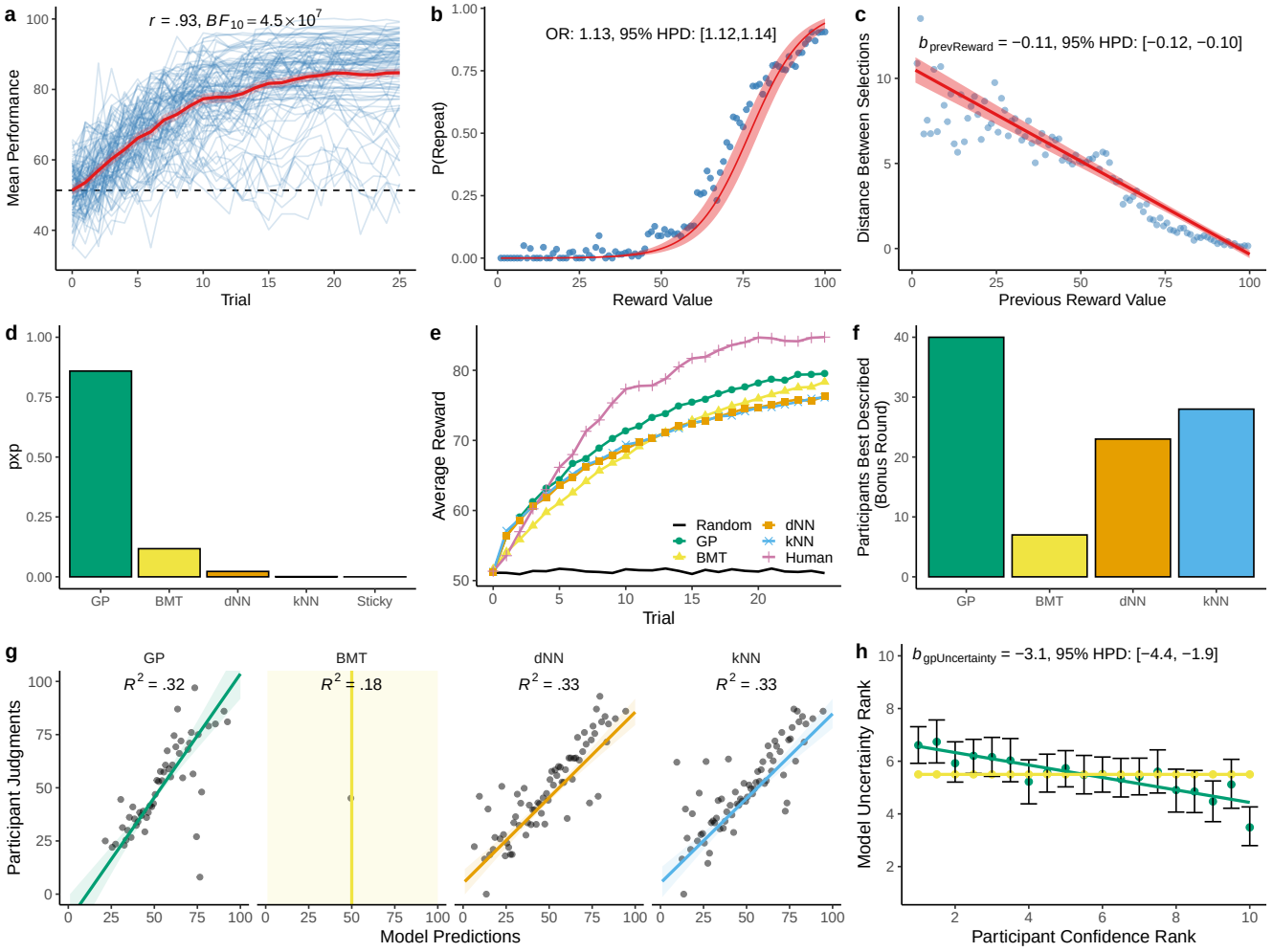

Figure 5. Experiment 2 results. **a**) Mean performance over trials, where each blue line is a single participant and the red line is the group mean (± 95% CI). The dashed line provides a comparison to a random baseline. **b**) The probability of repeating a selection as a function of the reward value. Each dot is the aggregate mean calculated at intervals of 1, while the red line is the group-effect of a Bayesian mixed effects logistic regression (Table A2), with the ribbon indicating the 95% CI. **c**) The relationship between reward value and the graph distance (shortest path) to the subsequent selection. Each dot is an aggregate mean, while the red line indicates the group-level effect of a Bayesian mixed model (Table A2). **d**) Model comparison based on out-of-sample predictive accuracy. The Y-axis shows the protected exceedance probability (pxp) describing the prevalence of each model in the population (corrected for chance). **e**) Simulated learning curves by sampling (with replacement) from participant parameter estimates (10k replications), where the black line shows a random baseline and the pink line shows mean participant performance. **f**) Participants best described on the bonus round data, where we used parameter estimates from the bandit task to make predictions of participant judgments about unobserved rewards on the bonus round and compared RMSE. **g**) The correspondence between each bonus round judgment and model predictions, where each dot is a data point and each line is the group-level effect of a Bayesian mixed model. The Bayesian R^2 of each mixed model is reported (see Table A3 for details). **h**) Only the GP and BMT make uncertainty estimates. Here we show the correspondence between rank ordered (per participant) confidence judgments and model uncertainty estimates. Dots indicate means with error bars showing 95% CI, and colored lines represent a linear regression. The regression coefficient corresponds to a Bayesian mixed model fit to the raw, untransformed data (see Table A3).

with participant confidence ratings. In order to test this relationship by accounting for individual differences in subjective ratings of confidence, we fit a mixed effects model to predict the raw confidence judgment (Likert scale 1-11)

using the GP uncertainty estimate as a fixed effect and participant as a random effect (Table A3). The results showed a strong correspondence between lower confidence ratings and higher GP uncertainty estimates ($b_{gpUncertainty} = -3.1$, 95% HPD: $[-4.4, -1.9]$, $BF_{10} = 4.5 \times 10^5$).

To summarize, Experiment 2 showed that participants leverage their functional knowledge over graph-structured reward environments to guide sampling decisions. Participants searched for rewards locally and found highly rewarding nodes much faster than would be expected under the assumption of independent options. We again found support for a GP model of function learning, augmented with a probabilistic action policy including both directed and undirected exploration. The GP provided the best predictive accuracy of choices, produced similar learning curves to human performance, and accurately predicted judgments about expected reward and confidence.

