EPFL

Thèse n° 10 399

Surprise-based model estimation in reinforcement learning: algorithms and brain signatures

Présentée le 11 mars 2021

Faculté des sciences de la vie
Laboratoire de calcul neuromimétique (SV/IC)
Programme doctoral en neurosciences

pour l'obtention du grade de Docteur ès Sciences

par

Vasiliki LIAKONI

Acceptée sur proposition du jury

Prof. K. Hess Bellwald, présidente du jury
Prof. W. Gerstner, Prof. K. Preuschoff, directeurs de thèse
Prof. D. Ostwald, rapporteur
Prof. E. Koechlin, rapporteur
Prof. A. Mathis, rapporteur

École polytechnique fédérale de Lausanne

2021

It is going dark.
There are bombs exploding.
Alice is losing patience.
She throws down the map and shouts at Gertrude:
"This is the wrong road!"
Gertrude drives on.

"Right or wrong,
this is the road and we are on it."

— Jeanette Winterson, *Why be happy when you could be normal?*,
rephrasing a passage from
Gertrude Stein, *The autobiography of Alice B. Toklas.*

To Eva, my sister
$\Sigma'\epsilon\upsilon\chi\alpha\rho\iota\sigma\tau\acute{\omega}$.

Acknowledgements

I would like to, first of all, thank my advisor Wulfram Gerstner and my co-advisor Kerstin Preuschoff for giving me the opportunity to pursue my PhD in such a unique and stimulating environment. Wulfram's passion for science, broad view, and ability to combine things and "see" what lies behind facts are truly inspiring and motivating. The freedom he provided and his always constructive feedback helped me grow professionally and personally. I am grateful to Kerstin for her help and trust, in particular during back when I started as an intern in the lab, for always believing in me and for trying – often intensely – to convince me about my potential. I would also like to thank my mentor Dimitri Van De Ville for his kind advice and support.

Doing a PhD is often a lonely and disheartening process. But I had the luxury to be surrounded by talented, kind and interesting people, to have support during the "I want to quit" moments, and to form fruitful collaborations. I feel deeply indebted to Johanni Brea and Alireza Modirshanechi, thanks to whom my PhD got an out-of-my-comfort-zone twist and became the enriching learning experience I was looking for. I warmly thank Johanni for his stimulating, restless scientific curiosity, his invaluable help and insights, his encouragement, as well as for introducing me to the beautiful world of the julia language and pgfplots. I am extremely thankful to Alireza for a great and smooth collaboration based on honesty and mutual help, for always being there to discuss and help, for his contagious scientific passion and rigour, and for his friendship. I am also deeply grateful to Marco Lehmann for a great collaboration, for his help and support, for numerous discussions about all sorts of topics, and for uncountable moments of laughter in Lausanne and elsewhere in conferences. I couldn't thank enough Samuele Muscinelli, whose friendship is tightly associated with my whole PhD experience, and whose enthusiastic nature has always brought people together and set a cheerful tone. I thank him for his advice, his professional and psychological support, for listening and caring, and for all the fun moments, including some movie scripting and shootings, that had not really been in my list of things to do in life originally.

I am thankful to all the aforementioned people, also for their careful reading of many parts of the thesis and for their valuable feedback on it.

Working in the LCN has been an amazing experience both in a scientific and a social level. For interesting discussions and feedback, as well as for nice times at Sat, at SV happy hours and in the town, I would like to thank all past and present lab members I have overlapped with: Moritz Deger, Friedemann Zenke, Christian Pozzorini, Carlos Stein

(also for his great advice), Lorric Ziegler, Laureline Logiaco, Mohammadjavad Faraji, Hesam Setareh, Alex Seeholzer, Aditya Gilra, Olivia Gozel, Dane Corneil, Ho Ling Li, Wilem Wybo, Marco Martinolli, Tilo Schwalger, Valentin Schmutz, Noé Gallice, Martin Barry, Florian Colombo (also for his ski-weekend cooking), Bernd Illing (also for the taxi-bike rides and the late night walking home discussions), Chiara Gastaldi, Guillame Belec, Christos Soumpris, Georgios Iatropoulos and Berfin Simcek (also for her lively attitude towards science and life). Many thanks also go to Leyla Loued-Khenissi for her help with fMRI scanning, and to Maya Jastrzebowska, Thomas Bolton and Jonas Richiardi for useful discussions.

Outside the lab, I am grateful to my friends and flatmates Rebekka Park and Frantisek Dlabac (and since recently Adam Eördögh) for their support and for great evenings with food, wine, and games at home. I am also grateful to Laura Bless for her valuable advice and support. I would like to thank all my friends from my master's, who made Lausanne the most beautiful place in the world, and especially Varun Sharma, who somehow placed the initial seed of the idea to pursue a PhD.

I am very grateful to my parents, Dora and Thanasis, who supported me during the beginning of my studies and allowed me, thus, to be where I am now, as well as for not doubting my choices, despite my work sounding often a bit vague. I feel incredibly grateful to my sister Eva, who always has the right words to make any problem seem lighter and who is my favourite human being in this world.

Last, but not least, I sincerely thank my boyfriend Leandre (also known to some as "Bob") for always being my shelter of support and calmness, as well as for all the cooking these last days before submission.

Lausanne, 31 August 2020 V. L.

Abstract

Learning how to act and adapting to unexpected changes are remarkable capabilities of humans and other animals. In the absence of a direct recipe to follow in life, behaviour is often guided by rewarding and by surprising events. A positive or a negative outcome influences the tendency to repeat some actions, and a sudden unexpected event signals the possible need to act differently or to update one's view about the world. Advances in computational, behavioral and cognitive neuroscience have indicated that animals employ multiple strategies to learn from interaction. However, our understanding of learning strategies and how they may be combined is still largely restricted. The main goal of this thesis is to study the use of surprise by ever-adapting biological agents, its contributions to reward-based learning, and its manifestation in the human brain.

We first study surprise from a theoretical perspective. In a probabilistic model of changing environments, we show that exact and approximate Bayesian inference give rise to a trade-off between forgetting old observations and integrating them with new ones, modulated by a naturally emerging surprise measure. We develop novel surprised-based algorithms that can adapt in the face of abrupt changes and accurately estimate the model of the world, and that could potentially be implemented in the brain.

Next, we focus on the contributions of surprise-based model estimation to reinforcement learning. We couple one of our adaptive algorithms as well as simpler non-adaptive methods with reinforcement learning agents and evaluate their performance on environments exhibiting different characteristics. Abrupt changes that directly affect the agent's policy call for surprise-based adaptation, in order to achieve higher performance. Often, however, the agent does not need to invest in maintaining an accurate model of the environment to obtain high reward levels. More specifically, in stochastic environments or in environments with distal changes, simpler methods, equipped with exploration capacity, perform equally well compared to more elaborate methods.

Finally, we turn to human learning behaviour and brain signals of surprise- and reward-based learning. We design a novel sequential decision making task of multiple steps where strategic use of surprising events allows us to dissociate fMRI brain correlates of reward learning and model estimation. We show that Bayesian inference on this task leads to the same surprise measure we found earlier, where the trade-off is now between ignoring new observations and integrating them with the old belief, and we develop reinforcement learning algorithms that perform outlier detection via this surprise-modulated trade-off. At the level of behaviour we find evidence for a model-free policy learning architecture,

with potential influences from a model estimation system. At the level of brain responses we identify signatures of both reward- and model estimation signals, supporting the existence of multiple parallel learning systems in the brain.
This thesis presents a comparative analysis of surprise-based model estimation methods in theory and simulations, provides insights in the type of approximations that biological agents may adopt, and identifies signatures of model estimation in the human brain. Our results may aid future work aiming at building efficient adaptive agents and at understanding the learning algorithms and the surprise measures implemented in the brain.

Keywords: Reinforcement learning, adaptive learning, surprise, human learning, sequential decision making, behaviour, brain imaging, fMRI.

Résumé

Apprendre à agir et s'adapter à des changements inattendus sont des capacités remarquables des humains et des autres animaux. En l'absence d'une recette directe à suivre dans la vie, le comportement des êtres vivants est souvent guidé par des événements gratifiants et surprenants. Un résultat positif ou négatif influence la tendance à répéter certaines actions, et un événement soudain et inattendu signale la nécessité éventuelle d'agir différemment ou d'actualiser sa vision du monde. Les progrès des neurosciences computationnelles, comportementales et cognitives ont montré que les animaux utilisent de multiples stratégies pour apprendre des interactions. Cependant, notre compréhension des stratégies d'apprentissage et de la façon dont elles peuvent être combinées est encore largement limitée. Le but principal de cette thèse est d'étudier l'utilisation de la surprise par des agents biologiques en constante adaptation, les contributions de la surprise à l'apprentissage basé sur la récompense, et sa manifestation dans le cerveau humain.

Nous étudions d'abord la surprise d'un point de vue théorique. Dans un modèle probabiliste d'environnements changeants, nous montrons que l'inférence bayésienne exacte et approximative donne lieu à un compromis entre l'oubli d'anciennes observations et leur intégration à de nouvelles, modulées par une mesure du degré de surprise qui émerge naturellement. Nous développons de nouveaux algorithmes basés sur la surprise qui peuvent s'adapter face à des changements brusques et estimer avec précision le modèle de l'environment, et qui pourraient potentiellement être mis en œuvre dans le cerveau.

Ensuite, nous nous concentrons sur les contributions de l'estimation de modèles basés sur la surprise à l'apprentissage par renforcement. Nous couplons l'un de nos algorithmes adaptatifs ainsi que des méthodes non adaptatives plus simples avec des agents d'apprentissage par renforcement et évaluons leurs performances sur des environnements présentant des caractéristiques différentes. Les changements brusques qui affectent directement la politique de l'agent exigent une adaptation basée sur la surprise, afin d'obtenir de meilleures performances. Souvent, cependant, l'agent n'a pas besoin d'investir dans le maintien d'un modèle précis de l'environnement pour obtenir des niveaux de récompense élevés. Plus précisément, dans les environnements stochastiques ou dans les environnements présentant des changements distaux, les méthodes plus simples, dotées d'une capacité d'exploration, sont tout aussi performantes que les méthodes plus élaborées.

Enfin, nous nous tournons vers le comportement de l'apprentissage humain et les signaux cérébraux d'un apprentissage basé sur la surprise et la récompense. Nous concevons une nouvelle tâche de prise de décision séquentielle en plusieurs étapes où l'utilisation

stratégique d'événements de surprise nous permet de dissocier les corrélats cérébraux de l'IRMf de l'apprentissage par récompense et de l'estimation du modèle. Nous montrons que l'inférence bayésienne sur cette tâche conduit à la même mesure du degré de surprise que nous avons trouvée plus tôt, où le compromis est maintenant entre l'ignorance des nouvelles observations et leur intégration à l'ancienne croyance, et nous développons des algorithmes d'apprentissage par renforcement qui effectuent la détection des aberrations via ce compromis modulé par la surprise. Au niveau du comportement, nous trouvons des preuves d'une architecture d'apprentissage des politiques sans modèle, avec des influences potentielles d'un système d'estimation de modèle. Au niveau des réponses du cerveau, nous identifions des signatures de signaux de récompense et d'estimation de modèle, ce qui confirme l'existence de multiples systèmes d'apprentissage parallèles dans le cerveau. Cette thèse présente une analyse comparative des méthodes d'estimation de modèle basées sur la surprise en théorie et en simulation, fournit des indications sur le type d'approximations que les agents biologiques peuvent adopter et identifie les signatures d'estimation de modèle dans le cerveau humain. Nos résultats peuvent aider les travaux futurs visant à construire des agents adaptatifs efficaces et à comprendre les algorithmes d'apprentissage et les mesures du degré de surprise mises en œuvre dans le cerveau.

Mots-clés : Apprentissage par renforcement, apprentissage adaptatif, surprise, apprentissage humain, prise de décision séquentielle, comportement, imagerie cérébrale, IRMf.

Contents

List of Figures

List of Tables

1 Introduction

"Life is sequential decision making and learning under uncertainty and risk". This is a quote from a talk of Máté Lengyel at the "Computational and Systems Neuroscience conference" (Cosyne) in 2017, that stayed in my mind. Indeed, few cognitive processes are as central in the lives of humans and other animals as the ability to learn and to adapt their future decisions. When a kitten opens it eyes and its visual system learns to recognize objects, or when a baby makes its first steps, when we avoid eating a suspicious-looking fruit in the fridge since a similar one made us ill in the past, up to when a trained clinician makes a diagnosis that saves someone's life, these are all examples of the fundamental role of the brain processes that allow the incorporation of experience to guide behaviour.

Sometimes there are clear instructions, direct feedback, or some form of a "teacher" that guide learning. But in many other cases feedback comes very late or is not explicit. For example, when we learn to ride a bike, there is a series of movements that may eventually lead to a fall, and it is hard to know what we should do differently. Through the observation of positive and negative outcomes (rewards) of our repeated efforts, we slowly learn how to keep our balance. And in other cases, learning occurs by observing patterns in the world and their violation. Feedback is, then, often implicit for the "hidden" process we try to understand. For example, if we start encountering traffic jams on our usual route to work, we are surprised. As a consequence, we may infer that there is construction work on some alternative route that causes the increased traffic, and adjust our schedule accordingly. Our understanding of how the various types of learning are implemented in the brain is quite limited. This thesis presents research on these two latter types of learning; learning from reward and from surprise.

We, first, investigate surprise and surprise-modulated learning from a theoretical perspective. We develop surprise-modulated algorithms both for building a model of the world and for obtaining reward in non-stationary environments, and study their behaviour and performance by means of simulations. We then present an experiment where we study signatures of different learning methods in the behaviour and in blood oxygenated

levels (BOLD) responses of human participants, during a reward-based sequential decision making task.

In this introduction, I first provide an overview of the theory of Reinforcement Learning, with particular focus on model-free and model-based reinforcement learning. Next, I briefly review the theoretical background of surprise and surprise-based learning, as one component of model learning. Then I present a short review of findings that have bridged these theories with neuroscience: behavioural and neural evidence for the implementation of these types of learning in the brain. Finally, I conclude with a comment on the shared nature and neural mechanisms between reward- and surprise-based learning and on the open issues this thesis aims at contributing to. A more detailed review of previous studies on the specific topic of each chapter will be provided at the Introduction or Discussion sections of the corresponding chapter, in a more targeted manner. The aim of this introduction is to lay out the general background and the fundamental concepts on which our work is based.

1.1 Reinforcement Learning theory

Reinforcement Learning (RL) (Sutton and Barto, 1998) is the mathematical formulation of learning from delayed feedback – reward or punishment – that has had an extremely influential impact in studying and understanding learning behaviour. The most widely used starting point for the framework of RL is viewing the world as a Markov Decision Process (MDP). In an MDP the world is made of a set of possible states $\mathcal{S}$, a set of possible actions $\mathcal{A}$ that the – biological or artificial – agent can perform, and a set of possible reward (scalar) values $\mathcal{R} \subset \mathbb{R}$, that the agent can receive. On a given time step t the agent is at (or observes) a state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$, and, on the next time step, transits to a state $s_{t+1} \in \mathcal{S}$ and possibly receives some reward $r_{t+1} \in \mathcal{R}$ (Fig. 1.1).

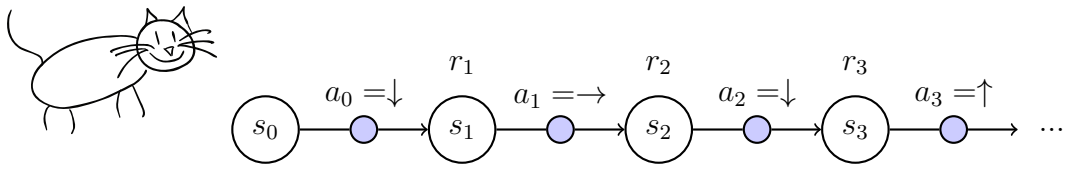


Figure 1.1 – **Markov decision process (MDP) and agent.** The interaction between the agent and the environment, modelled as an MDP, gives rise to a sequence of states, actions, and rewards. In this simple example there are four available actions: up $\uparrow$, down $\downarrow$, left $\leftarrow$, and right $\rightarrow$. At every time step, from a state $S_t = s_t$, and upon the selection of an action $A_t = a_t$, the agent transits to a state $S_{t+1} = s_{t+1}$ and observes some reward $R_{t+1} = r_{t+1}$.

