LANGUAGE RESHAPES EXPERIENTIAL LEARNING IN HUMANS AND MACHINES

Ellen Su[1,*] **Divya Srinivasan**[2] **Mark K. Ho**[1,2] **Todd M. Gureckis**[1,2]

[1]Center for Data Science, New York University, New York, NY, USA
[2]Department of Psychology, New York University, New York, NY, USA
[*]Correspondence: ellensu@nyu.edu

ABSTRACT

Humans acquire knowledge through two profoundly different routes: explicit instruction through language, and implicit experience accumulated through action. How these sources reconcile to guide decisions is poorly understood. To study their integration, we introduce the Instructed Bandit Tasks (IBT), eight sequential decision-making problems in which participants earn rewards while given natural-language hints conveying partial information about the task environments. Across two large ($N = 337$ and $N = 300$) behavioral experiments, linguistic hints improved performance and systematically reshaped choice trajectories. We propose modeling these effects with a hierarchical Bayesian RL agent, which treats language as a probabilistic constraint over beliefs, and compare it against Centaur, a language model fine-tuned on human behavior. Both models predicted which hints improve learning, and Centaur additionally reproduced when language hurt performance or lost its benefit under additional experience. Their complementary strengths suggest that normative and data-driven cognitive models can together capture the nuances of human learning.

1 Introduction

Human decision making is influenced by at least two types of information, which each tug at the processes of deliberation and choice. One, sometimes known as value-based decision making, is acquired through direct experience [1, 2], and is implicit, statistical, and accrued incrementally across time [3]. The other, communicated rapidly and abstractly through language, is semantically rich, communicative, and often vague [4–6]. Though seemingly distinct, language and direct experience are deeply integrated in everyday decision making and learning [7, 8]. When deciding where to get coffee, for example, an individual can rely on their own past experiences at different cafes while also considering a recommendation of a friend that "Coffee Project is the best."

The integration of language with experiential learning is non-trivial, as these two sources rely on somewhat incommensurate cognitive processes and representations [9, 10]. Prior research has largely overlooked the direct integration of language and experience, in part due to the relative scarcity of theoretical frameworks for combining language with other forms of learning (but see [9–11]). For instance, research on the "description–experience gap" has, in separate tasks, explored how *experienced* versus *described* gambles influence risky choice behavior [12] while leaving their integration relatively underexplored [8, 10]. However, recent

Preprint. This manuscript is under review.

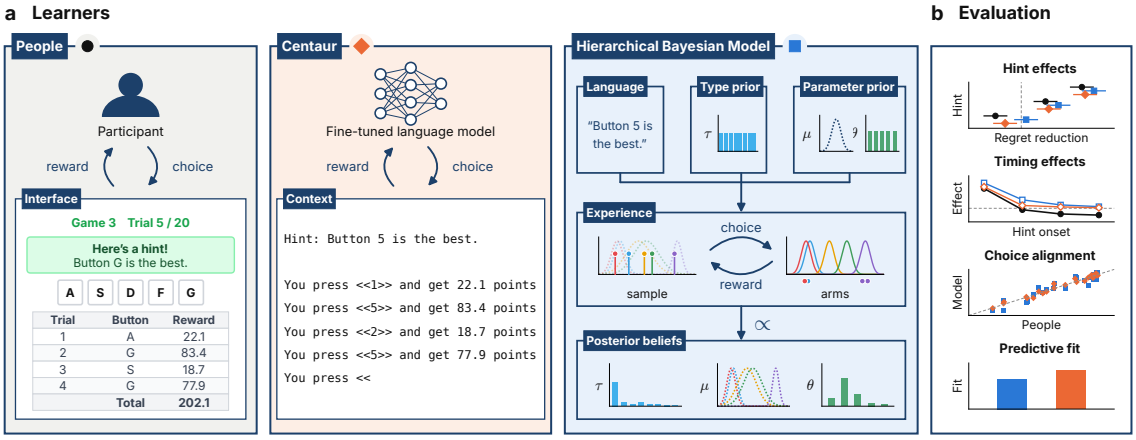


Figure 1. The Instructed Bandit Tasks and modeling approach. a) Participants see a hint and their reward history in the IBT task interface. Centaur, a language model fine-tuned on human behavior, receives the same task instructions, hint, and history as text. The Bayesian model combines the hint with priors over task types (τ), arm means (μ) and task-specific parameters (θ), and updates its beliefs after each choice and reward. This paradigm yields choice trajectories for all three learners under each task condition. **b)** The three learners are compared on the effect on performance for each hint, hint effects based on timing, choice patterns, and fit to held-out human choices.

advances in AI, particularly the linguistic competence of large language models (LLMs), make capturing the influence of language on other types of processes more amenable to comprehensive study [13].

In this paper, we conceptualize the integration of language and experience in the form of "hints" provided to a learner during experiential learning. Hints are a common form of language-based input, resembling the advice, tips, and recommendations that accompany everyday learning. Like a friend's verbal hint on how to play a board game, hints need not convey the relative values of all options, carry explicit measures of confidence, or agree with the learner's own experience to be informative [8], and it is yet unknown how such partial descriptions of a problem influence decision making and choice.

To understand *how* people integrate language and experience requires probing for the different effects language can have on choice behavior. We establish a new approach to studying the integration process by augmenting a popular experimental learning paradigm, the multi-armed bandit (MAB) task [14, 15], with natural language hints to create a set of Instructed Bandit Tasks (IBT). In the IBT, learning agents receive a hint providing partially constraining information about the structure or parameters of a sequential decision-making task, make a series of choices, and observe evaluative feedback on those choices in the form of rewards (Fig. 1). The hints guide learners towards better options or to adopt certain policies, vary in their informativeness, and are experimentally provided at different times to systematically interrogate the learning process.

In concert with our novel empirical task, we develop a theory of how language input reshapes a learner's world model by constraining their beliefs over the task structures [similar to 16]. We model this via a hierarchical Bayesian reinforcement learning (RL) agent, in which natural language hints translate into functions which contribute an additional prior over the learning agent's belief states. By modeling human choices across combinations of hints, task structures, and experimental conditions, we found that humans integrate natural language and experience in ways that reflect Bayesian principles, and that language models aligned to human behavior reproduce these tendencies. This work provides a formal and more expansive account of how language and experience interact to guide choice.

Table 1. Examples of each hint type for the *stationary* and *Bernoulli* tasks. In the behavioral task, the arm choices are given by buttons on the screen. The placeholders in the hints map a given arm ranking to the corresponding button letter in that game (see note).

Bandit	Type	Hint
Stationary	V	{max1} is better than {min2}, {min1}, and {mid}.
	S	One button is the best.
	P	Explore all buttons, then stick with the one that gave the most points.
Bernoulli	V	{max2} is most likely to return 100 points.
	S	The probability for returning points is fixed for each button.
	P	Choose a button multiple times to learn its probability of returning points.

Note: aliases for the arms in ranked order are {min2}, {min1}, {mid}, {max1}, {max2}.
Key: V = Value of arms; S = Structure of problem; P = Policy of optimal solution.

1.1 Instructed Bandit Tasks (IBT)

To study the integration of language with experience, we added task-relevant natural language hints to variants of the multi-armed bandit. MABs formalize sequential, value-based decisions under uncertainty, have been studied computationally in economics, optimization, and RL for several decades [2, 14, 15, 17], and are a popular framework for studying animal and human learning [18–20], since simple reward-maximizing tasks can be performed regardless of language competence.

In the simplest case of a stationary N-armed bandit, a learning agent repeatedly chooses between N arms, each returning a reward sampled from a fixed Gaussian distribution with unknown mean μ and variance σ^2, and must balance exploration, in observing the rewards from each arm, with exploitation, in choosing the arm with the highest expected reward [15]. Building off of the stationary MAB, we study eight tasks which include bandit problems with non-stationary arm means (*drifting*, *patch foraging*), as when a favorite restaurant changes quality; context dependencies (*two context* and *three context*), as when the fastest route home depends on the day of the week; time dependencies (*time delayed*), as when effective changes in diet and exercise are only observed weeks later; alternative reward distributions (*Bernoulli*), as when one bus line proves more reliably on time than another; and feature dependencies (*linear contextual*), as when an apartment is judged by a renter's personalized priorities. We selected this set from a survey of bandit varieties in the neuroscience, psychology, and human RL literature to span commonly studied dynamics. We include formal definitions of the task types in Appendix A and Table A1.

In contrast to standard MAB scenarios, language comprehension is critical in our *Instructed* Bandit Tasks. In the IBT, the formulation of the problem and the agent's learning objective are fixed, but the agent additionally receives a natural language hint, such as "Arm 1 is greater than 25". This hint provides only partial information, as it is uninformative about the relative values of the other arms. If the learning agent effectively harnesses the information provided by the hint as a constraint over the problem space, then their choice behavior should change in a systematic and measurable way.

We created a custom set of hints for each task by conducting pilot studies with participants, the Centaur model (a language model fine-tuned on human behavior), and the Bayesian model and searching for hints which affected learning behavior. This set should be understood as a sample of hints that can influence learning rather than a representative sample of all possible hints. Each hint can also be roughly classified by whether it contributed information about the value of the arms (V), structure of the problem (S), or policy of the optimal solution (P). The hints are single sentences and include placeholder references to the arm choices, as we rotated the position of the arms across the games to avoid numerical or positional biases. Examples of each hint type are provided in Table 1 and the full dataset of hints is in Appendix B and Table A2.

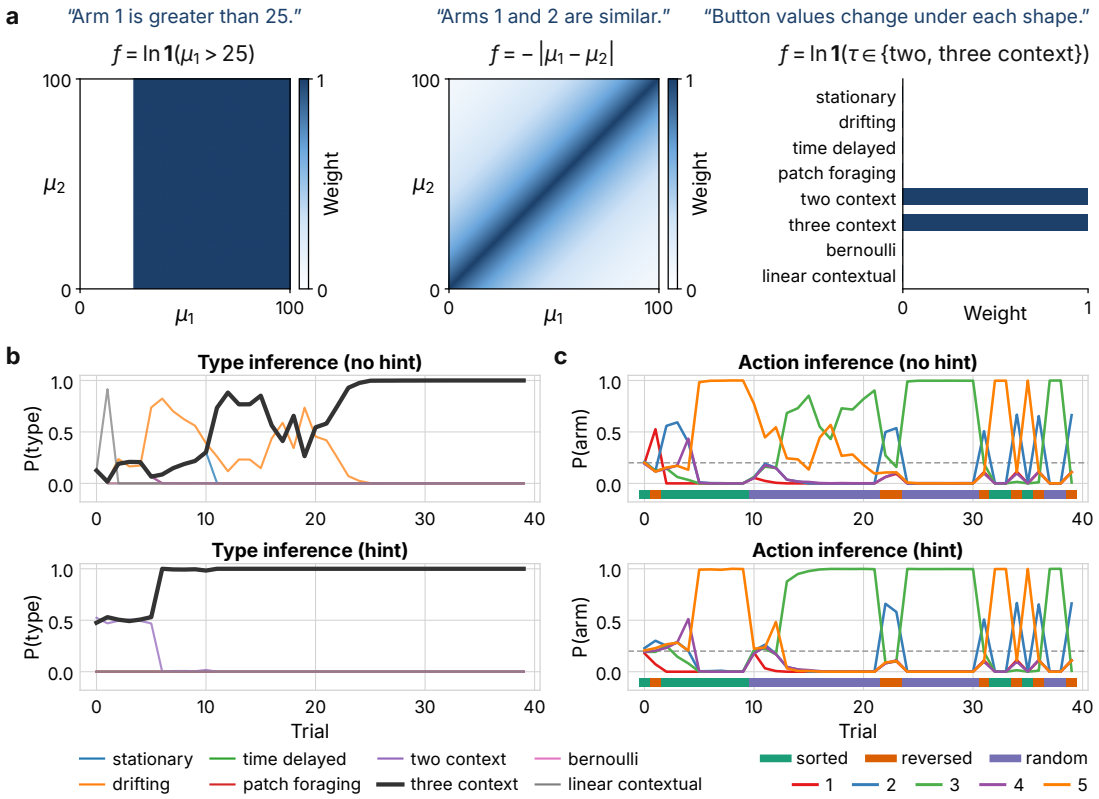


Figure 2. Formalizing language in the Bayesian model. **a)** Each linguistic hint translates into a logical meaning function f, which operates over the parameter space of the problem. $\exp(\gamma f)$ then gives a probabilistic weighting over sampled parameter configurations. Two toy value hints weight the arm means μ_1 and μ_2 (left, center). A structure hint weights the task type τ, here placing equal weight on the two task types with contexts (right). **b)** One simulated *three context* game with and without the hint "The button values change under each shape." The top row plots beliefs computed from the observations alone, and the bottom plots those reweighted by the hint. The bold line is the true type. **c)** The resulting action distributions for the same simulated *three context* game, with (bottom) and without (top) the hint. The dashed line marks chance, and the color strip along the bottom of each plot shows context changes over the trials.

We use the IBT to experimentally manipulate the integration of language and experience in humans through two large-scale studies. Study 1 evaluated participants on all tasks and hints in the IBT. Study 2 additionally manipulated the timing of the hint in two task types (*stationary* and *patch foraging*) to explore whether the amount of accumulated experience changes how participants integrate the information from language. Implementation details for both behavioral experiments are in Section 4.1.

1.2 Theoretical approach

We characterize the integration process by comparing human behavior to two computational models which derive from contrasting assumptions about human learning (Fig. 1). The first is a hierarchical Bayesian reinforcement-learning agent that offers a normative account of the integration process by treating each hint as a prior constraint over the space of possible task representations (Fig. 2a) [4, 21, 22]. As the Bayesian model maintains explicit posteriors over task types, task parameters, and arm values, it is also highly interpretable and allows for inspection into how a hint changes its beliefs and actions at the individual trial level (Fig. 2b,c).

