Title: A way around the exploration-exploitation dilemma

Authors: Erik J Peterson,[1,2*] Timothy D Verstynen[1,2,3,4]

Affiliations:
[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

[*]To whom correspondence should be addressed; E-mail: Erik.Exists@gmail.com

Abstract: The exploration-exploitation dilemma is a fundamental but intractable problem in the learning and decision sciences. In this problem the goal of exploration and exploitation is to maximize reward. Here we challenge this basic form. We conjecture the dilemma can be viewed as a competition between exploiting known rewards or exploring to learn a *world model*, a simplified concept of memory borrowed from computer science. We prove this competition is tractable and can be solved by a simple greedy algorithm. This solution has the properties expected of an optimal solution to original dilemma–finding maximum value with no regret.

One sentence summary: We prove the intractable exploration-exploitation dilemma can actually be solved by viewing exploitation and exploration as separate goals, that are in competition with each other.

Main text: Let's imagine a bee foraging in a meadow. Our bee has a decision to make. It could go back to the location of a flower it has visited before (exploitation) or go somewhere else (exploration). In a traditional reinforcement learning account when our bee explores it acts to maximize tangible rewards (*1*), such as finding a plant with more flowers (Figure 1, **A**). This reinforcement learning account however leads to a mathematically intractable dilemma over whether it is optimal to explore or exploit a any given moment (*2–5*).

Resource gathering is not the only reason animals explore. Many animals, like our bee, learn simplified models of the world to make decisions and plan actions (*6, 7*). Borrowing from the field of artificial intelligence we refer to these as world models (*1, 8, 9*). When learning a world model, information about the environment is intrinsically valuable and is why animals are intrinsically curious (*10–15*) and prone to explore even when no rewards are present or expected (*16*). In some cases information seeking is known to happen even if it explicitly leads to a loss of reward (*17*).

Here we conjecture that the only kind of exploratory behavior an animal needs to do is that which builds its world model. With this conjecture we can propose an alternative to the classic dilemma, breaking exploration and exploitation into independent objectives that compete to either exploit known rewards or explore to learn a world model (Figure 1**B**). We prove this alternative has a tractable and optimal solution. Optimal here means maximising rewards while minimizing regrets.

Our contribution is threefold. We offer five axioms that serve as a general basis to estimate the value of any learned observation, given a memory. This prospective leads to a surprising result. Information theory can be formally disconnected from information value, producing a new universal theory. Next we prove that the computer science method of dynamic programming (*1, 18*) provides an optimal way to maximize this kind of information value. Finally, we describe a simple greedy scheduling algorithm that can maximize both information value and

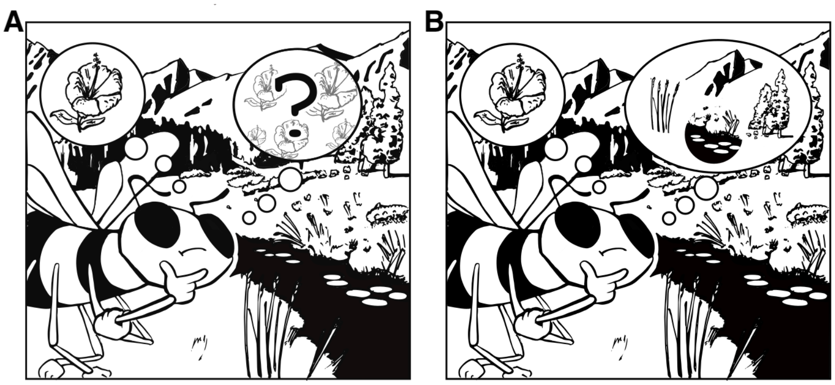

Figure 1: Two views of exploration and exploitation. **A**. The classic dilemma is depicted as two options for reward maximization: exploit an action with a known reward likelihood (e.g., return to the previous plant) or stochastically explore other actions on the chance they return better outcomes (e.g., 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 model of the world by learning new information (e.g., layout of the environment). Notice how exploration here is based not on finding rewards per se (e.g., flowers), but on learning in general. *Artist credit*: Richard Grant.

reward.

Rewards and information are fundamentally distinct concepts. Rewards are a conserved resource. Information is not. For example if a rat shares potato chip with a cage-mate, she must necessarily split the chip up 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.

To formally separate the value of information from the value of reward we look to the field of information theory (*19*). Rather than focus on the statistical problem of transmitting symbols, as was Shannons goal, we focus on remembering symbols, in order to produce a learning and memory view of information value.

World models are memories that 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 (*32–35*). In all of these examples, the value of any observation made by an agent who is learning a world model depends entirely on what the agent learns by making that observation.