The MDP is characterized by its dynamics: the probability distribution of observing the

state s_{t+1} and reward r_{t+1} at the time step $t+1$, when choosing a_t from s_t at time t, i.e.

$$p(s_{t+1}, r_{t+1}|s_t, a_t) \doteq \mathbf{P}(S_{t+1} = s_{t+1}, R_{t+1} = r_{t+1}|S_t = s_t, A_t = a_t), \tag{1.1}$$

where capital letters indicate random variables and small letters indicate values, and $\mathbf{P}$ stands either for probability mass function (for the discrete variables, which is the case here), or for probability density function (for the continuous variables). The next state s_{t+1} and the reward r_{t+1} depend only on the current state s_t and action a_t, and not on the whole history of states and actions, i.e. the dynamics assume the Markov property $p(s_{t+1}, r_{t+1}|s_{1:t}, a_{1:t}) = p(s_{t+1}, r_{t+1}|s_t, a_t)$.

From these dynamics one can define the so-called *transition matrix* [1]

$$T(s_t, a_t, s_{t+1}) \doteq \mathbf{P}(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t) = \sum_{r\in\mathcal{R}} p(s_{t+1}, r|s_t, a_t), \tag{1.2}$$

that is the probability to transit to s_{t+1} from s_t when choosing a_t, i.e. a three-argument function $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, as well as the *reward function*

$$\begin{aligned}\bar{R}(s_t, a_t, s_{t+1}) &\doteq \mathbb{E}[R_{t+1}|S_t = s_t, A_t = a_t, S_{t+1} = s_{t+1}] \\ &= \sum_{r\in\mathcal{R}} r\mathbf{P}(R_{t+1} = r|S_t = s_t, A_t = a_t, S_{t+1} = s_{t+1}) \\ &= \sum_{r\in\mathcal{R}} r\frac{p(s_{t+1}, r|s_t, a_t)}{T(s_t, a_t, s_{t+1})},\end{aligned} \tag{1.3}$$

that is the reward that is expected to be received when the agent was at s_t, chose a_t, and transitioned to s_{t+1}. $\bar{R}$ is defined here as a three-argument function, i.e. $\bar{R} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$, but oftentimes it is a function of the landing state only.

The goal of an agent is to maximize the total reward that he or she will receive from the environment in the future. Sometimes rewards that occur very far off in the future are less important. The quantity that takes into account this subjective weighting between immediate and long-term rewards is the *return*, i.e. the discounted total future reward calculated from time step t onwards

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{1.4}$$

The parameter $\gamma \in [0, 1]$ is called *discount rate* and discounts the importance of distant in time rewards.

The agent, thus, seeks to find the behaviour or the policy that maximizes the above discounted sum in expectation. A policy is defined as the probabilistic mapping from

[1] Throughout this thesis we assume discrete rewards. If rewards are continuous quantities, all sums over rewards in the following equations should be replaced by integrals.

states to actions, i.e.

$$\pi(s_t, a_t) \doteq \mathbf{P}(A_t = a_t | S_t = s_t). \tag{1.5}$$

The policy that fulfils the goal of the agent, i.e. that maximizes the expected discounted sum G_t is called "optimal policy" and is denoted as π^*.

There are several algorithms in order to find the optimal policy and we will briefly review them. Some of them achieve this goal, through the calculation of values. *Value* is a core concept in RL, and quantifies the "goodness" of selecting a certain action from a certain state. The state-action value $Q^\pi(s, a)$ given a policy π is defined as

$$Q^\pi(s, a) \doteq \mathbb{E}_\pi[G_t | S_t = s, A_t = a], \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}, \tag{1.6}$$

and expresses the expected future discounted reward when the agents chooses a at s and follows the policy π thereafter. Sometimes it is useful to think of the values of being in a certain state, and not of a state and an action jointly, in which case we have

$$\begin{aligned} V^\pi(s) &\doteq \mathbb{E}_\pi[G_t | S_t = s] \\ &= \sum_a \pi(s, a) Q^\pi(s, a), \text{ for all } s \in \mathcal{S}. \end{aligned} \tag{1.7}$$

Equation 1.6 can be re-written in the following way

$$\begin{aligned} Q^\pi(s, a) &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \sum_{s'} \mathbf{P}(s' | s, a) \big[\sum_r r \ \mathbf{P}(r | s, a, s') + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \big] \\ &= \sum_{s'} T(s, a, s') \big[\bar{R}(s, a, s') + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \big] \\ &= \sum_{s'} T(s, a, s') \big[\bar{R}(s, a, s') + \gamma V^\pi(s') \big]. \end{aligned} \tag{1.8}$$

This recursive self-consistent expression is a fundamental property of values, and it is called the *Bellman equation*. If we spell out this equation in words, it means that the "goodness" of a state and action pair is equal to the immediate reward we expect to receive at the next step, plus the "goodness" of the state we expect to land on, discounted by one factor γ. The Bellman equation for the V values can similarly be written as

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} T(s, a, s') \big[\bar{R}(s, a, s') + \gamma V^\pi(s') \big]. \tag{1.9}$$

The optimal policy π^*, which is better than or equally good to all other policies is the

one associated with the optimal value function Q^*, where

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) = \sum_{s'} T(s,a,s')\big[\bar{R}(s,a,s') + \gamma V^*(s')\big], \tag{1.10}$$

where $V^*(s') = \max_{a'} Q^*(s',a')$. Equation 1.10 is called the "Bellman optimality equation".

If the dynamics of the environment $p(s_{t+1}, r_{t+1}|s_t, a_t)$ are known, then the set of equations like the one of Equation 1.10 for all state and actions in the environment, form a set of non-linear equations, that one could solve. After having the solution Q^*, one automatically knows the optimal policy π^*; the optimal thing to do is to be greedy with respect to $Q^*(s,a)$, i.e. choose the action a for which $Q^*(s,a)$ is maximal (or, if there is not only one best action, randomly select one among the equally best ones).

However, in most cases, such as in real life, we often do not know the dynamics of the environment. In fact, even if we knew them, solving the set of Bellman equations is often intractable when the state-action space is large (e.g. for the game of backgammon or Go, but also in less extreme cases). In the following subsections, I briefly review some RL methods that attempt to approximately solve the Bellman equation as they experience the environment. All these methods learn through interaction with the world, but employ different ways to do so.

1.1.1 Model-free reinforcement learning

Value-based learning

Equation 1.7 and Equation 1.9 mean that *after learning* the following should *in expectation* hold true

$$V(s_t) = r_{t+1} + \gamma V(s_{t+1}). \tag{1.11}$$

As long as learning has not occurred, the above equation does not hold. Some value-based RL algorithms exploit this idea and turn Equation 1.11 into an update rule, using the discrepancy between the left-hand side and the right-hand side of Equation 1.11. This discrepancy is defined as the *reward prediction error* (RPE)

$$RPE_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t). \tag{1.12}$$

As we will see later, the RPE has played an extremely important role in the neuroscience of reinforcement learning.

Using the RPE, $V(s_t)$ is then updated, at every time step, as

$$V(s_t) \leftarrow V(s_t) + \alpha\, RPE_t, \tag{1.13}$$

where $\alpha \in (0, 1]$ is the learning rate, controlling the amount of the update.

Equation 1.12 and Equation 1.13 are the simplest form of the *temporal-difference* (TD) algorithms. The name of these algorithms comes from the fact that the updates are based on the difference in the agent's estimation between two consecutive time points; the predicted $V(s_t)$ at time t and the experienced or new estimation $r_{t+1} + \gamma V(s_{t+1})$ at time $t + 1$.

TD algorithms belong to the category of *model-free* RL algorithms. They try to approximate the Bellman equation by sampling and averaging over the experienced reward at each step, without explicitly learning the dynamics of the environment, such as the transition matrix T.

A TD algorithm, commonly used in human learning studies, is the SARSA algorithm (Sutton and Barto, 1998)

$$\begin{aligned} RPE_t &= r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \\ Q(s_t, a_t) &\leftarrow Q(s_t, a_t) + \alpha\, RPE_t, \end{aligned} \tag{1.14}$$

and a closely related and famous TD algorithm is the Q-learning algorithm (Watkins and Dayan, 1992; Watkins, 1989)

$$\begin{aligned} RPE_t &= r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \\ Q(s_t, a_t) &\leftarrow Q(s_t, a_t) + \alpha\, RPE_t. \end{aligned} \tag{1.15}$$

The name "SARSA" comes from the fact that the update depends on the tuple (S)tate - A(ction) - R(eward) - next S(tate) - next A(ction). In Q-learning, on the other hand, the update is based on the current estimate of the value of the next best – greedy – action ($\max_{a'} Q(s_{t+1}, a')$), rather than the value of next action chosen according to the agent's policy ($Q(s_{t+1}, a_{t+1})$). Hence, Q-learning directly approximates the optimal Q^* independently of the policy the agent follows, whereas SARSA estimates Q^π for a policy π. In order to find the optimal policy π^* with SARSA, the policy π should be changing according to the estimated Q values, and the convergence of SARSA to the optimal policy depends on the relationship between policy and Q values. The fact that SARSA takes into account the agent's policy in its updates leads often to better online performance – while the agent is learning and has not converged yet – to lower variance across updates and to faster convergence (Sutton and Barto, 1998).

In general, the convergence of model-free RL algorithms to the optimal policy π^* can theoretically be ensured, given infinite number of visits of all state-action pairs in the environment, and under other appropriate conditions (e.g. decreasing learning rate α) (Singh et al., 2000; Sutton and Barto, 1998; Watkins and Dayan, 1992). The requirement of "infinite" number of visits in the previous statement maybe makes already apparent the

problem, or the dilemma, an agent is faced with: the goal is to maximize the obtained reward, i.e. find the optimal policy π^*, but to do so one needs to fully explore the environment. That is, the agent needs to spend time being suboptimal, and possibly obtain less reward, in order to eventually be optimal. This is called the *exploration-exploitation* dilemma, which is an active field of research with many open questions. Typically, the policy that is employed is an ϵ-greedy policy, that is choosing the action $a = \text{argmax}_a Q(s,a)$ with probability $1-\epsilon$ and another action with probability ϵ, or a softmax policy with respect to the Q values, i.e.

$$\pi(s,a) = \frac{e^{Q(s,a)/\tau}}{\sum_b e^{Q(s,b)/\tau}}, \tag{1.16}$$

with a temperature parameter τ.

According to Equation 1.13, Equation 1.14 and Equation 1.15, at each time step, the state that was just visited is updated in the light of new experience. However, when something good happens, for example when we finally get an ice cream, it is not only the state and action just before that led to it (entering the ice cream shop), but possibly a whole series of previous states and actions (turning left at the previous block to reach the shop). In other words, the *credit assignment* should be done to the history of choices as well, and not only to the last one before the reward.

A mechanism to achieve this is the use of *eligibility traces*. Eligibility traces are decaying memory traces of previous states and actions. Upon the receipt of the reward, all states and actions in the memory trace are updated and reinforced in a weighted fashion that depends on the time lapsed since they were experienced. For example, in the SARSA algorithm with eligibility traces, i.e. SARSA-λ, the calculated RPE of Equation 1.14 is used in the following way[2] (Sutton and Barto, 1998)

$$\begin{aligned} &Q(s,a) \leftarrow Q(s,a) + \alpha\, RPE_t\, e_t(s,a), \quad \forall\, s \in \mathcal{S}, a \in \mathcal{A} \\ &e_t(s,a) = \begin{cases} 1, & \text{if } s_t = s, a_t = a \\ \gamma\lambda e_{t-1}(s,a), & \text{otherwise,} \end{cases} \end{aligned} \tag{1.17}$$

where $e_t(s,a)$ are the exponentially decaying eligibility traces, for each state and action, that are initialized to zero, and decay with a decay factor $\lambda \in [0,1]$. Thus, all state-action pairs are updated at each step, according to the decaying memory of visits. This mechanism can also be applied to Q-learning in the same way, and give rise to the Q-λ algorithm. The eligibility traces accelerate learning and are relevant for biological implementations of learning (Gerstner et al., 2018) and for human behaviour (Lehmann et al., 2019).

[2] This is the version of *replacing traces* (Sutton and Barto, 1998), which we use also later in Chapter 4.

Policy learning

In all the aforementioned methods, the policy is a function of the values and the agent tries to find the optimal policy by estimating the optimal values. Other methods, e.g. the *policy gradient* methods, work directly in the policy space, without using the values as a proxy (Peters, 2010; Schulman et al., 2015; Sutton and Barto, 1998). In policy gradient mehtods, the main idea is changing the policy by performing gradient ascent on some objective measure, e.g. the expected return. The policy is still a function (e.g. softmax) parametrized by some parameters, often called *preferences*, but estimating directly the policy offers often higher flexibility; value-based methods will converge to the optimal values, whereas policy gradient will converge to the optimal stochastic or deterministic policy directly. For example, if ones uses a softmax policy and the optimal policy is deterministic, the policy preferences in the softmax are free to reach any high value, whereas the Q values will reach the true Q^* values, and may thus still allow for some stochastic behaviour. Moreover, an often disregarded feature of policy gradient is that, in most of its versions, the policy preferences from a particular state are not independent and jointly estimated, whereas in value-based learning, the value of each (s, a) pair is estimated independently. For example, learning that we should select a certain action usually means that we should not select the other available actions, hence, this feature may add to the flexibility of these methods. The simplest example of a policy gradient method is the REINFORCE algorithm (Williams, 1992) that at each step updates the policy preferences $p(s, a)$ of all preceding decisions with gradient ascent using the return G_t (see Chapter 4 subsection 4.4.4 for more details).

Some policy gradient methods do calculate values as well, because this offers better properties during learning, such as reduced variance across updates. One such algorithm is the Actor-critic algorithm (Sutton and Barto, 1998), where one "compartment" of the algorithm, called the *critic*, estimates V values and another compartment, called the *actor* estimates the policy preferences. The RPE calculated by the critic is inserted to the actor and influences its updates (see Chapter 4 subsection 4.4.4 for more details).

1.1.2 Model-based reinforcement learning

Model-free RL is appealing due to its simplicity and has played a crucial role in understanding animal and human learning. It is however slow and sample-inefficient, i.e. requires a very large number of state-action visits. Furthermore, it is agnostic to the topology between states, which makes it inflexible to changes. In other words, all that a model-free agent learns is a value or a policy landscape with respect to reward. After learning, the agent still does not know about how states are related to each other, beyond their relation with respect to the reward. Thus, if, for example, the reward changes location, a model-free agent needs to start learning from scratch.

A family of algorithms with better features in this respect are the *model-based* (MB) reinforcement learning algorithms. In model-based RL, the agent learns explicitly the model of the world, i.e. estimates the transition matrix T and the reward function $\bar{R}$, and then directly solves the Bellman equation (Equation 1.10). As we saw, the value of each state depends on its successors, thus this is not an easy calculation to do. One way to find the solutions to the Bellman optimality equation is *value iteration*, that is, at each time step, going through the whole state-action space many times and repeatedly calculating all Q values, until a stationary solution is reached (no significant change from one repetition to the next). This is, however, computationally expensive. Other approximate methods, such as Prioritized Sweeping (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013) mitigate the computational cost by updating only the (s, a) pairs whose predecessors changed more than some priority threshold.