For the second model, the integration of language and experience is emergent and data-driven. Centaur is a large open-weight language model (Llama 3.1 70B base [24]) fine-tuned on a large-scale text corpus of psychological experiments [25]. Centaur is argued to be a "foundation model" for human behavior, meaning

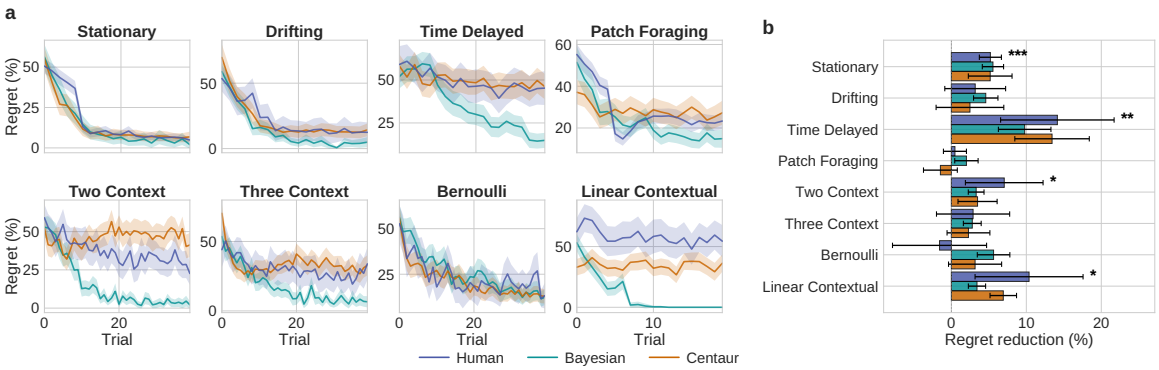


Figure 3. Humans learn in each task and hints improve performance on most task types. a) Trial-by-trial regret (as a percentage of the game's arm spread, where 0% is always choosing the best arm and 100% the worst), averaged across participants for no-hint games (purple), generally decreases over trials for every task type. The Bayesian model (teal) and Centaur (orange) simulations are overlaid. Shaded regions are ± 1 SEM. **b)** Per-type hint advantage for humans and both models, computed as the difference in mean regret between no-hint and hint games, in percentage points of the arm spread (positive values mean hints benefit). Error bars are 95% confidence intervals from a linear mixed model for people and from ordinary least squares with robust standard errors for the models (Section 4.5.2). The stars *, **, *** denote Benjamini–Hochberg adjusted two-sided $p < .05$, .01, and .001, respectively, for the human hint effects.

it is able to capture a wide range of hints and natural language instructions without hand-engineering. The operation of Centaur is more opaque, and its predictions are agnostic to the explicit model of the reward generating process in each task. Its value is then less as an explanation than as a human choice simulator and a benchmark to identify the aspects of human learning that the formal cognitive model fails to capture [26]. These two models, stemming from quite different traditions, thus provide complementary analyses of the integration of language and experience.

2 Results

2.1 Behavioral experiments

To study the integration of language and experience through the IBT, each task must be learnable. Even without hints, we expect participants to show decreasing trial-by-trial average regret—or difference between the mean of the arm they chose and that of the optimal arm—over the game trials. We normalized raw regret as a percentage of a game's arm value range, and report hint effects throughout as differences in this normalized regret, in percentage points, which makes them comparable across games and task types. In the no-hint games, regret decreased significantly from the first to last five trials across all eight task types (Fig. 3a, purple curves; paired t-tests, t range 2.25–31.07, all Holm-adjusted $p < .05$, see Appendix C and Table A3), supporting the case that participants learn the reward structures of each task through experience alone.

2.1.1 Hints improve learning

Because the hints in IBT are always true (they conveyed information about the actual parameters of the problem to avoid deception), an ideal Bayesian learner should use them to reduce uncertainty about the task and improve performance. Accordingly, we computed the difference in average task regret between participants given a hint and those given no hint for each of the eight tasks. Fig. 3b (purple bars) shows that participants who received a hint achieved lower mean regret throughout the games than those without in seven of eight task types (a mean relative reduction of 6.8% overall and 13.9% in the first five trials). We estimate each effect with a linear mixed-effects model (Section 4.5.2) and report 95% confidence intervals (CI, in brackets throughout) and p-values adjusted for multiple comparisons. Hints significantly reduced regret for the *stationary* (5.2 percentage points, 95% CI [3.7, 6.7], $p < .001$), *time delayed* (14.1, [6.6, 21.7], $p = .001$),

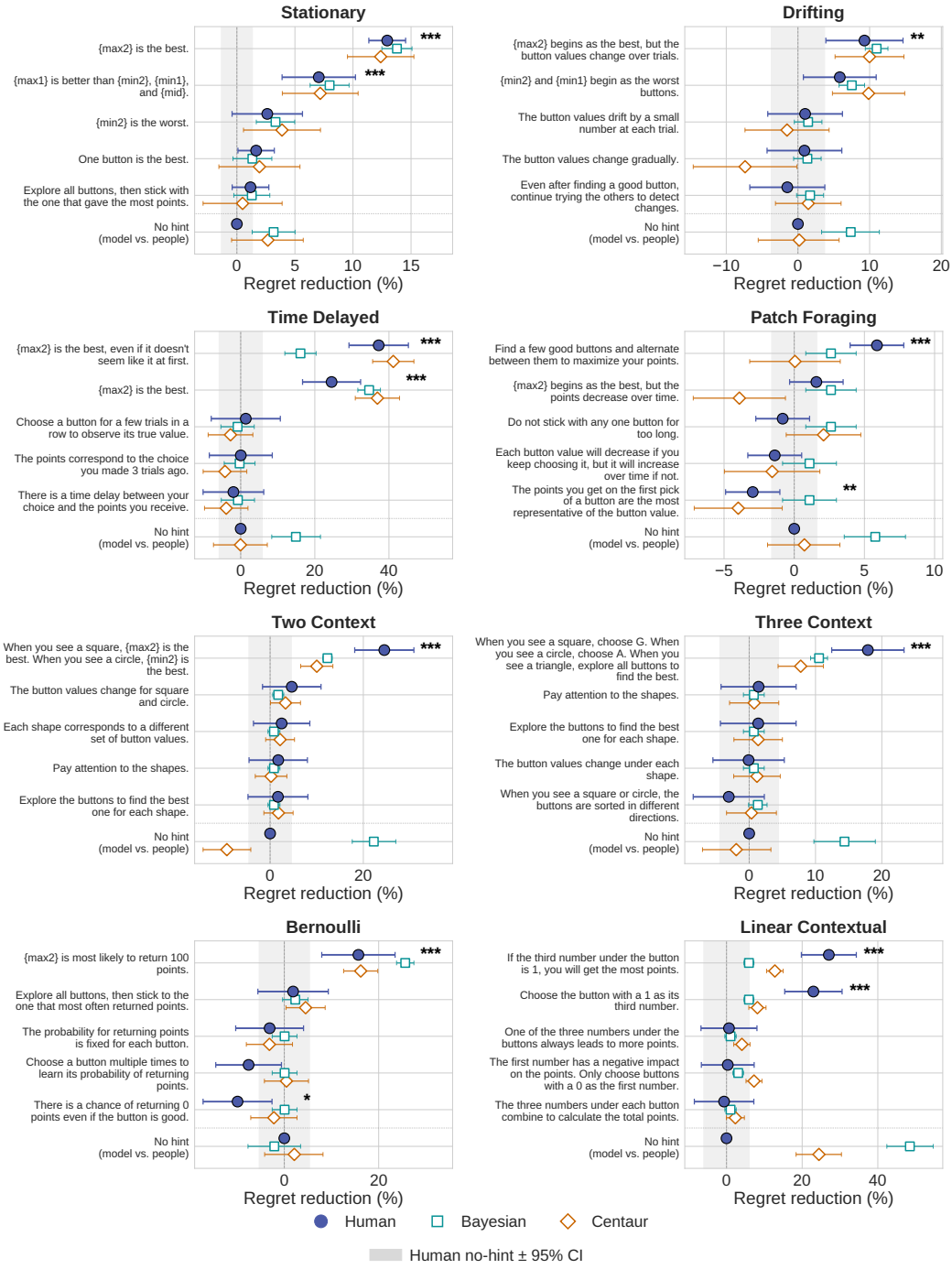


Figure 4. Hint effects vary by hint and task type for both humans and models. Each point represents the difference between the mean negative regret of the hint condition and the no hint baseline for that task (gray band, human no-hint mean $\pm$ 95% confidence interval), in percentages of the arm spread. The circles represent human participants, the squares the Bayesian model predictions, and the diamonds the Centaur model predictions. The bottom row of each panel ("No hint") shows each model's no-hint regret relative to the human no-hint baseline. Error bars show 95% confidence intervals from a linear mixed-effects model for people and from ordinary least squares with robust standard errors for the models (Section 4.5.2), and participant counts per hint are given in Appendix D and Table A4. The stars *, **, *** denote Benjamini–Hochberg adjusted two-sided $p < .05$, .01, and .001, respectively, for the human effects.

two context (7.1, [1.9, 12.2], $p = .015$), and *linear contextual* (10.4, [3.2, 17.6], $p = .013$) tasks. No reliable benefit was observed for the *drifting*, *patch foraging*, *three context*, or *Bernoulli* tasks on the aggregate level.

2.1.2 Hints modulate performance differently in each task type

This aggregate view, however, collapses across individual hints that varied substantially in information content. We therefore decompose the effect of language on participant performance by task and individual hint, relative to the no-hint baseline for that task (purple in Fig. 4; details in Section 4.5.2), and defer the model comparisons to Section 2.2.

Fig. 4 shows that hints varied in their effects on task performance. The hints which directly contributed information about the arm values most consistently improved participant performance. As many of these value-based hints identified the best arm (e.g. "{max2} is the best."), they eliminated the need to search and reduced the decision problem to simple exploitation. The strongest effect appeared in the *time delayed* environment, where the k-trial reward delay makes learning from experience challenging. In this case, participants given "{max2} is the best." achieved a reduced regret of 24.5 percentage points ($[16.7, 32.4]$, $p < .001$). Notably, the hint was even more helpful to participants (reduced regret by 37.2 percentage points, $[29.2, 45.2]$, $p < .001$) when it additionally referenced the task structure ("{max2} is the best, even if it doesn't seem like it at first."). Large value-based hint effects were similarly observed in the *stationary* task ("{max2} is the best."), where regret reduced by 12.9 percentage points ($[11.4, 14.5]$, $p < .001$), and in the *Bernoulli* task ("{max2} is most likely to return 100 points."), where regret reduced by 15.7 percentage points ($[7.9, 23.5]$, $p < .001$).

Because some tasks shared the same hint, we could also see that a hint's effect varied with the task type even under fixed language. "{max2} is the best" was roughly twice as beneficial in the *time delayed* environment as in the *stationary* one. This suggests that the amount of uncertainty a hint can resolve depends on the task structure and how much experience is required to learn its parameters. When beliefs converge slowly, such as with delayed feedback, language may accomplish more inferential work.

Next, hints did not only substitute for evidence on arm values, but also guided learners toward more efficient search strategies. In the *patch foraging* task, the hint "Find a few good buttons and alternate between them to maximize your points" reduced average regret by 5.9 percentage points ($[4.0, 7.8]$, $p < .001$). Since the optimal strategy in a depleting-rewards environment is to rotate among top choices rather than exploit a single arm, participants evidently extracted the policy-level information from the hint and adapted their behavior accordingly.

Finally, in some cases, participants given certain hints performed significantly worse than those given no hint. For example, in the *patch foraging* task, the hint "The points you get on the first pick of a button are the most representative of the button value" also reliably increased regret (hint effect of -3.0 percentage points, $[-4.9, -1.0]$, $p = .009$). As all hints in the IBT are designed to be helpful and to strictly contribute positive information, this demonstrates a case where ostensibly helpful linguistic information can actively mislead or distract from experiential learning.

2.1.3 Providing hints after experience alters their effect

To further investigate how people weigh linguistically-provided priors against accumulated reward experience, Study 2 varied the trial (0, 4, 8, or 12) at which a hint appeared relative to accumulated experience. The trial-0 conditions reproduced the direction and approximate magnitude of the beneficial effects in Study 1 for both task types (trial 0 in Fig. 5b,c), although they partly share games with Study 1 (Appendix D).

For Study 2, the hint effects were computed only on post-onset trials. Fig. 5a shows where each onset falls in the course of learning, and Fig. 5b shows that the benefit of a hint can be modulated by when it is delivered. Most hints had small effects at every onset, but five of the 40 hint-by-onset effects were reliable. In the *stationary* task, hints helped participants only when introduced at the start (e.g. "{max2} is the best" reduced regret by 11.3 percentage points at trial 0, $[8.6, 14.0]$, $p < .001$; Fig. 5c1). The hint "{max1} and {max2} are better than the others" helped at trial 0 (5.6, $[2.9, 8.3]$, $p < .001$) but had slightly negative, non-significant

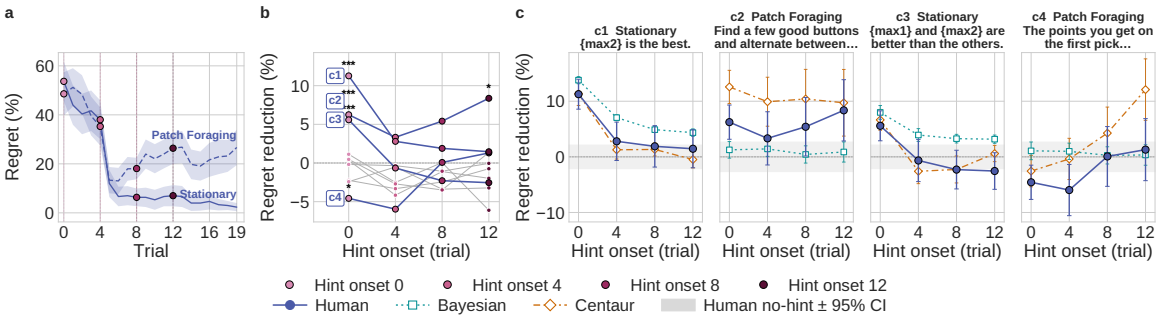


Figure 5. Humans integrate hints differently depending on prior experience. a) Average trial-by-trial regret of participants in Study 2 for the *stationary* (solid) and *patch foraging* (dashed) tasks, with markers at the four hint onsets (trials 0, 4, 8, and 12). Shaded regions are 95% confidence intervals across participants. **b)** Human hint effects for all 40 hint-by-onset conditions, as the normalized regret reduction relative to the no-hint baseline. Each line is one hint. **c)** Human, Bayesian model, and Centaur hint effects for the four highlighted hints as a function of hint onset. The gray band is the human no-hint baseline $\pm$ 95% confidence interval. All effects are computed over post-onset trials. Error bars show 95% confidence intervals from a linear mixed model for people and from ordinary least squares with robust standard errors for the models. The stars *, **, *** denote Benjamini–Hochberg adjusted two-sided $p < .05$, .01, and .001, respectively, for the human effects.

effects when it arrived at trials 4, 8, and 12 (Fig. 5c3). Only these two value hints changed reliably with onset, yielding reduced effects at every later onset than at trial 0 (all six contrasts $p \leq .043$). For all other hints, the differences between onsets were not reliable (all $p > .1$). This indicates diminishing returns, as earlier hints incorporate easily into learning before strong behavioral tendencies form, whereas later hints may conflict with already-formed strategies and fail to update people's behaviors.