We do not prefer any one kind of world model to any other. So we adopt a broad definition of memory, which overlaps with nearly any world model. We assume that time t is continuous quantity, and denote increases in time using the differential quantity dt. We generally then express changes in M (our memory, defined below) as a differential equation (e.g., $\frac{dM}{dt}$), although for non-differential memories difference equations can be used (e.g., $\frac{\Delta M}{\Delta t}$).

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 generally binary and always provided by the external environment.

Preliminaries aside, we can formally define a memory M as a finite set of real numbers, whose maximum size is also N (M^N). We say that learning of s at time t by M happens by an invertible encoder function f, $M_{t+dt} = f(M_t, s_t)$ and $M_t = f^{-1}(M_{t+dt}, s_t)$. (Invertibility here is equivalent to saying that any observations which be stored can be forgotten). Memories $\hat{s}_t$ about s_t are recalled by a decoder function g, such that $\hat{s}_t = g(M_t, s_t)$.

The details of f and g define what kind of world model M is. To make this more concrete, let's consider some examples. If f simply adds states to the memory and g tests whether s_t is in M, then M models 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 we have recovered a classic frequently used information theory account of information value (*23, 29, 30*). In this account the decoded state probabilities could be the current state, or for future states or actions (*9, 15, 23*). Or if M is much smaller than the size of the space S^N, then f could learn a latent or compressed representation (*28, 31, 33–38*), with g decoding a reconstruction of current ($\hat{s}_t$) or future states ($\hat{s}_{t+dt}$).

We define a real valued distance E to measure how M changes with both time and observations. Different f and g pairs will naturally need different ways to exactly express this distance. For example, a novelty model (*20*) would produce binary values, as would a count model (*21, 22*). A latent memory (*9, 15*) might use its own error gradient. A probabilistic memory (*23, 28*) would likely use the KL divergence. All that matters is the chosen distance meet our five axioms.

Axiom 1 (Axiom of Memory)**.** *$E(s_t)$ depends* only *on how the memory M changes when making an observation s_t.*

Axiom 2 (Axiom of Novelty)**.** *An observation s_t that doesn't change the memory M has no value. $E(s_t) = 0$ if and only if $\frac{dM}{dt} = 0$.*

Axiom 3 (Axiom of Scholarship)**.** *All learning in M about s_t is valuable. $E(s_t) \geq 0$; $\frac{dM}{dt} \geq 0$.*

Axiom 4 (Axiom of Specificity)**.** *If the total change in memory M due to observation s_t is held constant ($\frac{dM}{dt} = h$), the more compact (Eq. 6) the change in memory the more valuable the observation.*

Axiom 4 adds two critical and intertwined properties. It ensures that if all else is held equal, more specific observations are more valuable that less specific observations (*39, 40*). It also ensures that an observation that leads to a simplifying or parsimonious insight (is equivalent to a compression of the memory, (*36*)) is more valuable than one that changes memory the same total amount but does not lead to compression.

Axiom 5 (Axiom of Equilibrium)**.** *An observation s_t must be learnable by M. $\frac{d^2M}{dt^2} \leq 0$.*

Technically speaking by learnable we mean learnable using the probably approximately correct (PAC) framework (*41*), a common tool of computer science used to formalize learning and inference. Any observation that cannot be learned, for whatever the reason, is not valuable because it cannot change behavior.

Having written down a positive definition, for clarity we'll also state what our theory is not. Information value is not based on the intrinsic complexity of an observation (that is, its entropy) (*42*), nor on its similarity to the environment (its mutual information; (*43*)), nor on its novelty or surprise (*3, 23, 29*).

Stimulus complexity and surprise have tremendous potential to drive learning, of course. In cases like Bayesian rule there is even a fixed relationship between learning and surprise (*23, 29, 44*). However, this does not hold for all learning rules. Complexity and surprise which can't be learned is not valuable; if it can't be learned it can't shape future actions.

Dynamic programming is a popular optimization approach because it can guarantee total value is maximized by a simple, deterministic, and greedy algorithm. In Theorem 1 (see Mathematical Appendix) we prove our definition of memory has one critical property, optimal substructure, that is needed for a greedy dynamic programming solution (*18, 45*). The other two needed properties, $E \geq 0$ and the Markov property (*18, 45*), are fulfilled by the Axioms 3 and 1 respectively.

