A way around the exploration-exploitation dilemma

Erik J Peterson[1,2*] **and Timothy V Verstynen**[2,3,4]

[1]Department of Psychology
[2]Center for the Neural Basis of Cognition
[3]Carnegie Mellon Neuroscience Institute
[4]Biomedical Engineering, Carnegie Mellon University, Pittsburgh PA
[*]Erik.Exists@gmail.com

ABSTRACT

The optimal decision to exploit existing rewards, or explore looking for larger rewards, is known to be a mathematically intractable problem. Here we challenge this fundamental result in the learning and decision sciences by showing that there is an optimal solution if exploitation and exploration are treated as independent, but competing, objectives. To make it independent we re-imagine exploration as an open-ended search for any new information, whose value we define with a new set of universal axioms. We prove a reward-information competition can be solved by a deterministic winner-take-all algorithm. This algorithm has the properties expected of an ideal solution to the original dilemma, maximizing total value with no regret. Our work suggests an answer to all exploration-exploitation questions can found by exploring simply to learn.

Introduction

Decision making in the natural world often leads to a dilemma. For example, let's imagine a bee foraging in a meadow. The bee could go to the location of a flower it's been to before (exploitation) or it could go somewhere else, to exploration. In most accounts of this situation, decisions to explore and exploit both try to maximize tangible rewards, like nectar[1] (Figure 1**A**). This account however contains in it a mathematically intractable problem. There is no way prove, for any given moment, whether it is optimal to explore or to exploit[2–5].

But resource gathering is not the only reason animals explore. Many animals, like our bee, explore to learn about their environment, developing an often simplified model that helps them in planning actions and making future decisions[6,7]. Borrowing from the field of artificial intelligence we refer to these models as "world models"[1,8,9]. World models offer a principled explanation for why animals are intrinsically curious[10–15], and prone to explore even when no rewards are present or expected[16].

Given animals explore just to learn, we wondered if exploration exclusively for reward is necessary. This lead us to make a bold conjecture that's the focus of this paper. The only exploratory behavior an animal needs is that which builds its world model (Figure 1**B**); exploration for reward is redundant.

Our contribution is then threefold. We offer two axioms that serve as a general basis to value of any observation, given a world model. This lets us define a separate objective for exploration. In fact, the axioms lets us formally disconnect Shannon's Information Theory from any notion of information value, which leads to a new universal theory. Next we prove that the computer science method of dynamic programming[1,17] provides an optimal way to maximize this kind of information value. Finally, we describe a simple winner-take-all algorithm that optimally maximizes both information value and reward.

Results

A definition of information value

Tangible rewards are a conserved resource, while learned information is not. This makes them distinct concepts or ideas. For example, if a rat shares potato chip with a cage-mate, she must necessarily split up the chip leaving less food for herself. Whereas if student shares the latest result from a scientific paper with a lab-mate, they do not necessarily forget a portion of that result.

In an optimization problem, like the dilemma, we assume distinct concepts should have separate objective functions. To do this we first looked to the field of information theory[18], but found it wanting. The problem of information value is not based in the statistical problem of transmitting symbols, as was Shannon's goal. It is based on the problem of learning and remembering them–which is a very different problem.

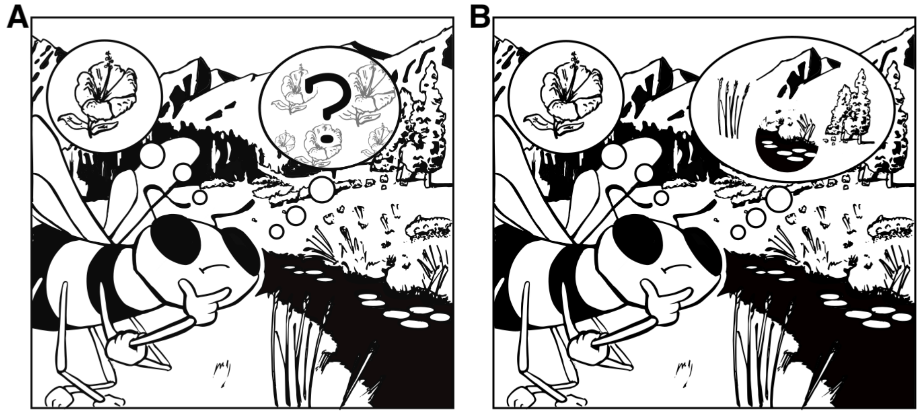

Figure 1. Two views of exploration and exploitation. **A**. The classic dilemma: either exploit an action with a known reward (e.g., return to the previous plant) or explore other actions on the chance they will return a better outcome (e.g., find a plant with more flowers). **B.** Here we offer an alternative view of the dilemma, with two different competitive goals: maximize rewards (e.g., keep returning to known flower locations) or build a world model by learning new information (e.g., layout of the environment). Exploration here focused on learning in general, not on reward learning specifically. *Artist credit*: Richard Grant.

A model of memory

World models are memories with some amount of simplification[9,19]. They can range from simple novelty signals[20], to location or state counts[21,22], state and action prediction[9,15,23], flow[24], learning progress[25], classic working or episodic memories[26,27], Bayesian and hierarchical Bayesian models[23,28–30], latent spaces[31] and recurrent neural networks[19,32–34].

We have no reason to prefer any one kind of world model over any other. So we adopt an abstract definition designed to overlap with nearly any world model. We assume that time is a continuous value and denote increases in time using the differential quantity dt. We then express changes in M (our world model, defined below) as a gradient, ∇M. We also assume that observations about the environment s are real numbers sampled from a finite state space $s \in S$, whose size is N (denoted S^N). Actions are also real numbers a, drawn from a finite space A^K. Rewards R_t, when they appear, are binary $(0,1)$ and are provided by the external environment.

Definition 1. We can now formally define a world model M as a finite set of real numbers, whose maximum size is also N (M^N). We say that every world model has a pair of functions f and g which function as encoder and decoder. Learning of s at time t (i.e. s_t) by M is done by the invertible encoder function f, $M_{t+dt} = f(M_t, s_t)$ and $M_t = f^{-1}(M_{t+dt}, s_t)$. Memories $\hat{s}_t$ about s_t are recalled by the decoder function g, $\hat{s}_t = g(M_t, s_t)$.

The invertibility of f, denoted as f^{-1}, is a mathematical way to ensure that any observations that be encoded in the world model can also be forgotten. This is both an important aspect of real memory, and a critical point for our analysis.

The details of f and g define what kind of world model M is. Let's consider some examples. If f adds states s_t to the memory, and g tests whether s_t is in M, then M is a model of novelty[20]. If f counts states and g returns those counts, then M is a count-based heuristic[21,22]. If f follows Bayes rule and g decodes the probability of s_t, then M is a Bayesian memory[9,15,23,23,29,30]. If the size of M is much smaller than the size of the state space S^N, then f can be seen as learning a latent or compressed representation im M[19,28,31,33–37], and g decodes a reconstruction of s ($\hat{s}_t$) or future states ($\hat{s}_{t+dt}$).

Axioms for information value

To formalize information value we use two axioms that define a real valued function, $E(s)$, that measures the value of any observation s_t given a world model M.

