Specific Machine Curiosity

by

Nadia Michelle Ady

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Curiosity appears to motivate and guide effective learning in humans, which has led to high hopes in the machine learning community for machine analogues of curiosity. While a variety of machine curiosity algorithms have been introduced, they are rarely compared with other existing curiosity algorithms. With a new family of experimental domains—'Curiosity Bandits'—I provide a means of observing curiosity methods on an even playing field, manipulating the curiosity mechanism while controlling for the learning algorithm and its environment. Observations using these domains, along with the study of human and animal curiosity, allowed me to clarify five properties that would offer important benefits for machine learners but have not yet been well-explored in machine intelligence—directedness towards inostensible referents, cessation when satisfied, voluntary exposure, transience, and coherent long-term learning. I further demonstrate how three of these properties can be implemented together in a proof-of-concept reinforcement learning agent. As a whole, this work presents a novel view into machine curiosity and how it might be integrated into the behaviour of goal-seeking, decision-making machine agents in complex environments.

Preface

I have been fortunate to be part of excellent collaborative teams while undertaking the research presented in this dissertation.

Chapter 3 and Appendix C are based on collaborative work with P. M. Pilarski, C. Linke, T. Degris, A. White, and M. White. Aspects of the first study have been shared at three workshops (Ady and Pilarski, 2016; Ady, 2017a; Ady and Pilarski, 2017b), while the second study has been published as part of a journal article (Linke et al., 2020). With regards to the journal article, I was responsible for the concept formation and initial experimental design and contributed to manuscript editing. The work presented with Linke et al. (2020) has also been presented at two conferences (Linke et al., 2021, 2019). Appendix B, which is also closely related to the work presented in Chapter 3, is based on a publicly-available technical report (Ady, 2017c).

Chapters 4 and 5 are under review with the *Journal of Artificial Intelligence Research*, as N. M. Ady, R. Shariff, J. Günther, and P. M. Pilarski, "Five Properties You Didn't Know Machines Should Have," which has also been published as a preprint on arXiv at `https://arxiv.org/abs/2212.00187`. An earlier presentation, which mainly focused on the experimental side of the work as presented in Chapter 5, was given at the Multidisciplinary Conference on Reinforcement Learning and Artificial Intelligence (Ady et al., 2022b). I held responsibility for the entirety of the work. R. Shariff assisted with concept formation and advised on implementation of the experiments. J. Günther contributed to manuscript editing.

In each of the contributions aside from the collaboration with Linke et al.

(2020), P. M. Pilarski was the supervisory author and was involved with concept formation, experiment design and implementation, and manuscript composition and editing.

> Ady, N. M. and Pilarski, P. M. (2016). Domains for investigating curious behaviour in reinforcement learning agents. 11th Women in Machine Learning Workshop (WiML 2016)
>
> Ady, N. and Pilarski, P. (2017a). Comparing reinforcement learning methods for computational curiosity through behavioural analysis. In *3rd Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, page 88, Ann Arbor, Michigan, USA
>
> Ady, N. M. and Pilarski, P. M. (2017b). Unifying curious reinforcement learners. In *Designing for Curiosity: An Interdisciplinary Workshop, ACM CHI Conference on Human Factors in Computing Systems (CHI 2017)*, Denver, CO, USA. May 6–11
>
> Ady, N. M. (2017c). Parameter screening for curious reinforcement learner motivated by unexpected error. doi.org/10.7939/R3G15TS0P. Department of Computing Science, University of Alberta Education & Research Archive
>
> Linke, C., Ady, N. M., White, M., Degris, T., and White, A. (2020). Adapting behavior via intrinsic reward: A survey and empirical study. *Journal of Artificial Intelligence Research*, 69:1287–1332
>
> Ady, N. M., Shariff, R., Günther, J., and Pilarski, P. M. (2022b). Prototyping three key properties of specific curiosity in computational reinforcement learning. In *5th Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, June 8-11, Providence, Rhode Island

Ady, N. M., Shariff, R., Günther, J., and Pilarski, P. M. (2022a). Five properties of specific curiosity you didn't know curious machines should have. arxiv.org/abs/2212.00187. Submitted to the *Journal of Artificial Intelligence Research*

This dissertation is designed to be a coherent story of my central contributions during my program. However, other research completed during this program is not included in this text.

Ady, N. M. (2017b). Curious actor-critic reinforcement learning with the dynamixel-bot. https://doi.org/10.7939/R3B853Z7S. Department of Computing Science, University of Alberta Education & Research Archive

Ventura, J., Ady, N. M., and Pilarski, P. M. (2017). An exploration of machine curiosity and reinforcement learning using a simple robot. https://doi.org/10.7939/R36W96Q00. WISEST Summer Research Program Posters, University of Alberta Education & Research Archive

Günther, J., Ady, N. M., Kearney, A., Dawson, M. R., and Pilarski, P. M. (2020). Examining the use of Temporal-Difference Incremental Delta-Bar-Delta for real-world predictive knowledge architectures. *Frontiers in Robotics and AI*

Ady, N. M. and Rice, F. (2023). Interdisciplinary methods in computational creativity: How human variables shape human-inspired AI research. In *Proceedings of the 14th International Conference on Computational Creativity (ICCC'23)*

Acknowledgements

Thank you so much to Patrick Pilarski, Janet Droppo, Michael Ady, Roshan Shariff, Faun Rice, Kate Pratt, Johannes Günther, Alex Kearney, James Austin, Brian Tanner, Anna Koop, Raksha Kumaraswamy, Niko Yasui, Nathan Wispinski, Aristotle Kollias, Juliana Virani, Martha White, Craig Chapman, Emma McDonald, Carrie Demmans Epp, Gerald Häubl, İlhan İnan, Marlos Machado, and David Silver. You have all contributed in your own ways to this dissertation and I am grateful for it! Thank you to the agencies that provided funding to support this research, including NSERC, CIFAR AI Chairs Program, Women Techmakers, and the Government of Alberta.

Contents

List of Tables

List of Figures

List of Symbols

(x, y) the open interval between x and $y \in \mathbb{R}$; a continuous subset of the real line $\mathbb{R}$ excluding both end points . . . . xv, 83

A_t action at time t . . . . xxii, 19, 22, 57, 72, 74, 76–78, 141

$C_{t,i}$ the value of target signal i at time t, drawn from distribution $\theta(t, i)$ . . . . xvi, 73, 74, 76, 84, 86, 87

$D_{t,i}$ the amount of drift between the mean of the distribution of target signal i at time t and at time t, not accounting for clipping to a closed range; that is, $\mu_{t+1,i} \leftarrow \Pi_{[x,y]} (\mu_{t,i} + D_{t,i})$ . . . . xiv, 74, 75

G_t return at time t, defined for the episodic setting in Equation 2.2 and for the discounted setting in Equation 2.3 . . . . 20–22, 142

H a numerical preference function, as used to determine a soft-max distribution . . . . xxiv, 77, 78, *Glossary:* soft-max

$I(X; Y)$ the mutual information of two random variables X and Y:

$$I(X; Y) \coloneqq \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

(Cover and Thomas, 1991, p. 6) . . . . 41–43, 45, *Glossary:* mutual information

N the number of learning tasks considered in a multi-prediction learning setting 72, 73, 76–78

O denotes an observation of the environment, which may not perfectly capture the state; may appear as O_t denoting the observation at time t, or, with slight abuse of notation, as $O(S_t)$, denoting a mapping from the state S_t to the information available at time t about the state 22

Q estimated or learned action-value function . . . . 22, 57, 62, *Glossary:* action-value

R_t^I intrinsic reward at time t . . . . 58, 60, 61, 74, 76, 83, 146, 251, *Glossary:* intrinsic reward (IR)

R_t (extrinsic) reward at time t . . . . xxii, 19–22, 57, 77, 141, 145, *Glossary:* extrinsic reward

S_t state at time t; $S_t \in \mathcal{S}$ . . . . xiv, xxii, 19, 21, 22, 45, 57, 141–143

T a timestep 20

$U(x, y)$ the continuous uniform distribution on the open interval (x, y), where every point in (x, y) has equal probability of being sampled; for detail, consider Wasserman (2005, pp. 27–28) 79

$Z_t^{(i)}$ the 'surprise' or 'unexpected prediction error' about general value function prediction i at time t, as defined by White et al. (2014); see Equation 2.9 . . . . 37, 38, *See:* UDE

$[x, y)$ the half open interval between x and $y \in \mathbb{R}$, including x but not y, a continuous subset of the real line, $\mathbb{R}$ 251, 252

$[x, y]$ the closed interval between x and $y \in \mathbb{R}$; a continuous subset of the real line $\mathbb{R}$ including both end points . . . . xv, 20, 44, 56, 75

$\Pi_{[x,y]}$ a projection function that bounds the input to the range $[x, y]$:

$$\Pi_{[x,y]}(z) := \begin{cases} x & \text{for } z < x \\ z & \text{for } z \in [x, y] \\ y & \text{for } x > y \end{cases} \tag{1}$$

. . . . xiv, 74, 75

α a step size/learning rate . . . . xvi, xxiv, 57, 62, 63, 73, 76, 77, 81, 82, 84, 85, 89, 91, 103

$\bar{R}_t$ the average of all the rewards up to time t, maintained using an unbiased exponential average initialized to zero (Sutton and Barto, 2018, Eq. 2.9) 77, 78

β the *smoothing parameter* or *averaging constant* (White, 2015, p. 121) used in an unbiased exponentially-weighted average . . . . . xvi, xvii, xxi, xxv, 83, 84, 87, 92, 93, 108, 109

δ_t the prediction error (typically temporal-difference (TD) error) at time t . . . xix–xxi, xxiv, 21, 22, 37, 57, 58, 60, 61, 73, 83–86, 89

η a smoothing parameter used in the computation of Learning Progress as specified by Oudeyer et al. (2007, p. 271). This smoothing is achieved by averaging the error samples over η time steps . . . . . . xi, xvi, xix, 35, 60, 83, 92, 93, 99, 100, 107

γ discount rate . . . . 20–22, 57, 62, 142, 143, 146, 162, 164, 166, 167, 189, 190, 249–253, 256, 258, 259, *Glossary:* discount rate

$\hat{C}_{t,i}$ a learner's estimate of target signal $C_{t,i}$ . . . . . . xvi, xx, xxv, 73, 74, 84, 85, 87

$\hat{\nu}_{t,i}$ an estimate of the variance, $\mathrm{var}\left[\hat{C}_{t,i}\right]$ (of the estimated target signal) obtained using an exponentially-weighted average variant of Welford's algorithm: $\nu^{(y)}_{t,i} = (1-\beta)\nu^{(y)}_{t-1,i} + \beta(C_{t,i} - \hat{C}_{t-1,i})(C_{t,i} - \hat{C}_{t,i})$ . . . . xxv, 84, 85

$\hat{v}_\pi$ estimated or learned value function, given the agent follows policy π . . . 21, 22

κ the meta learning-rate for Autostep, which controls how quickly the step size, α, changes 80, 81

$\mathbb{E}_\pi$ the expected value of a random variable given the agent follows policy π . . . 21, 22

$\mathbb{E}$ the expected value 73, 74, 249

$\mathbb{R}_{>0}$ the set of positive real numbers, $\{x \in \mathbb{R} \mid x > 0\}$ 74

$\mathbb{R}$ the set of real numbers . . . . xiv–xvi, xxiv, 19, 21, 73, 74, 77, 80, 141, 249

$\mathcal{N}$ the normal (or Gaussian) distribution with mean (or center) μ and variance σ^2 . . . . 74, *See also:* normal

$\mathfrak{E}_t$ the empowerment at time t 43

$\mathcal{A}(s)$ set of all possible actions available from state $s \in \mathcal{S}$ . . . . xxiv, 19, 56

$\mathcal{S}$ set of all possible states . . . . xv, xvi, 19, 21, 56, 249, 250

μ the mean of a distribution . . . . xiv, 74, 75, 79

$\nu^{(y)}_{t,i}$ an estimate of the variance, $\mathrm{var}\left[C_{t,i}\right]$, of target signal $C_{t,i}$ obtained using an exponential average variant of Welford's algorithm: $\nu^{(y)}_{t,i} = (1-\beta)\nu^{(y)}_{t-1,i} + \beta(C_{t,i} - \hat{C}_{t-1,i})(C_{t,i} - \hat{C}_{t,i})$ . . . . xvi, xvii, 84, 87, 88

$\nu_{t,i}$ the posterior variance estimate at time t for task i used in the computation of Bayesian Surprise (see Table 3.3); defined by $\max(\nu^{(y)}_{t,i}/d_t, 10^{-3})$ .... xx, 84, 88

π a policy .... xvi, xviii, 20, 21, 77, 78

σ the standard deviation or 'spread' of a distribution 74, 75, 78, 79, 87, 88

τ a time window parameter used in the computation of Learning Progress as specified by Oudeyer et al. (2007, p. 271). The computation of Learning Progress compares the averaged (smoothed) error of two time windows τ apart ... xi, xvii, xix, 35, 60, 83, 92, 93, 99, 100, 107

Pr the probability of .... xxii, xxiv, 19, 78

var the variance, or sample variance, of a distribution, as denoted with each use in the text .... xvi, 37, 61, 84

`MSE` Mean Squared Error (MSE) 73, 77

$\theta(t, i)$ a distribution associated with learning task i at time t ... xiv, xvii, 35, 73, 74, 76, 78

ε probability of taking a random action in an ε-greedy policy 29, 57, 58, 61, 62, 251–253, 256–259

ξ a standard deviation parameter to control the rate of drift of the distribution, $\theta(t, i)$ of task i 74, 75, 78, 79

a an action .... xvii, 19, 22, 29, 44, 56, 57, 62, 77, 78, 249, 250

c a constant, typically a parameter 29, 84, 248

d_t a count of time that decays by a factor of $1 - \beta$ at each time step, defined by $d_t = (1 - \beta)d_{t-1} + 1$ and used to compute Bayesian Surprise as shown in Table 3.3 .... xvii, 84, 87, 88

i index denoting one of a discrete set of learning tasks ..... xiv, xv, xvii, xix–xxi, xxiv, xxv, 37, 72–76, 78, 79, 81, 83–89, 91, 103

$n_{s,a}$ a count of the number of time steps that have passed since the action a was last taken from state s 29

n a sequence-length parameter used in the computation of empowerment (Klyubin et al., 2005, p. 130) 43

q_π action-value function, given the agent follows policy π .... 22, *Glossary:* action-value

s a state .... xvii, 19, 21, 22, 29, 56, 57, 62, 249, 250

t time step..... xiv–xvii, xix–xxi, xxiv, xxv, 19, 20, 22, 37, 43, 56, 57, 61, 72–78, 83–89, 103, 141, 146, 174, 248, 249

v_π value function, given the agent follows policy π 21, 24

w a weight .... xxv, 73, 74, 76, 85, 86, 89, 91

$:=$ is defined to be........................ xiv, xv, xxiv, 20–22, 41, 73–75, 78, 89

Glossary

Absolute Error an intrinsic reward proposed by Schmidhuber (1991b) formalizing a measure of violated expectations using prediction error, defined in this work as:

$$|\delta_{t,i}|$$

. . . . xxi, 85, 89, 90

Absolute Value of Learning Progress an intrinsic reward approximating the amount of recent change in a learner's predictions defined in this work by:

$$\left| \frac{1}{\eta+1} \sum_{j=0}^{\eta} \delta^2_{t-j-\tau,i} - \frac{1}{\eta+1} \sum_{j=0}^{\eta} \delta^2_{t-j,i} \right|$$

. . . . xi, xii, 83, 88, 92–94, 99–101, 104–107, *See also:* Learning Progress

action-value the expected return when starting in a particular state, taking a particular action, and following a given policy thereafter (Sutton and Barto, 1998, p. 68), also known as the Q-value . . . xv, xviii, xix, xxiii, 29, *Symbols:* q_π & Q

ADADELTA a meta-learning algorithm, introduced by Zeiler (2012), for adapting the step-size parameter for gradient descent 82

Adam a meta-learning algorithm, introduced by Kingma and Ba (2015), for adapting the step-size parameter for stochastic optimization methods . . . 82, 110

Arcade Learning Environment a popular evaluation platform consisting of a "software framework for interfacing with emulated Atari 2600 game environments" (p. 254) proposed by Bellemare et al. (2013) that allows for the evaluation of artificial intelligence agents on numerous Atari 2600 video games (Machado et al., 2018) 2, 70, 244

Autostep a meta-learning algorithm, introduced by Mahmood et al. (2012), for adapting the step-size parameter of a least-mean-squares (LMS) learner over time . . . . xi, xvi, 80–82, 92, 93, 95, 103–107, 110

Bayesian Surprise an intrinsic reward attributed to Itti and Baldi (2006) formalizing the idea of amount of learning; in this work, our learners are not Bayesian, so we use an approximation of Bayesian Surprise defined by:

$$\frac{1}{2}\log_2\left(\frac{\nu_{t,i}}{\nu_{t-1,i}}\right)+\frac{\nu_{t-1,i}+(\hat{C}_{t-1,i}-\hat{C}_{t,i})^2}{2\nu_{t,i}}-\frac{1}{2}$$

. . . . 72, 84, 86–88, 90, 93, 99, 101, 105

channel capacity a measure of how much information from a transmitted signal can possibly be received given a particular channel 42, 43

cumulant a signal that is added up in a general value function prediction, analogous to the way the reward signal is added up for a value function prediction (see Sutton and Barto, 2018, p. 459); also called a *pseudo-reward* (e.g., Sutton et al., 2011, p. 761) 54, 68

Curiosity Bandit an instance of a family of experimental domains, introduced by Ady and Pilarski (2016), for highlighting differences between methods inspired by curiosity . . . . x, xi, 11, 54–56, 62, 71, 72, 111, 113, 276

discount rate a parameter of a learning problem, also called a *continuation probability*, that determines the present value of future rewards, weighting "rewards in the near future more than those in the far future" (Sutton and Barto, 2018, p. 55); roughly reflects the traditional 'exponential discounting' model of intertemporal choice from behavioural economics—our tendency to care more about near-term gains and losses over those further into the future (Berns et al., 2007, pp. 482–483) . . . . 20, 21, 142, 249, 252, *Symbol:* γ

Dyna-Q a simple architecture which allows the integration of planning, acting, and learning (Sutton and Barto, 1998, p. 230) 28–30, 152, 153

Error Reduction an intrinsic reward inspired by Schmidhuber (1991a) meant to measure the "(positive or negative) change of assumed reliability caused by the [most recent] observation" (p. 1461), in this work defined as:

$$|\delta_{t-1,i}|-|\delta_{t,i}|$$

. . . . 84, 86, 90, 99, 101

exploration–exploitation dilemma the challenge of determining how to choose between an "exploitative" action with the best expected payoff and an "exploratory" action which learns more about the environment and might result in finding an action with an even better payoff (Sutton and Barto, 2020, p. 3) .... xxi, 23, 29

Expected Error an intrinsic reward designed to measure violated expectations, but with less noise than Absolute Error or Squared Error, defined by:

$$\left|\overline{\delta_{t,i}}^{\beta}\right|$$

.... 83, 85, 93, 97, 104, 108, 109

exploration bonus a term added to the reward or value function computation for the purpose of encouraging explorative behaviour; to the best of our knowledge (see Section 2.3.1.1), the first exploration bonus was proposed by Sutton (1990b) 28–30, 39, 70, 110, 152

exponentially-weighted average an estimate of the average of a sequence of samples where more recent samples are weighted more heavily, following an exponential decay (Sutton and Barto, 2018, Eq. 2.9); the exponentially weighted average, $\overline{x_t}^{\beta}$, of a sequence of real numbers, $x_0, x_1, x_2, \ldots$, is computed as $\overline{x_t}^{\beta} \leftarrow (1-\beta)\overline{x_{t-1}}^{\beta} + \beta x_t$, where β is a smoothing parameter ... xv, xvi, 37, 84, 85, 87, 108

extrinsic motivation being moved to do something because it leads to a separable outcome (Ryan and Deci, 2000, pp. 54–55) ..... xxii, 25, 68, *See also:* extrinsic reward

extrinsic reward a term used in contrast to intrinsic reward; used in the context of psychology to refer to separable consequences of behaviour such as food or money; in the computational literature, usually refers to the reward provided by the environment or as part of the problem a learner is set to solve that neither the computational learner nor the designer of its algorithm has any control over .... 25–27, 56–61, 68, *See also:* extrinsic motivation

gradient bandit algorithm a bandit algorithm described by Sutton and Barto (2018, pp. 37–40) which learns action preferences and probabilistically chooses actions based on a soft-max distribution (p. 42) 76–78, 81, 91, 93, 97–100, 104, 107, 110

intrinsic motivation (IM) when used in the context of machine intelligence—that is to say, *computational* intrinsic motivation—refers to a motivational structure defined based on changes in internal computational structures, as opposed to computational extrinsic motivation, which refers to a motivational structure defined as part of the external problem the learner is set to solve; in the context of biological intelligence, refers to "general motivations that push [some animals] to explore, manipulate or probe their environment, fostering curiosity and engagement in playful and new activities" (Oudeyer and Kaplan, 2007, p. 1)..... xxvi, 8, 9, 14, 16, 18, 25–28, 32–34, 40, 49, 52, 85, 210

intrinsic reward (IR) in psychology, this term has been used to refer to the "internal condition" brought about by engagement in intrinsically motivated activities (Deci, 1975, p. 118), whereas in machine intelligence, generally refers to reward signals that can be computed from the sensorimotor context, reflecting changes in the "knowledge and know-how" of the agent (Oudeyer and Kaplan, 2007, p. 12), independent of any actual meaning a designer might attribute to the sensor observations, allowing intrinsic rewards to be agnostic to the environment; compare with Section 2.3.2............... xi, xx, xxi, xxiv, xxv, xxvii, 17, 26, 27, 31, 33, 39, 40, 48, 49, 55, 57, 58, 60–62, 69–75, 77, 79, 81, 83, 85, 86, 88–112, 145–147, 149–152, 156, 159, 210, 211, 247, 250, 251, 258, *See also:* intrinsic motivation (IM)

introspective learner a learner that can autonomously increase its learning rate when progress is possible, and decrease learning when progress is not—or cannot—be made....................... xii, 72, 79, 80, 82, 83, 89, 91–111

Learning Progress Oudeyer et al. (2007) defined Learning Progress as part of their Intelligent Adaptive Curiosity (IAC) mechanism..... xvi, xvii, 27, 35, 60, 65, 70, 274

Markov property an environment has the Markov property if the current state and action provide no less information about the next state and reward than do all preceding states and actions; formally:

$$\begin{aligned}&\Pr\left\{S_{t+1}=s, R_{t+1}=r \mid S_t=s_t, A_t=a_t\right\}\\=&\Pr\left\{S_{t+1}=s, R_{t+1}=r \mid S_0=s_0, A_0=a_0, ..., S_t=s_t, A_t=a_t\right\}\end{aligned}$$

.... 19, 20, 22

mutual information a measure of the dependence between the two random variables; the reduction in uncertainty due to another random variable..... 41, 42, 45, 86, *Symbol:* $I(X;Y)$

normal a normal (or Gaussian) distribution is a standard distribution that is sometimes referenced by its “bell curve” shape; Wasserman (2005, p. 28) offers a formal definition .... xvi, 74, *Symbol:* $\mathcal{N}$

option a mathematical abstraction of short-term policies attributed to Sutton et al. (1999) 68, 157

partial monitoring a family of learning problems that generalizes the multi-armed bandit framework to include problems in which the loss may not be “directly observed by the learner” (Lattimore and Szepesvári, 2019, p. 1)—in contrast to typical bandits, in which the learner receives the loss of the chosen action as feedback at each step (Lattimore and Szepesvári, 2020, Ch. 37) 76

prediction learner an instance of a reinforcement learning algorithm that only predicts aspects of its environment (e.g. estimates a general value function) and does not select actions or optimize behaviour ...... xxv, 56, 57, 62, 64, 71–77, 80, 82, 83, 86–89, 91–97, 101, 103, 108, 110, 111, 249, 250

Predictive Power a component of the information-theoretic approach to curiosity-driven reinforcement learning designed by Still and Precup (2012) 114, 276

project a shorthand originally used by Simpson (1976) in reference to behaviour that gives the impression of practice towards a particular desired outcome behaviour (in the original case, used to describe “captive rhesus monkey infants of six to 12 weeks [repeating] sequences of activity involving particular leaps, particular climbs and particular routes,” p. 386) 39, 40, 276

Q-learning an algorithm for learning action-values for an optimal policy using a temporal-difference update rule; according to Sutton and Barto (2018, p. 140), the algorithm was introduced by Watkins (1989) 22, 29, 57–59, 61, 62, 68, 71, 110, 251

Q-value synonym for action-value .... xix, 29, *See:* action-value

RMSProp (thought to be short for ‘root mean square propagation’) an optimization method for the training of artificial neural networks (ANNs) involving the adjustment of step sizes, commonly credited to Hinton et al. (2012) ... 82

Sarsa an algorithm for predicting action-values (usually for the purpose of control) using a temporal-difference update rule; according to Sutton and Barto

(2018, p. 139), the algorithm was introduced by Rummery and Niranjan (1994) and dubbed ‘Sarsa’ by Sutton (1995) 22, 56, 62, 112

Simple Simulated Robot Experiment an experiment in which the open-ended behaviour of a simple simulated robot (a box with two wheels, a sound emitter, and a sensor allowing it to perceive its distance from a small toy) is observed, originally showcased by Oudeyer et al. (2007, p. 273) . . . xi, xxiv, 35, 36, 274, 276

soft-max a soft-max distribution, also known as a Gibbs or Boltzmann distribution, defines probabilities for a discrete set of alternatives $\mathcal{A}$ parameterized by a numerical preference function $H : \mathcal{A} \to \mathbb{R}$ such that a random variable A with a soft-max distribution parameterized by H is defined by the following probabilities:

$$\Pr\{A = a\} \coloneqq \frac{e^{H(a)}}{\sum_{b \in \mathcal{A}} e^{H}(b)} \tag{2}$$

. . . . xiv, xxi, 29

specific curiosity the cognitive, emotional, and embodied condition humans imply by saying, “I am curious to know X.” 118, 121, 159

Squared Error an intrinsic reward used by Gordon and Ahissar (2011), formalizing a measure of violated expectations using prediction error, defined in this work as:

$$\delta^2_{t,i}$$

. . . . xxi, 84, 85, 95, 99, 101, 106, *See also:* Absolute Error

Step-size Change an intrinsic reward based on the amount of change in the step-size parameter, defined by:

$$|\alpha_{t-1,i} - \alpha_{t,i}|$$

. . . . 84, 103

tracking a form of memory adaptation that occurs by allowing new, current information to replace information that was previously in memory (Koop, 2008, p. 15) 59, 75

Uncertainty Change an intrinsic reward designed to reflect the degree to which the prediction learner is settling on a stable prediction, in this work measured as how much the Variance of Prediction estimate has changed since the preceding time step, defined by:

$$|\hat{\nu}_{t-1,i} - \hat{\nu}_{t,i}|$$

. . . . 84, 88, 89, 91, 93, 101, 106

Uncertainty Reduction an intrinsic reward designed to encourage the learner to settle on a stable prediction by decreasing the variance in their predictions, in this work defined by:

$$\hat{\nu}_{t-1,i} - \hat{\nu}_{t,i}$$

. . . . 85, 89, 97

Variance of Prediction an intrinsic reward measuring the amount of recent variability in the prediction learner's estimate, in this work defined incrementally by:

$$\hat{\nu}_{t,i} \leftarrow (1-\beta)\hat{\nu}_{t-1,i} + \beta\left(\hat{C}_{t,i} - \overline{\hat{C}_{t-1,i}}^{\beta}\right)\left(\hat{C}_{t,i} - \overline{\hat{C}_{t,i}}^{\beta}\right)$$

. . . . xxv, 84, 85, 88, 91, 93, 97, 101

Weight Change an intrinsic reward that measures the amount of change in the prediction learner's weights, defined by

$$\|w_{t,i} - w_{t-1,i}\|_1$$

. . . . xi, 81, 82, 85, 88, 89, 97, 99, 101, 104–107, 110, 111

Acronyms

AI artificial intelligence . . . . xix, 3, 5, 6, 118, 209–211, 216, 244, 245

ANN artificial neural network xxiii

ANOVA analysis of variance 253, 256

CCI inscribed central composite 253

CPM Competence Progress Motivation 50

DTS Dynamic Thompson Sampling 76, 93

EWMA Exponentially Weighted Moving Average 61, 62

GVF general value function . . . xv, xx, xxiii, 24, 31, 37, 38, 54, 67, 68, 115, 242, 243

HRA Hybrid Reward Architecture 67

IAC Intelligent Adaptive Curiosity . . . . xxii, 34–37, 51, 60, 149

ICM Intrinsic Curiosity Module 149

IDBD Incremental Delta-Bar-Delta 82

IGM Information Gain Motivation 49, 243

IM intrinsic motivation . . . xxii, 8, 9, 11, 14, 16, 18, 25–28, 32–34, 39, 40, 49, 50, 52, 85, 112, 114, 210, *Glossary:* intrinsic motivation (IM)

IMAGINE Intrinsic Motivations And Goal INvention for Exploration 157

IMRL intrinsically-motivated reinforcement learning . . . . . 8, 9, 25, 26, 112, 211, 212, 243, *Glossary:* intrinsic motivation (IM)

IR intrinsic reward … xi, xx, xxi, xxiv, xxv, 11, 17, 26, 27, 31, 33, 39, 40, 48–52, 55, 57, 58, 60–62, 69–77, 79, 81, 83, 85, 86, 88–112, 145–147, 149–152, 156, 159, 210, 211, 247, 250, 251, 258, *Glossary:* intrinsic reward (IR)

IUA Intentional Unintentional Agent … 67

KL Kullback–Leibler … 45, 86, 90

LMS least-mean-squares … xx, 73, 80, 81, 93, 100, 102–104, 110, 112

LPM Learning Progress Motivation … 49, *Glossary:* Learning Progress

MDP Markov decision process … 18, 19, 31

MSE Mean Squared Error … xvii, 73–75, 77, 94, 95, 97, 100–102, 105, 106, 111

PACE Prediction, Appraisal, Curiosity, and Exploration … 120

RL reinforcement learning … xxiii, 4, 8, 14, 18, 24–26, 31, 47, 48, 56, 59, 60, 82, 111, 112, 141, 159, 210, 211, 246–248

RSM response surface method … 253

SAC-X Scheduled Auxiliary Control … 67, 68

TD temporal-difference … xi, xvi, xxiii, 21, 22, 37–39, 57, 58, 60–62, 64, 86, 110, 242, 243, 249, 250

TIDBD Temporal-Difference Incremental Delta-Bar-Delta … 242

UCB upper confidence bound … 29

UDE Unexpected Demon Error … xi, 37, 60, 61, 66, 84, 85, 92, 93, 97, 101, 105, 106, 108

UNREAL UNsupervised REinforcement and Auxiliary Learning … 67, 68, 109

Chapter 1

Introduction

1.1 Curiosity in a Computer[1]

From reducing human wait-time on our computers to controlling prosthetic limbs, intelligent systems are rapidly improving to the point that they can significantly improve our quality of life. Historically, human designers have designated the exact procedures followed by their systems, but the designer cannot always be expected to determine the best way for the system to operate in every environment that it could encounter. In truth, the system itself is in the ideal position to determine how to act. If we have a system deciding its own actions, we want it to make its decisions based on sufficient and appropriate—perhaps even thorough—knowledge of the environment. The term *curiosity* refers to a desire to learn or to know more. We value curious behaviour because it leads to becoming more knowledgeable. To empower learning systems to adaptively and effectively become more knowledgeable, we could aim to implement *computational* curiosity in intelligent systems. While computing scientists hope to offer the benefits of curiosity machine intelligence, we do not yet understand the mechanisms behind human curiosity (Kidd and Hayden, 2015, p. 449)—much like many other aspects of human intelligence. As a result, the idea of designing machine curiosity is an interesting and open

[1] Some of this text has been adapted from Ady et al. (2022a, p. 2) and is under review with the *Journal of Artificial Intelligence Research*.

challenge.

Computational curiosity, or machine curiosity—the two terms will be used interchangeably throughout this dissertation—has been used to refer to mechanisms to give computational systems a desire to learn or know more. However, computational curiosity can also be thought of as building computational models of the abstract concept we call curiosity. One way we might better understand curiosity is to better understand the ways we are inclined to model it.

Humans have thought about their own curiosity for thousands of years, dating back at least to Aristotle in 350 BCE (Loewenstein, 1994, p. 76, in reference to *Metaphysics*, Bk. 1, Ch. 2). The study of human curiosity remains an active area of research with many diverse interpretations; recent psychological, neuroscientific and philosophical accounts by Kidd and Hayden (2015) and Zurn (2015, pp. 1–28) review some of this diversity of thought. Over the last three decades, curiosity has also caught the attention of researchers seeking to create increasingly intelligent non-biological learners. Select ideas from the study of human curiosity have inspired fantastic breakthroughs in machine intelligence, from a robot dog shifting its own learning focus across progressively complex situations (Oudeyer et al., 2007) to a simulated player achieving higher-than-ever-before scores in Montezuma's Revenge, one of the so-called "hard exploration" games in the Arcade Learning Environment suite (Bellemare et al., 2016, pp. 4, 7; Dvorsky, 2016; Burda et al., 2019b, pp. 1, 4; Cobbe et al., 2018). Work on machine curiosity is expected to continue to play an influential role in machine intelligence research.

This thesis is about machine learning algorithms designed with the purpose of exhibiting properties of curiosity. There are two main lines of reasoning for why we, as a scholarly community, might want to undertake such an endeavour.

The first is because we expect curiosity will be valuable for machines, just as it is thought to be valuable for humans. There have been calls to foster curiosity in far-reaching domains such as education (Engel, 2011, pp. 625, 628, 643; Schmitt and Lahroodi, 2008, p. 125) and business (Gino, 2018, p. 48). Of course, the valuing of curiosity is not uncomplicated, nor is it universal. There exists a long history of curiosity being alternately—even sometimes simultaneously—praised and blamed (Benedict, 2001, p. 23), seen as vice or virtue (Loewenstein, 1994, p. 76). The function of curiosity is popularly hypothesized to relate to motivation and facilitation of learning (Kidd and Hayden, 2015, p. 450). Moreover, the benefits in humans seem to be even broader, as Gino's (2018) research suggests that curiosity makes for better team members: it improves decision-making—reducing susceptibility to confirmation bias—and also increases innovation, engagement, collaboration, and ability to adapt while reducing group conflict (pp. 48–49). With the growing conversation around incorporating artificial intelligence (AI) tools as team members in the workforce (Chen et al., 2022, p. 2, Duin and Pedersen, 2021) or companions in homes (Ramadan et al., 2021), the design of AI that reflects these beneficial aspects of curiosity is incredibly relevant and timely.

The second reason we might want to design machine learning algorithms reflecting curiosity is to improve our understanding of curiosity as a whole. There is hope that we can do so through the process of developing machine models. There is a natural relationship between machine curiosity and the curiosity-focused fields in other disciplines, as all of these fields share an interest in explaining different aspects of the same phenomenon (Darden and Maull, 1977, pp. 48–49). A variety of views about the strength of the relationship (if any) between artificial intelligence and psychology have been put forward (with some gathered by Newell, 1970), but one view is that some of the work generated in the field of artificial intelligence is "simply part of theoretical psychology" (Moore and Newell, 1974, p. 1), especially when its designers seek to remain close to their concepts of inspiration within psychology. Indeed, the process of developing models or simulations of curiosity can highlight many questions about biological curiosity that have been, thus far, left unanswered (Hunt, 1971, p. 93; Lieto and Radicioni, 2016, pp. 1, 2), and should motivate new experiments and lines of research in other fields that have the appropriate tools and techniques to answer those questions (Darden and Maull, 1977,

p. 50).

With the value expected from computational curiosity, both in intelligent systems and in understanding our own curiosity, it is unsurprising that our community has proposed a variety of methods for implementing computational curiosity. It is natural that many methods for machine curiosity employ existing frameworks for *motivating* systems, because curiosity is often conceptualized as motivation to learn (e.g., Szumowska and Kruglanski, 2020, p. 36). In particular, reinforcement learning (RL), an application of learning by trial and error to predict and control, is commonly employed for computational curiosity. RL has a well-developed literature (Sutton and Barto, 2018; Dayan and Niv, 2008; Kaelbling et al., 1996) and is very effective for implementing machine behaviour to maximize an objective value, making it an ideal choice for "motivating" a system (Barto, 2013, pp. 17–18, 19). Readers familiar with RL might question the This dissertation focuses on RL methods for producing computational curiosity, both because of the prevalence of such methods thus far and for deeper reasons explored later, in Section 2.3.

1.2 What do we mean by *curiosity*?

For the purpose of this thesis, I am embracing curiosity in its vernacular form, as a word that is used day-to-day in communication between people of varied backgrounds. Now, this choice is not without controversy. Fiske (2020) has warned that "it is pernicious to use one language's dictionary as the source of psychological constructs" (p. 95). Fiske tells the story of how his interest was piqued by experiences of shedding tears during movie scenes depicting human kindness, despite feeling happy (p. 95). At the time, he and colleagues felt such emotions were best described as *being moved* (p. 95). He and his multilingual collaborators, however, soon realized that the terms that most commonly denoted the emotion in different languages failed to "map onto each other one-to-one" (p. 96). Starting from this experience, he came to argue that researchers "need to coin new technical names for scientifically derived [psychological] constructs" (p. 95) and his team came to coin the technical term *kama muta*, the emotion evoked by sudden intensification of communal sharing (p. 96). Fiske's arguments apply equally to the word *curiosity*. Such concerns are recognized by curiosity researchers like, Kidd and Hayden (2015) who similarly note that, in psychology and neuroscience, the lack "of a widely agreed upon delineation of what is and what is not curiosity" limits our understanding (p. 449).

Yet, this lack of agreement need not be a limitation within the realm of AI. In fact, it is in some ways a strength. If our field is inspired by a wide range of different ideas about curiosity, we could achieve a great mosaic of useful mechanisms helping us achieve a variety of our potential goals in building intelligent systems. Murayama et al. (2019) have made a similar point that, despite the likely impossibility of a "correct" definition of curiosity, its study can still lead to "significant theoretical and practical implications" (p. 877).

Indeed, that curiosity researchers across disciplines are "beguiled by our language" (Fiske, 2020, p. 98) may be an energizing factor behind our research. The growing research interest in curiosity may rely on curiosity as an "everyday concept" (Schmitt and Lahroodi, 2008, p. 126), one we experience in our life and bring into our communication with others. The hypothesis that curiosity plays a critical role in scientific discovery is widespread (Inan, 2012, p. 3; Djerassi, 2011; Zuss, 2011, pp. 64–65). We should take this hypothesis in conjunction with Inan's (2012) contentions that, not only are we curious exclusively "about things that we are interested to know," but "the limits of what we can be curious about are set by the limits of what we can attempt to refer to within our idiolect" (p. 183). Together, these ideas suggest that it is valuable for researchers who study curiosity-related concepts to stay connected to the everyday concept of curiosity. After all, it is the vernacular concept that sparked our initial curiosity, not any technical term coined for the purpose of precision. Getting rid of the word curosity is not the way to achieve precision. Indeed, even Kidd and Hayden (2015), despite their characterization that lacking “a single widely accepted definition” has “hindered” development, consider the present “diversity of definitions” to be healthy (p. 449).

However, we still need to work towards being precise in communication while embracing curiosity as inspiration. Without this precision, we are likely to struggle to effectively achieve *generativity*. The term generativity, as attributed to Shulman (1999), refers to the ability to build on knowledge that has come before us. Careful language around what aspects of curiosity we are capturing faithfully (or not) will allow future authors to appropriately connect their ideas to our own. We must achieve a balance between diversity and precision, or else even AI, which does not always need to be perfectly locked to its inspirational psychological construct, will suffer negative consequences in terms of “integration of studies ... and communication of scientific ideas” (p. 97), two core areas of concern for Fiske (2020). So let us make our connections to the everyday concept that inspires our curiosity, but be precise enough that we will be able to weave the threads of our work together with the knowledge that has come before to prepare for new threads of the future.

1.3 Research Aims

The overarching aim of the project described in this document was to bring us closer to providing the benefits of human-like curiosity to machine learning systems. When I say ‘the benefits of human-like curiosity,’ I think of the reasons I delight in children’s many questions—given I am not among those stuck answering them all day—and uphold the importance of curiosity-driven science. I believe that curiosity in humans has value. I also believe that some learning systems could benefit from aspects of curiosity. The core argument I make in this document is that **we must experiment and think beyond the most commonly used frameworks being used for machine curiosity if we want to secure the benefits of human-like curiosity for our machine learners.**

Unifying the View I began this line of research in 2016. At the time, there was already a burgeoning literature of novel methods for machine learning systems inspired by curiosity. Some of this historical background, including summaries of influential ideas and publications, will be covered in Chapter 2. As these new methods emerged, so too did new testing domains to help capture the most valuable and noteworthy aspects of their behaviour. Indeed, some of the work in this dissertation follows that tradition (as we will see in Chapter 5). In this way, a landscape of computational curiosity methods was beginning to emerge, with each new method showing promise in producing some desirable behaviour in machine learners. Each method by itself formed a strong starting point for further research, but to understand how the landscape was developing, I felt that it was necessary to acknowledge the structure of the field and how existing methods might be compared.

To see the benefits we believe curiosity promises, we must design methods that adopt advances in understanding from other disciplines; we must build upon a unified understanding of existing methods. Through the design of and experimentation with understandable domains, I aimed to contribute some of the relationships among methods that are needed to unify our understanding of existing curiosity methods. Chapter 3 describes my contributions towards this aim.

My Novel Approach for Machine Curiosity In the process of designing comparative experiments, I sought the roots of existing methods—the ideas that originally inspired their designs. These ideas flowed from disciplines that study humans, like psychology and behavioural economics. Indeed, I found that the human-centred disciplines have carefully built up theories of curiosity. While aspects of these theories inspired many machine curiosity methods, other aspects of the theories seemed to have been neglected.

The process of working to unify our view of existing machine curiosity methods gave me the opportunity to see these methods in relation to one another. This particular perspective led me to a realization: Despite critical differences in behaviour and primary task performance (Chapter 3), one of the core frameworks for machine curiosity methods makes the methods in this cohort more similar to than different from one another. This core framework is called intrinsically-motivated reinforcement learning (IMRL), which will be described in more detail in Section 2.3. In particular, I believe the structure of IMRL limits our ability to realize some of the core beneficial properties of human curiosity in our computational systems—properties that I was discovering through my multidisciplinary survey of curiosity.

This viewpoint motivated the second core research aim: to develop an argument for some of the key properties of human curiosity that could most benefit machine learning systems. I aimed to integrate ideas from multiple disciplines that include the study of curiosity, as the complexity of curiosity invites multidisciplinary methodologies (Repko and Szostak, 2020, p. 8). Chapters 4 and 5 describe my contributions towards this aim.

The Present and the Future of Machine Curiosity In the intervening years since I began this work, how has the field changed? There has been a surge of interest within the core reinforcement learning community for using and developing forms of computational intrinsic motivation (IM), with one likely cause being the role of curiosity-inspired approaches in the achievement of impressive, and in some cases, state-of-the-art performance on challenge problems recognized by the wider reinforcement learning community, like the achievements of Bellemare et al. (2016) and Pathak et al. (2017). IMRL has shown itself to be a useful framework for increasing the exploratory power of existing reinforcement learning methods.

Concurrently, interest has risen in understanding the complex landscape of different approaches to computational IM, e.g., Biehl et al. (2018, p. 1) and Matusch et al. (2020, pp. 1, 2). There is huge potential for future work to be done in this area, as we would benefit from a diversity of comparative views of different related methods to match the diversity of ways humans understand curiosity.

While the research field studying computational IM and machine curiosity has grown, computational intrinsic reward remains the most prevalent approach for developing curiosity-inspired algorithms. My contribution of five key properties of specific curiosity offers a distinct direction for imagining new forms of machine curiosity. This distinct direction is not based in the IMRL framework—and yet, in the original meaning of *intrinsic motivation* in psychology, specific curiosity as it is understood through the lens of these properties *is* a kind of intrinsic motivation.

At the time I am writing this work, there is not an strongly established community[2] for computational curiosity or machine curiosity—in keeping, a favoured name for this kind of research has yet to be solidified. However, a growing interdisciplinary community around curiosity welcomes viewpoints from the computer scientists who draw inspiration from curiosity in their systems and their science.

1.4 Contributions[3]

At a high level, this dissertation describes two important contributions.

The first key contribution is a set of experimental comparisons of multiple intrinsic-reward algorithms—the first of its kind. When new machine curiosity or computational intrinsic motivation algorithms are introduced, they are rarely compared with other existing curiosity algorithms. By manipulating the curiosity mechanism while controlling for the learning algorithm and its environment, I was able to observe and describe key differences between machine curiosity algorithms

[2] Computational creativity provides an example of a cohesive community that has grown around a concept from psychology being ported into artificial intelligence (cf. Guckelsberger, 2020, p. 13). There may be a future in which machine curiosity develops its own cohesive community in much the same way. For now, the study of machine curiosity appears to function as an emerging interdiscipline (Repko and Szostak, 2020, p. 6) crossing back and forth between computer science—particularly machine learning and artificial intelligence—and curiosity studies, which might itself be considered an emerging interdiscipline.

[3] Some of this text has been adapted from Ady et al. (2022a, p. 2) and is under review with the *Journal of Artificial Intelligence Research*.

that are built upon the idea of an intrinsic reward (Ady and Pilarski, 2016, 2017a, cf. Ady, 2017c). What followed from this work was the first comprehensive empirical comparison of different intrinsic reward mechanisms (Linke et al., 2020). While the landscape of different computational intrinsic motivation methods still requires substantial investigation to develop a coherent understanding of its hills and valleys, this work provided a novel experimental framework. It remains one of the only published experiments drawing on early psychological theories of curiosity to better understand existing machine curiosity algorithms. This first key contribution an be summarized as follows:

- A new family of experimental domains, Curiosity Bandits, which allow comparison across different approaches to computational curiosity or computational intrinsic motivation.
- A comprehensive empirical comparison of different intrinsic reward mechanisms that, for the first time, puts them in context with each other.

These detailed studies highlighted serious limitations of the primary class of 'machine curiosity' methods today: In particular, these methods lack critical properties of human curiosity as it has been studied in behavioural economics, psychology, philosophy, and neuroscience. This work, therefore, presents a landmark synthesis and translation of specific curiosity to the domain of machine learning and reinforcement learning and provides a novel view into how specific curiosity operates and in the future might be integrated into the behaviour of goal-seeking, decision-making computational agents in complex environments. The second key contribution of this thesis is an argument for which specific properties of curiosity will most benefit machines (Ady et al., 2022a). Beyond translating important recent contributions from the study of human curiosity to machine intelligence, this work clarifies and consolidates a unique theoretical understanding of human curiosity. In particular, it represents, to the best of my knowledge, the most robust and detailed characterization of specific machine curiosity to date. This second key contribution can be summarized as follows:

- The definition of five key properties of specific curiosity: 1) directedness towards inostensible referents, 2) cessation when satisfied, 3) voluntary exposure, 4) transience, and 5) coherent long-term learning.
- A proof-of-concept reinforcement learning agent, demonstrating how the properties manifest in the behaviour of this agent in a simple non-episodic grid-world environment that includes curiosity-inducing locations and induced targets of curiosity.

Readers of this work may also be interested in some of the additional research that I completed alongside the contributions included in this dissertation, as it informed the ideas and approaches I used with the work described herein, adjacent to the main storyline of this work. These contributions can be found in Appendix A.

1.5 Structure of the Thesis

This dissertation comprises six chapters. This chapter has provided an overview of the project described in this thesis. Chapter 2 both introduces some of the history of the study of machine curiosity and serves as a reference that explains some of the core concepts relied upon in following chapters. Chapter 3 describes the development of a novel family of experimental domains designed specifically to compare computationally intrinsically motivated learners—Curiosity Bandits—and the first comprehensive empirical comparison of different intrinsic rewards, which was designed using members of that family. By illustrating patterns in the behaviour across different approaches to computational IM, the studies detailed in Chapter 3 highlight key limitations of identifying curiosity with intrinsic motivation. In particular, these computational intrinsic reward methods fail to model critical properties of human curiosity. Chapter 4 details five key properties of specific curiosity and includes an argument for their value in the design of future computational learners. To demonstrate the feasibility of these properties in a computational system, Chapter 5 presents an initial proof-of-concept algorithm that achieves three of the five key properties presented in the preceding chapter. Further, Chapter 5 includes an ablation study demonstrating that the three properties work together to result in behaviour characteristic of specific curiosity. I conclude the story in Chapter 6, summarizing my research contributions and presenting ideas for potential future work.

The format of this dissertation largely conforms to a paper-based structure, in which the central chapters of the thesis (here, Chapters 3–5) are independent papers in publication format. These central chapters are bookended by introductory and background material on the front (Chapters 1 and 2) and a conclusion on the back (Chapter 6). I have made certain adjustments to the original texts. Specifically, Chapter 3 collects together Ady and Pilarski (2016), Ady and Pilarski (2017a), Ady and Pilarski (2017b), and Linke et al. (2020), as these four publications provide complementary perspectives on the proposal and development of Curiosity Bandits. The original text of these publications has been modified to improve clarity and eliminate redundancies. By contrast, Chapters 4 and 5 are presented here as separate chapters, despite their original presentation as a single document (Ady et al., 2022a). These editorial decisions have been made carefully to improve the reading experience.

1.6 Researcher Identity Statement

I am a curiosity researcher. I suspect curiosity is universally shared across the human species. However, research in affect science suggests that every emotional facet of human psychology is to a large extent actively constructed by the mind—neurally, psychologically, and very importantly, socially. Curiosity is no exception. Your experiences of curiosity are shaped by your past, your relationships, and your culture.

My experiences of curiosity are similarly shaped by my culture. I completed this research where I grew up: Edmonton, on Treaty 6 territory and in Métis Region 4. I live and perform my research in a context that was designed through violence and the systematic erasure of Indigenous ways of knowing. My research has been shaped by this violence, as the stories of curiosity that are most salient and easily accessed continue to be those that privilege Western ways of knowing.

As I work to develop a precise understanding of curiosity, I am challenged to uncover how the culture I live and work in has hidden away non-Western viewpoints on curiosity, skewing the systems that I design. In machine learning, we must continually confront the biases introduced by history. It rings true to me that while I have worked to encompass a diverse set of ideas, I have not escaped the culture I live in and it also rings true to me that the systems I am designing are likely raced, gendered, classed, and otherwise biased.

As you read this document, I invite you to reflect on how my position has influenced this research and how your own endeavours are shaped by the histories of the land on which you pursue them.

Chapter 2

Background

This chapter will be devoted to presenting the relevant background regarding the study of curiosity and intrinsic motivation in psychology and the use of reinforcement learning techniques as approaches to both computational curiosity and computational intrinsic motivation. First, we will briefly overview how the study of curiosity first developed in the field of psychology, as this development largely preceded and strongly influenced researchers in computational curiosity. This section on the field of psychology will be followed by an outline of the framework underlying the approaches to computational curiosity on which I have chosen to focus: reinforcement learning. Once the necessary vocabulary is in place, we will look at an important superclass of curiosity: intrinsic motivation.

2.1 Origins of Curiosity in Psychology

Both intuition about curiosity and curiosity research done in psychology have influenced the way computational curiosity has been approached. These influences will be explored further in Section 2.3, in which we will attend to the details of several examples of machine curiosity approaches. Wu and Miao, in their 2013 survey on curiosity, provide an eloquent explanation of the benefit of the background in psychology on any exploration of computational curiosity:

> In order to provide a more complete model of human cognition . . . we believe that it is beneficial to go back to the research in psychology to understand how human curiosity can be aroused. (Wu and Miao, 2013, p. 18:2)

A historical view of curiosity shows that the initial recorded discussions on the topic were largely focused on whether or not the concept should be upheld as a virtue (Loewenstein, 1994, p. 76). While the ancient Greeks championed the quality of curiosity, the concept fell into disfavour in the Middle Ages following St. Augustine's criticism of curiosity as a weakness of vanity and whimsy (Reio et al., 2006, p. 119). The recognition of the role of curiosity in Galileo's astronomical discoveries brought the notion back into a positive light in the seventeenth century (Loewenstein, 1994, p. 76).

George Loewenstein, in his 1994 review of the psychology of curiosity, explains that a first wave of study in psychology of curiosity grew through the 1950s into the 1960s (p. 75–76). He observed that these studies were predominantly concerned with: first, the cause of curiosity; second, why humans seek both situations which result in curiosity and to resolve those situations (reducing our curiosity in them); and finally, what conditions can be observed in the environment before and after one experiences curiosity. Loewenstein (1994) further offers that a second wave of research ran through the later half of the 1970s into the 1980s; this time, the focus was on measuring curiosity. Bridging both waves of research was the work of Daniel E. Berlyne, who would eventually be an important influence on the researchers who would later suggest methods for machine curiosity.

2.1.1 Berlyne & Csikszentmihalyi

Daniel E. Berlyne first published on on curiosity-related motivation in humans in the late 1940s (Konečni, 1978). He developed his ideas and directed the field of psychological research on curiosity until his death in 1976 (Appley, 1978). According to a biography by Konečni (1978, p. 135), Berlyne's arguably most influential publication was *Conflict, Arousal, and Curiosity* (1960). Berlyne's influence has extended to work in curiosity and intrinsic motivation in machine intelligence, where his work is widely referenced, including in a number of the papers we will discuss throughout this chapter (e.g., Barto, 2013; Schmidhuber, 2010; Oudeyer et al., 2007).

In the late 1950s, relatively early in his career, Berlyne rekindled an idea which had been introduced several decades prior: the Wundt curve (Kubovy, 1999, p. 139; Berlyne, 1960, pp. 200–201 in reference to Wundt, 1874). An image of the Wundt curve is recreated in Figure 2.1. This idea would prove to have an important role in the future treatment of computational curiosity. Berlyne (1960) hypothesized that "human beings and higher animals will normally strive to maintain an intermediate amount of arousal potential" and that this goal is an important part of the mechanism behind curiosity (p. 200).

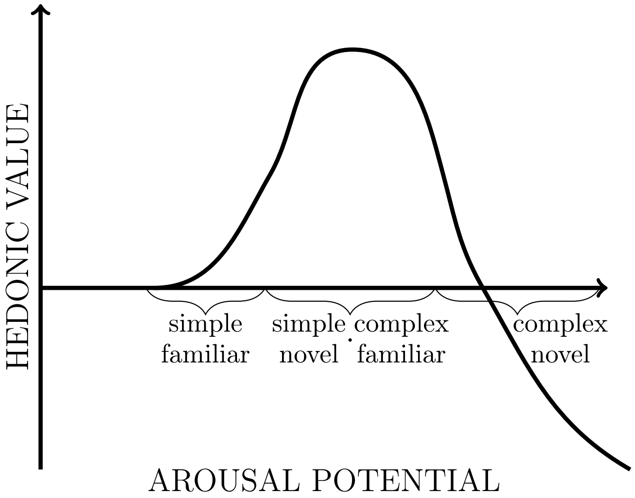


Figure 2.1: Wundt curve recreated following Berlyne (1970, p. 284).

According to Oudeyer et al. (2007), another psychologist, Mihalyi Csikszentmihalyi, similarly theorized that "the internal reward is maximal when the challenge is not too easy but also not too difficult" (p. 266). Csikszentmihalyi would later prove to be influential in the development of computational curiosity. In particular, Csikszentmihalyi (1991) developed the highly popularized concept of *flow*, which denoted the state realized by humans when their experience is most enjoyable—so called because several of the subjects of his study felt that "it was like being carried away by a current, everything moving smoothly without effort" (1993, p. xiii). Csikszentmihalyi (1993) suggested that achieving flow is intrinsically rewarding for humans (p. xiii). He further theorized about the requirements for flow to occur: the key requirement is that the experience is highly challenging to the point where "personal skills are used to the utmost" (p. xiii). Most often, this occurs when the subject has few distractions and has goals and feedback which are clear and well-defined (Csikszentmihalyi, 1993, p. xiv). Because the balance required to achieve flow shifts as personal skills develop, and personal skills necessarily develop when flow is achieved, flow is generally achieved in previously unexperienced states. An agent optimally seeking flow must be exploratory.

In developing his theory, Csikszentmihalyi (1993, p. 190) expressed a belief that the intrinsic reward of flow developed through evolution:

> Apparently humans who experience a positive state of consciousness when they use their skills to the utmost in meeting an environmental challenge improve their chances of survival. The connection between flow and enjoyment may have been at first a fortunate genetic accident, but once it occurred, it made those who experienced it much more likely to be curious, to explore, to take on new tasks and develop new skills. And this creative approach, motivated by the enjoyment of facing challenges, might have conferred so many advantages that with time it spread to the majority of the human population.

In particular, Csikszentmihalyi (1993) suggested that the enjoyment of flow in humans is a component of an overall evolutionary preference for *complexity* (p. 175). Csikszentmihalyi (1993) designated the complexity of a system as its level of differentiation and integration, which in turn refer to "the degree to which [the] system ... is composed of parts that differ in structure or function from one another" and "the extent to which the different parts communicate and enhance one another's goals" (p. 156). Flow motivates the development of more complex skills over our lifetimes, just as we might desire from curiosity.

Berlyne also associated curiosity with *complexity*, but by a different definition. In his case, he eventually began to pair the arousal potential of stimuli with information-theoretic measures of their complexity (Kubovy, 1999, p. 140). At the time, information theory was a relatively recent development by Claude Shannon (1948). Interestingly, Berlyne's emphasis on intermediate levels of arousal and on the information-theoretic measurement of stimuli have set two different pathways of research for computational curiosity. We will explore each of these paths in Sections 2.3.3 and 2.3.4, respectively. However, I want to first touch on the concept of intrinsic motivation, because it provides context into its subconcept, curiosity. For adequate detail, we must first develop the foundational vocabulary for reinforcement learning.

2.2 Reinforcement Learning Framework

Reinforcement learning (RL) is an approach to learning by which machines and animals learn about their world through trial and error, changing their expectations about the world to match their experiences. Machine intelligence researchers recognize RL as a very effective means for implementing a "motivation" for a machine to maximize an objective value and so have posited that RL might be the ideal choice for implementation of an agent motivated by a form of machine curiosity (Barto, 2013, pp. 20, 40). Because I too am interested in exploring the potential of RL for machine curiosity, this document is centred around RL approaches to machine curiosity, yet informed by knowledge of curiosity from other disciplines. This section provides the notation and preliminary ideas from the RL framework that will be used throughout this document.

A standard formal model used in RL, depicted in Figure 2.2, is that interaction with the world can be modelled as a Markov decision process (MDP) (Sutton and

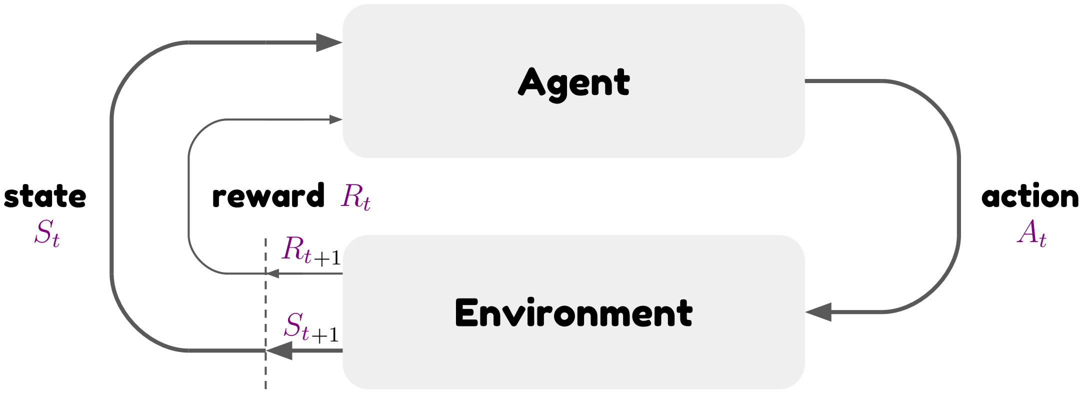


Figure 2.2: Interaction between the agent and its environment, formalized as a decision process. Adapted from Sutton and Barto (2018, p. 48).

Barto, 2018, p. 2). Within this formalization, we think of time occurring in discrete *time steps*. At each time step t, the agent is in a current *state* $S_t \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible states. Intuitively, the state describes the circumstances that the agent is in and the context that the agent can use to make decisions. The agent must select an *action* $A_t \in \mathcal{A}(S_t)$, where $\mathcal{A}(S_t)$ is the set of actions available to the agent from state S_t. The agent receives a real-valued *reward* $R_{t+1} \in \mathbb{R}$ and enters a new state $S_{t+1} \in \mathcal{S}$. Many theoretical results in reinforcement learning rely on the *Markov property*, which holds when the reward and new state depend on S_t and A_t, but S_t and A_t provide no less information about the future than do all preceding states and actions (Sutton and Barto, 2018, p. 49). Formally, this notion can be written:

$$\begin{aligned} &\Pr\{S_{t+1} = s, R_{t+1} = r \mid S_t = s_t, A_t = a_t\} \\ =&\Pr\{S_{t+1} = s, R_{t+1} = r \mid S_0 = s_0, A_0 = a_0, ..., S_t = s_t, A_t = a_t\} \end{aligned} \tag{2.1}$$

The Markov property, which applies to MDPs, is typically assumed as part of the formal framework for reinforcement learning (Sutton and Barto, 2020, p. 13), particularly because it helps make developing theory more tractable (Sutton and Barto, 1998, Ch. 3.5). However, the Markov property is not always a realistic assumption. Particularly when working with robots and other complex systems, it can be difficult to engineer a Markov state from the sensor and motor configurations that we can obtain from the environment. Without assuming the Markov property, we can still describe environments using the same interactive framework depicted in Figure 2.2, herein referred to as simply *decision processes*, recognizing that much of the theory developed for reinforcement learning cannot be assumed to hold.

The problem defined by a decision process is: How should the agent decide which action to take in each state to accumulate the most reward over time? The goal is to determine how an agent should behave, which we define as a *policy*, π, which maps each state to the probabilities of taking each action available from that state. However, we need to qualify what it means to accumulate the most reward over time. Should receiving \$100 in a thousand years be the same as receiving \$100 today? Sometimes we want our agents to value different situations differently. There are multiple ways of formulating the objective function, that is, the way our agents value the accumulation of reward. Two classical settings include the episodic setting and the discounted setting.

In the *episodic setting*, the *return* at time t is defined as

$$G_t := \sum_{k=1}^{T-t} R_{t+k} \tag{2.2}$$

where T is a final time step. The episodic setting is commonly used in games, where there is a clear, repeating end point: you must get the most points before the end of the game. As many machine curiosity approaches were designed in the context of developmental robotics, it is common for there to be no clear end point (no obvious final time step T), and so a continuing setting is typically considered more appropriate, and the discounted setting is one option. In the *discounted setting*, we include a *discount rate*, $\gamma \in [0, 1]$, to weight rewards in the near future more than those in the far future, and define the return at time t as

$$G_t := \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.3}$$

The practical effect of the discount rate is that, for $\gamma < 1$, we can be safely assured that G_t is finite, despite being the sum of an infinite number of terms (Sutton and Barto, 2018, p. 55). In terms of modeling biological learners, the formulation of return in the discounted setting (Equation 2.3) roughly reflects the traditional—though flawed[1]—'exponential discounting' model of intertemporal choice from behavioural economics—our tendency to care more about near-term gains and losses over those further into the future (Berns et al., 2007, pp. 482–483).

In either setting, our objective, originally stated informally as "how to accumulate the most reward over time" becomes "how to maximize return, G_t." A reinforcement learning agent, towards that goal, typically maintains a 'learned' value function $\hat{v}_\pi : \mathcal{S} \to \mathbb{R}$ estimating the true value function $v_\pi : \mathcal{S} \to \mathbb{R}$, defined by

$$v_\pi(s) \coloneqq \mathbb{E}_\pi \left[G_t \mid S_t = s \right], \tag{2.4}$$

where $\mathbb{E}_\pi[\cdot]$ denotes the expected values of a random variable, given the agent follows policy π.

One of the most standard ways of learning a value function is *temporal-difference (TD) learning*, which has also been found to well-model some animal learning (Niv, 2009). At each time step in TD learning, the learner makes a new estimate for the estimated value $\hat{v}_\pi(S_t)$ that ropes in its newest sample of reward, R_{t+1} and uses its estimate of the future return $\hat{v}_\pi(S_{t+1})$, given its new state. Simply using that new estimate, $R_{t+1} + \hat{v}_\pi(S_{t+1})$, would not take into account any prior information held in the original estimate $\hat{v}_\pi(S_t)$, so the agent typically only changes the value maintained in memory by a "step" towards the new estimate. The difference between the new and old estimates is called the TD error, defined as

$$\delta_t \coloneqq R_{t+1} + \gamma \hat{v}_\pi(S_{t+1}) - \hat{v}_\pi(S_t) \tag{2.5}$$

in the episodic and discounted settings.

[1] Behaviour of animals, including humans, is better modelled by a hyperbolic discounting curve than an exponential discounting curve (Berns et al., 2007, p. 483). However, recent work like that of Kurth-Nelson and Redish (2009, p. 1) has suggested that the hyperbolic discounting curve also reflects the behaviour of an agent making decisions using a combination of multiple value functions with different exponential discount rates.

Instead of (or in addition to) maintaining estimated values for each state, a learner could maintain estimates of *action-value* functions, which are closely related to standard value functions. Value functions average over the potential actions the learner could take from a state, while the outputs of action-value functions depend on which action the learner is takes from a given state. Analogous to Equation 2.4, an action-value function q_π is defined by

$$q_\pi(s, a) \coloneqq \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a\right], \tag{2.6}$$

Similar to a value function, an estimated action-value function, Q, can be learned by variations of TD learning. One option is Sarsa, for which the TD error is defined by

$$\delta_t \coloneqq R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t). \tag{2.7}$$

Another is Q-learning, for which the TD error is defined by

$$\delta_t \coloneqq R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t). \tag{2.8}$$

While Sarsa learns action-values for the policy the learner is following (*i.e.*, it is *on-policy*), Q-learning approximates the optimal action-value function (Sutton and Barto, 2018, p. 131).

When we do not have the Markov property, it can be useful to work with a model where we can obtain an *observation*, $O(S_t)$, relating the information available at time t about the state S_t. We can then use some kind of function approximation for our estimated value function $\hat{v}_\pi$. The formal model of decision processes, whether Markov or not, allows for the design of clear experiments which are easy for a human to observe and understand—at least in comparison to experiments involving the complexities of numerous sensors or rich visual observations. In Chapter 3, I will provide further discussion of the versatility of this set-up in the context of improving our understanding of behaviour produced through computational curiosity methods.

2.2.1 Exploration

When aiming to maximize return, there is always a difficult challenge: determining how to choose between an "exploitative" action with the best expected payoff and an "exploratory" action which learns more about the environment and might result in finding an action with an even better payoff (Sutton and Barto, 2020, p. 3). This challenge is known as the exploration–exploitation dilemma. Numerous exploration techniques specific to different potential situations (e.g., different value estimates, uncertainties, and numbers of remaining steps) exist (Sutton and Barto, 2020, p. 26), but the specifics of a given situation in reinforcement learning are often so complicated or little-known that reinforcement learning algorithms are implemented without attempting to balance exploration and exploitation in a sophisticated way—they are implemented only with a concern for "balancing them at all" (Sutton and Barto, 2020, p. 27).

Curiosity has been imagined by authors like Still and Precup (2012) as a way to address the exploration–exploitation dilemma (p. 142), and yet many curiosity-inspired methods still need a traditional exploration mechanism alongside their mechanisms proposed as analogues of curiosity (e.g., Oudeyer et al., 2007, pp. 271–272; Pathak et al., 2017, p. 2780, with reference to Mnih et al., 2016, p. 1931; Burda et al., 2019b, p. 13, with reference to Schulman et al., 2017, p. 5). Such traditional exploration techniques usually involve injecting randomness into the policy in some way (called *undirected exploration* techniques by Thrun, 1992). The distinction between curiosity and exploration, at least within reinforcement learning, is crucial for understanding the landscape of curiosity-inspired methods we see today.

2.2.2 Beyond the Maximization of Return

While this section only provides the preliminaries needed for understanding this document, the reinforcement learning framework is very flexible and goes beyond maximization of return, extending to offer many possibilities for prediction and control (Sutton and Barto, 2020, p. 460). However, the maximization of return is a component of the majority of existing approaches to machine curiosity using reinforcement learning. It has been suggested (for example, by Stojanov and Kulakov, 2006) that the prevalent use of the RL paradigm to implement computational curiosity has led to an understanding which is "rather one dimensional and rather reductionist in spirit" (p. 46). However, through a review of the development of RL methods for computational curiosity, I intend to demonstrate that the understanding is already multi-dimensional by exhibiting a sample of such dimensions which can be developed in the RL framework. This level of complexity is by no means fully explored.

One framework by which reinforcement learning demonstrates its flexibility is that of general value functions (GVFs), which generalize the standard value function, v_π, defined in Equation 2.4. Where value functions describe the accumulation of reward over time, a general value function (GVF) describes the accumulation of a chosen scalar signal over time—the construction can be equally well applied to any scalar signal obtained at each time step. GVFs have been used to answer many goal-oriented questions in parallel, as can be found in the 2011 paper by Sutton et al. describing the Horde architecture. Modayil et al. (2014) further provide a detailed account of using GVFs for predictive representations of knowledge. Since their introduction, initial work using GVFs has shown that real-time prediction learning is a practical way to support intuitive joint control in complex robotics systems like those found in prosthetic systems (Pilarski et al., 2013a,b). In connection to the focus of this thesis, we will see in Section 2.3.3.2 a way in which GVFs have been leveraged to develop behaviour inspired by curiosity.

2.3 Reinforcement Learning Approaches to Machine Curiosity

Over the past decade, machine curiosity research has boomed, and many of these research ideas have been developed within the RL framework. RL seems like a strong contender in the pursuit of computational curiosity, particularly since curiosity seems to motivate many human decisions, and RL is sometimes thought of as offering a computational analogue for motivation. Inspired by curiosity, researchers have developed different methods to modify the reward delivered to a learner or to modify other parts of an RL algorithm. Many of their methods have shown promise in real-world or simulated domains, as noted in Section 1.1.

In particular, much of the recent work on machine curiosity has been developed as a branch of research on *intrinsically-motivated reinforcement learning (IMRL)*. It is no surprise that computational curiosity, as a field of research, is intertwined with the study of computational *intrinsic motivation*. The concept of intrinsic motivation was originally studied in psychology, where the term encompasses the "why" behind engagement in activities "for which there is no apparent reward" (Deci, 1975, pp. 3, 23). Curiosity, in humans, is often considered to be a kind of intrinsic motivation (Ryan and Deci, 2000, p. 56). Study of the mechanisms involved in intrinsic motivation began in the 1950s in animal psychology (Deci and Ryan, 1985, p. 13). The term 'intrinsic motivation' tends to be used in opposition to *extrinsic motivation*, which refers to motivation to engage in activities resulting in separable consequences like food or money, where those separable consequences are also known as *extrinsic rewards*. Research in psychology has suggested intrinsic motivation plays an important role in the development of general intelligence (Singh et al., 2010, p. 70).