Nevertheless, we also found that the two nearest neighbor models closely matched the GP in terms of predicting choices and participants' judgements of unobserved nodes. However, only the GP model can generate predictions of uncertainty, which we found to match well with participants' confidence judgments. Given that we also found lower levels of α compared to the ground truth, participants likely generalized only very locally. Yet they still tracked their uncertainty about different options, and used that uncertainty to guide their exploration and to rate the confidence of their own predictions. The ability to model these characteristics of human behavior is what makes the GP model a superior model of human behavior in our task.

General discussion

We studied how people learn and exploit graph-structured functions in two experiments. In Experiment 1, we studied how people make predictions about the values of unobserved nodes on a graph and estimated their level of confidence. In Experiment 2, we studied how people searched for rewards in a multi-armed bandit task with graph-correlated rewards. In both experiments, we found that participants made inferences, rated confidence, and navigated the exploration-exploitation dilemma consistent with a Bayesian model of human function learning. This model is implemented using GP regression, which has previously been shown to accurately describe function learning in continuous domains. Here, we replaced the RBF kernel commonly used in continuous domains with a diffusion kernel, where connectivity rather than feature similarity defines relationships in structured environments. The diffusion kernel in turn contains the RBF kernel as a special case, where any Cartesian feature space is equivalent to an infinitely fine undirected lattice graph. Thus, our model expands upon past research on human function learning to richer, graph-structured domains.

Our work also relates directly to classical work on human generalization (Shepard, 1987). Just as in Shepard's original theory, the diffusion kernel defines a distance-dependent similarity measure which assumes that the similarity between nodes decays with their (graph) distance. Similar mechanisms have permeated theories of category learning, where participants learn about a stimulus class given its features (Kruschke, 1992; Love, Medin, & Gureckis, 2004; Medin & Schaffer, 1978; Nosofsky, 1984). Indeed, similar models have also been used to explain human decision making. For example, Gureckis and Love (2009) showed that participants can use the covariance between changing payoffs and the systematic change of a state cue to generalize experiences from known to novel states, and that a linear network learning using similarities between features matched well with participants' behavior. By further combining the diffusion kernel with traditional models of generalization and category learning, we hope to pave the way towards a truly unifying theory of human generalization (Shepard, 1987; Tenenbaum & Griffiths, 2001; Wu, Schulz, Garvert, et al., 2018).

Most directly related to our work is the theory of property induction developed by Kemp and Tenenbaum (2009), who showed how different assumptions about graph-structured functions can lead to different patterns of generalization, consistent with human data. For example, assumptions about genetic transmission through a taxonomic tree license different patterns of generalization compared to assumptions of disease transmission through a food chain. Whereas Kemp and Tenenbaum studied binary property induction, we have focused on real-valued properties in this paper. We have also gone beyond induction to study the role of structured function learning in decision making.

Recent work in reinforcement learning has also developed models related to the diffusion kernel. In particular, cumulative rewards can be estimated efficiently using the successor representation (SR), which represents states of the environment in terms of the expected future occupancy of other states (Dayan, 1993; Gershman, 2018b). For example, a particular subway station would be represented by a vector encoding the expected future occupancies of other stations in the network. When an agent follows a random walk in state space (approximating a diffusion process), the SR is equivalent to the inverse graph Laplacian. Thus, while it does not make probabilistic predictions about cumulative reward values, the SR is able to generalize based on the diffusion of cumulative rewards in a graph-structured state space.

One limitation of our current model implementation is that we assumed the graph structure to be known *a priori*. While this may be a reasonable assumption in problems such as navigating a subway network, where maps are readily available, this is not always the case. One promising avenue for future research is to combine our model with other approaches that learn the underlying structure from experience. The Bayesian structure learning framework pro-

posed by Kemp and Tenenbaum (2008) learns a "conceptual universe" of different graphs. The Bayesian model assigns a score to each candidate graph based on its prior probability and its likelihood. The prior is specified by a generative model for graphs (which can generate grids, trees, and chains, among other graphs) that favors simple, regular graphs over complex ones. The likelihood is based on the match between the observed data and the graph structure, under the assumption that the feature values of the data vary smoothly over the graph. In particular, the features are assumed to be multivariate Gaussian distributed with a covariance function defined by a variant of the regularized Laplacian kernel (Smola & Kondor, 2003; Zhu, Lafferty, & Ghahramani, 2003), which is closely related to the diffusion kernel used here.

Another limitation of our current study is that several variants of a simple nearest-neighbor averaging rule were also surprisingly effective heuristics to capture human behavior in our tasks. Both the dNN and kNN can be understood as binarized simplification of the similarity metric used by the GP. While the GP predicts expected rewards using a similarity-weighted sum of previous observations (Eq. 4), the dNN and kNN use either a distance or count-based threshold of similarity, such that nodes are either similar if considered a neighbor, or dissimilar otherwise. Similar nodes are then averaged, equivalent to an equal-weight regression model (Lichtenberg & Simsek, 2016; Wesman & Bennett, 1959). Although these heuristics are able to efficiently capture many aspects of judgments and choices, our results also show that human behavior is sensitive to the uncertainties about their own predictions, using them to rate their own confidence and to preferentially explore more uncertain options when searching for rewards. This aspect could only be captured by the GP model, which can generate Bayesian uncertainties about its own predictions. Future studies could try to create further heuristic models that also calculate uncertainties of different nodes, for example by using Bayesian versions of the nearest neighbor algorithms (Behmo, Marcombes, Dalalyan, & Prinet, 2010).

Currently, we have only focused on using the diffusion kernel for modeling smooth functions on graph structures. However, the real world contains mixes of continuous and discrete structures, such as in our example of Darwin's finches. How could we model these more complex mixtures of structures? Participants' ability to learn more complex yet highly structured functions in a continuous domain have been explained by using compositional kernels (Schulz, Tenenbaum, et al., 2017). Compositional kernels learn about functions through combining different structures, starting from simple building blocks that can be composed. Thus, one avenue for future research could be to model human function learning using kernel that compose together these mixtures of graph and continuous structures.

Finally, even under the assumption that participants know the full graph structure, our model additionally assumes that they have direct access to the value of each node during inference. Future studies could try to further enrich our tasks to scenarios in which participants have to plan moves over the graph, or to require that participant remember previously observed outputs. While studying how people plan on known graphs would connect our work further to past research on hierachical planning in human reinforcement learning (Balaguer, Spiers, Hassabis, & Summerfield, 2016; Tomov, Yagati, Kumar, Yang, & Gershman, 2018), adding a forgetting component to our model and task would connect it to memory-based models of learning (Bornstein & Norman, 2017; Collins & Frank, 2012) and decision making (Bhui, 2018; Stewart, Chater, & Brown, 2006).

In summary, our behavioral results and proposed model considerably expand past studies of human function learning to graph-structured domains, and emphasize the importance of function learning and uncertainty-guidance to explain human behavior in such domains.