Axiom 1 (Axiom of Change)**.** *The value of $E(s_t)$ depends only on the total distance M moves by making observation s_t.*

This simple axiom does three important things. It ensures that value depends only on the world model, that value is a distance in memory, and that value learning has the Markov property[1].

Distance in memory is the natural choice. When value is distance, learning is always a valuable thing that cannot lead to negative value (*i.e.*, $E(s_t) \geq 0$). When value is distance, value is zero only when ∇M is zero. That is, only no learning means there is no value.

Different f and g pairs will naturally need different ways to measure distances in M. For example, in a novelty world model[20] either the hamming or Manhattan distance are applicable and would produce binary distance values, as would a count model[21,22]. A latent memory[9,15] might instead use the euclidean norm of its own error gradient[38]. While a probabilistic or

Bayesian memory would likely use the Kullback–Leibler (KL) divergence[23,28]. From the axiomatic point of view, all of these are equally good choices.

Axiom 1 does not require that the distance in memories from M to M' be the same as from M' to M. For the technically inclined this makes the distance metric d underlying E a formally only a pre-metric. For many choices of M and d this symmetry will be present, e.g., Euclidean norm). But such symmetry is not necessary and in some cases not desirable (e.g., the KL divergence on a Bayesian memory).

Axiom 2 (Axiom of Equilibrium). *To be valuable an observation s_t must be learnable by M;* $\mathbb{E}\left[\nabla^2 M\right] \leq 0$ *for all* $s \in S$.

By learnable we mean two things. First, with every (re)observation of s, M should change. Second, the change in M must eventually reach a learned equilibrium. Learnability is necessary for information value to have certain meaning.

Most attempts to value information rest their definition on information theory. Value might rest on the intrinsic complexity of an observation (i.e., its entropy)[39] or on its similarity to the environment (i.e., mutual information)[40], or on some other salience signal[41]. In our analysis though learning alone drives value. This is because learning might happen on a true world model or with a faulty world model, or be about a fictional narrative. The observation might be simple, or complex. From the subjective point of view, which is the right point of view, the value for all these can be the same. Value depends only on the total knowledge gained.

Exploration as a dynamic programming problem

Dynamic programming is a popular optimization method because it guarantees value is maximized using a simple algorithm that always chooses the largest option. In Theorem 1 (see *Mathematical Appendix*) we prove that our definition of memory has one critical property, optimal substructure, that is needed for an optimal dynamic programming solution[17,42]. The other two required properties, $E \geq 0$ and the Markov property[17,42], are fulfilled by the *Axiom 1*.

To write down our dynamic programming solution we introduce a little more notation. We let π denote an action policy, a function that takes a state s and returns an action a. We let δ denote the transition function, which takes a state-action pair (s_t, a_t) and returns a new state, s_{t+dt}. This function acts as an abstraction for the actual world. For notational consistency with the standard Bellman approach we also redefine $E(s)$ as a *payoff function*, $F(M_t, a_t)$[17].

$$\begin{aligned} F(M_t, a_t) &= E(s) \\ \text{subject to the constraints} & \\ a_t &= \pi(s_t) \\ s_{t+dt} &= \delta(s_t, a_t), \\ M_{t+dt} &= f(M_t, s_t) \end{aligned} \tag{1}$$

The value function for F is,

$$V_{\pi_E}(M_0) = \left[\max_{a \in A} \sum_{t=0}^{\infty} F(M_t, a_t) \,\middle|\, M,\ S,\ A \right]. \tag{2}$$

And the recursive Bellman solution to learn this value function is,

$$V^*_{\pi_E}(M_t) = F(M_t, a_t) + \max_{a \in A} \Big[F(M_{t+dt}, a_t) \Big]. \tag{3}$$

For the full derivation of Eq 3 see the *Mathematical Appendix*, where we also prove that Eq 3 leads to exhaustive exploration of any finite space S (Theorems 2 and 3).

Scheduling a way around the dilemma

Remember that the goal of reinforcement learning is to maximize reward, an objective approximated by the value function $V_R(s)$ and an action policy π_R.

$$V_R^{\pi_R}(s) = \mathbb{E}\Big[\sum_{k=0}^{\infty} R_{t+k+1} \big| s = s_t \Big] \tag{4}$$

Remember too that our overall goal is to find an algorithm that maximizes both information and reward value. To do that we imagine the policies for exploration and exploitation are possible "jobs" competing to control behavior. We know that, by definition, each of these jobs produces non-negative values: E for information or R for reinforcement learning. So our goal is to find an optimal scheduler for these two jobs.

To do this we further simplify our assumptions. We assume each action takes a constant amount of time, and has no energetic cost. We assume the policy can only take one action at a time, and that those actions are exclusive. Most scheduling solutions also assume that the value of a job is fixed, while in our problem information value changes as the world model improves. In a general setting however, where one has no prior information about the environment, the best predictor of the next value is the last or most recent value[42,43]. We assume this precept holds in all of our analysis.

With these assumptions in place, the optimal solution to this kind of scheduling problem is known to be a purely local, winner-take-all, algorithm[17,42]. We state this winner-take-all solution here as a set of inequalities where R_t and E_t represent the value of reward and information at the last time-point.

$$\begin{aligned} \pi_\pi(s_t) &= \begin{cases} \pi_E^*(s_t) & : E_t - \eta > R_t \\ \pi_R(s_t) & : E_t - \eta \leq R_t \end{cases} \\ &\text{subject to the constraints} \\ & p(\mathbb{E}[R]) < 1 \\ & E - \eta \geq 0 \end{aligned} \tag{5}$$

To ensure that the default policy is reward maximization, Eq. 5 breaks ties between R_t and E_t in favor of π_R. In stochastic environments, M can show small continual fluctuations. To allow Eq. 5 to achieve a stable solution we introduce η, a boredom threshold for exploration. Larger values of η devalue information exploration and favor exploitation of reward.

The worst case algorithmic run time for Eq 5 is linear and additive in its policies. So if in isolation it takes T_E steps to earn $E_T = \sum_{T_E} E$, and T_R steps to earn $r_T = \sum_{T_R} R$, then the worst case training time for π_π is $T_E + T_R$. It is worth noting that this is only true if neither policy can learn from the other's actions. There is, however, no reason that each policy cannot observe the transitions (s_t, a_t, R, s_{t+dt}) caused by the other. If this is allowed, worst case training time improves to $\max(T_E, T_R)$.

Exploration without regret

Suboptimal exploration strategies will lead to a loss of potential rewards by wasting time on actions that have a lower expected value. Regret G measures the value loss caused by such exploration. $G = \hat{V} - V_a$, where $\hat{V}$ represents the maximum value and V_a represents the value found by taking an exploratory action rather than an exploitative one[1].

Optimal strategies for a solution to the exploration-exploitation dilemma should maximize total value with zero total regret.