In some computational work, researchers like Guckelsberger (2020) have attempted close integration of psychological theories of intrinsic motivation, but the term 'computational intrinsic motivation' is often used in a more simplistic way, to reference methods centred on computational intrinsic reward. In his 2013 review, Barto essentialized computational intrinsic reward, writing that "all of the approaches to using RL to obtain analogs of intrinsic motivation ... [become] simply a matter of defining specific mechanisms for generating reward signals" (p. 19, referencing an idea proposed by Schmidhuber, 1991b, p. 226). Returning to the idea of reinforcement learning as an analogue of motivation, we can think of designing a reward function as allowing us to design different motivations for our systems. With a new reward function designed, existing RL algorithms can be used to efficiently learn which actions maximize future reward.

In a sense, many attempts to implement computational curiosity in an intrinsically-motivated reinforcement learning framework motivate us to agree to a measurement of curiosity. The most obvious way to use reinforcement learning to motivate an agent to behave with the highest level of curiosity is to give a value to the curiosity exhibited by the agent by its actions. Of course, what it means for curiosity to be exhibited by an agent is non-obvious and may be difficult to define. However, once a measure of curiosity has been decided by the designer, it can be used as a reward in the reinforcement learning framework, so a curious agent would aim to maximize the accumulation of this reward over time.

In the translation of 'intrinsic motivation' from psychology to a computational reinforcement learning setting, 'intrinsic reward' has morphed to specifically refer to rewards computed from the agent's sensorimotor context without regard to any concrete meaning of the sensorimotor context (Oudeyer and Kaplan, 2007, p. 12). The phrase *sensorimotor context*, stemming from the combination of *sensor* and *motor*, references the combination of what an agent can perceive using its sensors and invoke through motor commands. Use of the phrase usually not only considers those observations at the current time step, but some history of sensor and motor observations. Computational extrinsic rewards, in contrast to intrinsic rewards, tend to be designed to relate closely to the concrete meaning of the sensorimotor context, with rewards for success in a task decided by the human designer. For example, extrinsic rewards might be provided for achievements like winning a game of Go, moving closer to a target, or connecting to a battery charger. Different examples of rewards that, instead, disregard the meaning of the sensorimotor context—that is, intrinsic rewards—will be explored throughout this document. Oudeyer and Kaplan (2007) have described this understanding of intrinsic rewards (IRs) by pointing to how IRs are based on changes in the "knowledge and know-how" of the agent due to changes in the sensorimotor observations made by the agent, independent of the meaning of those observations (p. 12). The phrase "knowledge and know-how" of an agent is vague—likely purposefully so!—but it allows us to consider constructs, computed from observations, that shift up a level of abstraction from the concrete world of the learner, like a reward based on the error in a prediction of that signal could.

The mechanisms that have been defined for curiosity vary widely, making use of a number of different internal constructions, with many utilizing prediction error or ideas from information theory, often in the interests of forming computational analogues of other constructs imagined to related to curiosity, like confidence (Schmidhuber, 1991a), counting (Bellemare et al., 2016), compression progress (Houthooft et al., 2016), learning progress (Oudeyer et al., 2007), surprise (White et al., 2014; Schembri et al., 2007a), novelty response (Singh et al., 2004), and information gain (Schaul et al., 2011; Still and Precup, 2012).

In this section, we will highlight one possible way of classifying the majority of existing approaches, providing detail on a couple of approaches from each class. Though there are many different ways of classifying approaches to computational curiosity,[2] I have chosen an objective function-centric classification because the topic of this chapter (and the next) is the behaviour produced by different objective functions. In particular, I have divided the approaches into those based on error and those based on measures of information, in the information-theoretic sense

[2] Oudeyer and Kaplan (2007) and Aubret et al. (2023), for example, have applied different classification approaches methods to intrinsic motivation, and in many cases, such classification approaches are applicable to forms of computational curiosity.

developed by Shannon (1948).

2.3.1 Early and Influential Ideas for Machine Curiosity

To set the stage, we will begin with three of the most influential works in computational curiosity and intrinsic motivation: Sutton's exploration bonus (1990b) and Schmidhuber's 'curious neural controllers' (1991b) and 'curious model-building control systems' (1991a). In 1991, Schmidhuber proposed a design for a learning system that he termed 'curious' (p. 222–223). His proposal that curiosity might be a useful attribute in learning systems (p. 224) appears to have set in motion the boom in study of computational curiosity that we are experiencing today. His goal was to build a learning system that could rapidly build a model of a signal from the world. At the time, there were few proposals for approaching this problem. In his demonstration of curious model-building control systems, Schmidhuber (1991a) compared his proposed algorithm with two baselines: one was giving a learner a phase of completely random behaviour, while another was called Dyna-Q+ (p. 1458). The design of Dyna-Q+, introduced only a year earlier by Sutton (1990b), included the exploration bonus, the structure of which has proved influential in the use of intrinsic motivation in recent years. In the next part of this chapter, we will summarize these influential works in more detail.

The publications described in this section were among the first to develop ideas for computational curiosity. They pushed the development of the field of computational intrinsic motivation and have provided a starting point for many approaches to take inspiration from and diverge from.

2.3.1.1 Sutton's Exploration Bonus

In the preceding section, I alluded to how some curiosity-inspired methods have drawn a connection between curiosity and novelty. The original exploration bonus provides an important example of behaviour motivated by a measure of novelty. The core idea behind the exploration bonus is that, we can implement an agent that prefers a type of novelty by adding value to states that have not been visited recently or as often; you might say the agent prefers a change in their surroundings over the tedium of a state they have seen recently. The exploration bonus was introduced by Sutton (1990b) as part of the Dyna-Q+ architecture (p. 221).

Dyna-Q, with its derivatives, is a simple architecture which allows the integration of planning, acting, and learning (Sutton and Barto, 1998, p. 230). The 'Q' in Dyna-Q stems from *Q-values*, another word for action-values (Eq. 2.6), which, as described in Section 2.2, is used to describe the expected return when starting in a particular state, taking a particular action, and following a given policy thereafter (Sutton and Barto, 1998, p. 68). Dyna-Q develops its policy using Q-learning. As a reminder to the reader, the basic idea behind Q-learning is that the agent maintains an estimate of the optimal action-value (Q-value) function so that it can adapt its policy based on these values.

The '+' augmenting Dyna-Q to make 'Dyna-Q+' refers to the use of an exploration bonus. In Dyna-Q+, the exploration bonus is added to the Q-value of a state-action pair to encourage exploration. The value of the bonus for a given state-action pair, (s, a), is computed from a count, $n_{s,a}$, of the number of time steps that have passed since the action a was last taken from that state s. This count is maintained for every state-action pair in the agent's world. This allows the computation of a measure of the uncertainty, $\sqrt{n_{s,a}}$, about the Q-value for that pair. The final bonus added to the Q-value update is a proportional value, $c \cdot \sqrt{n_{s,a}}$, where c is a small positive parameter (Sutton, 1990b, p. 221). The more time that has passed since that particular state-action pair last occurred in real experience, the larger the bonus.

The exploration bonus was introduced as part of a longstanding concern for the exploration–exploitation dilemma, along with other exploration heuristics like the upper confidence bound (UCB) algorithm, optimistic initialization, soft-max/Boltzmann exploration, and ε-greedy exploration. For a review of exploration for learning control as it was understood in the early 1990s, the reader is referred to Thrun's chapter on the topic (1992).

In Sutton's initial experiments, the exploration bonuses in the Dyna-Q+ system tended to improve performance in environments where we might expect a curious learner to perform well: that is, in initial learning and in non-stationary environments. It also, however, led to performance decline where we might expect: in small stationary environments, where exploration ceases to be valuable once the entire state space has been explored (Sutton, 1990b, p. 222).

In the wider context of curiosity-inspired methods, it is valuable to note that Dyna-Q+ built a model and planned its behaviour using that model. Model-building and planning have rarely been highlighted as key aspects of more recent curiosity-inspired algorithms. In the case of Dyna-Q+, planning was done with the model returning not the estimated *true* reward, but the estimated true reward augmented with an exploration bonus. We will see the concept of modifying a value function via use of a model for the sake of curiosity once again in Chapter 5.

2.3.1.2 Schmidhuber's Curious Model-Building Control System

Sutton's (1990b) introduction of the exploration bonus appears to have increased interest in improving exploration methods. Jürgen Schmidhuber (1991a) referred to the Dyna architecture (Sutton, 1990a) as using an "ad-hoc method" for establishment of a world model, and stressed that it failed to address the real-world challenges of "uncertain environments" (Schmidhuber, 1991a, p. 1458). He emphasized that the Dyna approach and other existing work in exploration for agent control had so far neglected to fully take advantage of two potential increases in learning efficiency (Schmidhuber, 1991a, p. 1458). These potential increases could be afforded by avoiding parts of the environment which are already well-understood and by avoiding those parts which have little potential for improving future understanding (Schmidhuber, 1991a, p. 1458). These ideas, first explored in Schmidhuber's 1991 work on curious model-building control systems Schmidhuber (1991a), brought the first experimental trials of reinforcement learning methods devised to produce curious behaviour to the forefront (Baldassarre and Mirolli, 2013, p. 6).

Schmidhuber's goal for his agent in this work was model-building. His definition of this model was simply to require that it must provide a prediction of the "reaction" of the agent's environment at each time step (p. 1459). If we look at this idea of "reaction" within the RL framework described in Section 2.2, it could be any signal available to the agent. In the basic MDP set-up, we could make this "reaction" refer to reward or state, or in a more complex system with GVFs we may want "reaction" to refer to the "pseudo-rewards" associated with some GVFs.

By requiring his agent to maintain such a model as part of its learning computations, Schmidhuber (1991a) allowed the agent to provide a distinction between parts of the world which are already well-understood and parts which are poorly understood. His aim was to leverage this distinction to provide his hypothesized increase in efficiency.

Schmidhuber's approach to curiosity in his 1991 work relied on the parallel concept of *confidence*. Confidence measured the reliability of the model maintained by the agent. The agent can be confident that parts of the world which are reliably modelled are well-understood but should not be confident in parts of the world which are unreliably modelled. He suggests four different approaches to computing the confidence in his model, all of which involve degree of prediction failure or expected error.

The implementation uses changes in error as intrinsic reward:

> The 'curiosity goal' of the control system (it might have additional 'pre-wired' goals) is to maximize the expectation of the cumulative sum of future positive or negative changes in prediction reliability (Schmidhuber, 1991a, p. 1461).

Schmidhuber (1991a) used the model's ability to quickly learn to predict the deterministic reactions of its environment as a measure of success. In particular, he found experimentally that the sum of the squared differences between the model's predictions of the deterministic reactions in the environment and the true reactions decreased much more quickly in an agent motivated by his curiosity goal (Schmidhuber, 1991a, p. 1462).

Researchers have since contended that "this kind of signal cannot cope with unpredictable situations: if it is not possible to anticipate what will be the future state, or the predictor has limited computational capabilities, the prediction errors will not disappear thus providing reward signals to the system that will so get stuck" (Santucci et al., 2012, p. 2). Despite this realization, related motivational functions have shown some success. For example, Ngo et al. (2013) made use of the simple idea of producing goals based on how reliably their system could predict the reaction of the environment to guide the behaviour of a robotic arm (Baldassarre et al., 2014, p. 3).

A further limitation of this method may be that a simple model for prediction of the "reaction" of a state does not necessarily define "understanding" state from the point of view of an agent which may have other goals beyond model-building. Schmidhuber's method encourages the agent to pursue the task of model-building when that task is expected to be fruitful; possibly other tasks should also be pursued when fruitful.

2.3.2 Evolutionary Approaches

Several computational researchers have been inspired by the idea that intrinsic motivation may have developed in response to evolutionary pressures (for example, Schembri et al., 2007b; Klyubin et al., 2005). In some cases, the resulting computational approaches to intrinsic motivation do not fall clearly within the realm of reward signals reflecting changes in the "knowledge and know-how" of the agent, independent of any actual meaning a designer might attribute to the sensor observations, as Oudeyer and Kaplan (2007, p. 12) describe intrinsic motivation. In this document, I focus on approaches to computational intrinsic motivation that *do* fit Oudeyer and Kaplan's description, so I will only briefly summarize evolutionary approaches to intrinsic motivation. Evolutionary approaches to intrinsic reward, in particular, instead focus on how reward signals that motivate agents to explore might have developed through an evolutionary process as improvements to agent fitness over multiple generations.

The "Evolutionary Perspective" developed by Singh et al. (2010) provides a representative discussion. The authors developed an evolutionary framework for intrinsic reward with a particular focus on factors distinguishing intrinsic motivation from extrinsic motivation (p. 70). Singh et al. used evolutionary search methods to choose the reward signal function as a whole (p. 73). They performed their experiments in environments with small numbers of simple fitness-increasing events, which they compare to biological reproductive or feeding events (p. 75). Of particular interest is their use of environments in which the fitness-increasing events change location, applying a kind of evolutionary pressure. The results presented by Singh et al. (2010) suggest that, with this kind of unpredictability in play, evolution selects for agents with reward signals encouraging exploratory behaviour (pp. 77, 79).

To Singh et al. (2010), what psychology normally recognizes as a distinction between extrinsic and intrinsic motivations may actually be a reflection of our poor understanding of the causal structure of our world. Their results support an evolutionary view of natural reward systems where all reward signals could be the result of a planning agent with imperfect knowledge of the world being best suited for evolutionary fitness by valuing exploratory choices. Evolutionary development of intrinsic motivation is a strong hypothesis for the development of curiosity in humans and animals.

Computational evolutionary approaches show promise for developing useful intrinsic reward systems for well-specified environments. Despite this promise, I would argue that human-developed intrinsic motivation systems have a different value, especially for developing our understanding of the connections between intrinsic motivation systems and behaviour; this is desirable not only for producing robots and other computational agents that can interact safely with humans, but also potentially for application to our understanding of human and animal intrinsic motivation.

It may be that intrinsic motivation and curiosity in humans have arisen from overarching reward functions evolving over time. Even if this is the case, we can still ask: Out of what information might these specific rewards be computed? In the next sections, we will look at some potential answers to this question.

2.3.3 Measures Based on Error

The first class of computational curiosity approaches I would like to introduce is the class of **error-based approaches**. Many error-based approaches motivate curiosity by focusing the agent on the unexpected. Looking to humans as a favourite example of a curious agent, discovering something new can feel inherently rewarding, driving us to exhibit curiosity. Because we are unlikely to correctly predict a novel stimulus, high prediction error can indicate a novel experience. Error-based approaches leverage this idea, encouraging agents towards further experience with surprising stimuli (experienced error) or poorly-explored states or novelty (expected error). In this section, we cover two case studies of error-based approaches.

2.3.3.1 Intelligent Adaptive Curiosity

Our first case study will be the Intelligent Adaptive Curiosity (IAC) mechanism presented by Oudeyer et al. (2007). IAC represents a key moment in the history of machine curiosity. Study of machine curiosity appears to have hit its stride with the growth in interest in developmental/epigenetic robotics at the turn of the century. Herrmann et al. (2000) offered initial ideas for machine curiosity for autonomous robots, but it was the line of research pursued by Oudeyer et al. to develop intrinsically motivated systems (Kaplan and Oudeyer, 2003; Oudeyer and Kaplan, 2004; Oudeyer et al., 2007; Baranes and Oudeyer, 2009) that appears to have begun the most influential push towards designing machine curiosity.

In their seminal paper, Oudeyer et al. (2007) they followed a line of thought inspired by Berlyne (1960) and Csikszentmihalyi (1991) who suggested that internal reward is maximized when the agent deals with a challenge which “is not too easy but also not too difficult” (Oudeyer et al., 2007, p. 266). We can describe the IAC mechanism as an error-based approach because for each pair of consecutive time steps, the agent makes a prediction of the state signal it expects to experience next, then compares it to the state it actually experiences, and computes the discrepancy between them (its error) using some distance function (pp. 270-271). These errors, kept in memory, are used to compute the value Oudeyer et al. (2007) call *Learning Progress* (p. 271). Learning Progress is computed based on a comparison between two windows of time steps. One window is the most recent η steps,[3] and the other is a window of the same size τ steps prior, where the η and τ are integer parameters, with η generally ≈ 25 and $\tau \approx 15$. Oudeyer et al. (2007) take the difference between the average squared error over each window of time, which is the increase in error, and negate it to compute Learning Progress (p. 271). For example, if the average error for the earlier window was 50, and the average error for the most recent window was 100, then the Learning Progress would be -50, since the agent's predictions have become worse. Oudeyer et al. (2007) use Learning Progress as the intrinsic (and only) reward for the IAC system (p. 272).

Oudeyer et al. (2007) tested their IAC mechanism using two different experiments. The first experiment is called the Simple Simulated Robot Experiment, an experiment in which the open-ended behaviour of a simple simulated robot (a box with two wheels, a sound emitter, and a sensor allowing it to perceive its distance from a small toy) is observed. The robot can control its wheels and sound emitter independently, setting a real value within a known interval for each. The wheels allow the robot to move around the room, but the key part of its control system

[3] The parameter here denoted by η is the smoothing parameter, denoted by θ in the original description by Oudeyer et al. (2007, p. 271); this change of notation was to avoid confusion with the symbol $\theta(t, i)$, which is used elsewhere in this document to denote a distribution.

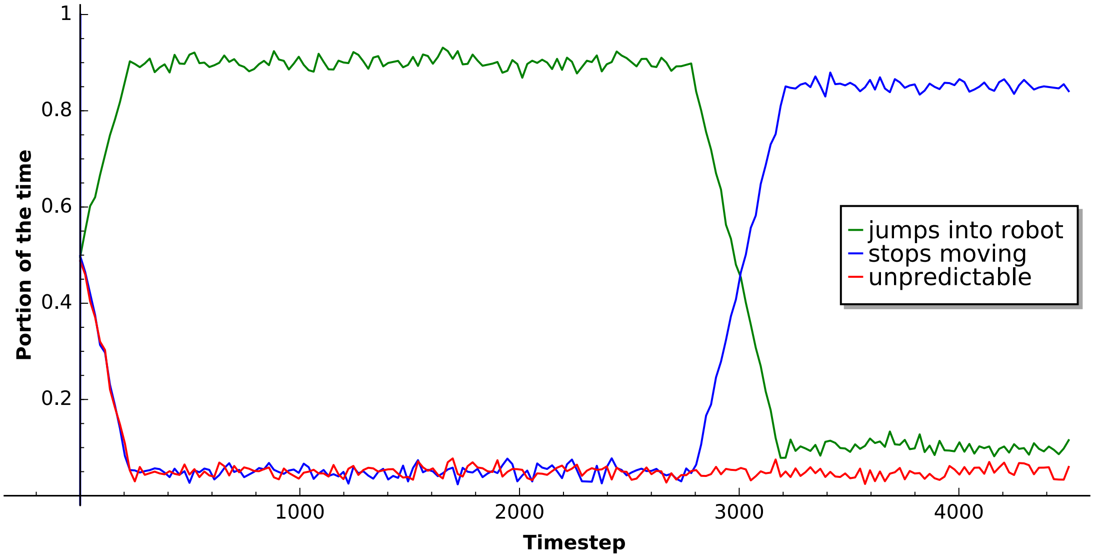


Figure 2.3: This plot, adapted from Fig. 4 from Oudeyer et al. (2007, p. 274), shows the approximate evolution of time spent: (in red) emitting a tone in the frequency range which results in the toy moving randomly, (in green) emitting a tone in the frequency range which results in the toy not moving, and (in blue) emitting a tone which results in the toy jumping into the robot.

is that the tone emitted by the robot determines the behaviour of the toy. Of the interval of possible tone frequencies available to the robot, one third causes the toy to move randomly, one third causes the toy to stop moving, and one third causes the toy to jump into the robot (p. 273).

The general shape of the behaviour followed by the Simple Simulated Robot controlled using the IAC mechanism is shown in Figure 2.3. For a short (approximately 250-step) phase of the 5000-step trial, the agent chose actions apparently randomly. For the next phase, the robot generally tended toward the frequency range where the toy would simply jump into the robot (simple because the robot's understanding of the toy is based on its distance away), and in the third phase, it tended toward the frequency range where the toy stopped moving. The robot was consistently disinterested in frequencies which resulted in the unpredictable, random movement of the toy (p. 273).

The second experiment used was called the Playground Experiment. Oudeyer et al. (2007) set up a baby play mat with objects that could be seen, bitten, or bashed. They implemented their IAC mechanism in a Sony AIBO robot agent. They found that the agent focused on particular "sensorimotor loops" for segments of time, with the complexity (in terms of dimensionality) of those loops increasing over time (p. 278).

In both experiments, the observed behaviour can be understood as pursuing increasing levels of complexity over time. Oudeyer et al. (2007) assert that this is evidence that the IAC mechanism has allowed the agent to "autonomously generate a developmental sequence" (p. 284).

2.3.3.2 General Value Function Surprise as Generating 'Curious' Behaviour

While the curious agent implemented by Oudeyer et al. (2007) chooses actions based on prediction areas where they seem to be making the most improvement, White et al. (2014), in proposing the method at the centre of our second case study, developed an approach to curiosity *without* the concept of improvement. Instead, White et al. (2014) use the term "curious behaviour" to refer to reactive changes in behaviour to encourage re-learning in the face of surprise. In their paper, they defined surprise as "unexpected prediction error" (p. 19). To compute the surprise about a single GVF prediction i, they made use of the TD-error, $\delta^{(i)}$, in their prediction of that sensor value. This allowed them to define the surprise, $Z_t^{(i)}$, about GVF prediction i at time t, in computational terms as

$$Z_t^{(i)} = \frac{\overline{\delta^{(i)}}}{\sqrt{\text{var}\left[\delta^{(i)}\right]}} \tag{2.9}$$

where $\bar{\cdot}$ refers to an exponentially-weighted average.[4]

White et al. (2014) suggest that they wanted to approach the problem of selecting actions which provide effective training data for "learners with diverse needs"

[4] In later work, White (2015) refers to the construct in Equation 2.9 as Unexpected Demon Error (UDE), where the "demon" is the prediction learner for prediction i. We will see UDE again in Chapter 3.

(p. 19). Though inspired by a bigger goal of making action choices for life-long learning, their work is restricted to enabling changes in behaviour based on surprise *without* providing mechanisms for the agent to decide what those changes should be.

We can see how White et al. (2014) enabled changes in behaviour by considering where they implemented checking the levels of surprise in a robotic agent. The experimental set-up used to test this measure of surprise used a robot making predictions about two of its sensor values. Once the robot had learned to accurately make both of these predictions, the humans suddenly made a change to the world that changed the pattern obtained by a single sensor. Their system design successfully allowed the robot to adaptively recognize the need to re-learn that prediction pattern.

In this implementation, the policies used to learn the given patterns are human-designed and fixed. However, the surprise measure $Z_t^{(i)}$ could feasibly be adapted help the agent choose a policy. As another potential avenue to build on their work, one could consider adapting what White et al. (2014) consider to be one of their major contributions: "the first measure of surprise based on off-policy GVF learning progress" (p. 22). In this case, the measure of surprise is based on instantaneous temporal-difference error in the off-policy prediction. We can imagine, however, applying other existing measures of curiosity to off-policy GVFs to achieve different behaviour.

What is referred to as 'curious behaviour' by White et al. (2014, pp. 19–20) might better be called determined or focused behaviour. As an example, perhaps if a human tried to pick up a cup and expected to be successful, and instead failed, they might realize that they want to be able to predict where their hand ought to be each time they attempt to pick up a cup. This way, they will be successful at this task when they need to be. As a result they might try to pick up the cup until they feel confident in their ability to complete the task consistently or until they realize (perhaps if the reason for their initial failure was a prank) that the task is so unpredictable that it is not worth trying any more. Although this tendency is definitely useful, and might fall under the umbrella term of intrinsically motivated behaviour, it does not seem to have the character of curiosity.

With respect to intrinsically motivated behaviour, on the other hand, animal psychologists have observed a somewhat similar kind of behaviour in young rhesus monkeys. In the case of rhesus monkeys, the monkeys were observed leaping, but varying where and how they leaped. Simpson (1976) has referred to this repetition of single tasks as 'projects' (p. 386). "Such patterns of behavior are often thought of as play because they appear to be circumscribed in time and they do not satisfy an immediate need" (Kubovy, 1999, p. 148). The defining difference between Simpson's 'projects' and White et al.'s 'curious behaviour' (pp. 19–20) may be the aspects of variation and development. The leaping project starts off simple, perhaps repeatedly jumping up to a low branch, but as the monkey develops, its projects become more complicated and progressive. How the monkey chooses to vary its leaping is the active component of play which might be seen as curiosity.

2.3.3.3 Discussion of Error-Based Measures

Though I have provided detail regarding only two studies in this section, error—in some cases expected and in others experienced—form the basis of many existing approaches. While I presented Sutton's (1990b) exploration bonus and Schmidhuber's (1991a) curious model-building control system in an earlier section (2.3.1), both belong to this family, as they used measures based on error to encourage curious behaviour. Sutton's (1990b) exploration bonus rewarded expected error: actions taken in states which have not been visited recently are more likely to have an unexpected reaction, particularly in non-stationary settings. Schmidhuber (1991a) explicitly used prediction error for intrinsic reward.

Even more proposals from this family of approaches might be considered. Schembri et al. (2007b) produced an example of similar work. They used error, specifically TD-error (Baldassarre and Parisi, 2000, p. 134), as the intrinsic reward (which they called ‘surprise’) as a component of their system evolving intrinsic reward signals. Stout and Barto (2010, p. 838) note that what Schembri et al. (2007b) refer to as ‘surprise’ and use as intrinsic motivation forms the simplest case for the intrinsic reward, ΔV, first presented by Şimşek and Barto in 2006. Stout and Barto used ΔV for their competence progress mechanism in 2010 (p. 835). The value of ΔV is a short-term error: the agent makes estimates for the value of its greedy policy at each time step, and ΔV is the amount that this estimate has changed over one time step (Stout and Barto, 2010, p. 835).

All of these approaches make use of error in their computation of intrinsic motivation, but the mechanisms still vary greatly. While Sutton (1990b), Schmidhuber (1991a), Schembri et al. (2007b), and Oudeyer et al. (2007) took approaches focusing on the intrinsic values of states and actions, Stout and Barto (2010) and White et al. (2014) took a view one layer up, using curiosity not to pick the best (most curious) next action, but to pick the next (policy) focus for the duration of an option.

Despite these differing perspectives, both sides had some similar results in their experiments. Both Oudeyer et al. (2007) and White et al. (2014) found that, in implementation, they observed their robotic agents focusing on a group of actions and states for periods of focused development. This behaviour was clearly engineered in the algorithm by White et al. (2014), whereas this behaviour follows in a more complicated matter from the design by Oudeyer et al. (2007) As mentioned while describing the approach by White et al. (2014) in Section 2.3.3.2, these types of focused behaviour share a certain resemblance to ‘projects’ Simpson (1976) observed in animal behaviour, seeming to be key for their development. In keeping, several error-based approaches are designed to push an agent towards ‘project’-like behaviours, which focus on one task for a while and then move on to another of greater complexity.

2.3.4 Measures Based on Information

Part of the philosophy of the reinforcement learning framework is to let the machine decide the best way to act to achieve a goal, avoiding human-designed behaviour choices. Unfortunately, our attempts to work within this philosophy towards computational curiosity are hindered by our uncertainty regarding what the goal of curiosity is. One possibility is the accumulation of information, and this leads to the next class of approaches.

As introduced with Berlyne's ideas regarding curiosity in Section 2.1.1, Claude Shannon's *Mathematical Theory of Communication* and the field of information theory have given a reasonable theoretical basis for the amount of information contained in data. The following approaches use a notion of the "amount" of information that their agent holds and try to increase it intelligently. For this reason, I classify these approaches as based on information.

One of the key concepts shared by several of the information based approaches is *mutual information*. The mutual information of two discrete random variables X and Y is

$$I(X;Y) \coloneqq \sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{2.10}$$

where where $p(x,y)$ is the joint probability distribution function of the random variables X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively (MacKay, 2003, p. 143). Intuitively, the mutual information measures, for each pair of possible events x, y from the two random variables X and Y, the decrease in the uncertainty of x if we know y occurs (Osteyee and Good, 1974, p. 10; Cover and Thomas, 1991, p. 18). Alternatively, the mutual information can be thought of as the decrease in the amount information we would gain by the occurrence of x caused by the occurrence of y (Osteyee and Good, 1974, p. 10).

2.3.4.1 Empowerment

In 2005, Klyubin et al. introduced ‘empowerment’ as an objective for knowledge acquisition; like Barto (2010), they take the line that “it is all about control” (2010, p. 3) but also value the agent-centric idea of perception, developing a measure for the amount of control that an agent *perceives* it has over its environment (2005, pp. 128–129). To develop such a measure mathematically, Klyubin et al. (2005) take advantage of the information-theoretical concept of communication systems, as developed by Shannon (1948).

The key communications systems concept used by Klyubin et al. (2005) is that of a *channel*. When describing the parts of a communication system, Shannon (1963) defines a channel as the medium over which a signal travels from the transmitter to the receiver (p. 5). The transmitter’s signal in a communication system may be deterministic, but in most situations, the signal is *random*, in that it is unpredictable—an observer would not know with exact certainty the signal’s future value, even with knowledge of its entire history (Rao, 2009, p. 2.1). Because of this uncertainty, we may represent a signal with a random variable, X. Furthermore, the channel over which the transmitter’s signal travels may be noisy, and so the signal retrieved by the receiver is generally represented by another random variable, Y.

When the signals X and Y are discrete, the channel is completely characterized by the conditional probability distribution $p\left(y|x\right)$, because this defines how “transmitted signals correspond to received signals” (Klyubin et al., 2005, p. 129). It is sometimes useful to consider how much information from the transmitted signal can possibly be received: this is called the *channel capacity*, $\max_{p(x)}$. Recall is the mutual information, as defined in Equation 2.10. By taking the maximum over all possible distributions of the transmitted signal, the capacity allows us to maintain a measure of the most information we could possibly receive by modifying the transmission signal.

With this mathematical framework in mind, Klyubin et al. (2005) translate the reinforcement learning framework into a communication system framework by treating the sequence of actions chosen by an agent as a transmitted signal to their later self, through the very noisy channel of the environment, to eventually be received by the agent's own sensors.

Like Oudeyer et al. (2007), Klyubin et al. (2005) make use of a window of time steps, writing specifically of the length of a sequence of actions (p. 130). For a window of length n, the signal starting at time step t is the sequence of actions taken within that window, $a_t^n = (a_t, a_{t+1}, ..., a_{t+n-1})$. That signal is received by the agent at time $t + n$ as their state signal s_{t+n} (p. 130). Klyubin et al. (2005) then write A_t^n and S_{t+n} as the random variables from which a_t^n and s_{t+n}, respectively, are outcomes.

Klyubin et al. (2005) define empowerment as the information-theoretic capacity of an agent's actuation channel (p. 128): the *empowerment* of the agent at time t is written as

$$\mathfrak{E}_t = \max_{p(a_t^n)} I(X;Y) \left(A_t^n; S_{t+n}\right) \tag{2.11}$$

Klyubin et al. (2005) approach the goal of maximizing empowerment by evolving its sensor and actuator, but we could just as easily imagine using this measure as an objective for a learning agent to maximize during its lifetime (p. 131).

2.3.4.2 Predictive Power with Limited Complexity

In 2012, Still and Precup described another example of an information-gain approach: predictive power with limited complexity. Like Klyubin et al. (2005), who emphasized an agent-centric approach, Still and Precup (2012) make use of information-theoretic measures of how much information is held by two probability distributions specific to the agent. Rather than the probability distributions defined by the agent's sequence of actions and sensor input, they consider the distributions implicit in a model maintained by the agent of the probabilities of being in each visited state, and the probability distribution defined by the agent's policy.

Still and Precup's (2012) choice of probability distributions allows them to approach curiosity through two concepts. The first is an objective to maximize the predictive power of the agent's policy—Still and Precup (2012) use a formal definition of predictive power which has been used in the literature to measure "how complex, surprising, or 'interesting,' a time series is." Intuitively, increasing predictive power should mean an increase in the agent's knowledge of the environment.

The second concept is limiting the complexity of the agent's policy. Limited complexity is obviously desirable with limited computational resources, but Still and Precup (2012) suggest that it is also useful in organizing an agent's development to tend towards more sophisticated behaviours as the agent attains more experience.

These two concepts are central to Still and Precup's (2012) aim to balance control and exploration (p. 143). The first concept, increasing predictive power, should push the agent to explore, while the second concept, limited complexity, adds more control. While the goal of the first part of Still and Precup's paper is similar to that of Şimşek and Barto's (2006) work in that they aims to optimize exploration, Barto (2010) might question this choice of balance. He argues that exploration is only useful to the extent that it later facilitates control, and so might question whether the goal of curiosity should indeed be to balance control and exploration, or instead be to balance present and future control.

Still and Precup (2012) give a simple example setting where the agent, in a two-state world, has a continuous action space, $[0, 1]$, where, if the agent takes action a, then with probability a, it will move to the other state (p. 143). An agent motivated using Still and Precup's (2012) approach has an optimal policy which only chooses actions 0 and 1, with equal probability. This example suggests that their curiosity mechanism seems to only motivate for exploration of state, not action. The optimal policy with regards to their curiosity measure stops 'exploring' all possible actions.

2.3.4.3 Discussion of Information-Based Measures

Intuitively, the goal of curiosity is to gain information. This makes information-based measures very appealing objective functions for computational curiosity. Though the detail in this chapter is limited to carefully developing only two approaches, a number of others have been invented and should be considered as part of the family of information-based approaches. For instance, Polani, one of the designers of the empowerment measure we explored in Section 2.3.4.1, has since collaborated with Tishby to publish a 2011 work that uses an information theoretic perspective to approach balancing the costs and benefits of information-sampling (Gottlieb et al., 2013, p. 587).

Before that, Ay et al. (2008) tried to motivate "behavior which is both explorative and sensitive to the environment" in a robotic agent (p. 329). Their 2008 work did not consider the actions of the agent, but considered the mutual information held between the sensor values in one step and the next, $I(X;Y)(S_{t+1};S_t)$ (pp. 329, 333).

Itti and Baldi (2005, 2006) developed a measure of 'surprise' and showed that their measure was closely correlated with the attention of humans in terms of eye tracking. They did not use their measure to motivate a computational agent, but they influenced Abdallah and Plumbley, who, in 2009, argued that an observation might be better valued not by its inherent surprisingness, but by the amount of "information it carries about the unobserved future," given what we know about the past (p. 93). They computed an information measure that they called *predictive information*. Like Still and Precup (2012), both Itti and Baldi (2005, 2006) and Abdallah and Plumbley (2009) used the Kullback–Leibler (KL) divergence.

More recently, Orseau (2014) published two objective functions to motivate knowledge-seeking, both based on measures of information. He published the algorithms for his Square Knowledge Seeking Agent and Shannon Knowledge Seeking Agent in 2014. In the same year, Gordon et al. (2014) developed an approach that modularly combined an information basis with ideas from predictive, error-based approaches.

Though it seems that increasing the quantity of information held by the agent is the goal of curiosity, we are not curiously motivated by all information. We certainly are not attracted to signals with the most information—that is to say, signals that are completely random. Both Klyubin et al. (2005) and Still and Precup (2012), in this light, restrict the information the agent is inclined to acquire. Klyubin et al. (2005)'s approach is to maximize information about future states, but limited to the information that can be held by the agent's sequence of actions. Still and Precup (2012)'s approach, on the other hand, is to maximize the information about the future state distribution carried by the agent's current state and action, while limiting the information about the actions carried by the state. The different methods for restricting which information is attractive to the agent form some of the key differences in information-gain approaches.

***

In this chapter, we have explored some of the key ideas that have shaped the study of machine curiosity. We have also sampled some of the variety of methods that have been proposed for computational curiosity. We may gain a better understanding of these methods by contrasting the behaviour each approach generates, with both information-based approaches and error-based approaches. In the chapter that follows, we will see the first main contribution of this thesis: a family of experiments designed for the empirical comparison of these already-existing methods.

Chapter 3

The Curiosity Bandits

3.1 Principled Study of Curious Behaviours

One way we hope to develop our understanding of curiosity is to model curiosity and employ these models of curiosity in computational systems. As curiosity appears to *motivate* many of the decisions made by animals, including humans, reinforcement learning (RL) is a strong candidate for helping us develop such models—an idea already suggested in Chapter 1 (Section 1.1). RL provides a natural approach for designing different motivations for our systems. In particular, RL allows machines and biological systems to learn, through trial and error, the value of situations and choices. In computational RL, we formalize this idea by requiring that a signal known as *reward* is delivered to the learner throughout its interactions with its environment. Reward can be used to provide a type of motivation for computational systems. Existing RL algorithms can be used to efficiently learn which actions maximize future reward. Designing our reward, therefore, offers an approach for designing different motivations for our systems. Researchers have developed different methods to modify the reward delivered to a learner or to modify other parts of an RL algorithm so as to evoke behaviours inspired by curiosity in their systems. Many of these methods have shown promise in real-world or simulated domains, as can be seen with the examples described in Section 2.3.

However, it is unclear whether different learning systems inspired by curiosity produce qualitatively different behaviour when placed in the same situation. To create a clear comparison of different curiosity methods, one approach is to hold both the domain and the majority of the agent's internal workings constant, varying only the curiosity method used to motivate the agent. This approach is particularly straightforward when working with curiosity methods primarily centred on the design of an intrinsic reward, and is analogous to a suggestion by Oudeyer and Kaplan (2007, p. 13). Repeatedly placing such agents in a single domain while varying their curiosity methods (intrinsic rewards) can allow us to clearly see how each agent's behaviour differs with respect to other agents in the cohort. To gain an understanding of how the resulting behaviours compare to what might be expected or desired given the methods' theoretical underpinnings, we suggest that initial experiments should be carefully designed, using uncomplicated domains with variations specifically chosen to untangle the differences between curiosity methods. The domains presented in this chapter are examples of this kind of carefully designed domain.

In this chapter, we describe experiments assessing how different approaches to curiosity in RL lead to different behaviours within a controlled experimental domain. The principal contribution of the work presented in this chapter is a family of domains that allows us to investigate the behaviour elicited by different computational curiosity approaches. Results from the study of curiosity in psychology have suggested that behaviour associated with curiosity may be prompted by, roughly, a degree of difference in "comparison of information from different sources" (Berlyne, 1963, pp. 290, 292; see also Berlyne, 1966, p. 30), sometimes stripped down to "the degree of novelty, surprisingness, and complexity" (Berlyne, 1963, pp. 292). In the domains presented in this chapter, an agent observes a variety of signals of differing variability—or complexity—and, importantly, which signal the agent observes depends on their own actions. By focusing the design of the domains on action-dependent observations, we provide insight into how different approaches to computational curiosity drive agent behaviour.

Our results represent the first look at computational curiosity in unified settings, and are therefore an important step toward a better understanding of curious behaviour in learning systems. We expect the principled understanding of computational curiosity will make significant contributions to the development of general machine intelligence.

3.2 Previous Comparative Studies

Despite a boom of new research on computational intrinsic motivation, little has been done to compare the many methods springing into existence. To the best of my knowledge, only three publications prior to the initial publication of the experiments presented in this chapter provide comparative analyses of different approaches. In this section, I will describe the work exhibited with those publications and contrast their goals with my own.

In 2007, Oudeyer and Kaplan explored a variety of ways to classify both existing and newly proposed approaches to intrinsic motivation in terms of the intuitive motivation behind their theoretical foundation. Their broad collation of different intrinsic reward approaches allowed them to present an important early definition of computational intrinsic motivation (IM):

> Each of the described models defines a certain interpretation of intrinsic motivation in terms of properties of the flow of sensorimotor values and of its relation to the knowledge and know-how of the system independently of the meaning of the sensorichannels that are involved. *(Oudeyer and Kaplan, 2007, p. 12)*

Additionally, their typology helped them to develop hypotheses about which types of IM “can lead to open-ended developmental trajectories” like those observed in human children: they suggest Information Gain Motivation (IGM), Learning

Progress Motivation (LPM), and Competence Progress Motivation (CPM) (p. 13). For experiments in more complex domains in which different IRs cannot be comprehensively tested, such hypotheses (along with the results of other comparative studies) may act as suggestions about which varieties of IR to prioritize. Though Oudeyer and Kaplan (2007) note the significance of the behavioural trajectories of different approaches, behavioural comparison and analysis is beyond the scope of their work (pp. 5, 13)—a gap to which we propose to contribute in this work.

Taking a more experimental tack, Santucci et al. (2012) investigated which IM signals would be best suited to develop a learner's "capacity to act so as to achieve a state of the world when it becomes desirable," which they called *competence* (pp. 1, 5). While their 2012 work used "a simple grid-world environment," (p. 1) they continued this line of work comparing different IM choices into 2013b, with their publications in the latter year focused on robotic experimental domains (Santucci et al., 2013b, p. 1; Santucci et al., 2013a, p. 1). The goal to have open-ended learners develop competencies or skills is an example of an intelligent-systems-designer's goal that curiosity or intrinsic motivation are thought to potentially support. Historically, we have developed systems whose operating procedures are designated by their human designers. But the reality is that, in many situations, a system designer cannot be expected to determine the best way to operate in every environment the system could encounter. Rather, the system itself is in the ideal position to determine its own capabilities. We can observe humans discovering their own capabilities—we see human infants testing their own motor functions and refining these functions into skills to apply in different situations. Curiosity has long been posited as supporting competence acquisition (White, 1959, p. 318), so metrics that evaluate competence offer one important perspective with which to compare different computational curiosity or intrinsic motivation approaches. However, there is further work to do: A small number of metrics cannot provide us with the breadth or depth of understanding we might want for the variety of potential applications of curious machine learners.

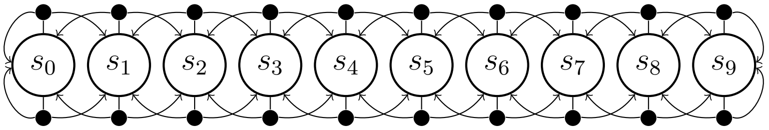


Figure 3.1: The environment used by Santucci et al. (2012) for their comparative study. It is a simple 1 × 10 gridworld. The agent has two options from each state, 'right' and 'left.' For either option, in any state, the agent has a 95% chance of moving in the direction specified, and a 5% chance of moving in the opposite direction (Santucci et al., 2012, p. 2).

In 2013, Santucci et al. wrote that their 2012 analysis was "limited" to "a simple grid-world environment" (p. 1). While simple domains with discrete states can feel quite disconnected from "animal, human and robotic learning which takes place in continuous states and actions" and may be of primary interest for many researchers (Santucci et al., 2013a, p. 2), the use of complex robotic domains has downsides. For one thing, a robotic domain adds further parameter decisions, and, as will be clear from Section B.1.3, the number of parameters involved in a simple discrete-state study is already substantial. For another, it complicates the researcher's ability to make the kinds of precise observations of behaviour we will use in Section 3.4.2. Yet, we too recognize that some cases of curiosity as it is observed in humans and other animals may only be applicable in environments that allow for generalization and temporally extended sequences of decisions (see Chapter 4), which is an important limitation of simple domains with discrete states. However, methods specifically centred on IR can often be faithfully ported to discrete-state domains and meaningfully compared, allowing us to take advantage of simplicity.

In this chapter, we will present experiments comparing multiple IRs on small discrete-state domains, just like Santucci et al. (2012), making their 2012 grid-world study the most closely related work to that presented in this chapter. They compared three different mechanisms (p. 4). These mechanisms included one based on the IAC mechanism by Oudeyer et al. (2007; see Section 2.3.3.1 for a summary), one analogous to the mechanism designed by Barto et al. (2004), and the last similar to work published by Hart and Grupen (2013). Santucci et al. (2012, p. 2) tested their implementations of these mechanisms in the decision process environment shown in Figure 3.1. For each approach, they ran the simulation for 100,000 trials (p. 4). At the beginning of each trial, the agent was placed randomly in one of state s_0, $s_1, ..., s_8$. The trial terminated when the agent reached s_9 for the first time or at a timeout of 20 time steps.

While the robotic experiments presented by Santucci et al. (2013a,b) are less closely related to our work than their predecessor, we still note here their conclusions, as they add to our general comparative understanding of IR approaches. Their first 2013a paper extended their setting to include multiple skills and continuous state and action spaces and the second included a couple more approaches in their comparison. Only error-based approaches were compared in their studies. The main conclusion offered by Santucci et al. (2013b) from across all three papers is that the best performance was achieved by "coupling the activity of the mechanism generating the IM signal to the competence of the system in performing the different tasks" (p. 1).

3.2.1 Evaluation of Behaviour

This subsection is about how we might evaluate behaviour, drawing ideas from the literature. Much like the experiments performed by Santucci et al. (2012), the experiments set out in this chapter manipulate the intrinsic motivation approach used by the learner in simple decision processes. However Santucci et al. (2012) set out a clear goal and performance metric for how well the agent achieves this goal under the influence of each motivational approach. Evaluating the behavioural trajectory of agents is not so cut and dry.

In the same way that Santucci et al. (2012) chose to measure how many skills an agent can learn quickly, other authors have suggested possible measurements of how well an agent explores. We recall that Schmidhuber (1991a) measured the speed at which the agent learned to predict deterministic parts of its environment.

Lim and Auer (2012) developed a measure of the "amount of exploration [the agent] uses to learn the environment" (p. 40.1). We can engineer agents which will attempt to maximize these measures, but these measurements give little insight into the behaviour we will observe from an agent over a long lifetime.

We are interested in the behaviour of the agent. When referring to this perspective, Oudeyer et al. (2007) specify this perspective as being an *external* point of view on the behaviour. We especially considered two suggestions about how to evaluate a learner's behaviour with regards to curiosity:

- In evaluating the behaviour of the agent, it is helpful to call on our intuition of what we expect curious behaviour to exhibit. As initially suggested by Schmidhuber (1991a, p. 2), intuitively, neither chasing patterns in the environment which are easily understood nor chasing (non-)patterns in the environment which are impossible to understand are curious choices.

- When studying the external behaviour of the simple simulated robot motivated by their own approach, Oudeyer et al. (2007) characterizes the frequency with which the agent takes actions with specific impacts on its sensor feedback (tones in each interval). It seems that the simplest measures of behaviour could be similar for our experiments.

Therefore, we propose quantifying what percentage of a trial the agent takes each action. Measuring these proportions will give a sense of important general tendencies, but for a more subtle understanding of the behaviour, it will be important to be able to see progression over a trial. Because this environment is so simple, it it be possible to test most approaches over multiple runs. Further, it allows us to ask, with what frequency is each action taken? Using this, we can produce and analyse a graphical representation of any tendencies of action over time. Further, we can go beyond the methods used by Oudeyer et al. (2007), we can ask similar questions regarding states and state-action pairs. These measures are simple and provide more insight into the behavioural development of learners motivated by existing computational curiosity approaches than has been explored in the prior literature.

3.3 The Curiosity Bandit Family

The major contribution of this chapter is the introduction of decision-making problems designed to provide opportunities for agents to exhibit behaviour characteristic of some conceptualizations of curiosity. These decision-making problems are the *Curiosity Bandits*, which we devised to showcase the behaviour elicited by variations in a real-valued signal (e.g., the cumulant used for the computation of a GVF). The Curiosity Bandit derives its name from bandit environments; an n-armed bandit decision problem has a single state (equivalently can be thought of as stateless) and n possible actions (the n arms) to choose from. For a detailed introduction to bandits, see Chapter 2 of Sutton and Barto (2018).

The Curiosity Bandit problems are designed to showcase a learner's ability to differentiate the interestingness of different signals sampled by choosing different actions. Examples of this setup are shown visually in Figures 3.2 and 3.8. One longstanding view of curiosity is that it motivates agents to maintain an intermediate level of 'arousal' (*cf.* the Wundt curve as described by Berlyne, 1960, pp. 200–201; Kidd et al., 2012 also provide evidence that human infants allocate more attention to visuals that are neither too simple nor too complex, p. 1). In the design of the Curiosity Bandits, the choice of signals (which are paired with actions) is meant to offer a variety of complexity levels from very simple (*e.g.*, a signal that is always constant) to more complex (*e.g.*, a high-variance, random signal). Example choices of signals are shown visually in Figures 3.2 and 3.9 and instances of curiosity bandits will be described in detail for the two studies that follow in this chapter.

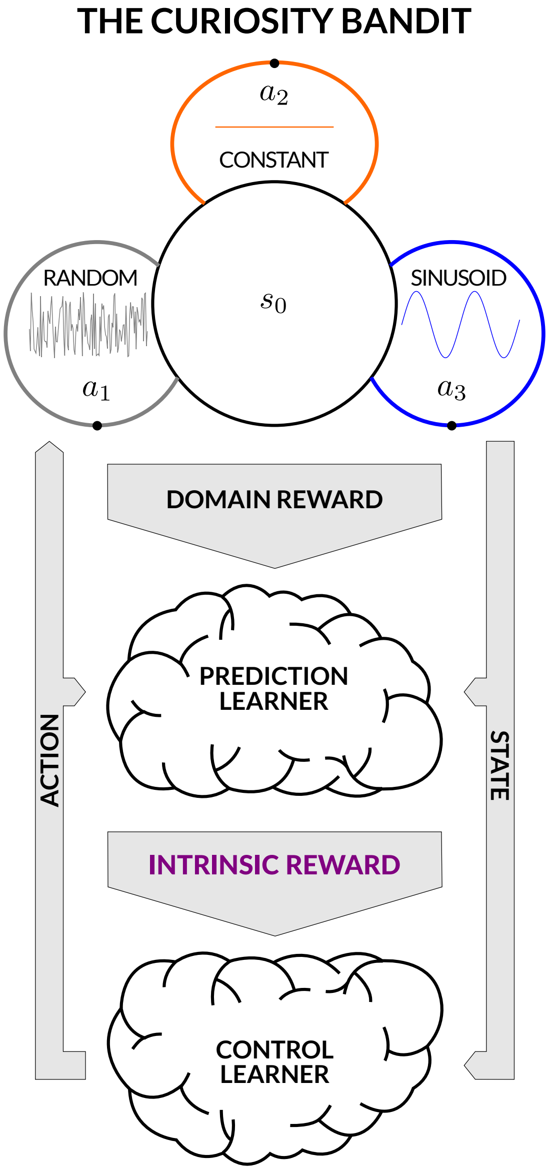


Figure 3.2: The Curiosity Bandit domain (upper) and its interaction with a two-part agent (lower) suitable for error-based intrinsic reward methods.

3.4 First Study: Behavioural Comparison

We devised the first Curiosity Bandit to showcase the behaviour elicited by variations in *extrinsic reward*. This original design is depicted in Figure 3.2. In this first study, the Curiosity Bandit has a single state ($\mathcal{S} = \{s_0\}$), but each of its three actions, $a_1, a_2, a_3 \in \mathcal{A}(s_0)$ result in a different pattern of rewards. If the agent takes a_1, its reward is drawn uniformly randomly from $[-1, 1]$—the *random* action. If the agent takes a_2, it always receives a reward of 0—the *constant* action. If the agent takes a_3, then it receives a reward of $\sin(c \cdot t)$—the *sinusoid* action, where c is a small constant (set to $c = 0.001$ in the included experiments) and t is the current timestep (starting at $t = 0$ in our experiments). All three arms deterministically return to the same state s_0 and produce reward between -1 and 1 with mean 0. The *constant* action deterministically results in a reward of 0. The *random* action stochastically returns any value within the continuous interval $[-1, 1]$. The reward resulting from taking the *sinusoid* action is a deterministic function of the current timestep t. The Curiosity Bandit is not Markov (this is in contrast with a standard bandit, which would have the Markov property). In reference to the idea that curiosity helps maintain an intermediate level of arousal, providing the *sinusoid* action as one form of regularity between the most simple (*constant*) and the most complex (*random*) was a natural starting point in exploring the agent's possible reactions to variations in domain-delivered (extrinsic) reward.

3.4.1 Agent Design

The design of our agent was motivated by emphasizing simplicity and consistency in the inner workings of the agent while allowing for the computations for a variety of curiosity methods.

Many RL control algorithms rely on maintaining an estimate of the value of each action. For this reason, we included a prediction learner, which used the Sarsa (prediction) algorithm (Sutton and Barto, 2018, pp. 129–130) to estimate the action-value $Q(s, a)$ of each action a, given it is taken from state s, where the action-value estimates are with respect to the extrinsic reward. At each time step, t, given $S_t, A_t, R_t, S_{t+1}, A_{t+1}$ as described in Section 2.2, the prediction learner computed the *temporal-difference error* (TD error), δ_t, as

$$\delta_t \leftarrow R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t). \tag{2.7}$$

where γ is the discount factor as described in Section 2.2 (set to $\gamma = 0.9$ in this study).

Essentially, the TD error is the difference between our *predicted value*, $Q(S_t, A_t)$, of taking action A_t from state S_t, and our *sample value*, $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$, which combines our sample reward R_{t+1} and the estimated value of our sample action A_{t+1}, as taken from our sample state S_{t+1}.

Because there may be an element of randomness to the domain's rule for determining the next state and reward given the current state and action, we do not necessarily want to change our new estimated value to the sample value—we only move it towards that value, so the estimated value for action A_t from state S_t is then updated as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\delta_t \tag{3.1}$$

where α is the learning rate (kept constant at $\alpha = 0.1$ in our experiments). We initialized $Q(s, a)$ to 0 for all states and actions.

Since we are already computing a form of prediction error—the TD error—we can conveniently implement several intrinsic reward methods based around prediction error. For the purposes of this initial experiment, we limited the tested methods to several which computed a new *intrinsic reward* from the prediction error for each transition. This meant we could control the overall agent by simply using a control learner aiming to maximize cumulative future intrinsic reward (*intrinsic value*).

For control, we used ε-greedy Q-learning (Sutton and Barto, 1998, p. 140). Q-learning maintains estimates of the value (in the cases of our intrinsic reward methods, intrinsic value) of each action, assuming the agent will choose the action with the highest value in the next step. If we had perfect estimates of our values, we would want to always act *greedily*, taking the action with the highest value. But to learn and maintain our estimates of each action-value, the agent must try every action occasionally. With probability ε, the agent will choose a random action, but otherwise, it really does choose the action with the highest value (hence the name, ε-greedy). In this experiment, we set $\varepsilon = 0.1$.

Baseline Methods We selected one baseline method for this experiment.

(a) **Extrinsically Motivated Actor**, referring to an agent using the same control strategy as used with the intrinsic reward methods (ε-greedy Q-learning) to maximize the extrinsic reward (as opposed to an intrinsic reward).

Intrinsic Reward Methods We selected four intrinsic reward methods for our initial experiments. Each of these methods relies on prediction error as a component of the computation of its intrinsic reward.

(a) **Absolute prediction error**, referring to the absolute value of some measure of prediction error, was one of the earliest learning signals optimized specifically for curiosity, suggested by Schmidhuber (1991b). The initial intuition: to improve prediction, the agent should spend more time in areas of high error. Unfortunately, such an agent might get stuck repeatedly choosing areas that are highly unpredictable or random (Schmidhuber, 1991a, p. 2).

$$R^I_{t+1} \leftarrow |\delta_t| \tag{3.2}$$

(b) Schembri et al. (2007a) specifically aimed to maximize **(signed) TD-error in domain value**, as opposed to general prediction error. The intuitive benefit of maximizing TD error is that the agent's choices should favour areas which seem to be better than expected (and so afford positive TD error) and avoid

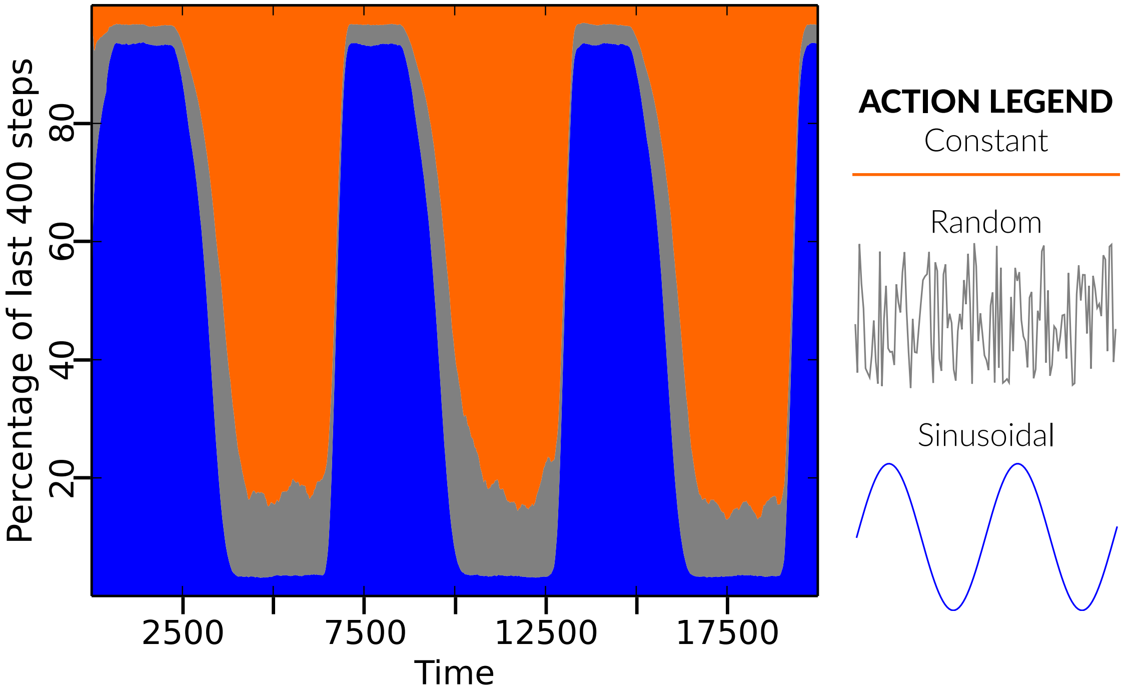


Figure 3.3: The Extrinsically Motivated Actor tracks the *sinusoidal* action, showing a greater preference for the *constant* action than the *random* action when the *sinusoial* action's rewards are negative. As noted in the text, the externally motivated actor is a typical RL agent: it uses standard Q-learning control to maximize accumulated discounted future extrinsic reward.

areas which seem to be getting worse (affording negative TD error) in terms of the usual RL goal of maximizing cumulative future extrinsic reward.

$$R^I_{t+1} \leftarrow \delta_t \tag{3.3}$$

(c) **Learning Progress**, as defined by Oudeyer et al. (2007), refers to the decrease in error over a recent window of time steps. Intuitively, an agent achieves high learning progress as its prediction improves.

I implemented Learning Progress in two different ways. The simpler implementation (for which results are shown in Figure 3.6) used a single shared buffer of recent squared errors: the buffer, of length $\eta + \tau$ holds the observed TD errors from $\delta_{t-\eta-\tau}$ to δ_t, where η is what Oudeyer et al. (2007, p. 271) call a *smoothing parameter*, which determines the size of the time windows that the squared errors are averaged (smoothed) over, and τ is what Oudeyer et al. (2007, p. 271) call a *time window* parameter, which determines the length of the time window between which the smoothed squared errors are compared. The intrinsic reward is then computed as follows:

$$R^I_{t+1} \leftarrow \frac{1}{\eta+1}\sum_{j=0}^{\eta}\delta^2_{t-j-\tau} - \frac{1}{\eta+1}\sum_{j=0}^{\eta}\delta^2_{t-j} \tag{3.4}$$

I also implemented Learning Progress with separate buffers for each action (results not shown). Using TD error is substantially different from the prediction error in next-state predictions using in the original formulation of IAC.

(d) **Unexpected Demon Error (UDE)** was designed by White et al. (2014) to measure the surprisingness of an observation and can be used as a curiosity reward. UDE is the ratio of a moving average of the prediction error and the square root of the variance in the prediction error. If the prediction error is consistently large then the samples have high variance, and we expect high error. If the error becomes larger than expected, something about the observation is surprising, and should be explored further.

$$R^I_{t+1} \leftarrow \frac{\mathrm{EWMA}(\delta)_t}{\sqrt{\mathrm{var}(\delta)_t}} \tag{3.5}$$

In this study,[1] the Exponentially Weighted Moving Average (EWMA) at time t of the TD error, $\mathrm{EWMA}(\delta)_t$, is initialized as the first TD error observed,

$$\mathrm{EWMA}(\delta)_0 \leftarrow \delta_0, \tag{3.6}$$

then computed incrementally at each time step as

$$\mathrm{EWMA}(\delta)_{t+1} \leftarrow \mathrm{EWMA}(\delta)_t + \alpha_{\mathrm{EWMA}} \cdot (\delta_t - \mathrm{EWMA}(\delta)_t), \tag{3.7}$$

and the sample variance at time t, of the TD error, $\mathrm{var}(\delta)_t$ is initialized to 0,

$$\mathrm{var}(\delta)_0 \leftarrow 0, \tag{3.8}$$

and then computed incrementally at each time step as

$$\mathrm{var}(\delta)_{t+1} \leftarrow (1 - \alpha_{\mathrm{EWMA}}) \left(\mathrm{var}(\delta)_t + \alpha_{\mathrm{EWMA}} \cdot (\delta_t - \mathrm{EWMA}(\delta)_t)^2\right) \tag{3.9}$$

where $0 < \alpha_{\mathrm{EWMA}} < 1$ is a weighting factor parameter. In these experiments, α_{EWMA} was set to 0.25. This algorithm for incrementally updating the Exponentially Weighted Moving Average (EWMA) and variance was provided by Finch (2009, p. 8).

3.4.2 Results and Conclusions

For comparison, in Figure 3.3 we show an illustration of the behaviour of an agent without a specific intrinsic reward method, simply aiming to maximize future domain reward. This agent is using the ε-greedy Q-learning control learner described

[1] In the paper introducing UDE, White et al. (2014) do not specify the algorithm used for computing the exponentially weighted average, so in this initial experiment we followed the algorithm provided by Finch (2009, pp. 7-8). However, White's thesis (2015) does specify the algorithm used (p. 121) and it differs from our choice. In the experiments in the second study (later in this chapter, in Section 3.5), White's algorithm is implemented as it was in the original experiments using White's choice of exponentially weighted average algorithm.

prediction learning algorithm	Sarsa	
control learning algorithm	Q-learning	
learning rate (applied in Eq. 3.1)	α	0.1
probability of taking a random action	ε	0.1
discount factor	γ	0.9
initial action values for all s, a	$Q(s, a)$	0
weighting factor for EWMA	α_{EWMA}	0.25

Table 3.1: Parameter settings for initial Curiosity Bandit experiment.

above, but it is estimating and making choices based on the domain reward. The plot shows, at each time step and for each action, the percentage of the last 400 actions (or up to 400 actions, for time steps earlier than 400) that the action was selected. Because the TD error is used to compute each IR, we initially thought that tracking the TD error might afford us some insight into the behaviour of the agent in relation to its computational intrinsic reward method. However, comparing the sample value to the predicted value provided better insight, as their difference is the TD error, but the trends in each vary, particularly depending on the *sinusoidal* reward signal. Interpretation of this comparison is described in the captions of Figures 3.4–3.7 and the sample value and the predicted value for the *constant* and *sinusoidal* actions are shown in the lower plot of each of those figures. Because the changes in the sample value and predicted value are subtle, but important for understanding the behaviour policy of the learner, these figures are presented in full-page format.

These initial experiments have already shown that we can more clearly discern differences in the behaviour motivated by different computational intrinsic reward methods by controlling other aspects of the experiment. We have initial insight into how regularities in the environment can impact the behaviour of agents motivated by different intrinsic reward methods.

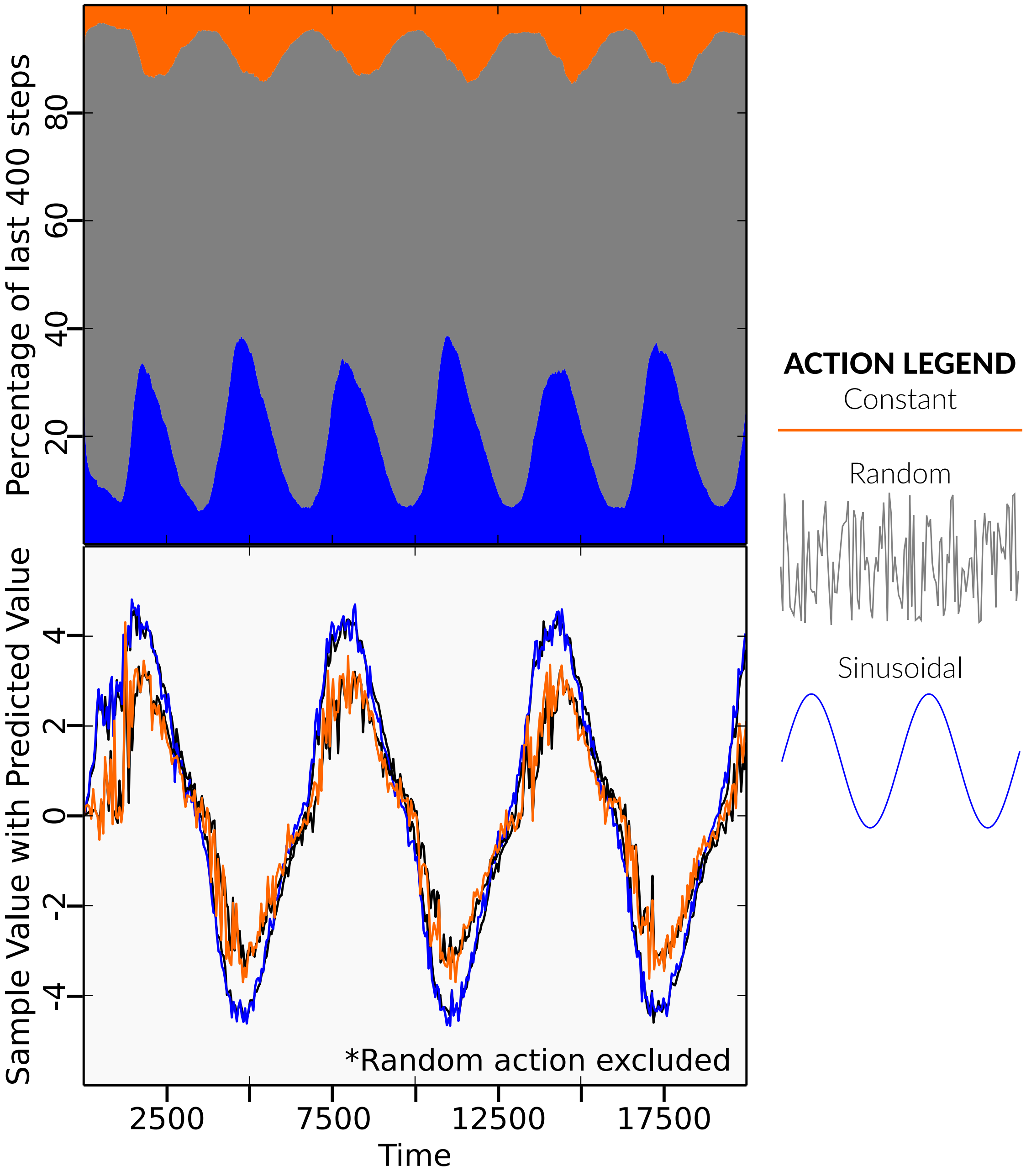


Figure 3.4: As predicted, when the agent is motivated by **absolute prediction error**, we observe an attraction to randomness. Though one might expect the sinusoidal action to be chosen most (hardest to predict) when it experiences the largest change, around zero, the sinusoidal action actually shows periodic spikes exactly at its crests and troughs. These spikes occur because α is small; it is at these peaks that the value estimate lags most.

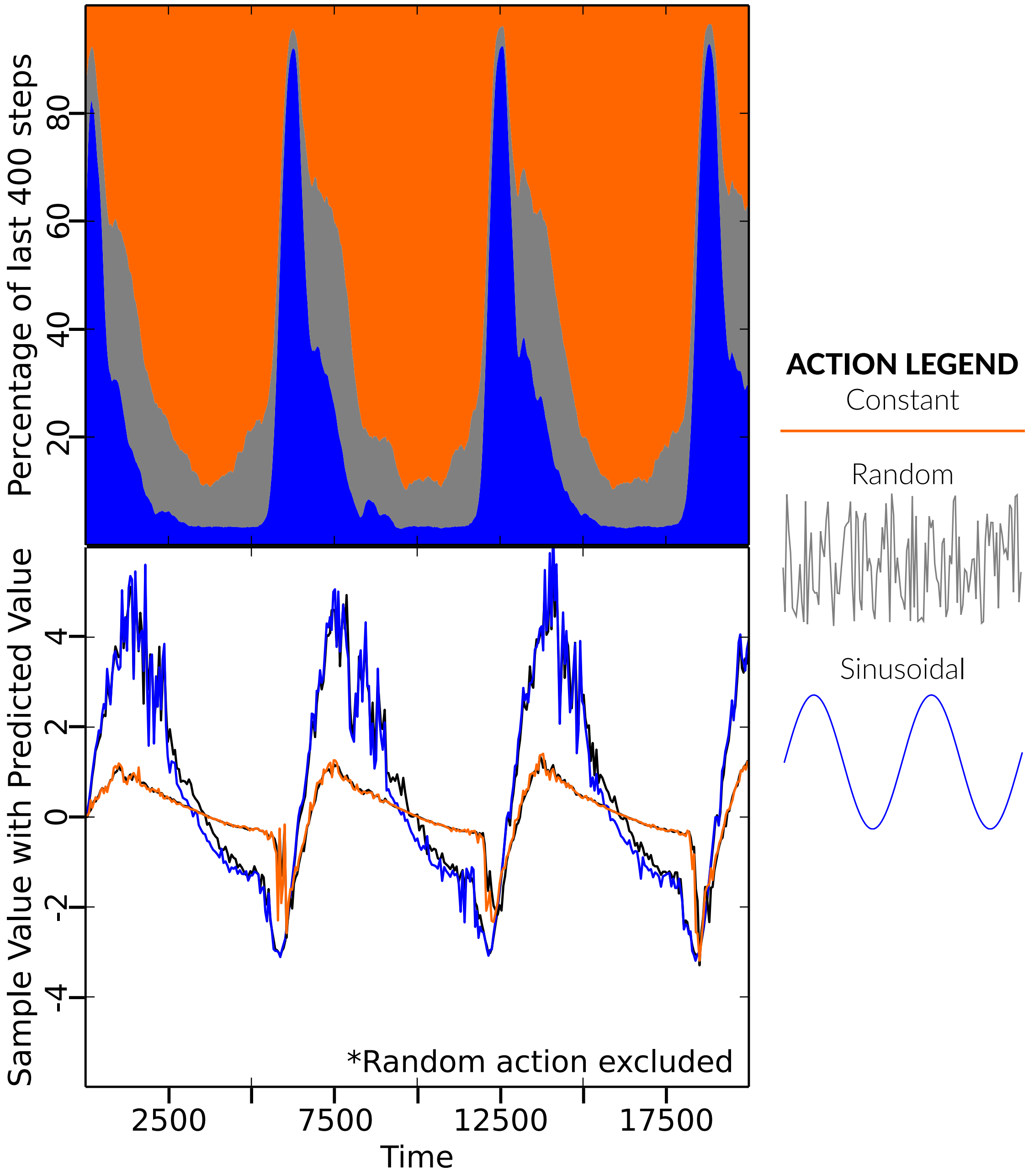


Figure 3.5: When the agent is aiming to maximize its **(signed) TD error**, it can guarantee the greatest error when it chooses the sinusoidal action at its crests, similar to the learner maximizing absolute error. In contrast, at the troughs for the sinusoidal action, the constant action is under-estimated. This under-estimation occurs because the prediction learner is taking the possibility of the sinusoidal action into account in its estimate. This results in the constant action giving the best error.

