

Program-Based Strategy Induction for Reinforcement Learning

Carlos G. Correa¹ (cgcorrea@princeton.edu)

Thomas L. Griffiths^{2,3} (tomg@princeton.edu)

Nathaniel D. Daw^{1,2} (ndaw@princeton.edu)

¹Princeton Neuroscience Institute, Princeton University, Princeton, New Jersey

²Department of Psychology, Princeton University, Princeton, New Jersey

³Department of Computer Science, Princeton University, Princeton, New Jersey

Abstract

Typical models of learning assume incremental estimation of continuously-varying decision variables like expected rewards. However, this class of models fails to capture more idiosyncratic, discrete heuristics and strategies that people and animals appear to exhibit. Despite recent advances in strategy discovery using tools like recurrent networks that generalize the classic models, the resulting strategies are often onerous to interpret, making connections to cognition difficult to establish. We use Bayesian program induction to discover strategies implemented by programs, letting the simplicity of strategies trade off against their effectiveness. Focusing on bandit tasks, we find strategies that are difficult or unexpected with classical incremental learning, like asymmetric learning from rewarded and unrewarded trials, adaptive horizon-dependent random exploration, and discrete state switching.

Keywords: program induction; reinforcement learning; heuristics; strategy discovery;

Classic models of reinforcement learning (RL) often assume that humans and other animals incrementally estimate expected rewards or other continuously-varying decision variables, and make noisy decisions based on them (Rescorla & Wagner, 1972; Sutton & Barto, 2018). However, there is increased recognition that the learning strategies actually adopted by humans (and even animals) deviate qualitatively from these models and instead reflect more discrete, idiosyncratic, and perhaps explicit heuristics: elaborating simple state-switching strategies like win-stay, lose-shift (Iigaya, Fonseca, Murakami, Mainen, & Dayan, 2018; Lau & Glimcher, 2005) and reflecting memories of individual trials rather than summaries (Collins & Frank, 2012; Plonsky, Teodorescu, & Erev, 2015; Duncan & Shohamy, 2016; Bornstein, Khaw, Shohamy, & Daw, 2017). Despite individual attempts to capture these behaviors by elaborating classic models, there is no general formalism spanning the space of such strategies or predicting what variants organisms adopt in particular situations, and how.

Recent work on computational approaches to strategy discovery has started to engage with these questions. For example, meta-RL has been used to train recurrent neural networks (RNNs) to have recurrent dynamics that implement a learning process (Duan et al., 2016; Wang et al., 2018) which approximate decision-making strategies that are either task-optimized (Ortega et al., 2019) or supervised to distill organisms' learning behavior (Dezfouli, Griffiths, Ramos, Dayan, & Balleine, 2019; Ji-An, Benna, & Mattar, 2023; Miller, Eck-

stein, Botvinick, & Kurth-Nelson, 2023). However, this approach is lacking in a number of respects. First, neural networks require considerable effort to interpret (making them awkward for theory discovery) and are data- and compute-intensive to optimize (making them unlikely candidates for a process model of how organisms discover strategies). Relatedly, it is not clear that they represent the most appropriate strategy space over which to conduct meta-learning for these purposes. While the divergence of task-optimized RNNs from observed behavior might be addressed by introducing resource costs (Moskovitz, Miller, Sahani, & Botvinick, 2023; Binz & Schulz, 2022), capacity-limited RNNs have low-dimensional continuous state spaces that do not capture the types of memory-dependent dynamics discussed above.

Accordingly, we propose instead to use Bayesian program induction to identify program-structured strategies for meta-learning in RL problems. This builds on approaches used to study cognitive representations (Goodman, Tenenbaum, Feldman, & Griffiths, 2008; Yang & Piantadosi, 2022; Ellis, Albright, Solar-Lezama, Tenenbaum, & O'Donnell, 2022) and discover machine learning algorithms (Real, Liang, So, & Le, 2020) that have been adapted to sequential decision making (Toussaint & Storkey, 2006; Wingate, Goodman, Roy, Kaelbling, & Tenenbaum, 2011). Our Bayesian framework has a natural interpretation as a resource-rational approach (Lieder & Griffiths, 2020) by letting the cognitive cost of a program correspond to its description length, so that simpler strategies are more probable under the prior but more effective strategies are more likely to result in optimal behavior under the likelihood. Importantly, we can explore the relationship between previously-reported behavior and the strategies resulting from different trade-offs between effectiveness and program simplicity. By focusing on program-structured strategies for RL, we explore a broader space of strategies than has previously been considered. These programs are interpretable and, since they are small and can be extended with modularity and reuse (Yang & Piantadosi, 2022; Ellis et al., 2021), may also lead to a process-level model of strategy induction.