One exception is that the *patch foraging* hint "The points you get on the first pick of a button are the most representative of the button value." showed a surprising but opposite effect. The hint undermined participant performance when introduced at trials 0 and 4 (reliably at trial 0, -4.6, $[-7.7, -1.5]$, $p = .029$) and had a neutral effect when introduced in later trials (Fig. 5c4). A possible explanation is that hints which change a participant's initial exploration strategy can prevent them from observing the evidence they need to learn, and become helpful only once some experience provides context for interpreting them. Consistent with this, the *patch foraging* hint "Find a few good buttons and alternate between them to maximize your points" helped at trial 0 (6.2, $[3.2, 9.3]$, $p < .001$) and remained beneficial when introduced as late as trial 12 (8.4, $[2.9, 13.9]$, $p = .029$; Fig. 5c2), after participants had experienced the rewards depleting.

In summary of the behavioral studies, people were able to use hint information to improve performance, but the effect of any given hint varied with task structure, strictly true information could be detrimental, and the helpfulness of some hints depended on how much experience had yet accumulated. A successful computational account must then capture these nuanced effects in fusing linguistic and experienced data to predict human choice.

2.2 Bayesian model and Centaur capture complementary facets of human learning

First, we fit the hierarchical Bayesian agent and variants of the Centaur language model to the trial-level human choices. We split the behavioral dataset from Study 1 by participant such that all trials from a given individual would appear in either the train or test set. For all models, we fit the parameters globally across all game types and individuals. Details of the model fitting and validation are described in Section 4.4. While all models outperformed random baselines (Fig. 7a; Section 2.4), Centaur achieved the highest likelihood score on held-out human choices (mean log-likelihood per trial of -1.25, followed by -1.30 for the Bayesian model), despite having no explicit model of the reward-generating process and relying entirely on in-context learning from the prompt.

2.2.1 Models reproduce hint benefits

Applying the best-fit parameters for both models, we forward-simulated each model to generate synthetic data for each of the IBT conditions (details in Section 4.4.1), which allows us to measure how well the models reproduce the basic behavioral signatures observed in Fig. 3. Without hints, the Bayesian model learned in all eight tasks and often reached lower regret than participants, while Centaur learned reliably in four of the eight tasks and showed no learning in *two context* or *linear contextual* (paired t-tests on no-hint games, Fig. 3a and Table A3). Consistent with the human participants, hints also induced performance improvements in most tasks for both models (Fig. 3b).

2.2.2 Bayesian belief updating supports human and Centaur choices

Beyond the aggregate view, Fig. 4 evaluates whether the models are able to replicate human behavioral patterns by comparing each model's predicted hint effect to the corresponding human effects for every task. The Bayesian model was generally able to predict the top two hints, most often contributing arm value information, which yielded the largest improvement in people. It additionally captured how semantically identical hints (e.g. "{max2} is the best.") altered human performance to different extents depending on the task (12.9 in *stationary* versus 24.5 in *time delayed* for people, and 13.8 versus 34.6 for the Bayesian model). One exception is the *time delayed* hint "{max2} is the best, even if it doesn't seem like it at first", which yielded a smaller predicted benefit from the Bayesian model than "{max2} is the best", the opposite of the human pattern. The Bayesian model's ability to both correctly identify the order and variability of hint effects across diverse tasks supports the case that the way people integrate language into learning approximates the mechanism within the hierarchical model.

Centaur also predicted the ordering of hints by helpfulness and the degree of each effect in humans to an impressive degree (Fig. 4), despite operating on latent states rather than formal beliefs. In cases where the hints harmed human performance, such as "The points you get on the first pick of a button are the most representative of the button value" in the *patch foraging* task, Centaur is also the only model to predict this interference effect. In contrast, our specific Bayesian model only benefits from hints (positive regret reduction), as they strictly contribute information.

Interestingly, Centaur largely reproduced the human sensitivity to hint timing, whereas the Bayesian model did not. Both models predict the human hint effect at trial 0 in the *stationary* task and smaller benefits for later onsets (Fig. 5c). Centaur closely follows the human decline, including the slightly negative effects of "{max1} and {max2} are better than the others" at later onsets (Fig. 5c1, c3), whereas the Bayesian model's predicted benefit shrinks but remains positive. For the *patch foraging* hint that Centaur had predicted would harm performance ("The points you get on the first pick of a button are the most representative of the button value."), both humans and Centaur show the negative effect subsiding when the hint is introduced in later trials, although Centaur predicts a large benefit at trial 12 that participants did not show (Fig. 5c4). For the "alternate" hint, Centaur predicts larger benefits than participants showed at every onset, and the Bayesian model predicts almost none (Fig. 5c2). The Bayesian model, which treats the hint as accurate evidence regardless of when it arrives, neither predicts harm at any onset nor captures how people change their weighting of language as experience accrues.

2.3 Centaur best reproduces fine-grained behavioral patterns

In Fig. 6, we go beyond how well our models can predict the overall performance changes induced by the hint toward how well they can predict the order of actions people made under each task and hint. We characterize the choice trajectories with three metrics, *switch rate*, *choice entropy*, and *exploitation onset* (defined in the Fig. 6 legend and Appendix E). Using the model simulations from Section 4.4.1, we averaged each metric across games to one value per task type and hint condition (48 per learner) and computed the Pearson correlations between human and model means separately for each metric and model.

Figure 6 shows that Centaur strongly reproduced human behavioral patterns across all three metrics ($r \approx 0.90$–0.95, $p < .001$). This is notable as IBT is absent from the Psych-101 fine-tuning dataset, and these met-

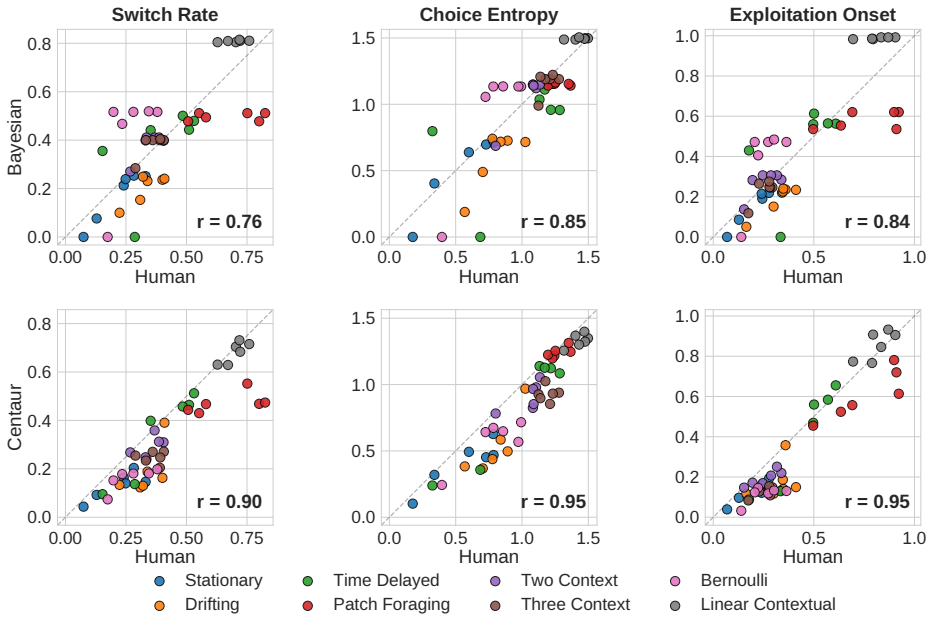


Figure 6. Centaur best reproduces patterns in human choice trajectories. Each panel shows a scatter plot of the condition-level means (48 points, one per task type and hint condition) for human participants (x-axis) versus the model simulated trajectories (y-axis). The three columns separately plot switch rate (the fraction of trials on which the agent changed its arm selection), choice entropy (the Shannon entropy of the empirical arm selection distribution over the full game), and exploitation onset (the normalized trial index of the first run of five consecutive identical choices), and the rows correspond to the Bayesian model and Centaur. The points are colored by task type. The dashed diagonal visualizes $y = x$, and the Pearson r values are shown in each panel to measure correlation agreement ($n = 48$ conditions, all $p < .001$).

rics were never optimized for during training, whose objective is only the trial-by-trial choice probabilities. The language model has thus captured qualitative aspects of how humans explore and learn across task and hint conditions in addition to their average performance.

The Bayesian model also showed strong alignment to humans across metrics ($r \approx 0.76$–0.85, $p < .001$), albeit achieving slightly weaker scores as compared to Centaur, especially in harder problems such as the *linear contextual* task. Likely, in environments where human beliefs are slow (requiring more experience) to converge, the Bayesian agent's mechanism for belief updating is too rigid in forcing abrupt and early changes in choice probability.

2.4 Model comparisons

The comparisons above evaluate each model against human behavior. To test whether the two models are complementary, we also compared them directly (Fig. 7). On held-out human choices, Centaur predicted the next choice best, with a negative log-likelihood of 1.25 nats per trial against 1.28 for CentaurIBT (Centaur fine-tuned to the IBT tasks, discussed further in Appendix F and Fig. A1), 1.30 for the Bayesian model, 1.43 for LLaMA, and 1.61 for chance (Fig. 7a). CentaurIBT fit slightly worse than Centaur despite fine-tuning on IBT choices, because fine-tuning sharpened its choice distributions (88.6% versus 79.0% of probability on the top-ranked arm), which is heavily penalized whenever a person chose a different arm.

The trajectory metrics show a similar difference (Fig. 7b). Averaged over task types, the Bayesian model's mean level on all three metrics was closer to people's than Centaur's, yet its values for individual tasks cross the human values, indicating that its agreement on the average arises from opposing errors in individual tasks. On the other hand, Centaur was slightly too exploitative on every metric, with a lower switch rate,

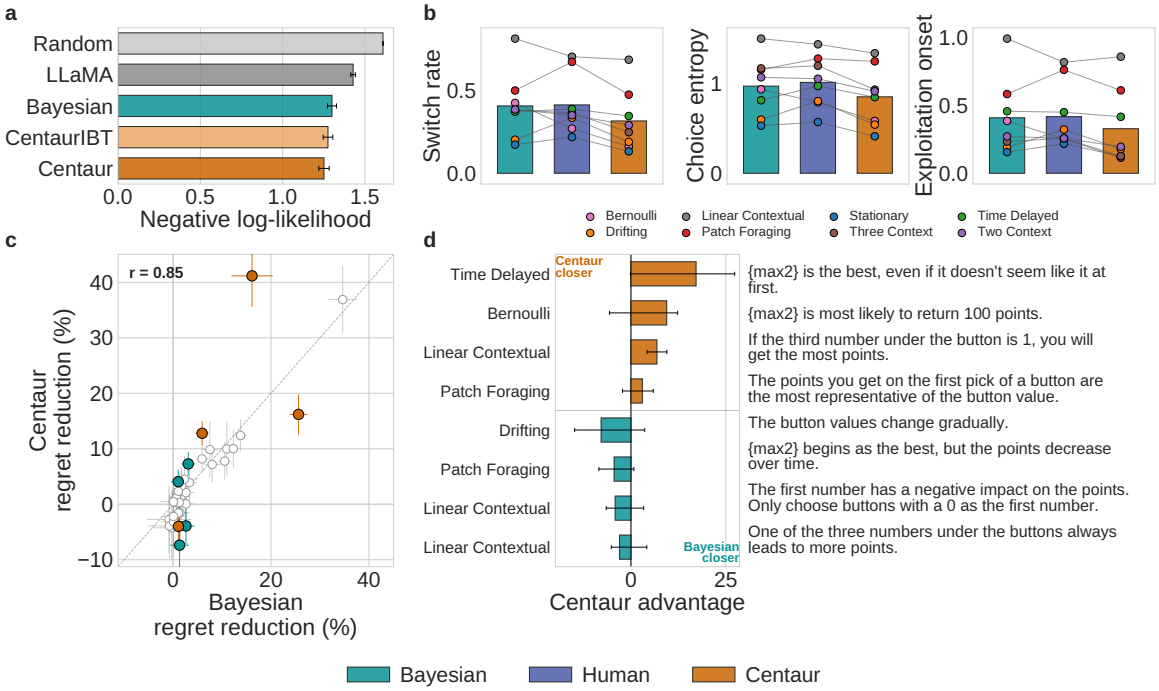


Figure 7. Bayesian and Centaur model comparison. a) Held-out negative log-likelihood per trial (lower is better) for Centaur, the fine-tuned CentaurIBT, the Bayesian model, the base LLaMA model, and a random baseline. Error bars are ± 1.96 standard errors across the five cross-validation folds. **b)** Switch rate, choice entropy, and exploitation onset for the Bayesian model, people, and Centaur. Bars show the mean across task types and points show each task type, with lines connecting the same task across learners. **c)** Predicted hint effects of the Bayesian model against those of Centaur for the 40 hints of Study 1, in percentage points of the arm spread. Filled points mark the eight hints for which the two models' predictions differ (the 95% confidence interval of their difference excludes zero and the difference is at least 1.5 percentage points), colored by the model closer to people. Open points mark the remaining 32. Error bars are 95% confidence intervals. **d)** For the eight hints in **c**, Centaur's advantage, the difference between the Bayesian model's and Centaur's absolute errors from the human hint effect (positive values mean Centaur is closer to people). Error bars are 95% intervals from a parametric bootstrap over the standard errors of the human, Bayesian, and Centaur estimates.

lower choice entropy, and earlier exploitation onset than people, but it preserved the ordering of the task types, so that its values run parallel to the human values.