To write down dynamic programming (or Bellman) solution for E we must introduce a little more notation. We let π denote the action policy, a function that takes a state s and returns an action a. We let δ be a transition function that takes a state-action pair (s_t, a_t) and returns a new state, s_{t+dt}. For notational simplicity we also redefine E as $F(M_t, a_t)$, and call this the *payoff function* (*18*).

$$F(M_t, a_t) = E(s)$$

subject to the constraints

$$\begin{aligned} a_t &= \pi(s_t) \\ s_{t+dt} &= \delta(s_t, a_t), \\ M_{t+dt} &= f(M_t, s_t) \end{aligned} \tag{1}$$

The value function for F is,

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

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 see the *Mathematical Appendix*. Eq. 3 implies that the optimal action policy π^*_E for E (and F) is a simple greedy policy. This greedy policy ensures that exploration of any finite space S is exhaustive (Theorems 2 and 3 in Mathematical Appendix).

Axiom 5 requires that learning in M converge. Axiom 4 requires information value increases with surprise, re-scaled by specificity. When combined with a greedy action policy like π^*_E, these axioms naturally lead to active learning (*14, 46, 47*) and to adversarial curiosity (*48*).

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.

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

To find an algorithm that maximizes both information and reward value we imagine the policies for exploration and exploitation acting as two possible "jobs" competing to control behavior. For exploration and exploitation we know (by definition) each of these jobs produces non-negative values which an optimal job scheduler could use: E for information or R for reward/reinforcement learning. Finding an optimal scheduler turns out to require we further simplify our assumptions.

We assume we are in a typical reinforcement learning setting, which is where the dilemma finds its simplest expression anyway. In this setting rewards are either present or absent (0, 1). Each action takes a constant amount of time and has no energetic cost. And each policy can only take one action at a time.

Most scheduling problems also assume that the value of a job is fixed, while in our problem information value changes as the memory learns and we expect that rewards are stochastic. However, in a general setting where one has no prior information about the environment the best predictor of the next future value is very often the last value (*45, 49*). We assume this precept holds in all of our analysis.

The optimal solution to a scheduling problem with non-negative values and fixed run times is a deterministic greedy algorithm (*45*). We restate this solution 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 exploration.

Reframing the exploration-exploitation dilemma as a scheduling problem comes at the cost of increasing overall computational complexity (*41*). The worst case run time for π_π is linear and additive in its policies. That is, 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 $T_E + T_R$. 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 $\max(T_E, T_R)$.

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.

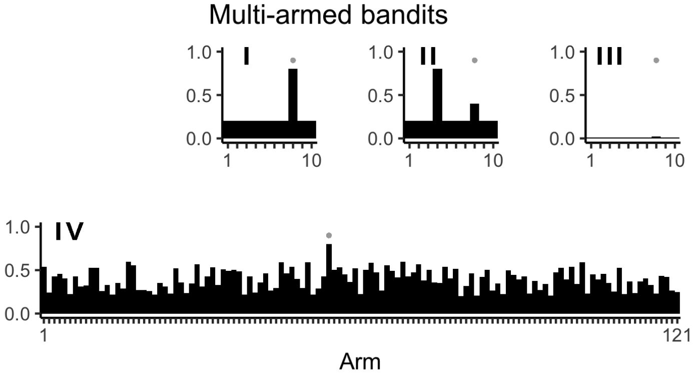

Figure 2: Bandits. Reward probabilities for each arm in bandit tasks I-IV. Grey dots highlight the optimal (i.e., highest reward probability) arm. See main text for a complete description.

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 task (*1*). Here

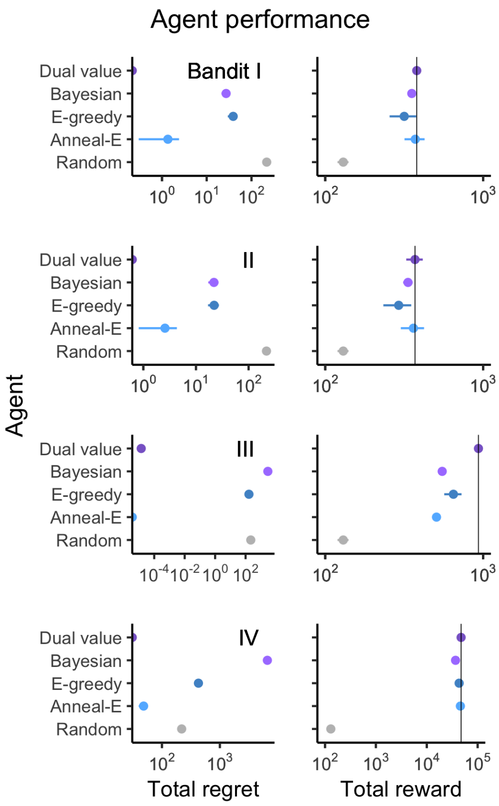

Figure 3: Regret and total accumulated reward across models and bandit task. Median total regret (left column) and median total reward (right column) for simulations of each model type ($N = 100$ experiments per model). See main text and Table 1 for description of each model. Error bars in all plots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

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 using the KL divergence (i.e. relative entropy; Figure 4**A**-**B**). The KL divergence is a standard way to measure the distance between two distributions (*44*) and is, by design, consistent with the axioms (see the Supplementary Materials for a more thorough discussion).