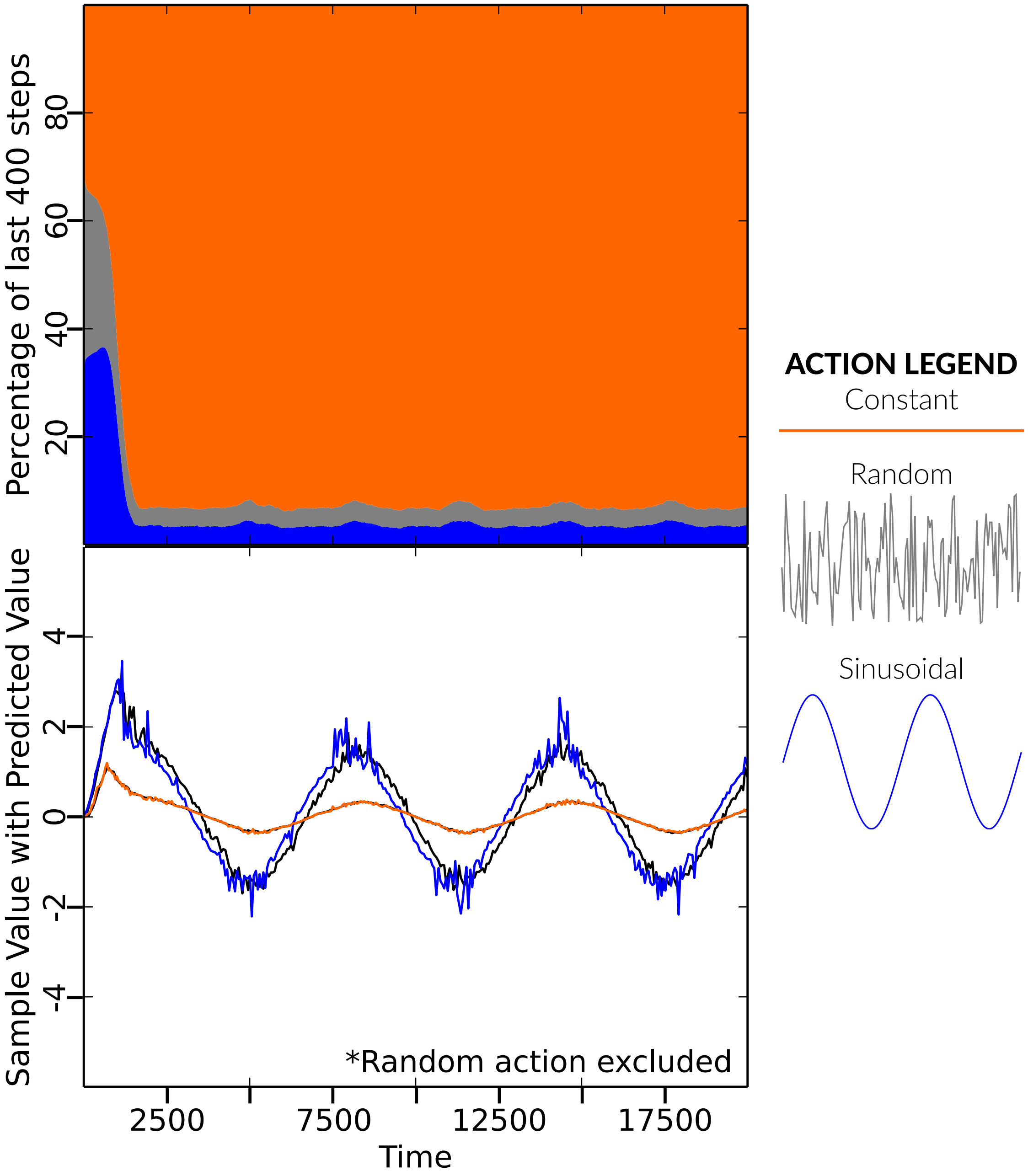


Figure 3.6: The behaviour of a control learner motivated by **learning progress**. In taking the constant action, the agent always shifts the value of the constant action closer to zero by increasingly smaller amounts. Therefore, the sinusoidal action only ever shows more Learning Progress at the crests and troughs, where the predicted and sample values cross, resulting in corresponding blips.

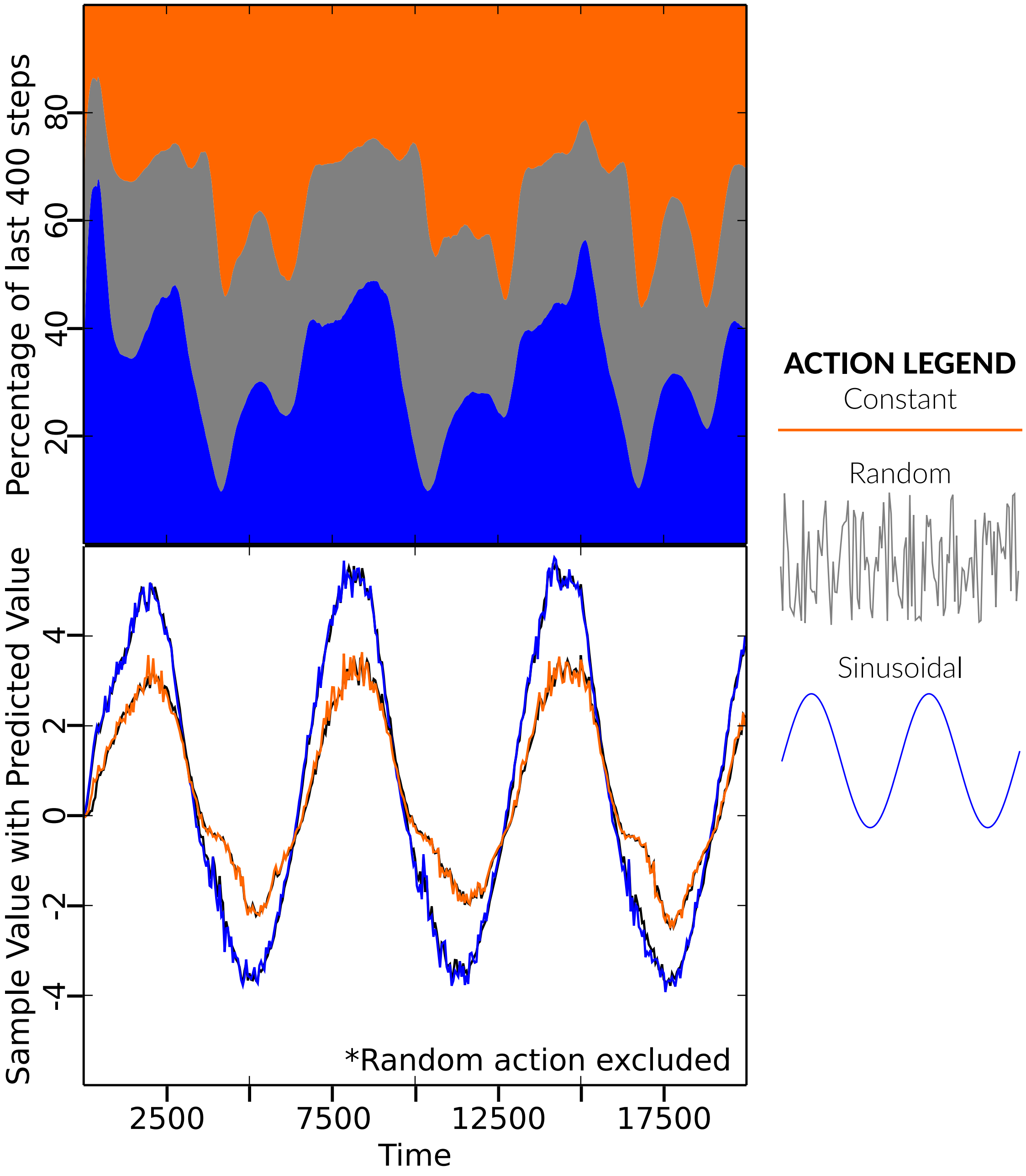


Figure 3.7: In the behaviour of an agent maximizing **Unexpected Demon Error** (UDE), we see two peaks of the constant action either side of each trough and two peaks of the sinusoidal action either side of each crest.

3.5 Second Study: Balancing Learning Many Things

Learning about many things can provide numerous benefits to a reinforcement learning system. For example, the UNsupervised REinforcement and Auxiliary Learning (UNREAL) and Intentional Unintentional Agent (IUA) architectures, designed by Jaderberg et al. (2017, pp. 1–3) and Cabi et al. (2017, p. 207), respectively, learn about many things, going above and beyond the more typical choice to learn a policy and value function for the extrinsic reward. In particular, they learn the optimal policies and value functions for a variety of GVFs (p. 1). Jaderberg et al. (2017) found that learning additional policies and value functions—which they called '*auxiliary tasks*'—improved their system's representation (p. 2–3). Learning GVFs for additional signals improved performance for both UNREAL and IUA (Jaderberg et al., 2017, p. 2; Cabi et al., 2017, p. 212). Similarly, the Hybrid Reward Architecture (HRA) designed by van Seijen et al. (2017) demonstrated state-of-the-art performance in problems in which the primary reward functions can be broken apart into components. Their design leverages learning a separate value function for each component and combining the results when determining behaviour (pp. 1–2, 8). Riedmiller et al. (2018) took the learning of numerous policies a step further, using a scheduler to decide which policy should be put to use at a given time (p. 4344). Along with learning a policy to maximize the primary reward associated with a practical robotic task, like stacking objects or tidying a table (p. 4345), the Scheduled Auxiliary Control (SAC-X) system designed by Riedmiller et al. (2018) learned multiple simple GVFs where the pseudo-rewards/cumulants are based on control of the robot's own sensory observations (p. 4344–4345). Scheduling these policies with a learned scheduler improved performance in the primary task, particularly by improving exploration. In all the examples above, a learning system updates a collection of GVFs (see Sutton et al., 2011) from a single stream of experience. The question we tackle in this study is how to sculpt that stream of experience—how to adapt the learning system's behaviour—when the system is learning a collection of value functions.

One option is to simply choose actions expected to maximize the environmental (extrinsic) reward. This was the approach explored by Jaderberg et al. (2017) with UNREAL and the simple addition of auxiliary learning problems shaping the learned representation resulted in significant performance improvements in challenging visual navigation problems. However, it is not hard to imagine situations where this approach would be limited. In general, the extrinsic reward may be delayed and sparse: what should the agent do in the absence of extrinsic motivation?

In their design of SAC-X, Riedmiller et al. (2018) suggested an answer to that question: the system scheduled and executed policies maximizing the GVFs for different cumulant signals to achieve exploration in the search for sparse external rewards (p. 4345, 4352). The learned scheduler, however, was designed to decide which policy to follow based on how the chosen policy contributes to maximizing the external reward (p. 4347). Similarly, Bagot et al. (2020) treated the policies associated with the maximization of different cumulants as options to be selected by a control learner aiming to maximize extrinsic reward (using Q-learning in their experiments).

Learning reusable knowledge such as skills (Sutton et al., 1999) or a model of the world might result in more long-term reward. Such auxiliary learning objectives could emerge automatically during learning (Silver et al., 2017). Most agent architectures, however, include explicit skill and model learning components. It seems natural that progress towards these auxiliary learning objectives could positively influence the agent's behaviour, resulting in improved learning overall.

Learning many value functions off-policy from a shared stream of experience—with function approximation in an unknown environment—provides a natural setting to investigate intrinsically motivated learning without extrinsic rewards. The basic idea is simple. The aim is to accurately estimate many value functions independently. Directly optimizing the data collection for all estimations jointly is difficult because we cannot directly measure this total learning objective and because actions have an indirect impact on learning efficiency. There is a large related literature in active learning (Cohn et al., 1996; Balcan et al., 2009; Settles, 2012; Golovin and Krause, 2011; Konyushkova et al., 2017) and active perception (Bajcsy et al., 2018), from which to draw inspiration for a solution but which do not directly apply to this problem. In active learning the agent must sub-select from a larger set of items to choose which points to label. Active perception is a subfield of vision and robotics. Much of the work in active perception has focused on specific settings—namely visual attention (Bylinskii et al., 2015), localization in robotics (Patten et al., 2018) and sensor selection (Satsangi et al., 2018, 2020)—or assumes knowledge of the dynamics of the world (see Bajcsy et al., 2018).

We can instead formulate our task as a reinforcement learning problem. We can use an intrinsic reward, internal to the learning system, that approximates the total learning across all learners. Behaviour can be adapted to choose actions that maximize the accumulation of intrinsic reward, towards the goal of maximizing the total learning of the system. The choice of intrinsic rewards can have a significant impact on the sample efficiency of such intrinsically motivated learning systems. This study provides the first formulation of parallel value function learning as a reinforcement learning task. Fortunately, there are many ideas from related areas that can inform our choice of intrinsic rewards.

Rewards computed from internal statistics about the learning process have been explored in many contexts over the years. Intrinsic rewards have been shown to induce behaviour that resembles the development stages exhibited by young humans and animals (Barto, 2013; Singh et al., 2004; Oudeyer et al., 2007; Lopes et al., 2012; Haber et al., 2018). Internal measures of learning have been used to improve skill or option learning (Singh et al., 2004; Schembri et al., 2007b; Barto and Şimşek, 2005; Santucci et al., 2013a; Vigorito, 2016), and model learning (Schmidhuber, 1991b, 2008). Most recent work has investigated using intrinsic reward as a bonus to encourage additional exploration in single task learning (Itti and Baldi, 2006; Stadie et al., 2015; Bellemare et al., 2016; Pathak et al., 2017; Hester and Stone, 2017; Tang et al., 2017; Andrychowicz et al., 2017; Achiam and Sastry, 2017; Martin et al., 2017; Colas et al., 2018; Schossau et al., 2016; Pathak et al., 2019). Few have investigated the impact of making these internal measures the main objective of learning (Berseth et al., 2021), however previous studies have noted that intrinsic reward is useful even in single-task problems with a well-defined external goal (Bellemare et al., 2016).

It remains unclear, however, which of these measures of learning would work best in our no-reward setting. Most prior work has focused on providing demonstrations of the utility of particular intrinsic reward mechanisms. One study, by Burda et al. (2019a), was on a large scale, but focused on a suite of complex control domains, holding the intrinsic reward constant (p. 1). In their experiments, Burda et al. (2019a) varied the feature space used for the learner's representation of its environment (pp. 3). For some of the experiments in that study, Burda et al. (2019a) trained their learner with pure intrinsic reward (p. 5), while in others they used the intrinsic reward as an exploration bonus (p. 9). In each of these experiments, Burda et al. (2019a, pp. 5–7) measured performance either based on accumulation of an external reward associated with the domain of interest (for example, in-game reward in Arcade Learning Environment video games) or with respect to a metric that was meaningful with respect to the environment (for example, count of ball bounces in a juggling domain).

Another large study, by Graves et al. (2017), on the other hand, compared eight different IRs, and so is more closely related to this work. In their work, their goal was to choose an intrinsic reward to accelerate the learning of a neural network, by using that IR to guide the system's choices about "which task to study next" (p. 1). has been conducted on Learning Progress measures for curriculum learning for neural networks (Graves et al., 2017), where the goal is to learn from which task to sample a dataset to update the parameters. Variants of their measures are related to the intrinsic rewards explored in this study, but their setting differs substantially in that learning is offline from batch supervised learning datasets and the underlying problems are stationary. To the best of our knowledge, there has never been a broad empirical comparison of intrinsic rewards for the online multi-prediction setting with non-stationary targets.

A computational study of intrinsic rewards is certainly needed, and we believe that insight can be gained by sidestepping function approximation and off-policy updating, at least to start. Estimating multiple value functions in parallel requires off-policy algorithms because each value function is conditioned on a policy that is different than the exploratory behaviour used to select actions. In problems of moderate complexity, these off-policy updates can introduce significant technical challenges. Popular off-policy algorithms like Q-learning and V-trace can diverge with function approximation (Sutton and Barto, 2018). Sound off-policy algorithms exist, but require tuning additional parameters and are relatively understudied in practice. Even in tabular problems, good performance requires tuning the parameters of each component of the learning system—a complication that escalates with the number of value functions. Finally, the agent must solve the primary exploration problem in order to make use of intrinsic rewards. Finding states with high intrinsic reward may not be easy, even if we assume the intrinsic reward is reliable and informative. To avoid these many confounding factors, the right place to start is in a simpler setting.

In this study, we investigate and compare different intrinsic reward mechanisms in a Curiosity Bandit designed as a parallel learning testbed. The testbed consists of a single state and multiple actions. Each action is associated with an independent scalar target to be estimated by an independent prediction learner. An ideal behaviour policy will focus on actions that generate the most learning across the prediction learners. However, the overall task is partially observable, and learning is never done. The targets change without any explicit notification to the agent, and the task continually changes due to changes in action selection and learning of the individual prediction learners. Different configurations of the target distributions can simulate unlearnable targets, non-stationary targets, and easy-to-predict targets. This new testbed provides a simple instantiation of a problem where *introspective* learners should help achieve low overall error. An introspective prediction learner is one that can autonomously increase its rate of learning when progress is possible, and decrease learning when progress is not—or cannot—be made.

This study summarizes a comprehensive empirical comparison of different intrinsic reward mechanisms, including several ideas from reinforcement learning and active learning. This study helps demonstrate the versatility of the Curiosity Bandit family, focusing on accurately learning explicitly non-reward signals for one potential quantitative evaluation of intrinsic rewards. In addition, this computational study highlighted a simple principle: intrinsic rewards based on the *amount of learning* (e.g., Bayesian Surprise and simple change in weights) can generate useful behaviour if each individual learner is introspective. Across a variety of problem settings we found that the combination of introspective learners and simple intrinsic rewards was most reliable, performant, and easy to tune. We conclude the description of this study with a discussion about how these ideas could be extended beyond our one-state prediction problem to drive behaviour in large-scale problems where off-policy learning and function approximation are required.

3.5.1 Problem Formulation

In this section we formalize a testbed for comparing intrinsic rewards using a stateless prediction task and independent learners. This formalism is meant to simplify the study of balancing the needs of many learners and facilitate comprehensive comparisons.

We formalize our multiple-prediction learning setting as a collection of independent, online supervised learning tasks. On each discrete time step $t = 1, 2, 3, ...$, the *control learner* selects an action $A_t \in \{1, \ldots, N\}$. Each action corresponds[2] one-

[2] To clearly separate the action selected by the agent and the prediction task, we use A_t to denote the action selected at time t and we use i to denote the prediction task. A_t is uppercase to indicate it is a random variable. In our setting, there is an equivalence between taking an

to-one to a task $i \in \{1, \ldots, N\}$. Choosing the action associated with task i lets the agent sample a target signal, $C_{t,i}$, with distribution, $\theta(t, i)$; that is, $C_{t,i} \sim \theta(t, i)$. This distribution, $\theta(t, i)$, is indexed by time to reflect that it can change on each time step; this enables a wide range of different target distributions to be considered, to model a non-stationary, multi-prediction learning setting. We define the particular distributions we use in our experiments later in this section, in Equation (3.12).

Associated with each prediction task, i, is a simple *prediction learner* that maintains a real-valued vector of weights to produce an estimate, $\hat{C}_{t,i} \in \mathbb{R}$, of the expected value of the target; that is, $\hat{C}_{t,i} \approx \mathbb{E}[C_{t,i}]$. The vector of weights could be updated using any standard learning algorithm at each time step its associated action/task is selected. In this work, we use a 1-dimensional weight vector, with the current weights for task i at time t denoted $w_{t,i}$ and so the update is a simple delta-rule/least-mean-squares (LMS) learning update:

$$w_{t+1,i} \leftarrow w_{t,i} + \alpha_{t,i}\delta_{t,i} \tag{3.10}$$

where $\alpha_{t,i}$ is a scalar step-size parameter and $\delta_{t,i} \coloneqq C_{t,i} - w_{t,i}$ is the prediction error of prediction learner i on step t. On a step where task i is not selected, $w_{t,i}$ is not updated, implicitly setting $w_{t+1,i}$ to $w_{t,i}$.

In alignment with the idea that, at any given time step, we want our learner to have an accurate estimate of every signal, we define the primary goal as the minimization of the Mean Squared Error (MSE) across both time and tasks. We define the MSE at time step t as

$$\texttt{MSE}(t) \coloneqq \frac{1}{N}\sum_{i=1}^{N}(\hat{C}_{t,i} - \mathbb{E}[C_{t,i}])^2. \tag{3.11}$$

The control learner does not observe the MSE, nor sufficient information to compute it; the control learner only observes an intrinsic reward signal on each time

action and observing data for its corresponding task i, and so there is an equivalence between the actions and tasks. More generally, such as in the full reinforcement learning setting, this is not the case; for extensions on this work, it is useful to clearly delineate between actions and prediction tasks.

step. Each prediction learner, on the other hand, observes one of the targets $C_{t,i}$ on each time step and the target signal observation is a noisy sample of the signal's true expected value, $\mathbb{E}\left[C_{t,i}\right]$, which, as noted above, $\hat{C}_{t,i}$ is meant to estimate.

Our problem can be naturally formulated as a sequential decision-making problem, where on each time step t, the control learner chooses an action A_t corresponding to a task i, resulting in a new sample, $C_{t,i}$, which the learner can use to update the associated weight, $w_{t,i}$. To design an agent to minimize the MSE, we must devise a way to choose which prediction task to sample at each time step. In this work, we choose to associate an intrinsic reward $R_t^I \in \mathbb{R}$ with the action choice at each time step as a basis for which to learn a preference over actions. In this work, we investigate different intrinsic rewards. Given a definition of an intrinsic reward, we can use a bandit algorithm suitable for non-stationary problems; our experiments included two such algorithms, but for the purposes of this dissertation we focus on one, as discussed in Section 3.5.1.1.

The targets for each prediction learner are intended to simulate the dynamics of targets that a real-world system learning parallel auxiliary tasks might experience, such as sensor values of a robot. To simulate a range of interesting dynamics, we construct the distribution for a target signal at a given time step, $\theta(t,i)$, as a normal (or equivalently, Gaussian) distribution with a mean that drifts over time:

$$\begin{aligned} \theta(t,i) &\coloneqq \mathcal{N}(\mu_{t,i}, \sigma_i^2) \\ &\text{for} \quad \mu_{t+1,i} \leftarrow \Pi_{[x,y]}\left(\mu_{t,i} + D_{t,i}\right) \\ &\text{for} \quad D_{t,i} \sim \mathcal{N}(0, \xi_i^2) \end{aligned} \tag{3.12}$$

where $\mu_{0,i}, x, y \in \mathbb{R}$, $\sigma_i^2, \xi_i^2 \in \mathbb{R}_{>0}$, are parameters of the problem that must be set as part of experimental setup. For visual depictions of example signals, see Figure 3.9, which shows the target data generated by one run of the problem in our experiments. The symbols $\mu_{t,i}$ and σ_i^2, respectively, denote the mean and variance of the target distribution $\theta(t,i)$. At each time step, the mean drifts by a stochastic amount represented by the random variable $D_{t,i}$. Since the distribution of the amount of drift, $D_{t,i}$, is normally distributed around zero, the parameter ξ_i^2 controls the rate of drift. Similarly, the parameter σ_i^2 controls the amount of noise in the samples. The function $\Pi_{[x,y]}$ is a projection function that bounds the input to the range $[x, y]$:

$$\Pi_{[x,y]}(z) := \begin{cases} x & \text{for } z < x \\ z & \text{for } z \in [x, y] \\ y & \text{for } x > y \end{cases} \tag{3.13}$$

In this way, $\Pi_{[x,y]}$ projects the drifting $\mu_{t,i}$ back to the range $[x, y]$ to keep it bounded. Note that $\mu_{t,i}$ is updated on each time step t *regardless* of which action is selected.

The two variance parameters, σ_i^2 and ξ_i^2, which control the target signal sample variance variance and the amount of drift, respectively, are indexed by t because we explore experimental settings where they change. These changes are not communicated to the control learner, and the individual prediction learners are prevented from storing explicit histories of the targets. The purpose of this choice was to simulate the partial observability common in many large-scale systems (e.g., Sutton et al., 2011; Modayil et al., 2014; Jaderberg et al., 2017; Silver et al., 2017). Given our setup, the learning tasks for both the control learner and the prediction learners are best treated as non-stationary, so we can expect better performance (lower MSE) from algorithms that track in comparison to algorithms that converge (Sutton et al., 2007), as long as ξ_i^2 is greater than zero. Our formalism is summarized in Figure 3.8.

3.5.1.1 Non-stationary Bandit Algorithms for Prediction Learning

This work is not focused on the formalism of bandits itself, nor bandit algorithms. Rather, our goal is to investigate *intrinsic rewards* and their utility for learning multiple predictions. In particular, we aim to complete this investigation in the simplest setting in which we can obtain meaningful insights: a bandit-like setting. Our choice of bandit algorithm, therefore, is simply to facilitate this investigation, rather than for the purpose of investigating the properties of the bandit algorithms

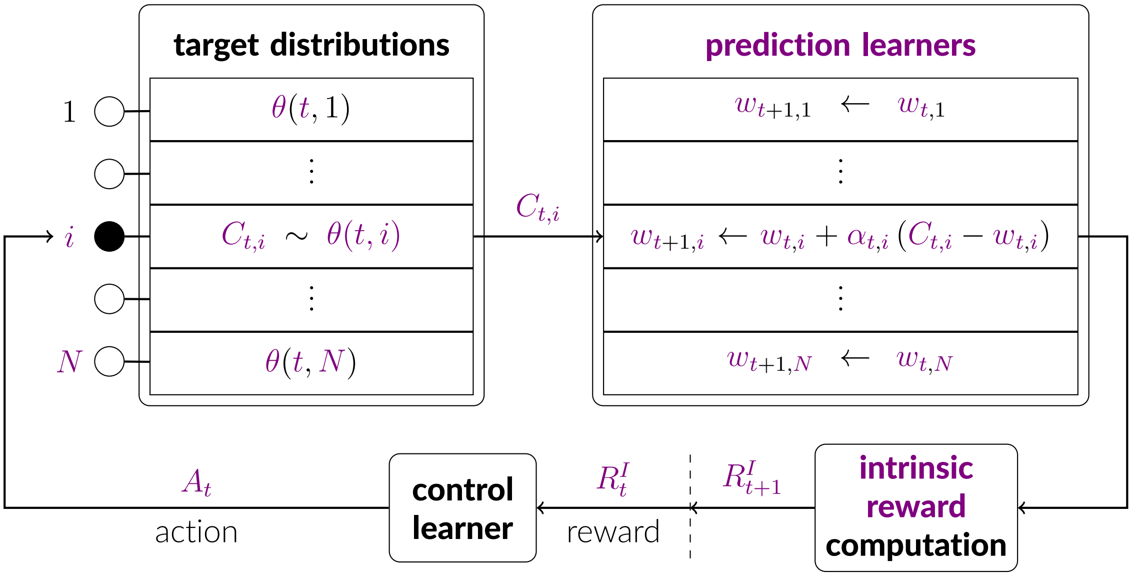

target distributions
1
i
N
$\theta(t,1)$
$\vdots$
$C_{t,i} \sim \theta(t,i)$
$\vdots$
$\theta(t,N)$
$C_{t,i}$
prediction learners
$w_{t+1,1} \leftarrow w_{t,1}$
$\vdots$
$w_{t+1,i} \leftarrow w_{t,i} + \alpha_{t,i}\,(C_{t,i} - w_{t,i})$
$\vdots$
$w_{t+1,N} \leftarrow w_{t,N}$
A_t
action
control learner
R^I_t
reward
R^I_{t+1}
intrinsic reward computation

Figure 3.8: Our parallel multi-prediction learning formulation.

themselves. To ensure our conclusions are not due primarily to the choice of bandit algorithm, we performed experiments applying two different bandit algorithms. We chose a gradient bandit algorithm (Sutton and Barto, 2018, p. 37–40) and an extension of Dynamic Thompson Sampling (DTS) (Gupta et al., 2011). In this dissertation we will focus only on the gradient bandit algorithm for simplicity, but more detail about the results with DTS is provided by Linke et al. (2020, pp. 1317–1319), but the differences between the results are minimal between the gradient bandit algorithm and DTS. Below, we describe the gradient bandit algorithm and briefly speak to why we chose it.

Our learning setting is not a typical multi-armed bandit problem; rather, it may be better considered a non-stationary partial monitoring problem where we care about pure exploration with anytime evaluation. Learners in typical bandit problems receive the loss of the chosen action as feedback at each step (Lattimore and Szepesvári, 2020, Ch. 37); *partial monitoring*, on the other hand, extends the multi-armed bandit framework to include problems "where the loss is not directly observed by the learner" (Lattimore and Szepesvári, 2019, p. 1). In our setting, the loss, or cost, at each time step is the MSE across tasks, $\texttt{MSE}(t)$ (Equation 3.11). Our setting presents a partial monitoring problem because, rather than directly observing the cost, the control learner only observes an intrinsic reward. Our setting is also non-stationary: the underlying target signals, from which the MSE is computed, are non-stationary.

We care about anytime evaluation because we are aiming for the prediction learners to *always* have estimates that are as accurate as possible. Anytime evaluation means that the agent needs to carefully choose when it selects a given arm, unlike in some pure exploration problems, where evaluation only occurs after some number of actions are taken (in the stationary case, one-time—as opposed to anytime—evaluation can essentially mean that the order the actions are taken doesn't really matter). These factors can be expected for real-world learners deciding what to observe in the world, especially if we want these learners to be prepared for unexpected situations where sufficiently accurate predictions allow them to behave appropriately.

Fortunately, for our prediction setting, the structure of our problem (described in Section 3.5.1) admits a simple approach that performs well in practice: to err on the side of taking an action periodically. There is no action selection which is detrimental, as it provides information about one of the targets. Particularly in a non-stationary setting, each action should be taken periodically, to check if expected reward estimates remain accurate. One reasonable strategy is to obtain a distribution over the actions—not find the single best action—and sample proportionally to that distribution, as is done by the gradient bandit algorithm.

The gradient bandit algorithm specified by Sutton and Barto (2018, Section 2.8) attempts to maximize the expected average reward by modifying a vector of action preferences $H_t \in \mathbb{R}^N$—indexed by action and incremented at each time step—based on the difference between the reward and average reward baseline:

$$H_{t+1}(a) \leftarrow \begin{cases} H_t(a) + \alpha(R_{t+1} - \bar{R}_t)(1 - \pi_t(a)) & \text{if } A_t = a; \\ H_\alpha(a) - \alpha(R_{t+1} - \bar{R}_t)\pi_t(a) & \text{otherwise.} \end{cases}$$

where $\bar{R}_t \in \mathbb{R}$ is the average of all the rewards up to time t, maintained using an unbiased exponential average (Sutton and Barto, 2018, Eq. 2.9), and $\bar{R}_t$ and $H_0(a)$ are both initialized to zero. Actions are selected probabilistically according to a softmax distribution which converts the preferences to probabilities:

$$\Pr\{A_t = a\} = \pi_t(a) := \frac{e^{H_t(a)}}{\sum_{b=1}^{N} e^{H_t(b)}}$$

The gradient bandit algorithm will sample all the actions infinitely often, though if an action preference is very low then that action will rarely be taken. It may be helpful for the reader to note that the gradient bandit algorithm is similar to policy gradient methods in reinforcement learning.

3.5.2 Simulating Parallel Prediction Problems

We considered several prediction problems as described by Linke et al. (2020), varying the problems by changing the the target signal variance, σ_i^2, and the drift variance, ξ_i^2. In this dissertation, we focus on a single problem setting with task distributions $\theta(t, i)$ set as defined in Equation (3.12). Figure 3.9 shows target data simulated for one run from each problem setting.

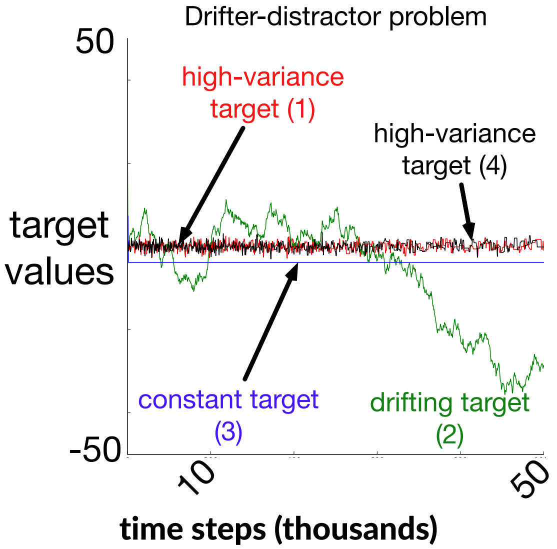


Figure 3.9: This figure shows the the target data generated in one run of the Drifter-Distractor problem.

target type	$\mu_{0,i}$	σ_i^2	ξ_i^2
constant	$\sim U(x,y)$	0	0
distractor	0	1	0
drifter	0	0	0.1

Table 3.2: These parameters define each target distribution used in the *Drifter-Distractor* problem. The parameter $\mu_{0,i}$ specifies the initial mean of each target, σ_i^2 is the sampling variance, and ξ_i^2 is the drift variance.

The *Drifter-Distractor* problem has four actions and, correspondingly, four target signals: (1) two (stationary) high-variance targets as *distractors* (2) a slowly *drifting* target and (3) a *constant* target, with ξ_i^2 and σ_i^2 for each of these types in Table 3.2. A distractor target is simply a noisy stationary target: the variance is high enough such that an agent might oversample the target even after the mean estimate is accurate. This is inspired by what Burda et al. (2019b) call the noisy-TV problem (p. 3; Burda et al., 2019a, p. 10), where forms of motivation designed to encourage learners to make observations where their models need improvement can also lead them to focus on natural sources of randomness, like a TV where the channel flips randomly, earlier articulated by Schmidhuber (1991a, p. 1460; 2008, p. 58).

3.5.3 Introspective Prediction Learners

The behaviour of a learning system that maximizes intrinsic rewards relies on the underlying prediction-learning algorithms as well as the definition of the intrinsic reward. In this section we introduce a distinction between two categories of learners, for which behaviour can be substantially different: introspective and non-introspective learners. We consider a learner to be **introspective** if the algorithm can modulate its own learning without help from an external process. More concretely, an introspective learner stops updating if it cannot improve. For example, in the case of prediction learning, an introspective learner would regulate its updates to mitigate noise in its prediction targets. A **non-introspective learner**, on the other hand, will continually update regardless of whether or not the incoming data is helping it make progress.

We experiment with both non-introspective and introspective learners for the prediction learners in our multi-prediction problem. We use basic LMS learners with a constant step-size parameter as our non-introspective learner. With a constant step-size parameter, the LMS algorithm will always try to adapt its estimates toward the sample targets on each time step. It does not matter if the target exhibits high variance or if the target is actually constant; the LMS algorithm will continue to adapt its estimates attempting to track each target in the online setting. Consider how a constant global step-size parameter would work on our Drifter-Distractor Problem discussed above. If the step-size parameter value is too large for the distractor target, then the prediction learner will continually make large updates due to the sampling variance, never converging to low error. If the step-size parameter is too small for the tracking target, then the prediction learner's estimate will often lag, causing high-error. A constant global step-size parameter cannot balance the need to track the drifter targets, and the need to learn slowly on the distractor targets.

To create a simple introspective learner for our setting, we simply combine our LMS predictors with a step-size adaption method called Autostep. Autostep is a simple meta-learning algorithm that adapts the step-size parameter of each LMS learner over time (Mahmood et al., 2012). The basic idea behind Autostep is to increase the step-size parameter when learning is progressing, and lower the step-size parameter value when learning is not progressing. It does so by keeping a trace, $h \in \mathbb{R}$, of the previous prediction errors. Roughly speaking, if the error changes sign often then the predictions are not improving and the step-size parameter value should be lowered. If the error is mostly of the same sign, then the step-size parameter value should not be reduced. Autostep has one key hyper-parameter, the meta learning-rate, κ: this controls how quickly the algorithm changes the step-size parameter (α). The full pseudocode, specialized to our stateless tracking tasks, is given in Algorithm 1.[3] Note that Autostep changes the step-size parameter with a multiplicative exponential, which allows geometric or rapid changes to the LMS learners step-size parameter.

Algorithm 1 : *The Autostep algorithm specialized to stateless prediction*
κ is the meta learning-rate parameter
n and h are scalar memory variables initialized to 1 and 0
δ is the prediction error and α_i (initialized to 1.0) the step-size parameter of predictor i

```
1: procedure AUTOSTEP($\delta$)
2:     $n \leftarrow \max(|\delta h|, n + \frac{1}{10000}\alpha_i(|\delta h| - n))$
3:     $\alpha_i \leftarrow \min(\alpha_i \exp(\kappa \frac{\delta h}{n}),\ 0.5)$
4:     $h \leftarrow h(1 - \alpha_i) + \alpha_i \delta$
```

To give some intuition about how Autostep changes the step-size parameter, consider what happens when we apply it to the Drifter-Distractor Problem in Figure 3.10. Here we simply plot α over time for four LMS learners—one for each target—with each step-size parameter adapted by Autostep. We used the gradient bandit algorithm and Weight Change reward[4] to generate the behaviour. The initial α of each LMS learner were set to one. The lines for the constant target (blue) and drifter target (green) are overlapping, and the lines for the distractor targets (red and black) are overlapping. Autostep progressively decreases α for the distractor targets, as the updates oscillate around zero. The update magnitude (or error) for the constant target goes to zero, and so Autostep stops changing α. This makes sense: why change the α if the prediction is perfect. Autostep keeps the α high for the learner estimating the drifter target, because continual progress is possible. On each time step the LMS learner moves its estimate towards the recent sample and most of these updates are in the same direction, at least over a recent window of time. In terms of prediction performance, Autostep significantly

[3] Our implementation of Autostep clips the step-size in step 3, given by Degris and White (2020), and so differs slightly from the form given by Mahmood et al. (2012).

[4] The details of the intrinsic reward function used to generate the data do not matter for the purpose of this experiment. Nevertheless, the Weight Change reward will be defined in the next section.

improves tracking, enabling different update rates for different prediction learners and reducing α on unlearnable targets or noisy targets once learning is complete—as you will see in the experiments included in this study.

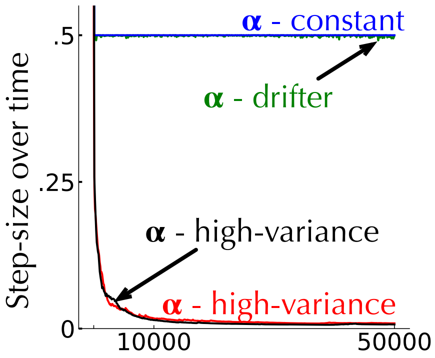


Figure 3.10: Sample run showing how Autostep changes the step-size parameters (α) over time with Weight Change reward. The lines for the constant target (blue) and drifter target (green) are overlapping, and the lines for the distractor targets (red and black) are overlapping.

We experimented with other step-size adaption methods, including ADADELTA (Zeiler, 2012) and RMSProp (Hinton et al., 2012), but the results were qualitatively similar. In this study we chose Autostep because (a) it was specifically designed for non-stationary, incremental, online tracking tasks like ours, (b) it uses a simple and easy to interpret update rule, and (c) there is a long literature demonstrating the practical utility of this method dating back to its origins in the Incremental Delta-Bar-Delta (IDBD) method (Sutton, 1992).

The choice of using meta-learning to obtain introspective learners not only works well in our multi-prediction tasks, but also should scale to larger tasks with function approximation in future work. Step-size adaption methods like Adam and RMSProp can speed up training in neural networks and make learning more robust to non-stationarity. In online reinforcement learning, extensions of Autostep have shown to improve prediction and control performance with function approximation (Kearney et al., 2018; Günther et al., 2020). We discuss these extensions and how our results go beyond stateless tracking at the end of the study. It is worth emphasizing that introspective learners are not optimal learners and that they are in fact the most common type of agent used in deep reinforcement learning. The main criterion is that an introspective learner should regulate its own updates based on an internal measure of improvement.

3.5.4 Intrinsic Rewards for Multi-prediction Learning

Many learning systems draw inspiration from the exploratory behaviour of humans and animals, uncertainty reduction in active learning, and information theory—and the resulting techniques could all be packed into the suitcase of curiosity and intrinsic motivation. In an attempt to distill the key ideas and perform a meaningful yet inclusive empirical study, we consider only methods applicable to our problem formulation of multi-prediction learning. Although few approaches have been suggested for off-policy multi-task reinforcement learning—approaches by Barto et al. (2004) and White et al. (2014) as notable exceptions—many existing approaches can be used to generate intrinsic rewards for multiple, independent prediction learners (see the excellent summary by Barto, 2013). We first summarize methods we evaluate in our empirical study. The specific form of each intrinsic reward discussed below is given in Table 3.3, with italicized names below corresponding to the entries in the table. We conclude by mentioning several rewards we did not evaluate, and why.

Reward Name	$R^I_{t,i}$
Absolute Value of Learning Progress (Oudeyer et al., 2007)	$\left\lvert \frac{1}{\eta+1}\sum_{j=0}^{\eta}\delta^2_{t-j-\tau,i} - \frac{1}{\eta+1}\sum_{j=0}^{\eta}\delta^2_{t-j,i} \right\rvert$
Parameter η specifies the length of the window and parameter τ the amount of overlap; $\tau \leq \eta < t$	
Expected Error (see explanation on p. 85)	$\left\lvert \overline{\delta_{t,i}}^{\beta} \right\rvert$
The exponentially weighted average, $\overline{\delta_{t,i}}^{\beta}$, is incrementally computed at each step as $\overline{\delta_{t,i}}^{\beta} \leftarrow (1-\beta)\overline{\delta_{t-1,i}}^{\beta} + \beta\delta_{t,i}$, where $\beta \in (0,1)$ is a parameter.	

Step-size Change
(see explanation on p. 103)

$$|\alpha_{t-1,i} - \alpha_{t,i}|$$

Error Reduction
(see explanation on p. 85;
inspired by Schmidhuber, 1991a)

$$|\delta_{t-1,i}| - |\delta_{t,i}|$$

Squared Error
(Gordon and Ahissar, 2011)

$$\delta_{t,i}^2$$

Bayesian Surprise
(Itti and Baldi, 2006)

$$\frac{1}{2}\log_2\left(\frac{\nu_{t,i}}{\nu_{t-1,i}}\right) + \frac{\nu_{t-1,i} + (\hat{C}_{t-1,i} - \hat{C}_{t,i})^2}{2\nu_{t,i}} - \frac{1}{2}$$

An estimate, $\nu_{t,i}^{(y)}$, of the target variance, $\mathrm{var}[C_{t,i}]$, is obtained using an exponential average variant of Welford's algorithm: $\nu_{t,i}^{(y)} \leftarrow (1-\beta)\nu_{t-1,i}^{(y)} + \beta(C_{t,i} - \hat{C}_{t-1,i})(C_{t,i} - \hat{C}_{t,i})$ from which the posterior variance estimate, $\nu_{t,i}$, is computed as $\nu_{t,i} = \max(\nu_{t,i}^{(y)}/d_t, 10^{-3})$ where $d_t = (1-\beta)d_{t-1} + 1$.

Unexpected Demon Error
(White et al., 2014; White, 2015)

$$\left|\frac{\overline{\delta_{t,i}}^{\beta}}{\sqrt{\mathrm{var}[\delta_{t,i}] + c}}\right|$$

Here c is a small constant, set to 10^{-6} in our experiments, and $\mathrm{var}[\delta_{t,i}]$ is a sample variance computed using the mean computed with an exponentially-weighted average.

Variance of Prediction
(see explanation on p. 88)

$$\hat{\nu}_{t,i}$$

Variance of Prediction, $\hat{\nu}_{t,i}$, denotes an estimate of $\mathrm{var}[\hat{C}_{t,i}]$, the variance in the estimate of the target signal, computed using an exponentially-weighted average variant of Welford's algorithm:

$$\hat{\nu}_{t,i} \leftarrow (1-\beta)\hat{\nu}_{t-1,i} + \beta\left(\hat{C}_{t,i} - \overline{\hat{C}_{t-1,i}}^{\beta}\right)\left(\hat{C}_{t,i} - \overline{\hat{C}_{t,i}}^{\beta}\right) \tag{3.14}$$

for $0 < \beta < 1$, with $\overline{\hat{C}_{t,i}}^{\beta} = (1-\beta)\overline{\hat{C}_{t-1,i}}^{\beta} + \beta\hat{C}_{t,i}$ the exponentially-weighted average of the predictions for task i.

Uncertainty Change
(see explanation on p. 88)

$$|\hat{\nu}_{t-1,i} - \hat{\nu}_{t,i}|$$

Variance of Prediction, $\hat{\nu}_{t,i}$, is computed as shown immediately above in this table.	
Weight Change (see explanation on p. 88)	$\lVert w_{t,i} - w_{t-1,i} \rVert_1 = \alpha_t \lVert \hat{C}_{t,i} - \hat{C}_{t-1,i} \rVert_1$
Absolute Error* (Schmidhuber, 1991b)	$\lvert \delta_{t,i} \rvert$
Uncertainty Reduction* (see explanation on p. 89) Variance of Prediction, $\hat{\nu}_{t,i}$, is computed as shown above in this table.	$\hat{\nu}_{t-1,i} - \hat{\nu}_{t,i}$

Table 3.3: Intrinsic rewards investigated in the second study. Separate statistics are maintained for each task i, and only updated when task i is selected by the control learner. Starred (*) rewards performed poorly and were excluded from the results. We compute sample averages using an unbiased exponentially-weighted average introduced by Sutton and Barto (2018, Eq. 2.9).

Several intrinsic rewards are based on **violated expectations**, or surprise. This notion can be formalized using the prediction error itself to compute the instantaneous *Absolute Error* (Schmidhuber, 1991b) or *Squared Error* (Gordon and Ahissar, 2011). We can obtain a less noisy measure of violated expectations with an exponentially-weighted average of the error, which we call *Expected Error*. Regardless of the specific form, larger error results in larger intrinsic reward, encouraging further sampling for that target. Such errors can be normalized, as was done for *UDE* (White et al., 2014), to mitigate the impact of noise in and magnitude of the targets.

Another category of methods focus on **improvement**, and assume that the learning system is capable of continually improving its policy or predictions. This is trivially true for approaches designed for tabular stationary problems (Barto et al., 2004; Still and Precup, 2012; Little and Sommer, 2013; Meuleau and Bourgine, 1999; Barto and Şimşek, 2005; Szita and Lorincz, 2008; Lopes et al., 2012; Schossau et al., 2016). The most well-known approaches for computational intrinsic motivation make use of rewards based on improvements in (model) error, including those proposed by Schmidhuber (1991a, 2008), and Oudeyer et al. (2007). In our experiments, we include an intrinsic reward inspired by the intrinsic reward that Schmidhuber (1991a) referred to as "the current change of confidence in [the learner's world model]'s current prediction" (p. 1460–1461), which we call *Error Reduction*. Error Reduction is a coarse analog of Schmidhuber's (1991a) proposal, as we compute the difference in absolute error between one time step and the next, meaning error is considered for two different observations, as opposed to measuring a difference in the error for the same observation before and after updating. A better analog, for this reason, might be $\delta_{t,i}^2 - (C_{t,i} - w_{t+1,i})^2 = (C_{t,i} - w_{t,i})^2 - (C_{t,i} - w_{t+1,i})^2$, but we do not include such a reward in this study due to time constraints. Improvement in an extrinsic value function can also be used to construct intrinsic rewards, such as by using the (signed) TD-error as an intrinsic reward (Schembri et al., 2007b; Mirolli and Baldassarre, 2013, p. 55) or by tracking improvement in the value function over all states (Barto and Şimşek, 2005). Our problem does not include an extrinsic reward, so these last methods are not clearly applicable.

An alternative to improvement is to reward **amount of learning**. Doing so does not penalize errors becoming worse, and instead only measures that estimates are changing: the prediction learner is still adjusting its estimates and so is still learning. *Bayesian Surprise* (Itti and Baldi, 2006) formalizes the idea of amount of learning. A Bayesian learner, here defined as a learner which maintains a distribution over its weights, allows for the computation of Bayesian Surprise, which corresponds to the KL-divergence between this distribution over parameters before and after the update. The KL-divergence, in this case, measures how much the distribution over parameters has changed. Bayesian Surprise can be seen as a stochastic sample of mutual information, which is the expected KL-divergence between prior and posterior across possible observed targets. Linke et al. (2020, pp. 1303–1306) discuss this further. Other measures based on 'information gain' have been explored (Still and Precup, 2012; Little and Sommer, 2013; Achiam and Sastry, 2017; de Abril and Kanai, 2018; Berseth et al., 2021). In the tabular case, different variations of information-gain reward perform similarly to Bayesian

Surprise empirically (Little and Sommer, 2013). In this study we use an approximation of Bayesian Surprise for non-stationary settings with non-Bayesian learners, detailed by Linke et al. (2020, p. 1300).

While Bayesian Surprise was originally derived assuming stationarity and Bayesian learners, we use an approximation of Bayesian Surprise more appropriate for our non-stationary setting with non-Bayesian learners. In our setting, the prediction learner's main objective is to estimate an unknown mean. A Bayesian learner would maintain a distribution representing its belief about what this unknown mean could potentially be. Bayesian learning requires the assumption of an initial such belief distribution; a simple choice is to use a Gaussian distribution with an assumed variance. This choice has the advantage that the posterior belief distribution will also be Gaussian. We can have this advantage assuming the observations, $C_{t,i}$, made by the learner are themselves drawn from Gaussian distributions with known variance. In our case, the observations are indeed drawn from Gaussian distributions (refer to Equation 3.12), but we do not actually know the variance, σ_i^2. To mitigate this issue, we maintain an estimate of the target distribution's variance, $\nu_{t,i}^{(y)}$, and use it as our "known" variance.[5] We estimate the variance using an exponentially-weighted average variant of Welford's algorithm: $\nu_{t,i}^{(y)} \leftarrow (1-\beta)\nu_{t-1,i}^{(y)} + \beta(C_{t,i} - \hat{C}_{t-1,i})(C_{t,i} - \hat{C}_{t,i})$; using the exponentially-weighted average helps account for the non-stationarity of the target distribution. At each time step, we construct analogs of the Bayesian prior and posterior distributions. For the prior distribution, we take the mean to simply be the prediction learner's estimate, $\hat{C}_{t,i}$, of the mean at time t (before the observation at time t) and the variance to be $\nu_{t,i}^{(y)}/d_t$, where $d_t = (1-\beta)d_{t-1} + 1$. The denominator, d_t is like a decayed version of t. If we were doing a real Bayesian update for a Gaussian prior with the observations drawn from a Gaussian with known, constant variance (that

[5] Typically, a Bayesian learner would simply maintain a distribution over both the mean and variance when both are unknown. Our goal here, however, is to approximate Bayesian surprise for a non-Bayesian learner. Since the learner only estimates the mean, we assume that the corresponding Bayesian learner can only maintain a distribution over the mean.

is, if we knew σ_i^2), then the posterior variance, $\nu_{t,i}$, would satisfy $\frac{1}{\nu_{t,i}} = \frac{1}{\nu_{prior}} + \frac{t}{\sigma_i^2}$, where ν_{prior} is the prior to the update at time $t = 1$. We consider ν_{prior} to be infinite,[6] so the preceding equation simplifies to $\nu_{t,i} = \frac{\sigma_i^2}{t}$. But because we need to estimate σ_i^2, we replace σ_i^2 with the estimate $\nu_{t,i}^{(y)}$ and because the distribution is non-stationary, we replace t with a decaying analog, d_t (each new sample does not provide as much increase in certainty as it would if the distribution was stationary) and prevent the variance from decreasing below 10^{-3}. We do not claim that this is an ideal approximation strategy for Bayesian Surprise for non-Bayesian learners, but employ it as a simple strategy that allows us to include an approximation of Bayesian Surprise in our experiments.

We can additionally consider non-Bayesian strategies for measuring amount of learning. In this work, we include four intrinsic rewards invoking such strategies. *Absolute Value of Learning Progress*, adapted from Oudeyer et al. (2007), reflects change in error over a recent time interval. *Variance of Prediction* similarly reflects amount of learning via measuring the amount of recent variability in the prediction learner's estimate. Rewarding Variance of Prediction can also be seen as encouraging behaviour analogous to uncertainty sampling from active learning: Settles (2012) describes uncertainty sampling as querying the "which the learner has the highest output variance in its prediction" (p. 17). Variance of Prediction can be manipulated to produce what we call *Uncertainty Change*: how much the Variance of Prediction estimate has changed since the preceding time step, which reflects the degree to which the prediction learner is settling on a stable prediction. Last, we also include an intrinsic reward based on the change in weights: Weight Change can be understood as measuring amount of learning when the control learner trusts that the prediction learners are using the observed data appropriately towards their primary responsibility of estimating their targets accurately.

[6] A normal distribution with infinite variance can be thought of as a uniform distribution over the entire real line, which implies no prior knowledge about the unknown mean, which is exactly our situation prior to making any observations. This 'distribution' is not a valid probability distribution, but it is what is known as an improper prior in Bayesian statistics, and still results in a valid posterior after the first observation.

With this assumption, the learning system can make use of intrinsic rewards based solely on the prediction learner's parameters, such as the change in the weights. In this work, we define Weight Change using the ℓ_1 norm,

$$\text{Weight Change}(w_{t,i}, w_{t+1,i}) \coloneqq \|w_{t,i} - w_{t+1,i}\|_1. \tag{3.15}$$

Note that, in our setting, Weight Change is simply Absolute Error scaled by the step-size parameter, emphasizing the role that learner capability plays in ensuring an effective reward.

$$\|w_{t,i} - w_{t+1,i}\|_1 = \|w_{t,i} - \underbrace{[w_{t,i} + \alpha_{t,i}\delta_{t,i}]}_{(3.10)}\|_1 \tag{3.16}$$

$$= \| - \alpha_{t,i}\delta_{t,i}\|_1 = \alpha_{t,i}|\delta_{t,i}| \tag{3.17}$$

If we instead assumed that the prediction learners could not be trusted, the intrinsic rewards would need to be computed to overcome poor learning. This approach would require the learning system to recognize when a prediction learner is non-introspective, and decrease the reward associated with that learner. If the learning system can measure this, though, then presumably so too can the prediction learner—they are, after all, part of the same system. The prediction learner should then be able to use the same measure to adjust its own learning.

The Uncertainty Change intrinsic reward can be modified to reflect improvement rather than amount of learning by removing the absolute value. We experimented with such a reward, calling it Uncertainty Reduction.

We omit several strategies because they either (1) would result in uniform exploration in our pure exploration problem, (2) require particular predictions about state to drive exploration, (3) are designed for the offline batch setting, or (4) are based on statistics of the targets rather than the statistics generated by the prediction learners. Count-based approaches (e.g., Brafman and Tennenholtz, 2002; Bellemare et al., 2016; Sutton and Barto, 2018) are completely unsupervised, rewarding visits to under-sampled states or actions—resulting in uniform exploration in our problem. Though count-based approaches are sometimes used in learning systems, they reflect novelty rather than improvement or surprise (compare Barto et al., 2013).

The second set of strategies we omit are methods that use a model to encourage exploration (Schmidhuber, 2008; Barto et al., 2004; Stadie et al., 2015; Pathak et al., 2017, 2019) such as by using Bayesian Surprise for next-state prediction (Houthooft et al., 2016). Subgoal discovery systems (Kulkarni et al., 2016; Andrychowicz et al., 2017; Péré et al., 2018) define rewards to reach particular states. Empowerment and state control systems are explicitly designed to respect and use the fact that some tasks or regions of the state-space cannot be well learned. Often such systems use only unsupervised signals relating to statistics of the exploration policy, ignoring the statistics generated by the learning process itself (Karl et al., 2022). Like count-based approaches, unsupervised measures like this would induce uniform exploration in our stateless task.

Curriculum learning—learning what task to sample next—is closely related to our multi-prediction problem. Graves et al. (2017) introduce several measures for batch curriculum learning that are related to the ideas underlying the intrinsic rewards discussed above. Most related, Prediction Gain corresponds to Error Reduction, albeit assuming a batch of data rather than an online instance. An approximation, called Gradient Prediction Gain, corresponds to the norm of the gradient; for our setting, this is the same as the Absolute Error. Several of the measures considered by Graves et al. (2017) require the ability to sample new batches of data, such as Supervised Prediction Gain and Target Prediction Gain. Finally, Graves et al. (2017) investigated several Complexity Gain measures for the neural networks, measuring KL divergence between the posterior and a learned prior. The prior is updated towards the previous posterior, and so the resulting KL is related to Bayesian Surprise. The KL itself, though, is not used: rather, the gain in complexity is measured by looking at the difference in two KLs, before and after an update. These approaches require Bayesian learners with a separate prior distribution to be learned just to measure the complexity. The most simple and computationally feasible of these is L2 Gain, which is simply the difference in ℓ_2 norm of the weights before and after and update: $\|w_{t,i}\|_2^2 - \|w_{t-1,i}\|_2^2$. L2 Gain rewards the learning system for making the weights smaller, and performed worse than random for curriculum learning (Graves et al., 2017).

Finally, we do not test intrinsic rewards based only on targets, such as variance of the target. To see why, consider a behaviour that estimates the variance for a constant target, and quickly determines it only needs to select that action a few times. The prediction learner, however, could have a poor estimate of this target, and may need many more samples to converge to the true value. Separately estimating *possible* amount of learning from *actual* amount of learning has clear limitations. Note that in the stationary bandit setting, with a simple sample average learner, the variance of the target provides a measure of uncertainty for the learned prediction (Audibert et al., 2009; Garivier and Moulines, 2011; Antos et al., 2008), and has been successfully applied in education applications (Liu et al., 2014; Clement et al., 2015). When generalizing to other learners and problem settings, however, variance of the target will no longer obviously reflect uncertainty in the predictions. We therefore instead directly test intrinsic rewards that measure uncertainty in predictions, including Uncertainty Change and Variance of Prediction.

3.5.5 Experimental Setup

We conducted five experiments in the Drifter-Distractor problem described in Section 3.5.2. The goal of these experiments is to (a) assess the utility of different intrinsic rewards in our testbed, and (b) to understand how the ability of the underlying prediction learners—introspective or not—impact the results.

Each component of the learning system is modulated by several hyper-parameters that interact in different ways. The control learner (using the gradient bandit algorithm) makes use of a step-size parameter α and the step-size parameter of its average reward estimate α_r. For non-introspective learners, each prediction learner makes use of a (shared) step-size parameter α_p, with $\alpha_i = \alpha_p$ for all i.

For introspective learners, the step-size adaption method Autostep uses a meta learning-rate parameter κ. Finally, many of the intrinsic rewards have their own tunable parameters. For example, UDE uses an exponential average of recent errors which requires a smoothing parameter, β. The Absolute Value of Learning Progress reward makes use of two windows of recent errors determined by scalar parameters η and τ. In most cases the key parameters of the prediction learner, control learner, and intrinsic reward correspond to different timescales—slower or faster—and so required noticeably different values. We show that these choices have a big impact on behaviour, so we needed extensive sweeps and analysis to gain insight into the methods. This warranted investigating each result deeply, to communicate a nuanced picture.

Hyper-parameters	
control learner (Gradient bandit algorithm)	**Step-size parameter** $\alpha \in \{2^{-8}, 2^{-7}, \ldots, 2^{-2}\}$ **Average reward rate** $\alpha_r \in \{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$
control learner (Dynamic Thompson Sampling)	**Step-size parameter** $\alpha \in \{2^{-8}, 2^{-7}, \ldots, 2^{-2}\}$ **Initial mean estimate** $m_a = 100$
Non-introspective prediction learner (LMS with a constant step-size parameter)	**Step-size parameter** $\alpha_p \in \{2^{-7}, 2^{-6}, \ldots, 2^{-2}\}$, with $\alpha_i = \alpha_p$
Introspective prediction learner (LMS with Autostep)	**Meta learning-rate** $\kappa \in \{0.01, 0.05, 0.1\}$ **Initial step-size** $\alpha_{0,i} = 1.0$
Smoothing parameter (Variance of Prediction, Uncertainty Change, Bayesian Surprise, UDE, Expected Error)	$\beta \in \{10^{-7}, 10^{-6}, \ldots, 10^{-1}\}$
Absolute Value of Learning Progress Window (all combinations s.t. $\eta > \tau$)	$\eta \in \{1, 5, 10, 25, 100, 1000\}$ $\tau \in \{1, 5, 10, 25, 100\}$

Table 3.4: The hyper-parameter configurations investigated across all experiments in the second study. There was a total of 50,000 combinations of intrinsic reward function and hyper-parameter setting, with each of these evaluated using 200 independent runs.

We extensively sweep all the key performance parameters of every learner and reward function, to ensure an accurate characterization of performance. Table 3.4 lists all the parameter settings we tested. In some cases we report results for several parameters to gain more specific insights into the behaviour induced by an intrinsic reward. When providing overall results, we report the best performance of the learning system for each intrinsic reward, using the best performing parameters across all parameters tested. The best performing parameters were those that achieved the lowest total MSE (defined in Equation 3.11) averaged over every timestep of the experiment and averaged over 200 independent runs. All told we tested over 50,000 parameter configurations 200 times each across our three experiments.

When reporting results under the best parameters, we jointly tune hyper-parameters for the intrinsic reward and the prediction learners. These hyper-parameters are all part of the agent; the best hyper-parameters reflect the best the agent could do for that intrinsic reward and prediction learner. Even under ideal circumstances, many intrinsic rewards can fail to induce the desired behaviour, highlighting issues with the intrinsic rewards or with the use of non-introspective learners.

Nonetheless, reporting the best parameters does not provide the full picture, and though we attempt to highlight certain key results for other hyper-parameters, we cannot and do not attempt to show the full picture. Ideally, we could slice down further, to provide this nuance. At the extreme, this could consist of showing all possible intrinsic rewards—Absolute Value of Learning Progress with the smallest step-size parameter, Absolute Value of Learning Progress with a the largest step-size parameter, and so on—for each of the many combinations of prediction learner and control learner (with different hyper-parameter settings). This is infeasible,[7] and part of the role of the empirical analysis is to summarize key outcomes. We

[7] To enable the reader to do this on their own, we have provided a Python notebook to explore the full set of data, at http://jair.adaptingbehaviour.com.

have provided what we believe are the key slices: different intrinsic rewards (under their ideal circumstances) with two types of prediction learners (non-introspective and introspective). When intrinsic rewards fail under idealized scenarios, this reflects how they might perform across hyper-parameter settings. When intrinsic rewards result in near-ideal behaviour, we then dig deeper to understand if this was an accident of idealized hyper-parameter tuning, or more generally a characteristic of the intrinsic reward.

We follow the same basic template in the presentation of the results. First we report the behaviour of the best configuration for each reward function using non-introspective learners (*i.e.,* without Autostep). For a given reward, the behaviour is depicted by the probability of selecting each action over time according to the control learner's policy. This gives us insight into how each reward drives action selection over time. We then investigate the MSE over time, plotting both the error of each predictor and the average. Finally, in each experiment we investigate the performance sensitivity of several intrinsic rewards with respect to the tunable parameters. This provides more detailed understanding of how the parameters interact and helps explain when some intrinsic rewards produce unexpected behaviours.

As a final note, the behaviour plots (showing the action selection probability) do not include error bars. The error bars over 200 runs are negligibly small, except in some cases where action probabilities across runs varied significantly for poorly performing intrinsic rewards (e.g., the selection of the distractor targets with Squared Error reward in Figure 3.15). The variance across runs, however, can both make the plots difficult to read and hides how the action probabilities can vary over time within one run. For these plots, therefore, we instead show a sampling of individual runs. All learning curves (plotting root MSE) include standard error bars. The error bars in all learning curves are not visible because they are smaller than the width of the mean line.

3.5.6 Experiment: The Drifter-Distractor Problem

The Drifter-Distractor problem has one constant target, two distractor targets and one drifter target (see Figure 3.9 in Section 3.5.2). This four-action problem highlights some key features we want from our learned behaviours. The behaviour should not be continually distracted by noisy or unlearnable things (the two distractor targets). It should be able to quickly learn about simple targets (the constant target), and ultimately focus action selection on targets that result in continual prediction improvement (the drifter target). We test if such a behaviour is learned, with non-introspective and introspective learners, under different intrinsic rewards.

Let us first be more precise about how the behaviour should look in this problem. Consider the ideal setting, where we have a Bayesian prediction learner. The behaviour should try out all the actions in the beginning. The prediction learner associated with the constant target should quickly reduce its error and control learner should almost completely stop selecting the corresponding action. The prediction learners associated with the high-variance distractor targets will take longer to learn due to the target variance, but eventually the posterior for these narrows as well and the learner converges to the correct prediction of zero. Once that happens the behaviour should stop choosing the actions corresponding to the distractor targets. Finally, the prediction learner corresponding to the drifter target cannot ever reduce its error to zero: unending prediction improvement is possible. A Bayesian learner for the drifter target is effectively performing filtering, and needs to see samples constantly to track the changing mean. Therefore the behaviour should eventually settle on selecting the action corresponding to the drifter target the majority of the time. This behaviour is the ideal, or correct, behaviour, in that it most efficiently gathers the data needed for each prediction task.

There are a few common degenerate behaviours that are possible in this problem. The first is over-selecting the actions corresponding to the distractor targets. Every time the behaviour takes one of these actions, the corresponding non-introspective prediction learner updates toward a random target and so its predictions can oscillate around the optimum. Over short windows of time, the variance of the drifter target is smaller than the distractor targets; within that window, the errors generated by the distractor targets will appear larger. This results in the behaviour frequently selecting the distractor targets, occasionally selecting the drifter target and cycling between the three. Any methods that rely on prediction learners to not chase noise, such as Weight Change, should exhibit this degenerate behaviour. With non-introspective learners, this can only be prevented if the intrinsic reward can somehow distinguish between distractor targets and drifter targets.

The other common degenerate behaviour is selecting all actions nearly equally. This strategy does not result in the lowest possible MSE, but it does result in lower MSE than other behaviours such as mostly selecting the actions corresponding to the distractor targets. The uniform strategy emerges because there is no setting of the parameters of the intrinsic reward to force the behaviour to follow the ideal strategy described above.

3.5.6.1 Results with Non-introspective learners

Figure 3.11 summarizes the behaviour of the gradient bandit algorithm with several intrinsic reward functions, with non-introspective learners. The bold dash lines reflect the probabilities averaged over 200 runs, while the light stroke solid lines depict probabilities of individual runs. Several rewards induced the ideal behaviour described above to varying degrees. Rewards based on simple moving averages of each learner's prediction error, including Expected Error and UDE, quickly latch on to the action corresponding to the drifter target. The parameter sweep chose a short averaging window, because the δ are more consistently the same sign for the drifter target, making the Expected Error higher for the drifter target. Using the variance of each predictors estimate, as in Variance of Prediction and Uncertainty Reduction, the behaviour also converges to mostly selecting the action

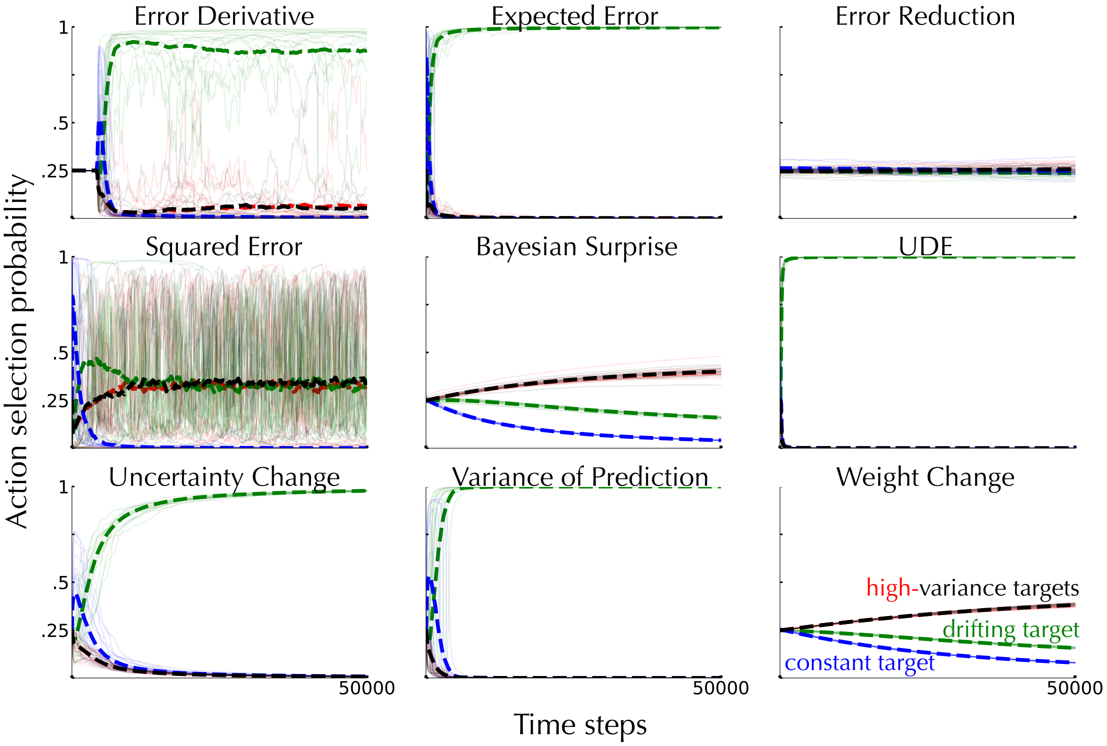


Figure 3.11: **Behaviour** in the **Drifter-Distractor** problem with **non-introspective learners**. Each subplot corresponds to the behaviour of the gradient bandit algorithm with a different intrinsic reward. Each line depicts the action selection probabilities learned by the control learner, over 50000 steps. The bold dashed lines show the mean probability of each action, averaged over 200 repetitions of the experiment. The light stroke solid lines show the probabilities computed by the gradient bandit algorithm for each action on individual runs—we only show a small random subset of 15 runs for readability. The **green line** corresponds to the drifter target, the **blue line** corresponds to the constant target, and the **red** and **black lines** correspond to the distractor targets. Intrinsic rewards based on variance estimates and averaging errors over time induce near-ideal action selection.

corresponding to the drifter target, after exploring the actions corresponding to the constant and distractor targets initially a bit longer. A parameter corresponding to a long window is used, because the predictions for the drifter target change much more over time than those for the distractor targets. Perhaps unsurprisingly the Squared Error and Error Reduction produce inappropriate behaviour. Bayesian Surprise and Weight Change cause the gradient bandit algorithm to be distracted by the distractor targets resulting in sub-optimal behaviour. The Absolute Value of Learning Progress reward induces behaviour that looks near-ideal in expectation, albeit there is more variance across runs than exhibited by other intrinsic reward functions.