The two models agreed closely on which hints help. Their predicted hint effects were strongly correlated across the 40 hints of Study 1 ($r = 0.85$), and for 32 of the 40 hints the two predictions did not differ reliably (Fig. 7c). Each model was also correlated with the human hint effects ($r = 0.73$ for the Bayesian model and 0.83 for Centaur). Despite their different constructions, both models therefore capture which hints help people and in which tasks, and this pattern is not what separates them.

On the eight hints where the predictions differed, each model was closer to people on four distinct conditions (Fig. 7d). Centaur's three largest advantages, of 7 to 17 percentage points, were all value hints identifying the best arm, which people acted strongly on and the Bayesian model underestimated. The Bayesian model's advantages were smaller, 3 to 8 percentage points, and came from hints describing the task's structure or dynamics. Of these eight differences, only the one for the *linear contextual* hint "If the third number under the button is 1, you will get the most points." was reliably different from zero on its own, so we read the split as a consistent pattern across hints rather than as eight separate findings.

The Bayesian model fits human choices slightly less well than Centaur. However, the Bayesian model, provided the generative structure of the eight task types and a hand-specified meaning function for each hint, gives a normative account of the integration of language and experience with only two fitted parameters (T and γ). Centaur, which receives only the text of the task, hint, and history and has one fitted parameter (β),

proves to be highly predictive of human choice patterns yet does not explain how these information sources combine. The two models thus serve as complementary tools for understanding how language shapes human experience and learning.

3 Discussion

Language and direct experience both contribute to the decisions we make. In this paper, we explore if there are coherent principles that guide their integration. We created a novel setting, the Instructed Bandit Tasks, to systematically study how people use language while learning from experience. The hints we studied occupy a different niche than the informationally equivalent descriptions studied in the description–experience gap [12] literature. In our setting, verbal descriptions were partial, arrived amid ongoing experience, and more closely resembled the advice and tips that accompany everyday learning.

We found that hints differed in helpfulness under different task types, hints that were strictly true sometimes made learners reliably worse, and the same hint could help at one point and have no effect at another depending on how much experience the learner had accrued. These effects suggest that natural language does not contribute a fixed amount of information into the learning process, but instead reshapes a learner's hypothesis space of the current task world.

From this view, the function of language in learning is to reshape the learner's world model and reweight their task representations. We attempted to capture this by constructing a hierarchical Bayesian agent which encodes the semantics of each hint as a secondary prior weighting of its beliefs. The Bayesian framing then offers a normative account of the integration process, as both linguistic and experienced evidence are cast into the common currency of probability and combined via Bayes' rule. The fitted Bayesian model captured which hints help and in which tasks, and was highly interpretable, allowing us to trace in time as a learner's beliefs changed about the nature of the task. One limitation of this approach is the manual curation of the hints and meaning functions, which ablations show account for most of the model's fit (Appendix G and Fig. A2). One way to address this is to use LLMs to translate arbitrary natural language inputs into these formal constraints, in the spirit of the model synthesis architectures [16], which would also scale the approach beyond the set of hints studied here.

A central contribution of our paper is to compare this Bayesian account to that of a fine-tuned large language model. Fit only by rescaling its choice distributions, Centaur assigned the highest likelihood to the held-out human choices and reproduced the fine-grained behavioral signatures of the human data with highest fidelity, despite never having been trained on IBT (though Psych-101 includes other bandit tasks) or optimized for these metrics. Most notably, while the normative Bayesian agent captured cases where language reduces uncertainty in a verifiable way, Centaur reproduced human errors, such as when informative hints harmed performance and when the benefit of a hint faded with experience. That a model trained only to predict the next token of human choice sequences is able to predict the cases in which humans will act suboptimally supports that Centaur has implicitly learned the behavioral regularities of how language governs experiential learning in humans.

These approaches capture opposite sides of growing debates about approaches to cognitive modeling. Our results suggest that they might not be so incommensurate. There was significant alignment between Centaur and the Bayesian model in terms of prediction ($r = 0.85$ across hint effects; Section 2.4). The two models can also be used together in the spirit of scientific regret minimization [26], in which the cases where a predictive model like Centaur succeeds and a normative model fails identify revisions for the latter.

This work opens several new avenues for future research. Future work could test hints that are conflicting or wrong (as everyday advice often is) and which an account of language as world-model constraint predicts should mislead learners in equally systematic ways (cf. ref. [10]). As everyday advice often comes from speakers of varying knowledge and intent rather than a trusted experimenter, the IBT can be extended to probe how source credibility and other aspects of social cognition impact how learners integrate language [4]. More broadly, almost all tasks begin with instructions which participants hear or read and which set the frame for the current experiment, yet cognitive models almost universally do not model the instructions directly,

instead assuming the model structure is in place already at the start of the task. This work starts to push at the boundaries of what we include in our psychological accounts of such tasks.

4 Methods

4.1 Behavioral experiment details

We conducted two behavioral studies with samples recruited on Prolific (https://prolific.com/). Study 1 evaluated participants across all task and hint pairings in the IBT, surveying the space of scenarios in which language can influence experiential learning. Study 2, designed to follow up on the findings of Study 1, manipulated the trial at which the hint appeared in order to test how accumulated experience changes the integration of linguistic information. Neither study was preregistered, and all analyses are exploratory.

We recruited 340 participants in Study 1 and 300 in Study 2. Participants self-reported their gender, which was not analyzed (Study 1: 176 men, 157 women, 3 preferred not to say, 1 other; Study 2: 140 men, 157 women, 2 preferred not to say, 1 other). Sample sizes were not determined by an a priori power analysis. A simulation-based sensitivity analysis (Section 4.6 and Appendix D) indicates that our design afforded 80% power to detect task-level hint effects of 2.1 to 10.6 percentage points of the arm spread depending on task. Minimum detectable effects for individual hints ranged from 2.2 to 13.3 percentage points, and for individual hint-by-onset conditions in Study 2 from 3.8 to 8.9.

We filtered out any participants who did not complete all games (2 people) or failed the instruction comprehension quiz more than twice (1 person), leaving N=337 and N=300 for analysis, respectively. We limited participation to adults from English-speaking countries (the US and UK) to ensure a baseline language competency. Participants were aged 18 to 80 (mean 42.7) in Study 1 and 19 to 79 (mean 40.9) in Study 2. We paid all participants at a rate of $15/hour with an additional bonus of up to $1.00, calculated proportionally to their performance in a randomly selected game trial (to avoid declining motivation later in the experiment). Participants completed the experiment in a median of 12.4 minutes in Study 1 and 9.1 minutes in Study 2. All procedures were approved by the NYU Institutional Review Board, and all participants provided informed consent before beginning the experiment.

In Study 1, participants played up to 15 games (see Appendix D for the composition of the Study 1 sample). Each game was assigned a random configuration of task type and hint, sampled from our pre-defined IBT dataset. In Study 2, each participant played 12 games. Each game was assigned one of two task types: *stationary*, in which experience alone rapidly identifies the best arm, and the *patch foraging* task, in which the non-stationary reward structure makes experiential learning substantially harder. This let us test the sensitivity of hint timing to task context while keeping the design tractable. Each game was also assigned a hint timing parameter, which determined the trial (of $[0, 4, 8, 12]$) on which the hint would appear on the screen. Both experiments used a within-subjects design, but because game conditions were assigned randomly, our analyses mix within- and between-participant comparisons. In each game, participants were instructed to maximize their point earnings over 20 or 40 trials, depending on game type (*two context*, *three context*, and *Bernoulli* games had double the number of trials due to task complexity). We kept the number of experiential trials relatively small in order to avoid overwhelming the information contributed by language.

The participants had to choose between 5 lettered buttons on the screen by clicking the buttons. Each time they made a choice, their action and observed reward were logged in a running history table under the buttons, to reduce the load on memory. In some games, a hint was provided at the top of the screen. In Study 1, the hint was always present to the participant, and in Study 2, the hint was made visible only after a predetermined number of choices, altering the amount of prior experiential learning the subject had before being given a hint. In the *two context* and *three context* cases, the participant also saw a shape (of circle, square, or triangle) on the screen at each trial to indicate which context they are in, and the context is additionally logged to the history table (Appendix H and Fig. A3). Both of our behavioral tasks were programmed using the Smile (https://smile.gureckislab.org/) project.

4.2 A Bayesian model of instructed learning

An IBT learner should best estimate the means of the Gaussian arms in order to make optimal selections and maximize their expected rewards. As rewards are noisy and observed only for the arm that was chosen, these estimates are inherently uncertain and built incrementally over trials. One way to model these beliefs is with a Bayesian RL framework, where the learner's knowledge is represented as a probability distribution over the arm means and is updated after every trial. Beyond providing a principled account of uncertainty, the Bayesian model allows for an additional language-informed prior to dually factor into the posterior belief.

At each timestep t, the learning agent chooses an action a_t and observes a reward r_t drawn from the task-specific reward function. The observation history is given by the full sequence of actions taken and rewards received, $h_t = (a_1, r_1, a_2, r_2, \ldots, a_t, r_t)$, and is sometimes annotated by additional contexts or features. In the absence of language, RL is purely evaluative [2], and the only learning signal comes from these observed rewards. In this case, we can use Bayes' Rule to represent the agent's posterior distribution over the N arm means given h_t as:

$$\underbrace{P(\mu_{1:N} \mid h_t)}_{Posterior} \propto \underbrace{P_0(\mu_{1:N})}_{Prior}\underbrace{P(h_t \mid \mu_{1:N})}_{Likelihood} \tag{1}$$

Under total uncertainty, the prior $P_0(\mu_{1:N})$ is broad, and we use a Normal prior centered on the typical arm value (Appendix I). In order to explore effectively, the agent must sample actions via Thompson sampling and iteratively update its beliefs to optimize the likelihood of the observations [14, 27].

In both real world settings and in the IBT, however, the learner faces an additional challenge of determining which problem structure they are learning under before estimating the parameters of the problem. We thus introduce a hierarchical Bayesian model which maintains uncertainty over both task types and parameters. Letting $\tau \in \{$*stationary*, *drifting*, *two context*, *three context*, *time delayed*, *patch foraging*, *Bernoulli*, *linear contextual*$\}$ denote the task type and θ_τ be the task specific parameters, the posterior of the hierarchical Bayesian model can be represented as:

$$P(\tau, \mu_{1:N}, \theta_\tau \mid h_t) \propto \underbrace{P(\tau)}_{Type}\underbrace{P(\mu_{1:N}, \theta_\tau \mid \tau)}_{Prior}\underbrace{P(h_t \mid \tau, \mu_{1:N}, \theta_\tau)}_{Likelihood} \tag{2}$$

The likelihood $P(h_t \mid \tau, \mu_{1:N}, \theta_\tau)$ encodes the generative model for each type. We use closed-form conjugate updates (Normal-Normal for most of the tasks) to compute this term. For types with unknown hyperparameters, the model uses Monte Carlo marginalization by sampling θ_τ from the current type-specific priors and averaging over the resulting likelihoods. After each trial, the hierarchical Bayesian agent updates its joint beliefs over task types and parameters, $P(\tau, \theta_\tau \mid h_t)$. Implementation details for each agent type are given in Appendix I. We additionally verify that the Bayesian hierarchical agent can correctly recover each of the task types in Appendix I and Fig. A4. To test which components of the agent account for its fit to human choices, we also fit ablated variants that drop the hint weight, replace the hierarchy over task types with a single stationary bandit, or fix the task type to its true value (Appendix G).

A key challenge is how to map the information from the natural language hints onto the learning problem. As a motivating example, consider a learner given the hint "The value of arm 1 is greater than 25". If the learner had first sampled arm 2 and observed a reward of 100, they would immediately recognize arm 2 to be a good choice, whereas if they had sampled arm 3 and received a reward of 0, they might be tempted to exploit arm 1 instead. The key point being, information contained in the hint, combined with individual experience, can rapidly shift choice behavior. Thus, we propose that the learning agent may use the information from the hint as a secondary Bayesian prior [21–23] to alter its beliefs over the task representations. In particular, this term, representing the natural language information, should remain distinct from the prior over tasks and over parameters (these reflect something about the learner's understanding of the games or the world at large), combine with the observations, and additionally inform the posterior beliefs.

To formally account for this inference at the computational level of analysis, we begin by considering all possible stationary bandits present to the learner. In the stationary N-armed bandit setting, the agent is tasked

with estimating the mean of each of the N Gaussian distributions corresponding to the arms. The parameter space that captures all possible arm mean configurations can then be defined as $\Theta = \mathbb{R}^N$. A natural language hint, l, places constraints on this space by essentially ruling out particular combinations of parameters from the possible worlds. To capture the semantics (i.e., which possible worlds are retained or excluded given a hint), we construct a "semantic meaning function" f_l, which is a mapping from the task parameterization to some real number indicating truth value [4, 28]. More concretely, we define $f_l : \Theta \to \mathbb{R}_{\leq 0} \cup \{-\infty\}$. The extended real-valued functions allow us to capture categorically true or false hints, represented by 0 or $-\infty$ respectively, as well as graded similarity judgments between parameter values in a manner analogous to fuzzy logic used in control engineering [29]. A toy visualization of how the hints translate to meaning functions and to probabilistic weightings over possible worlds is shown in Fig. 2a. The meaning functions of the Bayesian model were fixed before Studies 1 and 2 were run.