We start with a simple experiment with 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 (*50–52*). 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 (*53*).

We compared the reward and regret performance of 6 artificial agents. All agents used the same temporal difference learning algorithm (TD(0), (*1*)). The only difference between the agents was their exploration mechanism (Table 1; for a complete description see the *Supplementary materials*).

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

Table 1: Artificial agents.

Agent	**Exploration mechanism**
Dual value	Our algorithm (Eq 5).
E-greedy	With probability $1 - \epsilon$ 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 soft-max policy. $\sum_T R_t + \beta E_t$
Random	Action are selected with a random policy (no learning)

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.

Since Schultz *et al*s (*54*) report showing dopamine neurons behave quantitatively like a reward prediction error signal, reinforcement learning has been central to the study of learning (*55*), as well as important to our conception of numerous neurological and psychiatric disorders, including autism (*56*) and schizophrenia (*57*). Reinforcement learning is also undergoing a renaissance in artificial intelligence research, achieving superhuman performance on Atari video games (*58*), chess (*52*), and Go (*59*). However, despite this progress all of reinforcement learning shares a common dilemma without a formal solution. Here we suggest a related problem, a simple competition, that both has a solution itself and has the properties of an ideal solution to the dilemma itself; maximum value with zero regret.

Naturalistic accounts of exploratory behavior at present rely on probabilistic models (*5,60–*

62). Our theory predicts 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.

Our theory also has may lead to new progress in artificial intelligence research, which is limited by three factors: data efficiency, exploration efficiency, and transfer learning (*1, 33, 63*) (see the *Supplementary Discussion* for a complete explanation of these points).

Our most important contribution is perhaps a better worldview on a hard and common problem.

Q: Should we eat at our favorite, or try the new restaurant down the street? What if it's bad? **A**: I'm not sure…

Even in a mundane setting like this question, and its dilemma, the potential loss from exploration is daunting and uncertain to think about. Well beyond the mundane, variations on this problem are universal appearing in psychology, neuroscience, biology, data science, artificial intelligence, game theory, economics, demography, and political science. Here we suggest an universal alternative.

The uncertainty of the unknown can always be recast as an opportunity to learn. But rather than being a trick of psychology, we prove this view is (in the narrow sense of our formalism anyway) mathematically optimal.

Q: Would I rather have this reward, or learn something new? **A**: Which do I value more right now? Pick the biggest.

References and Notes

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

2. S. Thrun, K. Möller, *Advances in neural information processing systems* pp. 531–538 (1992).

3. P. Dayan, T. J. Sejnowski, *Machine Learning* **25**, 5 (1996).

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

5. S. J. Gershman, *Cognition* **173**, 34 (2018).

6. S. Ahilan, *et al.*, *PLOS Computational Biology* **15**, e1007093 (2019).

7. B. Poucet, *Psycholocial Review* **100**, 162 (1993).

8. J. Schmidhuber, *arXiv:1906.04493 [cs]* (2019).

9. Schmidhuber, *Proc. of the international conference on simulation of adaptive behavior: From animals to animats* pp. 222–227 (1991).

10. K. Mehlhorn, *et al.*, *Decision* **2**, 191 (2015).

11. A. A. Gupta, K. Smith, C. Shalley, *The Academy of Management Journal* **49**, 693 (2006).

12. O. Berger-Tal, J. Nathan, E. Meron, D. Saltz, *PLoS ONE* **9**, e95693 (2014).

13. J. Gottlieb, P.-Y. Oudeyer, *Nature Reviews Neuroscience* **19**, 758 (2018).

14. P. Schwartenbeck, *et al.*, *eLife* p. 45 (2019).

15. D. Pathak, P. Agrawal, A. A. Efros, T. Darrell, *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, Honolulu, HI, USA, 2017), pp. 488–489.