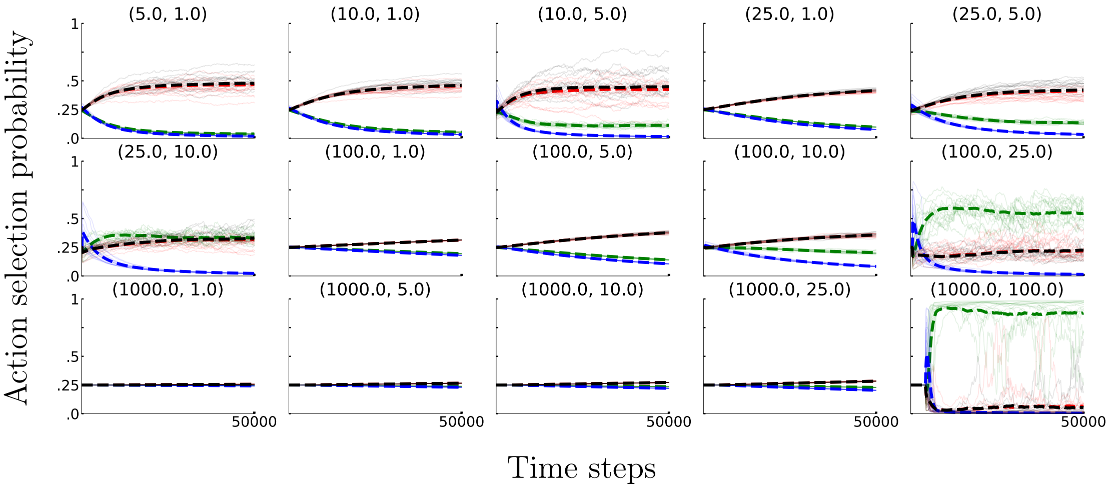


Figure 3.12: The impact of varying the key parameters η and τ of the **Absolute Value of Learning Progress** reward, in the **Drifter-Distractor** problem, with **non-introspective learners**. Each subplot depicts the behaviour of the gradient bandit algorithm with the Absolute Value of Learning Progress reward for many combinations of η, τ as indicated by the labels. As in Figure 3.11, each subplot shows both the average action selection probability for each action over time, and a small subset of individual runs. A large diversity of behaviours can be induced by changes to the window length parameters. Only one setting induced correct behaviour: $\eta = 1000, \tau = 100$. This explains why the initial action selection was uniform in Figure 3.11: the reward is zero until the windows fill, which takes 1000 steps for $\eta = 1000$.

Performance in the Drifter-Distractor problem with non-introspective learners is largely dependent on setting the hyper-parameters of the each reward correctly. To illustrate this sensitivity, consider the Absolute Value of Learning Progress reward, which is parameterized by two scalars η and τ. The η parameter controls the size of the window used to average recent errors, and τ controls how much each of the two windows overlap. Figure 3.12 shows the behaviour of the gradient bandit algorithm, in terms of action selection probability over time, for every combination of η and τ. For each pair of (η, τ) we selected all the other hyper-parameters in the learning system to minimize the total MSE; each subplot of the figure represents the best performance possible for a given (η, τ) pair according to MSE. Across these combinations, we see the full gamut of behaviours. Only one setting out of twelve exhibited the described good behaviour; most were uniform or focused on the distractor targets.

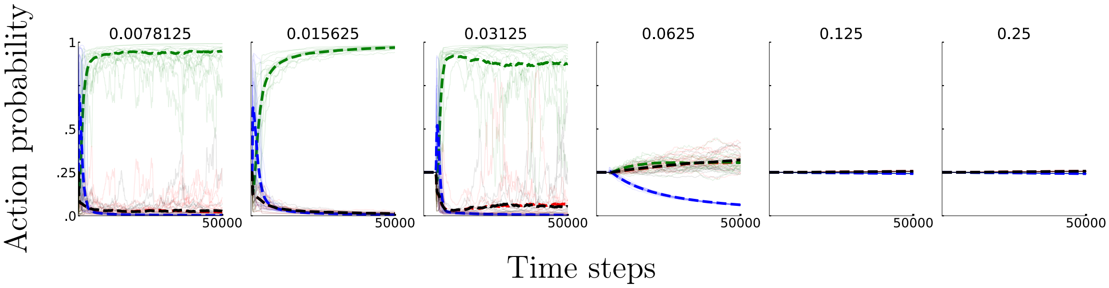


Figure 3.13: The impact of varying the LMS step-size parameter, α_p, with the **Absolute Value of Learning Progress** intrinsic reward, in the **Drifter-Distractor** problem, with **Non-introspective learners**. Each subplot depicts the behaviour of the gradient bandit algorithm with Absolute Value of Learning Progress reward for for different values of α_p as indicated by the labels. Large α_p—faster target tracking—induces a uniform behaviour, and smaller α_p produce action selection more similar to the ideal behaviour but MSE is higher because predictions are learned slowly. The third subplot, corresponding to $\alpha_p = 0.03125$, achieved the lowest total MSE, because it allowed for somewhat faster learning for the predictions, but was still slow enough for the behaviour to estimate learning.

The hyper-parameters of the other components of the learning system also interact with the reward function. Figure 3.13 shows the best behaviour—in terms of MSE—of the gradient bandit algorithm for different values of the LMS predictor step-size parameter α_p. As the predictors learn faster, the Absolute Value of Learning Progress reward induces nearly uniform action selection. If we slow the prediction learners updates with a smaller step-size parameter value, then the behaviour strongly favors the action for the drifter target. This makes sense because with a small α_p, the intrinsic reward for the distractor targets becomes smaller and much bigger for the drifter target because the step-size parameter value is not large enough to track quickly. Though the action selection by the behaviour is correct, this is not what we want from the learning system: we want the prediction learners to learn quickly, rather than artificially slowly so that the behaviour can more easily track what they know. In fact, with small step-size parameter values, the MSE is much worse than we can get with the introspective learners, where it is much easier to estimate prediction improvement and prediction learners can learn more aggressively.

Finally, let us investigate the error over time for each intrinsic reward. Figure 3.14 shows the exponential average of the root MSE over time for each reward function. We choose an exponential average to smooth the results (with a decay constant of 0.999). We plot both the error of each target, and the error average across targets. All rewards except UDE result in perfect prediction of the constant target; even UDE has near-zero error, indicating only minor under-selection of the action for the constant target. Rewards that induce nearly uniform action selection generate larger prediction error in aggregate (Error Reduction and Bayesian Surprise). Reward functions that do not induce a strong preference for the drifter target exhibit high or growing error (Weight Change). Rewards that induce strong preference for the distractor targets do achieve better error on those predictions at the cost of accuracy in predicting the drifter targets (Squared Error). Achieving the lowest overall error requires first selecting the actions for the constant and distractor targets at first, and then focusing on the drifter target (i.e., UDE, Uncertainty Change, and Variance of Prediction).

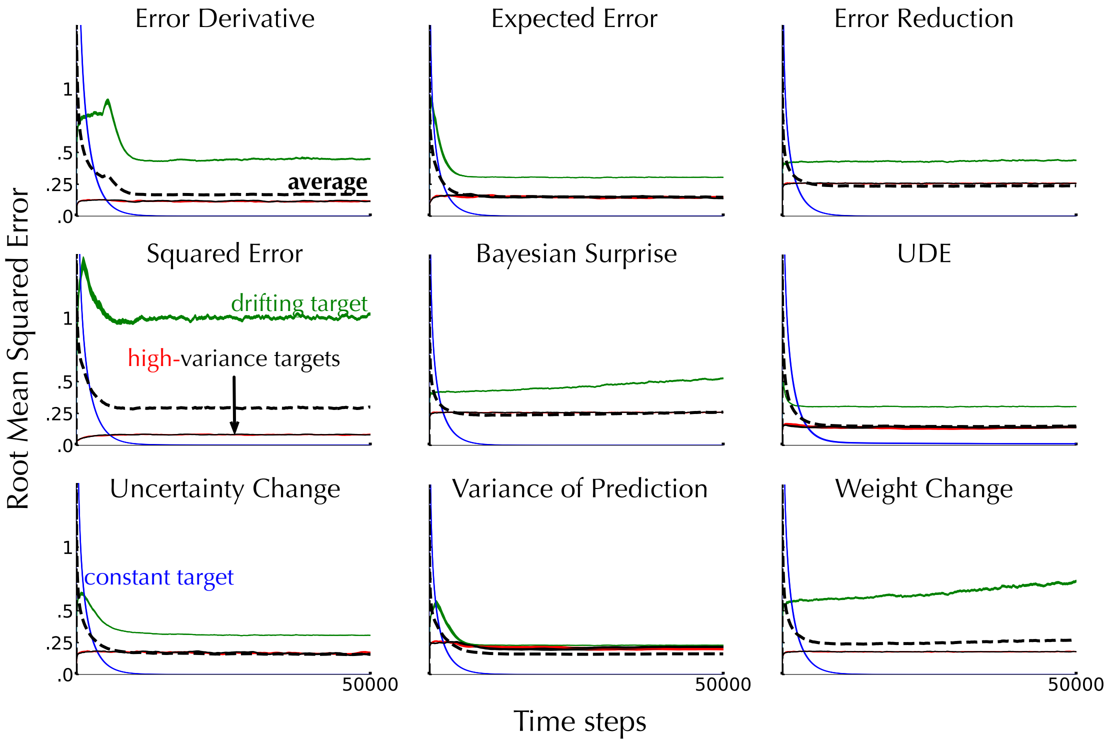


Figure 3.14: **Root MSE** over time corresponding to each intrinsic reward function in the **Drifter-Distractor** problem with **non-introspective learners**. Each subplot corresponds to a different reward as labelled. The line colors correspond exactly as in the previous plots: **green drifting**, **black** and **red high-variance**, and **blue constant**. Each line is the exponentially weighted moving average of the LMS predictor's root MSE. The root MSE is computed with an exponential average, with a decay 0.999. The final results are averaged over 200 independent runs (standard error bars are plotted but not visible). The heavy stroke black dashed line reports average of the other four. Although many rewards induce similar action selection strategies, they can produce different root MSE curves.

3.5.6.2 Results with Introspective Learners

In this section we analyze the impact of different intrinsic rewards with introspective learners. We use LMS learners with Autostep, a step-size adaption method, to obtain introspective prediction learners. First let us recall how the step-size parameter for each LMS learner might change over time (see Figure 3.10 in Section 3.5.3 for reference), based on the errors generated by each of our three target types. The distractor targets are noisy—even if the mean is stable—so the LMS learner will experience positive and negative errors. The Autostep algorithm will reduce the step-size parameter corresponding to these targets, allowing each LMS learner to mitigate the variance and converge to the correct prediction of zero. The constant target on the other hand is easy to predict. Autostep will keep the step-size parameter large because the errors will be of the same sign. However, the error on the constant error can easily be reduced to zero with repeated sampling. Once the prediction error is zero Autostep will modify the step-size parameter no further. The drifter target has noise, like the distractor targets, but the mean is not centered at zero, and it exhibits temporal structure. Consequently, the Autostep algorithm will keep the step-size parameter value high for the duration of the experiment. It is not hard to see that introspective learners should efficiently reduce error across all the targets, at least compared with a global, constant step-size parameter value. More subtly, an intrinsic reward that takes into account the dynamic values of the step-size parameter could exploit this additional information to adapt behaviour to reduce error even faster.

The setup of our second experiment was identical to the first except that each LMS learner maintained its own step-size parameter, $\alpha_{t,i}$, updated via Autostep. We also include an intrinsic reward based on the amount of change in the step-size parameter, *Step-size Change*, to assess the utility of rewarding action choices that caused changes in the step-size parameter values. This reward only makes sense if the step-size parameter can change over time, and thus was not included in the previous experiment.

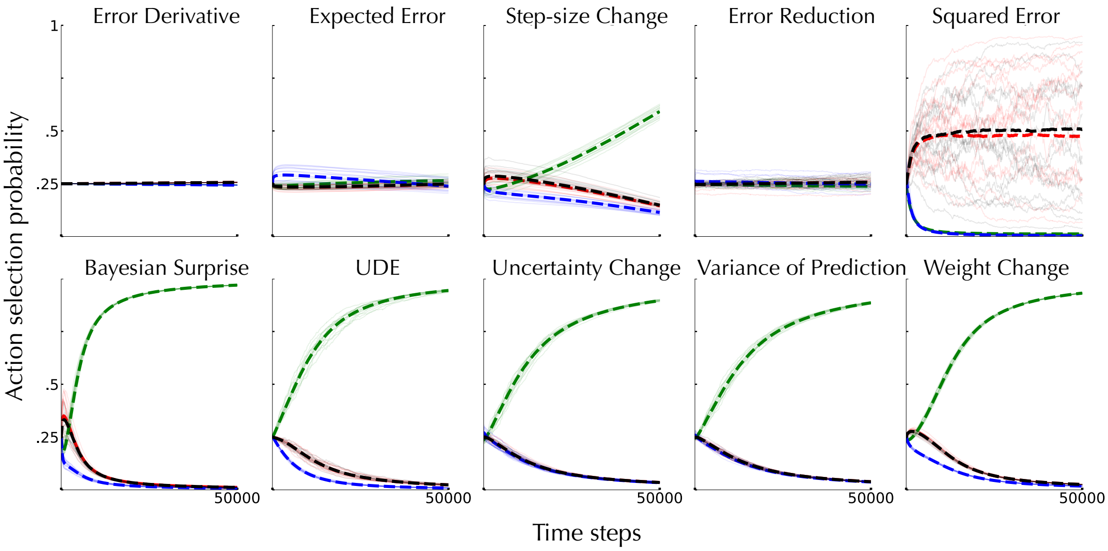


Figure 3.15: **Behaviour** in the **Drifter-Distractor** problem with **introspective learners**. Each subplot corresponds to the behaviour of the gradient bandit algorithm with a different intrinsic reward. Each LMS learner uses the Autostep algorithm to adapt the step-size parameter over time. The line coloring, labelling, and semantics mirror Figure 3.11. With Autostep, Weight Change induces near-ideal action selection. Absolute Value of Learning Progress and Expected Error rewards, on the other hand, induce inappropriate action selection.

The results of our second experiment are summarized in Figure 3.15. As before we plot the action selection probabilities to summarize the behaviour. Weight Change reward now induces near-ideal action selection. The step-size parameters for the distractor targets decay to a relatively small values causing Weight Change to decrease—those actions become less and less rewarding. Autostep keeps the step-size parameter value relatively high for the drifter target, on the other hand, and the change in weights remains relatively high. Finally, even though the step-size parameter does not decay to zero for the constant target, the prediction error for the constant target does go to zero. Consequently, the magnitude of the update also goes to zero, meaning Weight Change goes to zero and preference for the constant target diminishes over time. Bayesian Surprise induces similar behaviour as Weight Change as suggested by analysis by Linke et al. (2020, pp.1303-1306). The variance-based rewards and UDE induce the same overall action preferences as without Autostep.

Across the board, MSE is reduced, as shown in Figure 3.16. The root MSE is about half of that for the non-introspective learners. The differences in root MSE between the intrinsic rewards appear more minor, but the differences are meaningful. The total root MSE is well correlated with our definition of ideal behaviour in this domain—reward functions that result in lower error exhibit the expected action preferences over time. To see larger differences, though, we need more actions. This first experiment was primarily designed to investigate qualitative behaviour; the final experiment uses more actions and provides a better insight into quantitative differences.

For non-introspective learners, we observed that careful tuning of hyper-parameters allowed for the correct behaviour for certain intrinsic rewards, by slowing prediction learning. This was the case for the Absolute Value of Learning Progress, where in Figure 3.13 we observed that if the predictors learned too quickly, the drifter target did not produce the highest Absolute Value of Learning Progress. For introspective learners, prediction learning cannot be slowed: they increase learning

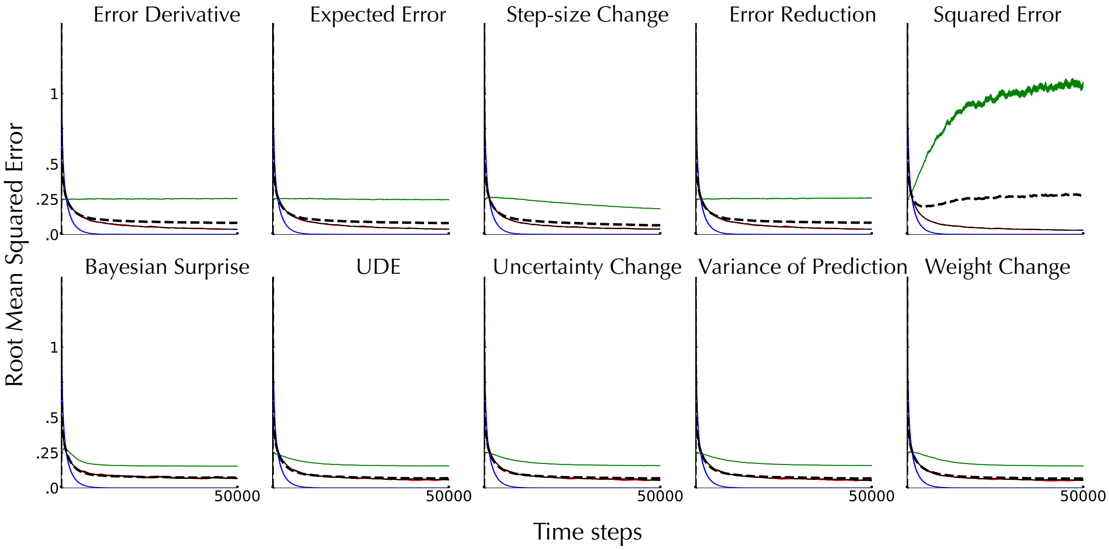


Figure 3.16: **Root MSE** over time corresponding to each intrinsic reward function in the **Drifter-Distractor** problem with **introspective learners**. Results averaged over 200 runs, and standard error bars included. In this experiment, reward functions that induce similar action preferences produce similar root MSE reduction over profiles. Using Weight Change reward produces the lowest root MSE (0.108), however both UDE (0.109) and Uncertainty Change (0.110) result in similar performance. Squared Error results in the worst performance overall (0.292), and rewards that induce uniform action selection like Absolute Value of Learning Progress result in larger error (0.124) compared with Weight Change.

when learning is possible. We might expect Absolute Value of Learning Progress to therefore perform poorly, and be unable to find an appropriate hyper-parameter setting. We find this is the case: Absolute Value of Learning Progress with Autostep does not induce the action selection preferences we expect—it causes nearly uniform action behaviour—and no setting of the window parameters resulted in appropriate action preferences (Figure 3.17).

Overall, the preference for the drifter target is less pronounced with introspective learners, as seen in Figure 3.15. Instead, the behaviour selects the actions for the distractor targets for longer. This is because step-size adaption is a meta (or second order) learning process, and so a non-trivial amount of data is required to recognize that learning is oscillating.

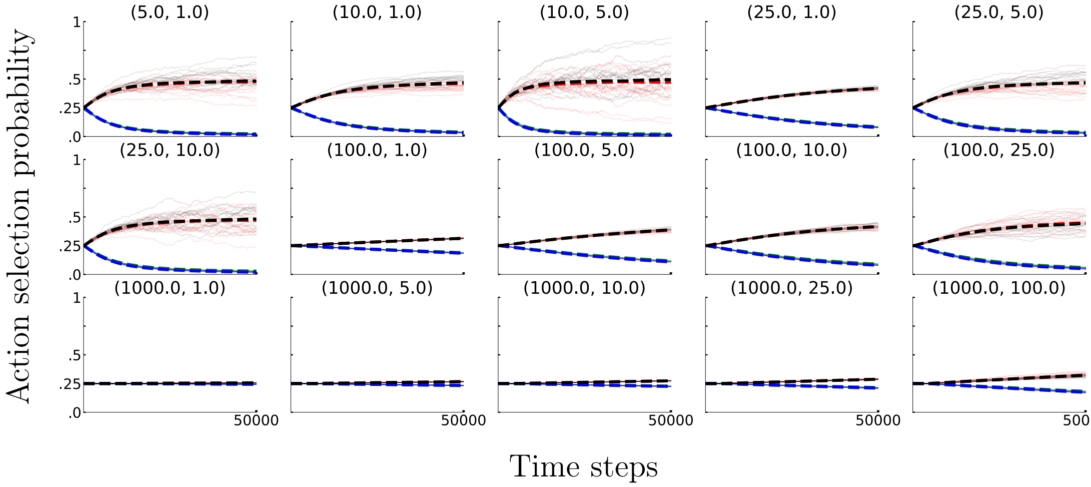


Figure 3.17: The impact of varying the window length parameters η and τ of **Absolute Value of Learning Progress** reward in the **Drifter-Distractor** problem with **introspective learners**.

One might therefore wonder if rewards like Weight Change simply hide the hyper-parameter tuning issue inside the step-size adaption algorithm. This seems not to be the case: the parameters of Autostep are straightforward to tune, and the behaviour is largely insensitive to these choices as shown in Figure 3.18. Small meta learning-rate parameter values slow learning but do not prevent preference for the drifter target. The results of our first experiment highlight the utility of both simple intrinsic rewards—one's without hyper-parameters—and introspective learners in multi-prediction learning systems.

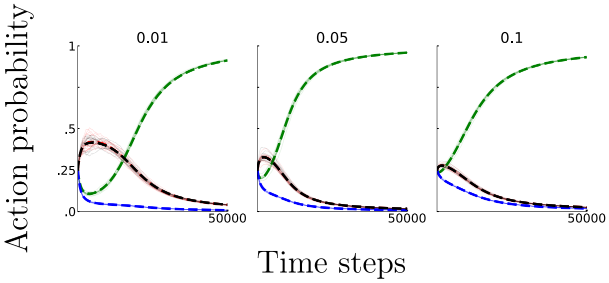


Figure 3.18: Action selection probabilities for the gradient bandit algorithm with **Weight-Change** reward under different meta learning-rate parameter values κ, in the **Drifter-Distractor** problem with **introspective learners**.

One final point of note is the surprising difference between UDE and Expected Error. In the previous experiment, with non-introspective learners, they performed similarly. In this experiment, with introspective learners, Expected Error results in uniform action selection whereas UDE provides the correct behaviour. This is surprising, as UDE corresponds to Expected Error divided by the long-run sample standard deviation of the target. If we look more closely at the behaviour induced by Expected Error with different smoothing parameters, β, in Figure 3.19, it becomes more clear why this is the case. A small β results in early errors dominating the exponentially-weighted average; consequently, the constant target is preferred, as it generates high error at first. A larger β is needed to avoid this issue, but this unfortunately causes poor estimates of the true expected error for the distractor targets (which should be zero). In fact, it makes the errors for those target appear higher. Consequently, for the four smaller β settings, the constant target is preferred and for the two large β, the distractor targets are preferred; there is no β amongst our set that lets the behaviour focus on the drifter target.

UDE, on the other hand, has a way to overcome this: the long-run variance estimate makes the drifter target appear better. The variance of the drifter target appears small in the beginning of learning, and it takes many steps to start to recognize that it is actually high variance. In contrast, the variance estimate for the distractor targets are learned quickly, and the variance for the constant target looks higher initially due to consistent decrease in the error. This behaviour is perhaps a bit accidental, and again highlights the complex interactions between all these hyper-parameters. This only further motivates the utility of intrinsic rewards without hyper-parameters, that instead rely on introspective prediction learners.

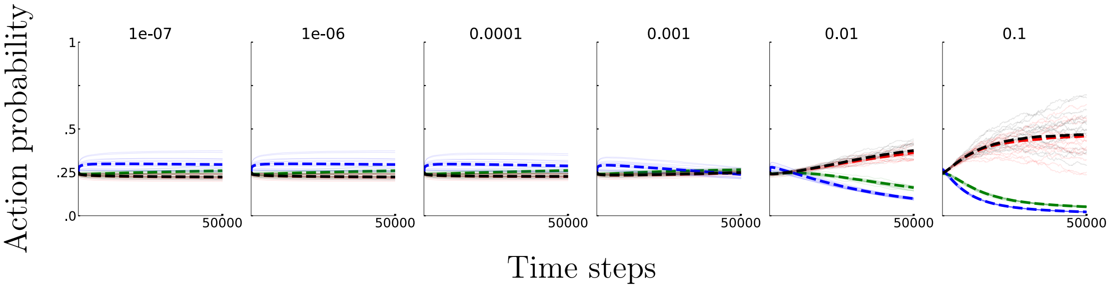


Figure 3.19: The impact of varying the smoothing parameter β of **Expected Error** reward in the **Drifter-Distractor** problem with **introspective learners**.

3.5.7 Discussion: Adapting the Behaviour of a Horde of Demons

The ideas and algorithms promoted in this study may be more impactful when combined with policy-contingent, temporally-extended prediction learning. Imagine learning hundreds or thousands of off-policy predictions from a single stream of experience, as in the UNREAL (Jaderberg et al., 2017) and Horde (Sutton et al., 2011) architectures. In these settings, the behaviour must balance overall speed of learning with prediction accuracy. That is, balancing action choices that generate off-policy updates across many predictions, with the need to occasionally choose actions in almost total agreement with one particular policy. In general we cannot assume that each prediction target is independent as we have done in this study; selecting a particular sequence of actions might generate useful off-policy updates to several predictions in parallel (White et al., 2012). There have been several promising investigations of how intrinsic rewards might benefit single (albeit complex) task learning (see Pathak et al., 2017; Hester and Stone, 2017; Tang et al., 2017; Colas et al., 2018; Pathak et al., 2019). However, to the best of our knowledge, no existing work has studied adapting the behaviour based on intrinsic rewards of a model-based or otherwise parallel off-policy learning system.

Simple intrinsic reward schemes and the concept of an introspective learning system should scale nicely to these more ambitious problem settings. We could swap our stateless LMS learners for Q-learning with experience replay, or gradient temporal-difference learning (Maei et al., 2010). The Weight Change reward could be computed for each predictor with computation linear in the number of weights. It would be natural to learn the behaviour policy with an average-reward actor-critic architecture, instead of the gradient bandit algorithm used here. Finally, the notion of an introspective learner still simply requires that each prediction learner can adapt its rate of learning. This can be achieved with quasi second order methods like Adam (Kingma and Ba, 2015), or extensions of the Autostep algorithm to the case of temporal difference learning and function approximation (Kearney et al., 2018, 2019; Günther et al., 2020; Jacobsen et al., 2019). It is not possible to know if the ideas advocated in this study will work well in a large-scale off-policy prediction learning architecture like Horde, however they will certainly scale up.

Maximizing intrinsic reward as presented in this study is not a form of exploration, its the objective of the learning system—the agent must explore in order to maximize the intrinsic reward. The intrinsic rewards do not provide a bonus to help improve exploration. In our stateless prediction task, sufficient exploration was provided by the stochastic behaviour policy. This will not always be the case, and additional exploration will likely be needed. Efficient exploration is an open problem in reinforcement learning. Combining the ideas advocated in this study with exploration bonuses or planning could work well, but this topic is left to future work.

3.5.8 Discussion: The Second Study in Context

The Drifter-Distractor is just one problem in the benchmark suite for comparing intrinsic rewards proposed by Linke et al. (2020). For a learner to perform successfully on this benchmark, it must demonstrate several important capabilities: avoiding dawdling on noisy outcomes, tracking non-stationary outcomes, and seeking actions for which consistent improvement is possible. This study included the empirical investigation of analogs of ten different well-known intrinsic reward schemes. In the process, we found that simple intrinsic rewards based on amount of learning can induce effective behaviour—avoiding classic degenerative behaviour—if the base prediction learners are introspective. Introspective prediction learners can decide for themselves when learning is done. Our results show the strength of focusing on more effective *learning* alongside simple (ideally parameter-free) intrinsic rewards. We found that intrinsic rewards based on amount of learning—like Weight Change—can perform well in problems specifically designed to distract the learning system.

This second study is a case study for how Curiosity Bandits can be used for both quantitative and qualitative comparison of different learners with open-ended behaviour. While this study was interested in *accurate* learners, so we used MSE as a measure of performance, other measures of performance can and should be considered.

3.6 Discussion: The Role of Interpretation

As I mentioned in Section 2.2, some authors, like Stojanov and Kulakov (2006), argue that curiosity requires a "more refined model for internal representation and thinking" than that used by the RL framework (pp. 46–47). However, through these experiments, we see that even simple decision-making problem models of the world still provide considerable complexity to consider.

The majority of intrinsic motivation methods are not designed for use in a simple decision process, and may use more information than differentiable states, actions, and rewards. Therefore, there are multiple ways to interpret each method when implementing an agent to make decisions in a simple single-state domain. For instance, many of the approaches I hope to test rely on other system components (like the expert learning machines in the approach devised by Oudeyer et al. (2007), which could be implemented many ways, including with neural networks, support-vector machines, Bayesian machines (Oudeyer et al., 2007, p. 270) or RL techniques). The choices made for these components necessarily have an impact on the behaviour of the overall system. For these reasons, these experiments do not give a complete evaluation of the possibilities inspired by each researcher's methods. However, it would be possible to explore the results achieved when these other components are manipulated. For example, how do different error-based IM methods compare when the prediction error is computed using different algorithms? In the first study we used Sarsa (temporally extended, policy-dependent predictions), and in the second we used LMS learning (myopic predictions), but one could consider other alternatives.

3.7 Conclusions

The goal of this work was to systematically investigate curiosity methods, and intrinsic reward methods more broadly. This chapter has two central contributions. The first is the introduction of curiosity bandits as a way to study intrinsic motivation algorithms in a unified context. The second is a comprehensive empirical comparison of different intrinsic reward mechanisms that, for the first time, puts them in context with each other. Along with a survey of intrinsically motivated learning systems, this work provides several new insights into the strengths and weakness of different intrinsic reward mechanisms, and may provide guidance for constructing larger, even more complex intrinsically-motivated reinforcement learning systems where an extensive and systematic study like ours is not feasible.

3.7.1 Fielding the Diversity of Curiosity

Curiosity is diversely defined. This is part of what makes computational curiosity so challenging and the approaches so widely varying. What kinds of behaviour should be considered curious is unclear, and the views on its benefits to an agent are discordant. So far, according to Santucci et al. (2012), most computational curiosity mechanisms have been designed either for "(a) the acquisition of knowledge, for example the acquisition of better prediction capabilities or the formation of object representations; (b) the acquisition of competence, i.e. the capacity to act so as to achieve a state of the world when it becomes desirable" (p. 1). These are not necessarily the only goals that are thought to be sought through curiosity, and of the knowledge and competencies available to an agent, it is often unclear which ones are 'better.'

However, this diversity of definition is also why studying the behaviour produced by different approaches is so valuable. In this chapter, with the context of the development of reinforcement learning approaches to computational curiosity as a whole, we have introduced simple experiments which will allow us to consider the developmental trajectories and behavioural tendencies of agents motivated to maximize different curiosity measures in a clear setting.

One way this work is limited is in that the choices which we intuitively believe to be most interesting in each world react with some function of time. We could imagine that patterns in the environment might not always be varying over time; they might vary spatially or with the activity of the agent. These patterns are generally still intuitively interesting, and should not be disregarded when testing computational curiosity approaches.

Further, the design of the Curiosity Bandits uses human intuition about which choices are curious and which are not. In future work, we may be able to develop more complex worlds for agents to explore based on existing studies in psychology, like those done by Berlyne and his successors. We may even be able to do our own studies of human choices in simulated microworlds with no effect on extrinsic reward (perhaps as very simple computer games).

In the proposed simple environments, a human evaluator will generally see only three choices. In future work, it would be interesting to place an agent in a world with more than three possible choices and see if varying the objective measure changes the order in which the choices are tackled.

3.7.2 Future Directions for Computational Curiosity

As we have already seen in Section 3.6, setting existing approaches in the extremely simple setting of a decision process pushes us as designers to recognize the choices available to us. This can provide excellent opportunities for modifying, modularizing, and combining different approaches. For example, the designers of the two information-theoretic measures discussed in detail in Chapter 2 do not use the sequence of reward observed in the computation of their curiosity intrinsic motivation in the agent. We may want to consider how we might incorporate the information provided by reward as part of these curiosity signals. The signal received by the agent in the current definition of empowerment is only $S_{t+\theta}$. It might be valuable to also include the sequence of rewards. In the context of Still and Precup (2012), the Predictive Power held by the agent is originally described as the mutual information carried by the current action and the current state about the future state, $I(\{A_t, S_t\}; S_{t+1})$ (p. 142). We might also want to consider including what information recent rewards might carry about the future state, or what information the current action and state might carry about future rewards.

Decisions in the development of new approaches can be made with the current context of computational curiosity in mind. One of the key open challenges in autonomous development is the intelligent restraint of an exploring agent. This direction has been emphasized by both Oudeyer (2010) and Doya (2010). Doya (2010) suggests that "the killer problem in designing curiosity is how not to let agents try to learn all the detailed structures in the world, most of which have nothing to do with their life" (p. 6). Oudeyer (2010) agrees, "intrinsically motivated exploration needs to be further constrained to harness the very large, potentially unbounded, volume and high-dimensionality of the space" (p. 7).

Much current work on exploration attempts to maintain the constraint where we must find *the* optimal policy, and the goal is to decrease the amount of time it takes to find that policy. For real-world, embodied agents like humans, animals, and robots, expecting to find *the* optimal policy is unreasonable. Exploring every possible state is infeasible (could lead to walking off a cliff), and many lifelong learning problems do not allow us to fulfill that kind of restraint, which is necessary if we want to achieve theoretical guarantees. What are more reasonable theoretical guarantees to strive for in a lifelong learning situation?

A challenge which has not been well-emphasized yet is the safety of computationally curious agents, both to themselves and to the humans we hope they can interact with. We recall from Section 2.1, the first recorded discussions on curiosity debated its value. Beyond highlighting the possible ethical debate which may overlap with the design of artificially curious agents, questioning the value of curiosity in humans should lead us to recognize the shortcomings and risks involved in successful implementation. As the stories of Pandora and Eve and the idiom about the cat[8] have been shared to remind us, unchecked curiosity can lead to undesirable consequences. For a computational curiosity mechanism to be successful in a general context, a balancing safety mechanism will be necessary.

While the field has some existing challenges, it also has potential. There are a number of important concepts being developed in the area of reinforcement learning which could be used in the development of approaches to computational curiosity. These include acquiring appropriate system biases, as demonstrated by Sutton (1992, p. 171). The rapid improvement of the GVF framework has also opened up problems whose solutions might come from curious behaviour, including choosing which policies would be updated, or followed, or which GVFs should be maintained at a particular moment in the agent's life. For an agent building up layers of abstractions as its representation of the world, how should it figure out what abstraction to build next? Though exploring these ideas in detail was beyond the scope of this chapter, developing these concepts is likely to be an important part of future work. Whether it is toward solving challenges or embracing area of potential, computational curiosity still has much room for growth. Analysis of behaviour is a key component of a full understanding of the field.

[8] "Curiosity killed the cat."

Chapter 4

Five Properties You Didn't Know Curious Machines Should Have

4.1 Books, Bookstores, and Machine Curiosity

Imagine you have a favourite corner bookstore near your home. You walk into the shop, browse the shelves for something new, take it home and read it end-to-end in less than a week—perhaps at the expense of sleep. The reading feels good; the unfolding plot makes you almost unable to put down the book. You close the last page and want more. There isn't any more book left, so you walk directly back to the bookstore for a chance at another great read. You don't buy and read the same book (that would be silly and you know how it ends); instead, you know that the bookstore can give you a new engaging reading experience. The more you read, the more you like reading (and that corner bookstore too!) This example is rooted in the properties of human curiosity. In this chapter, we focus on improving the specificity of how we think about curiosity with the goal of facilitating the implementation of key properties of human curiosity in machines.

As we described in the introduction to this dissertation, there is a long history of human thought and discussion around curiosity, culminating most recently with important advances in machine intelligence spurred by using curiosity as inspiration. In that section, we further emphasized the value of curiosity to our future machine learning systems and their human collaborators, and potential value to our understanding of curiosity as a whole. In this way, advancing curiosity in the domain of machine intelligence is expected to have substantial reciprocal benefits for research domains focused on human and animal curiosity, like those in psychology, education, philosophy, neuroscience, and behavioural economics. After all, advances in machine intelligence have long supported the development of new theories of biological intelligence (MacPherson et al., 2021, pp. 603, 604; Newell, 1970, pp. 363–370; Sutton and Barto, 2018, Chs. 14–15). We propose that the understanding of curiosity, as a facet of intelligence, can be similarly bolstered through the development of models of curiosity for machine intelligence. Given curiosity's political role "equip[ping] us to pursue a more intellectually vibrant and equitable world" (pp. xi–xii), scholars like Zurn and Shankar (2020) have emphasized the urgency of transdisciplinary conversation on curiosity.

To readers focused on curiosity in domains beyond computing: this chapter is in large part addressed to you. One goal of this chapter is to provide a new perspective on curiosity applicable to any learner, whether human, animal, or machine. Implementing any concept as an algorithm requires a different way of thinking. The abstractions that have been used thus far to improve our understanding of biological curiosity are different from those needed to build a curious machine. This approach and this work provide a unique perspective that may help researchers from multiple disciplines understand curiosity more deeply. This dissertation is meant to contribute as much to the field of curiosity studies (see Zurn and Shankar, 2020, p. xii–xiii) as to that of machine intelligence.

The synthesis completed in this work has led us to define and explore five key properties needed to capture the full value of human curiosity for machine curiosity. While, as we noted in Chapter 1, existing frameworks for curiosity in machine intelligence have offered clear successes, we suggest that learners implementing those frameworks would not exhibit the full range of curious behaviours exhibited in our bookstore example[1]—and we *do* want to see that full range. In contrast, with our five properties, we posit that a learner would exhibit recognizably curious behaviour.

Our five properties are all drawn from the study of *specific curiosity*. Specific curiosity has been described by Loewenstein (1994, p. 77) as "the desire for a particular piece of information." In this chapter, the term specific curiosity refers to one of the most intuitive uses of the unadorned term *curiosity*, as it is the cognitive, emotional, and embodied condition humans imply by saying, "I am curious to know X." However, we aim not to consider specific curiosity through the lens of any particular definition in this work. Instead, we focus on detailed descriptions of its properties, allowing for future proposals of aspects of specific curiosity that this list does not include.

We have seen numerous recent successes in artificial intelligence (AI), generating popular expectations that machines can learn efficiently and effectively. Yet, artificial intelligence systems often fall short, particularly when they must direct their own learning. For humans in such situations, much of their learning is driven by specific curiosity, a temporary desire to find out a specific piece of information. We expect AI to similarly benefit from specific curiosity, yet it has not been explored directly as inspiration for mechanisms for machine intelligence.

To understand why we refer to *specific* curiosity in particular, it helps to know some history of curiosity research. The term curiosity has been used as an umbrella term to describe a number of phenomena, generally information-seeking and knowledge-seeking behaviours in humans and other animals (Kidd and Hayden, 2015, p. 449). When a particular subset of these phenomena is studied, it often acquires a distinct name to define the scope of the study and clarify that the behaviours of interest may or may not represent a "different" phenomenon than other phenomena under the curiosity umbrella (e.g., Berlyne, 1954, p. 180). This choice

[1] This argument can be found in 4.3.3.2, but we recommend understanding the five key properties in Section 4.2 first.

allows authors to leave open the possibility that the phenomenon may be "different" in any of a number of ways, such as having different underlying mechanisms. Specific curiosity is one such subset of the curiosity umbrella.

In particular, the identifier 'specific' is typically used in contrast with 'diversive' (Loewenstein, 1994, p. 77). While only later used with the word 'curiosity,' the specific–diversive division derives from Berlyne (1960), who differentiated taking exploratory action for the purpose of learning something specific (specific exploration) versus for the purpose of relieving boredom or increasing stimulation (diversive exploration) (Berlyne, 1960, p. 80; 1966, p. 26). While Berlyne (1966) felt that diversive exploration seemed "to be motivated by factors quite different from curiosity" (p. 26), the terms 'specific curiosity' and 'diversive curiosity' have been used by other authors over the intervening years. We have chosen to adopt the term specific curiosity not only to emphasize that we are interested in a motivation to learn something specific, but also to differentiate our goals from those of works on machine curiosity typical today (see Section 4.3.3 for an overview).

In Section 4.2, we describe the five key properties of specific curiosity in detail, specifically considering in Section 4.3 their translation to the domain of reinforcement learning and related curiosity methodologies therein. In Section 4.3.3.4, we then offer an experimental demonstration of a computational specific curiosity agent inspired by those properties, along with detailed analysis of the resulting behaviour both with the properties intact and when each individual property is ablated in turn. Would a machine learner exhibit behaviour similar to that of the curious reader (you!) if placed in a similar setting? In our experiment, we will show how including just three of the key properties already helps a machine learner exhibit behaviour analogous to yours in the bookstore example. A machine learner might indeed return to the bookstore with the addition of a few specific and possibly easy to implement computational properties of specific curiosity.

4.2 Understanding Specific Curiosity and the Five Properties

As you were reading your book, why did you have trouble putting it down? A clever author can walk the reader from question to question along the narrative. Each individual question seems to be a variation on: "What's going to happen next?" but each question is new and specific to the moment (How did they get out of the locked room? What is that character's motivation? Did the butler do it?) You know how to find each answer and in doing so, satisfy your curiosity—keep reading!

4.2.1 A Framework for Expressing Specific Curiosity

Our first major act of synthesis in this manuscript will be to conceptually separate the moment where curiosity is induced from the moment where curiosity is satisfied. A curious learner cycles between these two types of situations. While Gruber and Ranganath (2019) proposed a similar cycle—the Prediction, Appraisal, Curiosity, and Exploration (PACE) cycle (p. 1015)—their focus was on the development of a neuroscientific framework. Their neuroscientific focus does not emphasize the conceptual options and multiplicity of theoretical positions that we believe will best support the machine intelligence research community. In contrast, our synthesis is designed to support exploration of the range of possibilities for effective machine curiosity.

Two key ideas are needed for understanding specific curiosity: (1) specific curiosity involves the consideration and manipulation of something the learner does not know: an inostensible concept; (2) inducing and satisfying curiosity require substantially different cognitive (and often physical) activities from a learner. Neither of these ideas are commonplace in the machine curiosity literature to date. Within this section, we set the stage by providing detail to develop the reader's intuition of these ideas, as this intuition will be needed to understand the five key properties that follow. From a computing perspective, where we aim to implement these ideas, some of the language we will use to describe our framework will be uncomfortably vague. This language include abstractions like *knowledge*, *concept*, and *object*. However, given the research community's current understanding of minds—both biological and machine—these abstractions are still necessary.[2] The challenging work of understanding mechanisms of mind is ongoing, and with progress towards solidifying these abstractions, our understanding and implementations of curiosity will improve as well.

4.2.2 Between Inducing and Satisfying Curiosity

4.2.2.1 Inostensible Concepts

As you asked each question about the narrative of your book, you were able to think about *something you wanted to know*. This experience follows the perspective put forward by Loewenstein (1994, p. 87) where specific curiosity[3] arises when a learner becomes focused on an information gap[4]—a gap between what they know and

[2] Our view on our inability to define these abstractions mirrors Frege's defense of using the word *concept* without definition (as translated by Geach; 1951, pp. 42-43): "If something has been discovered that is simple, or at least must count as simple for the time being, we shall have to coin a term for it, since language will not contain an expression that exactly answers."

[3] Loewenstein (1994, p. 92) actually expresses the information-gap perspective as a description of *specific epistemic state curiosity*, a term which delineates his concept of interest on traditional axes of types of curiosity: specific vs. diversive, perceptual vs. epistemic, and state vs. trait. The specific-diversive axis stems from the difference between taking exploratory action for the purpose of learning something specific (specific exploration) versus the purpose of relieving boredom or increasing stimulation (diversive exploration) (Berlyne, 1960, p. 80; 1966, p. 26). In this chapter, we focus on *specific state curiosity*, but primarily use the simplified term *specific curiosity* with 'state' implied throughout. See Footnote 8 for more description of the state-trait distinction. Finally, the perceptual-epistemic axis is meant to provisionally differentiate motivation relieved by perception from motivation relieved "by the acquisition of knowledge" (Berlyne, 1954, p. 180; 1957, pp. 399–400; 1960, p. 274). For the purposes of this chapter, we need not make a distinction along this axis, as the properties effectively describe either perceptual or epistemic forms. Even Berlyne, who made the original distinction, suggested that epistemic and perceptual curiosity seem to be closely related (1960, p. 280).

[4] While Loewenstein (1994) appears to have popularized the information gap as a theory of curiosity, the connection between curiosity and "gaps in information" goes back at least to Berlyne, who extended Bartlett's (1958) suggestion that thinking arises as a "reaction to a gap" (Berlyne, 1960, p. 280) to suggest that such "gaps in information" (Berlyne, 1960, p. 280; cf. Bartlett, 1958, pp. 22, 24) are similar to his own idea of *conflict*, and not only evoke thinking,

what they want to know. However, the term information gap is not well-specified (cf. Berlyne, 1960, p. 281)[5] and needs to be clarified before we can implement it algorithmically.

We can partially clarify the meaning of *information gap* via the term *inostensible concept*, as coined by Inan (2010, p. 30; 2012, p. 34).[6] An inostensible concept can be simplified as a "known unknown": something you know you do not know. If you are experiencing specific curiosity, you have an inostensible concept at the focus of that specific curiosity. In thinking about something you don't know, you are manipulating a *concept* of that something you don't know.

To make this more concrete, let's think through an example. As you were reading, perhaps you stumbled upon the following description:

> Except for an odd splash of some dark fluid on one of the white-papered walls, the whole place appeared neat, cheerful and ordinary.

If you're anything like me, you might ask yourself, "Why is there dark fluid on the wall?" An inostensible concept is implicit to this question.[7] The inostensible concept could be approximated as "how dark fluid ended up on the wall." If we knew the story of the dark fluid, our question would be answered: the concept would be *ostensible*, rather than inostensible.

but other knowledge-seeking behaviours (Berlyne, 1960, p. 280).

[5] The prevalence of the term 'information gap' in the study of curiosity, and the breadth of definitions and posited types of curiosity have led to the definition of 'information gap' occasionally being stretched beyond our area of interest in this chapter. For example, Pekrun (2019) has recently extended the term to include "a gap between current knowledge and the as yet unknown, expanded knowledge that could be gained by unspecific exploration" (p. 908) to account for *diversive curiosity*. We do not include this non-specific viewpoint in our treatment of the idea, a choice which we believe is appropriate, as multiple authors have called into question whether diversive 'curiosity' should be considered a form of curiosity at all (Markey and Loewenstein, 2014, pp. 229–230; Loewenstein, 1994, pp. 77-78).

[6] Beyond Loewenstein (1994)'s idea of an information gap, Inan's term, 'inostensible concept,' follows earlier work that describes ideas similar to the inostensible concept. Berlyne (1954) described questions evoking "mediating 'concepts' or 'meaning' responses" (p. 182).

[7] Here, we use the word *question* loosely, to refer to a feeling of recognizing an inostensible concept, because these can often be reasonably approximated as questions in the linguistic sense. We do not assume that curiosity requires linguistic abilities, and suggest that curiosity can arise prior to, or without, putting such a feeling into words.

"*How dark fluid ended up on the wall*" can be manipulated like any other concept in the mind. Note that you don't need to know how dark fluid ended up on the wall to be able to think about the inostensible concept "*how dark fluid ended up on the wall.*" Your inostensible concept (your known unknown) is composed of other concepts you are already familiar with: you have enough of an idea of what "fluids" and "walls" are and what "dark" and "ended up" mean to roughly conceptualize what it would mean to know "how dark fluid ended up on the wall." This rough concept cobbled together from concepts you already know is the inostensible concept. It is in this sense that Whitcomb (2010) indicates that curiosity does not require you "to conceive of its satisfier," rather, "curiosity requires you to conceive only of everything your questions are about" (p. 671). The inostensible concept is the concept formed from everything your question is about.

Each inostensible concept has an *object* that it refers to, also called an *inostensible referent*. For our example concept, the inostensible referent is the story of how dark fluid ended up on the wall. The term *object* is not well-defined from a computational perspective, but we can think instead about what we are trying to achieve. While we might talk about trying to acquire this object, this story, we're really thinking of a particular objective: we want to incorporate the story of how the dark fluid ended up on the wall into our knowledge base. This incorporation is the act of making an inostensible concept ostensible. There may be multiple approaches to make the inostensible concept ostensible: while we could read the next few pages of the book, we could also ask someone who has read this book before. It is computationally relevant that there are likely many different possible sets of observations one could make through different sensory apparatuses (e.g., eyes or ears) to make a given inostensible concept ostensible.

The term *inostensible concept* gives us additional power over the information gap perspective alone, as it gives us a foundation for satisfying our curiosity, for closing the gap. Our inostensible concept is defined by properties of the object of our curiosity (for example, the story must involve dark fluid ending up on the white-papered walls) that help us differentiate our particular object of interest from others.

This foundation allows us to use mental simulation—"the capacity to imagine what will or what could be" (Hamrick, 2019, p. 8)—to plan out sequences of actions we could take to make an inostensible concept ostensible. But before focusing on satisfying curiosity, we should talk about *inducing curiosity.*

4.2.2.2 Inducing and Satisfying Curiosity

Within the context of this work, we are focused on specific curiosity as *temporary*[8] and emphasize that specific curiosity is binary (it can be 'on' or 'off'). In particular, each time specific curiosity is induced, it is associated with exactly one inostensible concept at its focus.When curiosity associated with a different inostensible concept arises, it is not a continuation of the same instance of specific curiosity.

For this reason, the recognition of an inostensible concept is key to an instance of specific curiosity being induced. Specific curiosity primarily arises after processing new observations, where we allow for both observations we might consider external, like your eyes falling across the phrase, "Except for an odd splash of some dark fluid on one of the white-papered walls," and observations we might consider internal, like a thought. We refer to a set of such observations as *curiosity-inducing observations* and the situation where we make such observations as a *curiosity-inducing situation.*

There are multiple theories about what kinds of situations induce curiosity. Berlyne theorized that curiosity was induced by observations resulting in his concept of *conflict*, where two or more incompatible responses to an observation are

[8] Curiosity has been studied both as occurring temporarily, as is our focus, and as a persistent personality characteristic (Litman and Spielberger, 2003, p. 76; Pekrun, 2019, p. 908). In the literature, the former is termed *state curiosity* while the latter is called *trait curiosity.* In the study of reinforcement learning—a field central to the computational components of this text—the term *state* has a formal meaning (see Section 4.3.1) to which we will want to refer. For this reason, we avoid using the term state curiosity in this work, despite recognizing it as the accepted term. Following the *Webster's New World College Dictionary* (2014) definition of 'state' as "a particular mental or emotional condition" we will occasionally use *condition* in places the word *state* might be used in other works on curiosity.

evoked and the brain lacks the information to reconcile which is more appropriate (1960, p. 10; 1966, p. 26). Inan (2012) contended that "a certain kind of interest" (p. 126) is needed for awareness of an inostensible concept to result in curiosity. Chater and Loewenstein (2016) suggested that curiosity arises when a learner either obtains new information they can't make sense of or becomes aware of a potential way of obtaining "information that could help make sense of existing, stored, information" (p. 145). Pekrun (2019) and Peterson and Cohen (2019, pp. 812, 815) theorize that a "*sense of control* that it will be possible to close the gap" (Pekrun, 2019, p. 909) is necessary to experience the condition of curiosity. However, while Pekrun (2019) considers a sense of control to be necessary, it isn't sufficient: the theory also includes "an urge to close the gap" (p. 906) as a separate necessary component of curiosity, leaving open the question of in which situations such an urge will occur. This diversity of theories parallels the diversity of mechanisms suggested for machine 'curiosity' that we will describe in Section 4.3.3. Despite this question being a longstanding area of study, we still don't precisely understand the situational determinants of curiosity.

Once induced, specific curiosity is thought to be able to end in two different ways: either attention is distracted (a possibility we discuss further in Section 4.2.6) or curiosity is *satisfied* (Berlyne, 1954, p. 183). Drawing from the terminology of inostensible concepts used by Inan (2012), curiosity is satisfied when the inostensible concept at the focus of that instance of curiosity is made ostensible (pp. 35–36). While it may seem obvious to some readers, we wish to draw attention to the point that the curiosity-inducing situation and the curiosity-satisfying situation *must* be different.[9]

As an example where this requirement may not seem to hold, imagine you're moving through the bookstore and notice a peculiar noise from the floorboard as

[9] Why is this distinction between curiosity-inducing situations and curiosity-satisfying situations so critical to us, the authors? In both the psychological literature on curiosity and the literature on intrinsic-reward-based computational curiosity, the curiosity-inducing situation is sometimes not differentiated from the curiosity-satisfying situation, limiting our understanding of how learning occurs through curiosity. We speak more to these limitations in Section 4.3.3.2.

you transfer your weight onto it. If you experienced curiosity focused on whether your weight transfer caused the noise, you might find yourself satisfying your curiosity by repeating the same action that seemed to generate the noise the first time, transferring your weight back onto the same spot. In this case, it might seem that the curiosity-satisfying observation is the same as the curiosity-inducing observation. We see this kind of "repeated trial" action for many scientific curiosity questions (Bonawitz et al., 2010, pp. 105–106). For curiosity to be induced, however, the learner needs an inostensible concept. Critically, this means the learner knows there is something they do not know. If the curiosity-inducing situation provided the right information to satisfy this instance of curiosity, specific curiosity would not have been entered to begin with, because the known unknown would not be unknown after all. An observation of the peculiar noise as you transferred your weight does not tell you that your weight transfer caused the noise. Rather, it is the intervention and *set* of repeated, consistent observations that when you transfer your weight onto that spot, the peculiar noise reoccurs that brings you enough confidence in your understanding for your curiosity to be satisfied. However, that's not to say that the difference between the curiosity-inducing situation and curiosity-satisfying situation is never very subtle. The difference between the curiosity-inducing situation and the curiosity-satisfying situation could be as minute as a tiny movement of the eyes to obtain a new observation that satisfies curiosity.

But what does it mean for curiosity to be satisfied? Turning back to our example inostensible concept of *how dark fluid ended up on the wall,* if I were to become curious about the content of this inostensible concept,[10] my curiosity might be satisfied when I read the phrase "'The two clergymen,' said the waiter, 'that threw soup at the wall,'" printed on the following page of the book; my observation of this phrase upon turning the page constitutes a curiosity-satisfying situation.

[10] Yes, a learner can think about an inostensible concept without experiencing curiosity to resolve it (Inan, 2012, pp. 42, 125–126). The question of whether curiosity will occur or not leads us back to the open question of what kinds of situations induce curiosity.

Assuming I considered the waiter sufficiently trustworthy, I may be satisfied that I now know that *two clergymen threw soup at the wall, leaving a dark stain* is *how a splash of dark fluid came to be on the wall*. My initially inostensible concept is now ostensible, and my curiosity is satisfied.[11]

We use both the term *observation* and the term *situation* with the descriptors *curiosity-inducing* and *curiosity-satisfying* because multiple observations may be needed to enter or exit specific curiosity. For curiosity to be induced by reading the phrase "Except for an odd splash of some dark fluid on one of the white-papered walls," you likely require multiple placements of gaze on the text. Similarly, you had to transfer your weight over that peculiar-sounding floorboard multiple times to be satisfied about the causal relationship. Without a sufficient set of the right observations, curiosity won't be induced or satisfied, respectively. Using the term *situation* allows us to refer to the moment of the final observation while recognizing that more observations beyond the final one may have been needed.[12]

The primary goal of this preliminary section was to clarify specific curiosity as a short-term condition and to differentiate the terms *inostensible concept*, *curiosity-inducing situation* and *curiosity-satisfying situation*. While these terms are not all broadly used, in this work they are meant to help us be specific, as the concepts they refer to have sometimes been described interchangeably in the literature. For example, Dember and Earl (1957) used the term *goal stimuli* as both curiosity-inducing (p. 92) and curiosity-satisfying (p. 91). Similarly, Dubey and Griffiths

[11] An illuminating description of the satisfaction of curiosity has been put forth by Inan (2012, p. 135), where curiosity is satisfied "only when the curious being gains some new experience that [they believe] to be sufficient to come to know a certain object as being the object of [their] inostensible concept," and the interested reader might look to Inan's Chapter 9 for more detail. The curious reader, on the other hand, who simply wants to know where our example inostensible concept was lifted from can instead be directed to *The Innocence of Father Brown* by G. K. Chesterton (1911).

[12] We considered following Isikman et al. (2016) in their use of the term *curiosity-evoking events* rather than *curiosity-inducing situations*. However, we felt that the connotations of the word *event*, while allowing for the inclusion of multiple observations, suggests that the observations happen "all at once"—temporally close together—while we mean for *situation* to imply that a complete set of curiosity-inducing observations might occur across more time than might be considered a single event.

(2020, p. 463) use the term *stimulus* as offering both the curiosity-inducing and curiosity-satisfying observations (e.g. a trivia question and its answer considered as one stimulus without differentiating which of the two the learner is seeking). In this preliminary section, we have contributed an argument for the importance of separating what occurs when curiosity is induced from what happens when curiosity is satisfied. We believe that future work—both the study of biological curiosity and the design of machine curiosity—can proceed with improved clarity with this separation recognized.

4.2.3 Directedness Towards Inostensible Referents

> Our first key property is *directedness towards inostensible referents.* When specific curiosity is induced, the learner is motivated to take actions directed towards satisfying their curiosity.

Directedness towards inostensible referents is inherent to many of the experiments used for studying curiosity. One of the most common experimental paradigms for this purpose is the *trivia task*. In trivia tasks, experimenters attempt to induce curiosity using a trivia question, which can theoretically be satisfied by showing the associated answer to the question. Many trivia task experiments require participants to take specific actions to gain access to a curiosity-satisfying situation, like paying a token (Kang et al., 2009, p. 970), breaking a seal to open an envelope (Litman et al., 2005, pp. 565, 567), or pressing a key to indicate they would like to wait a short period to see the answer rather than skip ahead to another question immediately (Marvin and Shohamy, 2016, p. 268; Dubey and Griffiths, 2020, p. 463). When curious, participants generally took the specified actions to gain access to the inostensible referent. Even using an experimental setup that simply displayed the answer after a delay, Baranes et al. (2015, p. 81) showed that, when curious, participants' behaviour was directed in anticipation of receiving the answer, as they moved their gaze to where the answer would be shown.

Another common experimental paradigm for studying curiosity requires participants to take an action to “uncover” a picture. For example, Nicki (1970, p. 390) required participants to press a key if they wanted to see an in-focus version of a just-seen blurred picture, while participants studied by Loewenstein et al. (1992, as described by Loewenstein, 1994, p. 89) and Hsee and Ruan (2016, p. 663) needed to click a computer mouse if they wanted to remove boxes occluding pictures of animals.

While the above paradigms elicit simple actions to satisfy curiosity, experimenters have also used more complex situations requiring participants to take more extended sequences of directed action to acquire curiosity-satisfying information. Polman et al. (2017, pp. 819–820) observed an increase in stairwell traffic when they placed a curiosity-inducing situation (a placard with a trivia question) by an elevator along with the explanation that the answer could be found in the nearby stairwell.

However, all of these experimental paradigms largely make use of what Polman et al. (2017, p. 818) call a “curiosity appeal,” where the experimenter or the context induces curiosity and offers a promise that a particular sequence of actions will lead to a curiosity-satisfying situation. However, outside of experiments, there isn’t always an obvious plan to follow to satisfy one’s curiosity. There have been multiple suggestions of a theoretical connection with creativity (Gross et al., 2020, pp. 77–78), at least in part because curiosity appears to often require the creation of non-obvious plans of actions to acquire appropriate curiosity-satisfying observations (Hagtvedt et al., 2019, p. 2). While the theory that curious learners can generate complex, adaptable plans of action to satisfy their curiosity remains understudied, this idea remains a strong starting point for thinking about how machine learners might demonstrate the directedness characteristic of specific curiosity.

4.2.4 Cessation When Satisfied

Our second key property is *cessation when satisfied.* This property refers to the instance of specific curiosity ending immediately once curiosity has been satisfied, so the learner's motivation is no longer directed towards the same kind of observations that were or would have been curiosity-satisfying when the learner was still curious.

Once a learner has achieved the goal of transforming an inostensible concept into an ostensible one, they do not need to seek the same curiosity-satisfying situation again. You didn't repeatedly read the page describing how the protagonist escaped their brush with death; once you knew the answer, curiosity did not drive you to experience it again. Instead, in the process of transforming that particular inostensible concept, you found yourself with a new question as to the relationship of the protagonist with their mysterious saviour, and while curiosity motivates you to investigate the same book, you're no longer interested in the preceding pages, only the following ones. Theories of specific curiosity regularly reference the satisfaction of curiosity (Loewenstein, 1994, p. 92; Schmitt and Lahroodi, 2008, p. 129; Gruber and Ranganath, 2019, p. 1015), and some authors consider the cessation of curiosity when "the information gap is closed or the conflict is resolved" inherent to curiosity's definition (Renninger and Hidi, 2016, p. 45).

This understanding has unsurprisingly influenced the empirical study of curiosity. A number of studies have explored differences in behaviour or physiological changes when curiosity is satisfied. On the behavioural side, results shared by (Wiggin et al., 2019, p. 1194) suggest that when curiosity is left unsatisfied, humans are more likely to make indulgent choices—choices that provide short-term pleasure but are not in the chooser's long-term interest, like "the consumption of luxuries, hedonics, and other temptations" (Wiggin et al., 2019, p. 1195). Fastrich and Murayama (2018) similarly manipulated whether participants were provided with curiosity-satisfying observations or not, but did not find any significant difference in participants' rating of curiosity (pp. 13, 15) or willingness to bid to satisfy their curiosity on the next, unrelated trivia question in a sequence (pp. 14–15). On the physiological side, in an fMRI (functional magnetic resonance imaging) experiment performed by Jepma et al. (2012), participants were shown blurred pictures, sometimes followed by the corresponding clear picture, sometimes followed by an unrelated clear picture. In the condition with the corresponding clear picture—where curiosity induced by the blurred picture was thought to be relieved—Jepma et al. (2012) found both striatal and hippocampal activations were stronger than in the unrelated clear picture condition. Similarly, Ligneul et al. (2018) performed an fMRI experiment in which participants were shown trivia questions, sometimes followed by the corresponding answer, sometimes followed by an unrelated filler screen. In the condition with the corresponding trivia answer, Ligneul et al. (2018) found that observing the answer yielded a ventral striatal response in the brain. In prior work, the striatum has been implicated in both pain relief and reward responses, while the hippocampus has been implicated in memory. These results align well with the theory that specific curiosity is an uncomfortable experience that can be relieved as well as with the evidence showing that curiosity improves memory.

Despite evident interest in physiological changes when curiosity is satisfied, there has been minimal empirical work to confirm that the behaviour and motivation directing a learner towards curiosity-satisfying observations (observations that would render the previously-inostensible concept ostensible, if it were still inostensible) do indeed cease when satisfied. A notable exception is an experiment performed by Wiggin et al. (2019). In this experiment, all participants were shown a blurred picture, but while the participants in one condition were then shown the clear version of the same picture, participants in the other condition were not. Participants in both conditions responded to the "10-item state curiosity scale of the State-Trait Personality Inventory (STPI) developed by Spielberger and Reheiser (2009)," and participants who had not been shown the clear picture rated higher on the scale in terms of "the intensity of feelings and cognitions related to curiosity" (p. 1198).

Note that *cessation when satisfied* contrasts with the properties of behaviour motivated by extrinsic rewards. Extrinsic rewards, in the terminology of psychology, are material outcomes of an activity like obtaining food, water, or money (Morris et al., 2022, p. 1801). Extrinsic rewards motivate behaviour repeatedly leading to the same target (Gruber and Ranganath, 2019, p. 1014). For example, animals confined to a box with a lever will learn to repeatedly press the same lever if pressing it results in the mechanism providing the same food reward (Skinner, 1963, p. 504). This kind of directly repetitive behaviour is not exhibited towards curiosity-satisfying observations, as specific curiosity is expected to provide no further motivation towards the same target if the target satisfies the learner's curiosity (Gruber and Ranganath, 2019, p. 1014).

One view that may appear to contradict cessation when satisfied is that proposed by Fastrich and Murayama (2018), who have suggested that curiosity may persist even after curiosity-satisfying observations have been provided. In their experiment, they found participants were more likely to demonstrate curiosity for the answer to a trivia question in a sequence of trivia questions if they had been curious for the answer to the preceding question in the sequence—whether or not they had been provided with the answer to that preceding question. Fastrich and Murayama (2018) explain their findings as suggesting that curiosity *persists* even after the associated answer is provided, and curiosity can transfer to a temporally contiguous information gap. They call this effect the curiosity carry-over effect. However, their results do not actually contradict the property of cessation when satisfied, as in our terminology, while the present *instance* of curiosity ceases when the associated inostensible concept becomes ostensible, this does not imply that a learner is unlikely to become immediately curious again, but for a different inostensible concept. Fastrich and Murayama's (2018) results rather suggest that human learners remain physiologically "ready" for curiosity for a time interval once curiosity has been induced.

Further, we can clarify that the property we are calling "cessation when satisfied" is not the same as the "knowledge satiation" described by Murayama et al. (2019), in which a learner feels that they "completely understand the topic" (p. 882). In our terminology, satisfaction occurs at the moment of making ostensible the inostensible concept associated with the current instance of specific curiosity—answering a single specific question—and does not imply a feeling of completely understanding an entire topic.[13]

4.2.5 Voluntary exposure

Our third key property is *voluntary exposure.* This property refers to a preference for curiosity-*inducing* situations, and that learners act on that preference to purposefully make themselves curious.

> *"An active striving to encounter new experiences, and to assimilate and understand them when encountered, underlies a huge variety of activities highly esteemed by society, from those of the scientist, the artist and the philosopher to those of the polar explorer and the connoisseur of wines."* Berlyne (1950, p. 68).

The experience of unresolved curiosity is inherently frustrating, as well-demonstrated when you finish a chapter that ends with a cliffhanger, but know that reading the next chapter would take you past your bedtime. Curious humans modify their behaviour to alleviate the feeling of unresolved curiosity.[14] Despite the *aversive quality* or discomfort associated with curiosity,[15] humans voluntarily expose them-

[13] *Topics* are understood as categorizations of related knowledge and activities (Krapp, 1994, p. 83; Renninger and Hidi, 2016, p. 11).

[14] FitzGibbon et al. (2020, pp. 21-22) provide an overview of the lengths people will go to to satisfy their curiosity, including paying for non-instrumental information (information that provides no benefit in terms of traditional extrinsic rewards, like money or food) or exposing themselves to pain or risk.

[15] The idea that being in a condition of curiosity is uncomfortable has sparked some debate. Silvia (2006, pp. 50, 190–191) has argued that the idea of curiosity as aversive is a longstanding assumption with little supporting evidence popularized by Loewenstein's seminal work (1994). The difficulty in disentangling evidence of an aversive quality to curiosity from other possible

selves to curiosity, choosing to pick up mystery novels and puzzles *because* they will pique curiosity (Loewenstein, 1994, p. 76). We aim to capture this tendency with our third property, *voluntary exposure*.

We want to remind you of the separation of curiosity-inducing observations from curiosity-satisfying observations as we introduced in Section 4.2.2.2. While your new book contains examples of both curiosity-inducing observations and curiosity-satisfying observations, if they are associated with the same inostensible concept, then they must be in different places in the book. Re-reading the passage about the butler's shifty behaviour during the officers' interrogation (a plausible curiosity-inducing situation) will not tell you what the butler has done that they don't want the officers to be aware of (the inostensible concept). It is instead in reading the passage where the officers confront the butler about damning evidence of the butler's theft of thousands of dollars worth of their employer's property (a curiosity-satisfying situation) that your curiosity about their behaviour is satisfied.

Voluntary exposure is perhaps best observed via the vast amount of time and money that people across the world devote to activities associated with curiosity. Two of the most obvious activities include engaging with puzzles and mysteries, both of which are hugely popular activities. As examples, the puzzle genre raked in the second-highest total revenue across mobile game genres in the United States and Canada in 2021 (NPD Group, 2022) and the mystery genre has held an enduring share of entertainment production over the years in multiple countries (Knobloch-Westerwick and Keplinger, 2006, pp. 193–194). While mysteries and puzzles are some of the most obvious curiosity-generating activities, narrative elements that induce curiosity are pervasive across genres of storytelling (Bermejo-Berros et al., 2022, p. 12). Since storytelling features across media (in-

motivating factors still stands in more recent work (Murayama et al., 2019, pp. 886–887). Indeed, recent accounts of how emotions are constructed in biological brains and bodies suggests that the experience of curiosity may vary by culture (Barrett, 2017, p. 149), and individual differences implicated in interpretation the experience of curiosity may account for some of the controversy. In the computational part of this work (Section 4.3.3.4), we take inspiration from the aversive quality of curiosity, but our computational analogue of aversive quality is not needed for our computational learner to demonstrate recognizably curious behaviour (Section 5.5.1).

cluding books, television, movies, games, and news), this single example of activities demonstrates a huge swatch of human life voluntarily engrossed in curiosity-inducing activities at any given time.

While humans, as a group, seem to be drawn to voluntarily expose themselves to curiosity-inducing activities, the type of curiosity-inducing activity seems to vary from individual to individual. While one person might be drawn to formulating mathematical proofs, another might prefer crosswords or language puzzles, and another might instead spend time on puzzles of shape and geometry, and yet another may select for mystery novels. All of these individuals demonstrate voluntary exposure to curiosity, yet they are selective (compare Krapp, 1994, pp. 92–93). This selectivity is a starting point for our hypothesis that voluntary exposure might be learned over time, as an individual learns a preference for curiosity-inducing situations related to their preferred topics, domains, or puzzle styles. We will discuss this preference further in Section 4.2.7, with the property of coherent long-term learning.

4.2.6 Transience

Our fourth key property, *transience*, refers to how an instance of curiosity ends when attention is distracted or diverted.