Using this approach, we identify a range of strategies in various bandit tasks, accounting for some previously observed behavioral phenomena that deviate from classical incremental learning. We find instances where strategies focus exclusively on reward instead of omission, resembling

behavioral signatures in rodents (Parker et al., 2016) and asymmetric learning more broadly (Palminteri, 2023). We show how this approach supports adaptive, horizon-specific randomness in exploration, as previously observed (Wilson, Geana, White, Ludvig, & Cohen, 2014). We also find strategies that switch between discrete decision states focused on either exploiting known options or exploring new options, as in previous studies (Ebitz, Albarran, & Moore, 2018). Our framework provides an interpretable alternative to existing methods for strategy discovery, which we hope can provide new insights in other domains.

Program-Based Strategy Induction

We formulate strategies as programs and use sampling-based Bayesian inference to find programs that are both simple and effective. We first introduce a formalism for tasks and strategies, then describe our framework for inference, and close with implementation details.

Tasks and Strategies

We study finite-horizon tasks where at every time step t an agent takes an action a_t , makes an observation o_t , and receives a reward r_t , resulting over time in a history of agent-environment interactions $h_t = (a_{1:t}, o_{1:t}, r_{1:t})$. Observations may only provide partial information about the state of the task (Kaelbling, Littman, & Cassandra, 1998), so future observations and rewards depend on the full history of agent-environment interactions, $p(o_{t+1}, r_{t+1} | h_t, a_{t+1})$. Agents are modeled as $p(a_{t+1} | h_t)$, producing a task-specific distribution of histories $p(h_{t+1} | h_t) = p(a_{t+1} | h_t)p(o_{t+1}, r_{t+1} | h_t, a_{t+1})$.

Strategies require performing two computations at every time step: updating internal representations to incorporate new observations and rewards, and using that updated internal representation to act. Formally, strategies generate an updated memory m_{t+1} , based on information from the previous trial (memory m_t , action a_t , observation o_t , and reward r_t), which we refer to as the state update function f ,

$$m_{t+1} = f(m_t, a_t, o_t, r_t). \quad (1)$$

Choice behavior is driven by the updated memory m_{t+1} and, for convenience, information from the previous trial, by way of a policy function g that returns unnormalized log probabilities for each action,

$$\log p(a_{t+1} | h_t) \propto g(m_{t+1}, a_t, o_t, r_t). \quad (2)$$

A completely specified strategy $\pi = (m_1, q_1, f, g)$ also requires an initial memory m_1 and unnormalized log policy $\log p(a_1) \propto q_1$ for the first time step. At subsequent time steps, memory is first updated with Eq. 1 and then action is taken with Eq. 2. Agents following a strategy are evaluated based on the expected sum of rewards they accrue over T steps,

$$V(\pi) = \mathbb{E} \sum_{t=1}^T r_t.$$

Table 1: Primitive operations for strategies.

Primitives	Description
Arithmetic, Logic	
0, ..., 49	Integers from 0 to 49 (inclusive)
+, *	Addition, multiplication
-, 1/(x)	Negation, multiplicative inverse
<, ==	Less than, equals
&&, , !	And, or, negation
if(c, x, y)	Returns x if condition c is true, y otherwise
Vectors	
vec_full(x)	A vector filled with the value x
vec_n(x1, ..., xn)	A vector where the first n entries are supplied and others are 0, e.g., vec_2(x, y) = [x, y, 0, 0]
v[i]	Returns i th entry of v
assign(v, i, x)	Updated copy of v, with v[i]=x
add_assign(v, i, x)	Updated copy of v, with v[i]=v[i]+x
Inputs	
prev_action	Previous action, a_t
reward	Previous reward, r_t
state	Memory from previous trial m_t for f or current trial m_{t+1} for g
Action probabilities	
logit(l)	For two-action tasks, $l = \log \frac{p(a=0)}{p(a=1)}$
softmax(w, v)	Uses unnormalized log probabilities in v, scaled by w
action(a)	Takes action a
argmax(v)	Takes action with earliest, maximum value in v

Inference

We use Bayesian program induction to search the space of strategies with a generative prior over strategies and a likelihood based on the effectiveness of the strategy.

Each component of a strategy is a program, a composition of the primitive operations defined in Table 1. The primitives correspond to simple arithmetic, logic, and vectors, as well as primitives specific to decision-making, such as previous actions, rewards, and action distributions. For simplicity, we let agent states be vectors of length 4. The space of programs consists of all valid combinations of primitive operations that return the appropriate type (either a vector for m_1 and f , or action probabilities for q_1 and g), which corresponds to a context-free grammar. The prior, $p(\pi)$, assumes expansions in the grammar are sampled uniformly at random, except terminals (like integers and inputs) are 8 times more likely than non-terminals in order to avoid excessive nesting. The integers, from 1 onward, are distributed according to a geometric distribution with probability 0.5. We represent actions with integers. Invalid programs (non-integral or

out of bounds actions/indices, computing $1/0$) were assigned a value of $-\infty$. Programs that compute starting values (m_1 , q_1) have a restricted set of primitives, excluding conditionals, vector indexing, and other input values.