16. R. N. Hughes, *Behavioural Processes* **41**, 213 (1997).

17. M. Z. Wang, B. Y. Hayden, *Cognition* **189**, 1 (2019).

18. R. Bellmann, *Bull. Amer. Math. Soc* **60**, 503 (1954).

19. C. Shannon, *The Bell System Technical Journal* **27**, 379 (1948).

20. S. Kakade, P. Dayan, *Neural Networks* **15**, 549 (2002).

21. M. G. Bellemare, *et al.*, *arXiv:1606.01868 [cs, stat]* (2016).

22. P. Dayan, *Neural Computation* **5**, 613 (1993).

23. K. Friston, *et al.*, *Neuroscience & Biobehavioral Reviews* **68**, 862 (2016).

24. H.-K. Yang, P.-H. Chiang, M.-F. Hong, C.-Y. Lee, *arXiv:1905.10071 [cs, stat]* (2019).

25. M. Lopes, T. Lang, M. Toussaint, P.-y. Oudeyer, *NIPS* **25**, 1 (2012).

26. G. Miller, *The Psychological Review* **63**, 81 (1956).

27. E. Tulving, *Annual Review of Psychology* **53**, 1 (2002).

28. I. M. Park, J. W. Pillow, Bayesian Efficient Coding, *Preprint*, Neuroscience (2017).

29. L. Itti, P. Baldi, *Vision Research* **49**, 1295 (2009).

30. J. B. Tenenbaum, T. L. Griffiths, C. Kemp, *Trends in Cognitive Sciences* **10**, 309 (2006).

31. D. P. Kingma, M. Welling, *arXiv:1312.6114 [cs, stat]* (2013).

32. S. Ganguli, D. Huh, H. Sompolinsky, *Proceedings of the National Academy of Sciences* **105**, 18970 (2008).

33. D. Ha, J. Schmidhuber, *arXiv:1803.10122 [cs, stat]* (2018).

34. J. Schmidhuber, *arXiv:1511.09249 [cs]* (2015).

35. V. Mante, D. Sussillo, K. V. Shenoy, W. T. Newsome, *Nature* **503**, 78 (2013).

36. J. Schmidhuber, *arXiv:0812.4360 [cs]* (2008).

37. H. Levi-Aharoni, O. Shriki, N. Tishby, Surprise response as a probe for compressed memory states, *Preprint*, Neuroscience (2019).

38. S. Ganguli, H. Sompolinsky p. 9 (2010).

39. D. Berlyne, *British Journal of Psychology* **41**, 68 (1950).

40. C. Kidd, B. Y. Hayden, *Neuron* **88**, 449 (2015).

41. L. Valiant, *Communications of the ACM* **27**, 1134 (1984).

42. T. Haarnoja, *et al.*, *arXiv:1812.05905 [cs, stat]* (2018).

43. A. Kolchinsky, D. H. Wolpert, *Interface Focus* **8**, 20180041 (2018).

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

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

46. P. Shyam, W. Jaśkowski, F. Gomez, *arXiv:1810.12162 [cs, math, stat]* (2018).

47. D. Pathak, D. Gandhi, A. Gupta, *Proceedings of the 36th International Conference on Machine Learning* p. 10 (2019).

48. J. Schmidhuber, *arXiv:1906.04493 [cs]* (2019).

49. D. Hocker, I. M. Park **15**, 24 (2019).

50. V. Mnih, *et al.* p. 9.

51. D. Silver, *et al.*, *Nature* **529**, 484 (2016).

52. D. Silver, *et al.*, *Science* **362**, 1140 (2018).

53. C. M. Wu, E. Schulz, M. Speekenbrink, J. D. Nelson, B. Meder, *Nature Human Behaviour* **2**, 915 (2018).

54. W. Schultz, *Journal of Neurophysiology* **80**, 1 (1998).

55. E. O. Neftci, B. B. Averbeck, *Nature Machine Intelligence* **1**, 133 (2019).

56. P. Sinha, *et al.*, *Proceedings of the National Academy of Sciences* **111**, 15220 (2014).

57. T. V. Maia, M. J. Frank, *Biological Psychiatry* **81**, 52 (2017).

58. V. Mnih, *et al.*, *Nature* **518**, 529 (2015).

59. D. Silver, *et al.*, *Nature* **529**, 484 (2016).

60. A. J. Calhoun, S. H. Chalasani, T. O. Sharpee, *eLife* **3**, e04220 (2014).

61. M. Song, Z. Bnaya, W. J. Ma, *Nature Human Behaviour* **3**, 361 (2019).

62. E. Schulz, *et al.*, Exploration in the wild, *Preprint*, Animal Behavior and Cognition (2018).

63. T. D. Kulkarni, A. Saeedi, S. Gautam, S. J. Gershman, *Arxiv* **1606.02396v1**, 1 (2016).

64. M. Keramati, B. Gutkin, *eLife* **3**, e04811 (2014).

65. K. Juechems, C. Summerfield, Where does value come from?, *Preprint*, PsyArXiv (2019).

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

67. A. Jaegle, V. Mehrpour, N. Rust, *Arxiv* **1901.02478**, 13 (2019).

68. J. López-Fidalgo, C. Tommasi, P. C. Trandafir, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**, 231 (2007).

69. S. Still, D. A. Sivak, A. J. Bell, G. E. Crooks, *Physical Review Letters* **109** (2012).

70. N. Ay, *Entropy* **17**, 2432 (2015).

71. L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, *arXiv:1603.06560 [cs, stat]* (2016).

72. I. Cogliati Dezza, A. J. Yu, A. Cleeremans, W. Alexander, *Scientific Reports* **7** (2017).

73. J. L. Kelly, *the bell system technical journal* p. 10 (1956).

74. I. M. de Abril, R. Kanai, *2018 International Joint Conference on Neural Networks (IJCNN)* (IEEE, Rio de Janeiro, 2018), pp. 1–6.