As you went to pay for your book, you became intensely curious to learn the current news of a Hollywood star's familial strife, but only while you paid attention to the magazines placed temptingly close to the checkout. Once you've torn yourself away to pay, your mind is happy to resume other functions, so once you're out the door and on your way home to start your new book, the star's struggles are as good as forgotten (example inspired by Loewenstein, 1994, p. 76).

When attention is distracted, the instance of curiosity ends, and this property is referred to as *transience* (Loewenstein, 1994, pp. 86, 92).[16] While some authors

[16] The properties *cessation when satisfied* and *transience* are similar in that both refer to the condition of curiosity ending, but we have separated them to better align with how the terms are

have written about curiosity as though it can be sustained over long periods, even over years (e.g., Engel, 2015, pp. 7-8), transience of curiosity is a frequently recognized property.[17]

Like cessation when satisfied, transience appears prominently in theories of curiosity and intuitive examples. Early on, Berlyne (1954) noted that curiosity can end if distraction occurs (p. 183). More recently, the property of transience appears to have shaped Loewenstein's (1994) information gap theory: one of the reasons that *attention* to an information gap is key to the formulation is that curiosity is thought to end when attention is distracted (p. 92).

In recent experiments, Golman et al. (2021, Experiments 1B and 2B: "Salience") had participants solve puzzles (1B) or identify emotions associated with facial expressions (2B). Golman et al. (2021) manipulated the amount of time before participants were offered the solution to one of the more challenging puzzles if they failed to solve it (1B) or the amount of time before participants were offered the chance to view their score on the facial emotion recognition test (2B). Participants who were immediately offered the opportunity to satisfy their curiosity were more likely to click multiple times or complete an unrelated task to obtain the solution/score than those who were offered the same opportunity 24 hours later. Distanced from the original context of a concerted effort to solve the puzzle or test questions, participants showed less impetus to acquire the solution or scores. While this experiment is only a partial demonstration of transience, since by offering the solution/score, Golman et al. (2021) draw attention back to the inostensible concept, this decrease in demonstrated curiosity suggests that for many participants, curiosity has ended and this simple return of attention is insufficient to rekindle curiosity.

used in the literature. There may also be ways the mechanisms for each property should offer different effects. For example, there are some theories that the satisfactory resolution of curiosity is actively rewarding (Shin and Kim, 2019, p. 863; Murayama et al., 2019, p. 879).

[17] While the term *transience* was used in Loewenstein's (1994, p. 76) seminal paper on the information-gap theory of curiosity, the property is sometimes simply referred to as dissipation or decline of curiosity, but specifically that caused by the distraction of attention (Markey and Loewenstein, 2014, p. 232; Shin and Kim, 2019, p. 856; Dan et al., 2020, p. 152).

Our use of the term transience refers to the specific behaviour of directedness towards inostensible referents ending when attention is distracted. We mentioned in Section 4.2.4 that Wiggin et al. (2019) did find evidence of a difference in behaviour when curiosity is left unsatisfied versus not.

4.2.7 Coherent Long-Term Learning

The property of *coherent long-term learning* refers to how specific curiosity works in concert with other mechanisms of attention and value to orient the learner towards inostensible concepts related to the learner's prior knowledge.

In this work, we have attempted to be very careful to model specific curiosity as a short-term motivational effect that begins when curiosity is induced and ends when curiosity is satisfied or when attention is diverted. However, curiosity is choosy. Moment-to-moment, humans are faced with a galaxy of unknowns, but the mechanisms of curiosity choose carefully—and it is not as though curiosity simply chooses the most readily available unknown; rather, curiosity often sends us out on a temporally extended plan to make our inostensible concept ostensible. Importantly, curiosity seems to be biased towards learning ideas related to the learner's pre-existing background knowledge (Wade and Kidd, 2019, p. 1377).

Zurn et al. have recently proposed a connectional account of curiosity (2022), explicitly critiquing the 'acquisitional' metaphors commonly used for curiosity in recent decades. Curiosity is often thought to drive us to acquire information (p. 259-261). The connectional model instead emphasizes curiosity as building connections between ideas (p. 261). The connectional account aligns with Wade and Kidd's (2019) notion that a learner's level of curiosity is well-predicted by their metacognitive estimates of their own knowledge (p. 1380). If a learner recognizes metacognitively that they have existing knowledge related to a potential learning opportunity, they are well-prepared to make that connection and integrate it into their knowledge base.

By including the property of coherent long-term learning in our list of key properties, we are formally emphasizing the importance of specific curiosity's integration with the learner's current knowledge base. In humans, this integration may occur via the mechanism of individual interest. Individual interest refers to a predisposition to repeatedly engage with a class of content, where a class of content usually refers to a domain or category of knowledge, objects, or ideas. The class of content may be thought of as broad as 'science' or 'playing tennis' (Renninger and Hidi, 2016, p. 6) or more narrow, like 'approaches to machine curiosity that offer the benefits of human curiosity'—the best description will be highly individual and depend on the learner's organization of their knowledge. The connectional account of curiosity can help us think of a class of content as a set of ideas that have been connected in the learner's mind, woven together by the relationships that the learner recognizes among them. Individual interest is distinguished from other motivational concepts by two components: stored knowledge and stored value, both for the particular class of content.

Curiosity → Individual Interest: Curiosity may shape individual interest by increasing both knowledge and value for content areas that a learner experiences curiosity in. By driving learning, curiosity increases knowledge. Rotgans and Schmidt (2017) have provided initial evidence that individual interest is a consequence of learning, showing small but significant effects that growing knowledge results in increased individual interest. Indeed, the process of continually developing knowledge (availability of "cognitive challenges") in the content area of interest is required to maintain an individual interest (Renninger, 2000, p. 379). Curiosity provides impetus for a process of continually developing your knowledge.

Curiosity may also play a role in increasing value, as individual interest in a class of content reflects high levels of not only knowledge but value for the content relative to other classes of content (Renninger, 2000, p. 375). Experimentally, Ruan et al. (2018, pp. 559, 566) found that, when subjects experienced the resolution of curiosity about particular well-known brands, they developed increased positive attitudes towards those brands. Such increases in positive attitudes may reflect increased value. In one experiment, Ruan et al. (2018) teased some participants with an animation of a gift card gradually being revealed from an envelope (so these participants needed to wait to find out where the card could be spent) and showed other participants the whole gift card immediately (so these participants immediately knew the card could be spent at Target) (p. 564). When surveyed after, the participants who had to wait for the gift card to be pulled from the envelope had a more positive average attitude toward Target (p. 565). Ruan et al. (2018) also found similar results with different manipulations creating and resolving uncertainty about different brands. Further research into this effect is needed, but we hypothesize that, more generally, learners may develop increased positive attitudes towards topics associated with the inostensible concept when curiosity is created then resolved.

Individual Interest → Curiosity: Individual interest may direct curiosity by directing a learner's attention. Individual interest schools our attention onto aspects of what we perceive that we relate to our pre-existing interests. Renninger (2000) has described individual interest as acting like a filter on a learner's perception (p. 380). For example, I have an individual interest in curiosity, so when a character in my book says, "No, I'm not curious," my attention is drawn to how curiosity fits into the situation and how my understanding of curiosity explains or fails to explain the character's lack of motivation. A learner with other individual interests would likely focus on other aspects of the same scene. In this way, individual interest can bias curiosity towards inostensible concepts that connect with existing knowledge.

Curiosity ⟷ Individual interest: Given the early evidence we have described, we hypothesize a bidirectional relationship between curiosity and individual interest. Related bi-directional proposals have been previously raised by

Arnone et al. (2011, p. 186) and Engel and Randall (2009, p. 185). There has been a recent surge in effort to understand curiosity's relationship with interest (e.g., Peterson and Hidi, 2019). More recent work has focused on the direction that experiences of curiosity may build individual interest (Shin and Kim, 2019, pp. 863–864; Peterson and Cohen, 2019, p. 814) and substantial work remains to develop a complete account. However, a relationship with some mechanism to re-engage learning related to prior knowledge is likely necessary to provide specific curiosity with the property of coherent long-term learning.

***

The property of coherent long term-learning, the last of our five properties, closes the loop of how curiosity can guide a learner over a lifetime. Our list of properties began with the impetus to satisfy our curiosity in a specific, directed way (1, Directedness), an effect that ends relatively quickly, either via being satisfied (2, Cessation when satisfied) or via attention being diverted (4, Transience). Our final two properties speak to aspects of curiosity relevant to a learner's entire lifetime: learners should seek curiosity-inducing situations (3, Voluntary exposure) and curiosity should build up knowledge and value, biasing the learner's future experiences of curiosity towards learning opportunities to build on what they already know (5, Coherent long-term learning).

In this section, we described five properties of curiosity, and in particular, of specific curiosity (defined by Loewenstein (1994, p. 87) as "an intrinsically motivated desire for specific information"). While specific curiosity is associated with other properties, particularly intensity, association with impulsivity, and a tendency to disappoint when satisfied (Chater and Loewenstein, 2016, p. 17; citing Loewenstein, 1994), the set of five properties we described above are expressly valuable to a learner. In the next section, we will provide the context of existing approaches for machine curiosity and leverage this context to argue the need for these five properties in Section 5.6. As researchers work to design curious machine agents, we believe that these properties are ones we should attain.

4.3 Specific Curiosity for Machine Intelligence

4.3.1 Reinforcement Learning

In the remainder of this chapter, we will rely on language and a choice of framework drawn from reinforcement learning (RL). In this subsection, we introduce the framework and define some of the language that we will help us both to express the differences between the key properties proposed in this chapter and existing related methods and to describe our case study in Section 4.3.3.4.

One way of representing an agent's experience of the world in a reinforcement learning framework is as an alternating sequence of observations and actions marked by time. We think of time as discrete, and at each time step, a single observation is made and a single action is taken, resulting in a sequence of the form

$$O_0, A_0, O_1, A_1, ..., O_t, A_t, O_{t+1}, A_{t+1}, ... \tag{4.1}$$

The agent has a set of actions, $\mathcal{A}$, available to them, so the action taken at time t is denoted $A_t \in \mathcal{A}$. Each observation, denoted O_t for the observation at time t, provides (possibly partial) information about the current *state* of the environment, S_t. Informally, the state of the environment is the situation that the agent finds itself in; depending on the situation (state), the agent's choice of action will have different effects and could lead to different next situations (Sutton and Barto, 1998, pp. 7, 47). If I'm standing in the open doorway of the bookstore, a step forward could lead me into the splendorous observation of mountains of books; if I'm standing in front of the closed door, a step forward might lead to me bumping my nose.

In classical reinforcement learning, each observation from $t = 1$ onwards includes a numerical reward signal $R_t \in \mathbb{R}$. The agent must choose actions to maximize how much[18] reward accumulates over time, a quantity called the *return*, G_t. There are several possible definitions of return, but for simplicity in this chapter, we use *discounted return*,[19] which relies on a *discount rate*, $\gamma \in [0, 1)$, to place less value on rewards the further into the future they occur.

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{4.2}$$

It is common for a reinforcement learning agent to keep a running estimate of how valuable different parts of the world are so that they can map their representation of the current state S_t (usually formed using the present observation O_t) to an estimate of its value and use these estimates to attempt to accumulate more value. A *value function*, denoted v_π, is defined as the expected return moving forward from that state, assuming the agent follows policy π.

$$v_\pi(s) := \mathbb{E}_\pi \left[G_t | S_t = s \right] \tag{4.3}$$

We denote an agent's estimated value function as V. Estimated value functions intuitively describe agent preferences: states with higher estimated value are preferred by the agent. In this way, we could algorithmically express the property of voluntary exposure as the agent estimating increased value for situations that are expected to induce curiosity.

While there are multiple ways that a reinforcement learning agent might maintain an estimated value function, one of the most important approaches is called *temporal-difference* (TD) learning (Sutton and Barto, 2018, Ch. 6). When the agent transitions from state S_t to state S_{t+1}, receiving reward R_{t+1}, we can form a new estimate for $V(S_t)$: $R_{t+1} + \gamma V(S_{t+1})$. However, since we may not always

[18] Grammatically, you may have expected "how many rewards" instead of "how much reward," but within reinforcement learning, each reward can be a different real number, and we are concerned with maximizing *return*, a function involving the sum of rewards over time. For instance, while the learner receives a reward at each time step, one reward of 76.243 is going to be more desirable than accumulating three rewards of −8, 0, and 0.3, so "how many rewards" wouldn't reflect the meaning of return.

[19] Sutton and Barto (2018, p. 55) offer further intuition about this choice.

arrive in the same next state or receive the same reward when leaving state S_t, we usually only want to shift our estimate of $V(S_t)$ towards $R_{t+1} + \gamma V(S_{t+1})$ by a small step. We use a parameter α, known as the *step size*, to determine the amount of shift, multiplying α by the difference between the new estimate and the old. This difference, the *TD error*, denoted δ, is defined as

$$\delta := R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \tag{4.4}$$

The simplest TD method (and the approach we take in the case study described in Section 4.3.3.4) updates the estimate of the value of state S_t upon transitioning from S_t to S_{t+1} and receiving a reward of R_{t+1} as follows:

$$V(S_t) \leftarrow V(S_t) + \alpha\delta \tag{4.5}$$

The reinforcement learning framework, with these ideas of the state of the world having a specific value to a learning agent and that an agent's behaviour in the future can influence how valuable the current state is to the agent, has been used to study not only what kind of algorithms make the best choices in such a problem setting, but also how humans and other animals choose actions, especially to consider which algorithms seem to best replicate biological decision-making.

Within the reinforcement learning framework, there has been a long-standing assumed or hypothesized link between curiosity and *exploration* (Fox et al., 2020, p. 109)—some researchers hope that the study of curiosity holds the solution for the exploration–exploitation dilemma. The exploration–exploitation dilemma is a long-studied challenge of reinforcement learning (Sutton and Barto, 2018, p. 3). Classically, reinforcement learning problems have an optimal solution: a policy of behaviour that can obtain the maximal return. However, the learner doesn't start out knowing what the right policy is. To learn a good policy, the learner has to balance taking an action that has offered the best value in their experience so far (exploiting what they've learned) against taking an action they haven't tried enough times to be certain of the actions' value (exploring alternative possibilities).

In the following section, we will describe a number of existing methods inspired by curiosity, and many of these methods are explicitly designed to improve exploration. In this work, however, we do not assume that curiosity should contribute to exploration in this return-driven sense. Indeed, curiosity might be most interesting in the context where the learner does not have a persistent objective.

4.3.2 Goals in Reinforcement Learning[20]

The word *goal* lives a conflicted life within the terminology of reinforcement learning. One traditional use of the word goal is specifically in reference to maximizing return (Sutton and Barto, 2018, pp. 6, 53), in reference to the *reward hypothesis*, stated by Sutton and Barto (2018, p. 53) as:

> *That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward).*

And yet, when speaking to the intuition around reinforcement learning, there is longstanding use of the the word goal to refer to abstract accomplishments like *grasp a spoon* or *get to the refrigerator* (Sutton and Barto, 1998, Ch. 1.2; Sutton and Barto, 2018, p. 5). If we assume the reward hypothesis holds for human learners, the reward signals generated in our bodies were evolved over millions of years to shape our behaviour towards such goals, and it isn't obvious on what basis our reward signal is generated (Sutton and Barto, 2018, p. 469).

The use of *goal* as specifically related to maximizing return is inspired by the way *goal* can be used in the context of human and animal motivation and behaviour, but defining *goal* this way is limiting. More recently, taking a computational approach has led authors like Grace and Maher (2015) to define specific curiosity as "the search for observations that explain or elaborate a particular goal concept" (p. 262). We suggest that further consideration of what is meant

[20] Some of this text is adapted from Ady et al. (2022a) and is under review at the *Journal of Artificial Intelligence Research*.

by goal is needed when approaching the relationships between objectives as they relate to both environment state and knowledge state, as described above, and when attempting to broker the relationship between human and machine curiosity literature.

4.3.3 Computational Approaches Inspired by Curiosity: Intrinsic Rewards

The argument that reinforcement learning is an appropriate framework for computational approaches to curiosity has been embraced by many authors over the past few decades. Mechanisms inspired by curiosity have varied widely, with many using the amount of error in their machine-learning predictions (‘prediction error’) or ideas from information theory in the interests of simulating other constructs, like confidence (Schmidhuber, 1991a), learning progress (Oudeyer et al., 2007, p. 269), surprise (White et al., 2014, p. 14), interest/interestingness (Gregor and Spalek, 2014, p. 435; Frank et al., 2014, pp. 5-6), novelty (Gregor and Spalek, 2014, p. 435; Singh et al., 2004, pp. 1, 5), uncertainty (Pathak et al., 2017, pp. 1-2), compression progress (Graziano et al., 2011, p. 44), competence (Oddi et al., 2020, pp. 2417-2418), and information gain (Bellemare et al., 2016, p. 4; Houthooft et al., 2016, pp. 2-3; Still and Precup, 2012, p. 139; Frank et al., 2014, pp. 5-6).

Most existing methods inspired by curiosity are centred on generating special reward-like signals, called *intrinsic reward*. In this section, we provide detail on intrinsic-reward methods, including their benefits and limitations. Specific curiosity may address some of the limitations of intrinsic reward and offer a better choice for some applications of machine curiosity. Our list of properties provides a specification for computational approaches that aligns with an interdisciplinary understanding of curiosity based in the literature, in part inspired by observing a poor alignment between intrinsic-reward methods and biological curiosity.

Computational intrinsic rewards are a spin-off of the formalism of reward (R_t), a term described in the preceding subsection. As you may recall, reward is given as part of the observations the agent makes of the environment. The designer of the agent's learning algorithm cannot change the reward and so their algorithm must solve the optimization problem as it stands. Intrinsic rewards, on the other hand, are defined as part of the agent's learning algorithm (they are 'intrinsic' to the agent), behaviour can be optimized optimized for intrinsic rewards just as it could be for the original reward signal. For clarity, the original reward signal is often called *extrinsic reward* to distinguish it from *intrinsic reward* in the intrinsic reward literature.[21]

Intrinsic reward is usually either (a) treated as a reward bonus added to the extrinsic reward provided by the environment, or (b) treated as the only reward signal, with the learner effectively ignoring any reward provided by the environment. If the intrinsic reward at time t is written R^I_t, then standard algorithms for maximizing return can be used on the new, modified return (compare with Equation 4.2):

$$\text{(a)} \quad \sum_{k=0}^{\infty} \gamma^k \left(R_{t+k+1} + R^I_{t+k+1}\right) \qquad\qquad \text{(b)} \quad \sum_{k=0}^{\infty} \gamma^k R^I_{t+k+1} \qquad (4.6)$$

While many intrinsic reward designs have been inspired by curiosity, there is a wider body of literature about intrinsic rewards that doesn't always reference curiosity. However, many intrinsic rewards in this wider literature are designed for the same reasons that researchers often want to include curiosity in their algorithms, like improved exploration of the environment or allowing for self-directed learning. For this reason, readers interested in learning more about current machine curiosity methods may wish to explore the larger literature on computational intrinsic rewards.[22]

To create a computational form of specific curiosity, we essentially want an

[21] Use of the terms intrinsic reward and extrinsic reward in computational reinforcement learning, as described here, differs from its use in psychology. Oudeyer and Kaplan (2007, pp. 1–4, 12) offer a discussion of how the terms extrinsic, intrinsic, external, and internal reward and motivation are used within the contexts of psychology versus computational systems.

[22] Oudeyer and Kaplan (2007), Baldassarre and Mirolli (2013), and Linke et al. (2020), offer overviews and surveys of intrinsically motivated computational systems.

algorithm to exhibit the properties of specific curiosity. This means that a robot or computer running such an algorithm would take actions reflecting these properties. In this section, we want to discuss what infrastructure is needed to make such an algorithm possible, and provide the preliminaries for the framework that we argue is most appropriate—reinforcement learning.

For an algorithm to exhibit the properties of specific curiosity, the machine running the algorithm should be able to decide how to act in the world and exhibit preferences about its available choices; we see this especially in the properties of *directedness* and *voluntary exposure*. Above the other properties, directedness involves a preference for a sequence of actions expected to satisfy curiosity, and voluntary exposure involves a preference for curiosity-inducing situations. It is valuable if the agent can learn what kinds of situations induce curiosity and which sequence of actions might lead to those particular situations and develop the appropriate preferences through learning. The capability to learn preferences and act on them is the primary reason we consider reinforcement learning to be especially appropriate for designing algorithms that reflect machine curiosity, as reinforcement learning centres around algorithms that use access to sensations of their environments (at least partial) to choose actions that affect the environment around them (Sutton and Barto, 2018, p. 3). Within the framework, instances of reinforcement learning algorithms are often called *agents*, because they have the agency to shape their own experience in the world and learn from their actions. This quality makes the framework well-suited for the design of machine curiosity algorithms.

4.3.3.1 Benefits of Intrinsic-Reward Approaches

Intrinsic rewards have been very useful for increasing exploration on some important testbeds (Pathak et al., 2017; Burda et al., 2019b; Bellemare et al., 2016), and they have been used to perform well on problems where the objective outcome metric is unavailable to the agent (Linke et al., 2020) or to generate developmental behaviour (Oudeyer et al., 2007). In intrinsic-reward approaches, the agent is rewarded for being in interesting (novel, surprising, uncertainty-reducing, etc.) states. These rewards encourage the agent to stay in or return to the same state repeatedly. Repeatedly visiting the same state has some important benefits for learning.

1. **Offers a simple way to recognize and remain on an exploration 'frontier.'** Repeatedly visiting states that have not yet been visited many times can mean staying on the frontier of a part of the world that the agent has yet to explore. *Frontier* here refers to states of the world from which, if the agent takes a particular action, they can end up in a state of the environment that they have never experienced before. If the agent occasionally takes a random action,[23] staying on a frontier makes it more likely that it will end up visiting unexplored parts of the world via such a random action[24] than if the agent largely stayed in the middle of the part of the world already explored.

2. **Offers a way to check if an action results in a consistent reaction.** Another perceived benefit of doing the same thing repeatedly is to check for consistency. In Section 4.2.2.2, we described repeating a test of the floorboard to decide if it was the source of a peculiar noise. Similarly, for the experimental section in Chapter 5, we completed multiple trials of each experiment because we are interested in patterns that hold over time, rather than one-time outliers. This benefit relates to an important assumption in many uses of reinforcement learning: that the world is a little bit random.

[23] Many reinforcement learning agents occasionally take random actions (Hauser, 2018, p. 7). Such agents learn about the world and develop estimates about which actions will let them accumulate the best return, and take a best action (according to that metric) most of the time. However, the agent occasionally takes one of its other possible actions, just in case its estimates were wrong. This design element to take a random action is considered a type of exploration strategy, and has good properties for ensuring the agent tries all possible actions from any state an infinite number of times (Sutton and Barto, 2018, p. 103), at least if the agent has infinite time! Two popular examples are ϵ-greedy exploration and soft-max/Boltzmann policy exploration.

[24] Or, with carefully designed search control, the agent could be biased to take actions it has never taken from a given state before.

When we want to estimate the value of a particular state, we are really interested in an *average* so we must observe a state multiple times to form a reasonable estimate. Because of this assumption, many exploration methods try to ensure the agent visits each state of its domain multiple times.

One way algorithm designers have encouraged agents to make exploratory visits to each state multiple times is through intrinsic rewards that decay over visits. This is an important area of study in exploration for reinforcement learning, and some notable approaches include Upper-Confidence bounds for Reinforcement Learning (UCRL, Ortner and Auer 2007 and UCRL2, Jaksch et al. 2010), Model-based Interval Estimation (MBIE, Strehl and Littman 2005, 2008), and Random Network Distillation (RND, Burda et al. 2019b)—even the early curiosity system by Schmidhuber (1991a) is based on a decaying bonus (assuming it is applied in a deterministic environment). The purpose of the decay is to only temporarily encourage visits to any given state, enough to obtain sufficient samples.

3. **No need for an explicit reward objective.** Designing reward signals is a longstanding challenge of applying reinforcement learning to computational systems (Sutton and Barto, 2018, p. 469). While algorithms for learning to maximize return can be designed without concern for what the reward signal will be, complete reinforcement learning systems need reward signals. Intrinsic rewards can act in place of an extrinsic reward signal.

   The creation of open-ended learners without extrinsic reward functions is currently an area of increased interest. The Intelligent Adaptive Curiosity system devised by Oudeyer et al. (2007, pp. 265, 269, 283) was designed to evoke developmental, progressive learning from an agent with no extrinsic reward function. Pathak et al. (2017, p. 1), in their first demonstration of their Intrinsic Curiosity Module—a system using intrinsic rewards—explicitly investigated the behaviour of their system without any extrinsic reward (see also Burda et al., 2019a, p. 1). These exploratory studies have demonstrated that interesting or useful behaviours, like walking or completing video game levels (Burda et al., 2019a, p. 10), can arise from purely intrinsically motivated agents.

The specific value of designing systems that can navigate the world and learn (develop) over their lifetimes has been well-argued by Oudeyer et al. (2007, pp. 265–267). In particular, Oudeyer et al. (2007, p. 265) draw attention to the idea that, while a system might have an easier time learning a task we want it to complete if the system followed an incremental sequence of increasingly difficult tasks (e.g. learn to stand before learning to walk), it would be completely impractical to design each task in the sequence by hand. Therefore, the creation of such sequences must eventually be made autonomous. Further, when reward signals are designed by humans, they are notoriously likely to be somehow misspecified (this problem is sometimes referred to as reward hacking , specification gaming (Krakovna et al., 2020), faulty reward functions, or as a case of the alignment problem). Essentially, the objective you're really trying to achieve is not always achieved via the system's solution to the objective function provided.

4.3.3.2 Limitations of Intrinsic-Reward Approaches

Although intrinsic-reward approaches have important benefits, they are limited in their ability to achieve those benefits and lack some of the benefits we might expect from an analogue of biological curiosity.

Detachment: One limitation relates to the first benefit we described—that an intrinsic-reward approach offers a simple way to recognize and remain on an exploration 'frontier,' because it doesn't always work. In particular, since most intrinsic rewards are designed to decrease as the agent returns to the same state over and over again, it is possible for the agent to essentially use up the intrinsic reward without ever taking the actions required to continue into the nearby unexplored part of the world. This is an example of the problem Ecoffet et al. (2021, Supplementary Material, p. 20) called *detachment*, where an agent leaves and fails to return to parts of its environment that are likely on a frontier—likely to be close to new parts of the environment. This problem of detachment makes intrinsic reward approaches to seeking never-before-seen states quite brittle and unable to achieve this desired benefit in some situations.

By being *specific*, curiosity has different goals than the methods that suffer from detachment. Specific curiosity does not attempt to cover an entire frontier and doesn't regret losing track of a state that is likely to be near novel states. Specific curiosity may be best-suited for huge environments where there is so much possible novelty that the learner needs to be choosy about which new information they seek. Intrinsic reward methods are generally not so choosy about the novelty they seek.

Reactivity: If the goal of including a mechanism is to encourage the agent to experience something *new*, intrinsic reward offers an inelegant approach, as it can only drive that goal indirectly. A reward can only be provided for observing a state once it has been observed—at which point it is no longer new. As Shyam et al. (2019, p. 1) put it, intrinsic reward methods are *reactive* and cannot direct a learner towards novel observations. The reward becomes associated with something already observed, not with novelty itself. To best achieve this goal, the agent should be directed towards the new part of the world, rather than pushed to dither near it. Of course, this is easier said than done, and so methods like Go-Explore (Ecoffet et al., 2021) that return to frontier states then focus on actions that may lead to novel states, instead of focusing on staying in such states, offer a useful interim measure.

A core aspect of specific curiosity is planning to go retrieve a particular piece of information to build the right knowledge when the learner is ready for it. Curiosity could seem reactive in that a learner does indeed seem to react to curiosity-inducing situations—quite suddenly, the learner is directed towards attempting to satisfy its curiosity, in reaction to what it just observed! But the type of reaction associated with curiosity is forward looking. Rather than "Oh, that was novel, I'd better experience it again!" (backward-looking) it is more like, "Oh, I have a question, I'd better go make an observation that will answer it—and by it's nature of its contents being unknown to me, that observation will be novel." Furthermore, the property of voluntary exposure means that even curiosity-inducing situations are not always accidentally stumbled upon, but often actively sought.

Lack of motivation in non-stationary environments Another limitation relates to the second benefit we described: offering a way to check for consistency. Many types of intrinsic reward—decay-based intrinsic rewards in particular—only offer a way to check for consistency if we assume the environment is stationary. By stationary, we mean that patterns and distributions in the environment never change, so once you have collected enough samples to be confident in a pattern or distribution, you never have to return to collect more. If the environment is non-stationary, the pattern could change completely while you're not looking, so you must regularly return to check if you want to be sure of your estimates.

Decay-based rewards, in particular, are generally not designed to encourage an agent to return to parts of the environment that it has already visited a sufficient number of times. However, there are intrinsic rewards designed to account for this concern: one of the earliest intrinsic rewards, used as part of Sutton's (1990b) Dyna-Q+ agent, was an additive intrinsic reward ('exploration bonus') that, for a given state, grew with the amount of time since the agent's last visit. Note that this exploration bonus was not a reward bonus, however; rather than being added to the *reward* observed by the learner, the exploration bonus was added directly to the value of the state. The longer it has been since the agent's last visit to that state, the more the value for the state would grow, motivating the agent to return.

Of course, Dyna-Q+ relies on a model of the environment. In reinforcement learning, a *model* of the environment, sometimes *transition model*, is traditionally refers to a function that takes a state and an action to take from that state and returns a next state and reward, mimicking the environment (Sutton, 1990b, pp. 217–218). Models of the environment are notoriously challenging to formulate for real-world applications where environments are so large and complex that building full models would extend beyond real memory and computational limitations. However, the benefits of Dyna-Q+ point to a need to address these challenges to achieve effective curiosity or exploration: without being able to "think about" or simulate experiences far from your current position in the world, it will likely be impossible to develop specific intentions to observe parts of the world containing the information that an agent needs or wants most. For this reason, our view of specific curiosity is that it does require a model or related method of simulation.

Returning our attention to lack of motivation in non-stationary environments, we note that specific curiosity is unlike the exploration methods that hope to maintain a complete collection of consistent value estimates for every state in the environment. Specific curiosity is similarly susceptible to never returning to a prior part of the environment again in a learner's lifetime. However, where specific curiosity differs is that it *can* drive a learner back to revisit a specific part of the environment, when driven by an inostensible concept.

A mechanism of inostensible concepts should be flexible. Ideally, it should equally be able to ask "Does the bookstore floor still make the same peculiar noise it made when I visited last week?" (a question about a previous experience) as "If I pried up the offending floorboard, would I find something underneath?" (a question about part of the environment never previously experienced).

Reliance on repetition of state: Our final limitation of interest connects to the second key benefit—that intrinsic-reward approaches are useful for checking if an action results in a consistent reaction. This benefit may not align with our goals for designing computational curiosity. Curiosity may be misaligned with an underlying assumption about *state* that is typical in computational reinforcement learning. We mentioned state in Section 4.3.1, describing it as a term used in the reinforcement learning framework to describe the situation that the agent finds themself in.

State repetition is central to reinforcement learning, as reinforcement learning is designed to evaluate how well an action went previously so the learner can adjust their behaviour next time they are in the same situation. If I didn't much like bumping my nose on the door last time, I might choose a different action when I'm next faced with a closed door.

State is usually thought of as essentially separate from the learner, and more importantly, as repeatable, meaning a learner can experience the same state multiple times. Of course, in large complex worlds, the exact same situation isn't likely to repeat multiple times, but with some generalization, this assumption is very helpful. Important features of the state can repeat multiple times and be useful for predicting reward. For example, imagine you're a rat in a box with a lever. Let's say that when a light in the box is turned on, pulling the lever results in the appearance of chocolate for you to eat, but when the light is off, nothing happens when the lever is pulled. In this case, thinking of *light on* and *light off* as repeatable features of state can prove very useful in optimizing your chocolate intake.

However, this assumption that state repeats should be complicated in the case of curiosity. Why? With curiosity, a learner's goal is to change their situation by making changes to their own *knowledge state*. By knowledge state, we mean the state of what the learner knows—what the agent has learned from its observations of the world. While reading your novel, you may find yourself curious, wanting to change your knowledge state from not knowing who the killer is to include knowing who the killer is. While wandering through the bookstore, you found yourself wanting to get into a knowledge state where you know if it was your own action that generated peculiar noises from the floorboard. In comparison to traditional reinforcement learning state, which we can call *environment state*, knowledge state is similar in that the learner can take actions to change it, but it is different in that it isn't helpful to think about returning to previous knowledge states.

A learner's knowledge state is continually changing and does not have the same repeatability as environment state: it is much more useful to think of the agent's knowledge growing and adapting with each new observation of the world. Sure, an agent might forget things, but that doesn't mean it ever returns to a prior knowledge state.

For example, when checking if an action results in a consistent reaction, the knowledge state of the agent actually changes after each trial. The inostensible concept of interest is not the result of a single trial, but actually some statistic about the distribution of possible results. For the agent trying to learn the value of a state, the inostensible concept might be the mean value, and for the scientist, the inostensible concept is more likely to be some underlying pattern or truth about the world. Appropriate directed behaviour, in this case, is to experience the same environmental state features multiple times, but each visit provides new information and leads to achieving a different knowledge state.

Of course, specific curiosity still needs to use repeatable features of environment state. In fact, we believe that an agent learning what features of the environment tend to repeatably lead to curiosity-inducing situations might be critical to the property of voluntary exposure, (e.g. sections of bookstores labelled 'Mysteries' could be a good feature). And without learning about repeating features of environment state, how could we plan directed action to satisfy our curiosity? (c.f. Berlyne, 1954, p. 183).

***

4.3.3.3 Specific Curiosity in Relation to the Limitations of Intrinsic Reward Approaches

We believe that specific curiosity can address some of the limitations of intrinsic reward approaches. However, we also recognize that specific curiosity appears to function for a different purpose than intrinsic reward methods and compare the functions and goals of each type of method in this discussion.

Detachment: By being *specific*, curiosity has different goals than the methods that suffer from detachment. Specific curiosity does not attempt to cover an entire frontier and doesn't regret losing track of a state that is likely to be near novel states. Specific curiosity may be best-suited for huge environments where there is so much possible novelty that the learner needs to be choosy about which new information they seek. Intrinsic reward methods are generally not so choosy about the novelty they seek.

Reactivity: Specific curiosity is less defined by reactivity and is a forward-thinking method. The core piece of specific curiosity is the planning to go retrieve a particular piece of information to create the right knowledge at the right time.

A curiosity-inducing situation seems to stem from an update to the learner's knowledge state that results in the agent recognizing an inostensible concept, or specific piece of knowledge that they don't have. In large, complex worlds where a learner can't expect to do everything it is possible to do, specific curiosity helps the agent to go get the right observations for the agent's knowledge state at the right time.

In summary, while it may sometimes be reasonable to think of learners returning to the same environment state and action, this is not a return to the same knowledge state.

4.3.3.4 Approaching the Five Properties in the Computational Literature

Computational reinforcement learning researchers have shown strong interest in aspects of the properties of directedness towards inostensible referents, cessation when satisfied, voluntary exposure, and transience. Their exploration has not always been done in the name of curiosity, however. For example, the idea of directedness (though not necessarily towards inostensible referents) parallels work done on options (as early as Sutton et al., 1999) and planning. The study of *options*, a mathematical abstraction of short-term policies, has resulted in a growing body of research. Part of the appeal of options is their potential to get an agent from point A to point B (which could be thought of as a goal) without emphasis on the path to get there. Purposeful exploration using options and related ideas has been actively pursued by researchers such as Machado (2019). The options framework, in particular, further reflects cessation when satisfied and aspects of transience via termination conditions for each option. Some termination conditions are naturally defined by goal states, so the directed behaviour ceases upon reaching a goal state, much like cessation when satisfied; other termination conditions can be based on when the option hasn't succeeded in reaching its goal state in a reasonable amount of time, one of the aspects of transience (Stolle and Precup, 2002, p. 212). However, as Colas et al. (2022) have pointed out, most work with options to date has largely only considered goals within the distribution of goals previously encountered (p. 1177). One notable exception is the Intrinsic Motivations And Goal INvention for Exploration (IMAGINE) architecture, in the design of which Colas et al. (2020) leveraged the compositionality of language to generate goals—which could be seen as a step towards leveraging the compositionality of concepts to generate inostensible concepts.

Prior work has further aimed to address the lack of directedness that is a characteristic of intrinsic-reward methods. For example, the Model-Based Active eXploration algorithm presented by Shyam et al. (2019) uses planning to allow the agent “to observe novel events” (p. 1). They care about unknowns and about creating paths to them. The Go-Explore family of algorithms also centres on the idea of taking a direct sequence of actions to move to a specific state for the purpose of exploring from it, as per Ecoffet et al. (2021). In these examples and others, it is clear that recent work has begun to seek ways to avoid the reactive approach to designing machine curiosity.

Murayama et al. (2019) developed a model rooted in reinforcement learning to describe the reward process involved in knowledge acquisition, designed to help explain curiosity and interest. Yasui (2020) “also found that methods which add a bonus to their value function tended to explore much more effectively than methods which add a bonus to their rewards” (p. ii). This is part of a growing body of evidence in the literature that additive reward bonuses do not in many cases reflect or lead to the same results as human curiosity. As stated by Gruber and Ranganath (2019, p. 1016), “the effects of reward and curiosity are not additive, and reward has been shown to undermine curiosity and its effect on memory” (in reference to Murayama et al., 2010; Murayama and Kuhbandner, 2011). Finally, active perception is a field of computing science concerned with building systems that take action to change what the system perceives towards specific goals. The needs that arise when considering how to design algorithms for specific curiosity overlap substantially with the concerns of active perception.

In summary, it is encouraging to see a wide body of literature begin to move toward effecting what could be well considered properties of specific curiosity. In the section that follows, we expand on some of the benefits that each of the properties of specific curiosity can bring to curious reinforcement learning agents by way of a concrete implementation and empirical study that highlight how multiple properties work together as a unified whole to generate curious behaviour in a learning machine.

***

This section discussed very generally the kind of machine learning context we think specific curiosity makes sense in and what we might expect these properties to look like in such a context. In particular, the reinforcement learning framework includes useful infrastructure for algorithmic decision making, planning, and preference. The potential of this framework for computational curiosity is well-regarded, but has primarily been explored via intrinsic-reward approaches. We discussed some important benefits and limitations of intrinsic-reward approaches and alluded to some of the differences between the behaviour generated by intrinsic rewards and that which we should aim for when designing specific curiosity. In the next chapter, we showcase our prototype computational agent with three of our five properties of specific curiosity, aiming to build further intuition about the possibilities of such a system.

Chapter 5

Fundamental Study: A Prototype of Specific Machine Curiosity

We now present a case study that illustrates one possible way that three of the key properties of specific curiosity might be implemented to shape the behaviour of a reinforcement learning agent. Our intent is for this example to help the reader more deeply understand the properties of specific curiosity identified above, and how the computational principles they represent might be translated to algorithms and implementation. To support this understanding, the case study is designed to model our running bookstore example, so the agent, like you, has the opportunity to discover its own analogue of your corner bookstore.

We specifically hope to show that, even in a simple and focused setting, using the properties of specific curiosity we've highlighted as guidelines allows us to see machine behaviour emerge that approximates specific curiosity from the animal learning domain. Further, we aim to depict how these properties are modular and amenable to extension as future, more insoluble aspects of specific curiosity become computationally clear and tractable. This example is not, however, to be interpreted as a recommendation for a final or definitive computational implementation of specific curiosity. We diverge from the more common practice of fully tackling a problem without domain knowledge, instead implementing hand-designed rules of thumb or expert knowledge as solutions for some of the more challenging, unsolved aspects of computational specific curiosity, such as the process for recognizing inostensible concepts. The intended purpose of this section is for the reader to gain insight and motivation to further investigate the way the properties of specific curiosity might be integrated into different machine learning frameworks and problem settings.

To this end, we offer three sets of experiments. Sections 5.1 and 5.2 describe the base agent and base domain, respectively, that will be used throughout—agent interactions with the base domain are directly explored in the first set of experiments (Sections 5.3.1 and 5.3.2). In our second set of experiments, we investigate agent behaviour when the domain is perturbed in terms of domain geometry and span (Sections 5.4.1 and 5.4.2). In our third and final set of experiments (Sections 5.5.1 and 5.5.2), we examine the ablation of individual properties of specific curiosity within the agent and the impact this has on agent behaviour.

5.1 Agent Implementation

In this section, we provide the specification for an agent that, if truly exhibiting the behaviour expected from the biological literature on specific curiosity, would be expected to:

1. take a largely direct route to a curiosity-satisfying situation, which we term a *target* (directedness),
2. not repeatedly return to situations that had satisfied curiosity (cessation when satisfied), and
3. develop a preference for (increased estimated value for) parts of the world that repeatedly offer curiosity-inducing observations (voluntary exposure).

The full algorithm followed by our curious agent is described in Algorithm 2. Sections 5.1.3–5.1.5 provide detail on how each property is included in the algorithm. The agent parameters used for our experiments are shown in Table 5.1.

Algorithm 2 A specific example of prototyping specific curiosity

```
Initialize $\alpha$, $\epsilon$, $\gamma$, $\gamma_{curious}$, $V$, $x$
Initialize $V_{curious}$, $R_{curious}$ to zeros
while alive do
    if agent observation $x$ induces curiosity then
        generate a new curiosity target
        generate $R_{curious} = \begin{cases} 0, & \text{if transitioning to target} \\ \mathbf{-1}, & \text{otherwise} \end{cases}$    ▷ Aversive Quality
        $V_{curious} \leftarrow ValueIteration(R_{curious}, \gamma_{curious})$
    if there is currently a curiosity target (i.e. the agent is curious) then
        $x'$, $R \leftarrow$ move greedily w.r.t. $\boldsymbol{V_{curious}(x)}$    ▷ Directed Behaviour
    else
        $x'$, $R \leftarrow$ move $\epsilon$-greedily w.r.t. $V(x)$    ▷ Ties broken uniform randomly
    $\delta \leftarrow R + \gamma \cdot V(x') - \boldsymbol{[V(x) + V_{curious}(x)]}$    ▷ Voluntary Exposure
    $V(x) \leftarrow V(x) + \alpha\delta$
    if agent observation $x'$ is the target then
        destroy the current target
        reinitialize $\boldsymbol{V_{curious}}$ to zeros    ▷ Cessation when Satisfied
    $x \leftarrow x'$
```

As a note on the scoping of our empirical work: In the design of the agent used in this case study, we aim to demonstrate interactions between the first three of the five key properties of specific curiosity we contributed in the sections above (directedness, cessation when satisfied, and voluntary exposure). This scope is deliberate: we place our initial focus on foundational properties of specific curiosity that for clarity of investigation can be well studied and perturbed in isolation from the experimental variability of long-term information search and the shifting focus (transience) related to life-long learning. We address these remaining two properties and their conceptual connection to our observed results in the discussion sections below, and explicitly in Section 5.6.

We further contain the scope of these initial experiments by limiting the comparison of secondary computational operations that are involved in specific curiosity but that might have a variety of possible algorithms and implementations—in such cases we chose the clearest, simplest implementation of the many possible alternatives. Specifically, in Section 4.2.2.2, we noted the importance of separating curiosity-inducing observations from curiosity-satisfying situations. However, recognizing appropriate curiosity-inducing observations and estimating where in the world the appropriate satisfying observations can be found are complex issues. In this initial case study we chose to isolate the key properties from these complexities so as to better see the impact of the properties themselves on agent behaviour. We achieved this isolation by assuming the existence of an oracle-like mechanism that indicates that curiosity has been induced and indicates the location of an observation that would satisfy it. In what follows, we often refer to this particular location in the domain as the *target* of curiosity, in reference to the idea that, while there may be many possible ways of making the inostensible concept of focus ostensible, the agent selects one potential curiosity-satisfying situation and then aims its behaviour towards experiencing that situation. We refer in what follows to the mechanism for recognizing curiosity-inducing situations and suggesting appropriate targets as a *curiosity-recognizer module.*

5.1.1 Base Algorithm

Since we are conceptualizing specific curiosity as resulting in a binary state of curiosity—at a given moment, the agent is either curious or not—we can start with a base algorithm in our experiments that determines the baseline agent behaviour when the agent is not curious. For simplicity, since the intent of this work is to explore behavioural change and not task optimality, we chose TD(0) (Sutton and Barto, 2018, pp. 120-121) as our base algorithm,[1] with an ϵ-greedy policy[2] with

[1] We herein do not rely on eligibility traces to prevent confounding their impact during analysis with the way a system might present its developed preference for curiosity-inducing situations; we expect the practical impact of accumulating or replacing eligibility traces to be one of speeding up the acquisition of preference for curiosity inducing situations, but this is a detailed comparison intended for future work.

[2] Epsilon-greedy (ϵ-greedy) behaviour refers to choosing the action that has the highest estimated value (being *greedy)* nearly all of the time, but a small percentage of the time, choosing randomly from the available actions. The 'epsilon,' ϵ, in ϵ-greedy is a parameter that sets how likely it is that a given action will be random rather than greedy. For more information on epsilon-greedy behaviour, see Sutton and Barto (2018, pp. 27-28).

respect to its estimated value function V, with ties broken by equiprobable choice. This behaviour is defined in Line 11 of Algorithm 2. Further, while the agent is not in a state of curiosity, its learning follows the standard TD(0) learning update:

$$V(x) \leftarrow V(x) + \alpha\delta \text{ where } \delta = R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \tag{5.1}$$

Note that, in Line 12 of Algorithm 2, when the agent isn't curious, our $V_{curious}$ is zero everywhere, so the learning update simplifies to the standard TD(0) update.

5.1.2 Recognizing Curiosity-Inducing Observations

To enter a state of curiosity, the algorithm relies on a curiosity-recognizer module, which, upon a curiosity-inducing observation, generates an associated target location (Line 4). In our bookstore analogue, looking around the bookstore offers a curiosity-inducing observation, like an intriguing back-of-book blurb, and upon this observation, the reader/agent automatically has a target observation or set of target observations in mind. The target might be observing the first page of the book, and based on the target, the agent can guess how best to act to achieve the target (open the book) and proceed.

As we mentioned earlier in Section 5.1, recognizing when an observation should induce curiosity and estimating where an appropriate satisfier might be found are complex issues with solutions beyond the scope of this paper. For this case study, we instantiated a specific location in the domain to induce curiosity and a set of locations of possible satisfiers. Each time curiosity is induced by visiting the curiosity-inducing location, one location for a satisfier is chosen randomly from the set; we refer to this location as the target. This simple target generator acts as the curiosity-recognizer module in our experiments. The exact locations used for our experiments will be described with the domains in Sections 5.2 and 5.4.1. We use this simplified curiosity-recognizer module to recognize when curiosity is induced.

5.1.3 Directedness Towards Inostensible Referents

Once curiosity is induced, the agent changes its behaviour. To achieve the property of directedness towards inostensible referents, the agent is no longer ϵ-greedy with respect to V and is instead fully greedy with respect to $V_{curious}$, a temporary value function. As mentioned earlier in Section 5.1, the key property of $V_{curious}$ is that it is a gradient leading the agent towards the target provided by the curiosity recognizer: if one location is fewer actions away from curiosity's satisfier than another, the former location has higher value. An agent acting greedily with respect to the temporary value function will travel directly to curiosity's satisfier.

In our implementation, the function $V_{curious}$ is generated via value iteration (Sutton and Barto, 2018) in Line 7. Value iteration generates appropriate gradations in the value function, even taking into account any known obstacles or required detours between the agent's current location—or any given location—and the location of the target. See Figure 5.4(a, $V_{curious}$) for a visualization of a gradient generated by value iteration. Value iteration is performed using the agent's transition model of the space, but uses a special reward model, $R_{curious} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, which maps a transition from any location other than the target to -1, but maps a transition from the target to 0. Equivalently:

$$R_{curious}(s, a, s') = \begin{cases} 0 & \text{if } s \text{ is the target} \\ -1 & \text{otherwise} \end{cases} \tag{5.2}$$

This choice was inspired by the characteristic aversive quality of curiosity mentioned in Section 4.2.5.

Note that in this simplified agent, we provided the agent a perfect transition model of the world, so that its value iteration produces an exactly direct gradient to the target. The agent could instead learn this model from experience. Future work will need to consider the implications of *not* giving the agent a perfect model, as using a perfect model is a simplification rarely possible in real-world settings.

5.1.4 Cessation When Satisfied

The property of cessation when satisfied refers to the agent's behaviour no longer being affected by curiosity once the agent has observed the target of its curiosity. Once the agent has visited the target (Line 14), the agent is no longer curious and returns to its base behaviour. In the algorithm, this return to base behaviour is achieved by removing the target (Line 15) and zeroing out $V_{curious}$ (Line 16). The agent will only become curious again when it has another curiosity-inducing observation as recognized by the curiosity-recognizer module.

5.1.5 Voluntary Exposure

After several cycles of curiosity being induced, followed, and satisfied, if a particular part of the world repeatedly induces curiosity, the agent can learn a preference for returning to that part of the world. This learning process exemplifies voluntary exposure. In our running example, if you visited your corner bookstore by largely random choice during a few strolls around your neighbourhood and each time you found your curiosity sparked by excellent reads, you might find yourself heading to the bookstore directly to shortcut the process.

While we designed directedness and cessation when satisfied as simple behaviours, voluntary exposure was more interesting because we wanted our design to let the agent learn where in the world it might repeatedly become curious, and therefore voluntarily expose itself to those parts of the world—in reference to our running example, returning to the bookstore. We made a simple change to the TD update that would let the temporary value function, $V_{curious}$, influence the enduring value function, V. This change can be found in Line 12 of Algorithm 2:

$$\delta \leftarrow R + \gamma \cdot V(x') - [V(x) + V_{curious}(x)]$$

This change means that when the agent is curious, the temporary value function, $V_{curious}$, affects the learning update to its estimated value function, V. Since $V_{curious}$

α	0.01
ϵ	0.2
γ	0.9
$\gamma_{curious}$	0.9
initial $V : \mathcal{S} \rightarrow \mathbb{R}$	$V(s) = 0 \; \forall s \in \mathcal{S}$

Table 5.1: Parameters used at initialization in our experiments.

is negative everywhere, the enduring value for any locations the agent visits while curious will increase.

This design choice came from intuition more than a strong theoretical underpinning. Our intuition was that the experience of curiosity should affect internal value estimates more enduringly, but the result should not be an enduring push towards curiosity's satisfiers, as we might see if visiting a satisfier were intrinsically rewarding. Instead, we hoped to see an enduring effect of increased preference for curiosity-inducing situations, or, algorithmically, increased value. Our initial experiments were designed to uncover whether this algorithmic choice would offer behaviour and value function estimates characterized by the property of voluntary exposure, which we would observe as a learned preference (increased value) for locations where the agent repeatedly makes curiosity-inducing observations.

***

In summary, the agent implemented in this section was designed to incorporate three of the five key properties highlighted in this paper: directedness, cessation when satisfied, and voluntary exposure. As a reminder, this design represents only an initial example of how these properties might be simply achieved, and meant to inspire other approaches to agents exhibiting specific curiosity. In the remainder of this experimental section, we describe experiments designed to help us better understand the effects of our algorithmic choices for the agent.

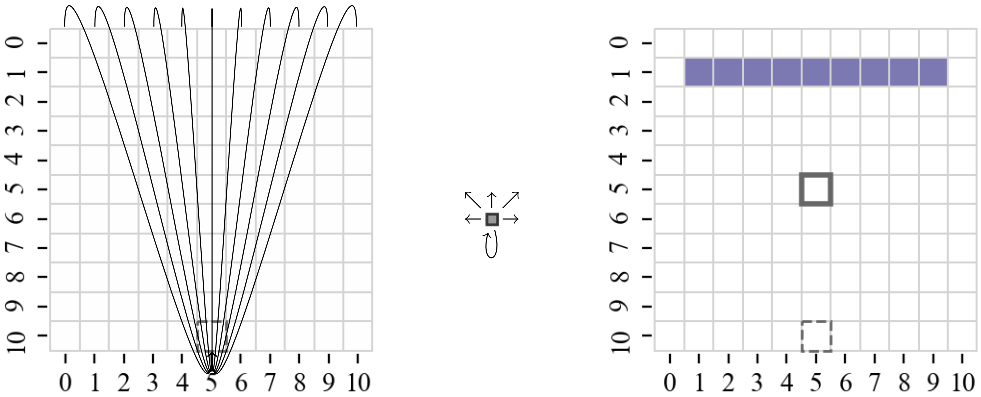
5.2 Primary Domain

With the goal that our experiments should use a simple and focused setting to highlight machine behaviour approximating biological specific curiosity, we designed a primary domain mirroring our running example of the corner bookstore. The domain is a simple gridworld, meaning that the agent occupies a single square in a grid. Our primary domain is an 11 by 11 grid, depicted in Figure 5.1. In this paper, our references to locations on the grid are 0-indexed using (*row*, *column*) notation.

Actions can move the agent one space per step either up, on either upward diagonal, left, or right—but never down or on a downward diagonal. The agent also has a stay-here action that allows it to stay on the same location. These actions are shown visually in Figure 5.1(b). Directly left or right actions that would take an agent beyond the left or right boundary of the grid instead return the agent to the square where it attempted the action. Similarly, diagonal actions that would take an agent beyond the left or right boundary of the grid instead simply move the agent up. Any action that would take the agent beyond the upper boundary of the grid teleports the agent to the midpoint of the lowest row of the grid, $(10, 5)$, which we will refer to as the *junction location*.[3]

A location in the centre of the grid at position $(5, 5)$ is considered a permanent *curiosity-inducing location*, analogous to the bookstore in our example. This choice makes the curiosity-recognizer module mentioned in Section 5.1 very simple. When the agent enters the curiosity-inducing location, the module generates a target.

[3] The choices to have no downward actions and to teleport off the top of the grid to the midpoint of the bottom of the grid may seem unexpected. This choice was made to allow greater clarity in the visual presentation of the outcomes of the case study. By removing backtracking, the visit counts for each state more clearly show where the agent chooses to move. There are other choices, such as standard cardinal direction actions. The choice to teleport to the *junction location* rather than treating the grid like a simple cylinder simplifies the learning problem by making it more likely that the agent will return to a state it has already learned about, speeding up the learning process. We evaluated a range of alternatives without some of these constraints on the domain and movement, but they have been omitted from this manuscript for what brevity we can hope to preserve; key observations in settings with backward motion, cardinal motion, cylindrical wrapping, and others are well captured by the presented results.



(a) Domain Mechanics **(b) Agent Actions** **(c) Agent Target Generation Mechanics**

Figure 5.1: This image visualizes the mechanics of the primary domain described in Section 5.2. The junction location is shown at the bottom with a grey dashed outline in (10, 5). The arrows in (a) from the top row back to the start location represent teleportation back to the junction location when the agent takes an upward action off the top of the grid. The grey rectangle shown in (b) will represent the agent in later figures, and (b) also visually shows the six actions available to the agent from any location. For clarity, the target generation mechanics needed for the curiosity-recognizing module (not considered inherent to the domain) are shown separately in (c). The curiosity-generating location has a thick solid grey outline. The possible locations for curiosity targets to be generated, across the second row from the top, are highlighted in purple.

Much like your corner bookstore, the curiosity-inducing location reliably induces curiosity in the agent when visited. Every target is generated in row 1 of the grid (the second row from the top) with equal probability of being placed in any of the 9 grid columns besides those directly neighbouring the left or right boundary, i.e., a target chosen from the locations in the row between and including $(1, 1)$ and $(1, 9)$. Different stories have different endings and different narratives to take the reader to them, so the targets generated at your corner bookstore vary.

While a reward function is usually included as part of an experimental domain for reinforcement learning, we did not include a reward function as part of the domain for the experiments described in this paper.[4] For a standard reinforcement learning agent that expects a reward for its learning algorithm, we could equivalently define R_t to be 0 at every time t. We leave exploring how best to balance $V_{curious}$ with the value generated by a nonzero reward function to future work.

In the experiments showcased in this paper, we initialized each trial with the agent located at the curiosity-inducing location, as the random behaviour to find the curiosity-inducing location is not especially relevant to the mechanics central to this paper. We did run experiments with the agent starting at other locations: the results are not meaningfully affected, but the learning time is extended.

The primary domain described in this section acted as the environment that the agent interacts with in our first and third sets of experiments and as a starting point for domain modifications in the second set of experiments. By using a clear analogue of the bookstore throughout, our intention was to make behaviours characterizing specific curiosity obvious.

5.3 First Set of Experiments: Primary Domain and Base Agent

5.3.1 Experimental Setup: Visit Count and Value Study in the Primary Domain with the Base Agent

As we noted early in Section 4.3.3.4, we wanted to design experiments to seek out machine behaviour approximating biological specific curiosity. To achieve this, we observed visit counts (where an agent goes) and the agent's estimated value function (what locations an agent learns to prefer).

[4] As a reminder from Section 4.3.1, typically the goal pursued by reinforcement learning agents is to maximize their accumulation of extrinsic reward—the reward provided by the environment, usually defined as part of the domain. This goal is not directly relevant to the core of this paper, which is more focused on isolated mechanics of curiosity.

To measure what the agent does or how the agent acts, we use *visit counts*. We use the term visit counts to refer to an array of integers, one integer for each location on the grid equal to the number of times the agent has visited that location. At any given time step, the values in the visit count array will be identical to the values in the preceding time step, except at the location that the agent visits, which will be larger by 1. At the end of a trial, the visit counts help us see where the agent spent more time and where it spent less time. Graphical examples of visit counts can be found in the right column of Figure 5.2.

To gain insight into what the agent learns and how its persistent value function changes over time, we can represent the persistent value function as an array with the value equal to the estimated value of that location. Graphical examples of the persistent value function can be found in the left column of Figure 5.2. The agent's curiosity value function can be represented similarly, and, in the context of Algorithm 2, a record of the curiosity value function at each time step can provide insight as to *why* the agent acted in a particular way or learned a particular change in the persistent value function. Graphical examples of the curiosity value function can be found in the second row of Figure 5.4.

As our initial experiment, we recorded the estimated value function V, the curiosity value function $V_{curious}$, and the visit counts of the agent described by Algorithm 2 in the primary domain in 30 trials of 5000 time steps each (with each time step referring to an iteration over the loop in Lines 3-17 of Algorithm 2). For each trial, we recorded the value functions at each time step. Recording these values allowed us to create frame-by-frame animations for each trial showing the agent's movement through the grid over time along with the changing value functions. An example of agent motion and value learning in video format is provided as supplementary material: `https://youtu.be/TDUpB7OefFc`.

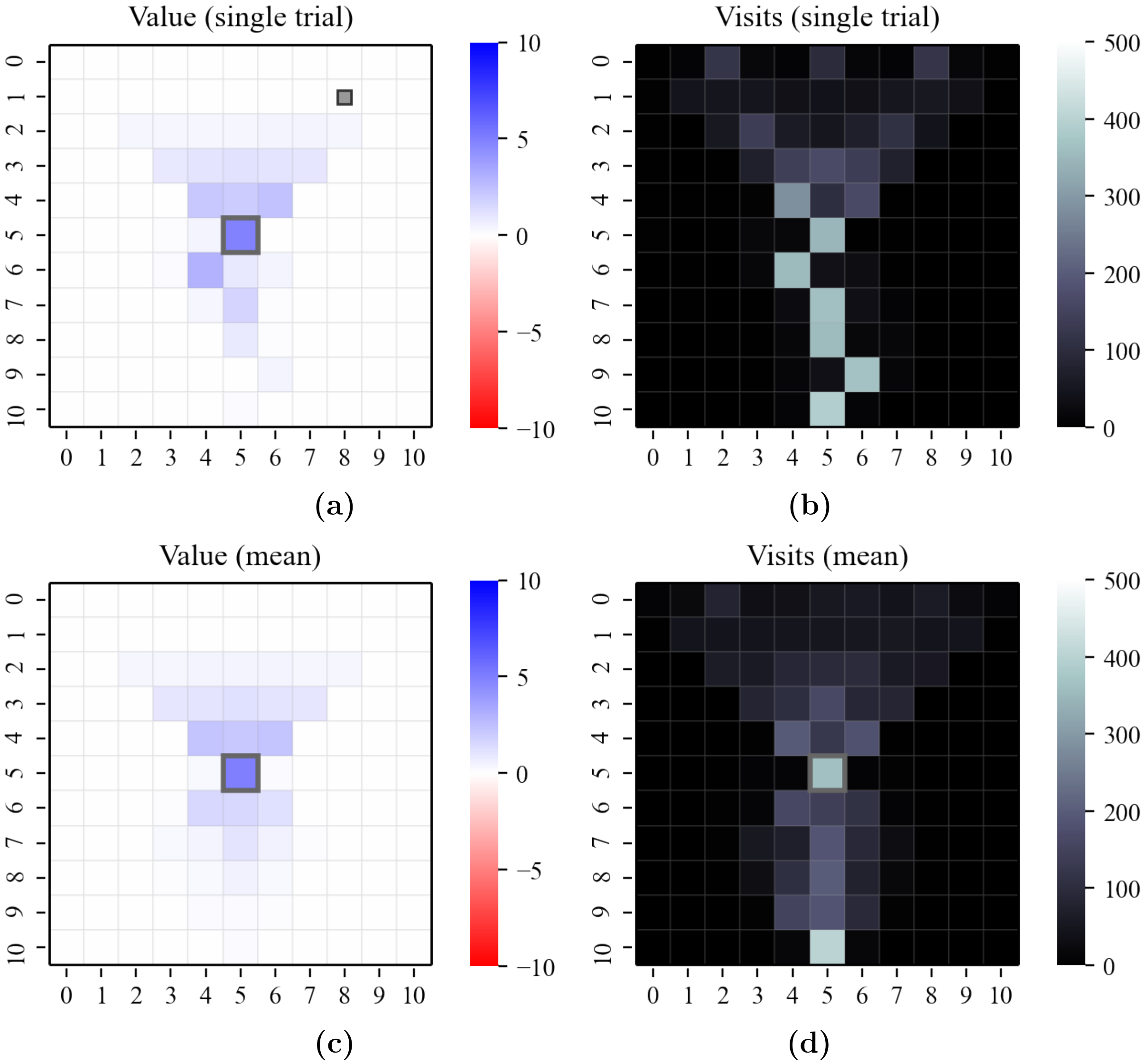
To account for stochasticity in the agent's behaviour, we also aggregated the final estimated value functions and visit counts (after 5000 time steps) over all 30 trials. Similarly, we aggregated the estimated value of the curiosity-inducing location and potential target locations at each time step over all 30 trials. Observing the changes in the estimated value function, in particular, allowed us to test our hypothesis of voluntary exposure: that the curiosity-inducing location would strongly accumulate value, while the locations of the targets would accumulate relatively little value.

Overall, this initial experiment in the primary domain allowed us to look for patterns in the agent's behaviour and learning and then compare those patterns to the expectations we developed through conceptual analysis of the properties of curiosity in Section 4.2.

5.3.2 Results and Discussion: Visit Count and Value Study in the Primary Domain with the Base Agent

One question that motivated these experiments was: Does the agent learn to value the curiosity-inducing location, emulating the property of *voluntary exposure*? In particular, an agent demonstrating specific curiosity would learn a preference to return to the curiosity-inducing situation (think the bookstore) and *not* learn a preference to return to the curiosity-satisfying targets (think the specific pages of each book). Figure 5.2(c) shows that the final value function, aggregated over all trials, had this property, with the curiosity-inducing location having the highest persistent value of all locations in the grid. We can also see a gradient leading from the bottom row of the grid up to the curiosity-inducing location, showing that, after 5000 steps, the agent had a persistent preference to move to the curiosity-inducing location. Figure 5.2(d) shows that this preference was reflected in the agent's behaviour: visits were concentrated between the junction location (where the agent starts each upward traversal of the grid) and at the curiosity-inducing location.

While it is promising to see this indication of voluntary exposure at the end of learning, we also would hope to see the difference in preference between the curiosity-inducing location and potentially curiosity-satisfying targets learned smoothly



(a) **(b)**

(c) **(d)**

Figure 5.2: This figure shows the persistent value function and visit counts in the primary domain for a simple reinforcement learning agent demonstrating properties of specific curiosity. From this figure, we can see that the agent learned to value the curiosity-inducing location and therefore follow a direct path to that location, but it does not learn to value the targets of its curiosity. Shown here are (a,c) the persistent value function V and (b,d) the total visits the agent made to each state in the 11 x 11 grid domain. Totals plotted for trials of 5000 steps, with (a) and (b) showing value and visit counts for one representative trial, while (c) and (d) are averaged over 30 independent trials.

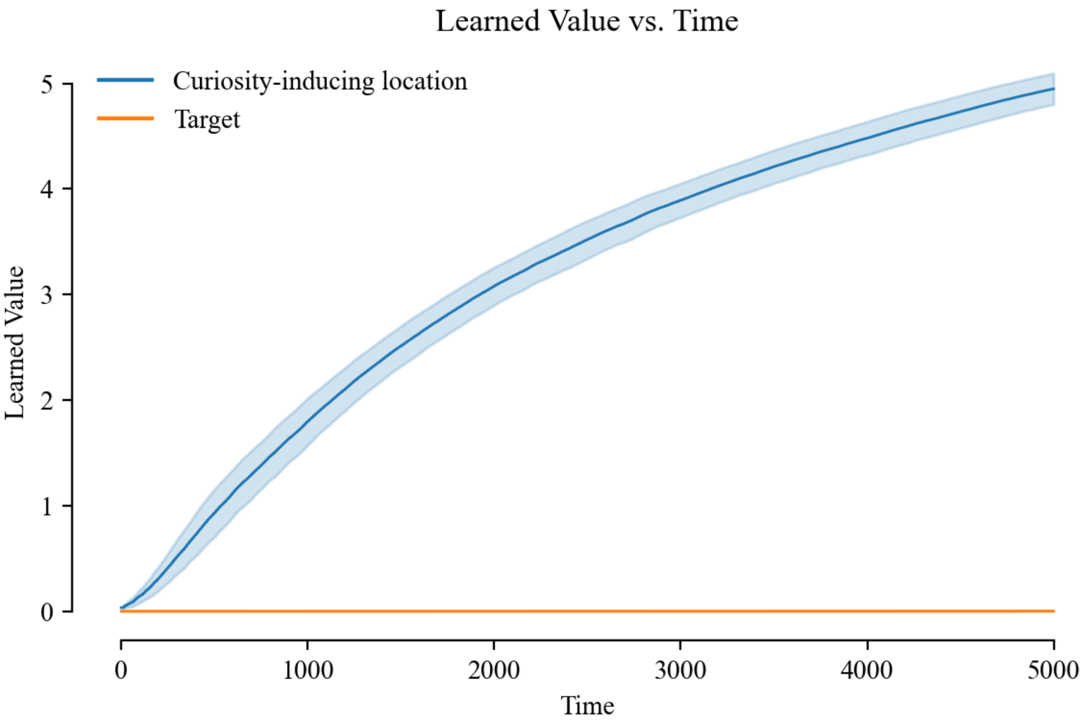


Figure 5.3: This figure shows the mean (line) and standard deviation (shaded area) of the persistent value of the curiosity-inducing location (in blue) and of all the possible target locations (in orange) over time, considering 30 trials. The estimated value of the curiosity-inducing location grows sublinearly while the learned values of the targets hover around 0 throughout with little variation or growth.

over time. Indeed, this desired pattern can be seen in the learning curves in Figure 5.3.

To understand how the agent learned to travel directly to the curiosity-inducing location, it can be helpful to follow the agent through a cycle of curiosity being induced, followed, and satisfied. The first such cycle in one trial is followed in Figure 5.4. The agent started at the curiosity-inducing location at $t = 0$, where curiosity is triggered. The leftmost column of Figure 5.4 shows the temporary reward function ($R_{curious}$), the temporary value function ($V_{curious}$), the persistent value function (V), and the visit counts at time $t = 0$. For the agent, the induction of curiosity meant generating a curiosity-satisfying target (in the figure, the target has a dashed line border and is located near the top right of the grid). An associated temporary reward function, $R_{curious}$, was generated, shown in panel (a), which was used to compute an appropriate temporary value function, $V_{curious}$, shown in panel (b).

Acting according to the property of *directedness*, the agent moved directly to the target and reached that target at $t = 3$, as shown in panel (h). At each step, the agent's persistent value function was updated according to Line 12, so we see the gradient we saw in $V_{curious}$, panel (b), reflected in the learned value in panel (g). The further from the target, which is where $V_{curious}$ is more negative, the more positive value was accumulated into the persistent value function.

When the agent observed the target at time $t = 3$, its curiosity was satisfied, and in accordance with the property of *ceases when satisfied*, the target-driven behaviour ended. This means that $R_{curious}$ and $V_{curious}$ were zeroed out for all locations, as shown in panels (e) and (f), respectively. In this initial cycle, the agent's behaviour was wandering and largely random (as can be observed via its visits in panels (l) and (p)) until the agent reached a location adjacent to a location that has accumulated some persistent value—in this case, the agent reaches a location adjacent to the curiosity-inducing location, where a greedy action would be to move to the curiosity-inducing location. At time $t = 16$, the agent visited the curiosity-inducing location where the cycle restarted with a new target.

We have seen the agent exhibit the properties of directedness, cessation when satisfied, and voluntary exposure, which was the desired result. However, this experiment was performed in a very small domain, so a next obvious question is whether these properties would still be exhibited in larger domains. Is the agent still able to learn a persistent preference for the curiosity-inducing location when the domain is larger, or when there are many possible targets? These questions motivated our second set of experiments, described in the next section.