References

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, *3*, 397–422.

Balaguer, J., Spiers, H., Hassabis, D., & Summerfield, C. (2016). Neural mechanisms of hierarchical planning in a virtual subway network. *Neuron*, *90*(4), 893–903.

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, *68*(3), 255–278.

Behmo, R., Marcombes, P., Dalalyan, A., & Prinet, V. (2010). Towards optimal naive bayes nearest neighbor. In *European conference on computer vision* (pp. 171–184).

Bhui, R. (2018). Case-based decision neuroscience: Economic judgment by similarity. In *Goal-directed decision making* (pp. 67–103). Elsevier.

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology*, *2*(1), 113–120.

Bornstein, A. M., & Norman, K. A. (2017). Reinstated episodic context guides sampling-based decisions for reward. *Nature neuroscience*, *20*(7), 997.

Bott, L., & Heit, E. (2004). Nonmonotonic extrapolation in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(1), 38.

Brehmer, B. (1974). Hypotheses about relations between scaled variables in the learning of probabilistic inference tasks. *Organizational Behavior and Human Performance*, *11*(1), 1–27.

Brehmer, B. (1976). Learning complex rules in probabilistic inference tasks. *Scandinavian Journal of Psychology*, *17*(1), 309–312.

Busemeyer, J. R., Byun, E., DeLosh, E. L., & McDaniel, M. A. (1997). Learning functional relations based on experience with input-output pairs by humans and artificial neural net-

works. In K. Lamberts & D. Shanks (Eds.), *Concepts and Categories* (p. 405-437). Cambridge: MIT Press.

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. doi: 10.18637/jss.v080.i01

Carroll, J. D. (1963). Functional learning: The learning of continuous functional mappings relating stimulus and response continua. *ETS Research Bulletin Series*, *1963*, i–144.

Christakou, A., Gershman, S. J., Niv, Y., Simmons, A., Brammer, M., & Rubia, K. (2013). Neural and psychological maturation of decision-making in adolescence and young adulthood. *Journal of Cognitive Neuroscience*, *25*, 1807–1823.

Cohen, J. D., McClure, S. M., & Yu, A. J. (2007). Should i stay or should i go? how the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *362*(1481), 933–942.

Collins, A. G., & Frank, M. J. (2012). How much of reinforcement learning is working memory, not reinforcement learning? a behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, *35*(7), 1024–1035.

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

Dayan, P. (1993). Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, *5*, 613–624.

DeLosh, E. L., Busemeyer, J. R., & McDaniel, M. A. (1997). Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*, 968–986.

Farashahi, S., Rowe, K., Aslami, Z., Lee, D., & Soltani, A. (2017). Feature-based learning improves adaptability without compromising precision. *Nature communications*, *8*(1), 1–16.

Fruchterman, T. M., & Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and experience*, *21*, 1129–1164.

Gershman, S. J. (2015). A unifying probabilistic view of associative learning. *PLoS Computational Biology*, *11*, e1004567.

Gershman, S. J. (2018a). Deconstructing the human algorithms for exploration. *Cognition*, *173*, 34–42.

Gershman, S. J. (2018b). The successor representation: Its computational logic and neural substrates. *Journal of Neuroscience*, *38*, 7193–7200.

Gershman, S. J. (2019). Uncertainty and exploration. *Decision*, *6*(3), 277–286.

Gershman, S. J., & Blei, D. M. (2012). A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, *56*(1), 1–12.

Gershman, S. J., & Niv, Y. (2015). Novelty and inductive generalization in human reinforcement learning. *Topics in Cognitive Science*, *7*, 391–415.

Gershman, S. J., Pesaran, B., & Daw, N. D. (2009). Human reinforcement learning subdivides structured action spaces by learning effector-specific values. *Journal of Neuroscience*, *29*, 13524–13531.

Griffiths, T. L., Lucas, C., Williams, J., & Kalish, M. L. (2009). Modeling human function learning with gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 553–560).

Gronau, Q. F., Singmann, H., & Wagenmakers, E.-J. (2017). Bridgesampling: An R package for estimating normalizing constants. *arXiv preprint arXiv:1710.08162*.

Gureckis, T. M., & Love, B. C. (2009). Learning in noise: Dynamic decision-making in a variable environment. *Journal of Mathematical Psychology*, *53*(3), 180–193.

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623.

Kalish, M. L., Lewandowsky, S., & Kruschke, J. K. (2004). Population of linear experts: knowledge partitioning and function learning. *Psychological Review*, *111*, 1072.

Kemp, C., & Tenenbaum, J. B. (2008). The discovery of structural form. *Proceedings of the National Academy of Sciences*, *105*, 10687–10692.

Kemp, C., & Tenenbaum, J. B. (2009). Structured statistical models of inductive reasoning. *Psychological Review*, *116*, 20.

Knox, W. B., Otto, A. R., Stone, P., & Love, B. (2012). The nature of belief-directed exploratory choice in human decision-making. *Frontiers in psychology*, *2*, 398.

Koh, K., & Meyer, D. E. (1991). Function learning: Induction of continuous stimulus-response relations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*, 811.

Kondor, R. I., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th International Conference on Machine Learning* (pp. 315–322).

Kruschke, J. K. (1992). Alcove: an exemplar-based connectionist model of category learning. *Psychological review*, *99*(1), 22.

Kwantes, P. J., & Neal, A. (2006). Why people underestimate y when extrapolating in linear functions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *32*, 1019.

Langville, A. N., & Meyer, C. D. (2011). *Google's pagerank and beyond: The science of search engine rankings*. Princeton University Press.

Leuker, C., Pachur, T., Hertwig, R., & Pleskac, T. J. (2018). Exploiting risk–reward structures in decision making under uncertainty. *Cognition*, *175*, 186–200.

Lichtenberg, J. M., & Simsek, Ö. (2016). Simple regression models. In *Proceedings of the NIPS 2016 Workshop on Imperfect Decision Makers: Admitting Real-World Rationality, Barcelona, Spain, December 9, 2016.* (pp. 13–25).

Love, B. C., Medin, D. L., & Gureckis, T. M. (2004). Sustain: a network model of category learning. *Psychological Review*, *111*, 309.

Lucas, C. G., Griffiths, T. L., Williams, J. J., & Kalish, M. L. (2015). A rational model of function learning. *Psychonomic Bulletin & Review*, *22*, 1193–1215.

McClelland, J. L., Rumelhart, D. E., Group, P. R., et al. (1986). Parallel distributed processing. *Explorations in the Microstructure of Cognition*, *2*, 216–271.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*, 207–238.