One example of such a MB algorithm is the Forward Learner (Daw et al., 2011a; Gläscher et al., 2010), where the transition matrix is estimated via the *state prediction error* (SPE) defined as

$$SPE_t = 1 - \hat{T}(s_t, a_t, s_{t+1}), \tag{1.18}$$

where $\hat{T}$ is the agent's estimate of the true transition matrix T. The above equation could be roughly interpreted as the difference between the experienced certainty of the occurrence of s_{t+1} and the agent's current estimated transition probability. Then the transition matrix is updated via a δ-rule in the following way

$$\begin{aligned} \hat{T}(s_t, a_t, s') &\leftarrow \hat{T}(s_t, a_t, s') + \alpha\, SPE_t, \quad \text{if } s_{t+1} = s' \\ \hat{T}(s_t, a_t, s') &\leftarrow \hat{T}(s_t, a_t, s') - \alpha\, \hat{T}(s_t, a_t, s'), \quad \text{otherwise,} \end{aligned} \tag{1.19}$$

where $\alpha \in (0, 1]$ is a learning rate. The SPE is a signal of important relevance in studying human learning, as we will discuss later.

Another way to estimate the transition matrix is via incrementally counting (Perfect Integration) the number of times state-action pairs and transition to other states were experienced (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013)

$$\begin{aligned} N(s_t, a_t, s_{t+1}) &\leftarrow N(s_t, a_t, s_{t+1}) + 1, \\ N(s_t, a_t) &\leftarrow N(s_t, a_t) + 1, \\ \hat{T}_t(s_t, a_t, s_{t+1}) &= \frac{N(s_t, a_t, s_{t+1})}{N(s_t, a_t)}, \end{aligned} \tag{1.20}$$

where $N(s_t, a_t, s_{t+1})$ is the number of times the agent transitioned to s_{t+1}, from s_t after taking a_t, and $N(s_t, a_t)$ is the number of times the agent took the action a_t from state s_t.

The reward function $\bar{R}$ can be estimated from experience in a similar spirit through

averaging the immediate reward values r_{t+1} observed

$$\begin{aligned} N(s_t, a_t, s_{t+1}) &\leftarrow N(s_t, a_t, s_{t+1}) + 1, \\ R^{sum}(s_t, a_t, s_{t+1}) &\leftarrow R^{sum}(s_t, a_t, s_{t+1}) + r_{t+1}, \\ \hat{\bar{R}}(s_t, a_t, s_{t+1}) &\leftarrow R^{sum}(s_t, a_t, s_{t+1})/N(s_t, a_t, s_{t+1}), \end{aligned} \tag{1.21}$$

where, as before, $N(s_t, a_t, s_{t+1})$ is the number of times the agent transitioned to s_{t+1}, from s_t after taking a_t, and $R^{sum}(s_t, a_t, s_{t+1})$ is the sum of the rewards received as a result of this transition.

As already mentioned, model-based algorithms learn faster, allow for planning and for adaptation to changes, but are more computationally costly. On the other hand, model-free algorithms are simple and straightforward, but slow and sample inefficient. In the next subsection we briefly describe computational approaches where model-free and model-based learning are combined and interact.

1.1.3 Hybrid reinforcement learning algorithms: model-free and model-based interaction

One of the earliest mentioned approaches to combine model-free and model-based features is the Dyna-Q algorithm (Sutton, 1990). The main idea in the Dyna architecture is that the model-based system feeds simulated ("imaginary") experience into the model-free system during offline updates. At each time step the agent updates both its Q values via Q-learning (Equation 1.15) and its model of the world (typically via Equation 1.20) based on the latest observation. At the same time, the agent randomly samples (or "imagines") possible outcomes of states and actions based on its estimated model, in a number of offline (background) steps. The agent uses then these simulated outcomes to update its Q values, again via Q-learning. Variations of Dyna-Q have been suggested, that use different algorithms for the updates, but follow the same idea of direct updating and action selection via model-free learning, and background simulating/planning via learning the model of the world (Sutton, 1990; Sutton and Barto, 2018). Notably, the original version of Prioritized Sweeping (Moore and Atkeson, 1993; Peng and Williams, 1993) was a hybrid method that can be seen as a variant of Dyna-Q, where the simulated experiences are not selected randomly, but based on how much change they would cause when updating the Q values of their preceding states.

Dyna is an example of cooperative interaction between model-free and model-based learning (Gershman et al., 2014a; Sutton and Barto, 1998). Other suggestions for a cooperative mechanism come from studies with more focus on animal and human behaviour, rather than high learning performance. A recent and influential algorithmic suggestion is the Hybrid Learner (Daw et al., 2005, 2011a; Gläscher et al., 2010) where Q values are computed as a weighted average of the SARSA Q values (cf. Equation 1.14)

and the Forward Learner Q values (cf. Equation 1.19 and Equation 1.10). Dezfouli and Balleine (2013) suggested a hierarchical organisation, where model-based learning function at a higher level on action sequences (options), and model-free learning at the level of actions, giving rise to action sequences or "habits". A hierarchical scheme has also been proposed by Cushman and Morris (2015), employing a nearly opposite organization, where the selection of the higher level option is done by the model-free system, and model-based planning is performed at the level of individual actions in order to fulfil it. In this way the model-free system helps the model-based one by reducing the search-space. An additional suggested mechanism for reducing the computational burden of the model-based system is via a (presumably serotonin-mediated) model-free system that "prunes" the planning horizon (tree) following large negative rewards, so that all the possibilities following a large loss are completely ignored during model-based computations (Huys et al., 2012).

Other approaches view the two systems as competing opponents. One suggestion has been that the two systems run in parallel, and the agent (or the brain) decides which one of the two to "trust" based on their estimation uncertainty (Daw et al., 2005; Lee et al., 2014). In this view, the two systems run in parallel and compete, and the system that is finally followed and controls behaviour is the one that is more certain about the value estimation. Other factors playing a role in the competition may be the availability of cognitive resources and amount of training. For example, human experiments have suggested that increasing the working memory load via another concurrent task may lead to the model-free system taking over (Otto et al., 2013a), but if the original task has been extensively learned then the model-based system can be resistant to the competition (Economides et al., 2015).

All the aforementioned RL algorithms, model-free value-based, model-free policy gradient, and model-based have their own strengths and weaknesses. Recently suggested hybrid approaches express the more recent thinking that multiple strategies co-exist and interact in the brain. It is still unclear which of these algorithms may be employed by animals and humans in which situations, and how they may be implemented in the brain. Before we present a review on findings and on the current understanding of the neuroscience of reinforcement learning, we will first briefly focus on computational approaches that aim at building the model of the world – and possibly use it for model-based reinforcement learning – and in particular on learning driven by signals of surprise.

1.2 Model learning

Building a model of the world, or in other words, understanding the environment's dynamics, in order to be able to make predictions about future events, is a vast topic that can be treated through the lenses of unsupervised learning, adaptive learning, Bayesian inference, change-point detection, statistics, signal processing, and other research fields. For this thesis, we focus on adaptive Bayesian or approximate online learning methods,

that are relevant in the study of how animals learn from sequences of observations. Moreover, for model building, we focus on passive inference processing, i.e. computations occurring when an agent is passively observing a sequence of events and tries to make sense of the world. Throughout the thesis, the "active" part of an agent comes from the reinforcement learning theory.

One exciting but also controversial approach to model learning is the so-called *Bayesian Brain hypothesis*, an idea traced back in the 19th century (Helmholtz, 1948). According to this idea, the brain is equipped with a generative model of the world, i.e. a set of probability distributions that relate sensory observations with hidden states or hidden causes in the world. Given this model, the brain tries to infer the hidden states and predict future events. For example, we may have some belief about how dice function, so that after a large number of throws resulting in a "6" while playing backgammon, we may start doubting the fairness of the dice (and our trust towards our opponent). A second important component of this hypothesis is that, in light of new information, the brain updates its beliefs about the world using the Bayes' rule. Formally, let us define as Θ_t some parameters or hidden states of the environment that the agent tries to estimate or infer (e.g. the probability of receiving the result "6" with the dice), and $y_1, y_2, ...y_t$ the observed stimuli (the dice marking after each throw). We then define the *belief of the agent* over the parameters

$$\mathbb{b}^{(t)}(\theta) \doteq \mathbf{P}(\Theta_t = \theta | y_{1:t}). \tag{1.22}$$

We recall that $\mathbf{P}$ is either a probability density function or a probability mass function, for continuous and discrete variables, respectively. Then, after seeing the new observation y_{t+1}, the updated posterior belief $\mathbb{b}^{(t+1)}(\theta) = \mathbf{P}(\Theta_{t+1} = \theta | y_{1:t+1})$ will be according to Bayes' rule

$$\mathbb{b}^{(t+1)}(\theta) = \frac{\mathbf{P}(y_{t+1}|\theta, y_{1:t})\mathbf{P}(\Theta_t = \theta | y_{1:t})}{\mathbf{P}(y_{t+1}|y_{1:t})}. \tag{1.23}$$

Even though the Bayesian brain hypothesis has provided account for a number of experiments (Behrens et al., 2007; Doya et al., 2007; Heilbron and Meyniel, 2019; Körding and Wolpert, 2004; Mars et al., 2008; Ostwald et al., 2012), there is still a heated debate on whether the brain is indeed a Bayesian inference machine. An important argument against this hypothesis is that in most cases, apart from toy tasks, performing inference or "inverting" the generative model, i.e. applying Equation 1.23, involves complicated intractable computations, of which the possible biological implementation is questionable (Mathys et al., 2011). Furthermore, Bayesian inference is a normative approach to the problem of model estimation; animals and humans are, however, often suboptimal and "irrational" (Glaze et al., 2015; Mathys et al., 2011; Nassar et al., 2010; Prat-Carrabin et al., 2020; Summerfield and Tsetsos, 2015; Wilson et al., 2013).

Thus, a variety of approximate Bayesian or heuristic approaches have been suggested (Faraji et al., 2018; Findling et al., 2019; Friston, 2010; Glaze et al., 2015; Mathys et al., 2011; Nassar et al., 2010, 2012; Schwartenbeck et al., 2013; Wilson et al., 2013), which have successfully explained aspects of behaviour and physiological signals. Among the simplest of the heuristic approaches, which will be important in this thesis, is Leaky Integration, namely the integration of a new observation with the agent's previous belief with a leak parameter, so that previous observations are gradually forgotten in an exponentially decaying fashion. For example, if we would like to estimate the number of times we got the observation $y_k = 6$ with the dice (denoted as $N^{(6)}$) with Leaky Integration, our online updating would be $N^{(6)}_{t+1} = [y_{t+1} = 6] + \eta N^{(6)}_t$, where $\eta \in [0, 1]$ is the leak parameter, and [.] is the Iverson bracket, which equals to 1 if the condition within the bracket is fulfilled, and 0 otherwise. Thus, at any time point t our estimation would be $N^{(6)}_t = \sum_{k=0}^{t-1} [y_k = 6]\eta^k$. Despite its simplicity and its non-adaptive nature – i.e. the parameter η is fixed in time and each observation is treated equally regardless of how unlikely it is –Leaky Integration of sensory input has successfully explained behaviour in multiple tasks (Gijsen et al., 2020; Maheu et al., 2019; Meyniel et al., 2016; Yu and Cohen, 2009) and, under certain circumstances, has been shown to approximate exact Bayesian inference (Ryali et al., 2018; Yu and Cohen, 2009).

In general, building the model of the world, that is learning the results of actions and not only the resulting values of actions (Koechlin, 2016), allows planning and can give rise to adaptive behaviour when the environment changes. This is, however, not always the case. In some situations when the estimated model is wrong and, in particular, when there are distal changes (Sutton and Barto, 2018) or when a change occurs long after the agent has converged to a solution, adapting to the change may require a long time or even never be achieved. It appears, thus, that being equipped with a model of how the environment functions may be necessary but not sufficient for adaptive behaviour. This raises the question of how a model should be estimated in order to give rise to adaptive behaviour, which is the topic of Chapter 2 and Chapter 3. A more detailed literature review on adaptive model learning algorithms is provided in the Introduction and in the Discussion sections of Chapter 2. Related work on the use of model learning for adaptive model-based reinforcement learning is discussed in Chapter 3.

A concept often evoked in adaptive model learning approaches and studies is the one of *surprise*. Yet, it is still unclear how surprise should contribute to learning and which measure of surprise is used by the brain. In the next section, we briefly present the notion of surprise and some of its mathematical definitions commonly used in neuroscience.

1.3 The notion of surprise

Surprise is a familiar psychological state occurring when something unexpected happens. More formally, surprise is a way to quantify the discrepancy between an agent's belief

about the world and reality, or the discrepancy between the agent's prediction and the observation. The SPE (Daw et al., 2011a; Gläscher et al., 2010) we saw in the previous section is actually a measure of surprise, although it was originally not defined as such.

Perhaps the most widely known measure of surprise is the Shannon surprise (Shannon, 1948). Shannon surprise comes from the field of information theory and is defined as the negative log likelihood of an event. Thus, the Shannon surprise as a result of observing y_{t+1} while the agent maintains the belief $\mathbb{b}^{(t)}$ is defined as

$$\mathbf{S}_{\mathrm{Sh}}(y_{t+1}; \mathbb{b}^{(t)}) = -\log\Big(\mathbf{P}(y_{t+1}|y_{1:t})\Big). \tag{1.24}$$

According to Shannon surprise, the more unlikely an observation is, the more surprising it will be. However, subjectively, the same event can be more or less surprising depending on our commitment to our belief. Such effects are not captured by Shannon surprise.

Another popular measure of surprise is the Bayesian surprise (Itti and Baldi, 2006; Schmidhuber, 2010; Storck et al., 1995), which measures the amount that the belief is changed after the agent has updated it in the light of new information. Intuitively, the farther away from our belief an observation lies, the more it will alter our belief about the world, and, according to Bayesian surprise, the more surprising it is. The Bayesian surprise is defined as

$$\mathbf{S}_{\mathrm{Ba}}(y_{t+1}; \mathbb{b}^{(t)}) = D_{KL}\Big[\mathbb{b}^{(t)} || \mathbb{b}_B^{(t+1)}\Big], \tag{1.25}$$

where D_{KL} is the Kullback-Liebler divergence and $\mathbb{b}_B^{(t+1)}$ is the updated belief following the Bayes' rule. Bayesian surprise, thus, entails a kind of chicken and egg problem: a surprising event will change our belief a lot, but we first need to update our belief to find out if we were surprised. It is therefore not suitable for a quantity used in online update rules, but it could potentially be used in order to measure the information that the agent gained after the update (Sun et al., 2011).

Finally, a more recent measure of surprise that includes the effect of certainty of the belief is the Confidence Corrected Surprise (Faraji et al., 2018), defined as

$$\mathbf{S}_{\mathrm{CC}}(y_{t+1}; \mathbb{b}^{(t)}) = D_{KL}[\mathbb{b}^{(t)} || \hat{p}_Y(\theta)], \tag{1.26}$$

where $\hat{p}_Y(\theta)$ is the posterior belief under a flat prior $\hat{p}_Y(\theta) = \frac{\mathbf{P}(y_{t+1}|\theta)}{\int_\theta \mathbf{P}(y_{t+1}|\theta)d\theta}$, i.e. disregarding all observations $y_{1:t}$. Intuitively, $\mathbf{S}_{\mathrm{CC}}$ compares at every step the current belief with a flat (or null) uninformative belief, and it can be shown that it includes the effect of commitment (Faraji et al., 2018). We may however have a more informed prior belief about a situation, rather than believing that everything is equally possible. $\mathbf{S}_{\mathrm{CC}}$ does not include this effect.

As we will see later, various studies have sought to identify the aforementioned surprise

measures in behaviour or in the brain. In a less mathematical and more neuroscientific perspective, surprise has been thought to serve three possible roles. First, it signals saliency and serves a mechanism of attention, so that the animal's focus is drawn towards something unexpected and possibly alarming (Fouragnan et al., 2018). Second, it may serve as a learning signal, so that the animal updates its belief accordingly and can make more accurate predictions in the future (Faraji et al., 2018; Fouragnan et al., 2018). A third, more recently identified role relates to memory formation; it has been suggested that surprise may be important for memory reactivation and reconsolidation, i.e. the process of making a memory liable to change, as well as for signalling the need for the creation of new memory (Rouhani et al., 2020; Sinclair and Barense, 2018) or the shift to a previous strategy stored in memory (Collins and Koechlin, 2012).