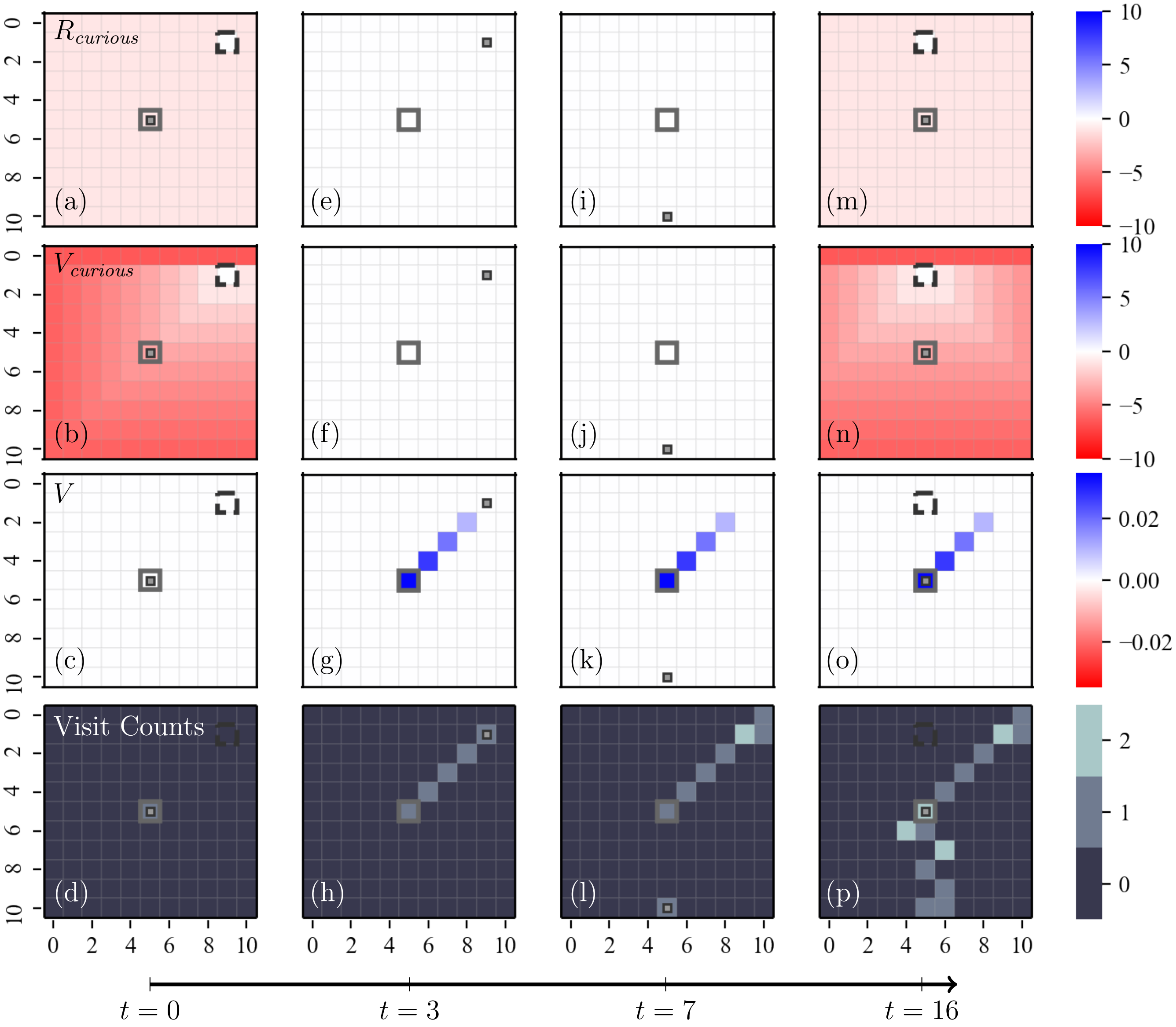


Figure 5.4: This figure is meant to offer intuition into the agent's learning behaviour by showing the agent's persistent value function V, curiosity value function $V_{curious}$, the curiosity reward function $R_{curious}$ used to generate $V_{curious}$, and the visit counts at the initialization of an example trial ($t = 0$), the first visit to an induced target ($t = 3$), after it has crossed off the top of the grid back to the bottom centre ($t = 7$) and the second visit to the curiosity-inducing location ($t = 16$). Note the difference in scale between V and $V_{curious}$. While it is not visually obvious, location (6, 4) has a value V of approximately 0.0003 at $t = 16$—the first step in learning a path to the curiosity-inducing location.

5.4 Second Set of Experiments: Domain Geometry Manipulations

5.4.1 Experimental Setup: Domain Geometry Manipulations

While the patterns we observed through the experiments described in Sections 5.3.1 and 5.3.2 are promising reflections of specific curiosity, we were curious about whether we would observe the same patterns in a larger domain. In a larger domain, there is more space for the agent to get 'lost,' and not pick up the patterns of behaviour demonstrating learned voluntary exposure and repeated cycles of curiosity. For this reason, in our second set of experiments, we manipulated the geometry, or shape, of our original 11×11 domain to make similar wide (11×101) and tall (101×11) domains. In these domains, we ran four experiments:

1. 30 trials of 5000 steps in wide (11×101) domain
2. 30 trials of 5000 steps in wide (11×101) domain without a junction location
3. 30 trials of 5000 steps in tall (101×11) domain with curiosity-inducing location near the bottom of the grid
4. 30 trials of 5000 steps in tall (101×11) domain with curiosity-inducing location in the centre of the grid

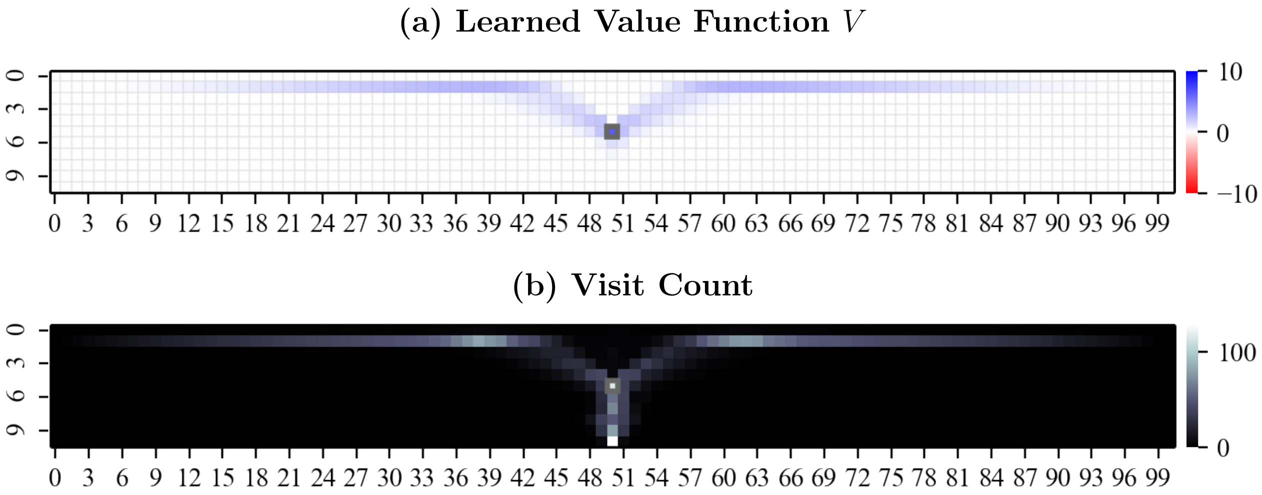
We explain these experiments in more detail in this section. In each of these domains with manipulated geometry, each key aspect of the primary domain has an analogue. The agent had the same six actions available (left, left-up diagonal, up, right-up diagonal, right, and stay-here). The targets were uniformly selected from the second row from the top of the grid: from $(1, 1)$ to $(1, 99)$ in the wide domain and from $(1, 1)$ to $(1, 9)$ in the tall domain.

In three of the four experiments, the junction location has an analogue: when the agent moves off the top of the grid, it is returned to the centre of the bottom row of the grid, which is $(10, 50)$ in the wide domain and $(100, 5)$ in the tall domain. In the second experiment, however, we removed the junction location, making the domain a true cylinder. When the agent moves off the top of the grid, it arrives at the bottom of the grid in the column it attempted to move into (e.g., if the agent moved on a left-up diagonal, it would arrive one column to the left of where it was along the top, unless it was against the left edge, in which case it would arrive in the bottom row in the same column).

Removing the junction location allowed us to explore how important it is for a curiosity-inducing location to be near the agent when the agent isn't curious. If the agent fails to find a distant curiosity-inducing location, it might not demonstrate the key properties of specific curiosity. Understanding this effect has important implications for the design of an appropriate curiosity-recognizing module. For example, we may need to ensure the module has a sufficiently low threshold for the induction of curiosity to obtain useful behaviour.

We further explored this concern by manipulating the location of the curiosity-inducing location in the tall domain. It was not obvious where to put the curiosity-inducing location in the tall domain: five rows up from the bottom, or in the vertical centre of the grid? As the third and fourth experiments of this set, we tried both natural possibilities for the curiosity-inducing location, with the third experiment performed with the curiosity-inducing location at $(95, 5)$ and the fourth with it at $(50, 5)$.

By manipulating the geometry of our original domain, we hoped to find out whether the initial patterns we observed in the first set of experiments generalized to larger domains. Further, larger domains might illuminate other patterns of behaviour that might improve our choices in the design of future, more sophisticated algorithms for machine curiosity.


(a) Learned Value Function V

(b) Visit Count

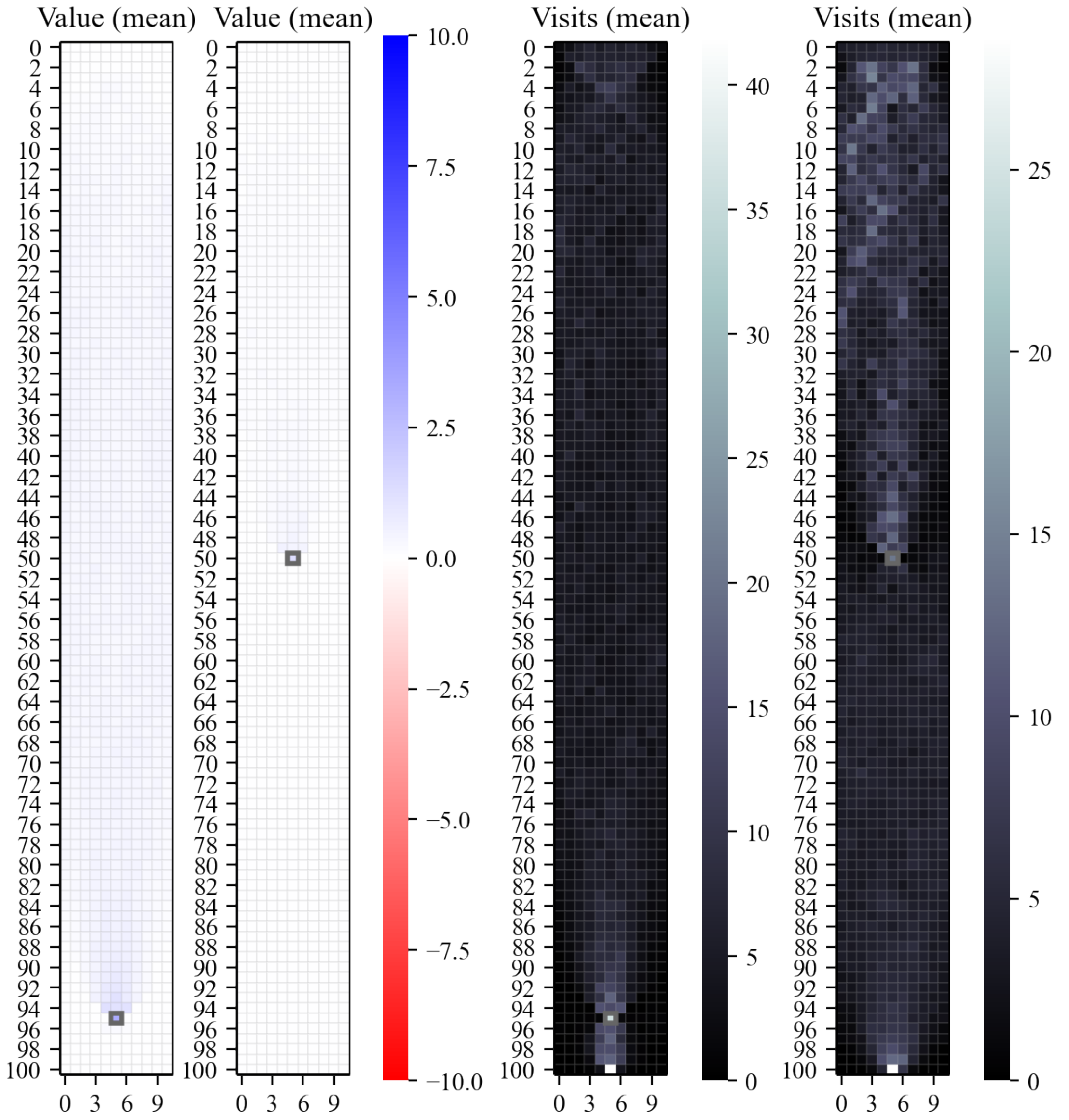
Figure 5.5: This figure shows the learned value function and visit counts in the wide domain for our simple reinforcement learning agent demonstrating properties of specific curiosity. This figure shows how any locations that are visited repeatedly while curious will accumulate value. Shown here are (a) the learned value function V and (b) the total visits the agent made to each location. Totals plotted for trials of 5000 steps and averaged over 30 independent trials. Note that the scale of the visit counts plot differs from that in Figure 5.2.

5.4.2 Results and Discussion: Domain Geometry Manipulations

Through these experiments with larger geometry-manipulated domains, we learned three key lessons:

1. **Even in expanded domains, following Algorithm 2 still results in properties of directedness, cessation when satisfied, and voluntary exposure.** Of these properties, we were least certain that we would observe voluntary exposure, but by the end of every trial of these experiments, the persistent value is highest at the curiosity-inducing location, which reflects this property. For an aggregate view, see Figures 5.5a and 5.6a,b.

   Directedness and cessation when satisfied are determined directly by the algorithm and do not rely on any learning, so it is unsurprising to see these



(a) **(b)** **(c)** **(d)**

Figure 5.6: This figure shows the learned value function and visit counts in the tall domain for our simple reinforcement learning agent demonstrating properties of specific curiosity. This figure shows that learning is slowed when time to complete a cycle of curiosity is increased, and slowed even more when the curiosity-inducing location isn't near any repeatedly visited location. Shown here are the learned value function, V, with (a) the curiosity-inducing location at $(95, 5)$ and (b) the curiosity-inducing location at $(50, 5)$, and the total visits the agent made to each location. Totals averaged over 30 independent trials of 5000 steps each.

properties reflected in videos of the agent's behaviour. The directed behaviour of the agent is also reflected in the visit counts for the wide and tall domains shown in Figures 5.5b and 5.6c,d, primarily in the upward-opening funnel shape from the curiosity-inducing location, which occurs because once the agent is in state curiosity, it only takes upward (or upward diagonal) actions to reach the targets at the top of the grid.

2. **Any part of the world that is repeatedly visited while the agent is in state curiosity acquires persistent value.** We already saw this phenomenon in the primary domain (Figure 5.2a,c), as persistent value accumulated in the funnel shape of locations leading from the curiosity-inducing location towards the targets. However, this phenomenon is more pronounced in the wide and tall domains: in Figures 5.5a and 5.6a,b, while a direct path from from the junction location to the curiosity-inducing location has accumulated some value, the magnitude of that value is imperceptible on the scale used for those figures, while the upward funnels are clear.

   In the wide domain, this upward funnel includes some of the potential target locations (see the distinctive 'bird-wing' shape in Figure 5.5a and the spread of orange lines in Figure 5.7), which might raise concern if you remember that we were aiming for targets *not* to accumulate value—remember, once you've satisfied your curiosity, you don't read the same page over and over again. However, this is a special case, where these locations accumulate value when they are visited for a different purpose: passing through them on the way to a curiosity-satisfying target. Depending on the context, there may be benefits to learning to value processes that have helped satisfy curiosity in the past, or this may be an undesirable side effect.

   From this phenomenon, note that while potential target locations do not accumulate persistent value by virtue of being targeted, they still may accumulate value if they are on direct paths to other targets. In the wide domain,

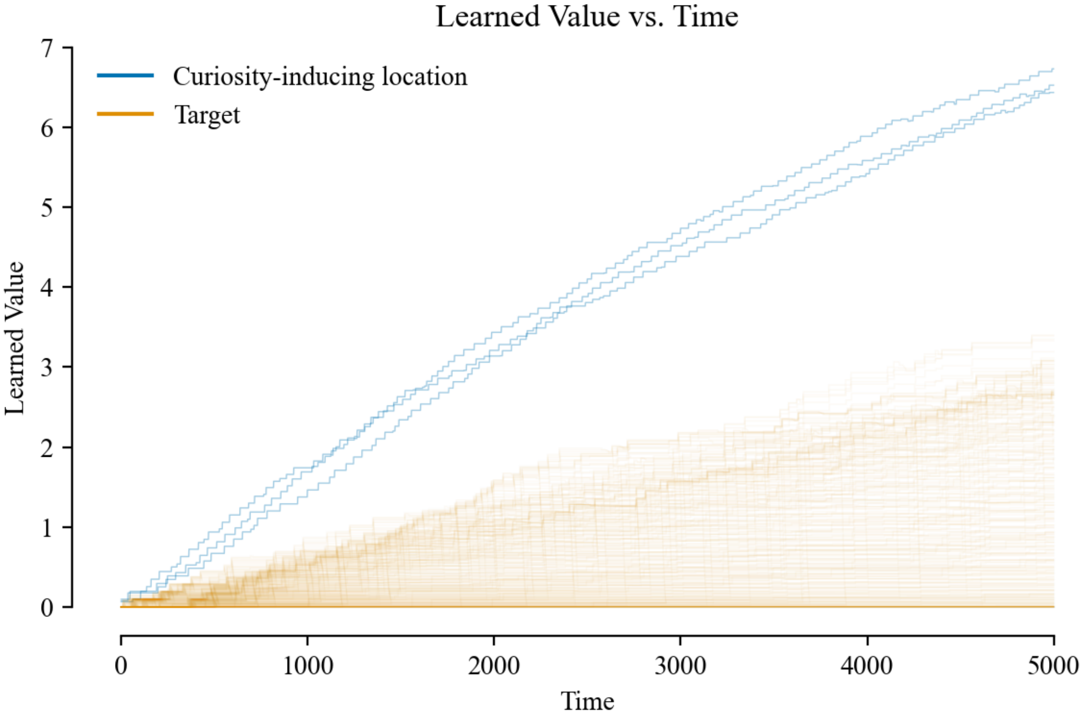


Figure 5.7: This figure shows the persistent value of the curiosity-inducing location (blue) and target locations (orange) in the wide domain over time for three trials. While the growth pattern for the curiosity-inducing location is similar to that seen for the primary domain (Figure 5.3), some of the target locations in the wide domain grow in value over time. There are three blue lines, with each showing the value of the curiosity-inducing location for a single trial. In orange, the value for each target is shown as a separate line (meaning there are 297 separate orange lines, 99 for each trial).

several of the potential target locations are on direct paths to other target locations, so they too accumulate value, as observed in the distinctive 'bird-wing' shape in Figure 5.5. While we pointed out that it is undesirable for targets to accumulate value (remember, once you've satisfied your curiosity, you don't read the same page over and over again),

Further, locations that are repeatedly visited between the curiosity-inducing location and induced targets (the funnel-shape above the curiosity-inducing location in the grid) accumulate more value than any path leading to the

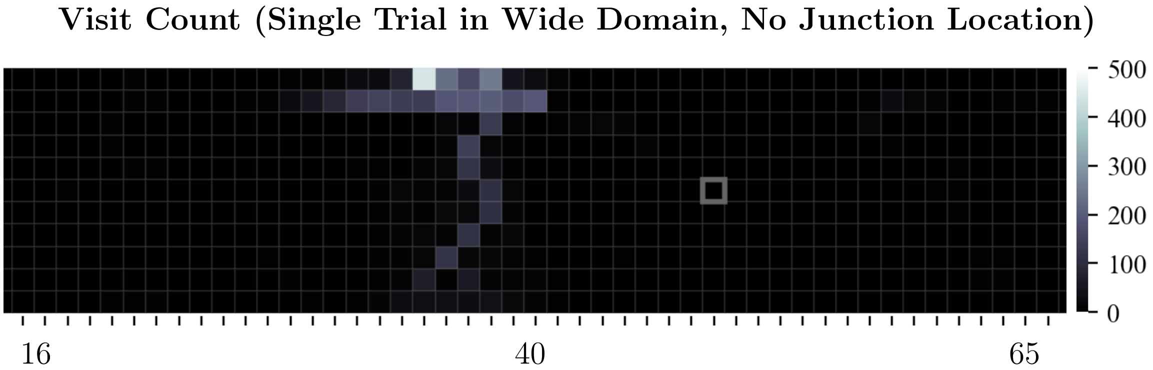


Figure 5.8: This figure shows the visit counts in a single trial in the wide domain with the junction location removed. While the agent's persistent value function *is* greatest at the curiosity-inducing location, as desired for voluntary exposure, the agent still does not find its way back to the curiosity-inducing location because it gets stuck re-visiting an area of the grid that accumulated value while the agent travelled from the curiosity-inducing location to a target. In this trial, the agent only visited the curiosity-inducing location twice (the first visit resulting in a target to the right of the curiosity-inducing location, and the second resulting in a target to the left of the curiosity-inducing location.

curiosity-inducing location from the junction location. While this difference was visible in the primary domain (Figure 5.2a,c) the difference in magnitude is much more apparent in the larger domains, especially the tall domain (Figure 5.6a,b).

The accumulation of value in any area visited by the agent while curious is important in the context of our exploration of whether an agent might 'get lost' if the curiosity-inducing location is too far away: the agent *can* get stuck in these areas of accumulated value and not find its way back to the curiosity-inducing location. We observed this exact problem when we removed the junction location from the wide world: the agent spends the majority of the trial example trial used to generate Figure 5.8 in an area to the left of the curiosity-inducing location, where it had previously accumulated value on the way to a target.

Thinking this scenario out beyond the 5000 steps of one trial, the agent should gradually learn that this ‘sticky’ area is not valuable. While the agent is not curious, the value of the locations it visits slowly return toward zero. In Figure 5.9, we see that the potential target locations visited repeatedly in this trial gradually decrease in value over time. Because the agent rapidly learned a persistent value function where the curiosity-inducing location has the highest persistent value, after many time steps, it should theoretically return to the curiosity-inducing location once the value of these areas had decreased sufficiently. However, we can see from the shape of those curves in Figure 5.9 that this decrease will be ineffectually slow.

In many cases, we suspect that this limitation would not pose a problem. For example, the existence of a junction location is typical to biological learners: where the curiosity-inducing location is like a bookstore, the junction location is much like a home—a place the agent returns to regularly. Once you've learned a path from your home to the bookstore, you are readily able to follow your desire to expose yourself to curiosity. If you didn't return home, however, you might not figure out how to get back to the bookstore, as we observed in our experiments. This observation of our agent getting stuck is the most extreme example of our third and following lesson.

3. **Learning voluntary exposure requires multiple visits, and the less likely the agent is to return to a curiosity-inducing location, the slower this learning process will be.** In the wide world with the junction location removed, the agent rarely followed any repetitive path to the curiosity-inducing location. In many trials, the agent visited the curiosity-inducing location more than once, but did not have the opportunity to learn a habitual path. In these grid worlds, a single visit to the curiosity-inducing location extends the learned path by only one location. For readers unfamiliar with the learning behaviour of model-free reinforcement learning algorithms, you can think that, every time the agent stumbles upon a path it has already noted, it notes where it was before entering the path, then follows

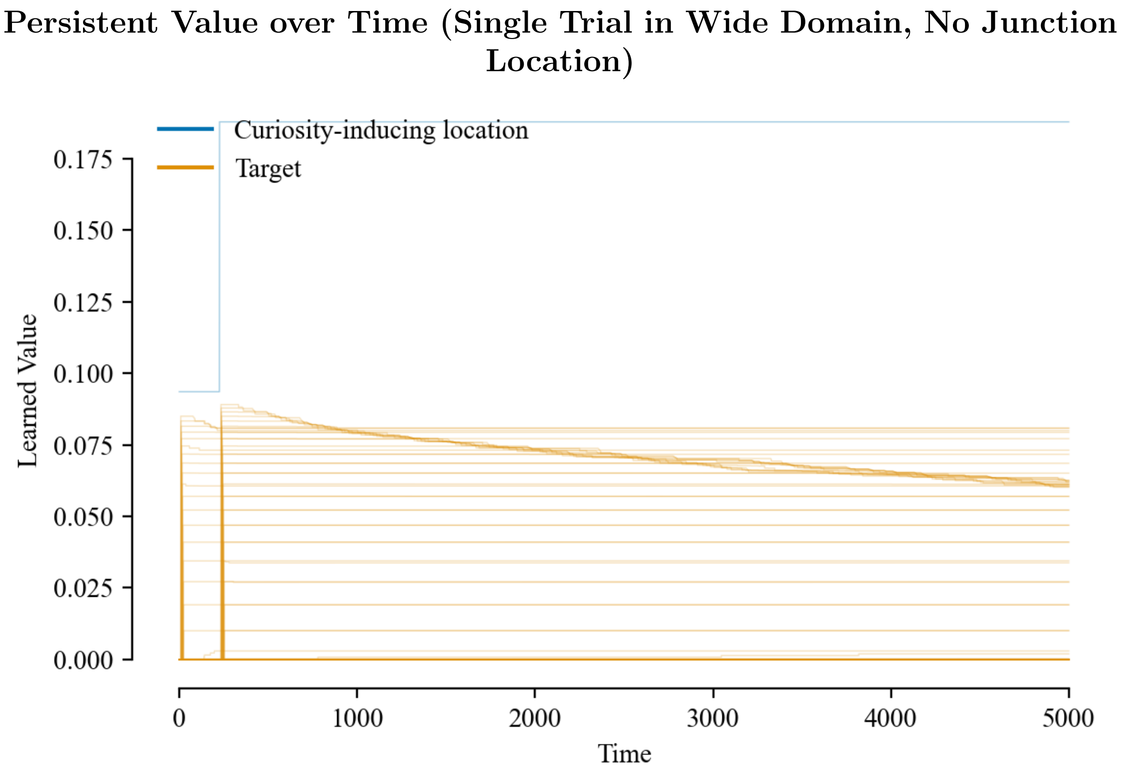


Figure 5.9: This figure shows the persistent value of the curiosity-inducing location (blue) and target locations (orange) over time for one trial in the wide domain with no junction location. While the curiosity-inducing location accumulates the most value, the agent gets stuck re-visiting a region of the grid that was on the way to a previous target. Some of the potential target locations are in this region, and so we can see their value grow when the agent visits them while curious, When the agent returns to these locations after curiosity has been satisfied, their value slowly declines. This decline is so slow that the agent will not unlearn its preference for them in a timeframe that we would consider reasonable. The value for each potential target location is shown in orange as a separate line (meaning there are 99 separate orange lines).

the path the rest of the way. This new note adds one more location to the path. Algorithmically, these 'notes' are made as increased persistent value. This procedure means that while developing increased value for the curiosity-inducing location occurs with even a single visit, developing *behaviour* that reflects voluntary exposure takes multiple visits.

The two experiments in the tall domain reflect our third lesson with more gradation. When the curiosity-inducing location is near the junction location, the agent learns a direct path between the two relatively quickly. When the curiosity-inducing location is placed further away, the agent skips by the curiosity-inducing location more often and spends more time wandering in the part of the domain above the curiosity-inducing location—slowed down by the 'sticky' parts of that region that have accumulated value by being visited when the agent is in a state of curiosity. As a result, the curiosity-inducing location accumulates more value and visits overall when it is placed close to the junction location (Figure 5.6a,c) than when it is placed further away (Figure 5.6b,d).

These lessons are valuable because, as described in Section 4.2.5, assuming curiosity can be used to direct agents towards fruitful learning opportunities, it is desirable for our agents to effectively and efficiently learn voluntary exposure to curiosity-inducing situations. Using Algorithm 2 or an adaptation of it will require recognizing the effect of domains on whether the agent will visit a curiosity-inducing location enough times while following its non-curious policy to learn habitual paths. With these lessons in mind, our next set of experiments probes the interplay of the properties within Algorithm 2.

5.5 Third Set of Experiments: Ablation of Properties

5.5.1 Experimental Setup: Ablation Study

The third and final set of experiments is an ablation study. The term *ablation* comes from neuroscience, where one way to experimentally learn about the function of part of the brain is to destroy that part and see how the behaviour of the learner changes. In our case, particular design elements were included in the algorithm to account for each of three key properties of specific curiosity and in this set of experiments, we ablated (i.e., removed) each of these design elements in turn—directedness, cessation when satisfied, and voluntary exposure—running the same experiment as described in Section 5.3.1 and observe what has changed from the results we observed in Section 5.3.2. For each property, a reminder from Sections 5.1.3-5.1.5 of how the property is incorporated into Algorithm 2 and a description of how the algorithm proceeds with the property removed is included in the latter part of this subsection.

Beyond using ablations to study the design elements for each key property, in this section we also include an experiment with an ablation of the design element included to account for the aversive quality of specific curiosity. While we pointed out that there is some controversy in whether specific curiosity should be characterized as aversive in Footnote 15 and did not argue for aversive quality to be a key property for the implementation of machine specific curiosity, we did include aversive quality in designing Algorithm 2, as described in Section 5.1.5. An aversive quality may not be necessary for specific curiosity generally, but removing it should have a notable effect on the results of using Algorithm 2, as the aversive quality both guides the agent to the target and determines the value that is learned in the persistent value function, so we tested its importance via an ablation of the associated algorithmic elements, detailed below.

However, since aversive quality defines the curiosity value function, $R_{curious}$, we expected that removing it completely for an ablation should result in uninteresting, random behaviour: the agent will neither have a guide to the target to use for directedness nor learn to value the curiosity-inducing location for voluntary exposure. For this reason, asking what happens when aversive quality is ablated entirely is less interesting than asking what happens if it is replaced with *positive* quality. How does the agent's learning and behaviour change if $R_{curious}$, rather than being negative everywhere except the target, is *positive* everywhere, most positive at the target? To answer this question, we additionally ran an experiment where we modified the curiosity reward function in this manner, as detailed below.

Running this series of ablations should allow us to better understand Algorithm 2 by demonstrating how each property contributes to the agent's learning and behaviour. Each of these experiments is described in more detail in the following subsections.

Ablation of Directedness To ablate directedness, we removed Line 9 and the **if** statement structure around it.

```
 8: if there is currently a curiosity target (i.e. the agent is curious) then
 9:     $x'$, $R \leftarrow$ move greedily w.r.t. $V_{curious}(x)$        ▷ Directed Behaviour
10: else
```

In Algorithm 2, the agent follows the gradient value function $V_{curious}$ greedily to the target, but in the ablation, the agent instead follows an ϵ-greedy policy with respect to V, whether or not a target exists. Equivalently, Line 11,

```
$x', R \leftarrow$ move epsilon-greedily w.r.t. $V(x)$        ▷ Ties broken uniform randomly,
```

always determines the agent's next action and get the next state, x', and reward, R.

Ablation of Cessation When Satisfied To ablate cessation when satisfied, we removed Lines 15 and 16 of Algorithm 2 and the **if** statement structure around them.

```
14: if agent observation $x'$ is the target then
15:     destroy the current target
16:     reinitialize $V_{curious}$ to zeros                ▷ Cessation when Satisfied
```

With these lines removed, if the agent visits the target, the target remains and the agent continues to greedily follow the gradient value function $V_{curious}$.

Ablation of Voluntary Exposure To ablate voluntary exposure, we removed the edit we made to the learning update in Line 12. As a reminder, Line 12 in the original algorithm was as follows:

$$\delta \leftarrow R + \gamma \cdot V(x') - [V(x) + V_{curious}(x)] \qquad \triangleright \textbf{ Voluntary Exposure}$$

The ablation reverts that line to the standard TD error, as follows:

$$\delta \leftarrow R + \gamma V(x') - V(x).$$

With the $V_{curious}(x)$ term removed, the temporary value function does not affect updates to the persistent value function.

Ablation of Aversive Quality To ablate aversive quality, we removed Line 6,

$$\text{generate } R_{curious} = \begin{cases} 0, & \text{if transitioning into target state} \\ \mathbf{-1}, & \text{otherwise} \end{cases} \qquad \triangleright \textbf{ Aversive Quality}$$

which accounts for the aversive quality of specific curiosity in Algorithm 2. Without Line 6, $R_{curious}$ remains zero for all state transitions, as $R_{curious}$ was initialized to zero in Line 2.

Replacing Aversive Quality with Positive Quality In addition to ablating aversive quality, we also tested replacing it with positive quality. To achieve this replacement, we modified $R_{curious}$. In the original algorithm, the special reward function, $R_{curious}$, is negative everywhere except at the target, inspired by the aversive quality of curiosity. A different, but still appropriate gradient (temporary value function) could be formulated using an alternative, positive reward model, $\tilde{R}_{curious}$, that would similarly direct the agent towards the target. While there are many possible definitions, we used the following definition:

$$\tilde{R}_{curious}(s,a,s') = \begin{cases} 1 & \text{if } s \text{ is the target} \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

When $\tilde{V}_{curious}$ is generated via value iteration from $\tilde{R}_{curious}$, it should guide the agent to the target much like the original $V_{curious}$ does. However, in the original learning update in Line 12, subtracting the non-positive $V_{curious}(x)$ meant that the agent learned a positive value. To get the same effect with the with the newly defined, non-negative $\tilde{R}_{curious}$, $\tilde{V}_{curious}(x)$ must be *added*; consequently, we modified Line 12 to the following.

$$\delta \leftarrow R + \gamma \cdot V(x') - V(x) + \tilde{V}_{curious}(x) \qquad \triangleright \textbf{Voluntary Exposure}.$$

5.5.2 Results and Discussion: Ablation Study

Our primary result from our ablation study was that ablating any algorithmic element that supports a key property or that supports the aversive quality results in behaviour that no longer reflects specific curiosity. In particular, the agent no longer exhibits the cycles of curiosity we observed in the primary domain or the wide and tall domains (with junction location). In this section, we will examine the resultant behaviour for each experiment in this set and their implications.

Ablation of Directedness When directedness is ablated, arbitrary paths through the domain accumulate value. This learning behaviour contrasts with what happens when using the original Algorithm 2, where direct paths from the curiosity-inducing location to the appropriate satisfier accumulate value (Figure 5.2a,c). Because the agent with the ablation chooses randomly when faced with equally-valuable maximally-valued alternatives, exactly which path accumulates value varies from trial to trial. This randomness results in the visual difference between the value function for a single trial (top panel of Figure 5.10a) and the aggregated

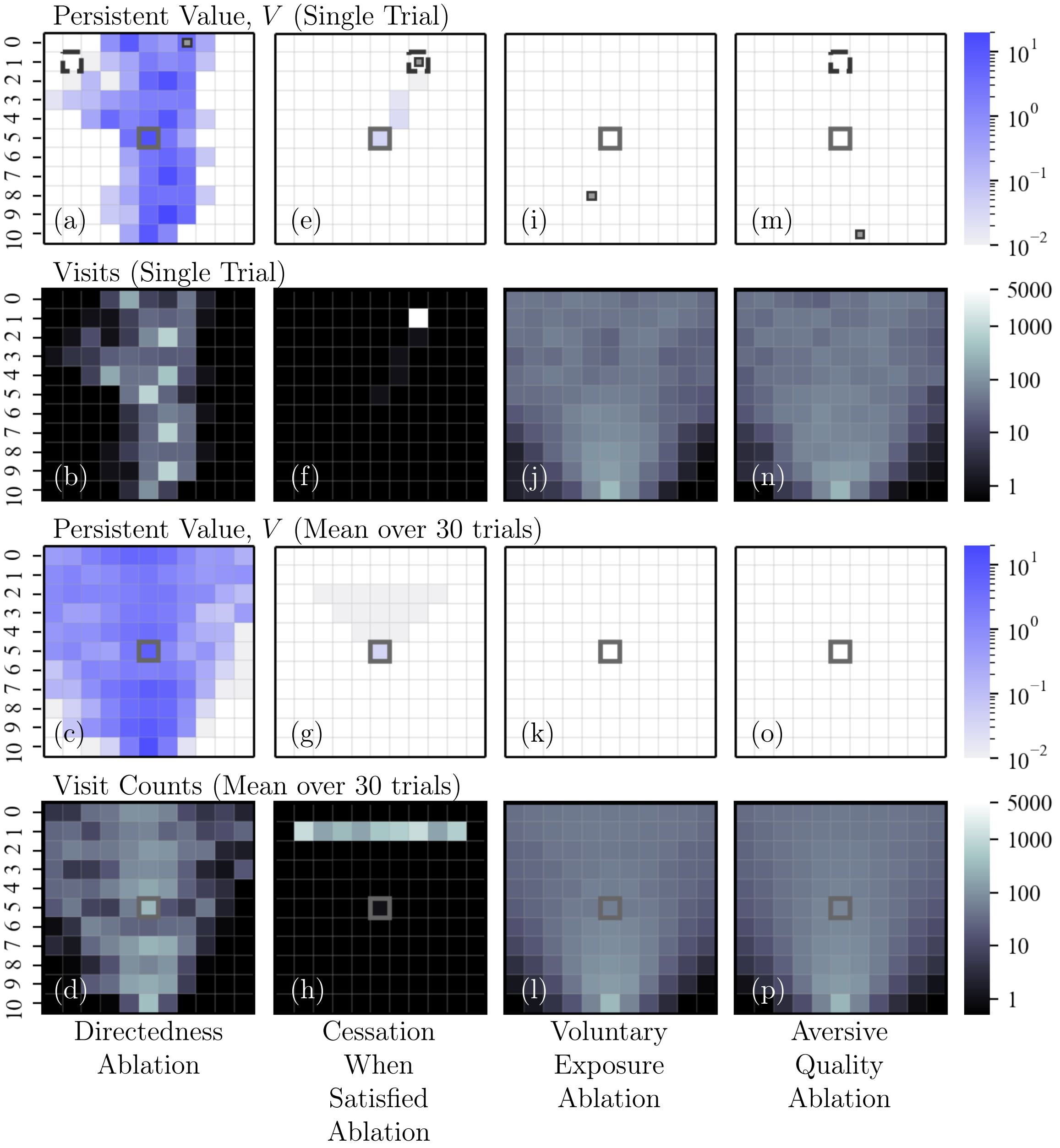


Figure 5.10: This figure shows the persistent value function and visit counts in the primary domain for each ablation. From this figure, we can see that all of the properties are used together to achieve behaviour that learns to value the curiosity-inducing location, but not the targets. A single ablation is shown in each column. The top and third rows show the learned value function V with zero-valued locations in white, while the second and bottom rows show the visit counts with zero-valued locations in black, each after 5000 time steps. The first two rows show a single representative trial for each ablation, while the bottom two rows are averaged over 30 trials. All subfigures are on logarithmic scales.

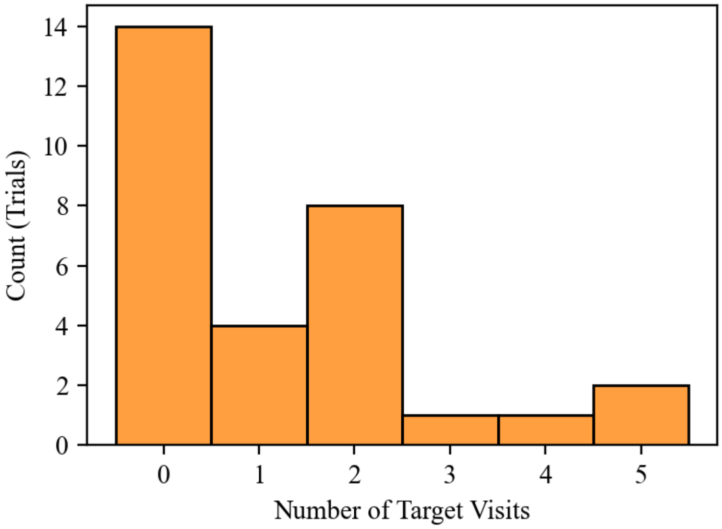
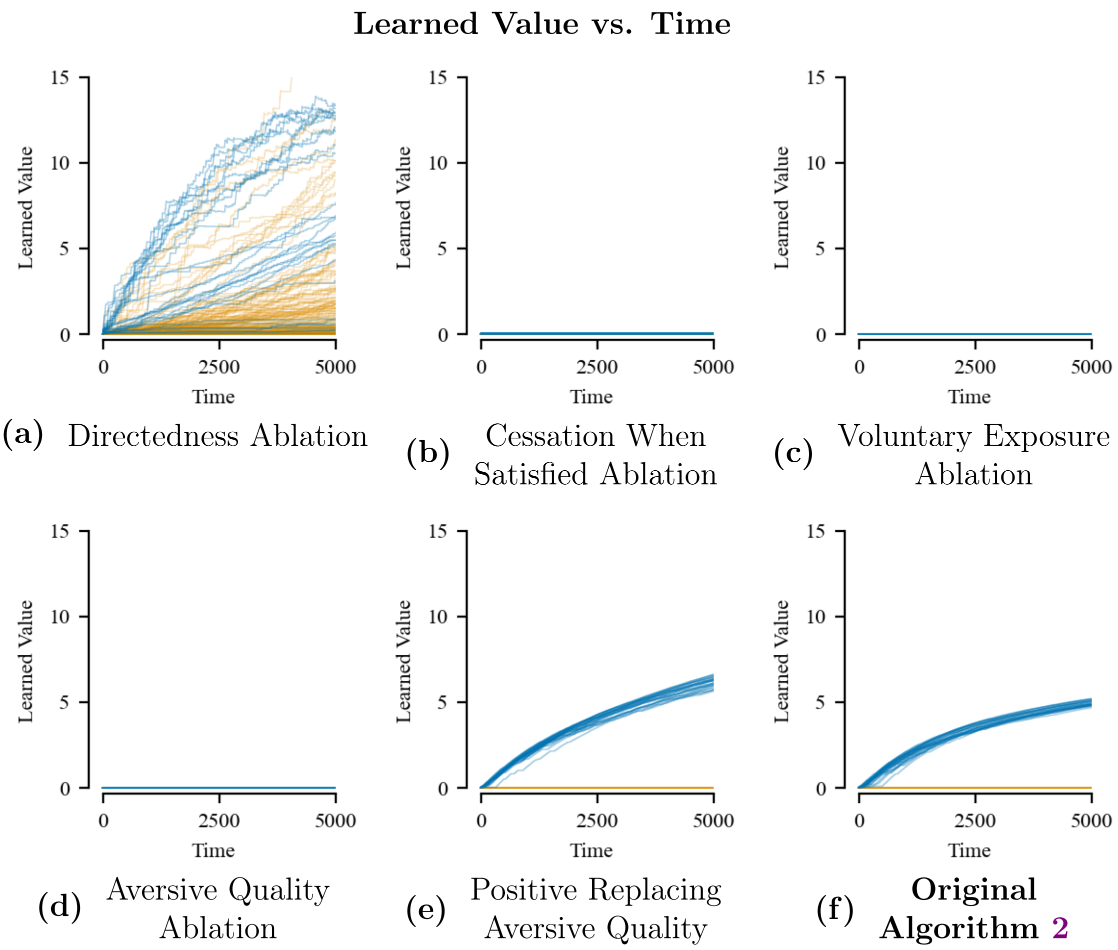


Figure 5.11: This figure shows a histogram of the number of visits to targets in each trial when directedness is ablated. The histogram is right-skewed. The height of a bar for a given number of target visits is the number of trials with exactly that number of visits. The experiment included 30 trials total.

value function across trials (third panel from the top of Figure 5.10a). Further, the persistent values for the curiosity-inducing location and the potential targets vary substantially from trial to trial, depending on which path through the domain the agent gets stuck on (Figure 5.12a).



(a) Directedness Ablation

(b) Cessation When Satisfied Ablation

(c) Voluntary Exposure Ablation

(d) Aversive Quality Ablation

(e) Positive Replacing Aversive Quality

(f) **Original Algorithm 2**

Figure 5.12: This figure shows the persistent value of the curiosity-inducing location (blue) and target locations (orange) over time for all thirty trials for each ablation and for the original Algorithm 2. Panel (f) shows the same data as Figure 5.3. In the directedness ablation (a), both the curiosity-inducing locations and targets grow over time, with large variation. When cessation when satisfied is ablated (b), the values of both the curiosity-inducing location and the targets remain constant over time, with the value of the curiosity-inducing location reaching 0.315 during the agent's first and only visit to the curiosity-inducing location at time $t = 0$ and the value of the targets remaining 0 throughout. The learned value for the ablations of voluntary exposure (c) and aversive quality (d) remains zero everywhere. When aversive quality is replaced with positive quality (e), the learned values for the curiosity-inducing location and the targets are similar to those in the original algorithm, but the value of the curiosity-inducing location grows slightly more quickly over time.

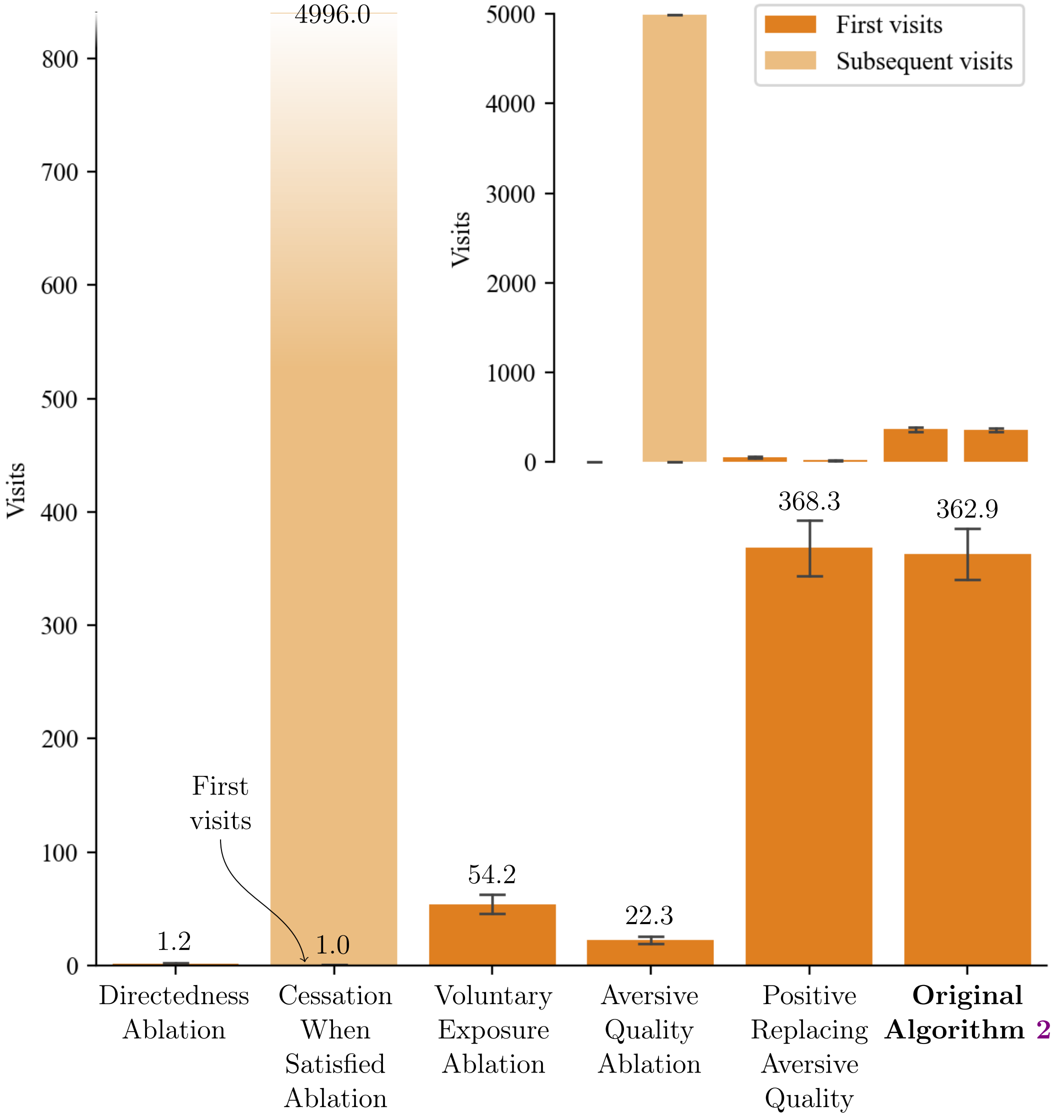


Figure 5.13: This figure allows for comparing the number of times the agent visits a target (specified by the curiosity recognizer, not just possible target locations) for each ablation in Section 5.5.1 with the original algorithm. The ablation of cessation when satisfied has two stacked bars: dark orange showing the number of "first visits" to a target (counting only one visit after the target has been generated) and light orange showing the number of subsequent visits. The upper right inset graph is zoomed out to show the full bar. The original algorithm and the modification replacing aversive quality with positive quality have similar target visit counts while the other ablations result in substantially fewer first visits. Error bars show the standard deviation across 30 trials.

On average, ablating directedness results in the fewest number of visits to generated targets as compared to the original algorithm and the other ablations (mean of 1.2, comparison shown in Figure 5.13). While the agent sometimes chooses a path that visits the target, that target is removed once it is visited (cessation when satisfied). Because the agent has already accumulated so much value on its meandering path, it tends to remain on that path. If the next target is not generated on or near that path, then the agent is unlikely to visit it. The result is that the distribution of the number of target visits across trials is right skewed, with the agent failing to visit any targets at all in nearly half of the trials (Figure 5.11).

With directedness ablated, the agent's behaviour is characterized not by cycles of curiosity, but by randomly chosen cycles which continually accumulate more value. The agent does not seek out a satisfier, so unless it stumbles on a satisfier by chance, it can stay in a state of 'curiosity,'[5] continually accumulating value in a randomly chosen region of the domain with no off switch.

Ablation of Cessation When Satisfied When cessation when satisfied is ablated, the agent takes a direct path from the curiosity-inducing location to the target and remains at that target for the remainder of the trial. In each trial, the agent has one first visit to a target, and 4996 subsequent visits (Figure 5.13). As an example, the visit counts and persistent value at the end of a single trial are shown in the top two panels of Figure 5.10, showing how the agent accumulated persistent value on its path to its first target much like the agent following Algorithm 2 shown in Figure 5.4g. Since this ablation agent's target is not removed, the agent does not move on from this location. The agent therefore only visits one target in every trial and does not benefit from curiosity motivating it towards multiple new experiences.

Of the ablation experiments, the ablation of cessation when satisfied is the only experiment where the agent consistently learns a persistent value function that is

[5] Of course, this state no longer reflects curiosity in any way, and is more reflective of wireheading (for one description of the term wireheading, see Yampolskiy, 2014).

maximal at the curiosity-inducing location. Such a value function would reflect voluntary exposure, but since the agent remains fixated on a target, it never has the opportunity to reflect the behaviour component of voluntarily visiting curiosity-inducing situations. Neither the value of the curiosity-inducing location nor the targets changes over time, with the value of the curiosity-inducing location reaching 0.315 during the agent's first and only visit to the curiosity-inducing location at time $t = 0$ and the value of the targets remaining 0 throughout (Figure 5.12). Because the agent remains fixated on a single target, the agent spends little time visiting areas with accumulated persistent value, instead spending the rest of its time at the target (Figure 5.10e–h).

The removal of cessation when satisfied might remind some readers of the reactive behaviour of intrinsic-reward learners, who are driven to visit a novel state repeatedly. Despite this parallel, the ablation of cessation when satisfied is not directly comparable to intrinsic reward methods. As we discussed in Section 4.3.3.1, multiple computational intrinsic rewards are designed to decay as the agent visits its target over and over. In our ablation, the level of motivation remains static throughout each trial. We experimented with a decaying motivation level, but do not include the (rather uninteresting) results here because the conceptual purpose of intrinsic rewards is so unlike that of specific curiosity that the comparison is inappropriate in our test domain. Again, two primary benefits of this decaying property of intrinsic rewards are promoting multiple visits to check for consistency (for example of a stochastic reward) or staying on an exploration frontier. In our simple, rewardless domain, there is no benefit to repeated visits, nor are the curiosity targets generated on an exploration frontier.

Ablation of Voluntary Exposure When voluntary exposure is ablated, no persistent value accumulates in any part of the domain (Figure 5.10c and the line plot in Figure 5.12 are zero everywhere). This occurs because the learning update step that flips value from the curiosity value function into the persistent value function has been removed. However, the agent does still demonstrate directed behaviour between the curiosity-inducing location and the targets. As a result, there is a faint but visible funnel shape above the curiosity-inducing location in the bottom panel of Figure 5.10c (compare with the bottom panel of Figure 5.10d, which reflects a true random walk through the domain). This directed behaviour helps the agent make more (first) target visits than any of the other ablations (mean of 54.2, see Figure 5.13), though still far fewer than an agent following the original Algorithm 2.

Ablation of Aversive Quality When the aversive quality of curiosity is ablated, $V_{curious}$ is not generated, so the agent experiences no difference in value or reward throughout the domain. For this reason, the agent acts randomly throughout each trial. The resulting estimated value function and visit counts are shown in Figure 5.10(d). No value is accumulated anywhere in the grid, as emphasized by Figure 5.12(d), which shows that the estimated value for all of the targets and the curiosity-inducing location remain zero throughout each trial.

Replacing Aversive Quality with Positive Quality More interesting than ablating aversive quality is replacing it with positive quality. In this experiment, the agent's behaviour is very similar to that of of the agent following original Algorithm 2 as described in Section 5.3.2. The number of visits to generated targets for the agent with this replacement are within error of that of the original algorithm, shown in Figure 5.13. Both agents' final persistent value functions and visit counts are similar (Figure 5.14). The main difference between the persistent value functions is a matter of scale, in that the estimated values for the experiment using positive quality are generally higher. This difference is also visible in the associated lineplot in Figure 5.12, where the value of the curiosity-inducing locations grows more quickly when aversive quality is replaced with positive quality. However, the difference is not only in scale; for example, note that squares $(7, 0)$ and $(4, 1)$ have different mean values between in Figure 5.14a (positive quality) and 5.14b

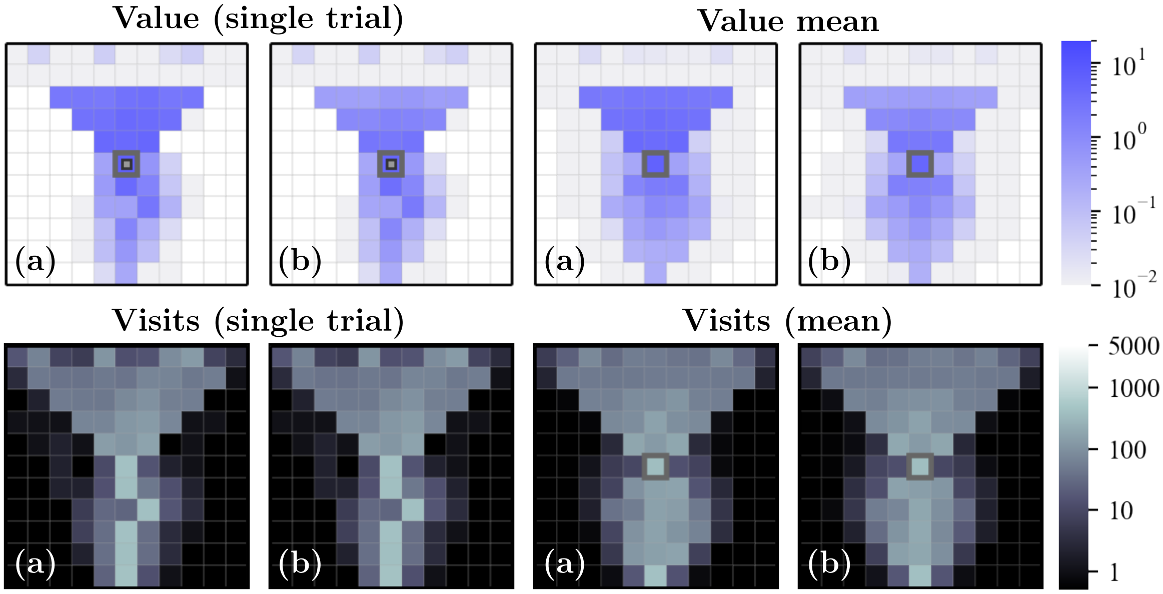


Figure 5.14: This figure shows the persistent value functions and visit counts for the experiment where (a) aversive quality is replaced with positive quality alongside the same for (b) the original algorithm (same data as Figure 5.2, but on a logarithmic scale) for visual comparison. The behaviour and value learned with positive quality (a) is very similar to that of the original algorithm (b)—indeed, given the same random seed, the behaviour is identical for 1071 steps—but value accumulates at different rates in each case, so the value functions do differ by more than just scale. All subfigures are on logarithmic scales.

(aversive quality).

The agent using positive quality should and does behave differently than the agent following the original Algorithm 2, because the value functions generated by $R_{curious}$ and $\tilde{R}_{curious}$ have different shapes. For this reason, the persistent value function accumulates value at different rates in each case. However, a takeaway from this experiment is that using a negative value function, or what we call the aversive quality of Algorithm 2, is not necessary for creating cycles of behaviour reflecting specific curiosity.

In humans, it may be true that the information seeking associated with specific curiosity "is motivated by the aversiveness of not possessing the information more than it is by the anticipation of pleasure from obtaining it" (Loewenstein, 1994,

p. 92), but from the perspective of our simplistic computational RL agent, our choice of implementation for each did not result in appreciably different behaviour.

***

Taken together, the experiments in our ablation study show us that, in the context of Algorithm 2, the properties of directedness, cessation when satisfied, and voluntary exposure work together, and that curious behaviour is noticeably impaired when any one property is missing.

5.6 General Discussion: Benefits of the Properties of Specific Curiosity

Our ablation study provides initial evidence for the interconnected nature of the properties of specific curiosity—effective learning behaviour isn't achieved via one or two properties; the properties work together. Indeed, the benefits of each property are so interwoven that they are best understood via their combined influence on the whole of specific curiosity.

Flexible specialization to a learner's context: In Section 4.2.7, we noted that the property of coherent long term-learning, the last of our five properties, closes the loop of how curiosity can guide a learner over a lifetime. Curious biological learners, including humans, live long lives, but certainly not long enough to experience every possible situation that the world could throw at them. Further, humans have found ways to survive in a diverse set of possible climates, cultures, and contexts. We believe specific curiosity supports that ability.

Some of what we learn is passive—we learn just by 'being there.' Our brains persistently and automatically take the observations from our senses and work to integrate them into our knowledge of the world (Chater, 2018, p. 138). This passive learning helps build up a foundation of knowledge that is somewhat local to the learner's particular context. Then, specific curiosity insists that we learn actively, almost any time we aren't attending to obvious needs to keeping our bodies going and species alive. And in particular, the property of coherent long-term learning biases our active learning towards specific concepts that we are ready to build onto our existing knowledge (Wade and Kidd, 2019), often towards new information defines a connection across a gap in our existing knowledge (Loewenstein, 1994), much of which may have been passively learned. The better connected our knowledge is, the more useful it is.

Very importantly, curiosity supports us when our context changes. By being biased to direct the learner towards information to support connections to the learners' existing knowledge, specific curiosity may direct us to learn new information that will help us transfer our existing skills and knowledge into a novel context. How many of our curiosity questions start by orienting on "Wait, that wasn't what I was expecting"? In those kinds of situations, whether we observed a toy performing an unexpected function or a suspect in our mystery performing a suspicious action, there is a waiting connection to be made. The jack-in-the-box doesn't appear except when ...? People don't dump heavy body-sized bags into the lake in the dead of night except when ...? Dark fluid doesn't end up on white-paper walls except when ...? In these situations, making our inostensible referent ostensible repairs the broken understanding created by our prior generalizations failing to hold in a new context, giving us a more accurate foundation of knowledge on which to act.

Specialization as contribution to societal knowledge: Looking at our favourite biological model of curiosity, the human, another key feature of humans is that they are social. Humans in particular seem to get an incredible benefit from individuals having different specialties (Hauser, 2018, p. 7). If each individual instead developed unspecialized, broad knowledge, then the overlap—the knowledge held by our entire society—would be similarly broad, but unfortunately shallow. We would know very little about many things, as a group. Instead, the overlap of all these narrow, deep specializations developed over time lends itself to providing not only broad, but deep knowledge for our larger society, networked together by humanity's social nature.

When a piece of specialized knowledge turns out to be generally applicable, it can be transferred via social contact across a connected network of learners, a more general societal benefit. While we noted that humans are our favourite model of curiosity, the societal transmission of new, specialized behaviours—innovations—appears to benefit social non-human animals too. One example involves birds, British blue tits, who famously discovered how to pierce the foil caps on milk bottles to access the cream on top. The behaviour was first observed in 1921, but by the end of the 1940s, the behaviour was widespread across the U.K. (Aplin et al., 2013, p. 1226, Yong, 2014). Experiments by Aplin et al. (2013) involving teaching new foraging behaviours to blue tits have provided further evidence that blue tits socially transmit new, useful behaviours across their communities (p. 1230).

As another example, researchers on the isolated Japanese islet of Koshima observed a macaque (a variety of monkey) washing the sand off of a potato—a new behaviour that they had never observed before (Kawai, 1965, pp. 2–3). In the years thereafter, the researchers observed a wave of social learning until nearly the whole colony seemed to clean their potatoes before eating (p. 4). Interestingly, the same macaque who seems to have come up with the potato-cleaning behaviour appeared to later be the first macaque to demonstrate a behaviour of 'wheat washing' (p. 13). Initially, when humans scattered wheat across the sand, the monkeys would painstakingly pick up each grain one by one. 'Wheat washing,' on the other hand, involves gathering up the sand with the grains and tossing them into water, which allows the sand to drift to the bottom while the wheat floats on top (p. 12). This behaviour also spread throughout the colony, though not quite as pervasively (p. 12). In analogy with the specialized human chef who might design new recipes and share them, the originator of these behaviours might have a specialized interest in food preparation, to the benefit of their community.

The need for directedness towards inostensible referents: Coherent long-term learning requires directedness towards inostensible referents. An inostensible concept, supported by the properties that the learner already knows will be true of the inostensible referent, is the form taken by the next—metacognitively most appropriate (Wade and Kidd, 2019)—learning opportunity to coherently build on existing knowledge. The only sensible activity to experience curiosity-satisfying observations is to take a systematic sequence of actions to obtain the specific information that will make their inostensible concept ostensible. Given that the learner will never have a perfect model of the world including the inostensible referent (it wouldn't be inostensible, in that case!), the learner must make a best guess and adapt their plan as they proceed.

The usefulness of cessation when satisfied: Cessation when satisfied creates efficiency by taking advantage of the following idea: what makes an appropriate answer depends on the question. For some inostensible referents, repetitive behaviour might be appropriate: just think back to the example with the peculiar-sounding floorboard. A reasonable way to acquire sufficient evidence to decide if your weight transfer caused the noise is indeed to try repeating that weight transfer several times—once might be a fluke, but three or four times seems sufficient to suggest you're causing the noise.

Our formulation of cessation when satisfied was directly inspired by the behaviour generated by intrinsic-reward methods and how it contrasts with specific curiosity. The reactive nature of intrinsic rewards motivate a learner to re-experience a state multiple times. Specific curiosity, on the other hand, doesn't require this kind of repetition for all inostensible concepts. For many questions, only single experiences of each curiosity-satisfying observation is required. After all, you don't need to re-read 'whodunnit' out of curiosity—once you've read that part once, your curiosity for that particular inostensible referent can end.

In most cases, there are multiple possible forms of evidence that we would accept as curiosity-satisfying. One of the seemingly most important for humans is testimony from others (Harris, 2012). This kind of evidence rarely requires repetitive behaviour (unless the person you're asking isn't listening). If anything, it may require probes into how reliable the source of information is, or seeking a second opinion via a different mode of behaviour. Not only does the kind of evidence required vary depending on the inostensible referent, the reliability required of an the answer varies even further. How important is it that we have the right answer, versus just a working theory?

In this way, humans demonstrate extreme flexibility when it comes to specifying what makes an acceptable curiosity-satisfying situation. While our next prototypes of curious machines may not have such beautifully tailored recognition systems for sufficient evidence for their curiosity to be satisfied, it is time to move away from simple repetition as a proxy for the satisfaction of curiosity.

The importance of transience: A close relative of cessation when satisfied, transience is necessary for functional curiosity in biological learners. After all, humans and animals can only (physically) be in one place at one time, and their attention is thought to be a similarly limited resource (Lloyd and Dayan, 2018, p. 2). Constantly reorienting those limited bodily and attentional resources is impractical, and so committing to a single goal for a period of time benefits the learner (Lloyd and Dayan, 2018, p. 2). Specific curiosity is one example of this kind of goal-directed behaviour. As detailed by Lloyd and Dayan (2018), goal-directed behaviour will be more effective in an uncertain environment if the behaviour of the agent can be interrupted by time-sensitive demands, like attending to a loud noise that might indicate danger, pangs of hunger (Simon, 1967, p. 35), or even the recognition that, in the past, you regretted a decision made in a similar situation (Hoch and Loewenstein, 1991, p. 498).

In this sense, transience also has a strong relationship with stay-switch decisions observed in animal decision making, wherein an animal constantly balances its near-term reward with its expectations of long-term average reward, thereby governing the persistence of its current behaviour (c.f., human patch foraging and the marginal value theorem; Constantino and Daw, 2015).

Even more critically, transience resolves some of the trouble that 'un-realizable' inostensible concepts could cause. When we say that some inostensible concepts are un-realizable, we are noting that the very nature of inostensible concepts is that, in some cases, they can't be made ostensible. Not everything that could be dreamt up by a learner is necessarily a thing that the learner could find, especially if the lifetime of the learner is limited. While I could find myself curious about the location of the nearest Earth-orbiting teapot, I would struggle to find out whether such a teapot exists, never mind its location. When asking about unknowns, it is necessary that a learner might sometimes ask the wrong questions, and so needs to be able to stop chasing curiosity-satisfying situations that don't exist.

The condition of specific curiosity is a concerted effort to make an inostensible concept ostensible. It requires adaptive planning, which is likely resource-heavy, and, in biological learners, active movement of the body towards perceiving curiosity-satisfying observations. Transience helps the learner manage an all-or-nothing effort to satisfy their curiosity, because it means that behaviour and use of attentional resources can be fully reallocated to other matters as needed.

Voluntary exposure over curiosity by chance: Accepting the premise that curiosity will be valuable to our machine agents, we certainly don't want our agents to avoid curiosity. But do we really want voluntary exposure, or would it be sufficient for the agent to stumble across curiosity-inducing observations without increased preference for them?

Before we provide our answer to that question, we would like to note some subtlety to the voluntary exposure that humans exhibit. Humans have been observed to voluntarily expose themselves to some observations that they are aware will be curiosity-inducing (Loewenstein, 1994, pp. 75-76), like a puzzle or the latest bingeable TV show, but there are other curiosity-inducing observations that humans will not choose to expose themselves to.

Ruan et al. (2018) presented the results of some experiments where humans exhibited specific curiosity, but not voluntary exposure. Their experiments centred on what they called an “uncertainty creation–resolution process” (p. 556). In their experiments, this process consisted of the learner being “first teased with some missing information” (e.g. presented with a trivia question) “and then given that information” (p. 556). In four experiments (see the discussions of *Choice* for Studies 1 through 4, pp. 561–565), they found that, given a choice between experiencing an ‘uncertainty creation–resolution process’ or not, most of their participants chose not, suggesting that they did not exhibit voluntary exposure.

The authors offered two hypotheses about why their participants failed to exhibit voluntary exposure. One hypothesis was that seeking uncertainty, or choosing to be exposed to curiosity-inducing observations, might be a trait exhibited by a minority of people (Ruan et al., 2018, p. 560). The very healthy industries producing puzzles, mysteries, and cliff-hanger-laden television series that we mentioned in Section 4.2.5 bring this hypothesis into doubt. Their other hypothesis was that, in cases where people voluntarily expose themselves to curiosity-inducing situations, they “have control over when they receive the missing information” (Ruan et al., 2018, p. 560), which merits further study.

Based on our computational case study, we suggest a novel hypothesis that voluntary exposure might be learned via multiple experiences of curiosity being induced in similar situations. It is possible that while these people have learned to predict the positive experience associated with their favourite forms of curiosity-inducing situations, be they crossword puzzles, mystery novels, or mathematical problems, the experimental setup might be too unfamiliar to lead to voluntary exposure. In this way, considering the value of voluntary exposure brings us back to coherent long-term learning. Tying voluntary exposure to individual interest enhances learner specialization, a key benefit of coherent long-term learning as we argued above.

Whatever domains we specialize our voluntary exposure towards, specific curiosity tends to drive us into a solving process. Whether racking our brains for the right word for a crossword or picking out the right clues to solve a murder mystery, curiosity helps us build and solidify our knowledge. In particular, human learning benefits from retrieval practice, and curiosity helps us when we're in danger of forgetting something we have already been exposed to, and if that something is coming up again, it is likely a somewhat consistent part of the context we interact in day-to-day. Learners have to practice to develop skills, so if we don't have to attend to a more pressing matter like food or sleep or whatever, practicing these kinds of solving processes, especially within an area of individual interest, so as to build up knowledge in a specialized, individual way, is a really good idea.

Most learners are thought to juggle many competing interests. Which of an learner's needs should be prioritized over another is probably situational and difficult to answer, but we argue that all else being equal, intelligent agents imbued with curiosity should choose to expose themselves to curiosity-inducing situations. With the right implementation, artificial curiosity should direct the agent towards fruitful learning opportunities, much as biological curiosity is thought to (Wade and Kidd, 2019, p. 1382). Assuming that our design of machine curiosity manages to do the same, we want our machine agents to seek curiosity, which starts with a preference for curiosity-inducing situations—that is, voluntary exposure.

5.7 Conclusion

Curiosity is central to biological intelligence, and machine curiosity is an area of emerging activity for machine intelligence researchers in their pursuit of learning agents that can engage in complex, information-rich environments like the natural world. Throughout this chapter and the preceding one, we have directly connected insight and empirical evidence from the study of human and animal curiosity to advances in machine intelligence. In particular, we have for the first time translated the idea of specific curiosity to the domain of machine intelligence and shown how it can lead a reinforcement learning machine to exhibit key behaviours associated with curiosity. As a first major contribution of this work, we presented a comprehensive, multidisciplinary survey of animal and machine curiosity. We then used that body of evidence to synthesize and define what we consider to be five of the most important properties of specific curiosity:

1. directedness towards inostensible referents;
2. cessation when satisfied;
3. voluntary exposure;
4. transience;
5. coherent long-term learning.

As a second main contribution of this work, we constructed a proof-of-concept reinforcement learning agent interleaving the most salient and immediate properties of specific curiosity. We then conducted empirical sweeps and ablations to probe the role that these integrated properties have on the agent's curious behaviour (and how the removal of individual properties substantially impacts this behaviour). Our computational specific curiosity agent was found to exhibit short-term directed behaviour, update its long-term preferences, and adaptively seek out curiosity-inducing situations. One major insight we draw from this work is that the separation of curiosity-inducing situations from curiosity-satisfying situations is critical to understanding curious behaviour.

We consider this study a landmark synthesis and translation of specific curiosity to the domain of machine learning and reinforcement learning. It is our hope that this exploration of computational specific curiosity will inspire a new frontier of interdisciplinary work by machine intelligence researchers, and that it will further provide new computational mechanisms to model and study the phenomenon of curiosity in the natural world.

Chapter 6

Discussions and Future Work

Throughout this dissertation, we focused on machine learning algorithms designed with the purpose of generating properties of curiosity. In the first chapter, I centred this document on the following argument:

> *We must experiment and think beyond the most commonly used frameworks being used for machine curiosity if we want to secure the benefits of human-like curiosity for our machine learners.*

In this final chapter, I will summarize the contributions that have been presented in this dissertation to support this argument and discuss potential future research directions that would build upon these contributions.