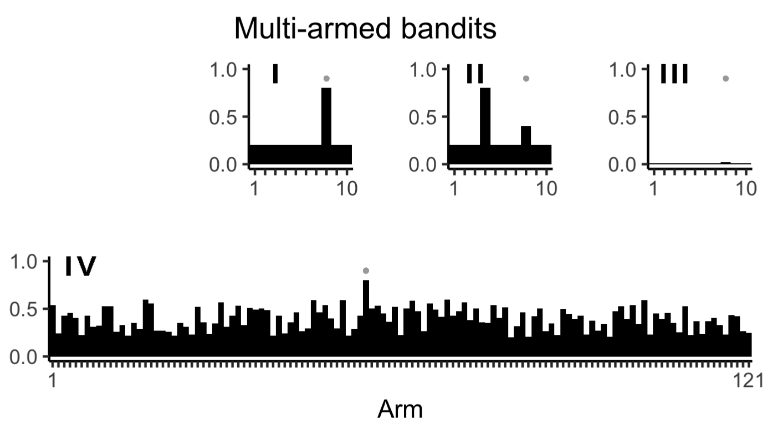

Figure 2. Bandits. Reward probabilities for each arm in bandit tasks I-IV. Grey dots highlight the optimal (i.e., highest reward probability) arm. See main text for a complete description.

To evaluate dual value learning (Eq. 5) we compared total reward and regret across a range of both simple, and challenging multi-armed bandit tasks. Despite its apparent simplicity, the essential aspects of the exploration-exploitation dilemma exist in the multi-armed bandit task[1]. Here the problem to be learned is the distribution of reward probabilities across arms (Figure 2). To estimate the value of any observation s_t, we compare sequential changes in this probabilistic memory, M_{t+dt} and M_t

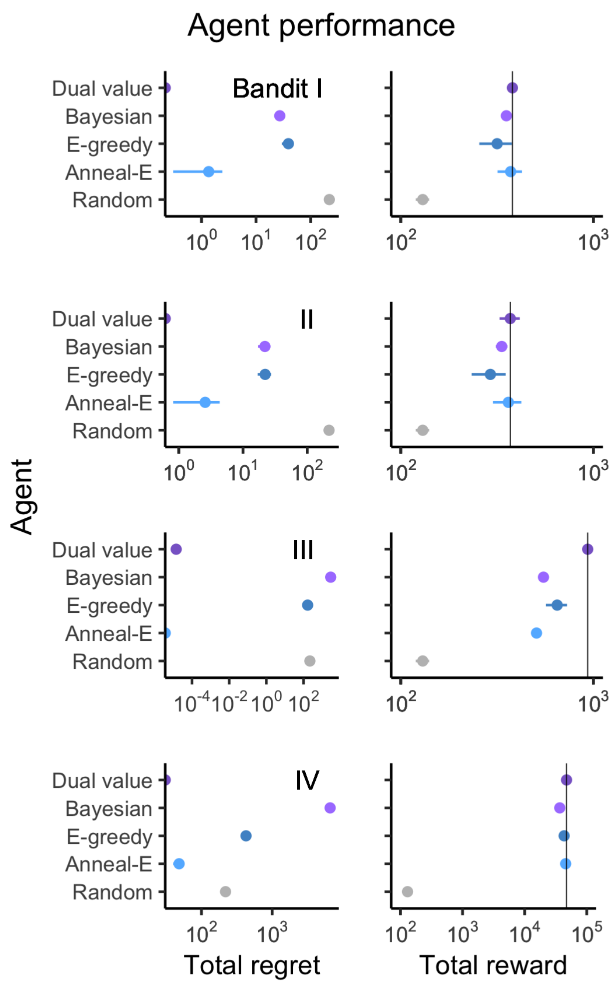

Figure 3. Regret and total accumulated reward across models and bandit task. Median total regret (left column) and median total reward (right column) for simulations of each model type ($N = 100$ experiments per model). See main text and Table 1 for description of each model. Error bars in all plots represent median absolute deviation.

Table 1. Artificial agents.

Agent	Exploration mechanism
Dual value	Our algorithm (Eq 5).
E-greedy	With probability $1-\varepsilon$ follow a greedy policy. With probability ε follow a random policy.
Annealed e-greedy	Identical to E-greedy, but ε is decayed at fixed rate.
Bayesian reward	Use the KL divergence as a weighted intrinsic reward, sampling actions by a softmax policy. $\sum_T R_t + \beta E_t$
Random	Action are selected with a random policy (no learning)

using the KL divergence (i.e. relative entropy; Figure 1**A**-**B**). The KL divergence is a standard way to measure the distance between two distributions[44] and is, by design, consistent with our axioms (see the *Supplementary Materials* for a more thorough discussion).

We start with a simple experiment involving a single high value arm. The rest of the arms have a uniform reward probability (Bandit **I**). This represents a trivial problem. Next we tried a basic exploration test (Bandit **II**), with one winning arm and one distractor arm whose value is close to but less than the optimal choice. We then move on to a more difficult sparse exploration problem (Bandit **III**), where the world has a single winning arm, but the overall probability of receiving any reward is very low ($p(R) = 0.02$ for the winning arm, $p(R) = 0.01$ for all others). Sparse reward problems are notoriously difficult to solve, and are a common feature of both the real world and artificial environments like Go, chess, and class Atari video games[45–47]. Finally, we tested a complex, large world exploration problem (Bandit (**IV**) with 121 arms, and a complex, randomly generated reward structure. Bandits of this type and size are near the limit of human performance[48].

We compared the reward and regret performance of 6 artificial agents. All agents used the same temporal difference learning algorithm (TD(0),[1]); see *Supplementary materials*). The only difference between the agents was their exploration mechanism (Table 1). The e-greedy algorithm is a classic exploration mechanism[1]. Its annealed variant is common in state-of-the-art reinforcement learning papers, like Mnih *et al* ([45]). Other state-of-the-art exploration methods are models that treat Bayesian information gain as an intrinsic reward and the goal of all exploration is to maximize total reward (extrinsic plus intrinsic)[9,49]. To provide a lower bound benchmark of performance we included an agent with a purely random exploration policy.

All of the classic and state-of-the-art algorithms performed well at the different tasks in terms of accumulation of rewards (right column, Figure 3). The one exception to this being the sparse low reward probability condition (Bandit **III**), where the dual value algorithm consistently returned more rewards than the other models. In contrast, most of the traditional models still had substantial amounts of regret in most of the tasks, with the exception of the annealed variant of the e-greedy algorithm during the sparse, low reward probability task (left column, Figure 3). In contrast, the dual value learning algorithm consistently was able to maximize total reward with zero or near zero (Bandit **III**) regret, as would be expected by an optimal exploration policy.

Discussion

Past work

We are certainly not the first to quantify information value[40,50], or use that value to optimize reward learning[3,9,29,51,52]. Information value though is typically framed as a means to maximize the amount of tangible rewards (e.g., food, water, money) accrued over time[1]. This means that information is treated as an analog of these tangible or external rewards (i.e., an *intrinsic reward*)[9,12,23,29]. This approximation does drive exploration in a practical and useful way, but doesn't change the intractability of the dilemma[2–5].