Mehlhorn, K., Newell, B. R., Todd, P. M., Lee, M. D., Morgan, K., Braithwaite, V. A., ... Gonzalez, C. (2015). Unpacking the

exploration–exploitation tradeoff: A synthesis of human and animal literatures. *Decision*, *2*(3), 191.

Nosofsky, R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*, 104–114. doi: 10.1037/0278-7393.10.1.104

Payzan-LeNestour, E., & Bossaerts, P. (2011). Risk, unexpected uncertainty, and estimation uncertainty: Bayesian learning in unstable settings. *PLoS computational biology*, *7*(1).

Radulescu, A., Niv, Y., & Ballard, I. (2019). Holistic reinforcement learning: the role of structure and attention. *Trends in cognitive sciences*.

Rasmussen, C. E., & Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press: Cambridge, MA.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, *2*, 64–99.

Rigoux, L., Stephan, K. E., Friston, K. J., & Daunizeau, J. (2014). Bayesian model selection for group studies—revisited. *Neuroimage*, *84*, 971–985.

Schulz, E., Bhui, R., Love, B. C., Brier, B., Todd, M. T., & Gershman, S. J. (2019). Structured, uncertainty-driven exploration in real-world consumer choice. *Proceedings of the National Academy of Sciences*, *116*(28), 13903–13908.

Schulz, E., & Gershman, S. J. (2019). The algorithmic architecture of exploration in the human brain. *Current opinion in neurobiology*, *55*, 7–14.

Schulz, E., Konstantinidis, E., & Speekenbrink, M. (2017). Putting bandits into context: How function learning supports decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *44*, 927–943.

Schulz, E., Speekenbrink, M., & Krause, A. (2018). A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, *85*, 1–16.

Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M., & Gershman, S. J. (2017). Compositional inductive biases in function learning. *Cognitive Psychology*, *99*, 44–79.

Schulz, E., Wu, C. M., Huys, Q. J., Krause, A., & Speekenbrink, M. (2018). Generalization and search in risky environments. *Cognitive Science*, *42*, 2592–2620.

Schulz, E., Wu, C. M., Ruggeri, A., & Meder, B. (2018). Searching for rewards like a child means less generalization and more directed exploration. *bioRxiv preprint*.

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

Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological monographs: General and applied*, *75*, 1.

Smola, A. J., & Kondor, R. (2003). Kernels and regularization on graphs. In *Learning theory and kernel machines* (pp. 144–158). Springer.

Speekenbrink, M., & Konstantinidis, E. (2015). Uncertainty and exploration in a restless bandit problem. *Topics in Cognitive Science*, *7*, 351–367.

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

Stewart, N., Chater, N., & Brown, G. D. (2006). Decision by sampling. *Cognitive psychology*, *53*(1), 1–26.

Stojic, H., Schulz, E., Analytis, P. P., & Speekenbrink, M. (2018). It's new, but is it good? how generalization and uncertainty guide the exploration of novel options.

Tenenbaum, J. B., & Griffiths, T. L. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences*, *24*, 629–640.

Tomov, M., Yagati, S., Kumar, A., Yang, W., & Gershman, S. (2018). Discovery of hierarchical representations for efficient planning. *BioRxiv*, 499418.

Wesman, A. G., & Bennett, G. K. (1959). Multiple regression vs. simple addition of scores in prediction of college grades. *Educational and Psychological Measurement*, *19*, 243–246.

Whittle, P. (1980). Multi-armed bandits and the gittins index. *Journal of the Royal Statistical Society: Series B (Methodological)*, *42*(2), 143–149.

Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., & Cohen, J. D. (2014). Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, *143*, 2074–2081.

Wimmer, G. E., Daw, N. D., & Shohamy, D. (2012). Generalization of value in reinforcement learning by humans. *European Journal of Neuroscience*, *35*(7), 1092–1104.

Wu, C. M., Schulz, E., Garvert, M. M., Meder, B., & Schuck, N. W. (2018). Connecting conceptual and spatial search via a model of generalization. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society* (pp. 1183–1188). Austin, TX: Cognitive Science Society.

Wu, C. M., Schulz, E., Garvert, M. M., Meder, B., & Schuck, N. W. (2020). Similarities and differences in spatial and non-spatial cognitive maps. *bioRxiv*. doi: 10.1101/2020.01.21.914556

Wu, C. M., Schulz, E., & Gershman, S. J. (2019). Generalization as diffusion: human function learning on graphs. In *Proceedings of the 41st Annual Conference of the Cognitive Science Society.*

Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (2018). Generalization guides human exploration in vast decision spaces. *Nature Human Behaviour*, *2*, 915–924.

Zhu, X., Lafferty, J. D., & Ghahramani, Z. (2003). Semi-supervised learning: From gaussian fields to gaussian processes.

Appendix A
Regression Models

All Bayesian mixed effects regression models were implemented in `brms` with default weak priors (Bürkner, 2017) using No-U-Turn sampling (Hoffman & Gelman, 2014) with the proposal acceptance probability set to .99. In all cases, participant id was used as a random intercept. All fixed effects were also entered as random slopes following a maximal random structure approach (Barr, Levy, Scheepers, & Tily, 2013). This allows us to compute a Bayes factor comparing each model against a null model, which used the same random effect structure but with the target fixed effect omitted. The Bayes factor was computed using bridge sampling (Gronau, Singmann, & Wagenmakers, 2017) as a method to approximate the marginal likelihood of both models. All models were estimated over four chains of 4000 iterations, with a burn-in period of 1000 samples.

Table A1
Experiment 1: Mixed Effects Models

	Judgment Accuracy	Judgment Accuracy	Confidence Rating	Confidence Rating
	Est.	*Est.*	*Est.*	*Est.*
Intercept	12.77	14.89	6.62	8.26
	[11.73, 12.80]	[13.50, 16.31]	[6.14, 7.11]	[7.95, 8.58]
Observed Nodes	-0.60		0.23	
	[-0.79, -0.41]		[0.17, 0.30]	
Confidence Rating		-0.66		
		[-0.83, -0.49]		
GP Uncertainty				-1.80
				[-2.50, -1.12]
Random Effects				
σ^2	4.86	4.77	2.61	2.40
τ_{00}	73.37	73.52	3.37	3.58
ICC	0.06	0.06	0.44	0.40
N	100	100	100	100
Observations	3000	3000	3000	3000
Bayesian R^2	.07	0.09	0.46	0.43

Note: We report the posterior mean and 95% highest posterior density (HPD) interval below in brackets. σ^2 indicates the individual-level variance, τ_{00} indicates the variation between individual intercepts and the average intercept, and ICC is the intraclass correlation coefficient.