75. Y. Burda, *et al.*, *arXiv:1808.04355 [cs, stat]* (2018).

76. C. Colas, P. Fournier, O. Sigaud, M. Chetouani, P.-Y. Oudeyer, *Arxiv* **1810.06284v3**, 1 (2019).

77. E. Even-Dar, S. Mannor, Y. Mansour, *Computational Learning Theory*, G. Goos, J. Hartmanis, J. van Leeuwen, J. Kivinen, R. H. Sloan, eds. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2002), vol. 2375, pp. 255–270.

78. E. Even-Dar, S. Mannor, Y. Mansour, *Journal of Machine Learning Research* **7**, 1 (2006).

79. A. L. Strehl, L. Li, M. L. Littman, *Journal of Machine Learning Research* **10**, 1 (2009).

80. P. Cisek, *Attention, Perception, & Psychophysics* (2019).

81. A. Epshteyn, A. Vogel, G. DeJong, *Proceedings of the 25th International Conference on Machine Learning - ICML '08* (ACM Press, Helsinki, Finland, 2008), pp. 296–303.

82. S. B. Thrun, *Technical Report* pp. 1–44 (1992).

83. S. Ishii, W. Yoshida, J. Yoshimoto, *Neural Networks* **15**, 665 (2002).

84. H. Zhang, A. J. Watrous, A. Patel, J. Jacobs, *Neuron* **98**, 1269 (2018).

85. A. Zhang, N. Ballas, J. Pineau, *arXiv:1806.07937 [cs, stat]* (2018).

86. A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, *Arxiv* **1606.05312v2**, 1 (2018).

Acknowledgments: EJP and TV wish to thank Jack Burgess, Matt Clapp, Kyle Donovank, Richard Gao, Roberta Klatzky, Jayanth Koushik, Alp Muyesser, Jonathan Rubin, and Rachel Storer for their comments on earlier drafts. EJP also wishes to thank Richard Grant for his illustration work in Figure 1. **Funding:**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. **Authors contributions:** EJP conceived of the idea, designed the research and developed the proofs, and wrote the manuscript. TV designed the research and wrote the manuscript. **Data and materials availability:** All code and data is available at `https://github.com/CoAxLab/infomercial`.

Mathematical Appendix.

Compactness

The compactness C of a hyper-cube has a simple formula, $C = \frac{P^2}{A}$ where P is the cube's perimeter and A is its area. Perimeter and area are of course defined by a set of distances D. Traditionally these would be euclidean distances, but any notion of distance should work.

To review, our world model M is simply a finite set of N real values. It therefore seems fair to assume one can always find, or define, a suitable axiomatic distance to measure how elements in M changed with time and learning, as M_t moves to M_{t+dt}. This distance might be measured on the elements of M directly or, as a convenient proxy, using memories taken from M by the decoder function g (which we define in the main text).

With any set of distances $D = \{d_i, d_{i+1}, d_{i+2}, \dots d_O\}$ defined for all observed states $Z \subseteq S$, we can imagine they form a O-dimensional hyper-cube ($O \leq N$). By measuring this imagined cube we find a geometric estimate for compactness (Eq. 6). Compactness defined this way encompass both specific changes to, and compression of, the representation in M.

$$C = \frac{\left(2 \sum_i^O d_i\right)^2}{\prod_i^O d_i} \tag{6}$$

Information value as a dynamic programming problem

To find greedy dynamic programming (*1, 45*) 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 (*45*).

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. 7).

$$
\begin{aligned}
V^*_{\pi_E}(M_0) &= \max_{a \in A} \Big[\sum_{t=0}^{\infty} F(M_t, a_t) \Big] \\
&= \max_{a \in A} \Big[F(M_0, a_0) + \sum_{t=1}^{\infty} F(M_{t+dt}, a_{t+dt}) \Big] \\
&= F(M_0, a_0) + \max_{a \in A} \Big[\sum_{t=1}^{\infty} F(M_{t+dt}, a_{t+dt}) \Big] \\
&= F(M_0, a_0) + V^*_{\pi_E}(M_{t+dt}) + V^*_{\pi_E}(M_{t+2}),\ \ldots
\end{aligned} \tag{7}
$$

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. 8).

$$a \leq b \Leftrightarrow \exists\, c;\ b = a + c \tag{8}$$

$$a > b \Leftrightarrow (a \neq b) \wedge (b \leq a) \tag{9}$$

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 $\mathbf{E} = (E_1, E_2, ...)$ be ranked series of E values for all states S, such that $(E_1 \geq E_2, \geq ...)$. To swap any pair of values $(E_i \geq E_j)$ so $(E_i \leq E_j)$ by Eq. 8 $E_i - c = E_j$.