At the other extreme from reinforcement learning are pure exploration methods, like curiosity[15,49,53] or PAC approaches[54]. Curiosity learning is not generally known to converge on rewarding actions with certainty, but never-the-less can be an effective heuristic[15,55,56]. Within some bounded error, PAC learning is certain to converge[54]. For example, it will find the most rewarding arm in a bandit, and do so with a bounded number of samples[57]. However, the number of samples is fixed and based on the size of the environment (but see[58,59]). So while PAC will give the right answer, eventually, its exploration strategy also guarantees high regret.

Animal behavior

In psychology and neuroscience, curiosity and reinforcement learning have developed as separate disciplines[1,53,60]. And they are separate problems, with links to different basic needs: gathering resources to maintain physiological homeostasis[61,62] and gathering information to plan for the future[1,54]. Here we suggest that though they are separate problems, they are problems that can, in large part, solve one another.

The theoretical description of exploration in scientific settings is probabilistic[5,63–65]. By definition probabilistic models can't make exact predictions of behavior, only statistical ones. Our approach is deterministic, and so does make exact predictions. Our theory predicts that it should be possible to guide exploration in real-time using, for example, optogenetic methods in neuroscience, or well timed stimulus manipulations in economics or other behavioral sciences.

Artificial intelligence

Progress in reinforcement learning and artificial intelligence research is limited by three factors: data efficiency, exploration efficiency, and transfer learning[19]. Our algorithm speaks directly to all three of these limits. By treating exploration as a problem in building a world model, our algorithm always ensures high quality exploration. The focus on the world model also means it can be naturally integrated with data efficient model-based reinforcement learning[1,66]. Finally, as it builds a world model that is free of any task specific bias–the model is ideal for later transfer or fine-tuning[67,68].

We describe here a simple and optimal algorithm to combine nearly any world model with any reinforcement learning algorithm. This effectively joins the two approaches to reinforcement learning–model-free and model-based–into an advantageous holistic whole where exploration is model-based but exploitation is model-free.

Acknowledgments.

EP and TV wish to thank Jack Burgess, Matt Clapp, Kyle "Donovank" Dunovan, Richard Gao, Roberta Klatzky, Jayanth Koushik, Alp Muyesser, Jonathan Rubin, and Rachel Storer for their comments on earlier drafts. EP also wishes to thank Richard Grant for his illustration work in Figure 1.

The research was sponsored by the Air Force Research Laboratory (AFRL/AFOSR) award FA9550-18-1-0251. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. government. TV was supported by the Pennsylvania Department of Health Formula Award SAP4100062201, and National Science Foundation CAREER Award 1351748.

References

1. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series (The MIT Press, Cambridge, Massachusetts, 2018), second edition edn.

2. Thrun, S. & Möller, K. Active Exploration in Dynamic Environments. *Adv. neural information processing systems* 531–538 (1992).

3. Dayan, P. & Sejnowski, T. J. Exploration bonuses and dual control. *Mach. Learn.* **25**, 5–22, DOI: 10.1007/BF00115298 (1996).

4. Findling, C., Skvortsova, V., Dromnelle, R., Palminteri, S. & Wyart, V. Computational noise in reward-guided learning drives behavioral variability in volatile environments. Preprint, Neuroscience (2018). DOI: 10.1101/439885.

5. Gershman, S. J. Deconstructing the human algorithms for exploration. *Cognition* **173**, 34–42, DOI: 10.1016/j.cognition.2017.12.014 (2018).

6. Ahilan, S. *et al.* Learning to use past evidence in a sophisticated world model. *PLOS Comput. Biol.* **15**, e1007093, DOI: 10.1371/journal.pcbi.1007093 (2019).

7. Poucet, B. Spatial cognitive maps in animals: New hypotheses on their structure and neural mechanisms. *Psycholocial Rev.* **100**, 162–183 (1993).

8. Schmidhuber, J. Unsupervised Minimax: Adversarial Curiosity, Generative Adversarial Networks, and Predictability Minimization. *arXiv:1906.04493 [cs]* (2019). 1906.04493.

9. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. *Proc. international conference on simulation adaptive behavior: From animals to animats* 222–227 (1991).

10. Mehlhorn, K. *et al.* Unpacking the exploration–exploitation tradeoff: A synthesis of human and animal literatures. *Decision* **2**, 191–215, DOI: 10.1037/dec0000033 (2015).

11. Gupta, A. A., Smith, K. & Shalley, C. The Interplay between Exploration and Exploitation. *The Acad. Manag. J.* **49**, 693–706 (2006).

12. Berger-Tal, O., Nathan, J., Meron, E. & Saltz, D. The Exploration-Exploitation Dilemma: A Multidisciplinary Framework. *PLoS ONE* **9**, e95693, DOI: 10.1371/journal.pone.0095693 (2014).

13. Gottlieb, J. & Oudeyer, P.-Y. Towards a neuroscience of active sampling and curiosity. *Nat. Rev. Neurosci.* **19**, 758–770, DOI: 10.1038/s41583-018-0078-0 (2018).

14. Schwartenbeck, P. *et al.* Computational mechanisms of curiosity and goal-directed exploration. *eLife* 45 (2019).

15. Pathak, D., Agrawal, P., Efros, A. A. & Darrell, T. Curiosity-Driven Exploration by Self-Supervised Prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 488–489, DOI: 10.1109/CVPRW.2017.70 (IEEE, Honolulu, HI, USA, 2017).

16. Hughes, R. N. Intrinsic exploration in animals: Motives and measurement. *Behav. Process.* **41**, 213–226, DOI: 10.1016/S0376-6357(97)00055-7 (1997).

17. Bellmann, R. The theory of dynamic programming. *Bull. Amer. Math. Soc* **60**, 503–515 (1954).

18. Shannon, C. A Mathematical Theory of Communication. *The Bell Syst. Tech. J.* **27**, 379–423, 623–656, (1948).

19. Ha, D. & Schmidhuber, J. World Models. *arXiv:1803.10122 [cs, stat]* DOI: 10.5281/zenodo.1207631 (2018). 1803.10122.

20. Kakade, S. & Dayan, P. Dopamine: Generalization and bonuses. *Neural Networks* **15**, 549–559, DOI: 10.1016/S0893-6080(02)00048-5 (2002).

21. Bellemare, M. G. *et al.* Unifying Count-Based Exploration and Intrinsic Motivation. *arXiv:1606.01868 [cs, stat]* (2016). 1606.01868.

22. Dayan, P. Improving Generalisation for Temporal Difference Learning: The Successor Representation. *Neural Comput.* **5**, 613–624 (1993).

23. Friston, K. *et al.* Active inference and learning. *Neurosci. & Biobehav. Rev.* **68**, 862–879, DOI: 10.1016/j.neubiorev.2016.06.022 (2016).

24. Yang, H.-K., Chiang, P.-H., Hong, M.-F. & Lee, C.-Y. Exploration via Flow-Based Intrinsic Rewards. *arXiv:1905.10071 [cs, stat]* (2019). 1905.10071.

25. Lopes, M., Lang, T., Toussaint, M. & Oudeyer, P.-y. Exploration in Model-based Reinforcement Learning by Empirically Estimating Learning Progress. *NIPS* **25**, 1–9 (2012).