Table A2
Experiment 2: Mixed Effects Models

	P(repeat)	Graph Distance	Reward	Eigen Centrality
	Odds Ratio	*Est.*	*Est.*	*Est.*
Intercept	0.00	10.59	78.61	0.15
	[0.00,0.00]	[9.85, 11.36]	[76.65, 80.51]	[0.14, 0.17]
Previous Reward	1.13	-0.11		
	[1.12,1.14]	[-0.12, -0.10]		
Trial				-0.003
				[-0.003, -0.002]
Eigen Centrality			-26.65	
			[-31.17, -22.04]	
Random Effects				
σ^2	-0.01	1.37	65.15	0.003
τ_{00}	0.25	18.47	496.55	0.04
ICC	-0.02	0.07	0.12	0.07
N	98	98	98	98
Observations	22050	22050	22050	22932
Bayesian R^2	.58	.42	.17	.08

Note: We report the posterior mean and 95% highest posterior density (HPD) interval below in brackets, except the first column, which is a logistic regression reporting the odds ratio and the respective 95% CI. σ^2 indicates the individual-level variance, τ_{00} indicates the variation between individual intercepts and the average intercept, and ICC is the intraclass correlation coefficient.

Table A3
Experiment 2: Bonus Round

	Judgment *Est.*	Judgment *Est.*	Judgment *Est.*	Confidence *Est.*
Intercept	-12.38	5.15	5.24	7.72
	[-26.16,-0.89]	[-1.04, 11.11]	[-2.37, 12.33]	[7.23,8.22]
GP Prediction	1.16			
	[0.93,1.43]			
GP Uncertainty				-3.13
				[-4.37, -1.92]
dNN Prediction		0.80		
		[0.69,0.92]		
kNN Prediction			0.79	
			[0.66,0.94]	
Random Effects				
σ^2	51.14	53.38	63.86	2.55
τ_{00}	275.06	274.38	272.84	4.37
ICC	0.16	0.16	0.17	0.37
N	98	98	98	98
Observations	980	980	980	980
Bayesian R^2	.32	.33	.33	.50

Note: Note that the BMT is not included, since it invariably makes the same prediction for all unobserved options based on the prior mean and prior variance. The bonus round only consisted of judgments about unobserved otions. We report the posterior mean and 95% highest posterior density (HPD) interval below in brackets. σ^2 indicates the individual-level variance, τ_{00} indicates the variation between individual intercepts and the average intercept, and ICC is the intraclass correlation coefficient.

Appendix B

Correspondence between participant judgments and model predictions in Experiment 1

In addition to comparing prediction accuracy (Fig. 3d-e), we also examined the correspondence between participant judgments, the true underlying values, and model predictions. Figure B1a shows a scatter plot comparing participant judgments to the true target value ($r = .59$, $p < .001$, $BF_{10} > 100$), which shows participants were well calibrated to ground truth. Figure B1b-d provides similar scatter plots showing the correspondence between each model's predictions and participant judgments. Overall, the GP had the highest correlation with participant judgments ($r = .71$, $p < .001$, $BF_{10} > 100$), followed by the dNN ($r = .68$, $p < .001$, $BF_{10} > 100$) and kNN models ($r = .67$, $p < .001$, $BF_{10} > 100$)

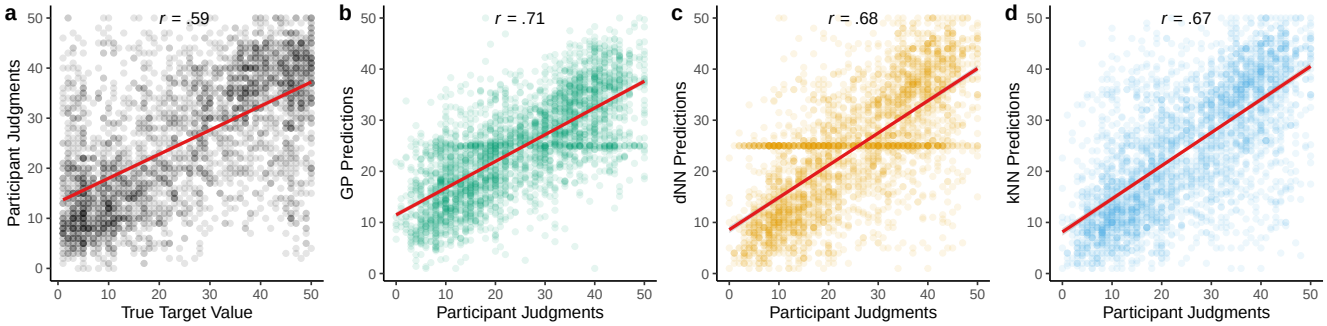

Figure B1. Experiment 1 correspondence between participant and model predictions. **a**) Participant judgments against the ground truth. **b-d**) Model predictions compared against participant judgments. Each dot is a single judgment and the red line is a linear regression. The Pearson correlation coefficient is shown above ($BF_{10} > 10^9$ in all cases).

Appendix C
Eigen Centrality

We analyzed search behavior from Experiment 2 as a function of the connectivity of the sample nodes, which we quantify using eigen centrality (EC; Bonacich, 1972). Intuitively, EC quantifies the connectivity of a node similar to how Google's PageRank (Langville & Meyer, 2011) quantifies webpages based on the number and quality of hyperlinks. Nodes with higher EC are those that exert higher influence on the network, by being connected to other nodes that are themselves highly central in the network. The ECs of each node in a graph $x_i \in \mathbf{x}$ are defined by the normalized eigenvector belonging to the largest eigenvalue λ of the adjacency matrix A, fulfilling the identity:

$$\lambda \mathbf{x} = A\mathbf{x} \tag{14}$$

Compared to the overall distribution of ECs in the task, participants systematically selected nodes with lower EC (one-sample t-test: $t(97) = -9.0$, $p < .001$, $d = 0.9$, $BF = 3.2 \times 10^{11}$; Fig. C1a). Participants also increasingly selected lower EC nodes over successive trials (Bayesian mixed model: $b_{trial} = -0.003$, 95% HPD: $[-0.003, -0.002]$, $BF_{10} = 1.6 \times 10^{11}$; Table A2; Fig. C1b). While EC was not predictive of expected rewards in the underlying task environment ($r = -.03$, $p = .109$, $BF = .17$; Fig. C1c), we found a systematic relationship in the choices made by participants: lower EC nodes sampled by participants had higher reward values ($b_{eigenCentrality} = -26.54$, 95% HPD: $[-31.17, -22.04]$, $BF_{10} = 1.9 \times 10^{20}$; Table A2; Fig. C1d). This is perhaps because nodes with lower EC tended to have more eccentric reward values. Indeed, the highest and lowest rewards across environments had very similar average EC values, of 0.06 and 0.05, respectively. Thus, this trend of preferentially sampling less central nodes may reflect a high-risk high-reward heuristic (Leuker, Pachur, Hertwig, & Pleskac, 2018), which combined with generalization proved to be an adaptive search strategy.