While the first two examples in Fig. 2a visualize the weighting functions for value-based hints, other hints might weight the agent's belief over the task types (as in the third example), the sampled parameters, or the agent's action and reward history. For example, a hint advising the agent to continue trying other arms throughout the game will use the trial time since the last selection of an arm as a weighting to upsample environments in which the best arm is one that the agent has not chosen recently. In this way, all types of hints in the IBT dataset change the agent's likelihoods for selecting possible environments, thereby constraining the agent to only worlds consistent with the hint. We can then adapt equation (2) by adding the meaning function weights as a secondary prior:

$$P(\tau, \mu_{1:N}, \theta_\tau \mid h_t) \propto \underbrace{P(\tau)}_{\textit{Type Prior}} \underbrace{P(\mu_{1:N}, \theta_\tau \mid \tau)}_{\textit{Parameter Prior}} \underbrace{P(h_t \mid \tau, \mu_{1:N}, \theta_\tau)}_{\textit{Likelihood}} \underbrace{\exp\{\gamma f_l(\tau, \mu_{1:N})\}}_{\textit{Hint Weight}} \tag{3}$$

In equation (3), $\exp(\gamma f_l(\tau, \mu_{1:N}))$ introduces the language dependency on the agent's posterior. The exponential maps the hint output (in $(-\infty, 0]$) into the probabilistic space for compatibility, and γ adds flexibility to the model by allowing variation in the weight of language on the learning process. Because hints that rule out possible worlds outright assign them a value of $-\infty$, and hence a weight of 0 (and a value of 0, or a weight of 1, to those consistent with the hint), γ only changes the influence of soft hints, such as those that weight samples by the agent's choice history. Algorithmically, the agent first computes the experience-only posterior, samples a large batch of K hypothetical environments from this standard posterior and passes them through the meaning function to evaluate their compatibility with the linguistic hint. Finally, the agent resamples a single environment from this weighted batch via importance sampling [30], selects the optimal action in that specific sampled environment with Thompson sampling [14], and executes the action to observe the next reward. Over the learning process, the agent maintains hierarchical beliefs to jointly infer task type and parameters. The language-informed hierarchical Bayesian RL agent is able to model the optimal learner through all of the problems in the IBT. In each scenario, the agent is provided no hint or one of the hints corresponding to the task type.

4.3 Centaur on IBT

To assess whether state-of-the-art large language models harness the information from linguistic hints in sequential decision-making tasks, we evaluate variants of the Centaur model on IBT. Centaur is a large language model fine-tuned on Psych-101, a large-scale dataset of past psychological experiments translated into text-based prompts [25]. Psych-101 includes a number of multi-armed bandit experiments, so Centaur has been trained on human choices in related paradigms, but the IBT tasks and hints are not part of it. The fine-tuning objective was to align the base model, Llama 3.1 70B [24], to human behavior by matching token likelihoods to human choices. One claim is that Centaur can predict human behavior for new behavioral tasks given only the relevant task instructions. We were interested in evaluating it on IBT to carefully measure the impact that different hints would have on prediction behavior for a model that relies heavily on linguistic input. We linearized our instructions and tasks following the protocol in [25], making only minimal changes to the text provided in the behavioral experiment to fit the query structure in the Psych-101 training dataset.

To simulate each of the games, we iteratively passed as input the task instructions, trial relevant contexts, and history of past choices and rewards to the models. We then extracted the predicted logits over the five valid action tokens (integers 1–5) and normalized via softmax (with generation temperature of 0.8 [25]) to obtain a probability distribution over the arm choices. We ran the models on each task 100 times while randomly rotating the arm mean configuration to avoid numerical biases, then averaged the per-trial choice distributions across simulation runs to yield mean model probabilities for each task type, hint, and trial condition.

4.4 Fitting models to human choice

We fit each model to the choices of 337 participants using 5-fold cross-validation and a maximum-likelihood approach to compare how well each model accounts for human decision-making. For the hierarchical Bayesian agent, we find the softmax temperature T and the weight of the linguistic hint γ which maximize the likelihood assigned to the true human choices at each trial. First, we iteratively pass each action and reward from the human trials to the hierarchical Bayesian agent, simulate the belief update, then draw 500 weighted posterior samples over task type and parameters. We then count the Thompson sampling actions across arms to construct a distribution over all actions. For the language-informed hierarchical Bayesian agents, we additionally reweighted the posterior samples by the hint function raised to the power of γ to simulate the effect of the hint. The reweighted action counts then pass through a softmax function with temperature T to produce the model's choice probabilities for that trial. We precomputed these statistics for every trial, so that fitting T and γ required only closed-form likelihood evaluations over the cache. We optimized both parameters using L-BFGS-B via the multivariate optimizer function `scipy.optimize.minimize` [31], with bounds $T \in [0.1, 20.0]$, $\gamma \in [0.1, 20.0]$. In every fold, γ sat at its lower bound. Refitting with the lower bound widened to 0.001 left T unchanged to two decimals and changed the held-out log-likelihood by at most 0.0002 nats per trial, so the bound does not affect the reported fits.

For Centaur, we introduced a rescaling parameter T that rescales the LLM's predicted choice distribution by a factor of $\beta = T_{\text{ref}}/T$:

$$P(a) = \frac{p(a)^{\beta}}{\sum_{a'} p(a')^{\beta}}$$

$\beta > 1$ sharpens and $\beta < 1$ flattens the rescaled probability distribution over action tokens. We fit β by extracting the choice probabilities at each trial of each game and scaling the distribution by this factor using the bounded scalar optimization function in Scipy (`scipy.optimize.minimize_scalar` with bounds $\beta \in [0.1, 20.0]$). We additionally fine-tuned Centaur on a training subset of the human bandit choices (CentaurIBT) using Low-Rank Adaptation (LoRA) by training with a cross-entropy loss over the five arm tokens. Additional implementation details on fine-tuning as well as an analysis of how additional training sharpened the choice distributions are in Appendix F. Finally, we include a LLaMA baseline, the untuned Llama 3.1 70B model on which Centaur is built, scored on the same trials as Centaur, to isolate the contribution of Centaur's training on the Psych-101 dataset. The latter two models were fit with the same temperature procedure as Centaur. We evaluate all models on the held-out fold with the mean log-likelihood per trial, or the probability the model assigns to the human's actual choice (results in Section 2.4).

4.4.1 Model simulations

After the fitting procedure, we set the best-fit hint weight for the hierarchical Bayesian agent and temperature for Centaur and forward simulated each model to reproduce the behavioral dataset. For the Bayesian model, we ran 100 independent agent simulations per MAB type and hint condition on freshly sampled bandit environments. For Centaur, we prompted the model with the same text prompts used in the human task and sampled choices at each trial under the fitted temperature, yielding 100 simulated games per MAB type and hint condition. We then analyze the hint effects of each of these datasets as compared to the behavioral dataset with the procedure in Section 4.5.2.

4.5 Statistical analyses

We use two complementary analyses matched to two questions. Paired t-tests on early versus late trial regret verify that each task is learnable from experience alone, and linear mixed models, with participant random intercepts, estimate the task-level hint effects, the effect of each individual hint, and, in Study 2, how each effect changes with hint onset, while accounting for repeated measures and unequal games per condition.

4.5.1 Human learning t-tests

To verify that participants learned within each task type from experience alone, we used only no-hint games and compared mean regret in the first five trials to mean regret in the last five trials using paired t-tests. For each task type, we computed the average regret (the difference between the mean of the optimal arm and that of the chosen arm, expressed as a percentage of the game's arm spread) of the participants over these two windows. A positive difference between mean regret in early and late trials indicates improvement over the course of the game. $t : \frac{\text{mean difference}}{\text{SE of difference}}$ is the difference relative to the variability. We report two-sided p-values, adjusted with the Holm–Bonferroni procedure [32] across the eight tasks. All eight human tests remain significant after correction. Full results are in Appendix C.

4.5.2 Linear mixed model fitting

We used a linear mixed model to quantify the effect of each hint on task performance for both humans and model simulations. For people, we fit the model (neg_regret $\sim$ hint + $(1|\text{subject})$) separately for each task type to predict the negative mean per-trial regret, expressed as a percentage of the game's arm spread (higher is better) with hint condition as a fixed effect and with a participant random intercept to correct for individual variation. Task-level hint effects were estimated with the same model, treating all hinted games as a single condition. For the models, which lack individual variation, we fit the model (neg_regret $\sim$ hint) by ordinary least squares with heteroscedasticity-robust standard errors. For Study 2, the model was fit separately for each task type and hint onset, with both hinted and no-hint games restricted to post-onset trials and, at onsets 4, 8, and 12, the game's mean regret before the onset included as a covariate (there are no pre-onset trials at onset 0). Each mixed model was estimated by restricted maximum likelihood using the statsmodels library [33]. We report estimated differences, 95% confidence intervals, and two-sided p-values for each task type and hint combination. The p-values are adjusted for multiple comparisons with the Benjamini–Hochberg false discovery rate procedure [34], applied separately to the 8 task-level effects, the 40 hint effects of Study 1, and the 40 hint-by-onset effects of Study 2. Adjusted p-values and significance stars are reported for the human effects only. To test whether hint effects in Study 2 depended on onset, we compared each hint's effect at trial 0 with its effect at each later onset (trials 4, 8, and 12) using a Wald test on the difference between the two estimates, and adjusted the 30 contrasts with the Benjamini–Hochberg procedure. We draw upon this modeling framework throughout our behavioral analyses.

4.6 Sensitivity analysis

We conducted a simulation-based sensitivity analysis [35] to estimate the minimum detectable effect (MDE) [36, 37] of our experiment design at each level of inference reported in the paper: per-task aggregate hint effects (Fig. 3b), per-hint effects within task (Fig. 4), and per-timing effects in Study 2 (Fig. 5). For each, we fit a reduced linear mixed model (neg_regret $\sim$ hint + $(1 \mid \text{subject})$) to the corresponding data to estimate the between-participant standard deviation τ and residual standard deviation σ. Holding fixed the number of participants, number of games per participant, and hint assignments for each condition, we then simulated synthetic datasets under a grid of true hint effects δ, refit the model to each, and tested for an effect using our reported convention (two-sided test at $\alpha = .05$, before the correction for multiple comparisons applied in Figs. 4–5).

Power at each δ was the proportion of 1,000 simulations in which the effect was detected, and the MDE was the value of δ at which power reached 80%, obtained by linear interpolation. Per-cell results are reported in

Appendix D. Task-level MDEs ranged from 2.1 percentage points of the arm spread (*stationary*) to 10.6 (*linear contextual*), with the two tasks played by the largest share of participants (*stationary* and *patch foraging*) resulting in the sharpest resolutions (MDEs of 2.1 and 2.3) owing to their larger samples. The per-hint MDEs ranged from 2.2 to 13.3, and per-hint-by-onset MDEs in Study 2 from 3.8 to 8.9. Of the four task-level hint effects (reported in Section 2.1.1), two exceeded their task's MDE and two (*two context* and *linear contextual*) fell just below it. At the per-hint level, 12 of the 13 effects meeting our reliability criterion exceeded their cell-specific MDE. The remaining one was detected despite falling below it, and we treat its magnitude as approximate (Appendix D).

4.7 Data availability

The anonymized behavioral datasets from both studies, including trial-level choices, reward, task and hint conditions, participant survey responses, and metadata are available at https://github.com/eysu35/IBT and archived on Zenodo at https://doi.org/10.5281/zenodo.23720109.

4.8 Code availability

Code for launching the online behavioral experiments, implementing the hierarchical Bayesian agent, simulating and fine-tuning Centaur, fitting both models, and replicating all analyses is available at https://github.com/eysu35/IBT.

Acknowledgements

We thank Pat Intara, Guy Davidson, Rheza Budiono, Solim LeGris, Misha O'Keefe, Maya Malaviya, and all other members of the Gureckis and Ho labs for helpful feedback on this manuscript. MKH was supported by the NSF under Grant No. 2434192. E.S. was supported by the Meta AIM PhD Fellowship.

Author contributions

E.S. conceived the project design, built the models, collected and analyzed the data, and wrote the manuscript. D.S. led the design, data collection, and analysis of Study 2. M.K.H. and T.M.G. supervised the research. All authors contributed to and approved the final manuscript.

References

[1] Rescorla, R. A. & Wagner, A. R. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In *Classical Conditioning II: Current Research and Theory* (eds Black, A. H. & Prokasy, W. F.) 64–99 (Appleton-Century-Crofts, 1972).

[2] Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT Press, 1998).

[3] Dayan, P. & Niv, Y. Reinforcement learning: The good, the bad and the ugly. *Current Opinion in Neurobiology* **18**, 185–196 (2008).

[4] Goodman, N. D. & Frank, M. C. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences* **20**, 818–829 (2016).

[5] Lupyan, G. & Bergen, B. K. How language programs the mind. *Topics in cognitive science* **8**, 408–424 (2016).

[6] Elman, J. L. On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive science* **33**, 547–582 (2009).

[7] Radulescu, A., Vong, W. K. & Gureckis, T. M. Name that state: How language affects human reinforcement learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 44 (2022).

[8] Weiss-Cohen, L., Konstantinidis, E., Speekenbrink, M. & Harvey, N. Incorporating conflicting descriptions into decisions from experience. *Organizational Behavior and Human Decision Processes* **135**, 55–69 (2016).