26. Miller, G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychol. Rev.* **63**, 81–97 (1956).

27. Tulving, E. Episodic Memory: From Mind to Brain. *Annu. Rev. Psychol.* **53**, 1–25, DOI: 10.1146/annurev.psych.53.100901.135114 (2002).

28. Park, I. M. & Pillow, J. W. Bayesian Efficient Coding. Preprint, Neuroscience (2017). DOI: 10.1101/178418.

29. Itti, L. & Baldi, P. Bayesian surprise attracts human attention. *Vis. Res.* **49**, 1295–1306, DOI: 10.1016/j.visres.2008.09.007 (2009).

30. Tenenbaum, J. B., Griffiths, T. L. & Kemp, C. Theory-based Bayesian models of inductive learning and reasoning. *Trends Cogn. Sci.* **10**, 309–318, DOI: 10.1016/j.tics.2006.05.009 (2006).

31. Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]* (2013). 1312.6114.

32. Ganguli, S., Huh, D. & Sompolinsky, H. Memory traces in dynamical systems. *Proc. Natl. Acad. Sci.* **105**, 18970–18975, DOI: 10.1073/pnas.0804451105 (2008).

33. Schmidhuber, J. On Learning to Think: Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models. *arXiv:1511.09249 [cs]* (2015). 1511.09249.

34. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84, DOI: 10.1038/nature12742 (2013).

35. Schmidhuber, J. Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes. *arXiv:0812.4360 [cs]* (2008). 0812.4360.

36. Levi-Aharoni, H., Shriki, O. & Tishby, N. Surprise response as a probe for compressed memory states. Preprint, Neuroscience (2019). DOI: 10.1101/627133.

37. Ganguli, S. & Sompolinsky, H. Short-term memory in neuronal networks through dynamical compressed sensing. 9 (2010).

38. Pascanu, R. & Bengio, Y. Revisiting Natural Gradient for Deep Networks. *arXiv:1301.3584 [cs]* (2013). 1301.3584.

39. Haarnoja, T. *et al.* Soft Actor-Critic Algorithms and Applications. *arXiv:1812.05905 [cs, stat]* (2018). 1812.05905.

40. Kolchinsky, A. & Wolpert, D. H. Semantic information, autonomous agency and non-equilibrium statistical physics. *Interface Focus.* **8**, 20180041, DOI: 10.1098/rsfs.2018.0041 (2018).

41. Tishby, N., Pereira, F. C. & Bialek, W. The Information Bottleneck Method. *Arxiv* **0004057**, 11 (2000).

42. Roughgarden, T. *Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming*, vol. 1 (2019).

43. Hocker, D. & Park, I. M. Myopic control of neural dynamics. **15**, 24 (2019).

44. MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms* (2003), 2 edn.

45. Mnih, V. *et al.* Playing Atari with Deep Reinforcement Learning. 9.

46. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489, DOI: 10.1038/nature16961 (2016).

47. Silver, D. *et al.* Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *Science* **362**, 1140–1144 (2018).

48. Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D. & Meder, B. Generalization guides human exploration in vast decision spaces. *Nat. Hum. Behav.* **2**, 915–924, DOI: 10.1038/s41562-018-0467-4 (2018).

49. Jaegle, A., Mehrpour, V. & Rust, N. Visual novelty, curiosity, and intrinsic reward in machine learning and the brain. *Arxiv* **1901.02478**, 13 (2019).

50. Cogliati Dezza, I., Yu, A. J., Cleeremans, A. & Alexander, W. Learning the value of information and reward over time when solving exploration-exploitation problems. *Sci. Reports* **7**, DOI: 10.1038/s41598-017-17237-w (2017).

51. Kelly, J. L. A New Interpretation of Information Rate. *bell system technical journal* 10 (1956).

52. de Abril, I. M. & Kanai, R. Curiosity-Driven Reinforcement Learning with Homeostatic Regulation. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–6, DOI: 10.1109/IJCNN.2018.8489075 (IEEE, Rio de Janeiro, 2018).

53. Berlyne, D. Novelty and curiosity as determinants of exploratory behaviour. *Br. J. Psychol.* **41**, 68–80 (1950).

54. Valiant, L. A theory of the learnable. *Commun. ACM* **27**, 1134–1142 (1984).

55. Burda, Y. *et al.* Large-Scale Study of Curiosity-Driven Learning. *arXiv:1808.04355 [cs, stat]* (2018). 1808.04355.

56. Colas, C., Fournier, P., Sigaud, O., Chetouani, M. & Oudeyer, P.-Y. CURIOUS: Intrinsically Motivated Multi-Task Multi-Goal Reinforcement Learning. *Arxiv* **1810.06284v3**, 1–15 (2019).

57. Even-Dar, E., Mannor, S. & Mansour, Y. PAC Bounds for Multi-armed Bandit and Markov Decision Processes. In Goos, G., Hartmanis, J., van Leeuwen, J., Kivinen, J. & Sloan, R. H. (eds.) *Computational Learning Theory*, vol. 2375, 255–270, DOI: 10.1007/3-540-45435-7_18 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2002).

58. Even-Dar, E., Mannor, S. & Mansour, Y. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *J. Mach. Learn. Res.* **7**, 1–27 (2006).

59. Strehl, A. L., Li, L. & Littman, M. L. Reinforcement Learning in Finite MDPs: PAC Analysis. *J. Mach. Learn. Res.* **10**, 1–32 (2009).

60. Kidd, C. & Hayden, B. Y. The Psychology and Neuroscience of Curiosity. *Neuron* **88**, 449–460, DOI: 10.1016/j.neuron.2015.09.010 (2015).

61. Keramati, M. & Gutkin, B. Homeostatic reinforcement learning for integrating reward collection and physiological stability. *eLife* **3**, e04811, DOI: 10.7554/eLife.04811 (2014).

62. Juechems, K. & Summerfield, C. Where does value come from? Preprint, PsyArXiv (2019). DOI: 10.31234/osf.io/rxf7e.

63. Calhoun, A. J., Chalasani, S. H. & Sharpee, T. O. Maximally informative foraging by Caenorhabditis elegans. *eLife* **3**, e04220, DOI: 10.7554/eLife.04220 (2014).

64. Song, M., Bnaya, Z. & Ma, W. J. Sources of suboptimality in a minimalistic explore–exploit task. *Nat. Hum. Behav.* **3**, 361–368, DOI: 10.1038/s41562-018-0526-x (2019).

65. Schulz, E. *et al.* Exploration in the wild. Preprint, Animal Behavior and Cognition (2018). DOI: 10.1101/492058.

66. Shyam, P., Jaśkowski, W. & Gomez, F. Model-Based Active Exploration. *arXiv:1810.12162 [cs, math, stat]* (2018). 1810.12162.

67. Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. How transferable are features in deep neural networks? *arXiv:1411.1792 [cs]* (2014). 1411.1792.

68. Barreto, A., Dabney, W., Munos, R., Hunt, J. J. & Schaul, T. Successor Features for Transfer in Reinforcement Learning. *Arxiv* **1606.05312v2**, 1–11 (2018).

69. López-Fidalgo, J., Tommasi, C. & Trandafir, P. C. An optimal experimental design criterion for discriminating between non-normal models. *J. Royal Stat. Soc. Ser. B (Statistical Methodol.* **69**, 231–242, DOI: 10.1111/j.1467-9868.2007.00586.x (2007).