Among these roles, the way the second one may be mediated is particularly unclear. In many studies on learning, surprise is used as a quantity with which behavioural and neurophysiological signals are correlated with (Maheu et al., 2019; Meyniel et al., 2016; Squires et al., 1976), but it is rarely used as part of the learning algorithm (Faraji et al., 2018). In Chapter 2, we suggest a possible way that surprise may influence learning at an algorithmic level and we connect Bayesian and approximately Bayesian approaches with surprise-based modulated learning.

In the remaining sections of this introduction we turn to neuroscience and review current ideas on how reinforcement learning, adaptive learning and surprise signalling may be implemented in the brain, as well as evidence of their manifestation in behaviour and neurophysiological recordings.

1.4 The neuroscience of learning

1.4.1 Three-factor learning

At a microscopic and mesoscopic level, learning is mediated by the change of synapses, namely connections between neurons, a phenomenon called *synaptic plasticity* (Purves et al., 2004). During experience, synapses are formed, strengthened, weakened or eliminated. This spectacular capability of the brain is considered to be the basis of learning and memory formation (Martin et al., 2000).

Neurons communicate with their post-synaptic neurons through small current pulses (spikes) and through the release of neurotransmitters. A fundamental mathematical description that has shaped the way we think about synaptic plasticity and that corresponds to multiple experimental observations is *Hebbian learning* (Hebb, 1949). According to Hebbian learning rules, a synapse becomes liable (or eligible) to get strengthened if there is simultaneous occurrence of pre-synaptic and post-synaptic activity. The word "activity" here may mean the presence of neurotransmitters at the site of the synapse or on the

membrane of the post-synaptic neuron, the presence of other molecules and ions at the spine, spiking activity, or an elevated voltage (Frémaux and Gerstner, 2016; Gerstner et al., 2018).

For the case of reinforcement learning though, learning that from state s the animal should select action a, should happen only if the result of this state-action combination is a positive outcome. There is, thus, a "third" factor that affects learning. Hebbian learning has been extended to the so-called *neoHebbian* or *three-factor learning* that explains this type of learning (Barto, 1985; Lisman et al., 2011). The idea in this framework is that as, with Hebbian learning, the co-occurence of two factors sets an "eligibility trace" – which draws connections to some algorithms we saw in a previous section – for a possible change at the synapse, and the change happens only if a third factor occurs, i.e. a rewarding or surprising event is observed. The biological realization of this third factor can be various neuromodulators, such as dopamine, norepinephrine, serotonin and others (Gerstner et al., 2018; Montague et al., 1996).

We, next, describe findings on the specific instances of three-factor learning for reward- and for suprised-based learning, going also at a more macroscopic description on the role of various brain regions and on behaviour.

1.4.2 The neuroscience of reinforcement learning

Behaviour

Historically, the study of animal learning behaviour begins with an almost incidental experimental observation. Ivan Pavlov was a physiologist studying the digestive system, and in one of his experiments on dogs he observed that if a certain stimulus, e.g. a sound, preceded the appearance of food repeatedly, then dogs would react, i.e. salivate, to the sound same as they did to the food (Pavlov, 1927). That is, a physiological response to food would now occur as a response to a previously neutral stimulus, that is not related to food per se. A new association between the two stimuli was learned, and a response was now evoked to the new stimulus, indicating a reward prediction capability. This can be seen as an instance of three factor learning; the reward (food) caused a strengthening of an association between a response (salivating) and a stimulus (sound) occurring in close timing proximity. This seemingly simple experiment laid the basic principles of many later experiments on learning, memory formation and fear, nowadays called *classical conditioning* experiments.

Another important hallmark in the field of animal learning, is the work of the psychologist Edward Thorndike. He performed various behavioural experiments, where animals had to perform a series of actions to obtain reward. Based on these experiments he formulated a set of "laws" that behaviour follows (Thorndike, 1911), most importantly the learning by trial-and-error, also now called model-free learning, according to which

the association between a stimulus and an action is strengthened if the action leads to a positive outcome and weakened otherwise. His experiments formed the basis of the various later *instrumental conditioning* experiments.

Other early experiments showed that animals are able to learn the structure of mazes in the absence of reward, i.e. to build a model or a "cognitive map" of the environment (Blodgett, 1929; Thistlethwaite, 1951; Tolman, 1948). These experiments were the precursors of studying model-based learning behaviour, and seem to mark the beginning of the model-free vs model-based dichotomy in learning (Daw et al., 2005).

Neural circuitries

The mid-90's mark a breakthrough in the neuroscience of reinforcement learning with the work of R. Montague, P. Dayan, T. Sejnowski, and W. Schultz, when it was demonstrated that the activity of dopamine-releasing neurons bore a striking similarity to a model-free RPE (Montague et al., 1996; Schultz et al., 1997). In short, when a monkey received a reward that was not expected, the phasic activity of midbrain dopaminergic neurons increased. When an expected reward was not received, there was a silencing in their activity at the moment when reward was expected. This work linked for the first time the theory of reinforcement learning with neurophysiology, and it has led to the so-called *Reward prediction error hypothesis of dopamine*. This idea has proven to be very influential, since it has shaped the way we think about learning in neuroscience, but, as we will discuss later, it has also been subjected to large controversy.

Although a lot about the function of dopamine remains unknown, multiple studies over the last few decades have confirmed the important role of dopamine in (three factor) learning. Dopamine is a neuromodulator released by neurons in the ventral tegmental area (VTA) and the substantia nigra pars compacta (SNc) in the midbrain. Dopamine neurons receive inputs from multiple brain areas. It has been found that the main excitatory inputs to VTA are the lateral oribitofrontal cortex and the lateral hypothalamus, whereas SNc receives inputs from the sensorimotor cortex and the subthalamic nucleus (Watabe-Uchida et al., 2017). At the same time, ventral and dorsal structures in the basal ganglia, i.e. a set of nuclei, at the base of the forebrain (striatum, pallidum, globus pallidus, entopeduncular nucleus, and substantia nigra reticulata) are the main inhibitory inputs to dopamine neurons in VTA and SNc, respectively (Watabe-Uchida et al., 2017). All these input areas are also connected with each other, forming a complex recursive network. There is evidence supporting that dopamine neurons, receiving diverse information from multiple brain areas, combine this input information and calculate a RPE, rather than passively receiving it and relaying it (Watabe-Uchida and Uchida, 2018; Watabe-Uchida et al., 2017). In particular, it is thought that the striatum encodes the predicted value that it then provides to VTA for the calculation of the RPE (Joel et al., 2002; Watabe-Uchida et al., 2017) (Fig. 1.2A).

Dopamine neurons project, in turn, to a wide range of targets (Avery and Krichmar, 2017; Eshel et al., 2015). Depending on the target (and the post-synaptic receptors on the target) dopamine can induce different effects. The main target of dopamine neurons in SNc is the dorsal striatum, whereas for VTA is the ventral striatum and the prefrontal cortex (Avery and Krichmar, 2017).

The striatum is a particularly important structure in the reward-processing machinery. It receives input from virtually all of the cerebral cortex, thus, a variety of information. It then projects back, through other basal ganglia structures and the thalamus, to the frontal cortex and to motor areas, influencing in this way abstract learning processing, action selection and movement (Sutton and Barto, 2018). There is a gradient of various kinds of differences across the dorsal to the ventral striatum in their cytological profiles, in their input regions (from SNc and VTA for dorsal and ventral striatum, respectively), in their output regions (less and more to dopaminergic neurons, respectively), and in the functions they influence (action selection and value prediction, respectively) (Haber, 2016). These observations have led to the hypothesis that the striatum forms an actor-critic architecture (Joel et al., 2002; Takahashi et al., 2008), with the ventral striatum implementing the critic and the dorsal the actor (Fig. 1.2B).

A part of the brain extremely important in learning, adaptive decision making, goal-directed behaviour and higher cognition is the prefrontal cortex (PFC). Some subdivisions of the prefrontal cortex, with distinct roles in learning, are the ventromedial prefrontal cortex (vmPFC), the oribitofrontal cortex (OFC), the lateral prefrontal cortex (lPFC), the anterior prefrontal cortex (aPFC) and the anterior cingulate cortex (ACC) (Coutureau and Parkes, 2018; Rushworth et al., 2011; Sharpe et al., 2019). Experimental findings on the functions of different parts of the PFC are often hard to reconcile and to interpret, partially due to confusion in anatomical analogues across species (Wallis, 2012).

Perhaps the most well-studied frontal cortical area is the ventromedial prefrontal cortex. A widely accepted and replicated finding is that vmPFC activity is associated with reward and expected reward (i.e. values) (Euston et al., 2012; Rushworth et al., 2011). Evidence from various experiments has led to the suggestion that vmPFC receives as inputs and combines external (sensory) and internal (emotional) cues and events as well as previous experience, and maps them to a response (Euston et al., 2012). It therefore has an important role in goal-directed (Rushworth et al., 2011) and adaptive behaviour (Domenech and Koechlin, 2015), memory formation, recall and consolidation (Myers-Schulz and Koenigs, 2012).

A particularly mysterious region of PFC is the orbitofrontal cortex. The OFC lies in the ventral part of the frontal cortex, just above the eyes (orbits) (Murray and Rudebeck, 2018; Rudebeck and Rich, 2018). Inactivation or damage of OFC impairs decision making and control over impulsive actions, but does not absolutely diminish or prevent any function (Sharpe et al., 2019). In other words, there is virtually nothing that animals and humans

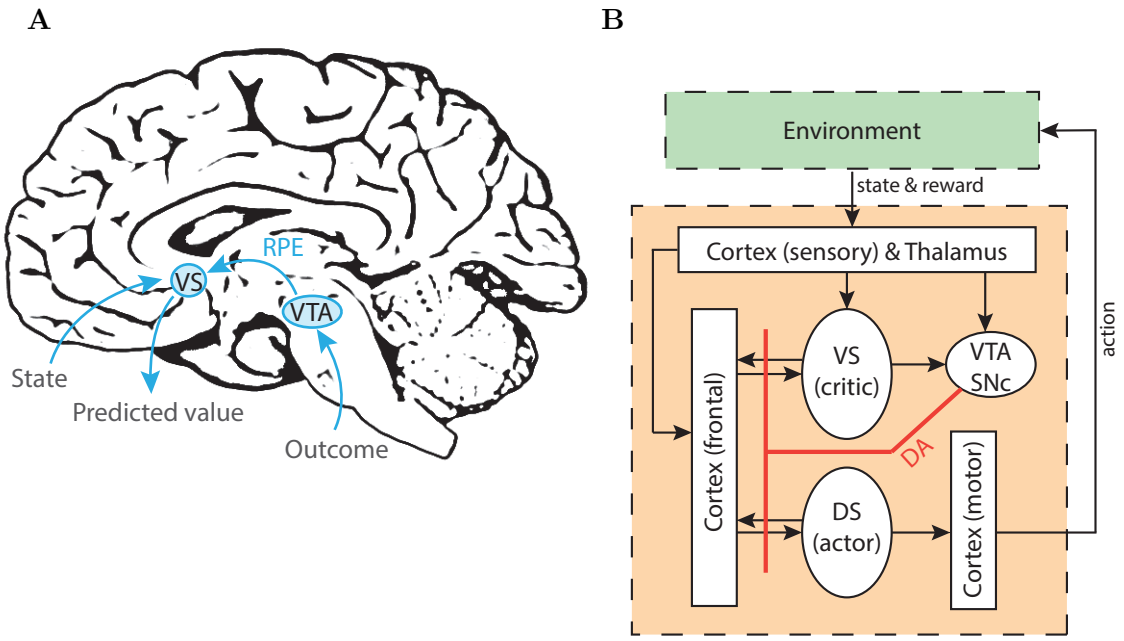


Figure 1.2 – **Neural pathways of the reward system. A**. Schematic of the implementation of model-free RL in the human brain. The reward prediction error is computed mainly by dopaminergic neurons in the ventral tegmental area (VTA) and provided as an input to the ventral striatum (VS), which in turn updates the predicted value. **B**. Hypothetical implementation of the Actor-critic architecture in the brain. The ventral striatum (VS) takes the role of the critic that calculates the values. The dorsal striatum (DS) implements the actor, that updates the policy parameters and influences action selection. Even though VS and DS are depicted as separate compartments, they belong to one single structure, the striatum. Dopamine (DA) is released by VTA, signals an RPE and is the third factor that modulates the synaptic plasticity between cortex and striatum in a three-factor learning rule. Figure based on (Takahashi et al., 2008). *Abbreviations:* VS – ventral striatum, VTA – ventral tegmental areas, DS – dorsal striatum, SNc – substantia nigra pars compacta, DA – dopamine.

cannot do without an OFC; they can still learn and erase or update previous associations, if only just more slowly and with more training (Sharpe et al., 2019; Wilson et al., 2014). The concrete role of OFC in reward-based learning is a topic of debate (Murray and Rudebeck, 2018). For example, neurons in OFC have been found to encode the value of cues (Padoa-Schioppa and Assad, 2006), but inactivation of the OFC does not prevent value-driven actions (Gardner et al., 2017). It has been suggested that OFC may be more sensitive to values of external cues, whereas the vmPFC more to values concerning internal events (Wallis, 2012). An influential theoretical suggestion that reconciles a number of puzzling experimental findings is that OFC encodes a *cognitive map* (Wilson et al., 2014). A cognitive map is a representation of the underlying structure of the task or the situation at hand, including both observed information and hidden information (e.g. task instructions, latent causes of observed events, previous actions) (Sharpe et al.,

2019; Stalnaker et al., 2015; Wilson et al., 2014).

Concerning the lateral PFC, parts of it lie (functionally) between sensory areas and motor execution areas, and have thus immediate influence in motor planning (Tanaka et al., 2015). lPFC is also thought to be involved in learning stimulus-stimulus associations (Pan et al., 2014), category learning, i.e. learning which things belong together (Tanaka et al., 2015), and strategy switching when aspects of the environment or the task change (Domenech and Koechlin, 2015; Sharpe et al., 2019).

Sometimes learning about the values of alternative actions can happen without actually taking these actions and the anterior PFC is thought to mediate such processes by encoding the value of unchosen actions (Rushworth et al., 2011). Its activity has been found to increase with the probability to switch to another action at the next trial (Boorman et al., 2009), with the reliability of the current strategy (Donoso et al., 2014), and with exploratory actions (Daw et al., 2006). It has been suggested that aPFC keeps a representation of possible alternative courses of action, different from the currently followed one, which a person might employ in the short-term future (Koechlin and Hyafil, 2007). Another region also encoding information about switching strategy and important for flexibility in behaviour is the ACC (Kolling et al., 2016). ACC has been reported to respond to errors, to quantities that track recent performance (e.g. expected reward rate) and to updates of the model of the world (Kolling et al., 2016; O'Reilly et al., 2013), as we also mention in the next section.

Much about the circuit of the reward-system, the role of the different areas, and the flow of information between them remains unknown. It is not possible to cover in this introduction the vast amount of experimental findings that attempt to slowly solve the puzzle of this brain network. In addition to the above brief review of subcortical and cortical structures, we simply list some main players of the reward processing system and their assumed role in Table 1.1, and refer to a (non-exhaustive) collection of corresponding references for more information.

The fact that dopamine plays a role in the calculation and signalling of model-free prediction errors is fairly established. There are, however, findings that seem inconsistent with the Reward prediction error hypothesis for dopamine, or suggest that the role of dopamine is multi-faceted and that the account of this hypothesis alone is incomplete. The same applies for other reward-processing brain structures that have been traditionally thought to be involved in model-free learning, such as the striatum. For example, the role of dopamine in negative rewards is unclear, some dopamine neurons respond to both positive and negative events (Menegas et al., 2018), or even to neutral events (Watabe-Uchida et al., 2017), and other dopamine neurons increase their activity at the onset of movement (Coddington and Dudman, 2019; Jin and Costa, 2010) or change their activity according to the direction of movement (Jin and Costa, 2010; Watabe-Uchida et al., 2017). Dopamine activity also increases with novel stimuli (Menegas et al., 2018),

and recent studies have shown that some dopamine neurons respond to model-based or purely sensory prediction errors (Langdon et al., 2018; Stalnaker et al., 2019; Takahashi et al., 2017). A recent exciting work provided evidence that there is a wide range of different dopaminergic responses to the same reward, that could correspond to not only the expected reward, but to probability distribution of reward (Dabney et al., 2020). This work points to an implementation of *distributional RL* in the brain, could potentially reconcile previous puzzling findings, and might – in my view – bring a paradigm shift in the neuroscience of reinforcement learning.