Following the standard approach for planning by inference (Toussaint & Storkey, 2006; Wingate et al., 2011; Levine, 2018), we formulate our likelihood by introducing a Bernoulli-distributed random variable Ω that indicates the optimality of a strategy, so that

$$\log p(\Omega = 1 \mid \pi) \propto \beta V(\pi),$$

with β a positive weight on the value. Notably, optimal strategies maximize this likelihood. Combining the likelihood and prior, the posterior probability that programs are optimal is

$$\log p(\pi \mid \Omega = 1) \propto \beta V(\pi) + \log p(\pi). \quad (3)$$

The value weight β controls the relative contribution of simplicity and task performance to the prior, so that simpler solutions are preferred when $\beta \rightarrow 0$ and more effective solutions are preferred when $\beta \rightarrow \infty$.

For inference, we use Markov chain Monte Carlo (MCMC) with Metropolis-Hastings acceptance to obtain samples from the posterior, as in prior work (Yang & Piantadosi, 2022; Goodman et al., 2008). We run five chains for 5×10^5 steps per weight β , which are [100, 300, 1000, 3000, 10000, 30000] for the Wilson et al. (2014) task and [10, 30, 100, 300, 1000, 3000] for others. Proposal distributions are subtree-regeneration (Goodman et al., 2008), and resampling a primitive (retaining arguments), swapping arguments, inserting a subtree between a parent and child, and deleting a subtree while retaining an ancestor (Yang & Piantadosi, 2022). A subset of m_1, q_1, f, g was proposed to, with each program independently sampled for inclusion (probability $\frac{1}{10}$ for m_1 and q_1 and $\frac{1}{5}$ for f and g), avoiding empty subsets by always including a program sampled proportionally to inclusion probabilities. Our implementation is built on the open-source Fleet library used in previous studies (Yang & Piantadosi, 2022).

Value was estimated from 2×10^3 rollouts for the Ebitz et al. (2018) task and 10^4 rollouts in other cases, using expected immediate rewards as well as fixed environmental and agent seeds per chain to decrease variance. To facilitate comparison across chains, we compare normalized value so that chance behavior corresponds to 0 and behavior guided by an oracle (i.e. complete knowledge of the reward probabilities) corresponds to 1. Pareto frontiers were computed by finding the maximal strategies in the space defined by normalized value and the prior, using the the top 100 strategies from each chain.

Exploring Strategies for a Simple Bandit Task

We first survey the strategies discovered for a two-armed bandit, with a finite horizon of 20 trials and stationary, Bernoulli reward probabilities drawn from a uniform distribution. The simplicity of this domain makes it possible to compute the value of strategies exactly. It is also possible to

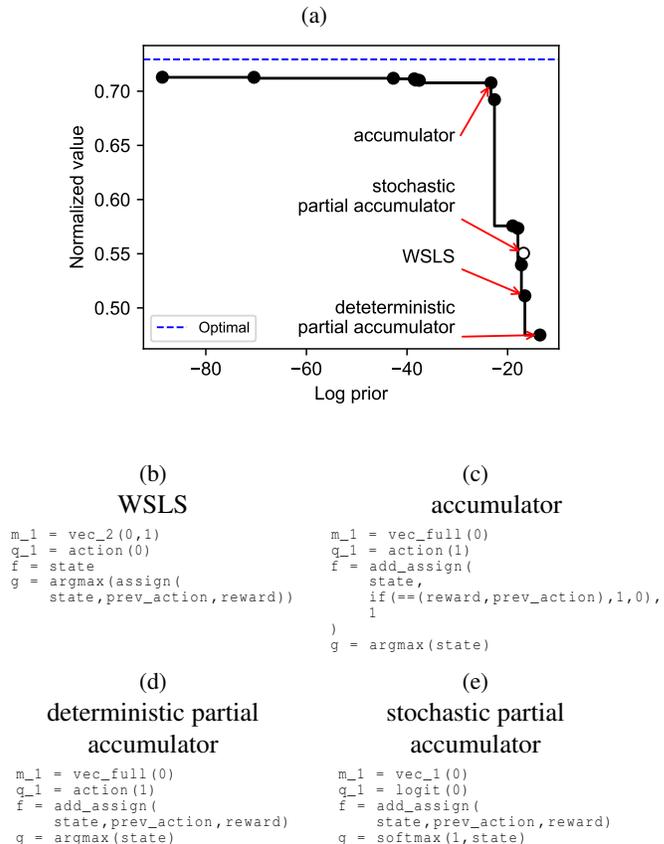


Figure 1: Strategies for the two-armed bandit with stationary, Bernoulli rewards. a) The Pareto frontier of deterministic strategies, which are maximal points in the space defined by the normalized value and prior. Points correspond to individual strategies. One stochastic strategy described in the text is included (unfilled point) for comparison. Value is normalized so that chance behavior has a value of 0 and behavior guided by an oracle has a value of 1. Also shown is the Bayes-optimal solution to the task (dotted line). Example strategies are shown in b-e), marked in a), and described in the text.

compare to the optimal solution, which has a simple analytic solution that can be reasonably computed for this horizon.