70. Still, S., Sivak, D. A., Bell, A. J. & Crooks, G. E. The thermodynamics of prediction. *Phys. Rev. Lett.* **109**, DOI: 10.1103/PhysRevLett.109.120604 (2012). 1203.3271.

71. Ay, N. Information Geometry on Complexity and Stochastic Interaction. *Entropy* **17**, 2432–2458, DOI: 10.3390/e17042432 (2015).

72. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *arXiv:1603.06560 [cs, stat]* (2016). 1603.06560.

73. Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Studies in Nonlinearity (Addison-Wesley Pub, Reading, Mass, 1994).

Supplementary materials.

Dual value implementation

Value initialization and tie breaking

The initial value E_0 for π_E^* can be arbitrary, with the limit $E_0 > 0$. In theory E_0 does not change π_E^*'s long term behavior, but different values will change the algorithm's short-term dynamics and so might be quite important in practice. By definition a pure greedy policy, like π_E^*, cannot handle ties. There is simply no mathematical way to rank equal values. Theorems 3 and 2 ensure that any tie breaking strategy is valid, however, like the choice of E_0, tie breaking can strongly affect the transient dynamics. Viable tie breaking strategies taken from experimental work include, "take the closest option", "repeat the last option", or "take the option with the highest marginal likelihood". We do suggest the tie breaking scheme is deterministic, which maintains the determinism of the whole theory. See *Information value learning* section below for concrete examples both these choices.

The rates of exploration and exploitation

In Theorem 4 we proved that π_π inherits the optimality of policies for both exploration π_E and exploitation π_R over infinite time. However this does proof does not say whether π_π will not alter the rate of convergence of each policy. By design, it does alter the rate of each, favoring π_R. As you can see in Eq. **??**, whenever $r_t = 1$ then π_R dominates that turn. Therefore the more likely $p(r = 1)$, the more likely π_R will have control. This doesn't of course change the eventual convergence of π_E, just delays it in direct proportion to the average rate of reward. In total, these dynamics mean that in the common case where rewards are sparse but reliable, exploration is favored and can converge more quickly. As exploration converges, so does the optimal solution to maximizing rewards.

Re-exploration

The world often changes. Or in formal parlance, the world is non-stationary process. When the world does change, re-exploration becomes necessary. Tuning the size of ε in π_π (Eq **??**) tunes the threshold for re-exploration. That is, once the π_E^* has converged and so π_R^* fully dominates π_π, if ε is small then small changes in the world will allow pi_E to exert control. If instead ε is large, then large changes in the world are needed. That is, ε acts a hyper-parameter controlling how quickly rewarding behavior will dominate, and easy it is to let exploratory behavior resurface.

Bandits

Design

Like the slot machines which inspired them, each bandit returns a reward according to a predetermined probability. As an agent can only chose one bandit ("arm") at a time, so it must decide whether to explore and exploit with each trial.

We study four prototypical bandits. The first has a single winning arm ($p(R) = 0.8$, Figure 2**A**); denoted as bandit **I**. We expect any learning agent to be able to consistently solve this task. Bandit **II** has two winning arms. One of these (arm 7, $p(R) = 0.8$) though higher payout than the other (arm 3, $p(R) = 0.6$). The second arm can act as a "distractor" leading an to settle on this suboptimal choice. Bandit **III** also has a single winning arm, but the overall probability of receiving any reward is very low ($p(R) = 0.02$ for the winning arm, $p(R) = 0.01$ for all others). Sparse rewards problems like these are difficult to solve and are common feature of both the real world, and artificial environments like Go, chess, and class Atari video games[45–47]. The fourth bandit (**IV**) has 121 arms, and a complex randomly generated reward structure. Bandits of this type and size are probably at the limit of human performance[48].

World model and distance

All bandits share a simple basic common structure. The have a set of *n*-arms, each of which delivers rewards in a probabilistic fashion. This lends itself to simple discrete n-dimensional world model, with a memory slot for each arm/dimension. Each slot then represents the independent probability of receiving a reward (Supp. Fig 1**A**).

The Kullback–Leibler divergence (KL) is a widely used information theory metric, which measures the information gained by replacing one distribution with another. It is highly versatile and widely used in machine learning[?], Bayesian reasoning[23,29], visual neuroscience[29], experimental design[69], compression[?,70] and information geometry[71], to name a few examples. KL has seen extensive use in reinforcement learning.

The Kullback–Leibler (KL) divergence satisfies all five value axioms (Eq. 6).

Itti and Baladi Itti2009 developed an approach similar to ours for visual attention, where our information value is identical to their *Bayesian surprise*. Itti and Baladi (2009) showed that compared to range of other theoretical alternative, information value most strongly correlates with eye movements made when humans look at natural images. Again in a Bayesian context, KL plays a key role in guiding *active inference*, a mode of theory where the dogmatic central aim of neural systems is make decisions which minimize free energy[14,23].

Let E represent value of information, such that $E := KL(M_{t+dt}, M_t)$ (Eq. 6) after observing some state s.

$$KL(M_{t+dt}, M_t) = \sum_{s \in S} M_{t+dt}(s) \log \frac{M_{t+dt}(s)}{M_t(s)} \tag{6}$$

Axiom **??** is satisfied by limiting E calculations to successive memories. Axiom **??**-**??** are naturally satisfied by KL. That is, $E = 0$ if and only if $M_{t+dt} = M_t$ and $E \geq 0$ for all pairs (M_{t+dt}, M_t).

To make Axiom 2 more concrete, in Figure 2 we show how KL changes between a hypothetical initial distribution (always shown in grey) and a "learned" distribution (colored). For simplicity's sake we use a simple discrete distribution representing a 10-armed bandit, though the illustrated patterns hold true for any pair of appropriate distributions. In Figure 2**C** we see KL increases substantially more for a local exchange of probability compared to an even global re-normalization (compare panels *A.* and *B.*).

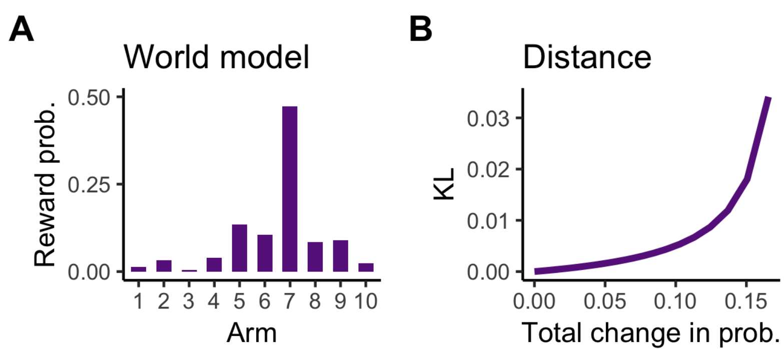

Figure 1. A world model for bandits. **B**. Example of a single world model suitable for all bandit learning. **B** Changes in the KL divergence–our choice for the distance metric during bandit learning–compared to changes in world model, as by measured the total change in probability mass.