[9] Ashby, F. G., Alfonso-Reese, L. A., Turken, A. U. & Waldron, E. M. A neuropsychological theory of multiple systems in category learning. *Psychological review* **105**, 442–481 (1998).

[10] Doll, B. B., Hutchison, K. E. & Frank, M. J. Dopaminergic genes predict individual differences in susceptibility to confirmation bias. *The Journal of Neuroscience* **31**, 6188–6198 (2011).

[11] Mu, J. *et al.* Improving intrinsic exploration with language abstractions. In *Advances in Neural Information Processing Systems*, Vol. 35, 33947–33960 (2022).

[12] Hertwig, R. & Erev, I. The description-experience gap in risky choice. *Trends in cognitive sciences* **13**, 517–523 (2009).

[13] Colas, C. *et al.* Language and experience: A computational model of social learning in complex tasks. *ArXiv* **abs/2509.00074** (2025).

[14] Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* **25**, 285–294 (1933).

[15] Slivkins, A. Introduction to multi-armed bandits (2024). URL https://arxiv.org/abs/1904.07272. arXiv:1904.07272.

[16] Wong, L. *et al.* From word models to world models: Translating from natural language to the probabilistic language of thought (2023). URL https://arxiv.org/abs/2306.12672. arXiv:2306.12672.

[17] Lattimore, T. & Szepesvari, C. *Bandit algorithms* (Cambridge University Press, 2020).

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

[19] McClure, S. M., Berns, G. S. & Montague, P. R. Temporal prediction errors in a passive learning task activate human striatum. *Neuron* **38**, 339–346 (2003).

[20] O'Doherty, J. P., Dayan, P., Friston, K. J., Critchley, H. D. & Dolan, R. J. Temporal difference models and reward-related learning in the human brain. *Neuron* **38**, 329–337 (2003).

[21] Ho, M. K. & Gureckis, T. M. Learning from language and experience. In *Proceedings of the 2023 Conference on Cognitive Computational Neuroscience* (Oxford, UK, 2023).

[22] Su, E., Ho, M. K. & Gureckis, T. M. Integration of language and experience via the instructed bandit task. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 47 (2025). URL https://escholarship.org/uc/item/1j746305.

[23] Srinivasan, D., Su, E., Ho, M. K. & Gureckis, T. M. Timing of guidance shapes experiential learning. In *Proceedings of the 2026 Conference on Cognitive Computational Neuroscience* (NY, NY, USA, 2026).

[24] Dubey, A. *et al.* The Llama 3 herd of models (2024). URL https://arxiv.org/abs/2407.21783.

[25] Binz, M. *et al.* A foundation model to predict and capture human cognition. *Nature* **644**, 1002–1009 (2025).

[26] Agrawal, M., Peterson, J. C. & Griffiths, T. L. Scaling up psychology via Scientific Regret Minimization. *Proceedings of the National Academy of Sciences* **117**, 8825–8835 (2020).

[27] Russo, D., Roy, B. V., Kazerouni, A., Osband, I. & Wen, Z. A tutorial on Thompson Sampling. *Found. Trends Mach. Learn.* **11**, 1–96 (2017).

[28] Cresswell, M. Formal semantics. In *The Blackwell Guide to the Philosophy of Language* (eds Devitt, M. & Hanley, R.) 131–146 (Wiley-Blackwell, 2006).

[29] Zadeh, L. A. Fuzzy sets. *Information and Control* **8**, 338–353 (1965).

[30] Neal, R. M. Annealed importance sampling. *Statistics and Computing* **11**, 125–139 (2001).

[31] Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17**, 261–272 (2020).

[32] Holm, S. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* **6**, 65–70 (1979).

[33] Seabold, S. & Perktold, J. statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference*, 92–96 (2010).

[34] Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B* **57**, 289–300 (1995).

[35] Green, P. & MacLeod, C. J. SIMR: an R package for power analysis of generalized linear mixed models by simulation. *Methods in Ecology and Evolution* **7** (2016).

[36] Lakens, D. Sample size justification. *Collabra: Psychology* **8**, 33267 (2022).

[37] Bloom, H. S. Minimum detectable effects: A simple way to report the statistical power of experimental designs. *Evaluation Review* **19**, 547–556 (1995).

[38] Steyvers, M., Lee, M. D. & Wagenmakers, E. A bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology* **53**, 168–179 (2009).

[39] Whittle, P. Restless bandits: activity allocation in a changing world. *Journal of Applied Probability* **25**, 287–298 (1988).

[40] Speekenbrink, M. & Konstantinidis, E. Uncertainty and exploration in a restless bandit problem. *Topics in cognitive science* **7**, 351–367 (2015).

[41] Langford, J. & Zhang, T. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, Vol. 20, 817–824 (2007).

[42] Li, L., Chu, W., Langford, J. & Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 661–670 (2010).

[43] Schulz, E., Konstantinidis, E. & Speekenbrink, M. Putting bandits into context: How function learning supports decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **44**, 927–943 (2018).

[44] Joulani, P., György, A. & Szepesvari, C. Online learning under delayed feedback. In *Proceedings of the 30th International Conference on Machine Learning*, 1453–1461 (2013).

[45] Charnov, E. L. Optimal foraging, the marginal value theorem. *Theoretical Population Biology* **9**, 129–136 (1976). URL https://www.sciencedirect.com/science/article/pii/004058097690040X.

[46] Constantino, S. M. & Daw, N. D. Learning the opportunity cost of time in a patch-foraging task. *Cognitive, affective & behavioral neuroscience* **15**, 837–853 (2015).

[47] Niv, Y. *et al.* Reinforcement learning in multidimensional environments relies on attention mechanisms. *The Journal of Neuroscience* **35**, 8145–8157 (2015).

[48] Abbasi-Yadkori, Y., Pál, D. & Szepesvari, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, Vol. 24, 2312–2320 (2011).

[49] Hu, E. J. *et al.* LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations* (2022). URL https://arxiv.org/abs/2106.09685.

[50] Mitchell, T. J. & Beauchamp, J. J. Bayesian variable selection in linear regression. *Journal of the American Statistical Association* **83**, 1023–1032 (1988).

Appendix

Appendix A IBT task definitions

To ensure our findings generalize beyond any single problem structure, we constructed the Instructed Bandit Tasks (IBT) to span a wide array of bandit variants commonly studied in the reinforcement learning and cognitive science literature. The eight task types were chosen to systematically vary the computational demands placed on the learner. Each task type admits a different family of linguistic hints about its underlying structure, making the set well suited for studying how instructions interact with qualitatively different learning problems. Formal definitions are in Table A1.

Task	Params	Arm Values	Reward Function	Desc.
Stationary	σ: noise	$\mu_{1:N}$	$r_t \sim N(\mu^t_{a_t}, \sigma)$	Rewards sampled from Gaussian [2, 18, 38].
Drifting	σ: noise ϵ: drift	$\mu^{t+1}_{1:N} = \mu^t_{1:N} + N(0, \epsilon)$	$r_t \sim N(\mu^t_{a_t}, \sigma)$	Gaussian means drift over time [18, 39, 40].
Two Context	σ: noise c: context	$\mu^t_{1:N} = \begin{cases} \mu_{1:N}, c_t = 0 \\ \mu'_{1:N}, c_t = 1 \end{cases}$	$r_t \sim N(\mu^t_{a_t}, \sigma)$	Context c inverts the arm means [41–43].
Three Context	σ: noise c: context	$\mu^t_{1:N} = \begin{cases} \text{sort}(\mu_{1:N}), c_t = 0 \\ \text{rev}(\mu_{1:N}), c_t = 1 \\ \text{rand}(\mu_{1:N}), c_t = 2 \end{cases}$	$r_t \sim N(\mu^t_{a_t}, \sigma)$	Context c sorts the arm means differently [41–43].
Time Delayed	σ: noise k: delay	$\mu_{1:N}$	$r_t \sim N(\mu^t_{a_{t-k}}, \sigma)$	Rewards correspond to actions taken k trials ago [44].
Patch Foraging	σ: noise λ: decay	$\mu^{t+1}_{1:N} = \begin{cases} \mu^t_a \times e^{-\lambda}, a = a_t \\ \mu^t_a \times e^{\lambda}, a \neq a_t \end{cases}$	$r_t \sim N(\mu^t_{a_t}, \sigma)$	Rewards deplete for chosen arm and replenish over time [45, 46].
Bernoulli		$\theta_{1:N}$: probabilities	$r_t \sim \text{Bern}(\theta_{a_t})$	Rewards sampled from Bernoulli [14, 38].
Linear Contextual	θ: weights η: noise	$a_{1:N} \in \{0, 1\}^3$: features	$r_t = \langle \theta, a_t \rangle + \eta$	Features linearly combine to compute reward [47, 48].

Table A1. Summary of all Instructed Bandit Tasks. Each row gives the task name, hyperparameters, arm values, reward function, and description of the task.

Appendix B IBT dataset

Table A2. List of hints by task type.

Task type	Hints
Stationary	1. {max2} is the best. **V** 2. {min2} is the worst. **V** 3. Explore all buttons, then stick with the one that gave the most points. **P** 4. {max1} is better than {min2}, {min1}, and {mid}. **V** 5. One button is the best. **S**

Continued on next page

Table A2 – continued from previous page

Task type	Hints
Drifting	1. {max2} begins as the best, but the button values change over trials. **V**, **S** 2. The button values drift by a small number at each trial. **S** 3. Even after finding a good button, continue trying the others to detect changes. **P** 4. {min2} and {min1} begin as the worst buttons. **V** 5. The button values change gradually. **S**
Two Context	1. The button values change for square and circle. **S** 2. Explore the buttons to find the best one for each shape. **P** 3. When you see a square, {max2} is the best. When you see a circle, {min2} is the best. **V** 4. Pay attention to the shapes. **P** 5. Each shape corresponds to a different set of button values. **S**
Three Context	1. When you see a square, choose G. When you see a circle, choose A. When you see a triangle, explore all buttons to find the best. **V**, **P** (the button letters are literal, as the square and circle arm orders in the *three context* task were fixed across games) 2. Explore the buttons to find the best one for each shape. **P** 3. The button values change under each shape. **S** 4. When you see a square or circle, the buttons are sorted in different directions. **S** 5. Pay attention to the shapes. **P**
Time Delayed	1. {max2} is the best. **V** 2. {max2} is the best, even if it doesn't seem like it at first. **V** 3. Choose a button for a few trials in a row to observe its true value. **P** 4. The points correspond to the choice you made 3 trials ago. **S** 5. There is a time delay between your choice and the points you receive. **S**
Patch Foraging	1. {max2} begins as the best, but the points decrease over time. **V**, **S** 2. Each button value will decrease if you keep choosing it, but it will increase over time if not. **S** 3. Find a few good buttons and alternate between them to maximize your points. **P** 4. The points you get on the first pick of a button are the most representative of the button value. **S** 5. Do not stick with any one button for too long. **P**
Bernoulli	1. {max2} is most likely to return 100 points. **V** 2. Explore all buttons, then stick to the one that most often returned points. **P** 3. The probability for returning points is fixed for each button. **S** 4. There is a chance of returning 0 points even if the button is good. **S** 5. Choose a button multiple times to learn its probability of returning points. **P**

Continued on next page

Table A2 – continued from previous page

Task type	Hints
Linear Contextual	1. If the third number under the button is 1, you will get the most points. V 2. One of the three numbers under the buttons always leads to more points. S 3. The three numbers under each button combine to calculate the total points. S 4. Choose the button with a 1 as its third number. P 5. The first number has a negative impact on the points. Only choose buttons with a 0 as the first number. S

Key: V: Value-based hint — S: Structural hint — P: Policy hint

Appendix C Learning in each task

This section reports the full learning-verification statistics summarized in the Methods of the main text. Table A3 gives the per-task paired t-tests for human participants and both model simulations, computed on no-hint games only. Fig. 3 in the main text shows the corresponding learning curves and hint effects. The Bayesian agent learned in all eight task types (t range 8.22–21.50, all $p < .001$). Centaur learned significantly in four (*stationary*, *drifting*, *three context*, *Bernoulli*) with marginal or absent learning in the *time delayed*, *patch foraging*, *two context*, and *linear contextual* tasks.

Appendix D Sensitivity analysis

We did not preregister our experiments or determine sample sizes via an a priori power analysis. Thus, we conducted a simulation-based sensitivity analysis [35] to understand the minimum detectable effect (MDE), or the smallest true hint effect that the design would detect with 80% probability [37], of each condition in our experimental design. This provides a condition-specific benchmark against which detected effects can be interpreted [36].

D.1 Composition of the Study 1 sample

The Study 1 sample combines two batches. In the first, 159 participants each played 15 games, each assigned a random configuration of task type and hint sampled from the pre-defined IBT dataset. The second comprises 178 participants who completed the first build of the Study 2 experiment, in which every hint appeared at trial 0; we included their games with hints from the Study 1 hint set (10 per participant, all *stationary* or *patch foraging*). Because Study 2 drew its trial-0 conditions from the same sessions, 74 of these participants also contribute games to Study 2, and 740 games appear in both datasets.

D.2 Procedure

We computed MDEs at three levels, namely (a) task-level aggregate hint effects (Fig. 3b; 8 cells), (b) per-hint effects within task (Fig. 4; 40 cells), and (c) per-hint effects at each onset in Study 2 (Fig. 5; 40 cells). For each, we fit a linear mixed model (neg_regret $\sim$ hint $+$ (1 | participant)) to the subset of the data to derive the between-participant standard deviation $\hat{\tau}$ and residual standard deviation $\hat{\sigma}$.