We first examine strategies when value is exactly computed, restricting attention to deterministic policies for computational reasons. We focus on the Pareto frontier defined by simplicity and utility, that is, solutions that are local maxima in the space defined by simplicity and utility (Fig. 1a). Overall, the model exposes a progression of more elaborate and more effective strategies parameterized by this trade-off.

One simple solution we identify is win-stay, lose-shift (WSLS; Fig. 1b), where an action is repeated if the previous trial was rewarded, or an alternative action is selected if the previous trial was not rewarded (Robbins, 1952). This solution has a trivial state update f that simply passes a starting vector onward, with the strategy instead implemented entirely in the policy function g . For the sake of clarity, we

briefly walk through the evaluation of this strategy. When `prev_action=0` and `reward=1`, the result of `assign` is `[1, 1]`. Since `argmax` returns the earliest maximum value, the subsequent action is 0, so winning results in repeating the action 0. On the other hand, if `reward=0`, the result of `assign` is `[0, 1]`, so `argmax` would result in the action 1, meaning that a loss resulted in a switch. The program behaves similarly for the other action. While a more human-readable implementation might use a conditional, this implementation’s careful use of indices avoids the need for a conditional and happens to be more probable under the prior.

A much more effective strategy accumulates the evidence indicated by rewards, so that wins increase the preference for an arm and losses decrease it. The algorithm in Fig. 1c implements this by incrementing the state at a conditional index, so that wins for an arm or losses for the other arm result in an increment as indexed by the arm. Since actions are selected by using `argmax`, the arm with the greatest accumulated evidence is selected at any time.

Other discovered strategies worse than the accumulator corresponded to partial accumulators that only incorporated part of the reward feedback in either an arm-specific or reward-specific way (like in Fig. 1d). Strategies better than the accumulator were typically minor variants of it, like giving greater weight to outcomes for the arm that was not initially selected.

For validation, we examined the strategies discovered when value was estimated and stochastic policies were permitted. We found similar policies as above, and use of stochastic policies that were often effectively deterministic due to large values of inverse temperatures.

One question is whether the resource-rational trade-offs implied by this model rationalize idiosyncratic behaviors in biological learning. One notable random policy is the stochastic partial accumulator in Fig. 1e, which shows a modest increase in value over WLS. Interestingly, this strategy ignores losses. This can result in poor performance with deterministic behavior, since choice is fixed to the first rewarding option (Fig. 1d, marked in Fig. 1a). However, partial accumulation can perform effectively with stochastic choice. This example is broadly consistent with findings of asymmetric learning from positive and negative prediction errors, such as exclusively learning from reward by rodents in a reversal learning task (Parker et al., 2016). It also echoes findings of optimistic learning biases in humans (Palminteri, 2023) that theoretical accounts (in a related task with counterfactual feedback) have shown to be normative in some settings (Lefebvre, Summerfield, & Bogacz, 2022).

To examine this more closely, we searched exhaustively over a small space of strategies to see whether exclusive accumulation of either rewarded or unrewarded trials was more effective. The space of strategies was a generalization of the partial accumulators in Fig. 1, testing starting conditions, inverse temperature, and whether policies were deterministic or not. When rewards were accumulated, the best strategy we