1.4.3 The neuroscience of model learning and surprise signalling

Behaviour

As we already mentioned in the previous section, early experimental evidence indicated that animals are able to learn a model of an environment (Tolman, 1948). More recent experiments have further explored spatial model learning. For example, in Bast et al. (2009), after one single visit to a new reward location in a familiar maze, animals can immediately re-visit it from different starting locations, which is an indication of structure learning in the absence of reward and of rapid model-based learning.

An important type of experiments studying model-based or *goal-directed* behaviour is the *outcome devaluation* experimental paradigm. In these, the experimenter changes after some time the value of a stimulus, or the desirability of a reward, and tests whether the animal would repeat the previously selected action or whether it infers that the stimulus does not lead to a desired outcome anymore (Adams and Dickinson, 1981). Based on this paradigm, multiple other studies have shown model learning adaptive capabilities and association of stimuli in animals depending on many factors (e.g. training time) (Balleine, 2005; Dolan and Dayan, 2013; Pearce and Hall, 1980; Wilson et al., 1992).

Adaptive model learning in humans has been a topic of active research and has been demonstrated in numerous experiments (Behrens et al., 2007; Glaze et al., 2015; Heilbron and Meyniel, 2019; Nassar et al., 2010, 2012). In fact, humans seem to have a natural tendency to build a model and try to find structure, even when there is none, that is when they are exposed to sequences of random stimuli (Huettel et al., 2002; Meyniel et al., 2016; Yu and Cohen, 2009). A popular paradigm that studies humans' capacity to build expectations about the next stimulus, as well as the manifestation of surprise, is the *oddball* task (Meyniel et al., 2016; Ostwald et al., 2012; Squires et al., 1976). In oddball experiments participants view (or listen to) a sequence of (usually) two stimuli, where one of them is more improbable to occur, and have to press a different button for each stimulus. A behavioural manifestation of surprise is longer reaction times (Vassena et al., 2020; Vossel et al., 2014). An unexpected (oddball) stimulus causes longer reaction times, in proportion to the number of non-oddball stimuli preceding it (Huettel et al.,

2002; Meyniel et al., 2016; Vossel et al., 2014).

Model learning in humans is also actively studied in the context of reward-based tasks, where it is generally thought that humans employ hybrid strategies of model-free and model-based learning (Daw et al., 2011a; Gershman et al., 2014a; Gläscher et al., 2010; Simon and Daw, 2011). More details on existing studies on adaptive model learning and model estimation for reinforcement learning in humans can be found in Chapters 2 and 4, respectively.

Neural circuitries

Surprise in the brain has been associated with the neuromodulator norepinephrine, also called noradrenaline (Aston-Jones and Cohen, 2005). Norepinephrine is released by a set of neurons located in the locus coeruleus (LC), a brain structure in the brainstem. LC receives inputs from other structures in the brainstem and from the prefrontal cortex, and projects to essentially all cortical and subcortical regions in the brain, apart from the basal ganglia (Avery and Krichmar, 2017). Bursts of activity of LC neurons have been reported as a response to salient, unexpected or novel stimuli (Sara et al., 1995; Vankov et al., 1995) and sudden changes in tasks (Aston-Jones et al., 1997). The function of LC neurons is thought to be important for fast adaptation to changes and belief updating (Aston-Jones and Cohen, 2005; Aston-Jones et al., 1994; Avery and Krichmar, 2017; Bouret and Sara, 2005; Nassar et al., 2012), as well as shifts from exploitation to exploration (Aston-Jones and Cohen, 2005).

LC activity has been found to robustly correlate with pupil dilation, therefore pupillary responses have been used as a surrogate measure of noradrenergic activity (Aston-Jones and Cohen, 2005). In humans, pupil dilation has been found to correlate with unexpected uncertainty (Preuschoff et al., 2011) and with an adaptive learning rate and belief updating (Lavín et al., 2014; Nassar et al., 2012). At the same time, other neuromodulators have also been implicated for surprise signalling; acetylcholine has been associated to expected uncertainty, i.e. known stochasticity in the environment, (Yu and Dayan, 2005) and dopamine has also been reported as a response to surprising and novel stimuli (Avery and Krichmar, 2017; Krugel et al., 2009; Langdon et al., 2018; Morrens et al., 2020; Stalnaker et al., 2019; Takahashi et al., 2017).

Surprise manifestation in humans is extensively being studied through electroencephalography (EEG) recordings. Unexpected or oddball stimuli robustly elicit certain evoked related potentials (ERPs), namely the P300, that is positive deflection around 300ms after the stimulus, observed at electrodes covering the parietal lobe, and the mismatch negativity (MMN), that is a negative deflection roughly 150ms post-stimulus, over sensory areas (Gijsen et al., 2020; Kolossa et al., 2013; Mars et al., 2008; Modirshanechi et al., 2019; Ostwald et al., 2012; Squires et al., 1976).

Brain regions related to model estimation, model updating and surprise signals are widespread in the brain. Some of them are parts of the prefrontal cortex, such as the OFC (see previous section) and the ACC, the insula, the inferior parietal cortex, and the hippocampus (Table 1.1). The ACC in particular has been found to increase its activity with surprise, with model updating, with an adaptive learning rate and with an increasing advantage of a behavioural change (Kolling et al., 2016; O'Reilly et al., 2013).

1.4.4 The thin line between reward- and model- learning

As we saw, early experiments and theory displayed a dichotomy between model-free and model-based learning. The more we investigate how the brain learns, the more this separation seems to be a simplification (Collins and Cockburn, 2020; da Silva and Hare, 2020; Daw, 2018; Langdon et al., 2018). Animal and human behaviour shows both or a mixture of the two strategies (Daw et al., 2005, 2011a; Gläscher et al., 2010) and neural circuits are often shared (Langdon et al., 2018). The very notion of reward and surprise as it is perceived by the brain may be elusive; a reward may be surprising and a surprising event may be internally rewarding (Juechems and Summerfield, 2019; Schmidhuber, 1991), and it is difficult to think of real life situations where the two are clearly separated (beyond maybe primary rewards – such as food – in animal experiments). We just begin to unravel how these signals and learning mechanisms may be implemented in the brain, in order to understand its extraordinary capability to learn from interaction with the world, and both theory and experiments are vital to this end.

How are such signals combined and used by the brain? How should a model of the world be estimated and when is model estimation itself beneficial for a biological or artificial agent? How is surprise perceived by the brain and how can it be used for adapting to changes? How does the brain combine model-free and model-based strategies and how are different brain regions involved and work together? These are some of the open questions in the field of learning in neuroscience, the elucidation of which this thesis aims at contributing to.

Brain region	Reported associated functions
ventral striatum	reward prediction (Joel et al., 2002; Menegas et al., 2018; Watabe-Uchida et al., 2017) reward prediction error (Daw et al., 2011a; Gläscher et al., 2010; O'Doherty et al., 2003)
dorsal striatum	action selection and policy learning (Joel et al., 2002; Takahashi et al., 2008) stimulus-response and response-outcome associations (Balleine, 2005; Miller and Venditto, 2020)
ventromedial prefrontal cortex (vmPFC)	reward expectation & values (Haber, 2016; Rouault et al., 2019; Vassena et al., 2020) comparison of evaluated options (Haber, 2016)
orbitofrontal cortex (OFC)	reward expectation (Padoa-Schioppa and Assad, 2006; Stalnaker et al., 2018; Watabe-Uchida et al., 2017) stimulus-stimulus and stimulus-outcome associations (Balleine, 2005; Doll et al., 2012; Haber, 2016; McDannald et al., 2011; Schoenbaum et al., 2009) representation of observed and hidden states (Niv, 2019; Schuck et al., 2018; Wilson et al., 2014)
lateral prefrontal cortex (lPFC)	state prediction error (Gläscher et al., 2010) belief updating (Visalli et al., 2019) surprise (Visalli et al., 2019)
dorsal prefrontal cortex (dPFC)	reward expectation (Haber, 2016) working memory (Haber, 2016) behavioural shifts (Bissonette and Roesch, 2017)
prefrontal cortex (PFC) – inferior and middle frontal gyrus	sequence violation (Huettel et al., 2002) Bayesian surprise (d'Acremont et al., 2013) posterior belief (d'Acremont et al., 2013)
hippocampus	spatial learning, navigation, planning (Dolan and Dayan, 2013; Johnson and Redish, 2007; Mathis et al., 2012; O'Keefe and Nadel, 1978; Pfeiffer and Foster, 2013) episodic memory (Gershman and Daw, 2017) learning latent causes (Gershman and Niv, 2010)
anterior cingulate cortex (ACC)	learning rate (Behrens et al., 2007) action selection (Haber, 2016) sequence violation (Huettel et al., 2002) surprise (Alexander and Brown, 2019; Hayden et al., 2011; Schwartenbeck et al., 2016; Vassena et al., 2020; Visalli et al., 2019) belief updating (O'Reilly et al., 2013) behavioural shifts and planning (Kolling et al., 2016)
Intraparietal sulcus (pIPS)	state prediction error (Gläscher et al., 2010) belief updating (Kolling et al., 2016; Visalli et al., 2019) surprise (O'Reilly et al., 2013; Visalli et al., 2019) confidence (Payzan-LeNestour et al., 2013)

Table 1.1 – **Brain regions implicated in reward processing and learning.** The reported associated functions are accompanied with a non-exhaustive list of references. See also (Doll et al., 2012; Doya, 2008; Fouragnan et al., 2018; Haber, 2016; Huang et al., 2020; Koechlin, 2016; Miller and Venditto, 2020; O'Doherty et al., 2015; Rushworth et al., 2011; Sharpe et al., 2019; Sutton and Barto, 2018) for reviews and meta-analyses.

1.5 Thesis contribution

This thesis summarizes the research I conducted during my Ph.D. from 2015 to 2020 in the Laboratory of Computational Neuroscience (LCN) in EPFL, under the supervision of Prof. Wulfram Gerstner and the co-supervision of Prof. Kerstin Preuschoff (University of Geneva). The main goal of my work was to investigate the contributions of model estimation and model learning signals to reinforcement learning in complex tasks, both in terms of algorithmic performance and in terms of neural manifestations in the human brain.

In Chapter 2, we investigate how the brain may measure surprise and how surprise may be used by ever-adapting biological agents. We present a theoretical framework that bridges Bayesian inference and surprise-driven learning. We show that exact Bayesian inference leads to an adaptive trade-off between abandoning the current belief and integrating it with a new piece of information, which is modulated by a naturally emerging measure of surprise. We present three novel scalable approximate algorithms that reach high levels of performance and may be implemented by biological agents. Finally, we demonstrate that both our proposed algorithms and various existing approaches use the same emerging measure of surprise in their update rules.

In Chapter 3, we are interested in the role of surprise in reward-driven learning and in building model-based reinforcement learning agents that can successfully adapt in non-stationary environments. We combine one of our surprise-based approximate algorithms of Chapter 2, as well as simpler methods, with Prioritized Sweeping, and we investigate the scenarios under which accurate and adaptive model estimation is beneficial for reinforcement learning. In environments with abrupt changes that directly affect the agent's policy, surprise-modulation leads to higher performance. In environments with distal changes, where exploration is crucial, a simple leaky integration with background forgetting is sufficiently successful.

In Chapter 4, we shift our attention to the manifestation of model-based and model-free learning in human behaviour and in BOLD responses. We design a novel multi-step task that decorrelates learning components of different strategies at the level of brain signals. We find that behaviour, in this task, is best described by a model-free Actor-critic algorithm, potentially influenced by model estimation, and we find signatures of both model-free and model-based prediction errors in brain responses.

Finally, I conclude with a short summary and general future directions and, in the Appendix, I provide the abstract of a related project to which I contributed (Lehmann et al., 2019).

The work presented in this thesis is a product of fruitful collaborations with Dr. Johanni Brea, Dr. Marco Lehmann and Alireza Modirshanechi. My specific contributions to each

project are stated at the end of each chapter, and summarized collectively for all projects at the end of the thesis, for convenience. An illustrative schematic of each chapter's content can be seen in Fig. 1.3.

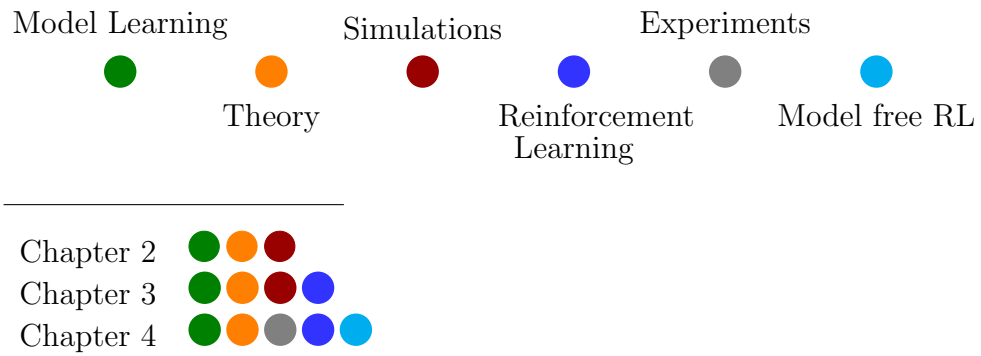


Figure 1.3 – **Chapters' content.** Schematic of the topics and the content of each chapter in this thesis. (Illustration inspired by a report written by Dr. Thomas Bolton.)

2 Learning in Volatile Environments with the Bayes Factor Surprise

This chapter presents research performed in collaboration with Alireza Modirshanechi, Dr. Johanni Brea and Prof. Wulfram Gerstner.
Alireza Modirshanechi and I contributed equally to this work [1].

2.1 Introduction

Animals, humans, and similarly reinforcement learning agents may safely assume that the world is stochastic and stationary during some intervals of time interrupted by change points. The position of leafs on a tree, a stock market index, or the time it takes to travel from A to B in a crowded city is often well captured by stationary stochastic processes for extended periods of time. Then sudden changes may happen, such that the distribution of leaf positions becomes different due to a storm, the stock market index is affected by the enforcement of a new law, or a blocked road causes additional traffic jams. The violation of an agent's expectation caused by such sudden changes is perceived by the agent as surprise, which can be seen as a measure of how much the agent's current belief differs from reality.

Surprise, with its physiological manifestations in pupil dilation (Nassar et al., 2012; Preuschoff et al., 2011) and EEG signals (Mars et al., 2008; Modirshanechi et al., 2019; Ostwald et al., 2012), is believed to modulate learning, potentially through the release of specific neurotransmitters (Gerstner et al., 2018; Yu and Dayan, 2005), so as to allow animals and humans to adapt quickly to sudden changes. The quick adaptation to novel situations has been demonstrated in a variety of learning experiments (Behrens et al., 2007; Glaze et al., 2015; Heilbron and Meyniel, 2019; Nassar et al., 2010, 2012; Yu and Dayan, 2005). The bulk of computational work on surprise-based learning can be separated into

[1] This work is currently accepted for publication in Neural Computation.

two groups. Studies in the field of computational neuroscience have focused on biological plausibility with little emphasis on the accuracy of learning (Behrens et al., 2007; Bogacz, 2017; Faraji et al., 2018; Friston, 2010; Friston et al., 2017; Nassar et al., 2010, 2012; Ryali et al., 2018; Schwartenbeck et al., 2013; Yu and Dayan, 2005), whereas exact and approximate Bayesian online methods (Adams and MacKay, 2007; Fearnhead and Liu, 2007) for change-point detection and parameter estimation have been developed without any focus on biological plausibility (Aminikhanghahi and Cook, 2017; Cummings et al., 2018; Lin et al., 2017; Masegosa et al., 2017; Wilson et al., 2010).