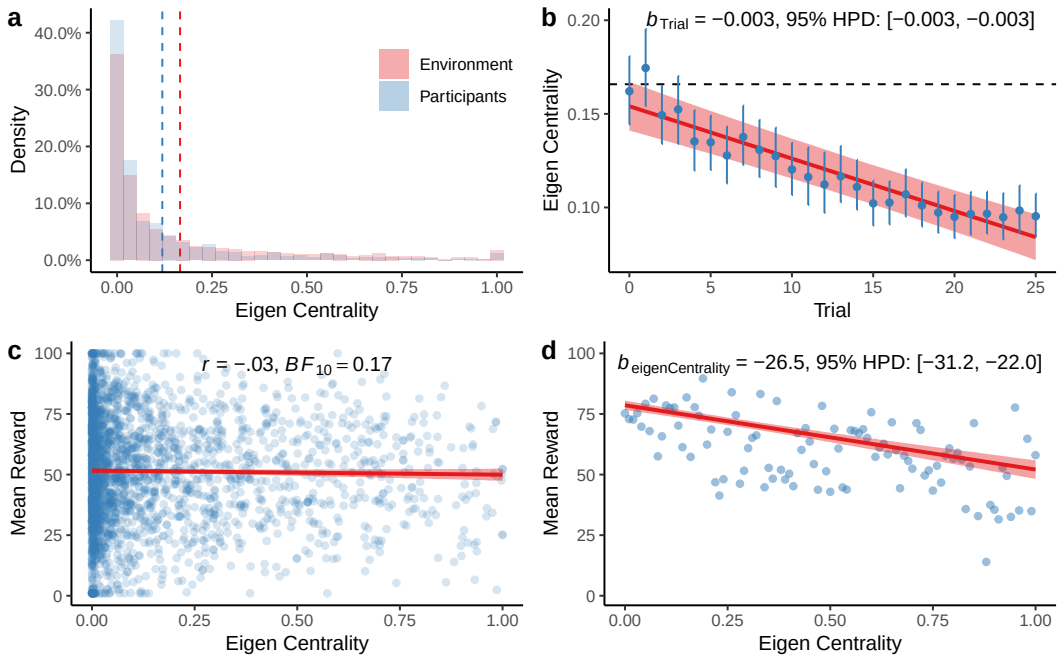

Figure C1. Eigen Centrality. **a**) Distribution of EC values of selected options (blue) compared to the ground truth of the task (red). The vertical dashed lines indicate the means of each distribution. **b**) Participants preferentially selected nodes with lower EC over subsequent trials. Each dot is the aggregate mean (± 95% CI) and the red line shows the group-level effect of a Bayesian mixed model (Table A2), with the ribbon showing the 95% CI. The dashed horizontal line indicates the mean eigen centrality across all nodes in the task. **c**) EC was not predictive of rewards in the task. **d**) However, from the nodes sampled by participants, those with lower EC corresponded to higher rewards. Each dot is dot is the aggregate mean (calculated at intervals of 0.01) and the red line is the group-level effect of a Bayesian mixed model (Table A2), with the ribbon indicating the 95% CI.

Appendix D
Experiment 2 model supplement

Figure D1 provides an overview of parameter estimates for each model, while Figure D2 shows how different parameter estimates were related to different levels of predictive accuracy. In addition, Figure D3 compares the difference in out-of-sample prediction error between the GP and each other model as a function of performance on the bandit task.

Parameter estimates

GP. The GP used the diffusion parameter (α) to define the extent of generalization, where larger values of α implied a wider influence of observed rewards over the graph structure. While the superior predictive accuracy of the GP over alternative models provided evidence for generalization, α estimates were systematically lower than the underlying value of $\alpha = 2$ used to generate the environments ($t(97) = -13.5$, $p < .001$, $d = 1.4$, $BF_{[}10] = 9.3 \times 10^{20}$). Thus, undergeneralization rather than overgeneralization was the norm, consistent with previous findings of a beneficial a bias towards undergeneralization in a similar search context (Wu, Schulz, Speekenbrink, et al., 2018). Additionally, the exploration bonus (β) estimated how participants traded off between exploring uncertain options vs. exploiting options with high expectations of reward. We found that the estimated β values were substantially larger than the lower bound ($t(97) = 4.5$, $p < .001$, $d = 0.5$, $BF_{10} = 949$), providing further evidence for directed exploration. Lastly, we also define a stickiness parameter (ω), which captures an aspect of the high rates of repeat clicks by adding an additional bonus to the value of the last selected options.

BMT. In comparison, while the BMT also made uncertainty estimates, these were defaulted to the prior variance ($\nu_0 = 500$) for all unobserved options. Thus, the BMT made the predictions uncertainty predictions for nodes near and far from previous observations. In contrast to the GP model, we found little evidence for directed exploration using the BMT, with β estimates only marginally different from the lower bound of .007 ($t(97) = 2.1$, $p = .038$, $d = 0.2$, $BF_{10} = .92$). The BMT also made similar use of the stickiness parameter compared to the GP ($t(97) = 0.7$, $p = .460$, $d = 0.1$, $BF_{10} = .15$).

Nearest-neighbors. The dNN generated predictions by averaging the rewards of observed nodes within a distance of d. The mean estimate of distance was $d = 2.4$, although the mode and median were both 1. Thus, the dNN predominately made predictions solely based on observations of directly connected nodes. Nonetheless, it was still able to predict participant choices fairly accurately. While the dNN had no access to directed exploration, we nevertheless find similar levels of random exploration (τ: $t(97) = 1.2$, $p = .230$, $d = 0.1$, $BF_{10} = .23$) and stickiness (ω: $t(97) = -1.0$, $p = .317$, $d = 0.1$, $BF_{10} = .18$) compared to the GP. Thus, one potential source of the gap in simulated learning performance compared to the GP (Fig. **??**b), could be due to the dNN lacking a form of directed exploration. The kNN model also performed similar to the dNN, by averaging the k nearest nodes rather than selecting nodes at a fixed distance. The mean number of neighbors was $k = 3$, and with a mode and median of 2. Thus, like the dNN, generalizations were on the basis of integrating a small number of other observations. The dNN and kNN also shared similar levels of both undirected exploration ($t(97) = 1.8$, $p = .077$, $d = 0.3$, $BF_{10} = .52$), and stickiness ($t(97) = 1.1$, $p = .290$, $d = 0.2$, $BF_{10} = .19$).