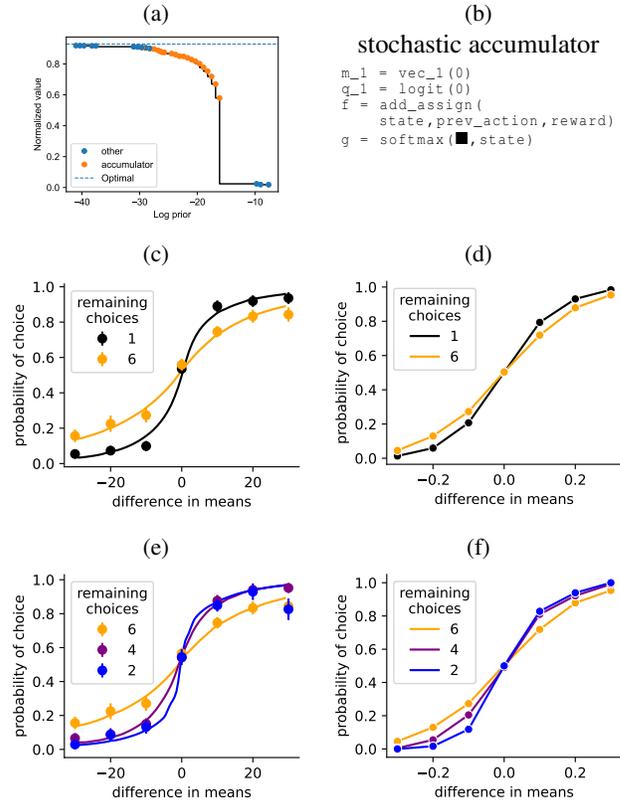


Figure 2: Adaptive random exploration, using a stochastic accumulator. a) The Pareto frontier for solutions in the long horizon condition. Many solutions simply accumulate rewards, but increase the determinism of policies (less probable under prior) to achieve greater value. b) The discovered accumulator. The “█” is a placeholder for the inverse temperature. c) Replotted from Fig. 2b in Wilson et al. (2014). Across conditions, there is a horizon-specific adjustment of decision noise. d) The optimal horizon-specific inverse temperature for the stochastic accumulator leads to more random behavior for the long horizon condition. Horizon-specific inverse temperature was selected to maximize Eq. 3 with $\beta = 300$. Rewards were scaled by $\frac{1}{100}$. e) Replotted from Fig. 3a in Wilson et al. (2014). In the long horizon condition, there is horizon-specific adjustment of decision noise. f) Memory magnitude in the accumulator grows over time, resulting in less randomness for later trials in the long horizon condition. Same inverse temperature as d).

found was Fig. 1e. When unrewarded trials were accumulated (replacing `reward` with `+(-1, reward)`), the best performing strategy was deterministic and implemented WLS. Thus, within this limited class of strategies, a bias towards positive information is rational.

Adaptive Random Exploration

A key feature of optimal behavior in bandit tasks is adaptive use of exploration—exploring is important early on, but over

time behavior favors the best choice given previous observations. Wilson et al. (2014) showed that humans explore adaptively in a bandit task, with increased preference for unfamiliar and less valuable arms for a longer horizon and decreased preference for both for a shorter horizon. While horizon-dependent exploration of unfamiliar arms is predicted by normative accounts, adaptive random exploration is hard to explain since optimal behavior is deterministic.

The task is a two-armed bandit with normal rewards that starts with four forced-choice trials, to control the information participants have about each arm, then proceeds to either one or six free-choice trials. Forced-choice trials ignored action probabilities, and an additional primitive was added so that strategies could test whether the previous trial was a forced-choice trial. Strategies were found for each horizon condition independently and, to facilitate comparing conditions, value was the average received reward on free-choice trials.

Applying our framework to search for stochastic strategies for this task, we primarily identified an accumulator along the Pareto front (long horizon shown in Fig. 2a, short horizon was similar), with varied inverse temperatures for the softmax (Fig. 2b, increasing with the value weight because the prior disfavors larger constants). This strategy performs best with a high inverse temperature, resulting in near-deterministic behavior. In this section, we will use this example program to account for the adaptive random exploration found by Wilson et al., focusing on the equal-information conditions where it is most directly exposed.

One pattern of adaptive random exploration was between different time horizons: on the first free-choice trial, participants were more random in the long horizon condition (Fig. 2c). Focusing on parametrically varying the strategy in Fig. 2b, we found horizon-specific inverse temperatures for the softmax that maximized the posterior (Eq. 3). The inverse temperature was larger for the short horizon, resulting in more deterministic behavior compared to the long horizon (Fig. 2d), consistent with the behavioral findings.

This horizon-dependent randomness results from an interplay between the accumulator’s memory dynamics and the representation cost of the policy. Accumulating rewards instead of weighting them means that the magnitude of entries in the agent’s state will increase over time, leading to greater determinism. So, for the long horizon condition, the inverse temperature primarily serves to make early choices more deterministic. Since integers are geometrically-distributed, larger values of the constant softmax temperature are less probable. This penalizes the determinism of a policy, resembling information-theoretic approaches to penalizing policy complexity (Piray & Daw, 2021; Lai & Gershman, 2021). Combined, this means that a strategy specific to the long horizon condition can afford to be more random early on, avoiding the penalty associated with being deterministic.