6.1 Summary of Contributions

From the beginning of this document, I emphasized two separable reasons that algorithms inspired by curiosity merit further research. The first reason was that it can be expected to benefit machine learning systems. The inspiration of curiosity as a concept has already inspired impressive breakthroughs in AI, particularly in improving performance in domains where exploration is known to be difficult and also in generating learning trajectories that more closely resemble those exhibited by young animals and humans. As human activities become more closely coupled with AI activities, humans can be expected to benefit from AI systems that exemplify great teammates, companions, and members of society—and arguably, that means AI systems that are curious (Gino, 2018; Kashdan et al., 2013, p. 150; Zurn, 2020, pp. 227–228). The second, then, is to contribute to our understanding of curiosity as a whole, allowing the processes that AI researchers use to create new systems develop precision in the language we use to describe curiosity and develop new theories of curiosity (cf. Newell, 1970).

Given these exciting motivators, it should come as no surprise that a number of RL-based approaches to machine curiosity have been proposed where the reward is specifically crafted or the RL algorithm is modified to generate curious behaviour, many examples of which have been discussed throughout this dissertation. While these existing methods have shown promise in a number of real-world and simulated targeted domains, there are few suggestions of unified ways to compare these different approaches. For us to be able to build on the growing number of curiosity-inspired algorithms, we need to work to develop ways to understand the landscape of such methods.

The first contributions of this thesis were in response to this gap, aiming to develop uncomplicated approaches allowing for comparison of many different curiosity-inspired RL algorithms, controlling for other variables where possible. In particular, the result was a new family of experimental domains, Curiosity Bandits (Chapter 3), which allow for comparison across computational intrinsic motivation methods—which encompass many of the curiosity-inspired approaches published to date. The introduction of this family of domains led to the first comprehensive empirical comparison of different intrinsic reward mechanisms (Chapter 3). The initial results using the Curiosity Bandit showed that different machine curiosity methods can result in very diverse behaviour (Chapter 3) and even a single curiosity method may result in widely varying behaviour dependent on how its various parameters are set (Appendix B). Furthermore, one cannot just set these parameters to maximize some objective measure of success. Unlike in traditional reinforcement learning, where the goal is to maximize return, the metric of a learner's success "at curiosity" is not obvious. To make progress possible, we need principles directing what we want to achieve through curiosity.

One potential line on this open-ended problem of understanding what we want to achieve through machine curiosity incorporates a better understanding of the beneficial properties of human curiosity. Along this line, this dissertation includes the novel integration of ideas from multiple disciplines on specific curiosity to lay out five key properties of specific curiosity (Chapter 4). The practice of delineating a concept by posing a series of design issues to be met, as shown with our five properties of specific curiosity, is an important practice in AI, historically undertaken by other scholars like Moore and Newell (1974, p. 1). Moving beyond this initial delineation, we further deepen our understanding of these properties as they might manifest in computational systems by including a proof-of-concept reinforcement learning agent and a careful ablation study demonstrating how the properties interact to result in behaviour characteristic of curiosity (Chapter 5).

Turning towards the central goal of this document, the proof-of-concept agent described in Chapter 5 does *not* engage the standard computational framework leveraged by the majority of curiosity-inspired computational RL agents: intrinsically-motivated reinforcement learning. Indeed, the chapter that sets the foundations for that demonstration, Chapter 4, emphasizes that, while intrinsic reward approaches offer a number of valuable benefits, they still have a number of limitations which make them unsuitable for achieving the proposed key properties. Recognizing this disjuncture between machine curiosity and biological curiosity was made possible by viewing the behaviour of multiple methods in aggregate, via the methods described in Chapter 3. Taken as a whole, the contributions described in this document support the central argument, that *we must experiment and think beyond the most commonly used frameworks being used for machine curiosity if we want to* *secure the benefits of human-like curiosity for our machine learners.* At the same time, this document represents a novel view into machine curiosity and, in particular, offers a robust and detailed characterization of specific machine curiosity.

6.2 Discussion, Reflection, and Future Directions

This section is is both backward- and forward-thinking. It represents a collection of reflections on the work in this dissertation. Some of these reflections speak to historical influences on aspects of this work, the inclusion of which may offer methodological value to readers and, additionally, provide a record to accompany these ideas as they go forward to have potential influence on our "constantly renegotiated" (IEEE TechXplore, 2023, p. 6) understanding of a concept like curiosity (Ady and Rice, 2023). The remainder—the forward-thinking parts—speak to research directions that arose from the work presented in this dissertation.

6.2.1 Challenges for Machine Curiosity Research

Through my work thus far, I have developed an understanding of both existing approaches to computational curiosity and key properties of curiosity recognized by non-computational perspectives. Given this understanding, I have identified several fundamental problems that existing methods for machine curiosity have in common.

These problems stem from structure that nearly all existing methods share. The majority of approaches use a standard IMRL approach: they develop a reward structure and use standard reinforcement learning methods to maximize the return. I wish to highlight the following three challenges:

1. Existing curiosity approaches have many parameters to set, but the objective is typically unclear and the effects of changes to these parameters are unknown.

2. The majority of proposed approaches to machine curiosity require the use of a separate exploration method. Both within computing science and beyond, curiosity and exploration are considered closely related. Indeed, the study of machine curiosity is often offered as a solution to the exploration-exploitation dilemma. The problem is not necessarily the separation of exploration and curiosity; in some sense, the choice of exploration method is simply another parameter whose effects we do not understand. However, I contend that we must consider that choice to be a critical implementation decision and share it as such, especially since many of the curiosity performance metrics we have used thus far are partially judging how effectively a curious learner explores, which is highly affected by how the exploration method behaves in conjunction with the curiosity approach.

3. The majority of existing methods require multiple visits to such interesting parts of the world before such a reward is considered more than a fluke, and behaviour is changed to motivate the learner to return to that part of the world. They do not motivate the agent to visit parts of the world that it has never visited before, which is a hallmark of curiosity.

These are important challenges, not only for the study of computational curiosity and exploration, but for the future of machine intelligence as a whole.

6.2.2 Unifying the Field of Machine Curiosity

The word 'curiosity' does not mean the same thing to all people. It has been related to a host of other concepts, like exploration (Fowler, 1965; Berlyne, 1966), information-seeking (Gottlieb et al., 2013), novelty-seeking (Kashdan et al., 2009), surprise (Charlesworth, 1964), learning progress (Gottlieb et al., 2013), boredom (Schmidhuber, 1991b), play (Ngo et al., 2012), virtuosity (Kubovy, 1999), confidence (Schmidhuber, 1991a), competence (Moulin-Frier and Oudeyer, 2012), flow (Malone, 1981, cited by Webster et al., 1993), and intrinsic motivation (Oudeyer et al., 2007).

It may seem that the most appropriate response is to choose an objective for curiosity, and design approaches to achieve it. However, I am making a specific philosophical choice in my development of a foundation for machine curiosity research: we are not ready to choose a single objective for curiosity. As already stated, one important takeaway from my preliminary work is that the different existing approaches to machine curiosity produce different behaviour. This is perhaps unsurprising, since if you ask any sample of people what curiosity is, you typically receive many different answers. Because of the many different intuitions about curiosity, a designer of an intelligent 'curious' system may have in mind any of multiple different objectives for curiosity. Having this choice not only empowers the creators of curious systems, but also allows researchers to benefit from the many advantages of a broad definition of curiosity, such as those argued by Kidd and Hayden (2015, p. 456).

Kidd and Hayden (2015, p. 456) further argue that there has been premature emphasis on "divide-and-conquer approaches" to the taxonomy of curiosity; in agreement, I propose that our understanding of curiosity will be improved by a system that does not try to label some definitions as not being curiosity and instead tries to recognize both similarities and differences between phenomena captured by a broad definition.

The future of the field of machine curiosity requires unification of the existing knowledge. Currently, the commonalities and differences between approaches are obscured; sometimes superficial differences hide deeper similarities, and vice versa. Even though the techniques appear disparate, they often make the same mistakes in different guises. Failing to unify existing approaches is detrimental to the progress of the field, as effort is put into 'new' approaches that are plagued by old problems.

6.2.3 Other Potential Properties

While the five properties described and explored in Chapters 4 and 5 are the best recommendations we can make with the given current state of curiosity research, we recognize that human curiosity is an active area of study, and there may be other properties that turn out to be critical for specific curious agents.

For example, Zurn (2019) has carefully recorded a number of recognizable properties, observing that curiosity "works at the limits of what we know," "deploys meticulous attention in its investigations," "allows new questions to develop out of old ones," "facilitates a network of collective inquiry," has elements of "childlike playfulness," shows "interest in what is uselessly strange," and results in "rabbit trails of distracted interest" (pp. 26-27). Zurn's components of *works at the limits of what we know* and *deploys meticulous attention in its investigations* parallel the specificity of our property of *directedness towards inostensible referents*. Similarly, *allows new questions to develop out of old ones*, together with *results in rabbit trails of distracted interest* and perhaps even *interest in what is uselessly strange* may stem from mechanisms for *coherent long-term learning*. It is, however, uncertain whether *childlike playfulness* is likely to arise from the five properties as given. This property may indeed be a salient feature of human curiosity to consider, as the "child-like nature" of curiosity also arose as a higher-level code in the qualitative study performed by Aslan et al. (2021, pp. 8, 9) focused on beliefs about curiosity and interest. The five properties as presented here represent an important characterization, but only a snapshot of our knowledge in time. As researchers, we should challenge ourselves to incorporate new understandings about what is important for machine curiosity as they develop, not only in our subfield communities, but across disciplines (Ady and Rice, 2023).

6.3 Ethical Commentary

At the beginning of this chapter, I made reference not only to the myriad of benefits that curiosity brings to humans but also to how machines imbued with curiosity might equally benefit humans. And yet, while the academic obligation to motivate this work compels an emphasis on curiosity as it is often commodified and made capitalisitically desirable, I want to echo Shankar's (2020) warning that the rhetoric of curiosity purely for its instrumental benefits dangerously constrains curiosity, delimiting and inhibiting our ability to explore our own curiosities (p. 114–115).

The warning holds with respect to the second reason for researching algorithms inspired by curiosity: helping us better understand curiosity as a whole. We walk a line as we strive to ethically build AI. If our work as AI scientists contributes to how a concept like curiosity is understood, do we have any ethical obligation to build curiosity as it is, rather than as we want it to be? Conversely, if we have an ethical obligation to build curiosity as it ought to be—for example, this could describe 'safe' curiosity, miraculously excluding those aspects that would kill the cat—then, who will decide what curiosity ought to be? Importantly, will our work as artificial intelligence scientists change what curiosity means?

6.4 The Title of this Thesis

Specific Machine Curiosity—what's that? Edward L. Walker, who edited Berlyne's posthumously published book fragment *Curiosity and Learning*, wrote in his 1978 introduction to the publication:

> Curiosity is defined as an internal state occasioned when subjective uncertainty generates a tendency to engage in exploratory behavior aimed at resolving or partially mitigating the uncertainty. Curiosity is always curiosity about something specific and it is pertinent only to specific exploration and not at all to diversive exploration.

In alignment with Berlyne's view, to me, specific curiosity is the one true curiosity.

Bibliography

Abdallah, S. and Plumbley, M. (2009). Information dynamics: patterns of expectation and surprise in the perception of music. *Connection Science*, 21(2-3):89–117.

Achiam, J. and Sastry, S. (2017). Surprise-based intrinsic motivation for deep reinforcement learning. arxiv.org/abs/1703.01732.

Ady, N. and Pilarski, P. (2017a). Comparing reinforcement learning methods for computational curiosity through behavioural analysis. In *3rd Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, page 88, Ann Arbor, Michigan, USA.

Ady, N. M. (2017a). Computational curiosity: A review and a proposal for future research. Technical report, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada. Completed for CMPUT 605: Computational Curiosity.

Ady, N. M. (2017b). Curious actor-critic reinforcement learning with the dynamixel-bot. https://doi.org/10.7939/R3B853Z7S. Department of Computing Science, University of Alberta Education & Research Archive.

Ady, N. M. (2017c). Parameter screening for curious reinforcement learner motivated by unexpected error. doi.org/10.7939/R3G15TS0P. Department of Computing Science, University of Alberta Education & Research Archive.

Ady, N. M. and Pilarski, P. M. (2016). Domains for investigating curious behaviour in reinforcement learning agents. 11th Women in Machine Learning Workshop (WiML 2016).

Ady, N. M. and Pilarski, P. M. (2017b). Unifying curious reinforcement learners. In *Designing for Curiosity: An Interdisciplinary Workshop, ACM CHI Conference on Human Factors in Computing Systems (CHI 2017)*, Denver, CO, USA. May 6–11.

Ady, N. M. and Rice, F. (2023). Interdisciplinary methods in computational creativity: How human variables shape human-inspired AI research. In *Proceedings of the 14th International Conference on Computational Creativity (ICCC'23)*.

Ady, N. M., Shariff, R., Günther, J., and Pilarski, P. M. (2022a). Five properties of specific curiosity you didn't know curious machines should have. arxiv.org/abs/2212.00187. Submitted to the *Journal of Artificial Intelligence Research*.

Ady, N. M., Shariff, R., Günther, J., and Pilarski, P. M. (2022b). Prototyping three key properties of specific curiosity in computational reinforcement learning. In *5th Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, June 8-11, Providence, Rhode Island.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. (2017). Hindsight experience replay. In *Advances in Neural Information Processing Systems*.

Antos, A., Grover, V., and Szepesvári, C. (2008). Active learning in multi-armed bandits. In *International Conference on Algorithmic Learning Theory*.

Aplin, L. M., Sheldon, B. C., and Morand-Ferron, J. (2013). Milk bottles revisited: social learning and individual variation in the blue tit, Cyanistes caeruleus. *Animal Behaviour*, 85(6):1225–1232.

Appley, M. H. (1978). Curiosity and learning. *Motivation and Emotion*, 2(2):97.

Arnone, M. P., Small, R. V., Chauncey, S. A., and McKenna, H. P. (2011). Curiosity, interest and engagement in technology-pervasive learning environments: a new research agenda. *Educational Technology Research and Development*, 59(2):181–198.

Aslan, S., Fastrich, G., Donnellan, E., Jones, D. J. W., and Murayama, K. (2021). People's naïve belief about curiosity and interest: A qualitative study. *PLOS ONE*, 16(9):1–20.

Aubret, A., Matignon, L., and Hassas, S. (2023). An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy*, 25(2):327.

Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902. Algorithmic Learning Theory.

Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B*, 63(3):329–339.

Bagot, L., Mets, K., and Latré, S. (2020). Learning intrinsically motivated options to stimulate policy exploration. In *4th Lifelong Machine Learning Workshop at ICML 2020*.

Bajcsy, R., Aloimonos, Y., and Tsotsos, J. K. (2018). Revisiting active perception. *Autonomous Robots*, 42(2):177–196.

Balcan, M.-F., Beygelzimer, A., and Langford, J. (2009). Agnostic active learning. *Journal of Computer and System Sciences*.

Baldassarre, G. and Mirolli, M. (2013). Intrinsically motivated learning systems: An overview. In Baldassarre, G. and Mirolli, M., editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 1–14. Springer.

Baldassarre, G. and Parisi, D. (2000). Classical and instrumental conditioning: From laboratory phenomena to integrated mechanisms for adaptation. In Meyer, J.-A., Berthoz, A., Floreano, D., Roitblat, H. L., and Wilson, S. W., editors, *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, volume 6 of *From Animals to Animats*, pages 131–139. MIT Press.

Baldassarre, G., Stafford, T., Mirolli, M., Redgrave, P., Ryan, R. M., and Barto, A. (2014). Intrinsic motivations and open-ended development in animals, humans, and robots: an overview. *Frontiers in Psychology*, 5.

Baranes, A. and Oudeyer, P.-Y. (2009). R-IAC: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1(3):155–169.

Baranes, A., Oudeyer, P.-Y., and Gottlieb, J. (2015). Eye movements reveal epistemic curiosity in human observers. *Vision Research*, 117:81–90.

Barrett, L. F. (2017). *How Emotions Are Made: The Secret Life of the Brain.* Houghton Mifflin Harcourt.

Bartlett, F. (1958). *Thinking: An experimental and social study.* Basic Books, New York.

Barto, A. (2010). What are intrinsic rewards signals? *The Newsletter of the Autonomous Mental Development Technical Committee*, 7(2):3.

Barto, A., Mirolli, M., and Baldassarre, G. (2013). Novelty or surprise? *Frontiers in Psychology*, 4.

Barto, A. G. (2013). Intrinsic motivation and reinforcement learning. In Baldassarre, G. and Mirolli, M., editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 17–47. Springer.

Barto, A. G. and Şimşek, O. (2005). Intrinsic motivation for reinforcement learning systems. In *Yale Workshop on Adaptive and Learning Systems*.

Barto, A. G., Singh, S., and Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In Triesch, J. and Jebara, T., editors, *Proceedings of the 3rd International Conference on Development and Learning*, pages 112–119. UCSD Institute for Neural Computation.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 1471–1479. Curran Associates, Inc.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.

Benedict, B. M. (2001). *Curiosity: A cultural history of early modern inquiry.* University of Chicago Press.

Berlyne, D. E. (1949). Interest as a psychological concept. *British Journal of Psychology*, 39(4):184.

Berlyne, D. E. (1950). Novelty and curiosity as determinants of exploratory behaviour. *British Journal of Psychology*, 41(1-2):68–80.

Berlyne, D. E. (1954). A theory of human curiosity. *British Journal of Psychology. General Section*, 45(3):180–191.

Berlyne, D. E. (1957). Conflict and information-theory variables as determinants of human perceptual curiosity. *Journal of experimental psychology*, 53(6):399–404.

Berlyne, D. E. (1960). *Conflict, arousal, and curiosity.* McGraw-Hill Book Company.

Berlyne, D. E. (1963). Motivational problems raised by exploratory and epistemic behavior. In Koch, S., editor, *Psychology: A study of a science. Study II: Empirical substructure and relations with other sciences. Volume 5. The process areas, the person, and some applied fields: Their place in psychology and in science*, pages 284–364. McGraw-Hill.

Berlyne, D. E. (1966). Curiosity and exploration. *Science*, 153(3731):25–33.

Berlyne, D. E. (1970). Novelty, complexity, and hedonic value. *Perception & Psychophysics*, 8(5):279–286.

Berlyne, D. E. (1978). Curiosity and learning. *Motivation and Emotion*, 2(2):97–175.

Bermejo-Berros, J., Lopez-Diez, J., and Gil Martínez, M. A. (2022). Inducing narrative tension in the viewer through suspense, surprise, and curiosity. *Poetics*, 93(101664):1–16.

Berns, G. S., Laibson, D., and Loewenstein, G. (2007). Intertemporal choice – toward an integrative framework. *Trends in Cognitive Sciences*, 11(11):482–488.

Berseth, G., Geng, D., Devin, C. M., Rhinehart, N., Finn, C., Jayaraman, D., and Levine, S. (2021). SMiRL: Surprise minimizing reinforcement learning in unstable environments. In *International Conference on Learning Representations.*

Biehl, M., Guckelsberger, C., Salge, C., Smith, C., and Polani, D. (2018). Free energy, empowerment, and predictive information compared. In *The Ninth International Conference on Guided Self-Organisation (GSO-2018): Information Geometry and Statistical Physics*, Leipzig, Germany. Max Planck Institute for Mathematics in the Sciences.

Bonawitz, E. B., Ferranti, D., Saxe, R., Gopnik, A., Meltzoff, A. N., Woodward, J., and Schulz, L. E. (2010). Just do it? Investigating the gap between prediction and action in toddlers' causal inferences. *Cognition*, 115(1):104–117.

Brafman, R. I. and Tennenholtz, M. (2002). R-max – A general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. (2019a). Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations (ICLR 2019)*.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. (2019b). Exploration by random network distillation. In *International Conference on Learning Representations (ICLR 2019)*.

Bylinskii, Z., DeGennaro, E., Rajalinghamd, R., Ruda, H., Zhang, J., and Tsotsos, J. (2015). Towards the quantitative evaluation of visual attention models. *Vision Research*.

Cabi, S., Colmenarejo, S. G., Hoffman, M. W., Denil, M., Wang, Z., and Freitas, N. (2017). The Intentional Unintentional Agent: Learning to solve many continuous control tasks simultaneously. In Levine, S., Vanhoucke, V., and Goldberg, K., editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 207–216. PMLR.

Charlesworth, W. R. (1964). Instigation and maintenance of curiosity behavior as a function of surprise versus novel and familiar stimuli. *Child Development*, pages 1169–1186.

Chater, N. (2018). *The Mind Is Flat: The Remarkable Shallowness of the Improvising Brain*. Yale University Press.

Chater, N. and Loewenstein, G. (2016). The under-appreciated drive for sense-making. *Journal of Economic Behavior & Organization*, 126(Part B):137–154. Special Issue: Thriving through Balance, Edited by Dennis J. Snower.

Chen, A., Xiang, M., Wang, M., and Lu, Y. (2022). Harmony in intelligent hybrid teams: the influence of the intellectual ability of artificial intelligence on human members' reactions. *Information Technology & People*, ahead-of-print.

Chesterton, G. K. (1911). *The Innocence of Father Brown*. Standard Ebooks. Retreived from https://standardebooks.org/ebooks/g-k-chesterton/the-innocence-of-father-brown/text/single-page.

Clement, B., Roy, D., Oudeyer, P.-Y., and Lopes, M. (2015). Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining*.

Cobbe, K., Nichol, A., Achiam, J., Isola, P., Ray, A., Schneider, J., Clark, J., Brockman, G., Sutskever, I., Barry, B., Storkey, A., Efros, A., Pathak, D., Darrell, T., Brock, A., Antoniou, A., Jastrzebski, S., Pilipiszyn, A., and Wang, J. J. (2018). Reinforcement learning with prediction-based rewards. Blog Post. Retrieved from https://openai.com/blog/reinforcement-learning-with-prediction-based-rewards/. OpenAI.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*.

Colas, C., Karch, T., Lair, N., Dussoux, J.-M., Moulin-Frier, C., Dominey, P., and Oudeyer, P.-Y. (2020). Language as a cognitive tool to imagine goals in curiosity driven exploration. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3761–3774. Curran Associates, Inc.

Colas, C., Karch, T., Sigaud, O., and Oudeyer, P. (2022). Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey. *Journal of Artificial Intelligence Research*, 74:1159–1199.

Colas, C., Sigaud, O., and Oudeyer, P.-Y. (2018). GEP-PG: Decoupling exploration and exploitation in deep reinforcement learning algorithms. In *International Conference on Machine Learning*.

Constantino, S. M. and Daw, N. D. (2015). Learning the opportunity cost of time in a patch-foraging task. *Cognitive, Affective, & Behavioral Neuroscience*, 15(4):837–853.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons.

Csikszentmihalyi, M. (1991). *Flow-the Psychology of Optimal Experience*. Harper Perennial, New York.

Csikszentmihalyi, M. (1993). *The evolving self: a psychology for the third millennium.* New York, NY: HarperCollins Publishers.

Dan, O., Leshkowitz, M., and Hassin, R. R. (2020). On clickbaits and evolution: curiosity from urge and interest. *Current Opinion in Behavioral Sciences*, 35:150 – 156. Curiosity (Explore vs Exploit).

Darden, L. and Maull, N. (1977). Interfield theories. *Philosophy of Science*, 44(1):43–64.

Dayan, P. and Niv, Y. (2008). Reinforcement learning: The Good, The Bad and The Ugly. *Current Opinion in Neurobiology*, 18(2):185–196. Cognitive neuroscience.

de Abril, I. M. and Kanai, R. (2018). Curiosity-driven reinforcement learning with homeostatic regulation. In *International Joint Conference on Neural Networks*.

Deci, E. L. (1975). *Intrinsic Motivation*. Springer US.

Deci, E. L. and Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer US.

Degris, T. and White, A. (2020). Automatic step-size adaptation for reinforcement. *Manuscript in Preparation*.

Dember, W. N. and Earl, R. W. (1957). Analysis of exploratory, manipulatory, and curiosity behaviors. *Psychological Review*, 64(2):91.

Djerassi, C. (2011). Foreword. In *Drive and Curiosity: What Fuels the Passion for Science*. Prometheus Books, Amherst, New York. by Istvan Hargittai.

Doya, K. (2010). Open challenges for autonomous cumulative learning. *The Newsletter of the Autonomous Mental Development Technical Committee*, 7(2):5–6.

Dubey, R. and Griffiths, T. L. (2020). Reconciling novelty and complexity through a rational analysis of curiosity. *Psychological Review*, 127(3):455–476.

Duin, A. H. and Pedersen, I. (2021). Working alongside non-human agents. In *2021 IEEE International Professional Communication Conference (ProComm)*, pages 1–5.

Dvorsky, G. (2016). Artificial curiosity allows this bot to triumph at Montezuma's Revenge. Retrieved from https://gizmodo.com/artificial-curiosity-allows-this-bot-to-triumph-at-mont-1781067908. Gizmodo.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. (2021). First return, then explore. *Nature*, 590(7847):580–586.

Engel, S. (2011). Children's need to know: Curiosity in schools. *Harvard Educational Review*, 81(4):625–645.

Engel, S. (2015). *The Hungry Mind: The Origins of Curiosity in Childhood*. Harvard University Press.

Engel, S. and Randall, K. (2009). How teachers respond to children's inquiry. *American Educational Research Journal*, 46(1):183–202.

Fastrich, G. M. and Murayama, K. (2018). Curiosity carry-over effect. OSF Preprint. Retrieved from https://doi.org/10.31219/osf.io/x4npq.

Fedorov, V. V. (1972). *Theory of Optimal Experiments*, volume 12 of *Probability and Mathematical Statistics: A Series of Monographs and Textbooks*. Academic Press, Inc., New York, New York. Translated and edited by W.J. Studden and E. M. Klimko.

Finch, T. (2009). Incremental calculation of weighted mean and variance. resreport, University of Cambridge. Retrieved from https://fanf2.user.srcf.net/hermes/doc/antiforgery/stats.pdf.

Fiske, A. P. (2020). The lexical fallacy in emotion research: Mistaking vernacular words for psychological entities. *Psychological Review*, 127(1):95.

FitzGibbon, L., Lau, J. K. L., and Murayama, K. (2020). The seductive lure of curiosity: information as a motivationally salient reward. *Current Opinion in Behavioral Sciences*, 35:21 – 27. Curiosity (Explore vs Exploit).

Fowler, H. (1965). *Curiosity and exploratory behavior*. Critical Issues in Psyhology. Macmillan, New York.

Fox, L., Dan, O., Elber-Dorozko, L., and Loewenstein, Y. (2020). Exploration: from machines to humans. *Current Opinion in Behavioral Sciences*, 35:104–111. Curiosity (Explore vs Exploit).

Frank, M., Leitner, J., Stollenga, M., Förster, A., and Schmidhuber, J. (2014). Curiosity driven reinforcement learning for motion planning on humanoids. *Frontiers in neurorobotics*, 7:25.

Frege, G. (1951). II.—On concept and object. *Mind*, LX(238):168–180. First published in the Vierteljahrsschrift für wissenschaftliche Philosophie, 16 (1892): 192-205. Translation by P. T. Geach, revised by Max Black.

Future of Life Institute (2017). The Asilomar AI principles. https://futureoflife.org/ai-principles/.

Garivier, A. and Moulines, E. (2011). On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*.

Gino, F. (2018). The business case for curiosity. *The Harvard Business Review*, 96(5):48–57.

Golman, R., Loewenstein, G., Molnar, A., and Saccardo, S. (2021). The demand for, and avoidance of, information. *Management Science*. Published online in Articles in Advance 15 Dec 2021.

Golovin, D. and Krause, A. (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*.

Gordon, G. and Ahissar, E. (2011). Reinforcement active learning hierarchical loops. In *The 2011 International Joint Conference on Neural Networks (IJCNN 2011 - San Jose)*. Institute of Electrical and Electronics Engineers (IEEE).

Gordon, G. and Ahissar, E. (2012). Hierarchical curiosity loops and active sensing. *Neural Networks*, 32:119–129. Selected Papers from IJCNN 2011. Edited by Jean-Philippe Thivierge, Ali Minai, Hava Siegelmann, Cesare Alippi and Michael Georgiopoulos.

Gordon, G., Fonio, E., and Ahissar, E. (2014). Learning and control of exploration primitives. *Journal of computational neuroscience*, 37(2):259–280.

Gottlieb, J., Oudeyer, P.-Y., Lopes, M., and Baranes, A. (2013). Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in cognitive sciences*, 17(11):585–593.

Grace, K. and Maher, M. L. (2015). Specific curiosity as a cause and consequence of transformational creativity. In Toivonen, H., Colton, S., Cook, M., and Ventura, D., editors, *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, pages 260–267, Park City, Utah. Brigham Young University.

Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. (2017). Automated curriculum learning for neural networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1311–1320. PMLR.

Graziano, V., Glasmachers, T., Schaul, T., Pape, L., Cuccu, G., Leitner, J., and Schmidhuber, J. (2011). Artificial curiosity for autonomous space exploration. *Acta Futura*, 4:41–52.

Gregor, M. and Spalek, J. (2014). Curiosity-driven exploration in reinforcement learning. In *2014 ELEKTRO*, pages 435–440.

Gross, M. E., Zedelius, C. M., and Schooler, J. W. (2020). Cultivating an understanding of curiosity as a seed for creativity. *Current Opinion in Behavioral Sciences*, 35:77–82. Curiosity (Explore vs Exploit).

Gruber, M. J. and Ranganath, C. (2019). How curiosity enhances hippocampus-dependent memory: The prediction, appraisal, curiosity, and exploration (PACE) framework. *Trends in Cognitive Sciences*, 23(12):1014 – 1025.

Guckelsberger, C. (2020). *Intrinsic motivation in computational creativity applied to videogames*. PhD thesis, Queen Mary University of London.

Günther, J., Ady, N. M., Kearney, A., Dawson, M. R., and Pilarski, P. M. (2020). Examining the use of Temporal-Difference Incremental Delta-Bar-Delta for real-world predictive knowledge architectures. *Frontiers in Robotics and AI*.

Gupta, N., Granmo, O.-C., and Agrawala, A. (2011). Thompson sampling for dynamic multi-armed bandits. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 1, pages 484–489.

Haber, N., Mrowca, D., Fei-Fei, L., and Yamins, D. L. (2018). Learning to play with intrinsically-motivated self-aware agents. In *Advances in Neural Information Processing Systems*.

Hagtvedt, L. P., Dossinger, K., Harrison, S. H., and Huang, L. (2019). Curiosity made the cat more creative: Specific curiosity as a driver of creativity. *Organizational Behavior and Human Decision Processes*, 150:1–13.

Hamrick, J. B. (2019). Analogues of mental simulation and imagination in deep learning. *Current Opinion in Behavioral Sciences*, 29:8–16.

Harris, P. L. (2012). *Trusting what you're told: How children learn from others*. Harvard University Press.

Hart, S. and Grupen, R. (2013). Intrinsically motivated affordance discovery and modeling. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 279–300. Springer.

Hauser, T. U. (2018). Is human curiosity neurobiologically unique? *IEEE CIS Newsletter on Cognitive and Developmental Systems*, 15(1):6–7.

Herrmann, J. M., Pawelzik, K., and Geisel, T. (2000). Learning predictive representations. *Neurocomputing*, 32–33:785–791.

Hester, T. and Stone, P. (2017). Intrinsically motivated model learning for developing curious robots. *Artificial Intelligence*.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6b: A bag of tricks for mini-batch gradient descent. Retrieved from https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf.

Hoch, S. J. and Loewenstein, G. F. (1991). Time-inconsistent Preferences and Consumer Self-Control. *Journal of Consumer Research*, 17(4):492–507.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. (2016). VIME: Variational information maximizing exploration. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 1109–1117. Curran Associates, Inc.

Hsee, C. K. and Ruan, B. (2016). The Pandora effect: The power and peril of curiosity. *Psychological science*, 27(5):659–666.

Hunt, E. (1971). What kind of computer is man? *Cognitive Psychology*, 2(1):57–98.

IEEE TechXplore (2023). Ask a scientist: How will AI affect creativity? https://techxplore.com/news/2023-04-scientist-ai-affect-creativity.html. Retrieved 24 April 2023.

Inan, I. (2010). Inostensible reference and conceptual curiosity. *Croatian Journal of Philosophy*, 10(28):21–41.

Inan, I. (2012). *The Philosophy of Curiosity*. Routledge.

Isikman, E., MacInnis, D. J., Ülkümen, G., and Cavanaugh, L. A. (2016). The effects of curiosity-evoking events on activity enjoyment. *Journal of Experimental Psychology: Applied*, 22(3):319–330.

Itti, L. and Baldi, P. F. (2005). A principled approach to detecting surprising events in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 631–637, San Siego, CA.

Itti, L. and Baldi, P. F. (2006). Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems 19*, pages 547–554, Cambridge, MA. MIT Press.

Jacobsen, A., Schlegel, M., Linke, C., Degris, T., White, A., and White, M. (2019). Meta-descent for online, continual prediction. In *AAAI Conference on Artificial Intelligence*.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2017). Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4).

Jepma, M., Verdonschot, R., van Steenbergen, H., Rombouts, S., and Nieuwenhuis, S. (2012). Neural mechanisms underlying the induction and relief of perceptual curiosity. *Frontiers in Behavioral Neuroscience*, 6(5).

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Kang, M. J., Hsu, M., Krajbich, I. M., Loewenstein, G., McClure, S. M., Wang, J. T.-y., and Camerer, C. F. (2009). The wick in the candle of learning: Epistemic curiosity activates reward circuitry and enhances memory. *Psychological science*, 20(8):963–973.

Kaplan, F. and Oudeyer, P.-Y. (2003). Motivational principles for visual know-how development. In *Proceedings of the Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems.*, volume 101 of *Lund University Cognitive Studies*, Boston, USA.

Karl, M., Becker-Ehmck, P., Soelch, M., Benbouzid, D., van der Smagt, P., and Bayer, J. (2022). Unsupervised real-time control through variational empowerment. In Asfour, T., Yoshida, E., Park, J., Christensen, H., and Khatib, O., editors, *Robotics Research*, pages 158–173, Cham. Springer International Publishing.

Kashdan, T. B., Gallagher, M. W., Silvia, P. J., Winterstein, B. P., Breen, W. E., Terhar, D., Steger, M. F., Kashdan, T. B., Gallagher, M. W., Silvia, P. J., Winterstein, B. P., Breen, W. E., Terhar, D., and Steger, M. F. (2009). The curiosity and exploration inventory-II: Development, factor structure, and psychometrics. *Journal of Research in Personality*, 43(6):987–998.

Kashdan, T. B., Sherman, R. A., Yarbro, J., and Funder, D. C. (2013). How are curious people viewed and how do they behave in social situations? from the perspectives of self, friends, parents, and unacquainted observers. *Journal of Personality*, 81(2):142–154.

Kawai, M. (1965). Newly-acquired pre-cultural behavior of the natural troop of Japanese monkeys on Koshima Islet. *Primates*, 6(1):1–30.

Kearney, A., Veeriah, V., Travnik, J., Pilarski, P. M., and Sutton, R. S. (2019). Learning feature relevance through step size adaptation in temporal-difference learning. arxiv.org/abx/1903.03252.

Kearney, A., Veeriah, V., Travnik, J. B., Sutton, R. S., and Pilarski, P. M. (2018). TIDBD: Adapting temporal-difference step-sizes through stochastic meta-descent. arxiv.org/abs/1804.03334.

Kidd, C. and Hayden, B. Y. (2015). The psychology and neuroscience of curiosity. *Neuron*, 88(3):449–460.

Kidd, C., Piantadosi, S. T., and Aslin, R. N. (2012). The Goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PLOS ONE*, 7(5):1–8.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: A universal agent-centric measure of control. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 128–135. IEEE.

Knobloch-Westerwick, S. and Keplinger, C. (2006). Mystery appeal: Effects of uncertainty and resolution on the enjoyment of mystery. *Media Psychology*, 8(3):193–212.

Konečni, V. J. (1978). Daniel E. Berlyne: 1924-1976. *The American Journal of Psychology*, pages 133–137.

Konyushkova, K., Sznitman, R., and Fua, P. (2017). Learning active learning from data. In *Advances in Neural Information Processing Systems*.

Koop, A. (2008). Investigating experience: temporal coherence and empirical knowledge representation. mathesis, University of Alberta.

Krakovna, V., Uesato, J., Mikulik, V., Rahtz, M., Everitt, T., Kumar, R., Kenton, Z., Leike, J., and Legg, S. (2020). Specification gaming: the flip side of AI ingenuity. Blog Post. Retrieved from https://www.deepmind.com/blog/specification-gaming-the-flip-side-of-ai-ingenuity.

Krapp, A. (1994). Interest and curiosity. The role of interest in a theory of exploratory action. In Keller, H., Schneider, K., and Henderson, B., editors, *Curiosity and Exploration*, pages 79–100. Springer, Berlin, Heidelberg.

Kubovy, M. (1999). On the pleasures of the mind. In Kahneman, D., Diener, E., and Schwarz, N., editors, *Well-being: The Foundations of Hedonic Psychology*, chapter 7, pages 134–154. Russell Sage Foundation.

Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683.

Kurth-Nelson, Z. and Redish, A. D. (2009). Temporal-difference reinforcement learning with distributed representations. *PLOS ONE*, 4(10):1–19.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:1–15, e253.

Lattimore, T. and Szepesvári, C. (2019). Cleaning up the neighborhood: A full classification for adversarial partial monitoring. In *International Conference on Algorithmic Learning Theory*.

Lattimore, T. and Szepesvári, C. (2020). *Bandit Algorithms*. Retrieved from https://tor-lattimore.com/downloads/book/book.pdf on June 2, 2022.

Lieto, A. and Radicioni, D. P. (2016). From human to artificial cognition and back: New perspectives on cognitively inspired AI systems. *Cognitive Systems Research*, 39:1–3. From human to artificial cognition (and back): new perspectives of cognitively inspired AI systems.

Ligneul, R., Mermillod, M., and Morisseau, T. (2018). From relief to surprise: Dual control of epistemic curiosity in the human brain. *NeuroImage*, 181:490–500.

Lim, S. H. and Auer, P. (2012). Autonomous exploration for navigating in mdps. In *JMLR: Workshop and Conference Proceedings*, volume 23, pages 40.1–40.24.

Linke, C., Ady, N. M., White, M., Degris, T., and White, A. (2019). Investigating curiosity for multi-prediction learning. In *4th Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, McGill University, Montréal, Québec, Canada.

Linke, C., Ady, N. M., White, M., Degris, T., and White, A. (2020). Adapting behavior via intrinsic reward: A survey and empirical study. *Journal of Artificial Intelligence Research*, 69:1287–1332.

Linke, C., Ady, N. M., White, M., Degris, T., and White, A. (2021). Adapting behavior via intrinsic reward: A survey and empirical study. In *30th International Joint Conference on Artificial Intelligence (IJCAI-21)*, Montréal (Virtual), Québec, Canada.

Litman, J., Hutchins, T., and Russon, R. (2005). Epistemic curiosity, feeling-of-knowing, and exploratory behaviour. *Cognition and Emotion*, 19(4):559–582.

Litman, J. A. and Spielberger, C. D. (2003). Measuring epistemic curiosity and its diversive and specific components. *Journal of personality assessment*, 80(1):75–86.

Little, D. Y.-J. and Sommer, F. T. (2013). Learning and exploration in action-perception loops. *Frontiers in Neural Circuits*, 7:37.

Liu, Y.-E., Mandel, T., Brunskill, E., and Popovic, Z. (2014). Trading off scientific knowledge and user learning with multi-armed bandits. In *International Conference on Educational Data Mining*.

Lloyd, K. and Dayan, P. (2018). Interrupting behaviour: Minimizing decision costs via temporal commitment and low-level interrupts. *PLOS Computational Biology*, 14(1):1–23.

Loewenstein, G. (1994). The psychology of curiosity: A review and reinterpretation. *Psychological Bulletin*, 116(1):75–98.

Loewenstein, G., Adler, D., Behrens, D., and Gillis, J. (1992). Why Pandora opened the box: Curiosity as a desire for missing information. Unpublished manuscript, Department of Social and Decision Sciences, Carnegie Mellon University, Pittsburgh, PA.

Lopes, M., Lang, T., Toussaint, M., and Oudeyer, P.-Y. (2012). Exploration in model-based reinforcement learning by empirically estimating learning progress. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 206–214. Curran Associates, Inc.

Machado, M. C. (2019). *Efficient Exploration in Reinforcement Learning through Time-Based Representations*. PhD thesis, University of Alberta, Edmonton, Alberta, Canada.

Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M., and Bowling, M. (2018). Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562.

MacKay, D. J. C. (2003). *Information theory, Inference and Learning Algorithms*. Cambridge University Press, fourth edition.

MacPherson, T., Churchland, A., Sejnowski, T., DiCarlo, J., Kamitani, Y., Takahashi, H., and Hikida, T. (2021). Natural and artificial intelligence: A brief introduction to the interplay between AI and neuroscience research. *Neural Networks*, 144:603–613.

Maei, H. R., Szepesvári, C., Bhatnagar, S., and Sutton, R. S. (2010). Toward off-policy learning control with function approximation. In *International Conference on Machine Learning*.

Mahmood, A. R., Sutton, R. S., Degris, T., and Pilarski, P. M. (2012). Tuning-free step-size adaptation. In *International Conference on Acoustics, Speech and Signal Processing*.

Malone, T. W. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive Science*, 5(4):333 – 369.

Markey, A. and Loewenstein, G. (2014). Curiosity. In Reinhard Pekrun, L. L.-G., editor, *International Handbook of Emotions in Education*, pages 228–245. Routledge, New York.

Martin, J., S., S. N., Everitt, T., and Hutter, M. (2017). Count-based exploration in feature space for reinforcement learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2471–2478.

Marvin, C. B. and Shohamy, D. (2016). Curiosity and reward: Valence predicts choice and information prediction errors enhance learning. *Journal of Experimental Psychology: General*, 145(3):266.

Matusch, B., Ba, J., and Hafner, D. (2020). Evaluating agents without rewards. In *Workshop on Biological and Artificial Reinforcement Learning (BARL 2020) at NeurIPS 2020*.

Meuleau, N. and Bourgine, P. (1999). Exploration of multi-state environments: Local measures and back-propagation of uncertainty. *Machine Learning*.

Mirolli, M. and Baldassarre, G. (2013). Functions and mechanisms of intrinsic motivations. In *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA. PMLR.

Modayil, J., White, A., and Sutton, R. S. (2014). Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior*, 22(2):146–160.

Moore, J. and Newell, A. (1974). How can Merlin understand? In Gregg, L. W., editor, *Knowledge and Cognition*, 9th Symposium on Cognition, Carnegie-Mellon University. Lawrence Erlbaum. Retrieved frome https://kilthub.cmu.edu/articles/journal_contribution/How_can_Merlin_understand_/6606152/1/files/12096647.pdf.

Morris, L. S., Grehl, M. M., Rutter, S. B., Mehta, M., and Westwater, M. L. (2022). On what motivates us: a detailed review of intrinsic v. extrinsic motivation. *Psychological Medicine*, 52(10):1801—-1816.

Moulin-Frier, C. and Oudeyer, P.-Y. (2012). Curiosity-driven phonetic learning. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, pages 1–8. IEEE.

Murayama, K., FitzGibbon, L., and Sakaki, M. (2019). Process account of curiosity and interest: A reward-learning perspective. *Educational Psychology Review*, 31(4):875–895.

Murayama, K. and Kuhbandner, C. (2011). Money enhances memory consolidation – but only for boring material. *Cognition*, 119(1):120–124.

Murayama, K., Matsumoto, M., Izuma, K., and Matsumoto, K. (2010). Neural basis of the undermining effect of monetary reward on intrinsic motivation. *Proceedings of the National Academy of Sciences*, 107(49):20911–20916.

Newell, A. (1970). Remarks on the relationship between artificial intelligence and cognitive psychology. In Banerji, R. B. and Mesarovic, M. D., editors, *Theoretical Approaches to Non-Numerical Problem Solving*, pages 363–400, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ngo, H., Luciw, M., Forster, A., and Schmidhuber, J. (2012). Learning skills from play: artificial curiosity on a katana robot arm. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE.

Ngo, H., Luciw, M., Förster, A., and Schmidhuber, J. (2013). Confidence-based progress-driven self-generated goals for skill acquisition in developmental robots. *Frontiers in Psychology*, 4:833.

Nicki, R. M. (1970). The reinforcing effect of uncertainty reduction on a human operant. *Canadian Journal of Psychology*, 24(6):389–400.

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154.

NPD Group (2022). The number of Americans and Canadians that play mobile games fell by 4% in 2021. https://www.npd.com/news/press-releases/2022/the-number-of-americans-and-canadians-that-play-mobile-games-fell-by-4-in-2021/.

Oddi, A., Rasconi, R., Santucci, V. G., Sartor, G., Cartoni, E., Mannella, F., and Baldassarre, G. (2020). Integrating open-ended learning in the sense-plan-act robot control paradigm. In *24th European Conference on Artificial Intelligence: ECAI 2020*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2417–2424. IOS Press.

Orseau, L. (2014). Universal knowledge-seeking agents. *Theoretical Computer Science*, 519:127–139.

Ortner, P. and Auer, R. (2007). Logarithmic online regret bounds for undiscounted reinforcement learning. *Advances in Neural Information Processing Systems*, 19:49.

Osteyee, D. B. and Good, I. J. (1974). *Information, Weight of Evidence, the Singularity between Probability Measures and Signal Detection*. Springer, Berlin, Heidelberg.

Oudeyer, P.-Y. (2010). Developmental constraints on intrinsically motivated exploration. *The Newsletter of the Autonomous Mental Development Technical Committee*, 7(2):7–8.

Oudeyer, P.-Y. and Kaplan, F. (2004). Intelligent adaptive curiosity: a source of self-development. In Berthouze, L., Kozima, H., Prince, C. G., Sandini, G., Stojanov, G., Metta, G., and Balkenius, C., editors, *Proceedings of the Fourth International Workshop on Epigenetic Robotics*, number 117 in Lund University Cognitive Studies, Genoa, Italy.

Oudeyer, P.-Y. and Kaplan, F. (2007). What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, 1. Article 6.

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2778–2787, International Convention Centre, Sydney, Australia. PMLR.

Pathak, D., Gandhi, D., and Gupta, A. (2019). Self-supervised exploration via disagreement. In *International Conference on Machine Learning*.

Patten, T., Martens, W., and Fitch, R. (2018). Monte Carlo planning for active object classification. *Autonomous Robots*.

Pekrun, R. (2019). The murky distinction between curiosity and interest: State of the art and future prospects. *Educational Psychology Review*, 31(4):905–914.

Péré, A., Forestier, S., Sigaud, O., and Oudeyer, P.-Y. (2018). Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations*.

Peterson, E. G. and Cohen, J. (2019). A case for domain-specific curiosity in mathematics. *Educational Psychology Review*, 31(4):807–832.

Peterson, E. G. and Hidi, S. (2019). Curiosity and interest: current perspectives. *Educational Psychology Review*, 31(4):781–788.

Pilarski, P. M., Dawson, M. R., Degris, T., Carey, J. P., Chan, K. M., Hebert, J. S., and Sutton, R. S. (2013a). Adaptive artificial limbs: A real-time approach to prediction and anticipation. *IEEE Robotics & Automation Magazine*, 20(1):53–64.

Pilarski, P. M., Dick, T. B., and Sutton, R. S. (2013b). Real-time prediction learning for the simultaneous actuation of multiple prosthetic joints. In *International Conference on Rehabilitation Robotics (ICORR 2013)*, pages 1–8. IEEE.

Polman, E., Ruttan, R., and Peck, J. (2017). Curiosity and want/should conflicts. In Gneezy, A., Griskevicius, V., and Williams, P., editors, *NA - Advances in Consumer Research*, volume 45, pages 818–821, Duluth, MN. Association for Consumer Research.

Ramadan, Z., F. Farah, M., and El Essrawi, L. (2021). From Amazon.com to Amazon.love: How Alexa is redefining companionship and interdependence for people with special needs. *Psychology & Marketing*, 38(4):596–609.

Rao, V. V. (2009). *Principles of Communication*. Lecture Notes retrieved from http://nptel.ac.in/courses/117106090/ on March 15, 2016.

Reio, Jr, T. G., Petrosko, J. M., Wiswell, A. K., and Thongsukmag, J. (2006). The measurement and conceptualization of curiosity. *The Journal of Genetic Psychology*, 167(2):117–135.

Renninger, K. A. (2000). Individual interest and its implications for understanding intrinsic motivation. In Sansone, C. and Harackiewicz, J. M., editors, *Intrinsic and Extrinsic Motivation*, Educational Psychology, chapter 13, pages 373–404. Academic Press, San Diego.

Renninger, K. A. and Hidi, S. (2016). *The Power of Interest for Motivation and Engagement*. Routledge.

Repko, A. F. and Szostak, R. (2020). *Interdisciplinary Research: Process and Theory*. SAGE Publications, Inc.

Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., van de Wiele, T., Mnih, V., Heess, N., and Springenberg, J. T. (2018). Learning by playing – Solving sparse reward tasks from scratch. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4344–4353. PMLR.

Rotgans, J. I. and Schmidt, H. G. (2017). The relation between individual interest and knowledge acquisition. *British Educational Research Journal*, 43(2):350–371.

Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In Carla E. Brodley, A. P. D., editor, *Machine Learning: Proceedings of the Eighteenth International Conference (ICML 2001)*, San Francisco, CA, USA. Morgan Kaufmann.

Ruan, B., Hsee, C. K., and Lu, Z. Y. (2018). The teasing effect: An underappreciated benefit of creating and resolving an uncertainty. *Journal of Marketing Research*.

Rummery, G. A. and Niranjan, M. (1994). On-line Q-learning using connectionist systems. techreport CUED/F-INFENG/TR 166, Cambridge University.

Ryan, R. M. and Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54–67.

Santucci, V. G., Baldassarre, G., and Mirolli, M. (2012). Intrinsic motivation mechanisms for competence acquisition. In *IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–6. IEEE.

Santucci, V. G., Baldassarre, G., and Mirolli, M. (2013a). Intrinsic motivation signals for driving the acquisition of multiple tasks: a simulated robotic study. In *Proceedings of the 12th International Conference on Cognitive Modelling (ICCM)*, pages 1–6.

Santucci, V. G., Baldassarre, G., and Mirolli, M. (2013b). Which is the best intrinsic motivation signal for learning multiple skills? *Frontiers in Neurorobotics*, 7:22.

Satsangi, Y., Lim, S., Whiteson, S., Oliehoek, F., and White, M. (2020). Maximizing information gain in partially observable environments via prediction rewards. In *International Conference on Autonomous Agents and Multiagent Systems*.

Satsangi, Y., Whiteson, S., Oliehoek, F. A., and Spaan, M. T. (2018). Exploiting submodular value functions for scaling up active perception. *Autonomous Robots*.

Schaul, T., Sun, Y., Wierstra, D., Gomez, F., and Schmidhuber, J. (2011). Curiosity-driven optimization. In *2011 IEEE Congress on Evolutionary Computation (CEC)*, pages 1343–1349. IEEE.

Schembri, M., Mirolli, M., and Baldassarre, G. (2007a). Evolution and learning in an intrinsically motivated reinforcement learning robot. In Almeida e Costa, F., Rocha, L. M., Costa, E., Harvey, I., and Coutinho, A., editors, *Advances in Artificial Life: 9th European Conference, ECAL 2007, Lisbon, Portugal*, volume 4648 of *Lecture Notes in Computer Science*, pages 294–303, Berlin, Heidelberg. Springer Berlin Heidelberg.

Schembri, M., Mirolli, M., and Baldassarre, G. (2007b). Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot. In *IEEE 6th International Conference on Development and Learning (ICDL 2007)*, pages 282–287. IEEE.

Schmidhuber, J. (1991a). Curious model-building control systems. In *IEEE International Joint Conference on Neural Networks*, pages 1458–1463. IEEE.

Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In Meyer, J. A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227. MIT Press/Bradford Books.

Schmidhuber, J. (2008). Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In *Workshop on Anticipatory Behavior in Adaptive Learning Systems*.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247.

Schmitt, F. F. and Lahroodi, R. (2008). The epistemic value of curiosity. *Educational Theory*, 58(2):125–148.

Schossau, J., Adami, C., and Hintze, A. (2016). Information-theoretic neurocorrelates boost evolution of cognitive systems. *Entropy*, 18(1).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. arxiv.org/abs/1707.06347.

Settles, B. (2012). *Active Learning*. Springer International Publishing, Cham.

Shankar, A. (2020). "the campus is sick": Capitalist curiosity and student mental health. In Zurn, P. and Shankar, A., editors, *Curiosity Studies: A New Ecology of Knowledge*, pages 106–125. University of Minnesota Press, Minneapolis.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423.

Shannon, C. E. (1963). The mathematical theory of communication. In Shannon, C. E. and Weaver, W., editors, *The Mathematical Theory of Communication*, pages 3–93. University of Illinois Press, Urbana.

Shin, D. D. and Kim, S.-i. (2019). Homo curious: Curious or interested? *Educational Psychology Review*, 31(4):853–874.

Shulman, L. S. (1999). Professing educational scholarship. In Lagemann, E. C. and Shulman, L. S., editors, *Issues in education research: Problems and possibilities*, pages 159–165. Jossey-Bass, San Francisco.

Shyam, P., Jaśkowski, W., and Gomez, F. (2019). Model-based active exploration. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5779–5788. PMLR.

Silver, D., van Hasselt, H., Hessel, M., Schaul, T., Guez, A., Harley, T., Dulac-Arnold, G., Reichert, D., Rabinowitz, N., Barreto, A., and Degris, T. (2017). The Predictron: End-to-end learning and planning. In *International Conference on Machine Learning*.

Silvia, P. J. (2006). *Exploring the Psychology of Interest*. Oxford University Press, New York.

Simon, H. A. (1967). Motivational and emotional controls of cognition. *Psychological Review*, 74(1):29.

Simpson, M. J. A. (1976). The study of animal play. In Bateson, P. P. G. and Hinde, R. A., editors, *Growing points in ethology*, pages 385–400. Cambridge University Press.

Şimşek, Ö. and Barto, A. G. (2006). An intrinsic reward mechanism for efficient exploration. In *Proceedings of the 23rd international conference on Machine learning*, pages 833–840. ACM.

Singh, S., Lewis, R. L., Barto, A. G., and Sorg, J. (2010). Intrinsically motivated reinforcement learning: An evolutionary perspective. *Autonomous Mental Development, IEEE Transactions on*, 2(2):70–82.

Singh, S. P., Barto, A. G., and Chentanez, N. (2004). Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288.

Skinner, B. F. (1963). Operant behavior. *American psychologist*, 18(8):503–515.

Spielberger, C. D. and Reheiser, E. C. (2009). Assessment of emotions: Anxiety, anger, depression, and curiosity. *Applied Psychology: Health and Well-Being*, 1(3):271–302.

Stadie, B. C., Levine, S., and Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. arxiv.org/abs/1507.00814.

Still, S. and Precup, D. (2012). An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131(3):139–148.

Stojanov, G. and Kulakov, A. (2006). On curiosity in intelligent robotic systems. In *Proceedings of the AAAI Fall Symposium on Interaction and Emergent Phenomena in Societies of Agents*, pages 44–51.

Stolle, M. and Precup, D. (2002). Learning options in reinforcement learning. In Koenig, S. and Holte, R. C., editors, *SARA 2002: Abstraction, Reformulation, and Approximation*, volume 2371 of *Lecture Notes in Computer Science*, pages 212–223, Berlin, Heidelberg. Springer Berlin Heidelberg.

Stout, A. and Barto, A. G. (2010). Competence progress intrinsic motivation. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pages 257–262. IEEE.

Strehl, A. L. and Littman, M. L. (2005). A theoretical analysis of Model-based Interval Estimation. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 856–863.

Strehl, A. L. and Littman, M. L. (2008). An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331. Learning Theory 2005.

Sutton, R. S. (1990a). First results with Dyna, an integrated architecture for learning, planning and reacting. In *Proceedings of the AAAI Spring Symposium*, pages 179–189. Retreived from http://www.incompleteideas.net/papers/first_results_with_dyna.pdf.

Sutton, R. S. (1990b). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 216–224.

Sutton, R. S. (1992). Adapting bias by gradient descent: An incremental version of delta-bar-delta. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 171–176. MIT Press.

Sutton, R. S. (1995). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In Touretzky, D., Mozer, M., and Hasselmo, M., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 1038–1044. MIT Press.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* MIT Press Cambridge, MA.

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

Sutton, R. S. and Barto, A. G. (2020). *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, second edition. Retrieved from http://incompleteideas.net/book/RLbook2020.pdf.

Sutton, R. S., Koop, A., and Silver, D. (2007). On the role of tracking in stationary environments. In *International Conference on Machine Learning*.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. (2011). Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181 – 211.

Szita, I. and Lorincz, A. (2008). The many faces of optimism: A unifying approach. In *International Conference on Machine learning*.

Szumowska, E. and Kruglanski, A. W. (2020). Curiosity as end and means. *Current Opinion in Behavioral Sciences*, 35:35–39. Curiosity (Explore vs Exploit).

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. (2017). #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*.

Webster's New World College Dictionary (2014). Houghton Mifflin Harcourt Publishing Company, fifth edition.

Thrun, S. B. (1992). The role of exploration in learning control. In White, D. A. and Sofge, D. A., editors, *Handbook of Intelligent Control*, chapter 14, pages 527–559. Van Nostrand Reinhold, New York, NY.

Tishby, N. and Polani, D. (2011). Information theory of decisions and actions. In Cutsuridis, V., Hussain, A., and Taylor, J. G., editors, *Perception-Action Cycle*, pages 601–636. Springer.

Travnik, J. B. and Pilarski, P. M. (2017). Representing high-dimensional data to intelligent prostheses and other wearable assistive robots: A first comparison of tile coding and selective Kanerva coding. In *International Conference on Rehabilitation Robotics (ICORR 2017)*, pages 1443–1450.

van Seijen, H., Fatemi, M., Romoff, J., Laroche, R., Barnes, T., and Tsang, J. (2017). Hybrid Reward Architecture for reinforcement learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ventura, J., Ady, N. M., and Pilarski, P. M. (2017). An exploration of machine curiosity and reinforcement learning using a simple robot. https://doi.org/10.7939/R36W96Q00. WISEST Summer Research Program Posters, University of Alberta Education & Research Archive.

Vigorito, C. M. (2016). *Intrinsically Motivated Exploration in Hierarchical Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst.

Wade, S. and Kidd, C. (2019). The role of prior knowledge and curiosity in learning. *Psychonomic Bulletin & Review*, 26(4):1377–1387.

Walker, E. L. (1978). Introduction, *Curiosity and Learning* by Daniel E. Berlyne. *Motivation and Emotion*, 2(2):98.

Wasserman, L. A. (2005). *All of Statistics: A Concise Course in Statistical Inference*. Springer. Corrected second printing.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, United Kingdom.

Webster, J., Trevino, L. K., and Ryan, L. (1993). The dimensionality and correlates of flow in human-computer interactions. *Computers in Human Behavior*, 9(4):411–426.

Whitcomb, D. (2010). Curiosity was framed. *Philosophy and Phenomenological Research*, 81(3):664–687.

White, A. (2015). *Developing a predictive approach to knowledge*. PhD thesis, University of Alberta.

White, A., Modayil, J., and Sutton, R. S. (2012). Scaling life-long off-policy learning. In *International Conference on Development and Learning and Epigenetic Robotics*.

White, A., Modayil, J., and Sutton, R. S. (2014). Surprise and curiosity for big data robotics. In *AAAI-14 Workshop on Sequential Decision-Making with Big Data, Quebec City, Quebec, Canada*.

White, R. W. (1959). Motivation reconsidered: The concept of competence. *Psychological Review*, 66(5):297–333.

Wiggin, K. L., Reimann, M., and Jain, S. P. (2019). Curiosity Tempts Indulgence. *Journal of Consumer Research*, 45(6):1194–1212.

Wu, Q. and Miao, C. (2013). Curiosity: From psychology to computation. *ACM Computing Surveys (CSUR)*, 46(2):18.

Wundt, W. (1874). *Grundzüge der physiologischen Psychologie*. Engelmann.

Yampolskiy, R. V. (2014). Utility function security in artificially intelligent agents. *Journal of Experimental & Theoretical Artificial Intelligence*, 26(3):373–389.

Yasui, N. W. (2020). An empirical study of exploration strategies for model-free reinforcement learning. mathesis, University of Alberta.

Yong, E. (2014). Scientists instil new cultural traditions in wild tits. *National Geographic*. Retrieved from https://www.nationalgeographic.com/science/article/scientists-instil-new-cultural-traditions-in-wild-tits.

Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. arxiv.org/abs/1212.5701.

Zurn, P. (2015). *Curiosity: philosophy and the politics of difference*. College of liberal arts & social sciences theses and dissertations, 183, DePaul University.

Zurn, P. (2019). Busybody, hunter, dancer: Three historical models of curiosity. In Papastephanou, M., editor, *Toward New Philosophical Explorations of the Epistemic Desire to Know: Just Curious About Curiosity*, pages 26–49. Cambridge Scholars Publishing.

Zurn, P. (2020). Curiosity and political resistance. In Zurn, P. and Shankar, A., editors, *Curiosity Studies: A New Ecology of Knowledge*, pages 227–245. University of Minnesota Press, Minneapolis.

Zurn, P. and Shankar, A. (2020). Introduction: What is curiosity studies? In Zurn, P. and Shankar, A., editors, *Curiosity Studies: A New Ecology of Knowledge*, pages xi–xxx. University of Minnesota Press, Minneapolis.

Zurn, P., Zhou, D., Lydon-Staley, D. M., and Bassett, D. S. (2022). Edgework: Viewing curiosity as fundamentally relational. In Cogliati Dezza, I., Schulz, E., and Wu, C. M., editors, *The Drive for Knowledge: The Science of Human Information Seeking*, pages 259–278. Cambridge University Press.

Zuss, M. (2011). *The practice of theoretical curiosity*, volume 20 of *Explorations of Educational Purpose*. Springer Science & Business Media.

Appendix A

Thesis-Adjacent Contributions

In this section, each of my adjacent contributions is listed with its bibliographic details, followed by a summary of its contents and context.

> Günther, J., Ady, N. M., Kearney, A., Dawson, M. R., and Pilarski, P. M. (2020). Examining the use of Temporal-Difference Incremental Delta-Bar-Delta for real-world predictive knowledge architectures. *Frontiers in Robotics and AI*

With Günther et al. (2020), I performed an empirical investigation of Temporal-Difference Incremental Delta-Bar-Delta (TIDBD), a meta-learning method appropriate for general value function (GVF) learning algorithms. Both GVFs and meta-learning will be described in Chapter 2, along with some of their connections to computational curiosity. This investigation demonstrated that TIDBD is a practical alternative for classic temporal-difference (TD) learning, even on a complex, sensor-rich system like a state of the art prosthetic limb. In particular, TIDBD eliminates the need for an extensive parameter search for appropriate step sizes.

> Ady, N. M. (2017b). Curious actor-critic reinforcement learning with the dynamixel-bot. `https://doi.org/10.7939/R3B853Z7S`. Depart-

ment of Computing Science, University of Alberta Education & Research Archive

In 2017b, I adapted and implemented Gordon and Ahissar's (2012) model of hierarchical curiosity loops for a simple robot. I explored an alternative formulation of their architecture, replacing a neural-network-based predictive learner with a TD(λ) GVF learner—in both Gordon and Ahissar's case and mine, the purpose of the 'learner' was to make predictions about the future state of the environment. In the hierarchical curiosity loops model, the control system is an example of a prediction-error-based intrinsically-motivated reinforcement learning (IMRL) system, a category of approaches to machine curiosity that will be described in more detail in Section 2.3.3. My adaptation offered a demonstration of how Gordon and Ahissar's original ideas could be adapted into a lightweight architecture using GVFs and selective Kanerva coding.[1]

Ventura, J., Ady, N. M., and Pilarski, P. M. (2017). An exploration of machine curiosity and reinforcement learning using a simple robot. https://doi.org/10.7939/R36W96Q00. WISEST Summer Research Program Posters, University of Alberta Education & Research Archive

In 2017, I supervised and structured a project, led by high school research intern Justine Ventura, which centred on the implementation of Information Gain Motivation (IGM) on a simple robot. IGM was named and defined by Oudeyer and Kaplan (2007, p. 6) with reference to earlier work by Fedorov (1972) and Roy and McCallum (2001). IGM is an example of an information-based IMRL system, another category of approaches to machine curiosity explored in more detail in Section 2.3.4. While a number of observations were recorded, one of the most interesting was a limitation of a naive implementation of IGM: the learner tended

[1] Selective Kanerva coding is a representation method that supports particularly computationally efficient representation of high-dimensional sensor observations (Travnik and Pilarski, 2017, p. 1443).

to learn to stay put in states where they were not 'gaining' information, but instead simply avoiding 'losing' it—leading to a 'stuck' behaviour. This 'stuck' behaviour is comparable to that observed in some challenging benchmark domains like Pitfall and Tennis in the Arcade Learning Environment (Machado et al., 2018, p. 543), or that observed in domains with misleading rewards, like Antishaping (p. 15) in Yasui's (2020) exploration suite. In all of these cases, many learning algorithms choose not to move so as to avoid negative rewards.

> Ady, N. M. and Rice, F. (2023). Interdisciplinary methods in computational creativity: How human variables shape human-inspired AI research. In *Proceedings of the 14th International Conference on Computational Creativity (ICCC'23)*

In our June 2023, I co-presented preliminary results from a new collaboration: *Humanness in artificial intelligence (AI)*. Personally, a foundational component of my own work is translating curiosity, a concept from human psychology, into machine algorithms. Similar translation processes are undertaken by numerous researchers around the globe for a variety of psychological concepts, including not only curiosity, but creativity, forgetting, depression, emotion, imagination, and more. Some authors, like Lake et al. (2017, p. 3), have articulated why they value drawing inspiration from natural intelligence, while others, like Newell (1970, p. 363), have reflected on the relationship between AI and psychology, discussing what the relationship could and should be. However, the decisions and thought processes made by humans while designing new human-inspired artificial intelligence systems have been little investigated. Indeed, coalitions like the Future of Life Institute (2017) have called for greater attention to design decision-making in AI. Therefore, there is a need to understand how AI researchers and technologists approach understanding psychological literature, their decisions in translating it, and then who their decisions influence (e.g., how their work influences the scholarly understanding of these concepts in their original context—in humans).

In our June 2023 paper, we presented the initial thematic analysis of our preliminary interview data with researchers and technologists performing this translational work at the interface of natural and artificial intelligence. We articulated some emergent methodological practices: common themes in AI researchers' processes (e.g., strategies for engaging with literature from other disciplines, strategies for rendering psychological theory into a software program), which may offer inspiration and novel approaches for readers, both in human and machine intelligence. Our presentation at ICCC'23 represents the initial findings from a long-term project engaging grounded theory to offer an important contribution at the interface of natural and artificial intelligence, providing researchers with a birds-eye view of the ethical and social implications of such work and a map of potential approaches.

Appendix B

Parameter Screening for Curious Reinforcement Learning Motivated by Unexpected Error

Humans seem to maintain knowledge of their environment and utilize this knowledge to make decisions. One perspective on curiosity is that it should motivate behaviour that helps an agent improve its environment knowledge. In computational systems, one method of maintaining environment knowledge is through estimates or *predictions* of what the system expects to observe in the near future (Modayil et al., 2014).

In 2014, White et al. suggested a measure of 'unexpected error' for the purpose of generating curious behaviour in a robot. Intuitively, exploring situations that the agent can already predict well (leading to low error) will not improve its environment knowledge. Neither will exploring situations that have already been explored, but have such high variance that we cannot expect predictions to improve. Motivating a system to maximize cumulative unexpected error observed over time should ideally benefit the system by leading to improved knowledge of its environment.

Reinforcement learning (RL) is a well-studied way for biological systems and machines to learn about the value of situations and choices through trial and error and then utilize those learned values to make decisions (Sutton and Barto, 2018). Given a *reward signal* provided to the system, there are standard RL algorithms to learn to predict and/or maximize cumulative reward over time. Using RL algorithms to predict observations, White's 'unexpected error' as the intrinsic reward signal, and another RL algorithm to choose actions, we can create a 'curious system.'

However, RL algorithms use multiple parameters, and it is unknown how varying those parameters changes the behaviour of the curious system. While there is no clear measure of the 'curiousness' of observed behaviour, we may simply hope to recognize when it is different. One partial measure of the behaviour in a finite-length run of the system is the cumulative sum of the observation signal, called *return*, G, which may call for maximization or minimization.

$$G = \sum_{t=1}^{\infty} r_t^{(p)} \tag{B.1}$$

where $r_t^{(p)}$ is the observation or *reward* received from the environment.

The agent in our design is modifying its behaviour to maximize its cumulative unexpected error, but at this stage of the study we have little interest in the total accrued; varying its parameters may impact the magnitude of error observed, and therefore it is unreasonable to compare accrued unexpected error for different parameters.

The objective of this study is to determine which parameters, and which interactions of parameters, impact the return.

B.1 Materials and Methods

B.1.1 Hardware and Software

All experiments were implemented in Python without parallelism and performed on a Lenovo Flex 3 laptop with four Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz processors running Ubuntu 16.04.1 LTS.

B.1.2 Experimental Setup

The experimental setup can be considered in three parts: domain (environment), curious system, and test. We describe each in detail in the following sections, and the Python code is included in Appendix B.7. We typically refer to an RL system as an 'agent', referencing the *agency* exhibited by making choices. The curious system will hereafter be called the agent.

Test Design Each test (also called a run) was composed of an initialization and 20,000 iterative discrete time steps. To initialize the test, the domain's initial state signal is recorded, and the agent takes its initial action. In each time step, the domain reward and new state are observed. The agent takes another action based on the observed state, then updates. The interaction between the agent and the domain in each time step is depicted in Figure 3.2. Our implementation of a test can be found in Appendix B.7 as `Test.run()`.

Within each test, we also maintain a count of how many times each action was taken and a running sum of the domain reward, and use both to compute our **response variables**.

Domain Design We devised the curiosity bandit, depicted in Figure 3.2, to showcase the behaviour elicited by variations in *domain-delivered reward*. The curiosity bandit has a single state ($\mathcal{S} = \{s_0\}$), but provides the learning agent with three actions ($\mathcal{A} = \{a_1, a_2, a_3\}$), each of which results in a different *domain reward* signal.

If the agent takes a_1, its reward is drawn uniformly randomly from $[-1, 1]$. If the agent takes a_2, it always receives a reward of 0. If the agent takes a_3, then it receives a reward of $\sin(c_1 \cdot t)$, where c_1 is a small constant, **held-constant** at $c_1 = 0.001$ in our experiment, and t is the current time step (starting at $t = 0$).

The algorithm followed to determine the output state signal and domain reward is **controlled** throughout the experiment. The pseudo-random generation of the reward received after taking action a_1 is **allowed to vary**, because it theoretically simulates noise in the reward signal.

Agent Design The agent has two components: a prediction learner and a control learner. In this section, we describe the algorithm followed and introduce the parameters varied as design factors.