Therefore, again by Eq. 8, $\exists \int \delta E(s) \to -c$.

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 (*45*).

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 (*64, 65*).

Optimal scheduling for dual value learning problems

In classic scheduling problems the value of any job is known ahead of time (*18, 45*). 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 (*66*). 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 (*49*). 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 (*45*) 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. □

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

Agents

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* ((*50*)). 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, 67*). To provide a lower bound benchmark of performance we included an agent with a purely random exploration policy.

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 S2**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 (*50–52*). 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 (*53*).

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 4**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 (*68*), compression (*?, 69*) and information geometry (*70*), to name a few examples. KL has seen extensive use in reinforcement learning.

The Kullback–Leibler (KL) divergence satisfies all five value axioms (Eq. 10).

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. 10) after observing some state s.

$$KL(M_{t+dt}, M_t) = \sum_{s \in S} M_{t+dt}(s) \log \frac{M_{t+dt}(s)}{M_t(s)} \tag{10}$$

Axiom 1 is satisfied by limiting E calculations to successive memories. Axiom 2-3 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 5 more concrete, in Figure S5 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 S5**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.*).

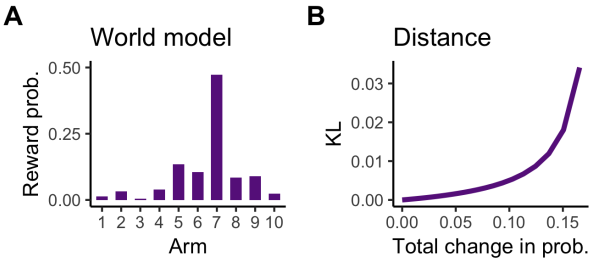

Figure 4: A world model for bandits. **B**. Example of a single world model suitable for all bandit learning. **B** Changes in the KL divergence–our choice for the distance metric during bandit learning–compared to changes in world model, as by measured the total change in probability mass.

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

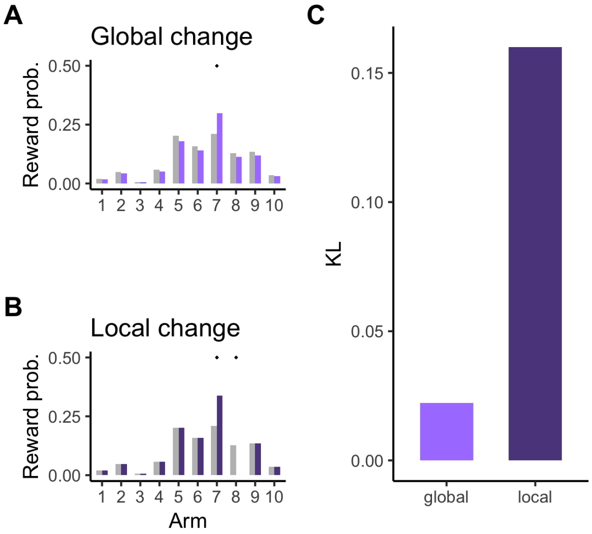

Figure 5: An example of observation specificity during bandit learning. **A**. A initial (grey) and learned (distribution), where the hypothetical observation s increases the probability of arm 7 by about 0.1, and the expense of all the other probabilities. **B**. Same as A except that the decrease in probability comes only from arm 8. **C**. The KL divergence for local versus global learning.

this uniform prior into the appropriate units for our KL-based E using Shannon entropy, $E_0 = \sum_K p(a_k) \log p(a_k)$.

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. 11). 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.

$$V(s) = V(s) + \alpha(\mathbf{R}_t - V(s) \tag{11}$$

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

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 (*71*). For a description of hyperparameters seen Table S1, and for the values themselves Table S**??**.

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 6**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 6**A**) we include an example of value learning value learning in our agent.