We then simulated synthetic datasets with the same participants, number of games per participant, and hint assignments as the original games and generated outcomes with $y_{ij} = \delta \cdot \text{hint}_{ij} + u_i + e_{ij}$, where $u_i \sim \mathcal{N}(0, \hat{\tau}^2)$ is a participant-level offset, $e_{ij} \sim \mathcal{N}(0, \hat{\sigma}^2)$ is game-level noise, and δ is a planted true hint effect. We analyzed each synthetic dataset with the same mixed model and recorded whether the hint effect was detected (criterion below). For each condition, we swept δ over a 13-point grid spanning 0 to 1.25 times

Type	Diff	t	p	p_{Holm}	d_z	N
Human participants						
Stationary	38.23	31.07	$< .001$	$< .001$	2.04	232
Drifting	30.55	6.75	$< .001$	$< .001$	0.94	51
Time Delayed	11.16	2.37	.022	.043	0.34	48
Patch Foraging	23.61	13.75	$< .001$	$< .001$	0.91	229
Two Context	20.44	4.98	$< .001$	$< .001$	0.71	49
Three Context	17.39	3.47	.001	.004	0.54	41
Bernoulli	22.82	4.77	$< .001$	$< .001$	0.79	36
Linear Contextual	9.87	2.25	.031	.043	0.39	34
Bayesian agent						
Stationary	32.46	12.24	$< .001$	$< .001$	1.22	100
Drifting	34.10	14.70	$< .001$	$< .001$	1.47	100
Time Delayed	38.49	17.45	$< .001$	$< .001$	1.75	100
Patch Foraging	22.64	8.22	$< .001$	$< .001$	0.82	100
Two Context	45.29	21.50	$< .001$	$< .001$	2.15	100
Three Context	35.66	13.40	$< .001$	$< .001$	1.34	100
Bernoulli	35.43	17.67	$< .001$	$< .001$	1.77	100
Linear Contextual	33.43	20.37	$< .001$	$< .001$	2.04	100
Centaur						
Stationary	26.71	13.53	$< .001$	$< .001$	1.35	100
Drifting	31.24	11.11	$< .001$	$< .001$	1.11	100
Time Delayed	5.63	1.65	.101	.304	0.17	100
Patch Foraging	4.49	1.84	.069	.274	0.18	100
Two Context	-1.26	-0.38	.707	1.000	-0.04	100
Three Context	15.54	5.38	$< .001$	$< .001$	0.54	100
Bernoulli	25.09	9.26	$< .001$	$< .001$	0.93	100
Linear Contextual	0.35	0.15	.879	1.000	0.02	100

Table A3. Learning from experience alone, verified per task type with two-sided paired t-tests comparing mean regret in the first versus last five trials of no-hint games. Diff is mean regret in the first five trials minus the last five, in percentage points of the arm spread; positive values indicate learning. p_{Holm} is Holm–Bonferroni adjusted across the eight tasks within each learner. $d_z = t/\sqrt{N}$ is the standardized within-subject effect size. N denotes participants for humans and independent simulated games for the models.

the cell's total outcome standard deviation $\sqrt{\hat{\tau}^2 + \hat{\sigma}^2}$ to address the differences in regret scale across tasks. Power at each δ was the proportion of 1,000 simulations in which the effect was detected, using common random seeds across the grid. We interpolated the per-condition MDEs from the 80% crossing point on the power curves. All 88 cells reached 80% power within the grid.

D.3 Detection criterion

The main analyses report hint effects from linear mixed models, starring effects whose adjusted two-sided p-value is below .05. Across the $88 \times 13 \times 1{,}000$ model fits, our sensitivity simulations use the same mixed model with a two-sided Wald test at $\alpha = .05$ ($|z| > 1.96$) as the detection criterion, without the correction for multiple comparisons applied in the main analyses. The two-sided test addresses both beneficial effects and harmful hint effects like those reported in Figs. 4–5.

D.4 Results

Per-cell results for Study 1 are reported in Table A4, and the Study 2 cells are available in the code repository (Code availability, main text). First, the two task types played by the largest share of participants (*stationary* and *patch foraging*) had the sharpest task-level resolution (MDEs of 2.1 and 2.3 percentage points of the arm spread), while the six task types assigned to participant subsets required larger effects (5.9 to 10.6 percentage points), with *time delayed* among the highest owing to its substantially larger residual variability ($\hat{\sigma} = 23.8$). Second, per-hint cells required correspondingly larger effects (MDEs of 2.2 to 13.3 percentage points across

tasks), due to the smaller sample sizes per hint. Third, Study 2's per-hint-by-onset cells had MDEs of 3.8 to 8.9 percentage points, and all five significant Study 2 effects exceeded their MDE.

Two of the four task-level effects reported as reliable in the main text exceeded their task's MDE, while two fell just below it (7.1 observed against an MDE of 7.4 for *two context*, and 10.4 against 10.6 for *linear contextual*), and we accordingly treat their magnitudes as approximate. The MDEs also indicate that only large hint benefits were reliably detectable in the *drifting*, *three context*, and *Bernoulli* tasks, so the lack of aggregate hint effects (see Results in the main text) should not rule out the possibility of moderate effects. At the per-hint level, 12 of the 13 significant effects (starred conditions in Fig. 4) exceeded their specific MDE. The remaining one was still detected despite an observed magnitude below its 80%-power threshold, as true effects below the MDE may still be detected with lower probability. This implies that this estimate of a hint effect is more susceptible to magnitude overestimation, as when power is low, the detected experiments are more likely to have noise inflation. We interpret this effect, the harmful structure hint in the *Bernoulli* task (observed 9.9 vs. MDE 10.8), whose direction is supported ($p = .024$), as evidence of effect direction rather than precise effect size.

Table A4. Per-cell sensitivity analysis. We report the number of contributing participants (N; for per-hint cells, the participants who received that hint), hint/no-hint game counts, and the minimum detectable effect at 80% power (MDE) in percentage points of the arm spread. Fitted variance components underlying each MDE are available in the code repository (Code availability, main text), together with the per-cell MDEs for Study 2. Observed effects and their confidence intervals are reported in Fig. 4 (Study 1), Fig. 5 (Study 2), and the Results of the main text.

Cell	N	Games (H/NH)	MDE_{80}
(a) Task-level hint effects (Study 1, Fig. 3b)			
Stationary	317	775/240	2.13
Drifting	141	247/59	5.91
Time Delayed	139	239/53	10.53
Patch Foraging	316	1143/237	2.28
Two Context	143	232/60	7.44
Three Context	138	258/46	7.10
Bernoulli	136	230/42	9.04
Linear Contextual	138	265/37	10.58
(b) Per-hint effects within task (Study 1, Fig. 4)			
Stationary			
{max2} is the best.	217	220/240	2.29
{max1} is better than {min2}, {min1}, and {mid}.	41	47/240	4.46
One button is the best.	228	237/240	2.22
Explore all buttons, then stick with the one that gave the most points.	216	222/240	2.32
{min2} is the worst.	40	49/240	4.31
Drifting			
{max2} begins as the best, but the button values change over trials.	40	45/59	7.82
{min2} and {min1} begin as the worst buttons.	47	52/59	7.35
The button values drift by a small number at each trial.	42	54/59	7.55
The button values change gradually.	43	48/59	7.70
Even after finding a good button, continue trying the others to detect changes.	44	48/59	7.45
Time Delayed			
{max2} is the best, even if it doesn't seem like it at first.	47	54/53	11.75
{max2} is the best.	49	59/53	11.01
Choose a button for a few trials in a row to observe its true value.	31	32/53	13.25
The points correspond to the choice you made 3 trials ago.	37	44/53	12.28
There is a time delay between your choice and the points you receive.	42	50/53	11.90
Patch Foraging			
Find a few good buttons and alternate between them to maximize your points.	225	233/237	2.79
{max2} begins as the best, but the points decrease over time.	228	236/237	2.80
Do not stick with any one button for too long.	218	226/237	2.81
Each button value will decrease if you keep choosing it, but it will increase over time if not.	220	227/237	2.87

Cell	N	Games (H/NH)	MDE_{80}
The points you get on the first pick of a button are the most representative of the button value.	217	221/237	2.79
Two Context			
When you see a square, {max2} is the best. When you see a circle, {min2} is the best.	41	45/60	9.25
The button values change for square and circle.	41	46/60	8.99
Each shape corresponds to a different set of button values.	39	47/60	8.56
Pay attention to the shapes.	46	52/60	9.09
Explore the buttons to find the best one for each shape.	39	42/60	9.18
Three Context			
When you see a square, choose G. When you see a circle, choose A. When you see a triangle, explore all buttons to find the best.	49	56/46	7.86
Explore the buttons to find the best one for each shape.	41	47/46	8.16
Pay attention to the shapes.	41	44/46	8.12
The button values change under each shape.	48	56/46	7.67
When you see a square or circle, the buttons are sorted in different directions.	48	55/46	7.71
Bernoulli			
{max2} is most likely to return 100 points.	33	35/42	11.14
Explore all buttons, then stick to the one that most often returned points.	37	42/42	10.59
The probability for returning points is fixed for each button.	44	51/42	10.37
Choose a button multiple times to learn its probability of returning points.	50	56/42	10.23
There is a chance of returning 0 points even if the button is good.	42	46/42	10.77
Linear Contextual			
If the third number under the button is 1, you will get the most points.	44	53/37	10.61
Choose the button with a 1 as its third number.	41	49/37	11.24
The first number has a negative impact on the points. Only choose buttons with a 0 as the first number.	59	69/37	10.32
The three numbers under each button combine to calculate the total points.	38	42/37	11.51
One of the three numbers under the buttons always leads to more points.	49	52/37	10.84

Appendix E Fine-grained choice trajectory metrics

To assess fine-grained behavioral alignment between models and human participants, we computed three metrics from the sequence of arm choices $a_1, a_2, \ldots, a_T$ over a game of T trials.

Switch Rate Switch rate measures the proportion of trials on which the agent changed its choice relative to the previous trial:

$$S = \frac{1}{T-1} \sum_{t=2}^{T} \mathbf{1}[a_t \neq a_{t-1}] \tag{4}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Higher values indicate more exploratory behavior.

Choice Entropy Choice entropy measures the uniformity of arm selection over the full game using the Shannon entropy of the empirical choice distribution:

$$H = -\sum_{k=1}^{K} \hat{p}_k \ln \hat{p}_k \tag{5}$$

where K is the number of arms and $\hat{p}_k = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[a_t = k]$ is the empirical frequency of arm k across trials. Higher entropy indicates more uniform exploration across arms.

Exploitation Onset Exploitation onset measures how early an agent committed to a single arm, defined as the normalized trial index of the first occurrence of five consecutive identical arm choices:

$$E = \frac{1}{T} \min \{t : a_t = a_{t+1} = a_{t+2} = a_{t+3} = a_{t+4}\} \tag{6}$$

If there was not a run of five consecutive choices within the game, we set $E = 1.0$. Lower values indicate earlier exploitation.

Appendix F Fine-tuning Centaur on IBT

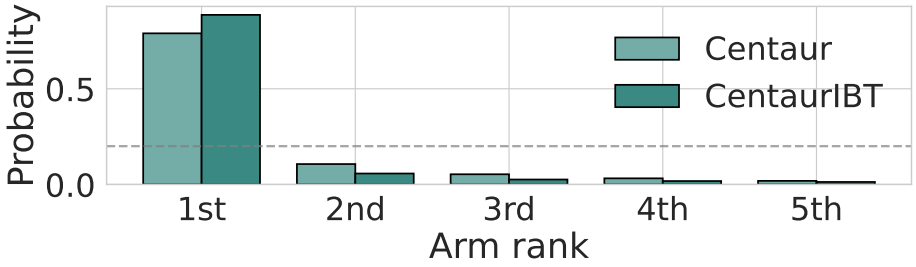


Figure A1. Fine-tuning sharpens Centaur's choice distributions. Mean probability assigned to each arm (from 1st, most probable, to 5th, least probable), averaged across all trials. CentaurIBT assigns higher probability to its top-ranked arm (0.89 vs. 0.79 for base Centaur). The dashed line indicates uniform baseline ($1/5 = 0.20$).

We fine-tuned Llama-3.1-Centaur-70B [25] on human bandit choices using Low-Rank Adaptation (LoRA) [49], implemented via Unsloth with 4-bit quantization. The model was loaded in 4-bit and adapted with LoRA applied to all attention and MLP projection matrices, with rank $r = 16$, $\alpha = 16$, and dropout $= 0.05$.

Fine-tuning followed the same participant-level 5-fold cross-validation as the other models (Methods, main text): in each fold, the model was fine-tuned on the training participants (about 269) and evaluated on the held-out participants (about 68), so that no participant's choices appeared in both. At each trial, the full game history up to that point was formatted into the same prompt used during simulation, and the model was trained to predict the human's next arm choice. Rather than standard cross-entropy over the full vocabulary, we used a choice-restricted loss, where logits were extracted only for the five arm-choice tokens, and label smoothing ($\epsilon = 0.05$) was applied to prevent overconfidence.

We fine-tuned Centaur for 2 epochs with AdamW (lr $= 2 \times 10^{-5}$, weight decay $= 0.01$), batch size 4 with gradient accumulation over 2 steps (effective batch size 8), gradient clipping at 1.0, and a cosine learning rate schedule with a 5% linear warm-up. Maximum sequence length was 1,024 tokens. Hyperparameter choices were largely set to default. Fine-tuning was run on a single NVIDIA H200 GPU with 128 GB of host memory and was allocated a 12-hour wall-time limit on an HPC cluster node.

Fig. A1 shows the rank-ordered average choice probability for the base Centaur model and the fine-tuned CentaurIBT model. We observe that fine-tuning on IBT sharpens the distribution. CentaurIBT concentrates 88.6% of probability mass on the top-ranked arm on average, as compared to the 79.0% for base Centaur. This overconfidence is the primary reason that CentaurIBT under-performs across IBT tasks in Fig. 7a. When the model's top-ranked arm does not match the human choice, the log-likelihood is severely penalized as the probability mass is low on all other arms. This indicates that either higher label smoothing or entropy regularization may be required during fine-tuning in order to improve model prediction.