In this work, we take a top-down approach to surprise-based learning. We start with a generative model of change points similar to the one that has been the starting point of multiple experiments (Behrens et al., 2007; Findling et al., 2019; Glaze et al., 2015; Heilbron and Meyniel, 2019; Nassar et al., 2010, 2012; Yu and Dayan, 2005). We demonstrate that Bayesian inference on such a generative model can be interpreted as modulation of learning by surprise; we show that this modulation leads to a natural definition of surprise which is different, but closely related to Shannon Surprise (Shannon, 1948). Moreover, we derive three novel approximate online algorithms with update rules that inherit the surprise-modulated adaptation rate of exact Bayesian inference. The overall goal of the present study is to give a Bayesian interpretation for surprise-based learning in the brain, and to find approximate methods that are computationally efficient and biologically plausible while maintaining the learning accuracy at a high level. As a by-product, our approach provides theoretical insights on commonalities and differences among existing surprise-based and approximate Bayesian approaches. Importantly, our approach makes specific experimental predictions.

In the Results section, we first introduce the generative model, and then we present our surprise-based interpretation of Bayesian inference and our three approximate algorithms. Next, we use simulations to compare our algorithms with existing ones on two different tasks inspired by and closely related to real experiments (Behrens et al., 2007; Mars et al., 2008; Nassar et al., 2010, 2012; Ostwald et al., 2012). At the end of the Results section, we formalize two experimentally testable predictions of our theory and illustrate them with simulations. A brief review of related studies as well as a few directions for further work are supplied in the Discussion section.

2.2 Results

In order to study learning in an environment that exhibits occasional and abrupt changes, we consider a hierarchical generative model (Fig. 2.1A) in discrete time, similar to existing model environments (Behrens et al., 2007; Nassar et al., 2010, 2012; Yu and Dayan, 2005). At each time point t, the observation $Y_t = y$ comes from a distribution with the time-invariant likelihood $P_Y(y|\theta)$ parameterized by $\Theta_t = \theta$, where both y and θ can be multi-dimensional. In general, we indicate random variables by capital letters, and values

by small letters. Whenever there is no risk of ambiguity, we drop the explicit notation of random variables to simplify notation. Abrupt changes of the environment correspond to sudden changes of the parameter θ_t. At every time t, there is a change probability $p_c \in (0,1)$ for the parameter θ_t to be drawn from its prior distribution $\mathbb{b}^{(0)}$ independently of its previous value, and a probability $1-p_c$ to stay the same as θ_{t-1}. A change at time t is specified by the event $C_t = 1$; otherwise $C_t = 0$. Therefore, the generative model can be formally defined, for any $T \geq 1$, as a joint probability distribution over $\Theta_{1:T} \equiv (\Theta_1, \ldots, \Theta_T)$, $C_{1:T}$, and $Y_{1:T}$ as

$$\mathbf{P}(c_{1:T}, \theta_{1:T}, y_{1:T}) = \mathbf{P}(c_1)\mathbf{P}(\theta_1)\mathbf{P}(y_1|\theta_1) \prod_{t=2}^{T} \mathbf{P}(c_t)\mathbf{P}(\theta_t|c_t, \theta_{t-1})\mathbf{P}(y_t|\theta_t)\,, \tag{2.1}$$

where $\mathbf{P}(\theta_1) = \mathbb{b}^{(0)}(\theta_1)$, $\mathbf{P}(c_1) = \delta(c_1 - 1)$, and

$$\mathbf{P}(c_t) = \text{Bernoulli}(c_t; p_c)\,, \tag{2.2}$$

$$\mathbf{P}(\theta_t|c_t, \theta_{t-1}) = \begin{cases} \delta(\theta_t - \theta_{t-1}) & \text{if} \quad c_t = 0\,, \\ \mathbb{b}^{(0)}(\theta_t) & \text{if} \quad c_t = 1\,, \end{cases} \tag{2.3}$$

$$\mathbf{P}(y_t|\theta_t) = P_Y(y_t|\theta_t)\,. \tag{2.4}$$

$\mathbf{P}$ stands for either probability density function (for the continuous variables) or probability mass function (for the discrete variables), and δ is the Dirac or Kronecker delta distribution, respectively.

Given a sequence of observations $y_{1:t}$, the *agent's belief* $\mathbb{b}^{(t)}(\theta)$ about the parameter θ at time t is defined as the posterior probability distribution $\mathbf{P}(\Theta_t = \theta|y_{1:t})$. In the online learning setting studied here, the agent's goal is to update the belief $\mathbb{b}^{(t)}(\theta)$ to the new belief $\mathbb{b}^{(t+1)}(\theta)$, or an approximation thereof, upon observing y_{t+1}.

A simplified real-world example of such an environment is illustrated in Fig. 2.1B. Imagine that every day a friend of yours meets you at the coffee shop, starting after work from her office (Fig. 2.1B left). To do so, she needs to cross a river via a bridge. The time of arrival of your friend (i.e. y_t) exhibits some variability, due to various sources of stochasticity (e.g. traffic and your friend's daily workload), but it has a stable average over time (i.e. θ_t). However, if a new bridge is opened, your friend arrives earlier, since she no longer has to take detour (Fig. 2.1B right). The moment of opening the new bridge is indicated by $c_{t+1} = 1$ in our framework, and the sudden change in the average arrival time of your friend by a sudden change from θ_t to θ_{t+1}. Even without any explicit discussion with your friend about this situation and only by observing her actual arrival time, you can notice the abrupt change and hence adapt your schedule to the new situation.

2.2.1 Online Bayesian inference modulated by surprise

According to the definition of the hierarchical generative model (Fig. 2.1A and Equation 2.1 to Equation 2.4), the value y_{t+1} of the observation at time $t+1$ depends only on the parameters θ_{t+1}, and is (given θ_{t+1}) independent of earlier observations and earlier parameter values. We exploit this Markovian property and update, using Bayes' rule, the belief $\mathbb{b}^{(t)}(\theta) \equiv \mathbf{P}(\Theta_t = \theta | y_{1:t})$ at time t to the new belief at time $t+1$

$$\mathbb{b}^{(t+1)}(\theta) = \frac{P_Y(y_{t+1}|\theta)\mathbf{P}(\Theta_{t+1} = \theta | y_{1:t})}{\mathbf{P}(y_{t+1}|y_{1:t})}\,. \tag{2.5}$$

So far, Equation 2.5 remains rather abstract. The aim of this section is to rewrite it in the form of a surprise-modulated recursive update. The first term in the numerator of Equation 2.5 is the likelihood of the current observation given the parameter $\Theta_{t+1} = \theta$, and the second term is the agent's estimated probability distribution of Θ_{t+1} before observing y_{t+1}. Because there is always the possibility of an abrupt change, the second term is not the agent's previous belief $\mathbb{b}^{(t)}$, but $\mathbf{P}(\Theta_{t+1} = \theta | y_{1:t}) = (1 - p_c)\mathbb{b}^{(t)}(\theta) + p_c\mathbb{b}^{(0)}(\theta)$. As a result, it is possible to find a recursive formula for updating the belief. For the derivation of this recursive rule, we define the following terms.

Definition 1. *The probability or density (for discrete and continuous variables respectively) of observing y with a belief $\mathbb{b}^{(t')}$ is denoted as*

$$P(y; \mathbb{b}^{(t')}) = \int P_Y(y|\theta)\mathbb{b}^{(t')}(\theta)d\theta\,. \tag{2.6}$$

Note that if $\mathbb{b}$ is the exact Bayesian belief defined as above in Equation 2.5, then $P(y; \mathbb{b}^{(t')}) = \mathbf{P}(Y_{t'+1} = y | y_{1:t'}, c_{t'+1} = 0)$. In Section 2.2.2 we will use also $P(y; \hat{\mathbb{b}}^{(t')})$ for an arbitrary $\hat{\mathbb{b}}^{(t')}$. Two particularly interesting cases of Equation 2.6 are $P(y_{t+1}; \mathbb{b}^{(t)})$, i.e. the probability of a new observation y_{t+1} with the current belief $\mathbb{b}^{(t)}$, and $P(y_{t+1}; \mathbb{b}^{(0)})$, i.e. the probability of a new observation y_{t+1} with the prior belief $\mathbb{b}^{(0)}$.

Definition 2. *The "Bayes Factor Surprise" $\boldsymbol{S}_{\mathrm{BF}}$ of the observation y_{t+1} is defined as the ratio of the probability of observing y_{t+1} given $c_{t+1} = 1$ (i.e. when there is a change), to the probability of observing y_{t+1} given $c_{t+1} = 0$ (i.e. when there is no change), i.e.*

$$\boldsymbol{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)}) = \frac{P(y_{t+1}; \mathbb{b}^{(0)})}{P(y_{t+1}; \mathbb{b}^{(t)})}\,. \tag{2.7}$$

This definition of surprise measures how much more probable the current observation is under the naive prior $\mathbb{b}^{(0)}$ relative to the current belief $\mathbb{b}^{(t)}$ (see the Discussion section for further interpretation). This probability ratio is the Bayes factor (Efron and Hastie, 2016; Kass and Raftery, 1995) that tests the prior belief $\mathbb{b}^{(0)}$ against the current belief $\mathbb{b}^{(t)}$. We emphasize that our definition of surprise is not arbitrary, but essential in order to write the exact inference in Equation 2.5 on the generative model in the compact recursive

form indicated in the Proposition that follows. Moreover, as we show later, this term can be identified in multiple learning algorithms (among them Nassar et al. (2010, 2012)), but it has never been interpreted as a surprise measure. In the following sections we establish the generality of this computational mechanism and identify it as a common feature of many learning algorithms.

Definition 3. *Under the assumption of no change $c_{t+1} = 0$, and using the most recent belief $\mathbb{b}^{(t)}$ as prior, the exact Bayesian update for $\mathbb{b}^{(t+1)}$ is denoted as*

$$\mathbb{b}_B^{(t+1)}(\theta) = \frac{P_Y(y_{t+1}|\theta)\mathbb{b}^{(t)}(\theta)}{P(y_{t+1}; \mathbb{b}^{(t)})}\,. \tag{2.8}$$

$\mathbb{b}_B^{(t+1)}(\theta)$ describes the incorporation of the new information into the current belief via Bayesian updating.

Definition 4. *The "Surprise-Modulated Adaptation Rate" is a function $\gamma : \mathbb{R}^+ \times \mathbb{R}^+ \to [0, 1]$ specified as*

$$\gamma(S, m) = \frac{mS}{1 + mS}\,, \tag{2.9}$$

where $S \geq 0$ is a surprise value, and $m \geq 0$ is a parameter controlling the effect of surprise on learning.

Using the above definitions and Equation 2.5, we have for the generative model of Fig. 2.1A and Equation 2.1 to Equation 2.4 the following Proposition.

Proposition. *Exact Bayesian inference on the generative model is equivalent to the recursive update rule*

$$\mathbb{b}^{(t+1)}(\theta) = \Big(1 - \gamma\Big(\boldsymbol{S}_{\mathrm{BF}}^{(t+1)}, \frac{p_c}{1 - p_c}\Big)\Big)\mathbb{b}_B^{(t+1)}(\theta) + \gamma\Big(\boldsymbol{S}_{\mathrm{BF}}^{(t+1)}, \frac{p_c}{1 - p_c}\Big)P(\theta|y_{t+1})\,, \tag{2.10}$$

where $\boldsymbol{S}_{\mathrm{BF}}^{(t+1)} = \boldsymbol{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)})$ is the Bayes Factor Surprise and

$$P(\theta|y_{t+1}) = \frac{P_Y(y_{t+1}|\theta)\mathbb{b}^{(0)}(\theta)}{P(y_{t+1}; \mathbb{b}^{(0)})} \tag{2.11}$$

is the posterior if we take y_{t+1} as the only observation.

The proposition indicates that the exact Bayesian inference on the generative model discussed above (Fig. 2.1) leads to an explicit trade-off between (i) integrating a new observation y^{new} (corresponding to y_{t+1}) with the old belief $\mathbb{b}^{\mathrm{old}}$ (corresponding to $\mathbb{b}^{(t)}$) into a distribution $\mathbb{b}^{\mathrm{integration}}$ (corresponding to $\mathbb{b}_B^{(t+1)}$) and (ii) forgetting the past observations, so as to restart with the belief $\mathbb{b}^{\mathrm{reset}}$ (corresponding to $P(\theta|y_{t+1})$) which

relies only on the new observation and the prior $\mathbb{b}^{(0)}$

$$\mathbb{b}^{\text{new}}(\theta) = (1-\gamma)\,\mathbb{b}^{\text{integration}}(\theta|y^{\text{new}}, \mathbb{b}^{\text{old}}) + \gamma\,\mathbb{b}^{\text{reset}}(\theta|y^{\text{new}}, \mathbb{b}^{(0)}). \tag{2.12}$$

This trade-off is governed by a *surprise-modulated adaptation rate* $\gamma(\mathrm{S}, m) \in [0, 1]$, where $\mathrm{S} = \mathbf{S}_{\mathrm{BF}} \geq 0$ (corresponding to the Bayes Factor Surprise) can be interpreted as the surprise of the most recent observation, and $m = \frac{p_c}{1-p_c} \geq 0$ is a parameter controlling the effect of surprise on learning. Because the parameter of modulation m is equal to $\frac{p_c}{1-p_c}$, for a fixed value of surprise S, the adaptation rate γ is an increasing function of p_c. Therefore, in more volatile environments, the same value of surprise S leads to a higher adaptation rate than in a less volatile environment; in the case of $p_c \to 1$, any surprise value leads to full forgetting, i.e. $\gamma = 1$.

As a conclusion, our first main result is that a split as in Equation 4.6 with a weighting factor ("adaptation rate" γ) as in Equation 2.9 is exact and always possible for the class of environments defined by our hierarchical generative model. This surprise-modulation gives rise to specific testable experimental predictions discussed later.

2.2.2 Approximate algorithms modulated by surprise

Despite the simplicity of the recursive formula in Equation 2.10, the updated belief $\mathbb{b}^{(t+1)}$ is generally not in the same family of distributions as the previous belief $\mathbb{b}^{(t)}$, e.g. the result of averaging two normal distributions is not a normal distribution. Hence it is in general impossible to find a simple and exact update rule for e.g. some sufficient statistic. As a consequence, the memory demands for $\mathbb{b}^{(t+1)}$ scale linearly in time, and updating $\mathbb{b}^{(t+1)}$ using $\mathbb{b}^{(t)}$ needs $\mathcal{O}(t)$ operations. In the following sections, we investigate three approximations (Algo. 1-3) that have simple update rules and finite memory demands, so that the updated belief remains tractable over a long sequence of observations.

As our second main result, we show that all three novel approximate algorithms inherit the surprise-modulated adaptation rate from the exact Bayesian approach, i.e. Equation 2.9 and Equation 4.6. The first algorithm adapts an earlier algorithm of surprise minimization learning (SMiLe, Faraji et al. (2018)) to variational learning. We refer to our novel algorithm as Variational SMiLe and abbreviate it by VarSMiLe (see Algo. 1). The second algorithm is based on message passing (Adams and MacKay, 2007) restricted to a finite number of messages N . We refer to this algorithm as MPN (see Algo. 2). The third algorithm uses the ideas of particle filtering (Gordon et al., 1993) for an efficient approximation for our hierarchical generative model. We refer to our approximate algorithm as Particle Filtering with N particles and abbreviate it by pf$N$ (see Algo. 3). All algorithms are computationally efficient, have finite memory demands and are biologically plausible; Particle Filtering has possible neuronal implementations (Huang and Rao, 2014; Kutschireiter et al., 2017; Legenstein and Maass, 2014; Shi and Griffiths, 2009), MPN can be seen as a greedy version of pf$N$ without sampling, and Variational

SMiLe may be implemented by neo-Hebbian (Gerstner et al., 2018; Lisman et al., 2011) update rules. Simulation results show that the performance of the three approximate algorithms is comparable to and more robust across environments than other state-of-the-art approximations.