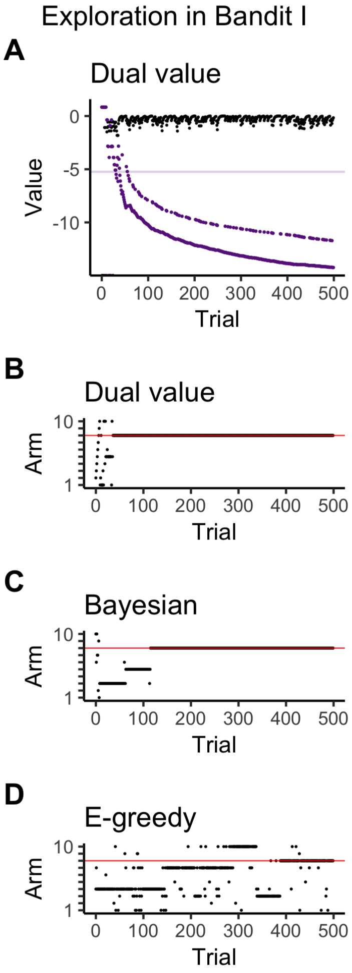

Figure 6: Exploration and value dynamics. **A**. An example of our dual value learning algorithm during 500 trials on Bandit. The light purple line represents the boredom threshold η (Eq. 5). **B.** An example of exploration dynamics (i.e arm selection) on Bandit. Note how the search is structured, and 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.

Supplementary discussion

Past work

We are certainly not the first to quantify information value (*43, 72*), or use that value to optimize reward learning (*3, 9, 29, 73, 74*). 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 information is treated as an analog of these tangible or external rewards. 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, 39, 67*) or PAC approaches (*41*). Curiosity learning is not generally known to converge on rewarding actions with certainty, but never-the-less can be an effective heuristic (*15, 75, 76*). Within some bounded error, PAC learning is certain to converge (*41*). For example, to find the most rewarding arm in a bandit, and will do so with a bounded number of samples (*77*). However, the number of samples is fixed and based on the size of the environment (but see (*78, 79*)). So while PAC will give the right answer, eventually, its exploration strategy also guarantees high regret.

Animal behavior

Cisek (2019) traced the evolution of perception, cognition, and action circuits from the Metazoan to the modern age (*80*). The circuits for reward exploitation and observation-driven exploration appear to have evolved separately, and act competitively–exactly the model we suggest. In particular he notes that exploration circuits in early animals were closely tied to the primary sense organs (i.e. information) and, historically anyway, had no input from the homeostatic circuits needed for reward valuation (*64, 65, 80*).

In psychology and neuroscience, curiosity and reinforcement learning have developed as separate disciplines (*1, 39, 40*). And indeed they are separate problems, with links to different basic needs–gathering resources to maintain physiological homeostasis (*64, 65*) and gathering information to plan for the future (*1,41*). Here we prove that though they are separate problems, they are problems that many ways they solve each other.

The theoretical description of exploration in scientific settings is probabilistic (*5, 60–62*). 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 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.

Data efficiency refers to how many samples or observations it takes to make a set amount of learning progress. The most successful reinforcement learning algorithms are highly data inefficient. For example, Q-learning (*58*). To make reinforcement learning data inefficient generally requires one include a (world) model in the reinforcement algorithm itself. In the challenging environments modern methods must learn, it this model exists. A dual algorithm offers a good compromise. Exploration learns a world model. As this model improves it can be used directly by the reinforcement learning policy, potentially leading to substantial improvement in data efficiency. The specialist could think of this as a loose generalization of the successor model (*6, 22, 63*).

In the large and complex environments modern machine learning operates in random exploration takes to long to be practical. So there is a critical need for more efficient exploration strategies (*33*), often known as active sampling or directed exploration. A range of heuristics for exploration have been explored (*13, 21, 42, 81–83*). Dual value algorithms offer a new and principled approach. Designing efficient exploration reduces to two questions: what should our agent remember? How should we measure change change in that memory? Subject to the axioms and Eq. 5, of course.

Deep reinforcement learning can match or exceed human performance in games like chess (*52*), Go (*51*), as well as less structured games like classic Atari video games (*58*). It It would be ideal to transfer performance from one task to another without (much) retraining (*1*). But with even minor changes to the environment, most artificial networks often cannot adapt (*84, 85*). This is known as the transfer problem. One thing which limits transfer is that many networks are trained end-to-end, which simultaneously (implicitly) learns a world model and a strategy for maximizing value. Disentangling these can improve transfer. We're therefore not the first to suggest using a world model for transfer (*33, 86*). What we offer is a simple and optimal algorithm to combine nearly any world model with any reinforcement learning scheme.

Cost

It's not fair to talk about benefits without also discussing costs. The worst-case run-time of a dual value algorithm is $\max(T_E, T_R)$, where T_E and T_R represent the time to learn to some criterion (see *Results*). In the unique setting where minimizing regret, maximizing data efficiency, exploration efficiency, and transfer do not matter, dual value learning can be a suboptimal choice.