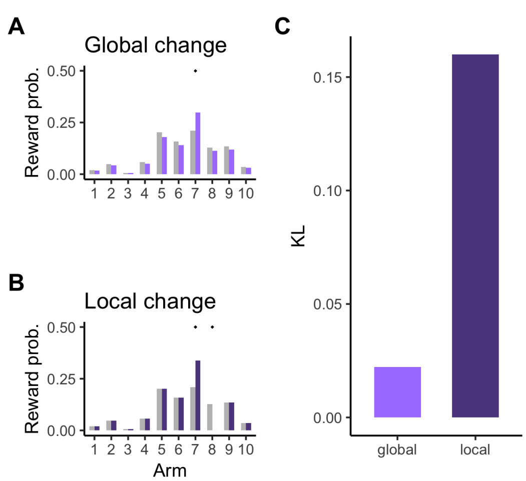

Figure 2. An example of observation specificity during bandit learning. **A**. A initial (grey) and learned (distribution), where the hypothetical observation s increases the probability of arm 7 by about 0.1, and the expense of all the other probabilities. **B**. Same as A except that the decrease in probability comes only from arm 8. **C**. The KL divergence for local versus global learning.

Initializing π_π

In these simulations we assume that at the start of learning an animal should have a uniform prior over the possible actions $A \in \mathbb{R}^K$. Thus $p(a_k) = 1/K$ for all $a_k \in A$. We transform this uniform prior into the appropriate units for our KL-based E using Shannon entropy, $E_0 = \sum_K p(a_k) \log p(a_k)$.

Table 2. Hyperparameters for individual bandits (**I-IV**).

Agent	Parameter	I	II	III	IV
Dual value	η	0.053	0.017	0.003	5.8e-09
Dual value	α	0.34	0.17	0.15	0.0011
E-greedy	ε	0.14	0.039	0.12	0.41
E-greedy	α	0.087	0.086	0.14	0.00048
Annealed e-greedy	τ_E	0.061	0.084	0.0078	0.072
Annealed e-greedy	ε	0.45	0.98	0.85	0.51
Annealed e-greedy	α	0.14	0.19	0.173	0.00027
Bayesian	β	0.066	0.13	0.13	2.14
Bayesian	α	0.066	0.03	0.17	0.13
Bayesian	γ	0.13	0.98	0.081	5.045

In our simulations we use a tie breaking "right next" heuristic which keeps track of past breaks, and in a round robin fashion iterates rightward over the action space.

Reinforcement learning

Reinforcement learning in all agent models was done with using the TD(0) learning rule[1] (Eq. 7). Where $V(s)$ is the value for each state (arm), $\mathbf{R}_t$ is the *return* for the current trial, and α is the learning rate $(0-1]$. See the *Hyperparameter optimization* section for information on how α chosen for each agent and bandit.

$$V(s) = V(s) + \alpha(\mathbf{R}_t - V(s)) \tag{7}$$

The return $\mathbf{R}_t$ differed between agents. Our dual value agent, and both the variations of the e-greedy algorithm, used the reward from the environment R_t as the return. This value was binary. The Bayesian reward agent used a combination of information value and reward $\mathbf{R}_t = R_t + \beta E_t$, with the weight β tuned as described below.

Hyperparameter optimization

The hyperparameters for each agent were tuned independently for each bandit using a modified version of Hyperband[72]. For a description of hyperparameters seen Table 1, and for the values themselves Table **??**.

Exploration and value dynamics

. While agents earned nearly equivalent total reward in Bandit I (Fig 3, *top row*), their exploration strategies were quite distinct. In Supp. Fig 3**B**-**D**) we compare three prototypical examples of exploration, for each major class of agent: ours, Bayesian, and E-greedy for Bandit *I*. In Supp. Fig 3**A**) we include an example of value learning value learning in our agent.

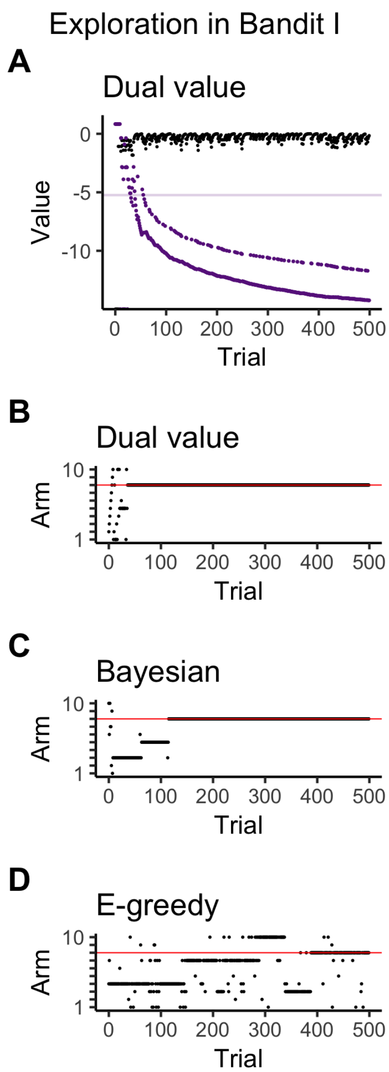

Figure 3. Exploration and value dynamics. **A**. An example of our dual value learning algorithm during 500 trials on Bandit. The light purple line represents the boredom threshold η (Eq. 5). **B.** An example of exploration dynamics (i.e arm selection) on Bandit. Note how the search is structured, and initially sequential. **C-D.** Exploration dynamics for two other agents. **C.** The Bayesian agent, which like our algorithm uses active sampling, and values information. Note how this shows a mixture of structures and repeated choices, mixed with seemingly random behavior. **D.** The E-greedy agent, which uses purely random sampling. Note how here the agent is either greedy, repeating the same arm, or seemingly random.

Mathematical Appendix.

Information value as a dynamic programming problem

To find greedy dynamic programming[1,42] answers we must prove our memory M has optimal substructure. By optimal substructure we mean that M can be partitioned into a small number, collection, or series of memories, each of which is itself a dynamic programming solution. In general by proving we can decompose some optimization problem into a small number of sub-problems whose optimal solution are known, or easy to prove, it becomes trivial to prove that we can also grow the series optimally. That is, proving optimal sub-structure nearly automatically allows for proof by induction[42].

Theorem 1 (Optimal substructure)**.** *Assuming transition function δ is deterministic, if $V^*_{\pi_E}$ is the optimal information value given by π_E, a memory M_{t+dt} has optimal substructure if the the last observation s_t can be removed from M_t, by $M_{t+dt} = f^{-1}(M_{t+dt}, s_t)$ where the resulting value $V^*_{t-dt} = V^*_t - F(M_t, a_t)$ is also optimal.*

Proof. Given a known optimal value V^* given by π_E we assume for the sake of contradiction there also exists an alternative policy $\hat{\pi}_E \neq \pi_E$ that gives a memory $\hat{M}_{t-dt} \neq M_{t-dt}$ and for which $\hat{V}^*_{t-dt} > V^*_{t-dt}$.