For the set of all available actions $\mathcal{A}$ and the set of all observable state signals $\mathbb{S}$, the prediction learner uses the TD(λ) algorithm (Sutton and Barto, 2018, p. 174) to estimate the *value* function $Q^\pi : \mathbb{S} \times \mathcal{A} \to \mathbb{R}$, where, for each state $s \in \mathbb{S}$ and each action $a \in \mathcal{A}$, the value of a, given it is taken from s, is defined by

$$Q^\pi(s,a) = \mathbb{E}\left\{ \sum_{k=0}^{\infty} \gamma_p^k r_{t+k+1}^{(p)} \middle| s_t = s, a_t = a \right\} \tag{B.2}$$

where $\mathbb{E}$ denotes the expected value, s_t, a_t, and $r_{t+1}^{(p)}$ are, respectively, the state observed and the action taken at time step t, and the resulting domain reward, and $0 \leq \gamma_p \leq 1$ is a constant parameter often called the *discount rate*.

For the TD(λ) algorithm, the agent starts with some initial estimation Q of Q^π. The initial Q is chosen by the agent designer and while it does typically affect behaviour, the effects are short-term and tend to be small in domains where $\mathbb{S} \times \mathcal{A}$ is small. Therefore, in our experiment it is a **held-constant** factor initialized with $Q(s,a) \leftarrow 0$ for all $s, a \in \mathbb{S} \times \mathcal{A}$.

The algorithm also makes use of an *eligibility trace*, $e : \mathbb{S} \times \mathcal{A} \to \mathbb{R}$. It is always initialized with $e(s,a) \leftarrow 0$ for all $s, a \in \mathbb{S} \times \mathcal{A}$.

At each time step, the agent observes some state s and takes some action a. At the beginning of the next time step, the agent observes a domain reward $r^{(p)}$ and a new state s' and takes a next action a'. The agent can then use the sum of the observed domain reward and $Q^\pi(s', a')$ to update $Q(s,a)$.

To update, the prediction learner uses two components: the eligibility trace, e, and the *temporal difference error* (TD error), δ. Eligibility is assigned to the action taken, via

$$e(s,a) \leftarrow \min\{e(s,a)+1, 1\} \tag{B.3}$$

and the TD error is computed as

$$\delta \leftarrow r^{(p)} + \gamma_p \, Q(s',a') - Q(s,a) \tag{B.4}$$

Because there may be an element of randomness to the domain's rule for determining the next state and reward given the current state and action, we do not necessarily want to change our new estimated value to the sample value—we only move it towards that value, so the estimated value for action a from state s is then updated as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha_p \delta e(s,a) \tag{B.5}$$

where α_p is a learning rate parameter. To complete the prediction learner's update, the eligibility trace then decays as follows:

$$e(s,a) \leftarrow \lambda_p e(s,a) \qquad\qquad \forall s,a \in \mathcal{S} \times \mathcal{A} \tag{B.6}$$

The intrinsic reward is computed using the predictor's TD-error. Essentially, we maintain a smoothly averaged estimate, ξ, of recent TD-error and divide by the root sample variance to obtain the intrinsic reward (unexpected error). This method was provided by White (2015, p. 121).

The algorithm to maintain ξ utilizes a single parameter, β_0, and a holding variable τ. Holding variable τ and estimate ξ are initialized to $\tau \leftarrow 0$ and $\xi \leftarrow 0$. To update ξ during each time step, we perform the following two steps:

$$\tau \leftarrow (1-\beta_0)\tau + \beta_0 \tag{B.7}$$

$$\xi \leftarrow \left(1 - \frac{\beta_0}{\tau}\right)\xi + \frac{\beta_0}{\tau}\delta \tag{B.8}$$

and we provide the estimate incorporating the most recent TD-error divided by the sample variance to the control learner as the intrinsic reward. Using one parameter

β_0, the estimate ξ is maintained using a holding variable, τ. The sample variance, $\text{var}(\delta)$, is maintained using a standard incremental algorithm (see Appendix B.7 for the implementation) and the final intrinsic reward for the time step is

$$R_t^I = \frac{\xi}{\sqrt{\text{var}(\delta)} + c_2} \tag{B.9}$$

where c_2 is **held constant** at $c_2 = 0.001$ in our experiment.

For control, we used ε-greedy Watkins's Q(λ) (Sutton and Barto, 2018, p. 312-313). Q(λ)-learning maintains estimates of the *optimal* curiosity value Q^*, assuming the agent will choose the action with the highest curiosity value in the next step.

Watkins's Q(λ) uses updates analogous to those shown in equations (B.3)-(B.5), so our control component also uses analogous parameters α_c, γ_c, and λ_c.

To select an action, a random number between 0 and 1 is drawn. If the random number is less than ε, the agent will choose an action randomly (so all actions have equal probability), but otherwise, it chooses the action with the greatest curiosity value (hence the name, ε-greedy).

Like the domain, the described algorithms used to make predictions and select actions are **controlled** throughout the experiment, while the pseudo-random generation is **allowed to vary** because it represents the metaphorical 'coin-flip' used to decide in a slightly-random policy. However, the parameters α_p, γ_p, $\lambda_p, \beta_0, \alpha_c, \gamma_c, \lambda_c$, and ε are our **manipulated** variables.

B.1.3 Parameter Factor Ranges

Learning rates within the interval $[0, 2)$ are generally stable. However, the purpose of the learning rate is to minimize error. For this reason, the learning rates, α_p, α_c, are defined on $(0, 1]$ which are theoretically sound with regards to this purpose (Sutton and Barto, 2018). Also, the learning rate is known to typically have a non-linear effect on prediction error and return (Sutton and Barto, 2018, pp. 155, 43).

Responding Variable	
Return, G	
Manipulated Variable	**Coding Function**
Learning rate	$\alpha_p = 0.51 + 0.49x_{\alpha_p}$
Discount rate	$\gamma_p = 0.49 + 0.49x_{\gamma_p}$
Trace decay parameter	$\lambda_p = 0.49 + 0.49x_{\lambda_p}$
Unexpected error parameter	$\beta_0 = 0.5 + 0.49x_{\beta_0}$
Learning rate	$\alpha_c = 0.51 + 0.49x_{\alpha_c}$
Discount rate	$\gamma_c = 0.49 + 0.49x_{\gamma_c}$
Trace decay parameter	$\lambda_c = 0.49 + 0.49x_{\lambda_c}$
Probability of random action	$\varepsilon = 0.5 + 0.49x_{\varepsilon}$

Table B.1: Summary of full-factorial design and coding for factor levels.

The discount rates chosen for continuing tasks like our domain fall in the interval $[0, 1)$ (Sutton and Barto, 2018, p. 53). Setting $\gamma_p, \gamma_c < 1$ ensures the the value functions Q^{π} and Q^{*} are bounded. Prior work has shown that γ_p has an important effect on behaviour (Ady and Pilarski, 2016).

The trace decay parameters λ_p, λ_c are used to assign most credit for an observation to the most recent choice, and decreasing amounts of credit to historical choices. The amount of credit decays by a factor of λ in each step. Like γ_p, γ_c, the parameter is set within the interval $[0, 1)$ for continuing tasks. The trace decay parameter has

The parameter ε represents a probability, so is bounded to $[0, 1]$. However, if $\varepsilon = 0$, the agent will get stuck taking the action whose curiosity value estimate first exceeds the estimates for the other actions. Similarly, if $\varepsilon = 1$, the agent will never utilize its learning; it will act randomly at every time step. Therefore, we bound ε to the range $(0, 1)$.

B.1.4 Experimental Design

For a design summary, see Table B.1.

Since many of the parameters of interest are already expected to affect the response variables, and in many cases to have non-linear effects, we were interested in a response surface method (RSM) design to help capture this information in a simplified model.

A full-factorial design capturing non-linear effects of even five of the eight factors would take nearly 2000 tests for a single replication. Further, we hoped to run twenty replications to account for the randomness impacting each run. While this is a feasible number of tests, given that a single test takes less than a second to complete, the analysis of the resulting data requires a great deal of computation, and a quadratic model can be realized much more efficiently using an RSM.

We chose to utilize an inscribed central composite (CCI) design. CCI designs are suitable RSM designs when the extrema of the included factors represent hard limits. As described in section B.1.3, the ranges of interest represent the reasonable limits for each parameter, so a CCI design was a reasonable choice. The test order was fully randomized.

B.2 Results

The linear models were created using STATISTICA 13.

B.2.4 Results for Return

An analysis of variance (ANOVA) was performed to determine the significant factors and interactions. Our initial model including all linear effects and quadratic interactions was validated using a run sequence plot of the residuals (B.4), a normal probability plot of the residuals (B.5), and a scatter plot of the predicted values against the residuals (B.6), shown in Appendix B. We defer discussion of the validation plots to section B.2.5. From the Effect Estimates (Table B.2), ANOVA (Table B.3), and Pareto chart of standardized effects (Figure B.1), we determined that the linear effects for ε, β_0, α_c, and γ_p were significant, along with the quadratic interactions between ε and β_0, ε and γ_p, β_0 and γ_p, and α_p and γ_p.

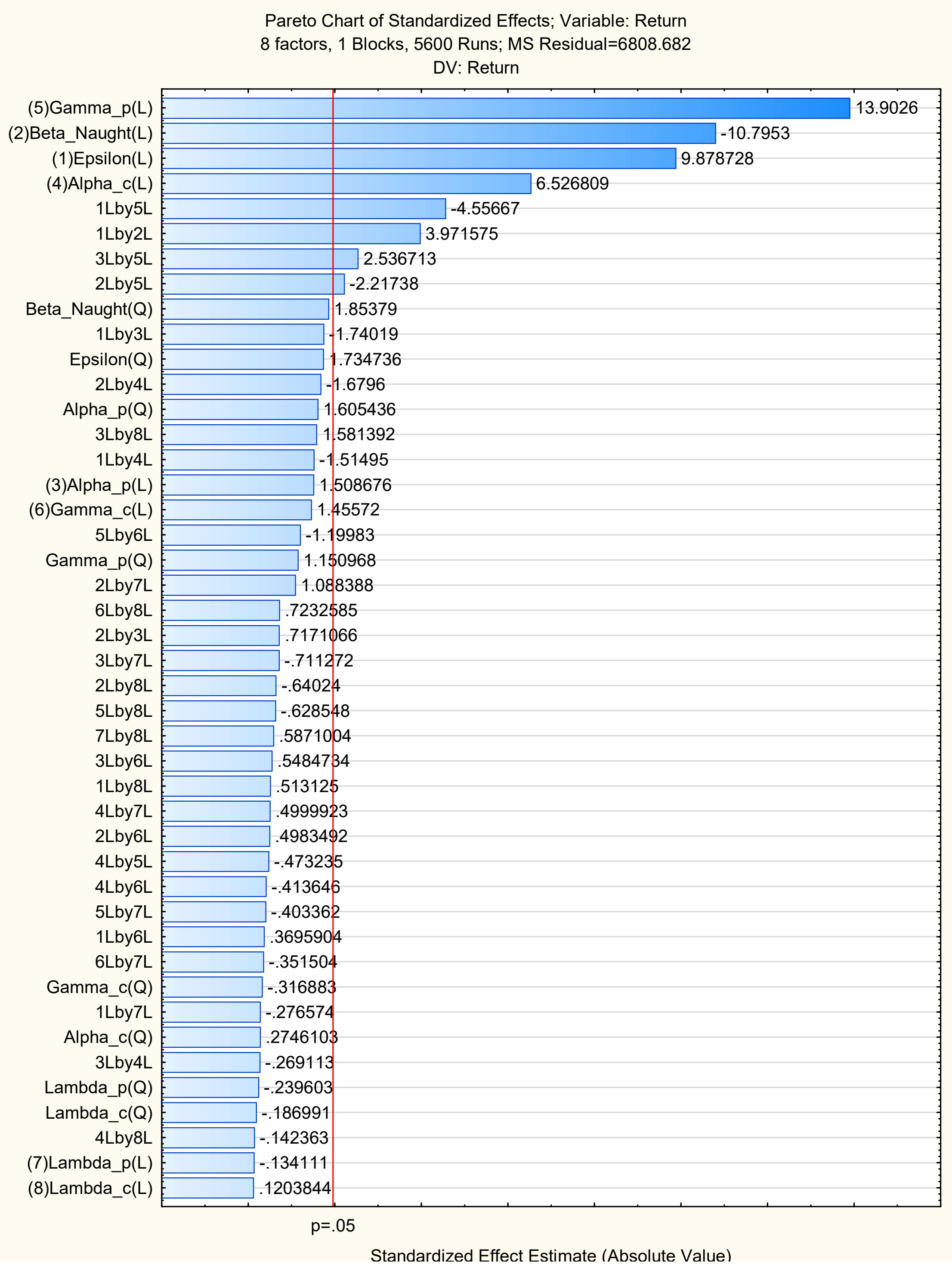


Figure B.1: A Pareto chart showing the standardized effects on return for the initial model with all linear and quadratic effects.

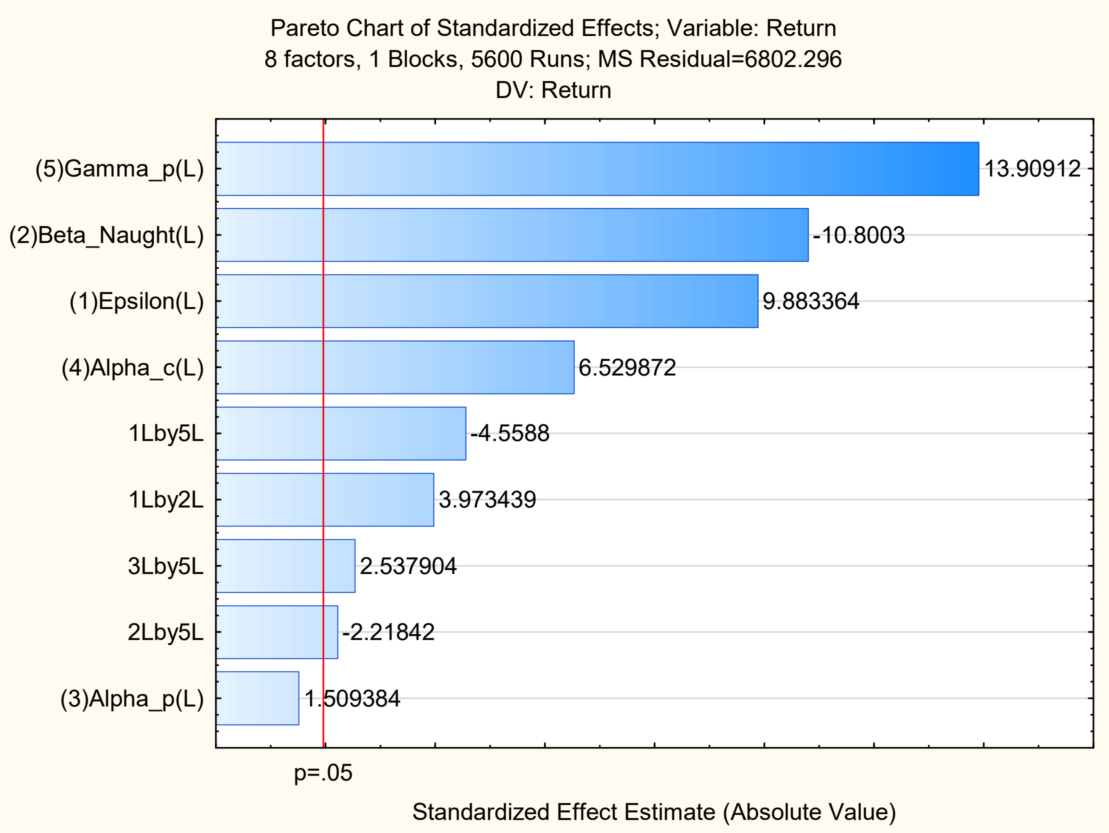


Figure B.2: In this Pareto chart, the effects with bars surpassing the red line (that is, those effects which have a component to the right of the red line) are significant in the reduced model.

We reduced our model by removing all insignificant effects, up to fulfilling the hierarchy principle (we keep the linear effect of α_p, despite its relatively high p-value). We then performed the same statistical analysis for our new model. Again, the model was visually validated (Figures B.5 and B.7, and B.8 in Appendix B.6) and the Effect Estimates (Table B.4) and ANOVA (Table B.5) computed. As is shown by the Pareto chart in Figure B.2, the effects listed in the previous paragraph remained significant.

With our reduced model, the return G can be written as a function of coded parameters as follows:

$$\begin{aligned} G = 139.5106 + 10.9775x_{\varepsilon} - 11.9960x_{\beta_0} + 1.6765x_{\alpha_p} + 7.2528x_{\alpha_c} + 15.4490x_{\gamma_p} \\ + 4.5799x_{\varepsilon}x_{\beta_0} - 5.2546x_{\varepsilon}x_{\gamma_p} - 2.5570x_{\beta_0}x_{\gamma_p} + 2.9253x_{\alpha_p}x_{\gamma_p} \end{aligned} \tag{B.10}$$

B.2.5 Model Validation

The assumptions of the ANOVA procedure require that the residuals of the linear model are normal.

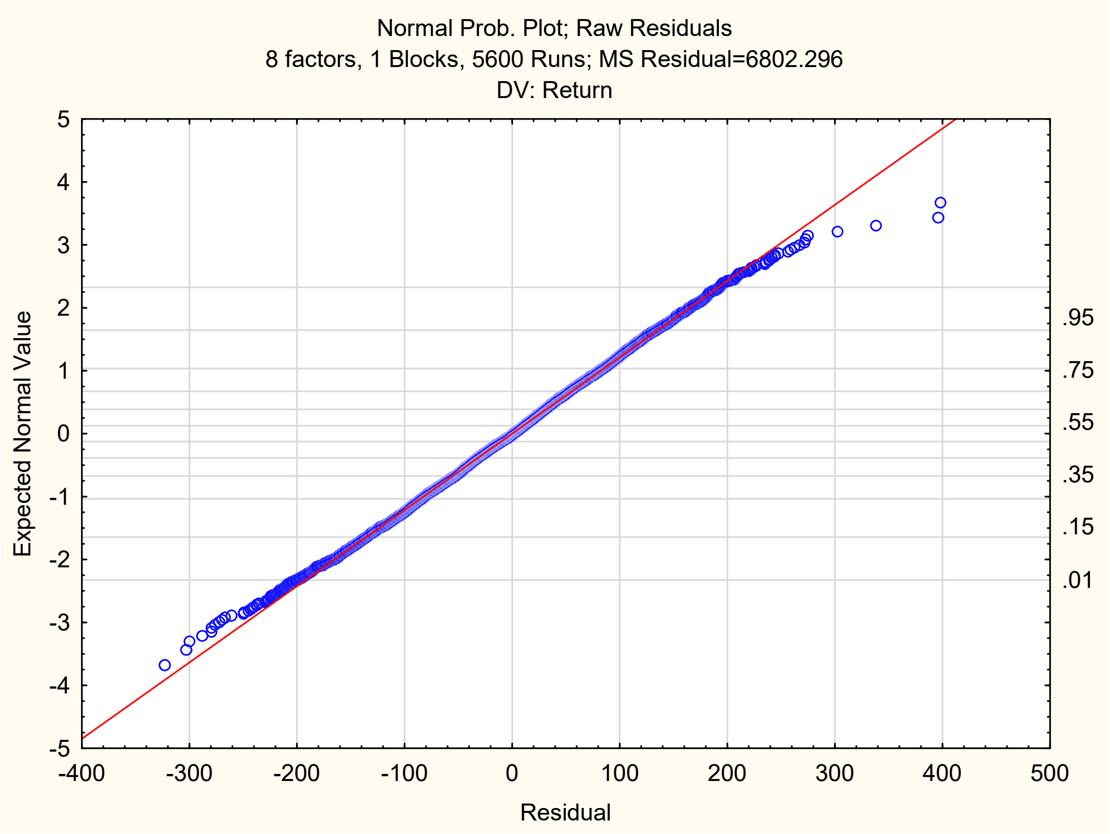


Figure B.3: This plot shows the normal probability plot for the residuals, given our reduced model.

Unfortunately, we can see in Figure B.3, that there are outliers. Interestingly, those outliers are runs where $x_\varepsilon = -1$, and all other coded variables are set to 0. This suggests that our model may fail in nearby cases.

On the other hand, neither the run-sequence plot of residuals in Figure B.7 nor the plot of residuals as a function of value in Figure B.8 appear to show any trend suggesting further invalidity of the model.

B.2.6 Optimization

A system designer may be interested in maximizing return while still using this curiosity method. We used `sqp` in Octave, and found that the following coded values maximized return according to our model, with a predicted expected return of 182.51.

$$x_{\varepsilon} = 1 \tag{B.11}$$

$$x_{\beta_0} = -1 \tag{B.12}$$

$$x_{\alpha_p} = 1 \tag{B.13}$$

$$x_{\alpha_c} = 1 \tag{B.14}$$

$$x_{\gamma_p} = 1 \tag{B.15}$$

B.3 Discussion

The initial objectives of this work were to find parameters that significantly affect the behaviour of a reinforcement learning agent controlled using White's unexpected error as a intrinsic reward.

To the best of the author's knowledge, there have been no prior attempts to determine which parameters in a curious agent impact its behaviour.

Using return as the response variable provided limited insight into this issue. Parameters which result in significantly different final return must have resulted in significantly different behaviour to do so. However, this experiment does not exhaust the possibility that some parameters which cause significantly different behaviour could still result in similar return.

Despite its limitations, return is an interesting response variable, as there could be situations where the system designer would like to maximize return while still requiring the learning system to utilize this kind of intrinsic reward.

In future work it will be crucial to utilize more descriptive measures of behaviour than the return. Such measures could include the average probability of each action or other measures of the agent's predictive error.

One response variable that was not tested in these experiments was the difference between the agent's prediction and the observed truth: the experimenter can compute, after the trial's completion, the actual value for a given time step and compare it to the prediction made by the agent. While this response variable measures how well the agent is predicting what it observes, it fails to measure the agent's knowledge of all possible situations. For example agent which always selects the constant action a_2 and never updates its predictions from $Q(s_0, a_2) = 0$ would always predict every situation it sees perfectly, but would be a poor predictor of its accessible environment.

B.4 Conclusions

We found that the significant factors were the linear effects for ε, β_0, α_c, and γ_p, along with the quadratic interactions between ε and β_0, ε and γ_p, β_0 and γ_p, and α_p and γ_p.

To maximize return, we found that the best values in our utilized ranges for the significant parameters were as follows:

$$\varepsilon = 0.99 \tag{B.16}$$

$$\beta_0 = 0.01 \tag{B.17}$$

$$\alpha_p = 1 \tag{B.18}$$

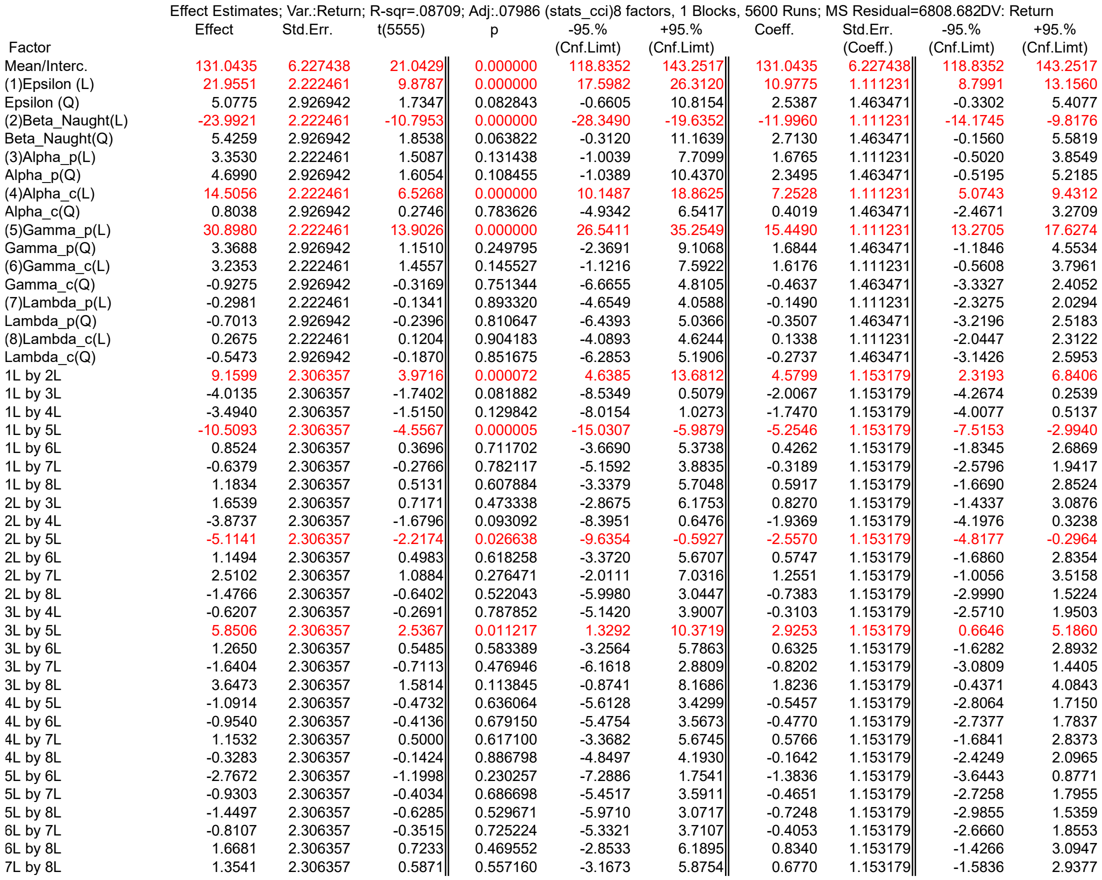
$$\alpha_c = 1 \tag{B.19}$$

$$\gamma_p = 0.98 \tag{B.20}$$

B.5 Raw Data

The raw data is available at https://drive.google.com/a/ualberta.ca/file/d/0B5rsyN1Hdb1qd1B4OGQ3dGxVc0U/view?usp=sharing

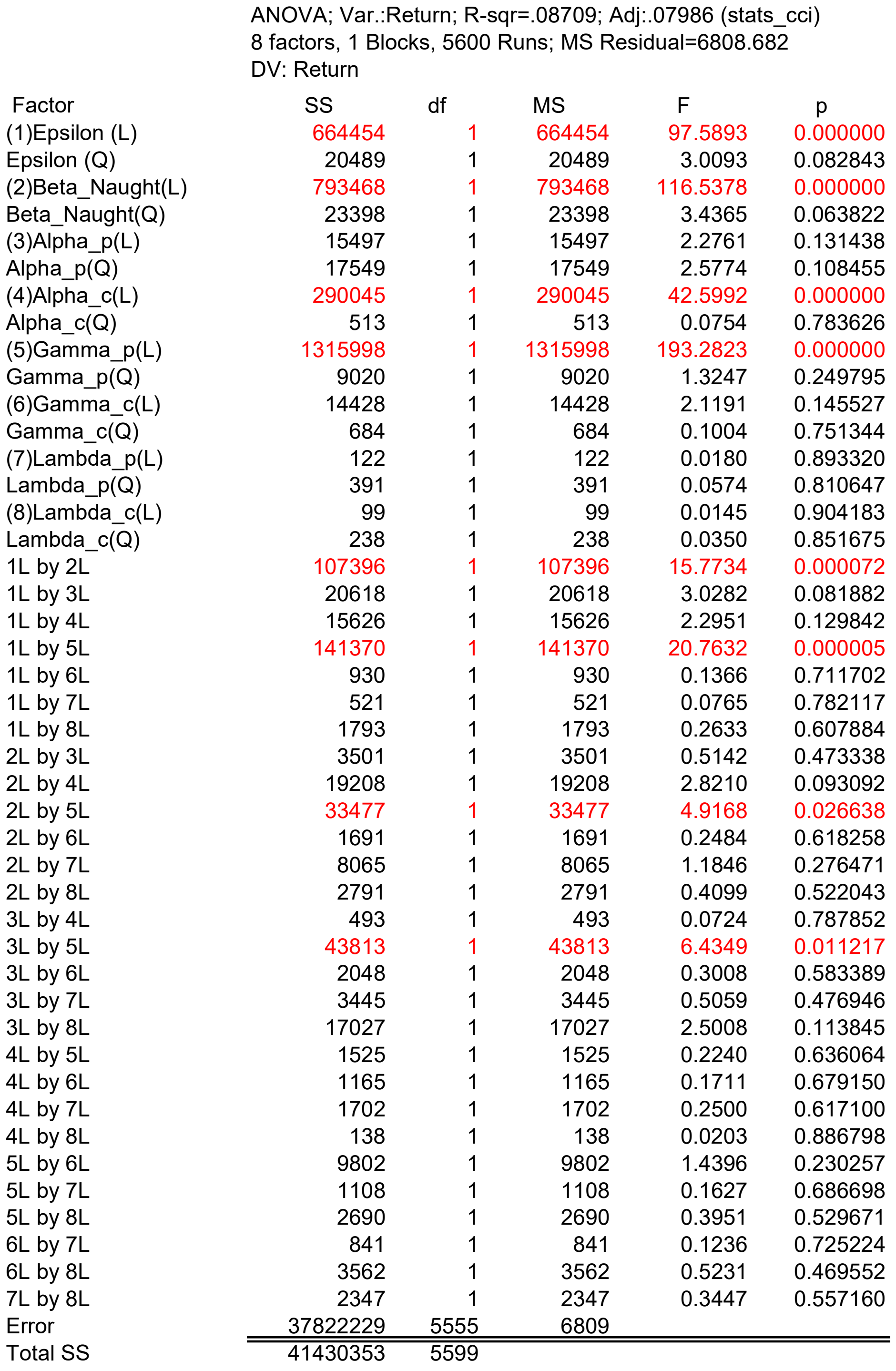
B.6 Additional Tables and Plots (Return)



Effect Estimates; Var.:Return; R-sqr=.08709; Adj:.07986 (stats_cci)8 factors, 1 Blocks, 5600 Runs; MS Residual=6808.682DV: Return

	Effect	Std.Err.	t(5555)	p	-95.%	+95.%	Coeff.	Std.Err.	-95.%	+95.%
Factor					(Cnf.Limt)	(Cnf.Limt)		(Coeff.)	(Cnf.Limt)	(Cnf.Limt)
Mean/Interc.	131.0435	6.227438	21.0429	0.000000	118.8352	143.2517	131.0435	6.227438	118.8352	143.2517
(1)Epsilon (L)	21.9551	2.222461	9.8787	0.000000	17.5982	26.3120	10.9775	1.111231	8.7991	13.1560
Epsilon (Q)	5.0775	2.926942	1.7347	0.082843	-0.6605	10.8154	2.5387	1.463471	-0.3302	5.4077
(2)Beta_Naught(L)	-23.9921	2.222461	-10.7953	0.000000	-28.3490	-19.6352	-11.9960	1.111231	-14.1745	-9.8176
Beta_Naught(Q)	5.4259	2.926942	1.8538	0.063822	-0.3120	11.1639	2.7130	1.463471	-0.1560	5.5819
(3)Alpha_p(L)	3.3530	2.222461	1.5087	0.131438	-1.0039	7.7099	1.6765	1.111231	-0.5020	3.8549
Alpha_p(Q)	4.6990	2.926942	1.6054	0.108455	-1.0389	10.4370	2.3495	1.463471	-0.5195	5.2185
(4)Alpha_c(L)	14.5056	2.222461	6.5268	0.000000	10.1487	18.8625	7.2528	1.111231	5.0743	9.4312
Alpha_c(Q)	0.8038	2.926942	0.2746	0.783626	-4.9342	6.5417	0.4019	1.463471	-2.4671	3.2709
(5)Gamma_p(L)	30.8980	2.222461	13.9026	0.000000	26.5411	35.2549	15.4490	1.111231	13.2705	17.6274
Gamma_p(Q)	3.3688	2.926942	1.1510	0.249795	-2.3691	9.1068	1.6844	1.463471	-1.1846	4.5534
(6)Gamma_c(L)	3.2353	2.222461	1.4557	0.145527	-1.1216	7.5922	1.6176	1.111231	-0.5608	3.7961
Gamma_c(Q)	-0.9275	2.926942	-0.3169	0.751344	-6.6655	4.8105	-0.4637	1.463471	-3.3327	2.4052
(7)Lambda_p(L)	-0.2981	2.222461	-0.1341	0.893320	-4.6549	4.0588	-0.1490	1.111231	-2.3275	2.0294
Lambda_p(Q)	-0.7013	2.926942	-0.2396	0.810647	-6.4393	5.0366	-0.3507	1.463471	-3.2196	2.5183
(8)Lambda_c(L)	0.2675	2.222461	0.1204	0.904183	-4.0893	4.6244	0.1338	1.111231	-2.0447	2.3122
Lambda_c(Q)	-0.5473	2.926942	-0.1870	0.851675	-6.2853	5.1906	-0.2737	1.463471	-3.1426	2.5953
1L by 2L	9.1599	2.306357	3.9716	0.000072	4.6385	13.6812	4.5799	1.153179	2.3193	6.8406
1L by 3L	-4.0135	2.306357	-1.7402	0.081882	-8.5349	0.5079	-2.0067	1.153179	-4.2674	0.2539
1L by 4L	-3.4940	2.306357	-1.5150	0.129842	-8.0154	1.0273	-1.7470	1.153179	-4.0077	0.5137
1L by 5L	-10.5093	2.306357	-4.5567	0.000005	-15.0307	-5.9879	-5.2546	1.153179	-7.5153	-2.9940
1L by 6L	0.8524	2.306357	0.3696	0.711702	-3.6690	5.3738	0.4262	1.153179	-1.8345	2.6869
1L by 7L	-0.6379	2.306357	-0.2766	0.782117	-5.1592	3.8835	-0.3189	1.153179	-2.5796	1.9417
1L by 8L	1.1834	2.306357	0.5131	0.607884	-3.3379	5.7048	0.5917	1.153179	-1.6690	2.8524
2L by 3L	1.6539	2.306357	0.7171	0.473338	-2.8675	6.1753	0.8270	1.153179	-1.4337	3.0876
2L by 4L	-3.8737	2.306357	-1.6796	0.093092	-8.3951	0.6476	-1.9369	1.153179	-4.1976	0.3238
2L by 5L	-5.1141	2.306357	-2.2174	0.026638	-9.6354	-0.5927	-2.5570	1.153179	-4.8177	-0.2964
2L by 6L	1.1494	2.306357	0.4983	0.618258	-3.3720	5.6707	0.5747	1.153179	-1.6860	2.8354
2L by 7L	2.5102	2.306357	1.0884	0.276471	-2.0111	7.0316	1.2551	1.153179	-1.0056	3.5158
2L by 8L	-1.4766	2.306357	-0.6402	0.522043	-5.9980	3.0447	-0.7383	1.153179	-2.9990	1.5224
3L by 4L	-0.6207	2.306357	-0.2691	0.787852	-5.1420	3.9007	-0.3103	1.153179	-2.5710	1.9503
3L by 5L	5.8506	2.306357	2.5367	0.011217	1.3292	10.3719	2.9253	1.153179	0.6646	5.1860
3L by 6L	1.2650	2.306357	0.5485	0.583389	-3.2564	5.7863	0.6325	1.153179	-1.6282	2.8932
3L by 7L	-1.6404	2.306357	-0.7113	0.476946	-6.1618	2.8809	-0.8202	1.153179	-3.0809	1.4405
3L by 8L	3.6473	2.306357	1.5814	0.113845	-0.8741	8.1686	1.8236	1.153179	-0.4371	4.0843
4L by 5L	-1.0914	2.306357	-0.4732	0.636064	-5.6128	3.4299	-0.5457	1.153179	-2.8064	1.7150
4L by 6L	-0.9540	2.306357	-0.4136	0.679150	-5.4754	3.5673	-0.4770	1.153179	-2.7377	1.7837
4L by 7L	1.1532	2.306357	0.5000	0.617100	-3.3682	5.6745	0.5766	1.153179	-1.6841	2.8373
4L by 8L	-0.3283	2.306357	-0.1424	0.886798	-4.8497	4.1930	-0.1642	1.153179	-2.4249	2.0965
5L by 6L	-2.7672	2.306357	-1.1998	0.230257	-7.2886	1.7541	-1.3836	1.153179	-3.6443	0.8771
5L by 7L	-0.9303	2.306357	-0.4034	0.686698	-5.4517	3.5911	-0.4651	1.153179	-2.7258	1.7955
5L by 8L	-1.4497	2.306357	-0.6285	0.529671	-5.9710	3.0717	-0.7248	1.153179	-2.9855	1.5359
6L by 7L	-0.8107	2.306357	-0.3515	0.725224	-5.3321	3.7107	-0.4053	1.153179	-2.6660	1.8553
6L by 8L	1.6681	2.306357	0.7233	0.469552	-2.8533	6.1895	0.8340	1.153179	-1.4266	3.0947
7L by 8L	1.3541	2.306357	0.5871	0.557160	-3.1673	5.8754	0.6770	1.153179	-1.5836	2.9377

Table B.2: Given our initial model, this table shows all linear effects and quadratic interactions.



ANOVA; Var.:Return; R-sqr=.08709; Adj:.07986 (stats_cci)
8 factors, 1 Blocks, 5600 Runs; MS Residual=6808.682
DV: Return

Factor	SS	df	MS	F	p
(1)Epsilon (L)	664454	1	664454	97.5893	0.000000
Epsilon (Q)	20489	1	20489	3.0093	0.082843
(2)Beta_Naught(L)	793468	1	793468	116.5378	0.000000
Beta_Naught(Q)	23398	1	23398	3.4365	0.063822
(3)Alpha_p(L)	15497	1	15497	2.2761	0.131438
Alpha_p(Q)	17549	1	17549	2.5774	0.108455
(4)Alpha_c(L)	290045	1	290045	42.5992	0.000000
Alpha_c(Q)	513	1	513	0.0754	0.783626
(5)Gamma_p(L)	1315998	1	1315998	193.2823	0.000000
Gamma_p(Q)	9020	1	9020	1.3247	0.249795
(6)Gamma_c(L)	14428	1	14428	2.1191	0.145527
Gamma_c(Q)	684	1	684	0.1004	0.751344
(7)Lambda_p(L)	122	1	122	0.0180	0.893320
Lambda_p(Q)	391	1	391	0.0574	0.810647
(8)Lambda_c(L)	99	1	99	0.0145	0.904183
Lambda_c(Q)	238	1	238	0.0350	0.851675
1L by 2L	107396	1	107396	15.7734	0.000072
1L by 3L	20618	1	20618	3.0282	0.081882
1L by 4L	15626	1	15626	2.2951	0.129842
1L by 5L	141370	1	141370	20.7632	0.000005
1L by 6L	930	1	930	0.1366	0.711702
1L by 7L	521	1	521	0.0765	0.782117
1L by 8L	1793	1	1793	0.2633	0.607884
2L by 3L	3501	1	3501	0.5142	0.473338
2L by 4L	19208	1	19208	2.8210	0.093092
2L by 5L	33477	1	33477	4.9168	0.026638
2L by 6L	1691	1	1691	0.2484	0.618258
2L by 7L	8065	1	8065	1.1846	0.276471
2L by 8L	2791	1	2791	0.4099	0.522043
3L by 4L	493	1	493	0.0724	0.787852
3L by 5L	43813	1	43813	6.4349	0.011217
3L by 6L	2048	1	2048	0.3008	0.583389
3L by 7L	3445	1	3445	0.5059	0.476946
3L by 8L	17027	1	17027	2.5008	0.113845
4L by 5L	1525	1	1525	0.2240	0.636064
4L by 6L	1165	1	1165	0.1711	0.679150
4L by 7L	1702	1	1702	0.2500	0.617100
4L by 8L	138	1	138	0.0203	0.886798
5L by 6L	9802	1	9802	1.4396	0.230257
5L by 7L	1108	1	1108	0.1627	0.686698
5L by 8L	2690	1	2690	0.3951	0.529671
6L by 7L	841	1	841	0.1236	0.725224
6L by 8L	3562	1	3562	0.5231	0.469552
7L by 8L	2347	1	2347	0.3447	0.557160
Error	37822229	5555	6809		
Total SS	41430353	5599			

Table B.3: Given our initial model for all linear effects and quadratic interactions, this table provides ANOVA data.

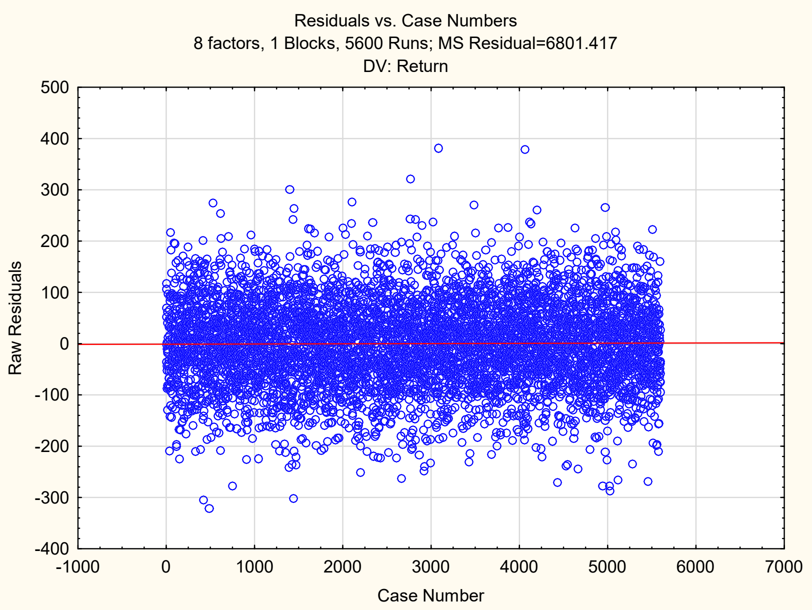


Figure B.4: This plot shows the raw residuals for the initial model including all linear and quadratic effects as a function of case number (equivalently, the raw residuals are shown in the run sequence order).

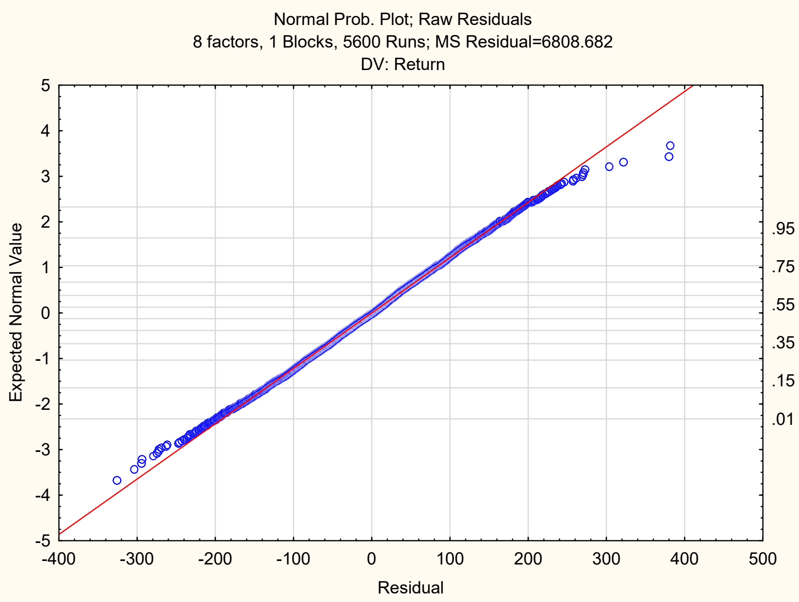


Figure B.5: This plot shows the normal probability plot for the residuals, given our initial model including all linear and quadratic effects.

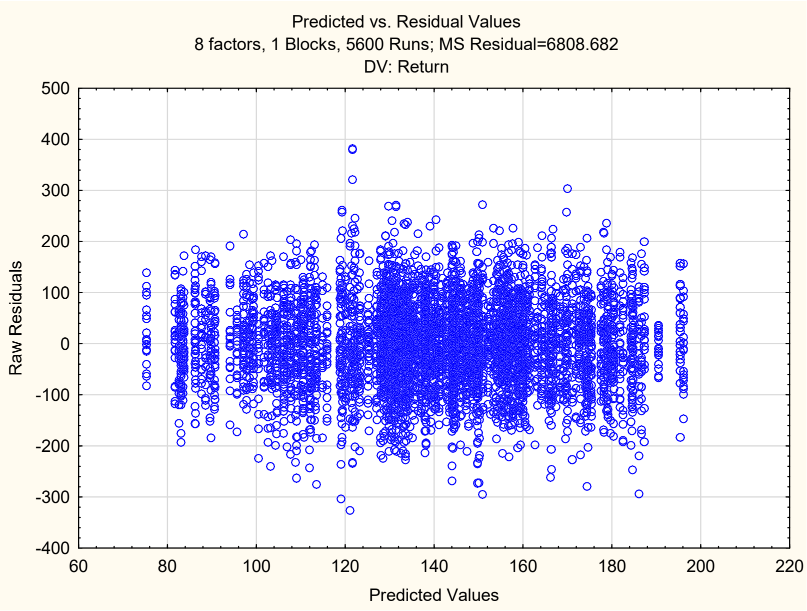
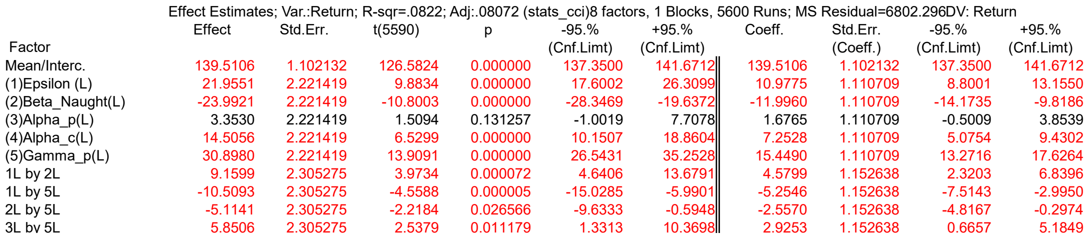


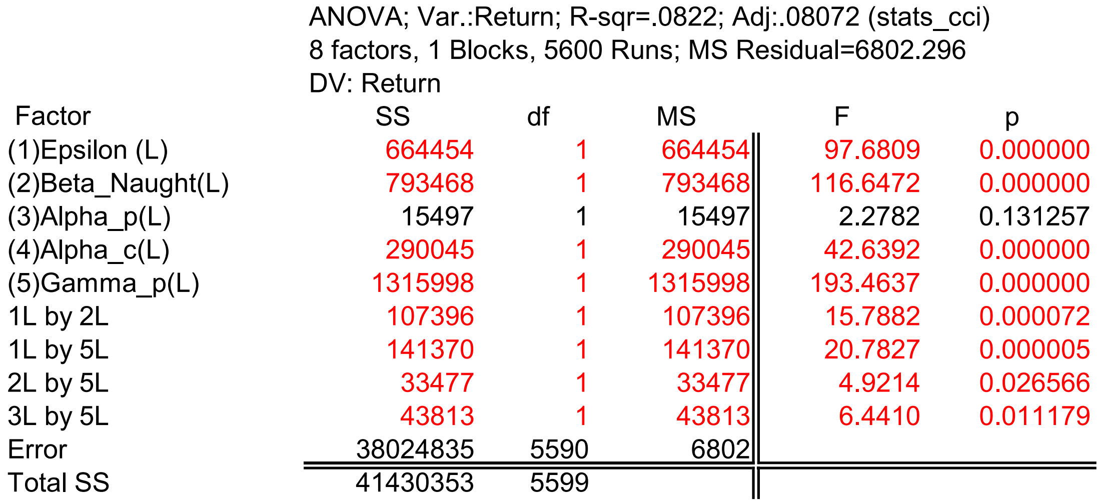
Figure B.6: This plot shows the raw residuals as a function of the predicted values for the initial model including all linear and quadratic effects.



Effect Estimates; Var.:Return; R-sqr=.0822; Adj:.08072 (stats_cci)8 factors, 1 Blocks, 5600 Runs; MS Residual=6802.296DV: Return

Factor	Effect	Std.Err.	t(5590)	p	-95.% (Cnf.Limt)	+95.% (Cnf.Limt)	Coeff.	Std.Err. (Coeff.)	-95.% (Cnf.Limt)	+95.% (Cnf.Limt)
Mean/Interc.	139.5106	1.102132	126.5824	0.000000	137.3500	141.6712	139.5106	1.102132	137.3500	141.6712
(1)Epsilon (L)	21.9551	2.221419	9.8834	0.000000	17.6002	26.3099	10.9775	1.110709	8.8001	13.1550
(2)Beta_Naught(L)	-23.9921	2.221419	-10.8003	0.000000	-28.3469	-19.6372	-11.9960	1.110709	-14.1735	-9.8186
(3)Alpha_p(L)	3.3530	2.221419	1.5094	0.131257	-1.0019	7.7078	1.6765	1.110709	-0.5009	3.8539
(4)Alpha_c(L)	14.5056	2.221419	6.5299	0.000000	10.1507	18.8604	7.2528	1.110709	5.0754	9.4302
(5)Gamma_p(L)	30.8980	2.221419	13.9091	0.000000	26.5431	35.2528	15.4490	1.110709	13.2716	17.6264
1L by 2L	9.1599	2.305275	3.9734	0.000072	4.6406	13.6791	4.5799	1.152638	2.3203	6.8396
1L by 5L	-10.5093	2.305275	-4.5588	0.000005	-15.0285	-5.9901	-5.2546	1.152638	-7.5143	-2.9950
2L by 5L	-5.1141	2.305275	-2.2184	0.026566	-9.6333	-0.5948	-2.5570	1.152638	-4.8167	-0.2974
3L bv 5L	5.8506	2.305275	2.5379	0.011179	1.3313	10.3698	2.9253	1.152638	0.6657	5.1849

Table B.4: This table provides the main effects and model coefficients for our reduced model.



ANOVA; Var.:Return; R-sqr=.0822; Adj:.08072 (stats_cci)
8 factors, 1 Blocks, 5600 Runs; MS Residual=6802.296
DV: Return

Factor	SS	df	MS	F	p
(1)Epsilon (L)	664454	1	664454	97.6809	0.000000
(2)Beta_Naught(L)	793468	1	793468	116.6472	0.000000
(3)Alpha_p(L)	15497	1	15497	2.2782	0.131257
(4)Alpha_c(L)	290045	1	290045	42.6392	0.000000
(5)Gamma_p(L)	1315998	1	1315998	193.4637	0.000000
1L by 2L	107396	1	107396	15.7882	0.000072
1L by 5L	141370	1	141370	20.7827	0.000005
2L by 5L	33477	1	33477	4.9214	0.026566
3L by 5L	43813	1	43813	6.4410	0.011179
Error	38024835	5590	6802		
Total SS	41430353	5599			

Table B.5: This table provides ANOVA data for our reduced model for which all insignificant effects have been removed.

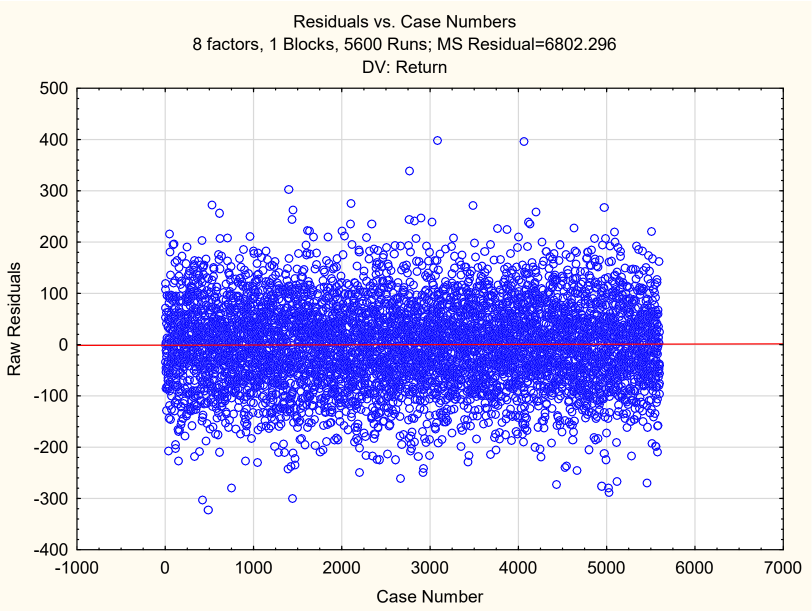


Figure B.7: This plot shows the raw residuals for the reduced model as a function of case number (equivalently, the raw residuals are shown in the run sequence order).

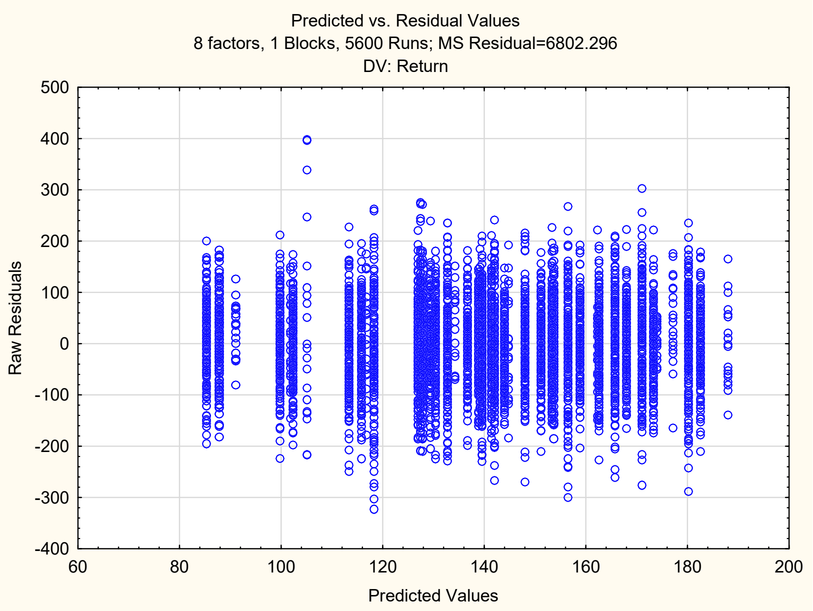


Figure B.8: This plot shows the raw residuals as a function of the predicted values for the reduced model.

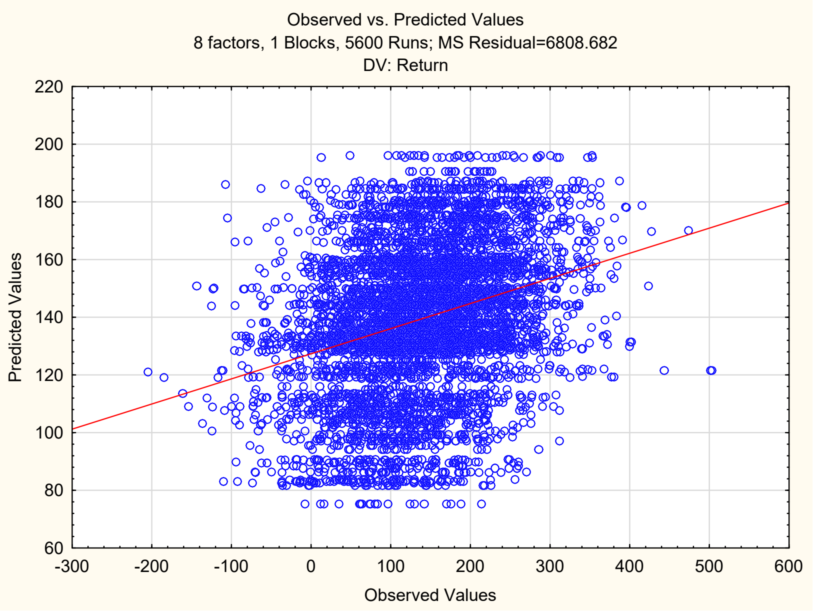


Figure B.9: This plot shows values predicted by the reduced model vs observed values.

B.7 Code for Parameter Screening Experiments

The following code runs all of the experiments and produces a csv (comma-separated-values) file containing the raw experimental data.

```
# parameter_screening.py
# Copyright (C) 2017 Nadia Ady
#
# This module is part of the curiosity project.
# To run, type: python parameter_screening.py

import numpy    # used for random functions


class Test(object):
    def __init__(self, domain, num_steps):
        self.domain = domain
        self.num_steps = num_steps
        self.return_sum = 0.0

        # tracks counts for each state-action pair
        self.count = {}
        for state in domain.get_state_set():
            self.count[state] = {}
            for action in domain.get_action_set():
                self.count[state][action] = 0

    def run(self, agent, initial_state):
        self.return_sum = 0.0     # reset before starting a new test
        state = initial_state
        action = agent.get_action(state)
        for step_num in range(self.num_steps):

            self.count[state][action] += 1

            domain_reward, new_state = self.domain.sample(state,
                                                          action)
            new_action = agent.get_action(new_state)

            agent.update(state, action, domain_reward,
                         new_state, new_action)

            state = new_state
            action = new_action
            self.return_sum += domain_reward


class CuriosityBandit(object):
    def __init__(self, offset=0.001, start_time=0,
                 update_sin_every_step=True, random_seed=None):
        # constant to affect the frequency of the sinusoidal action
        self.offset = offset
        # sin_argument holds self.offset * timestep
        self.sin_argument = start_time
        # if False, sinusoid action only shifts when action is taken
        self.update_sin_every_step = update_sin_every_step

        if random_seed is None:
            numpy.random.seed()
        else:
            numpy.random.seed(random_seed)

    def sample(self, state, action):
        if self.update_sin_every_step or action == 1:
            self.sin_argument += self.offset
        reward = self.get_reward(action)
        return reward, state

    def get_reward(self, action):
        if action == 0:      # random
            return 2 * numpy.random.random_sample() - 1
        elif action == 1:    # sinusoidal
            return numpy.sin(self.sin_argument)
        elif action == 2:    # constant
            return 0

    def get_state_set(self):
        """The returned list of states is a singleton."""
        return [0]

    def get_action_set(self):
        return [1, 0, 2]

    def get_initial_state(self):
        return 0


class ActionValuedAgent(object):
    def __init__(self, domain, params, random_seed=None):
        self.params = params
        self.domain = domain
        if random_seed is not None:
            numpy.random.seed(random_seed)

        self.gamma = self.params[’gamma’] if ’gamma’ in \
                                             self.params else 0.9
        self.alpha = self.params[’alpha’] if ’alpha’ in \
                                             self.params else 0.1
        self.epsilon = self.params[’epsilon’] if \
            ’epsilon’ in self.params else 0.1
        self.initial_value = self.params[’initial_value’] if \
            ’initialValue’ in self.params else 0
        self.decay = self.params[’lambda’] if ’lambda’ in \
                                              self.params else 0
        self.UDE_keeper = UDE(self.params[’beta_naught’]) if \
            ’beta_naught’ in self.params else UDE(0.1)
        self.Q = {s: {a: self.initial_value for a in
                      domain.get_action_set()} for s in
                  domain.get_state_set()}
        self.trace = {state: {action: 0 for action in
                              domain.get_action_set()} for state in
                      domain.get_state_set()}
        self.delta = 0

    def get_action(self, state):
        if state not in self.Q:
            self.Q[state] = {a: self.initial_value for a in
                             self.domain.get_action_set()}
        if numpy.random.random() < self.epsilon:
            action_set = self.domain.get_action_set()
            action = numpy.random.choice(action_set)
        else:
            action = numpy.random.choice(
                [k for k, v in self.Q[state].iteritems() if
                 v == max(self.Q[state].values())])
        self.last_action = action
        return action


class WatkinsQLearningAgent(ActionValuedAgent):
    def __init__(self, domain, params):
        super(WatkinsQLearningAgent, self).__init__(domain, params)

    def update(self, state, action, reward, state_new, action_new):

        astar = max(self.Q[state_new], key=self.Q[state_new].get)
        if self.Q[state_new][action_new] == self.Q[state_new][astar]:
            astar = action_new

        self.delta = reward + \
                     self.gamma * self.Q[state_new][astar] - \
                     self.Q[state][action]

        self.trace[state][action] += 1
        self.trace[state][action] = 1 if \
            self.trace[state][action] >= 1 else \
            self.trace[state][action]

        for s in self.trace:
            for a in self.trace[s]:
                self.Q[s][a] += self.alpha * self.delta * \
                                self.trace[s][a]
                if action_new == astar:
                    self.trace[s][a] *= self.gamma * self.decay
                else:
                    self.trace[s][a] = 0

        # update used to compute the intrinsic reward.
        self.UDE_keeper.update(self.delta)


class TDLambdaAgent(ActionValuedAgent):
    def __init__(self, domain, params):
        super(TDLambdaAgent, self).__init__(domain, params)

    def update(self, state, action, reward, state_new, action_new):

        self.delta = reward + \
                     self.gamma * self.Q[state_new][action_new] - \
                     self.Q[state][action]
        for s in self.trace:
            for a in self.trace[s]:
                self.trace[s][a] *= self.gamma * self.decay
        self.trace[state][action] += 1
        self.trace[state][action] = 1 if \
            self.trace[state][action] >= 1 else \
            self.trace[state][action]
        for s in self.Q:
            for a in self.Q[s]:
                self.Q[s][a] += self.alpha * self.delta * \
                                self.trace[s][a]

        self.UDE_keeper.update(self.delta)
```

```
class MultiBrainedAgent(object):
    def __init__(self, domain, params):
        self.domain = domain
        self.params = params

        assert 'control' in self.params
        self.control_agent = \
            self.params['control']['agent_type'](domain,
                        self.params['control']['params'])

        assert 'predictor' in self.params
        self.predictor_agent = \
            self.params['predictor']['agent_type'](domain,
                        self.params['predictor']['params'])

        assert 'control_reward' in self.params
        self.control_reward = self.params['control_reward']

    def get_action(self, state):
        return self.control_agent.get_action(state)

    def update(self, state, action, reward, state_new, action_new):
        self.predictor_agent.update(state, action, reward,
                                    state_new, action_new)

        curiosity_reward = self.control_reward(self.predictor_agent)

        self.control_agent.update(state, action, curiosity_reward,
                                  state_new, action_new)


class UDE(object):
    def __init__(self, beta_naught, small_constant=0.0001):
        self.beta_naught = beta_naught
        self.small_constant = small_constant
        self.knower_of_variance = SampleHolder()
        self.tau = 0
        self.learned_avg_delta = 0
        self.beta = None

    def update(self, delta):
        # tau_{t+1}
        self.tau = (1-self.beta_naught)*self.tau + self.beta_naught
        self.beta = self.beta_naught/self.tau

        # learn variance of delta
        self.knower_of_variance.add_variable(delta)
        v = self.knower_of_variance.get_variance()

        # learn exponentially weighted moving average of delta
        self.learned_avg_delta = (1 - self.beta) * \
                                 self.learned_avg_delta + \
                                 self.beta * delta

    def get_output(self):
        return abs( float(self.learned_avg_delta) /
                    (self.knower_of_variance.get_variance() +
                     self.small_constant))


# https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance
class SampleHolder(object):
    def __init__(self):
        self.K = 0
        self.n = 0
        self.ex = 0
        self.ex2 = 0

    def add_variable(self,x):
        if self.n == 0:
            self.K = x
        self.n += 1
        self.ex += x - self.K
        self.ex2 += (x - self.K) * (x - self.K)

    def remove_variable(self,x):
        self.n -= 1
        self.ex -= (x - self.K)
        self.ex2 -= (x - self.K) * (x - self.K)

    def get_mean(self):
        return self.K + self.ex / self.n

    def get_variance(self):
        if self.n == 0:
            return 0
        if self.n == 1:
            return (self.ex2 - (self.ex*self.ex) / self.n) / self.n
        return (self.ex2-(self.ex * self.ex)/self.n)/(self.n-1)


if __name__ == "__main__":
```

```
init_random_seed = 2017
num_steps = 20000
num_replicates = 20

filename = 'stats.csv'

epsilons = {-1: 0.01, 0: 0.5, 1: 0.99}
beta_naughts = {-1: 0.01, 1: 0.99}
alpha_ps = {-1: 0.01, 0: 0.5, 1: 0.99}
alpha_cs = {-1: 0.01, 0: 0.5, 1: 0.99}
gamma_ps = {-1: 0, 0: 0.49, 1: 0.98}
gamma_cs = {-1: 0, 0: 0.49, 1: 0.98}
lambda_ps = {-1: 0, 0: 0.49, 1: 0.98}
lambda_cs = {-1: 0, 0: 0.49, 1: 0.98}

with open(filename, 'w') as f:
    f.write('Epsilon,Beta_Naught,Alpha_p,Alpha_c,Gamma_p,' +
            'Gamma_c,Lambda_p,Lambda_c,Percent␣Periodic,' +
            'Percent␣Random,Percent␣Constant,Return\n')

code_combos = [(epsiloncode, b0code, apcode, accode,
                gpcode, gccode, lpcode, lccode)
               for epsiloncode in epsilons
               for b0code in beta_naughts
               for apcode in alpha_ps
               for accode in alpha_cs
               for gpcode in gamma_ps
               for gccode in gamma_cs
               for lpcode in lambda_ps
               for lccode in lambda_cs]*num_replicates
numpy.random.seed(init_random_seed)
numpy.random.shuffle(code_combos)

for code in code_combos:
    epsilon = epsilons[code[0]]
    beta_naught = beta_naughts[code[1]]
    alpha_p = alpha_ps[code[2]]
    alpha_c = alpha_cs[code[3]]
    gamma_p = gamma_ps[code[4]]
    gamma_c = gamma_cs[code[5]]
    lambda_p = lambda_ps[code[6]]
    lambda_c = lambda_cs[code[7]]

    # make spreadsheet
    with open(filename, 'a') as f:
        for c in code:
            f.write(str(c) + ',')
```

```
test_domain = CuriosityBandit(random_seed=init_random_seed)
test_system = \
    MultiBrainedAgent(test_domain,
                      {'predictor':
                           {'agent_type': TDLambdaAgent,
                            'params': {'gamma': gamma_p,
                                       'alpha': alpha_p,
                                       'lambda': lambda_p,
                                       'beta_naught':
                                           beta_naught,
                                       'initial_value': 0}},
                        'control':
                           {'agent_type':
                                 WatkinsQLearningAgent,
                            'params': {'gamma': gamma_c,
                                       'alpha': alpha_c,
                                       'epsilon': epsilon,
                                       'lambda': lambda_c,
                                       'beta_naught':
                                           beta_naught,
                                       'initial_value': 0}},
                            'control_reward': lambda agent:
                            agent.UDE_keeper.get_output()})
test = Test(test_domain, num_steps)
test.run(test_system, test_domain.get_initial_state())

with open(filename, 'a') as f:
    for action in test_domain.get_action_set():
        f.write(str(float(test.count[0][action])/num_steps)
               + ',')
    f.write(str(test.return_sum) + '\n')
```

Appendix C

Curiosity Ring World

While the Curiosity Bandit problem is designed to showcase an agent's ability to differentiate the interestingness of different rewards, we would also like to showcase an agent's ability to differentiate the interestingness of changes in *state*, or state-action trajectories. Because, as described in Section 2.3, different approaches use different internal constructions, the types of changes in the world that they react to will be different.

The second environment, shown in Figure C.1, is designed to showcase approaches that allow agents to differentiate the interestingness of changes in state, or state-action trajectories.

The choice of five states is somewhat arbitrary. As long as there are at least two states, the same notions of *constant*, *random*, and *sinusoid* can be included as actions. Intuitively, the 'sinusoidal' pattern becomes more and more difficult for an agent to see, if there are more states in the ring.

The number of states could vary as a parameter to the environment. If we look to have an analogous parameter in the curiosity bandit, we could vary the period of the function for the sinusoidal arm's reward. We may later want to modularly combine rings of different sizes to see what size of ring is most preferable to a curious agent or if it developmentally prefers rings of increasing size.

Another possible variation to this decision process could be the addition of an action which does not move the agent, that is, that takes the agent from s_i to s_i; it is not necessarily clear whether this action would (should?) be more 'boring' to a curious agent than the one we have defined as 'boring.'

In the ring-world, not only do we want to know about which actions are taken, but from where the agent seems to find them most valuable at any given time.

In Section 2.3.3.1, we described the Simple Simulated Robot Experiment by Oudeyer et al. (2007). As demonstrated by Figure 2.3, the Learning Progress agent implemented by Oudeyer et al. (2007) spent a non-negligible length of time

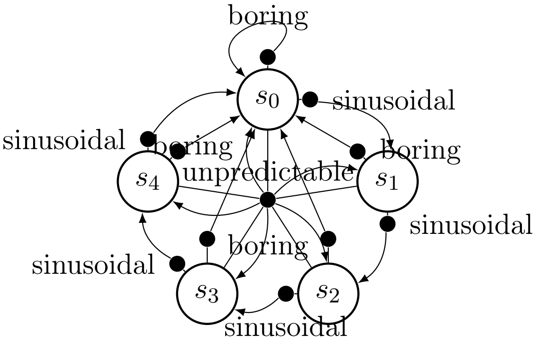


Figure C.1: The states and actions of the Curiosity Ring World and their dynamics. The decision process shown has five states and the same three actions can be taken from each state. We omit reward from the specification, but we can also simply assume that every action deterministically results in a reward of 0. For convenience, the unpredictable action, though it can be taken from every state, is only shown once, since the result of that action always has the same probabilities. The 'unpredictable' action takes the agent to any of the five states with equal probability. The 'boring' action deterministically returns the agent to s_0. The 'sinusoidal' action progresses the agent around the ring, *i.e.,* from state s_i it takes the agent to $s_{i+1 \bmod 5}$.

focused on the simplest tone frequency, possibly due to there still being substantial exploration space along with the simplest tone frequency. In contrast, after taking the *constant* action once, a learner faced with the Curiosity Bandit will learn essentially nothing new. In the ring-world, on the other hand, the agent is offered the complexity of multiple states from which to try the analogous *constant* action, so we might observe a progression more similar to that observed with the Simple Simulated Robot Experiment.

We also noted that neither empowerment, by Klyubin et al. (2005), nor Predictive Power, by Still and Precup (2012) have any mechanism to vary behaviour based on reward, or any simple signal excluded from the state. In the ring-world, on the other hand, we expect that both empowerment and Predictive Power will differentiate the different states and actions.

The simple translation of the approach designed by White et al. (2014) that we used for the Curiosity Bandit—simply computing the surprise regarding the reward and using the surprise as the intrinsic reward—cannot be expected to result in interesting behaviours in the ring-world. However, we could consider alternative translations such as providing the state signal to the agent as a 5-dimensional vector $\vec{v}$ with binary values

$$v_i = \begin{cases} 1 & \text{if the agent is in } s_i \\ 0 & \text{otherwise.} \end{cases} \tag{C.1}$$

The agent could predict these values for different policies, allowing us to compute the surprise regarding these patterns. A variation on this would be to use the reward value as a sixth dimension. We might expect to observe 'projects,' as we called the behaviour observed by White et al. (2014) in Chapter 2, alluding to Simpson's terminology for behaviour exhibited by rhesus monkeys (1976, p. 385).