Variational SMiLe Rule (Algo. 1)

A simple heuristic approximation to keep the updated belief in the same family as the previous beliefs consists in applying the weighted averaging of the exact Bayesian update rule (Equation 2.10) to the logarithm of the beliefs rather than the beliefs themselves, i.e.

$$\log\big(\hat{\mathbb{b}}^{(t+1)}(\theta)\big) = (1-\gamma_{t+1})\,\log\big(\hat{\mathbb{b}}_B^{(t+1)}(\theta)\big) + \gamma_{t+1}\,\log\big(P(\theta|y_{t+1})\big) + \text{Const.}\,, \quad (2.13)$$

where $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\text{BF}}(y_{t+1};\hat{\mathbb{b}}^{(t)}), m\big)$ is given by Equation 2.9 with a free parameter $m > 0$ which can be tuned to each environment. By doing so, we still have the explicit trade-off between two terms as in Equation 2.10, but in the logarithms; yet an advantageous consequence of averaging over logarithms is that, if the likelihood function P_Y is in the exponential family, and if the initial belief $\mathbb{b}^{(0)}$ is its conjugate prior, then $\hat{\mathbb{b}}^{(t+1)}$ and $\mathbb{b}^{(0)}$ are members of the same family. In this particular case, we arrive at a simple update rule for the parameters of $\hat{\mathbb{b}}^{(t+1)}$ (see Algorithm 1 for pseudocode and Methods for details). As it is common in variational approaches (Beal, 2003), the price of this simplicity is that, except for the trivial cases of $p_c = 0$ and $p_c = 1$, there is no evidence other than simulations that the update rule of Equation 2.13 will end up at an approximate belief close to the exact Bayesian belief.

One way to interpret the update rule of Equation 2.13 is to rewrite it as the solution of a constraint optimization problem. The new belief $\hat{\mathbb{b}}^{(t+1)}$ is a variational approximation of the Bayesian update $\hat{\mathbb{b}}_B^{(t+1)}$ (see Methods)

$$\hat{\mathbb{b}}^{(t+1)}(\theta) = \arg\min_q \mathbf{D}_{KL}\big[q(\theta)||\hat{\mathbb{b}}_B^{(t+1)}(\theta)\big], \quad (2.14)$$

with a family of functions $q(\theta)$ constrained by the Kullback-Leibler divergence

$$\mathbf{D}_{KL}\big[q(\theta)||P(\theta|y_{t+1})\big] \leq B_{t+1}\,, \quad (2.15)$$

where the bound $B_{t+1} \in \big[0, \mathbf{D}_{KL}[\hat{\mathbb{b}}_B^{(t+1)}(\theta)||P(\theta|y_{t+1})]\big]$ is a decreasing function of the Bayes Factor surprise $\mathbf{S}_{\text{BF}}(y_{t+1};\hat{\mathbb{b}}^{(t)})$ (see Methods for proof), and $P(\theta|y_{t+1})$ is given by Equation 2.11.

Because of the similarity of the constraint optimization problem in Equation 2.14 and Equation 2.15 to the Surprise Minimization Learning rule "SMiLe" (Faraji et al., 2018), we call this algorithm "Variational Surprise Minimization Learning" rule, or in short

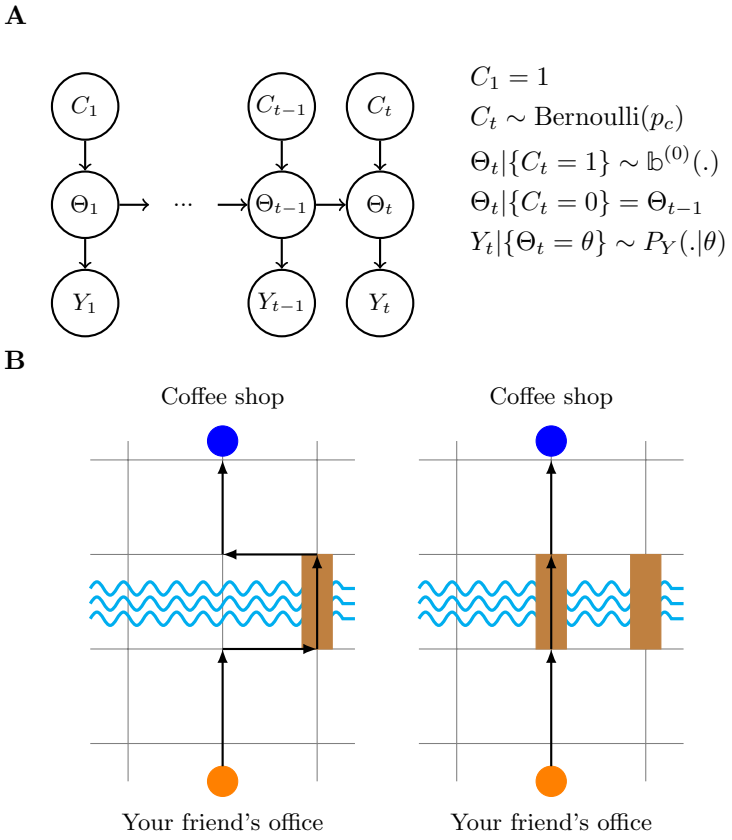


Figure 2.1 – **Non-stationary environment. A**. The generative model. At each time point t there is a probability $p_c \in (0, 1)$ for a change in the environment. When there is a change in the environment, i.e. $C_t = 1$, the parameter Θ_t is drawn from its prior distribution $\mathbb{b}^{(0)}$, independently of its previous value. Otherwise the value of Θ_t retains its value from the previous time step $t - 1$. Given a parameter value $\Theta_t = \theta$, the observation $Y_t = y_t$ is drawn from a probability distribution $P_Y(y_t|\theta)$. We indicate random variables by capital letters, and values by small letters. **B**. Example of a non-stationary environment. Your friend meets you every day at the coffee shop (blue dot) starting after work from her office (orange dot) crossing a river. The time of arrival of your friend is the observed variable Y_t, which due to the traffic or your friend's workload may exhibit some variability, but has a stable expectation (i.e. θ). If, however, a new bridge is opened (i.e. $C_t = 1$ where t is the moment of change), your friend no longer needs to take a detour. There is, then, a sudden change in her observed daily arrival times.

"Variational SMiLe" rule. The differences between SMiLe and Variational SMiLe are discussed in the Methods section.

Our variational method, and particularly its surprise-modulated adaptation rate, is complementary to earlier studies (Masegosa et al., 2017; Özkan et al., 2013) in machine learning which assumed different generative models and used additional assumptions and different approaches for deriving the learning rule.

Algorithm 1 Pseudocode for Variational SMiLe (exponential family)

Specify $P_Y(y|\theta)$, $\mathbf{P}_{\mathbb{b}}\big(\Theta = \theta; \chi, \nu\big)$, and $\phi(y)$
where $P_Y \in \{\text{exponential family}\}$, $\mathbf{P}_{\mathbb{b}} \in \{\text{conjugate priors of } P_Y\}$ parametrized by χ and ν, and $\phi(y)$ is the sufficient statistic.
Specify m.
Initialize $\chi^{(0)}$, $\nu^{(0)}$, and $t \leftarrow 0$.
while the sequence is not finished **do**
    Observe y_{t+1}
    # Surprise
    Compute $\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mathbb{b}}^{(t)})$ using Equation 2.83
    # Modulation factor
    Compute $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mathbb{b}}^{(t)}), m\big)$
    # Updated belief
    $\boxed{\chi^{(t+1)} \leftarrow (1 - \gamma_{t+1})\chi^{(t)} + \gamma_{t+1}\chi^{(0)} + \phi(y_{t+1})}$
    $\boxed{\nu^{(t+1)} \leftarrow (1 - \gamma_{t+1})\nu^{(t)} + \gamma_{t+1}\nu^{(0)} + 1}$
    $\hat{\mathbb{b}}^{(t+1)}(\theta) = \mathbf{P}_{\mathbb{b}}\big(\Theta = \theta; \chi^{(t+1)}, \nu^{(t+1)}\big)$
    # Iterate
    $t \leftarrow t + 1$

Message-Passing N (Algo. 2)

For a hierarchical generative model similar to ours, a message passing algorithm has been used to perform exact Bayesian inference (Adams and MacKay, 2007), where the algorithm's memory demands and computational complexity scale linearly in time t. In this section, we first explain the idea of the message passing algorithm of Adams and MacKay (2007) and its relation to our Proposition. We then present our approximate version of this algorithm which has a constant (in time) computational complexity and memory demands.

The history of change points up to time t is a binary sequence, e.g. $c_{1:t} = \{1, 0, 0, 1, 0, 1, 1\}$, where the value 1 indicates a change in the corresponding time step. Following the idea of Adams and MacKay (2007), we define the random variable $R_t = \min\{n \in \mathbb{N} : C_{t-n+1} = 1\}$ in order to describe the time since the last change point, which takes values between 1 to t. We can write the exact Bayesian expression for $\mathbb{b}^{(t)}(\theta)$ by marginalizing $\mathbf{P}(\Theta_t = \theta, r_t | y_{1:t})$

over the t possible values of r_t in the following way

$$\mathbb{b}^{(t)}(\theta) = \sum_{k=0}^{t-1} \mathbf{P}(R_t = t-k|y_{1:t})\mathbf{P}(\Theta_t = \theta|R_t = t-k, y_{1:t}). \tag{2.16}$$

For consistency with Algorithm 3 (i.e. Particle Filtering), we call each term in the sum of Equation 2.16 a "particle", and denote as $\mathbb{b}_k^{(t)}(\theta) = \mathbf{P}(\Theta_t = \theta|R_t = t-k, y_{1:t})$ the belief of the particle corresponding to $R_t = t-k$, and $w_t^{(k)} = \mathbf{P}(R_t = t-k|y_{1:t})$ its corresponding weight at time t, i.e.

$$\mathbb{b}^{(t)}(\theta) = \sum_{k=0}^{t-1} w_t^{(k)} \mathbb{b}_k^{(t)}(\theta). \tag{2.17}$$

For each particle, the term $\mathbb{b}_k^{(t)}(\theta)$ is simple to compute, because when r_t is known, inference depends only on the observations after the last change point. Therefore, the goal of online inference is to find an update rule for the evolution of the weights $w_t^{(k)}$ over time.

We can apply the exact update rule of our Proposition (Equation 2.10) to the belief expressed in the form of Equation 2.17. Upon each observation of a new sample y_{t+1}, a new particle is generated and added to the set of particles, corresponding to $P(\theta|y_{t+1})$ (i.e. $\mathbb{b}^{reset}$), modelling the possibility of a change point occurring at $t+1$. According to the proposition, the weight of the new particle (i.e. $k = t$) is equal to

$$w_{t+1}^{(t)} = \gamma_{t+1}, \tag{2.18}$$

where $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)}), \frac{p_c}{1-p_c}\big)$ (cf. Equation 2.9). The other t particles coming from $\mathbb{b}^{(t)}$ corresponds to $\mathbb{b}_B^{(t+1)}$ (i.e. $\mathbb{b}^{integration}$) in the proposition. The update rule (see Methods for derivation) for the weights of these particles (i.e. $0 \leq k \leq t-1$) is

$$w_{t+1}^{(k)} = (1-\gamma_{t+1}) w_{B,t+1}^{(k)} = (1-\gamma_{t+1}) \frac{P(y_{t+1}; \mathbb{b}_k^{(t)})}{P(y_{t+1}; \mathbb{b}^{(t)})} w_t^{(k)}. \tag{2.19}$$

So far, we used the idea of (Adams and MacKay, 2007) to write the belief as in Equation 2.17 and used our proposition to arrive at the surprise-modulated update rules in Equation 2.18 and Equation 2.19.

The computational complexity and memory requirements of the complete message passing algorithm increase linearly with time t. To deal with this issue and to have a constant computation and memory demands over time, we implemented a message passing algorithm of the form of Equation 2.17 to Equation 2.19, but with a fixed number N of particles, chosen as those with the highest weights $w_t^{(k)}$. Therefore, our second algorithm adds a new approximation step to the full message passing algorithm of Adams and

MacKay (2007): Whenever $t > N$, after adding the new particle with the weight as in Equation 2.18 and updating the previous weights as in Equation 2.19, we discard the particle with the smallest weight (i.e. set its weight equal to 0), and renormalize the weights. By doing so, we always keep the number of particles with non-zero weights equal to N. Note that, for $t \leq N$, our algorithm is exact, and identical to the message passing algorithm of (Adams and MacKay, 2007). We call our modification of the message passing algorithm of Adams and MacKay (2007) "Message Passing N" and abbreviate it by "MPN".

To deal with the computational complexity and memory requirements, one may alternatively keep only the particles with weights greater than a cut-off threshold (Adams and MacKay, 2007). However, such a constant cut-off leads to a varying number (smaller or equal to t) of particles in time. Our approximation MPN can therefore be seen as a variation of the thresholding algorithm in Adams and MacKay (2007) with fixed number of particles N, and hence a variable cut-off threshold. The work of Fearnhead and Liu (2007) follows the same principle as Adams and MacKay (2007), but employs stratified resampling to eliminate particles with negligible weights, in order to reduce the total number of particles. Their resampling algorithm involves solving a complicated non-linear equation at each time step, which makes it unsuitable for a biological implementation. In addition, we experienced that in some cases, the small errors introduced in the resampling step of the algorithm of Fearnhead and Liu (2007) accumulated and led to a worse performance than our MPN algorithm which simply keeps the N particles with the highest weight at each time step.

For the case where the likelihood function $P_Y(y|\theta)$ is in the exponential family and $\mathbb{b}^{(0)}$ is its conjugate prior, the resulting algorithm of MPN has a simple update rule for the belief parameters (see Algorithm 2 and Methods for details). For the sake of comparison, we also implemented in our simulations the full message passing algorithm of Adams and MacKay (2007) with an almost zero cut-off (machine precision), which we consider as our benchmark "Exact Bayes", as well as the stratified optimal resampling algorithm of Fearnhead and Liu (2007), called "SORN".