The second pattern of adaptive random exploration identified by Wilson et al. (2014) is within the long horizon condition: participant choice was more random when there were

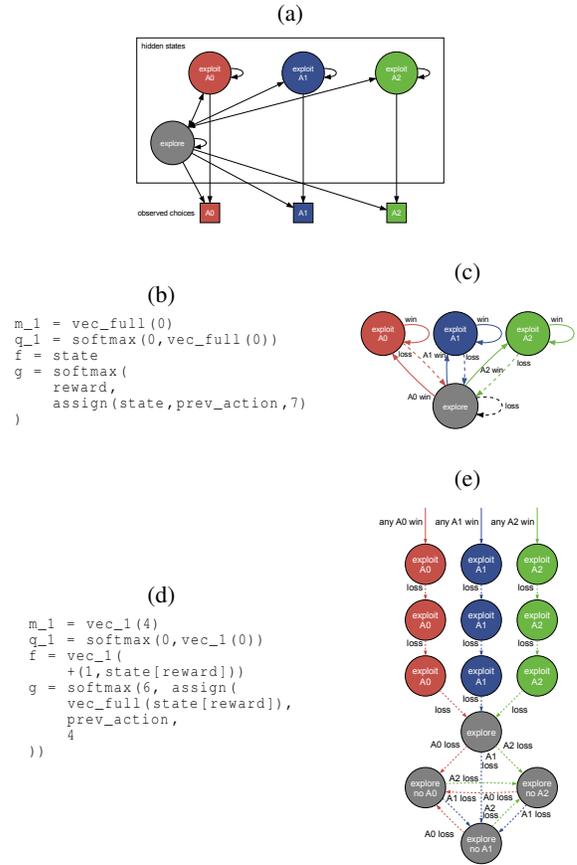


Figure 3: Discovering strategies with discrete decision states for a bandit task with non-stationary reward. a) The state-based choice model in Ebitz et al. (2018), featuring distinct states where behavior is either more exploratory or exploitative. Adapted from Fig. 1c in Ebitz et al. (2018). b) A WLSL strategy that randomly samples from actions after a loss. c) A state machine for WLSL strategy. States correspond to distinct action distributions. Edge color indicates action, dashed edges correspond to losses, and edges with action probability of less than 1% were excluded. The initial conditions of the strategy were modified to simplify the state machine. d) A more complex strategy that exploits after a single win, but requires consecutive losses to switch. e) State machine for d). Wins always lead to a state at top (as indicated), so wins are excluded elsewhere. States with 4+ consecutive losses (at bottom) were collapsed because they have similar action probabilities and identical transitions.

more remaining trials (Fig. 2e). While these results incorporated free-choice trials, they attempted to avoid a confound between information and reward by focusing on cases where each arm had been selected an equal number of times. Using the best inverse temperature for the long horizon, we saw similar results (Fig. 2f). As described above, the accumulator’s entries grow over time, resulting in a natural shift toward more deterministic behavior.

Discrete Decision States

We next examine strategies that emerge for learning in a bandit task with non-stationary rewards, which is difficult to solve with the accumulation models presented in previous sections. Ebitz et al. (2018) found that both choice behavior and neural activity from monkeys in this setting were captured by a slightly generalized WSLs model that involved switching between distinct states: exploring by random sampling, and exploiting by selecting a preferred action (Fig. 3a). Quickly switching between several distinctive regimes of behavior is difficult to model using error-based learning.

We study the task reported by Ebitz et al., which is a 3-armed bandit with Bernoulli rewards. The probability of reward for each arm is in $(0.1, 0.2, \dots, 0.9)$. After every trial, the probability has a 10% chance of drifting up or down by 0.1, while staying in bounds. We set the horizon to 500 trials.

Our framework identifies strategies with similar motifs as the one proposed by Ebitz et al., so we qualitatively examine two in this section. The simpler program in Fig. 3b implements WSLs where losses lead to random sampling among options—a loss means $\text{reward}=0$, so softmax returns a uniform distribution. We also show the program’s dynamics as a state machine in Fig. 3c, where each state corresponds to a unique distribution over actions, and the arrows show how wins and losses change the policy. In the state machine, the explore state results in random choice. Wins lead to an action-specific exploit state, and losses lead back to the explore state. This strategy precisely mirrors the schematic proposed by Ebitz et al. However, while their detailed results are consistent with state switching, they also show that both neural and behavioral measures somewhat violate the Markov conditional independence properties implied by their WSLs model, suggesting the actual state space is more complicated. We explore one such strategy next.

The strategy in Fig. 3d (state machine in Fig. 3e) elaborates both the exploration and exploitation phases. In particular, when exploiting (top) it switches only after three successive losses; for exploration (bottom) it tries random arms until a win, but avoids the most recent choice. The code implements this by counting consecutive losses in the 0^{th} entry of memory, resetting the counter to 1 following a win, and using the counter to inform choice probabilities. Importantly, this strategy retains the discrete, state-like nature of the proposal in Ebitz et al. Similar to the bandit algorithms discussed above, this strategy also responds more readily to positive feedback, which is also consistent with theoretical findings (in a related task) that choice-confirmation bias is adaptive when reward is non-stationary (Lefebvre et al., 2022).