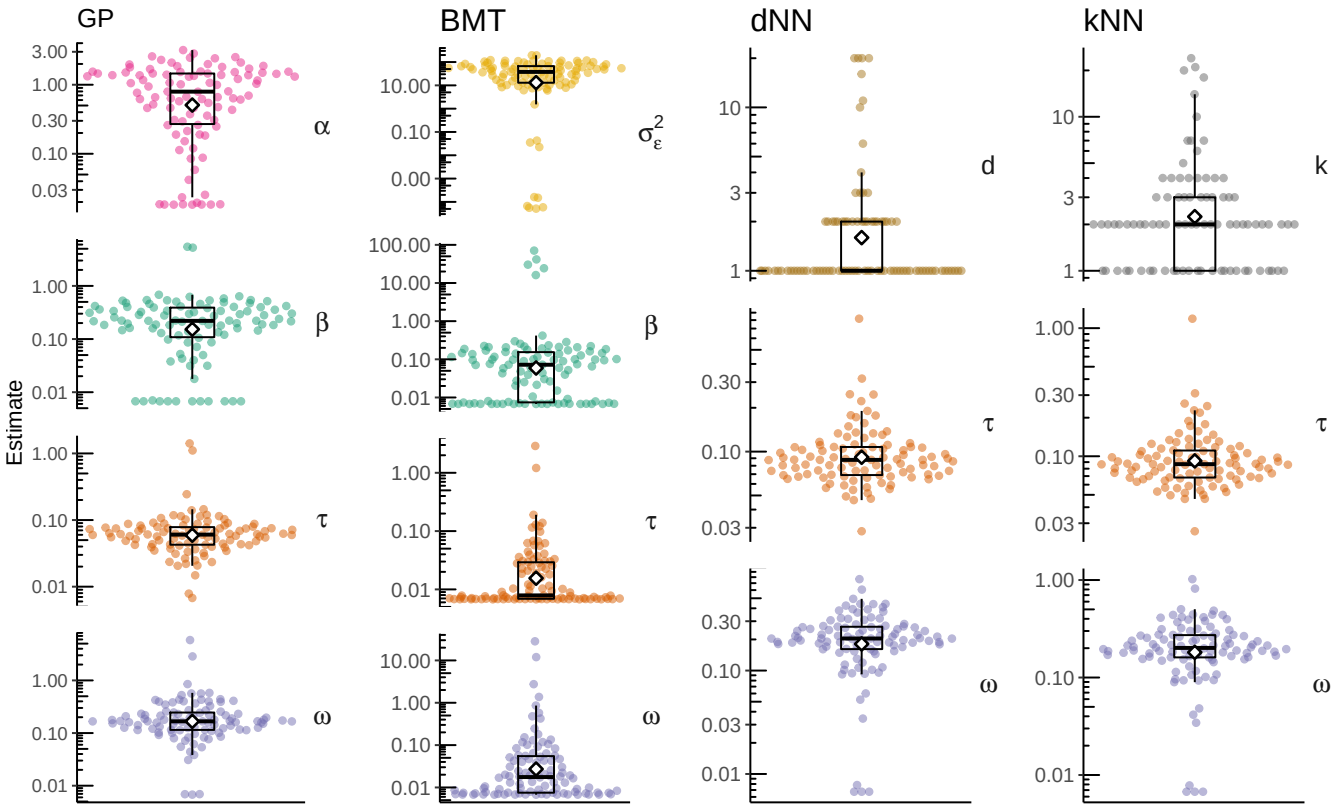

Figure D1. Experiment 2 parameter estimates. Each dot is the median cross-validated estimate for a single participant. The diamond indicates group mean and Tukey box plots show the median and 1.5 inter-quartile range. Note that the y-axis is log-scaled for parameters except *d* and *k*, which are natural numbers.