Appendix G Ablations of the Bayesian model

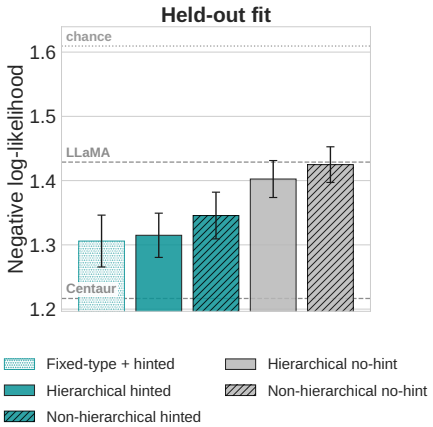


Figure A2. Language, not task-structure inference, accounts for most of the Bayesian model's fit. Held-out negative log-likelihood per trial (lower is better) for the full hierarchical Bayesian model and four ablated variants, averaged over the eight tasks with equal weight. Teal bars use the hint weight γ fitted to human choices; gray bars drop the hint term, so the agent learns from experience alone. Hatched bars replace the hierarchy over eight task types with a single stationary bandit (non-hierarchical), and the dotted bar gives the agent the true task type (fixed type). Error bars are ± 1.96 standard errors across the five cross-validation folds. Dashed lines show Centaur and the base LLaMA model for comparison, and the dotted line shows chance ($\log 5 \approx 1.61$).

We asked which components of the Bayesian model account for its fit to human choices and for its predicted hint effects. To do so, we fit ablated variants of the model with the same five-fold cross-validation and fitting procedure as the full model (Methods, main text), changing only the ablated component.

G.1 Ablated variants

- **No hint.** Drops the hint weight from equation (3) entirely, including for hard hints, and fits only the softmax temperature. The agent learns from experience alone.
- **Non-hierarchical.** Replaces the hypothesis space of eight task types with a single stationary bandit. Value and history hints still act on the sampled arm means, but hints about the task type or about parameters a stationary bandit lacks (e.g., the delay or the linear-contextual features) have no effect.
- **Fixed type.** Gives the agent the true task type, so it infers only the parameters within that type. Hints that act only on the task type have no effect.
- **Shuffled hint.** A control for the content of each hint. Each hinted game is scored with the meaning function of a different hint from the same task, one the participant did not see but that is still true of the game. We cycled through each task's hint list, giving four shuffled variants.

G.2 Fitting and comparison

Held-out negative log-likelihood is averaged over the eight tasks with equal weight, because *stationary* and *patch foraging* have about twice as many participants as the other tasks. (Averaged over trials instead, as in the main text, the full model's value is 1.300 nats per trial.) Differences between variants are tested with paired t-tests over participants. Because the task-weighted means and the per-participant differences weight the data differently, the gap between two task-weighted means need not equal the mean per-participant difference.

In every hinted variant, the hint weight γ sat at its lower bound of 0.1 in all five folds. This is expected: γ scales only the soft, history-based hints (Methods, main text), and the fit prefers them as weak as allowed. Widening the bound to 0.001 gave $\gamma = 0.001$ in every fold, left T unchanged to two decimals, and changed the held-out negative log-likelihood by at most 0.0002 nats per trial. The bound is therefore immaterial.

G.3 Results

First, we find that language accounts for most of the fit (Fig. A2). Removing the hint weight raised the held-out negative log-likelihood from 1.315 to 1.403 nats per trial (per-participant difference 0.11 [0.10, 0.12], $p < .001$). The hint-blind model also predicted no hint effects: its correlation with the human per-hint effects was $r = 0.09$, against 0.73 for the full model.

Next, we verified that hint benefits depend on their task content. Scoring games with a different, still-true hint from the same task fit almost as poorly as no hint (1.383–1.391 nats), and the predicted hint effects no longer tracked those of people (r between -0.25 and 0.25).

Removing the hierarchy cost very little overall (1.346 nats; per-participant difference 0.007 [-0.001, 0.015], $p = .086$), but this near-zero difference hides two opposing effects. The hierarchy fit choices better in tasks whose structure a stationary learner cannot represent (*three context*, *time delayed*, and *linear contextual*) and worse in tasks that people treated as simple bandits (*Bernoulli*, *drifting*, and *patch foraging*), where uncertainty over task type made the agent explore more than people did. Without the hierarchy, model correlation with human effects also dropped from 0.87 to 0.05 for policy hints and from 0.43 to 0.12 for structure hints, while the correlation for value hints did not drop (0.49 and 0.67).

Finally, the skyline Bayesian model is task-specific. In this case, giving the agent the true task type indeed improved fit (1.306 nats; per-participant difference 0.011 [0.008, 0.014], $p < .001$) and the correlation with human hint effects ($r = 0.79$), at the cost that hints acting only on the task type have no effect.

Appendix H Behavioral experiment interface

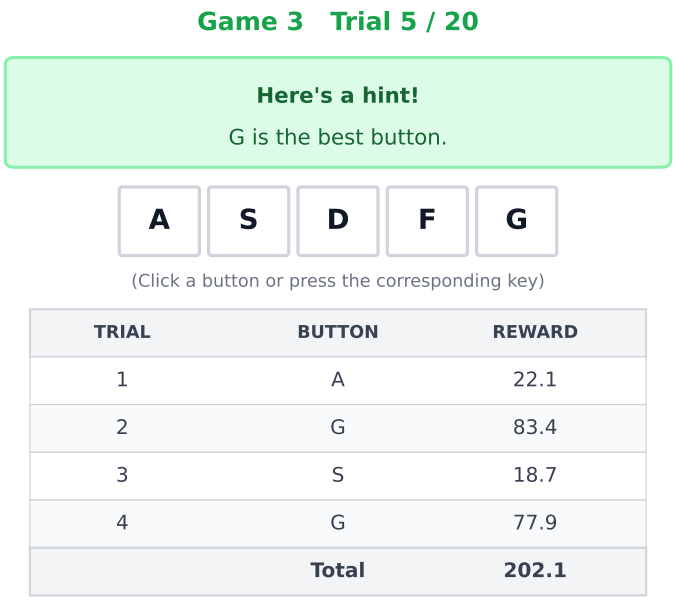


Figure A3. Mid-game screenshot of the experiment interface. Participants chose between five buttons (A/S/D/F/G) on each trial to maximize their total rewards. In hint conditions, a hint was displayed at the top of the screen. A running history of past choices and rewards was shown below the buttons.

Appendix I Implementation details of hierarchical Bayesian agents

The hierarchical model maintains a posterior $P(\tau, \theta_\tau \mid h_t)$ over eight task types. Each type implements a conjugate or Monte Carlo likelihood. All types use a shared Normal-Normal arm mean prior with $\mu_0 = 45$, $\sigma_0 = 20$, which approximates the mean and spread of the given arm values.

I.1 Computing type specific likelihoods

Stationary The agent runs a closed-form Normal-Normal update. At each trial t, the predictive variance for arm a is $\sigma_a^2 + \sigma_{\text{obs},a}^2$, where σ_a^2 is the current posterior variance of arm a's mean, and $\sigma_{\text{obs},a}^2$ is the known observation noise, which is set to the bandit arm's standard deviation. After observing reward r, the posterior precision is updated as $\lambda'_a = \lambda_a + 1/\sigma_{\text{obs},a}^2$ and the mean as $\mu'_a = (\mu_a \lambda_a + r/\sigma_{\text{obs},a}^2)/\lambda'_a$.

Drifting For *drifting* bandits, the agent uses a Kalman filtering framework [40] to track non-stationary means by marginalizing over a distribution of drift rates $\epsilon \sim \mathcal{N}(\mu_\epsilon, \sigma_\epsilon^2)$ (prior $\mu_\epsilon = 2$, $\sigma_\epsilon = 1$) and growing the posterior variance by the sampled ϵ^2 at each timestep to enforce a recency bias on observations. The log marginal likelihood is estimated via Monte Carlo over 10 sampled ϵ values. The posterior belief over ϵ is updated via importance-weighted moment matching on the samples.

Time delayed The agent maintains a categorical distribution over delay values $k \in \{1, 2, 3, 4, 5\}$ with uniform prior. For each candidate k, it computes the Normal-Normal log marginal likelihood by pairing each observed reward at trial t with the action taken at trial $t - k$. The posterior $P(k \mid h_t)$ is updated via Bayes' rule.

Patch foraging The agent models a deterministic transition where arm means deplete when chosen and replenish otherwise by marginalizing over a change rate $\lambda \sim \mathcal{N}(\mu_\lambda, \sigma_\lambda^2)$ with prior $\mu_\lambda = 0.2$, $\sigma_\lambda = 0.08$, using 40 Monte Carlo samples. After observing a chosen arm a, its mean is multiplied by $e^{-\lambda}$ and all other arms are multiplied by $e^{+\lambda}$, capped at a running estimate of each arm's equilibrium mean, updated online from observations. The posterior over λ is again updated via importance-weighted moment matching.

Two context and three context The agent maintains independent Normal-Normal arm mean posteriors for each context (2 or 3, respectively). Only the beliefs associated with the current context index (provided as side information from the trial history) are updated at each trial. The log marginal likelihood sums over context-appropriate observations only.

Bernoulli For *Bernoulli* bandits, the agent performs a Beta-Bernoulli conjugate update to estimate the success probabilities of each arm, with prior $\text{Beta}(\alpha_0, \beta_0) = \text{Beta}(1, 1)$ per arm. Rewards are in $\{0, 100\}$, and any other observed reward sets the log likelihood to $-\infty$, falsifying the Bernoulli hypothesis. Wins increment α_a, and losses increment β_a.

Linear contextual Finally, for the *linear contextual* bandits, the posterior is a multivariate normal over the weights of the linear model. The agent performs standard Bayesian linear regression [50] to maintain a covariance matrix for the 3-dimensional features, with a Gaussian prior $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, 100\mathbf{I})$. After each trial, the posterior is updated in closed form from the observed feature and reward pairs. The expected reward for an arm is then the dot product of its features with the sampled weights.

I.2 Updating posteriors and selecting actions

After computing log likelihoods $\ell_\tau = \log P(h_t \mid \tau)$ for each type, the type posterior is updated as:

$$P(\tau \mid h_t) \propto P(\tau) \exp(\ell_\tau)$$

with uniform prior $P(\tau) = 1/8$ at $t = 0$, using log-sum-exp normalization for numerical stability.

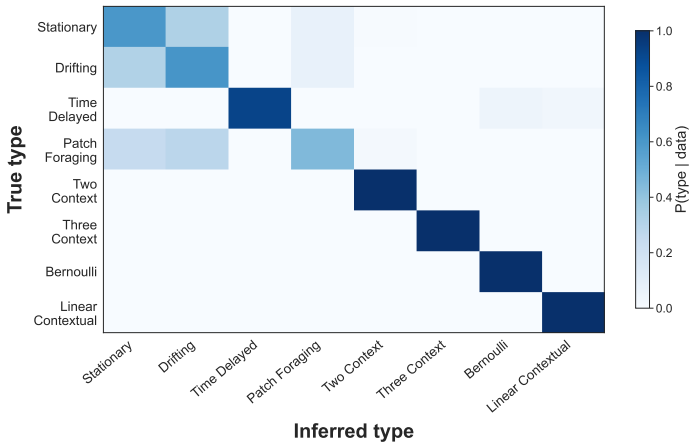


Figure A4. The hierarchical Bayesian agent correctly identifies most task types. Rows indicate the true task type used to generate the data. Columns show the agent's mean posterior $P(\text{type} \mid \text{data})$ after observing one game under random arm selection, averaged over 20 simulated games per type. Darker colors correspond to higher probability assignments.

I.3 Action selection with and without hints

The agent uses Thompson sampling to select actions. At each trial, $K = 500$ samples $(\tau^{(i)}, \mu_{1:N}^{(i)}, \theta_\tau^{(i)})$ are drawn from the current posterior $P(\tau, \mu_{1:N}, \theta_\tau \mid h_t)$. When a hint l is present, each sample is reweighted by:

$$w^{(i)} \propto \exp\left(\gamma \cdot f_l\left(\tau^{(i)}, \mu_{1:N}^{(i)}\right)\right)$$

where f_l is the hint meaning function defined in the Methods of the main text and γ is the fitted hint strength parameter. A single hypothesis is then resampled from this weighted population (or uniformly if no hint) via importance sampling [30], and the agent's action is the greedy arm, $a^{(i)} = \arg\max_a \mu_a^{(i)}$, of that resampled hypothesis. The resulting reward is observed and the trial data (a, r) is appended to the history h_t before the next belief update.

I.4 Type recovery

We conducted a type recovery analysis using the actual parameters from the behavioral experiment (five arms with means $[10, 20, 40, 60, 80]$ and $\sigma = 5$) to assess whether the hierarchical Bayesian agent can correctly identify which of the eight task types it is observing. For each task, we simulated 20 independent games using a random policy. After observing the full sequence of actions and rewards for each game, we performed a single batch belief update and recorded the agent's posterior $P(\text{type} \mid \text{data})$ across the eight candidate types. We averaged the agent's posteriors across the 20 repetitions per type.

Fig. A4 shows the resulting confusion matrix. Each row corresponds to the true task type used to generate the data, and each column corresponds to the agent's posterior belief of a given type. Four task types, *Bernoulli*, *two context*, *three context*, and *linear contextual*, are identified with near-perfect accuracy. *Time delayed* yielded strong posteriors (0.92). The agent was moderately confused between *stationary* and *drifting* as expected. With only 20 trials and a drift rate of $\varepsilon = 2$, the reward variance attributable to drift is small relative to observation noise, making the two distributions nearly indistinguishable. Finally, *patch foraging* is the most difficult to identify (0.46 correct), with residual mass split between stationary (0.25) and drifting (0.27). This is justified by the fact that depletion and replenishment signals are extremely difficult to detect under random exploration, as they require arms to be chosen repeatedly. Overall, the analysis confirms that the agent is well-calibrated for the structurally distinctive task types and that confusions are confined to the tasks that are genuinely hard to distinguish from random-policy observations.