To recover the known optimal memory M_t we lift $\hat{M}_{t-dt}$ to $M_t = f(\hat{M}_{t-dt}, s_t)$. This implies $\hat{V}^* > V^*$ which in turn contradicts the purported original optimality of V^* and therefore $\hat{\pi}_E$. □

Bellman solution

Armed with optimal substructure of M we want to do the next natural thing and find a recursive Bellman solution to maximize our value function for F (Eq. 1). (A Bellman solution of F is also a solution for E (Eq.2). We do this in the classic way by breaking up the series for F into an initial value F_0, and the remaining series in the summation. We can then apply this same decomposition recursively (Eq 3) to arrive at a final "twp-step" or recursive form which is shown Eq. 8).

$$
\begin{aligned}
V^*_{\pi_E}(M_0) &= \max_{a \in A} \left[\sum_{t=0}^{\infty} F(M_t, a_t) \right] \\
&= \max_{a \in A} \left[F(M_0, a_0) + \sum_{t=1}^{\infty} F(M_{t+dt}, a_{t+dt}) \right] \\
&= F(M_0, a_0) + \max_{a \in A} \left[\sum_{t=1}^{\infty} F(M_{t+dt}, a_{t+dt}) \right] \\
&= F(M_0, a_0) + V^*_{\pi_E}(M_{t+dt}) + V^*_{\pi_E}(M_{t+2}),\ \ldots
\end{aligned}
\tag{8}
$$

A greedy policy explores exhaustively

To prevent any sort of sampling bias, we need our exploration policy π_E (Eq.3) to visit each state s in the space S. As our policy for E is a greedy policy, proofs for exploration are really sorting problems. That is if a state is to be visited it must have highest value. So if every state must be visited (which is what we need to prove to avoid bias) then under a greedy policy every state's value must, at one time or another, be the maximum value.

We assume implicitly here the action policy π_E can visit all possible states in S. If for some reason π_E can only visit a subset of S, then the following proofs apply only to exploration of that subset.

To begin our proof, some notation. Let Z be the set of all visited states, where Z_0 is the empty set $\{\}$ and Z is built iteratively over a path P, such that $Z_{t+} = \{s | s \in P \text{ and } s \notin Z_t\}$. As sorting requires ranking, we also need to formalize ranking. To do this we take an algebraic approach, are define inequality for any three real numbers (a, b, c) (Eq. 9).

$$a \leq b \Leftrightarrow \exists\, c;\ b = a + c \tag{9}$$

$$a > b \Leftrightarrow (a \neq b) \land (b \leq a) \tag{10}$$

Theorem 2 (State search: breadth)**.** *A greedy policy π is the only deterministic policy which ensures all states in S are visited, such that $Z = S$.*

Proof. Let $\mathbf{E} = (E_1, E_2, \ldots)$ be ranked series of E values for all states S, such that $(E_1 \geq E_2, \geq \ldots)$. To swap any pair of values $(E_i \geq E_j)$ so $(E_i \leq E_j)$ by Eq. 9 $E_i - c = E_j$.

Therefore, again by Eq. 9, $\exists \int \delta E(s) \to -c$.

Recall: Axiom 5.

However if we wished to instead swap $(E_i \leq E_j)$ so $(E_i \geq E_j)$ by definition $\nexists c; E_i + c = E_j$, as $\nexists \int \delta \to c$.

To complete the proof, assume that some policy $\hat{\pi}_E \neq \pi^*_E$. By definition policy $\hat{\pi}_E$ can be any action but the maximum, leaving $k-1$ options. Eventually as $t \to T$ the only possible swap is between the max option and the kth, but as we have already proven this is impossible as long as Axiom 5 holds. Therefore, the policy $\hat{\pi}_E$ will leave at least 1 option unexplored and $S \neq Z$. □

Theorem 3 (State search: depth)**.** *Assuming a deterministic transition function Λ, a greedy policy π_E will resample S to convergence at $E_t \leq \eta$.*

Proof. *Recall*: Axiom 5.

Each time π^*_E visits a state s, so $M \to M'$, $F(M', a_{t+dt}) < F(M, a_t)$

In Theorem 2 we proved only a deterministic greedy policy will visit each state in S over T trials.

By induction, if π^*E will visit all $s \in S$ in T trials, it will revisit them in $2T$, therefore as $T \to \infty$, $E \to 0$. □

Optimality of π_π

In the following section we prove two things about the optimality of π_π. First, if π_R and/or π_E had any optimal asymptotic property for value learning before their inclusion into our scheduler, they retain that optimal property under π_π. Second, we use this Theorem to show if both π_R and π_E are greedy, and π_π is greedy, then Eq 5 is certain to maximize total value. This is analogous to the classic activity selection problem[42].

Independent policy convergence

Theorem 4 (Independence policy convergence under π_π)**.** *Assuming an infinite time horizon, if π_E is optimal and π_R is optimal, then π_π is also optimal in the same senses as π_E and π_R.*

Proof. The optimality of π_π can be seen by direct inspection. If $p(R=1) < 1$ and we have an infinite horizon, then π_E will have a unbounded number of trials meaning the optimally of P^* holds. Likewise, $\sum E < \eta$ as $T \to \infty$, ensuring pi_R will dominate π_π therefore π_R will asymptotically converge to optimal behavior. □

In proving this optimality of π_π we limit the probability of a positive reward to less than one, denoted by $p(R_t = 1) < 1$. Without this constraint the reward policy π_R would always dominate π_π when rewards are certain. While this might be useful in some circumstances, from the point of view π_E it is extremely suboptimal. The model would never explore. Limiting $p(R_t = 1) < 1$ is reasonable constraint, as rewards in the real world are rarely certain. A more naturalistic to handle this edge case is to introduce reward satiety, or a model physiological homeostasis[61,62].

Optimal scheduling for dual value learning problems

In classic scheduling problems the value of any job is known ahead of time[17,42]. In our setting, this is not true. Reward value is generated by the environment, *after* taking an action. In a similar vein, information value can only be calculated *after* observing a new state. Yet Eq. 5 must make decisions *before* taking an action. If we had a perfect model of the environment, then we could predict these future values accurately with model-based control. In the general case though we don't what environment to expect, let alone having a perfect model of it. As result, we make a worst-case assumption: the environment can arbitrarily change–bifurcate–at any time. This is, it is a highly nonlinear dynamical system[73]. In such systems, myopic control–using only the most recent value to predict the next value– is known to be an robust and efficient form of control[43]. We therefore assume that last value is the best predictor of the next value, and use this assumption along with Theorem 4 to complete a trivial proof that Eq. 5 maximizes total value.

Optimal total value

If we prove π_π has optimal substructure, then using the same replacement argument[42] as in Theorem 4, a greedy policy for π_π will maximize total value.

Theorem 5 (Total value maximization of π_π)**.** *π_π must have an optimal substructure.*

Proof. *Recall*: Reinforcement learning algorithms are embedded in Markov Decisions space, which by definition have optimal substructure.

Recall: The memory M has optimal substructure (Theorem 1.

Recall: The asymptotic behavior of π_R and π_E are independent under π_π (Theorem 4

If both π_R and π_E have optimal substructure, and are asymptotically independent, then π_π must also have optimal substructure. □