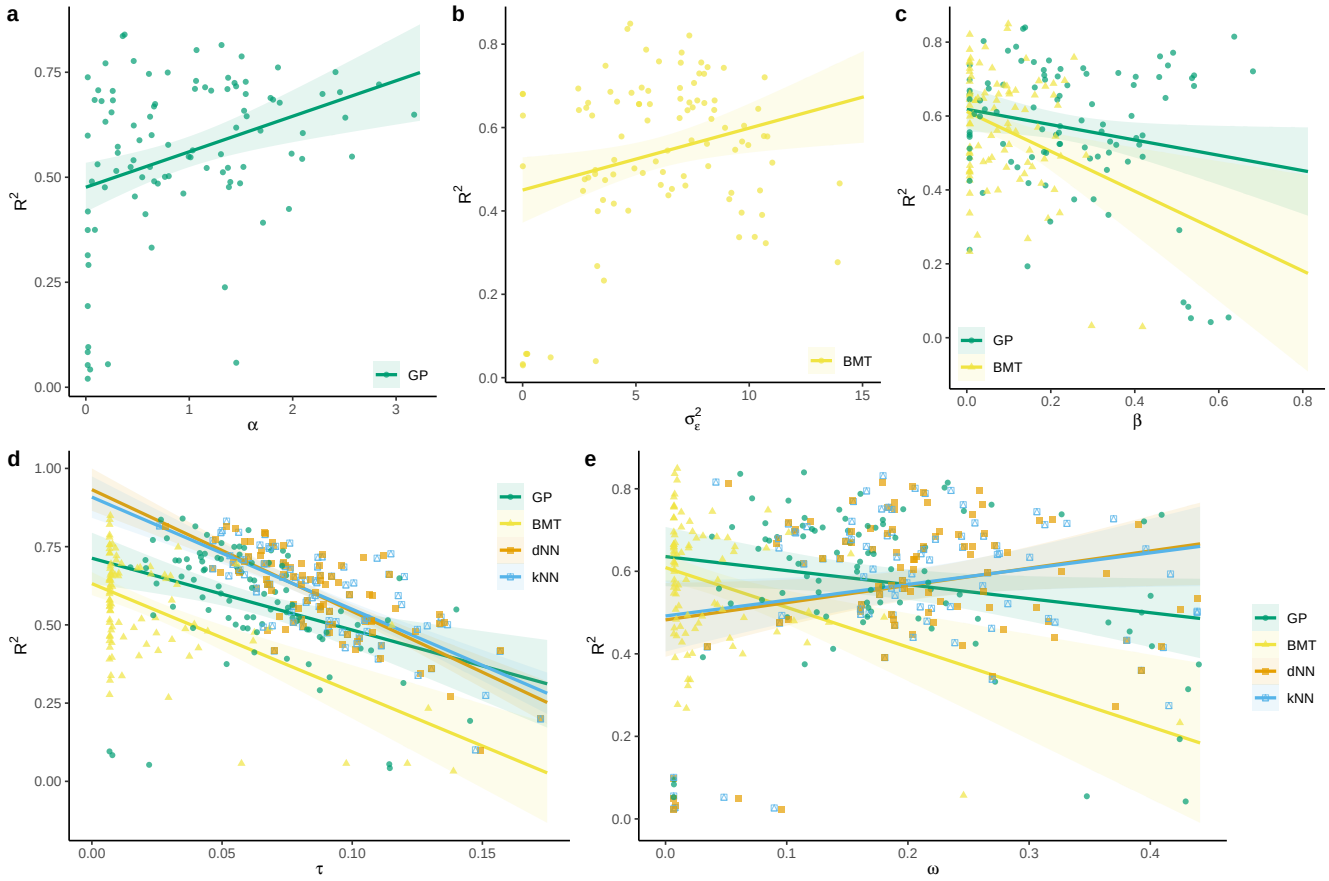

Figure D2. Experiment 2 model parameters and predictive accuracy. Each panel compares median per-participant parameter estimates against predictive accuracy (R^2), as an intuitive measure of objective model performance. Predictive accuracy compares the out-of-sample log loss of any given model k against a random model: $R^2 = 1 - \log \mathcal{L}_k / \log \mathcal{L}_{rand}$. Intuitively, $R^2 = 0$ indicates a model is equivalent to random chance, while $R^2 = 1$ is a theoretically perfect model. Each dot is a single participant, with linear regression lines added (ribbon indicates standard error). **a**) The diffusion parameter α measuring the level of generalization, where higher levels of generalization corresponded to better model predictions ($r = .34$, $p < .001$, $BF_{10} = 54$) **b**) The inverse sensitivity parameter σ^2_ϵ, where higher estimates (i.e., smaller learning updates) corresponded to better model predictions ($r = .25$, $p = .012$, $BF_{10} = 4.7$). **c**) The exploration bonus β controls the level of *directed* exploration. For both GP and BMT models, higher levels of directed exploration corresponded to worse model predictions ($BF_{10} > 100$). **d**) The softmax temperature parameter τ controls the level of *random* exploration. For all models, temperatures corresponded to worse model predictions ($BF_{10} > 100$). **e** The stickiness parameter ω added a bonus to the previously selected option, making it more likely to choose the same option on the next trial. For the GP and BMT models, stickiness was correlated with worse model model predictions ($BF_{10} > 100$), whereas there was no relationship for the dNN and kNN models ($BF_{10} < 1$).

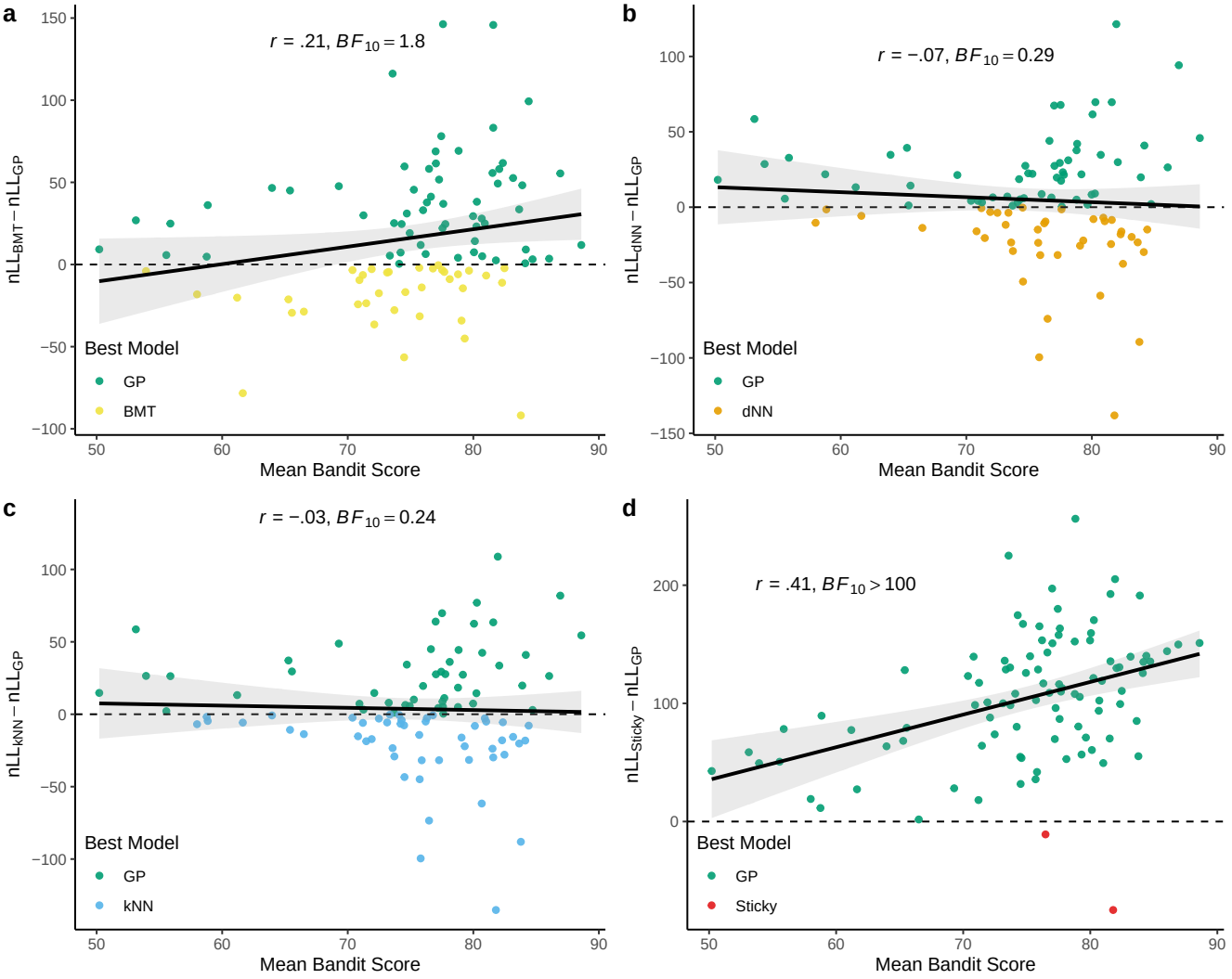

Figure D3. Experiment 2 Model Differences and Score. Each panel compares the difference in out-of-sample prediction error, measured as negative log likelihood (nLL), of two models as a function of mean performance on the bandit task. Each dot is a single participant, with the Pearson correlation shown above and a linear regression line added to the plot (ribbon indicates standard error). The color of each dot indicates the model with the lower nLL. **a**) GP vs. BMT. **b**) GP vs. dNN. **c**) GP vs. kNN. **d**) GP vs. Stickiness and softmax model.