Discussion

We have proposed a framework for strategy induction, using inferential methods to identify strategies that are both simple and effective. By searching for program-structured strategies, our approach is able to find strategy representations that are more interpretable than results from other approaches. We

use this framework to examine strategies for RL tasks that trade off between simplicity and effectiveness. We were able to discover strategies consistent with previously observed behavior, like asymmetric sensitivity to prediction errors, adaptive random exploration, and use of discrete decision states.

We were able to justify adaptive, horizon-dependent random exploration in our framework, as previously observed (Wilson et al., 2014). Within a fixed class of strategies, our framework can cast this adaptive behavior as the result of a resource-rational trade-off. Other approaches could explain this by penalizing the divergence of policies from a uniform distribution (Lai & Gershman, 2021; Piray & Daw, 2021; Levine, 2018). However, our approach goes beyond simple information-theoretic penalties on action distributions since it can identify structured, task-specific strategies and penalize them based on their description length.

One limitation of our current approach is that the primary cost of a program is its representational cost. This neglects any execution-related costs the program might incur, like memory use. Penalizing trial-specific computation costs could encourage strategies that conditionally perform fewer computations in some trials, which, for example, could result in habits which can be adaptive under a speed-accuracy trade-off (Dezfouli & Balleine, 2012). Future research could also adjust the primitive operations available to reflect cognitive resource limitations. For example, many kinds of memory are known to be subject to capacity limits or noise, like working memory (Collins & Frank, 2012) and action values (Findling, Skvortsova, Dromnelle, Palminteri, & Wyart, 2019).

Our approach is a computational-level account (Marr, 1982) of strategy induction, leaving open questions about the algorithms underlying strategy discovery. Future work could examine the process-level bias induced by varying the number of samples taken in MCMC or for value estimation. Another question is how library learning, which adds common subprograms as primitives (Ellis et al., 2021), could accelerate inference. Library learning offers a simple way to perform continual learning, which, by contrast, is an open challenge for neural networks (e.g., requiring task-specific network dynamics to be identified and frozen to avoid catastrophic interference in research by Duncker, Driscoll, Shenoy, Sahani, & Sussillo, 2020).

While we have focused on resource-rational strategies for learning in a handful of bandit tasks, future research could apply this framework to identify interpretable strategies for sequential decision making in settings like planning, problem solving, and other learning tasks. Future work could also apply this framework to discover strategies directly from behavioral data, as others have done with RNNs (Miller et al., 2023; Ji-An et al., 2023). We hope that the continued development of strategy discovery methods can accelerate our understanding of cognition by identifying novel features of behavior (Collins & Cockburn, 2020) and avoid misinterpretation of behavioral signatures (Collins & Frank, 2012; Akam, Costa, & Dayan, 2015).

Acknowledgments

We thank Evan M. Russek, Flora Bouchacourt, and Mark K. Ho for helpful discussions about early versions of this project. This research was supported by grants from the U.S. Army Research Office (ARO W911NF-16-1-0474, awarded to NDD), National Institute of Mental Health (R01MH135587, awarded to NDD), and the NOMIS foundation (awarded to TLG). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

- Akam, T., Costa, R., & Dayan, P. (2015). Simple plans or sophisticated habits? State, transition and learning interactions in the two-step task. *PLOS Computational Biology*, *11*(12), 1–25.
- Binz, M., & Schulz, E. (2022). Modeling human exploration through resource-rational reinforcement learning. In *Advances in Neural Information Processing Systems* (Vol. 35).
- Bornstein, A. M., Khaw, M. W., Shohamy, D., & Daw, N. D. (2017). Reminders of past choices bias decisions for reward in humans. *Nature Communications*, *8*(1), 15958.
- Collins, A. G. E., & Cockburn, J. (2020). Beyond dichotomies in reinforcement learning. *Nature Reviews Neuroscience*, *21*(10), 576–586.
- Collins, A. G. E., & Frank, M. J. (2012). How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, *35*(7), 1024–1035.
- Dezfouli, A., & Balleine, B. W. (2012). Habits, action sequences and reinforcement learning. *European Journal of Neuroscience*, *35*(7), 1036–1051.
- Dezfouli, A., Griffiths, K., Ramos, F., Dayan, P., & Balleine, B. W. (2019). Models that learn how humans learn: The case of decision-making and its disorders. *PLOS Computational Biology*, *15*(6), 1–33.
- Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., & Abbeel, P. (2016). RL²: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.
- Duncan, K. D., & Shohamy, D. (2016). Memory states influence value-based decisions. *Journal of Experimental Psychology: General*, *145*(11), 1420–1426.
- Duncker, L., Driscoll, L., Shenoy, K. V., Sahani, M., & Susillo, D. (2020). Organizing recurrent network dynamics by task-computation to enable continual learning. In *Advances in Neural Information Processing Systems* (Vol. 33).
- Ebitz, R. B., Albarán, E., & Moore, T. (2018). Exploration disrupts choice-predictive signals and alters dynamics in prefrontal cortex. *Neuron*, *97*(2), 450–461.e9.
- Ellis, K., Albright, A., Solar-Lezama, A., Tenenbaum, J. B., & O’Donnell, T. J. (2022). Synthesizing theories of human language with Bayesian program induction. *Nature Communications*, *13*(1), 5024.
- Ellis, K., Wong, C., Nye, M., Sablé-Meyer, M., Morales, L., Hewitt, L., ... Tenenbaum, J. B. (2021). Dream-coder: Bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation* (pp. 835–850).
- Findling, C., Skvortsova, V., Dromnelle, R., Palminteri, S., & Wyart, V. (2019). Computational noise in reward-guided learning drives behavioral variability in volatile environments. *Nature Neuroscience*, *22*(12), 2066–2077.
- Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, *32*(1), 108–154.
- Iigaya, K., Fonseca, M. S., Murakami, M., Mainen, Z. F., & Dayan, P. (2018). An effect of serotonergic stimulation on learning rates for rewards apparent after long intertrial intervals. *Nature Communications*, *9*(1), 2477.
- Ji-An, L., Benna, M. K., & Mattar, M. G. (2023). Automatic discovery of cognitive strategies with tiny recurrent neural networks. *bioRxiv preprint 2023.04.12.536629*.
- Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1), 99–134.
- Lai, L., & Gershman, S. J. (2021). Policy compression: An information bottleneck in action selection. In K. D. Federmeier (Ed.), *The psychology of learning and motivation* (Vol. 74, pp. 195–232). Academic Press.
- Lau, B., & Glimcher, P. W. (2005). Dynamic response-by-response models of matching behavior in rhesus monkeys. *Journal of the Experimental Analysis of Behavior*, *84*(3), 555–579.
- Lefebvre, G., Summerfield, C., & Bogacz, R. (2022). A normative account of confirmation bias during reinforcement learning. *Neural Computation*, *34*(2), 307–337.
- Levine, S. (2018). Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.
- Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, *43*, e1.
- Marr, D. (1982). *Vision*. W. H. Freeman.
- Miller, K. J., Eckstein, M., Botvinick, M. M., & Kurth-Nelson, Z. (2023). Cognitive model discovery via disentangled RNNs. *bioRxiv preprint 2023.06.23.546250*.
- Moskovitz, T., Miller, K., Sahani, M., & Botvinick, M. M. (2023). A unified theory of dual-process control. *arXiv preprint arXiv:2211.07036*.
- Ortega, P. A., Wang, J. X., Rowland, M., Genewein, T., Kurth-Nelson, Z., Pascanu, R., ... Legg, S. (2019). Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*.
- Palminteri, S. (2023). Choice-confirmation bias and gradual

- perseveration in human reinforcement learning. *Behavioral Neuroscience*, 137(1), 78–88.
- Parker, N. F., Cameron, C. M., Taliaferro, J. P., Lee, J., Choi, J. Y., Davidson, T. J., . . . Witten, I. B. (2016). Reward and choice encoding in terminals of midbrain dopamine neurons depends on striatal target. *Nature Neuroscience*, 19(6), 845–854.
- Piray, P., & Daw, N. D. (2021). Linear reinforcement learning in planning, grid fields, and cognitive control. *Nature Communications*, 12(1), 4942.
- Plonsky, O., Teodorescu, K., & Erev, I. (2015). Reliance on small samples, the wavy recency effect, and similarity-based learning. *Psychological Review*, 122(4), 621–647.
- Real, E., Liang, C., So, D., & Le, Q. (2020). AutoML-zero: Evolving machine learning algorithms from scratch. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning* (Vol. 119, pp. 8007–8019).
- Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II: Current research and theory* (pp. 64–99). New York: Appleton-Century-Crofts.
- Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5), 527–535.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.
- Toussaint, M., & Storkey, A. (2006). Probabilistic inference for solving discrete and continuous state Markov decision processes. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 945–952).
- Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., . . . Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6), 860–868.
- Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., & Cohen, J. D. (2014). Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6), 2074–2081.
- Wingate, D., Goodman, N. D., Roy, D. M., Kaelbling, L. P., & Tenenbaum, J. B. (2011). Bayesian policy search with policy priors. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two* (pp. 1565–1570).
- Yang, Y., & Piantadosi, S. T. (2022). One model for the learning of language. *Proceedings of the National Academy of Sciences*, 119(5), e2021865119.