Particle Filtering (Algo. 3)

Equation 2.17 demonstrates that the exact Bayesian belief $\mathbb{b}^{(t)}$ can be expressed as a sum of two factors, i.e. as the marginalization of $\mathbf{P}(\Theta_t = \theta, r_t|y_{1:t})$ over the time since the last change point r_t. Equivalently, one can compute the exact Bayesian belief as the marginalization of $\mathbf{P}(\Theta_t = \theta, c_{1:t}|y_{1:t})$ over the history of change points $c_{1:t}$, i.e.

$$\begin{aligned} \mathbb{b}^{(t)}(\theta) &= \sum_{c_{1:t}} \mathbf{P}(c_{1:t}|y_{1:t})\mathbf{P}(\Theta_t = \theta|c_{1:t}, y_{1:t}) \\ &= \mathbb{E}_{\mathbf{P}(C_{1:t}|y_{1:t})}\big[\mathbf{P}(\Theta_t = \theta|C_{1:t}, y_{1:t})\big]. \end{aligned} \tag{2.20}$$

Algorithm 2 Pseudocode for MPN (exponential family)

Specify $P_Y(y|\theta)$, $\mathbf{P}_{\mathbb{b}}\big(\Theta = \theta; \chi, \nu\big)$, and $\phi(y)$
where $P_Y \in \{\text{exponential family}\}$, $\mathbf{P}_{\mathbb{b}} \in \{\text{conjugate priors of } P_Y\}$ parametrized by χ and ν, and $\phi(y)$ is the sufficient statistic.
Specify $m = p_c/(1-p_c)$, and N.
Initialize $\chi_1^{(0)}$, $\nu_1^{(0)}$, $w_0^{(1)} = 1$ and $t \leftarrow 0$.
Until $N = t$, do the exact message passing algorithm of Equation 2.19 and Equation 2.18
while the sequence is not finished and $N < t$ **do**
    Observe y_{t+1}
    # Surprise per particle i
    for $i \in \{1, ..., N\}$ **do**
        Compute $\mathbf{S}_{\mathrm{BF}}(y_{t+1}, \hat{\mathbb{b}}_i^{(t)})$ using Equation 2.83 with $\chi_i^{(t)}$, $\nu_i^{(t)}$
    # Global surprise
    Compute $\mathbf{S}_{\mathrm{BF}}(y_{t+1}, \hat{\mathbb{b}}^{(t)})$ as the weighted $(w_t^{(i)})$ harmonic mean of $\mathbf{S}_{\mathrm{BF}}(y_{t+1}, \hat{\mathbb{b}}_i^{(t)})$
    # Modulation factor
    Compute $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}, \hat{\mathbb{b}}^{(t)}), m\big)$
    # Weight per particle i
    for $i \in \{1, ..., N\}$ **do**
        Compute the Bayesian weight $w_{B,t+1}^{(i)}$ using Equation 2.22
        $\boxed{w_{t+1}^{(i)} \leftarrow (1-\gamma_{t+1}) w_{B,t+1}^{(i)}}$
    # Weight for the new particle
    $\boxed{w_{t+1}^{(N+1)} \leftarrow \gamma_{t+1}}$
    # Updated belief per particle i
    for $i \in \{1, ..., N\}$ **do**
        $\chi_i^{(t+1)} \leftarrow \chi_i^{(t)} + \phi(y_{t+1})$ and $\nu_i^{(t+1)} \leftarrow \nu_i^{(t)} + 1$
    $\chi_{N+1}^{(t+1)} \leftarrow \chi^{(0)} + \phi(y_{t+1})$ and $\nu_{N+1}^{(t+1)} \leftarrow \nu^{(0)} + 1$
    # Approximation
    Keep the N particles with highest weights among $w_{t+1}^{(1:N+1)}$, rename and normalize their weights
    # Updated belief
    $\hat{\mathbb{b}}^{(t+1)}(\theta) = \sum_{i=1}^{N} w_{t+1}^{(i)} \mathbf{P}_{\mathbb{b}}\big(\Theta = \theta; \chi_i^{(t+1)}, \nu_i^{(t+1)}\big)$
    # Iterate
    $t \leftarrow t + 1$

The idea of our third algorithm is to approximate this expectation by particle filtering, i.e. sequential Monte Carlo sampling (Doucet et al., 2000; Gordon et al., 1993) from $\mathbf{P}(C_{1:t}|y_{1:t})$.

We then approximate $\mathbb{b}^{(t)}$ by

$$\hat{\mathbb{b}}^{(t)}(\theta) = \sum_{i=1}^{N} w_t^{(i)} \hat{\mathbb{b}}_i^{(t)}(\theta) = \sum_{i=1}^{N} w_t^{(i)} \mathbf{P}(\Theta_t = \theta | c_{1:t}^{(i)}, y_{1:t}) \,, \tag{2.21}$$

where $\{c_{1:t}^{(i)}\}_{i=1}^N$ is a set of N realizations (or samples) of $c_{1:t}$ (i.e. N particles) drawn from a proposal distribution $\Psi(c_{1:t}|y_{1:t})$, $\{w_t^{(i)}\}_{i=1}^N$ are their corresponding weights at time t, and $\hat{\mathbb{b}}_i^{(t)}(\theta) = \mathbf{P}(\Theta_t = \theta | c_{1:t}^{(i)}, y_{1:t})$ is the approximate belief corresponding to particle i.

Upon observing y_{t+1}, the update procedure for the approximate belief $\hat{\mathbb{b}}^{(t+1)}$ of Equation 2.21 includes two steps: (i) updating the weights, and (ii) sampling the new hidden state c_{t+1} for each particle. The two steps are coupled together through the choice of the proposal distribution Ψ, for which, we choose the optimal proposal function (Doucet et al., 2000) (see Methods). As a result, given this choice of proposal function, we show (see Methods) that the first step amounts to

$$\begin{aligned} w_{t+1}^{(i)} &= (1 - \gamma_{t+1}) w_{B,t+1}^{(i)} + \gamma_{t+1} w_t^{(i)} \,, \\ w_{B,t+1}^{(i)} &= \frac{P(y_{t+1}; \hat{\mathbb{b}}_i^{(t)})}{P(y_{t+1}; \hat{\mathbb{b}}^{(t)})} w_t^{(i)} \,, \end{aligned} \tag{2.22}$$

where $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mathbb{b}}^{(t)}), m\big)$ with $m = \frac{p_c}{1-p_c}$ (cf. Equation 2.9), and $\{w_{B,t+1}^{(i)}\}_{i=1}^N$ are the weights corresponding to the Bayesian update $\hat{\mathbb{b}}_B^{(t+1)}$ of Equation 2.8 (see Methods). In the second step, we update each particle's history of change points by going from the sequence $\{c_{1:t}^{(i)}\}_{i=1}^N$ to $\{c_{1:t+1}^{(i)}\}_{i=1}^N$, for which we always keep the old sequence up to time t, and for each particle i, we add a new element $c_{t+1}^{(i)} \in \{0, 1\}$ representing no change $c_{1:t+1}^{(i)} = [c_{1:t}^{(i)}, 0]$ or change $c_{1:t+1}^{(i)} = [c_{1:t}^{(i)}, 1]$. Note, however, that it is not needed to keep the whole sequences $c_{1:t+1}^{(i)}$ in memory, but instead one can use $c_{t+1}^{(i)}$ to update $\hat{\mathbb{b}}_i^{(t)}$ to $\hat{\mathbb{b}}_i^{(t+1)}$. We sample the new element $c_{t+1}^{(i)}$ from the optimal proposal distribution $\Psi(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1})$ (Doucet et al., 2000), which is given by (see Methods)

$$\Psi(c_{t+1}^{(i)} = 1 | c_{1:t}^{(i)}, y_{1:t+1}) = \gamma\Big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mathbb{b}}_i^{(t)}), \frac{p_c}{1-p_c}\Big) \,. \tag{2.23}$$

Interestingly, the above formulas entail the same surprise modulation and the same trade-off as proposed by the Proposition Equation 2.10. For the weight update, there is a trade-off between an exact Bayesian update and keeping the value of the previous time step, controlled by a adaptation rate modulated exactly in the same way as in Equation 2.10. Note that in contrast to Equation 2.10, the trade-off for the particles'

weights is not between forgetting and integrating, but between maintaining the previous knowledge and integrating. However, the change probability (Equation 2.23) for sampling is equal to the adaptation rate and is an increasing function of surprise. As a result, although the weights are updated less for surprising events, a higher surprise causes a higher probability for change, indicated by $c_{t+1}^{(i)} = 1$, which implies forgetting, because for a particle i with $c_{t+1}^{(i)} = 1$, the associated belief $\hat{\mathbb{b}}_i^{(t+1)} = \mathbf{P}(\Theta_{t+1} = \theta | c_{t+1}^{(i)} = 1, c_{1:t}^{(i)}, y_{1:t+1})$ is equal to $\mathbf{P}(\Theta_{t+1} = \theta | c_{t+1}^{(i)} = 1, y_{t+1}) = P(\theta | y_{t+1})$ (see Fig. 2.1A), which is equivalent to a reset of the belief as in Equation 4.6. In other words, while in MPN and the exact Bayesian inference in Proposition Equation 2.10, the trade-off between integration and reset is accomplished by adding at each time step a new particle with weight γ_{t+1}, in Particle Filtering, it is accomplished via sampling. As a conclusion, the above formulas are essentially the same as the update rules of MPN (c.f. Equation 2.19 and Equation 2.18) and have the same spirit as the recursive update of the Proposition Equation 2.10.

Equations 2.21 and 2.22 can be applied to the case where the likelihood function $P_Y(y|\theta)$ is in the exponential family and $\mathbb{b}^{(0)}$ is its conjugate prior. The resulting algorithm (Algorithm 3) has a particularly simple update rule for the belief parameters (see Methods for details).

The theory of particle filter methods is well established (Doucet et al., 2000; Gordon et al., 1993; Särkkä, 2013). Particle filters in simpler (Brown and Steyvers, 2009) or more complex (Findling et al., 2019) forms have also been employed to explain human behaviour. Here we derived a simple particle filter for the general case of generative models of Equation 2.2, Equation 2.3, and Equation 2.4. Our main contribution is to show that the use of the optimal proposal distribution in this particle filter leads to a surprise-based update scheme.

Surprise-modulation as a framework for other algorithms

Other existing algorithms (Adams and MacKay, 2007; Faraji et al., 2018; Fearnhead and Liu, 2007; Nassar et al., 2010, 2012) can also be formulated in the surprise-modulation framework of Equation 2.9 and Equation 4.6 (see Methods). Moreover, in order to allow for a transparent discussion and for fair comparisons in simulations, we extended the algorithms of Nassar et al. (2010, 2012) to a more general setting. Here we give a brief summary of the algorithms we considered. A detailed analysis is provided in subsection "Surprise-modulation as a framework for other algorithms" in the Methods.

The algorithms of Nassar et al. (2010, 2012) were originally designed for a Gaussian estimation task (see Simulations for details of the task) with a broad uniform prior. We extended them to the more general case of Gaussian tasks with Gaussian priors, and we call our extended versions Nas10* and Nas12* for Nassar et al. (2010) and Nassar et al. (2012) respectively (for a performance comparison between our extended algorithms and

Algorithm 3 Pseudocode for Particle Filtering (exponential family)

Specify $P_Y(y|\theta)$, $\mathbf{P}_{\mathbb{b}}\big(\Theta = \theta; \chi, \nu\big)$, and $\phi(y)$
where $P_Y \in \{\text{exponential family}\}$, $\mathbf{P}_{\mathbb{b}} \in \{\text{conjugate priors of } P_Y\}$ parametrized by χ and ν, and $\phi(y)$ is the sufficient statistic.
Specify $m = p_c/(1-p_c)$, N, and N_{thrs}
Initialize $\chi^{(0)}$, $\nu^{(0)}$, $w_0^{(i)}$ $\forall i \in \{1...N\}$, and $t \leftarrow 0$.
while the sequence is not finished **do**
    Observe y_{t+1}
    # Surprise per particle i
    for $i \in \{1, ..., N\}$ **do**
        Compute $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mathbb{b}}_i^{(t)})$ using Equation 2.83 with $\chi_i^{(t)}$, $\nu_i^{(t)}$
    # Global surprise
    Compute $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mathbb{b}}^{(t)}) = \big[\sum_{i=1}^{N} w_t^{(i)} [\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mathbb{b}}_i^{(t)})]^{-1}\big]^{-1}$
    # Modulation factor
    Compute $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mathbb{b}}^{(t)}), m\big)$
    # Weight per particle i
    for $i \in \{1, ..., N\}$ **do**
        Compute the Bayesian weight $w_{B,t+1}^{(i)}$ using Equation 2.22
        $\boxed{w_{t+1}^{(i)} \leftarrow (1-\gamma_{t+1}) w_{B,t+1}^{(i)} + \gamma_{t+1} w_t^{(i)}}$
    # Hidden state per particle i
    for $i \in \{1, ..., N\}$ **do**
        Sample $\boxed{c_{t+1}^{(i)} \sim \text{Bernoulli}\big(\gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mathbb{b}}_i^{(t)}), m)\big)}$
    # Resampling
    $N_{\text{eff}} \leftarrow (\sum_{i=1}^{N} w_{t+1}^{(i)^2})^{-1}$
    If $N_{\text{eff}} \leq N_{\text{thrs}}$: resample
    # Updated belief per particle i
    for $i \in \{1, ..., N\}$ **do**
        if $c_{t+1}^{(i)} = 0$ **then**
            $\chi_i^{(t+1)} \leftarrow \chi_i^{(t)} + \phi(y_{t+1})$ and $\nu_i^{(t+1)} \leftarrow \nu_i^{(t)} + 1$
        else
            $\chi_i^{(t+1)} \leftarrow \chi^{(0)} + \phi(y_{t+1})$ and $\nu_i^{(t+1)} \leftarrow \nu^{(0)} + 1$
    # Updated (output) belief
    $\hat{\mathbb{b}}^{(t+1)}(\theta) = \sum_{i=1}^{N} w_{t+1}^{(i)} \mathbf{P}_{\mathbb{b}}\big(\Theta = \theta; \chi_i^{(t+1)}, \nu_i^{(t+1)}\big)$
    # Iterate
    $t \leftarrow t+1$

their original versions see Supplementary Fig. 2.12 and Supplementary Fig. 2.13). Both algorithms have the same surprise-modulation as in our Proposition (Equation 2.10). There are multiple interpretations of the approaches of Nas10* and Nas12* and links to other algorithms. One such link we identify is in relation to Particle Filtering with a single particle (pf1). More specifically, one can show that pf1 behaves in expectation similar to Nas10* and Nas12* (see Methods and Supplementary Material).

To summarize, the algorithms Exact Bayes and SORN come from the field of change-point detection, and whereas the former has high memory demands, the latter has the same memory demands as our algorithms pf$N$ and MPN. The algorithms Nas10*, Nas12*, and SMiLe, on the other hand, come from the human learning literature and are more biologically oriented.

2.2.3 Simulations

With the goal of gaining a better understanding of different approximate algorithms, we evaluated the departure of their performance from the exact Bayesian algorithm in terms of mean squared error (MSE) of Θ_t (see Methods), on two tasks inspired by and closely related to real experiments (Behrens et al., 2007; Maheu et al., 2019; Mars et al., 2008; Nassar et al., 2010, 2012; Ostwald et al., 2012): a Gaussian and a Categorical estimation task.

We compared our three novel algorithms VarSMiLe, Particle Filtering (pf$N$, where N is the number of particles), and Message Passing with finite number of particles N (MPN) to the online exact Bayesian Message Passing algorithm (Adams and MacKay, 2007) (Exact Bayes), which yields the optimal solution with $\hat{\Theta}_t = \hat{\Theta}_t^{\text{Opt}}$. Furthermore, we included in the comparison the stratified optimal resampling algorithm (Fearnhead and Liu, 2007) (SORN, where N is the number of particles), our variant of Nassar et al. (2010) (Nas10*) and of Nassar et al. (2012) (Nas12*), the Surprise-Minimization Learning algorithm of Faraji et al. (2018) (SMiLe), as well as a simple Leaky Integrator (Leaky - see Methods).

Gaussian estimation task

The task is a generalized version of the experiment of Nassar et al. (2010, 2012). The goal of the agent is to estimate the mean $\theta_t = \mu_t$ of observed samples, which are drawn from a Gaussian distribution with known variance σ^2, i.e. $y_{t+1}|\mu_{t+1} \sim \mathcal{N}(\mu_{t+1}, \sigma^2)$. The mean μ_{t+1} is itself drawn from a Gaussian distribution $\mu_{t+1} \sim \mathcal{N}(0, 1)$ whenever the environment changes. In other words, the task is a special case of the generative model of Equation 2.2, Equation 2.3, and Equation 2.4, with $\mathbb{b}^{(0)}(\mu_t) = \mathcal{N}(\mu_t; 0, 1)$ and $P_Y(y_t|\mu_t) = \mathcal{N}(y_t; \mu_t, \sigma^2)$. An example of the task can be seen in Fig. 2.2A.

We simulated the task for all combinations of $\sigma \in \{0.1, 0.5, 1, 2, 5\}$ and $p_c \in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0001\}$. For each combination of σ and p_c, we first tuned the free parameter of each algorithm, i.e. m for SMiLe and Variational SMiLe, the leak parameter for the Leaky Integrator, and the p_c of Nas10* and Nas12*, by minimizing the MSE on three random initializations of the task. For the Particle Filter (pf$N$), the Exact Bayes, the MPN, and the SORN we empirically checked that the true p_c of the environment was indeed the value that gave the best performance, and we used this value for the simulations. We evaluated the performance of the algorithms on ten different random task instances of 10^5 steps each for $p_c \in \{0.1, 0.05, 0.01, 0.005\}$ and 10^6 steps each for $p_c \in \{0.001, 0.0001\}$ (in order to sample more change points). Note that the parameter σ is not estimated and its actual value is used by all algorithms except the Leaky Integrator.

In Fig. 2.2B we show the $\mathbf{MSE}[\hat{\Theta}_t | R_t = n]$ in estimating the parameter after n steps since the last change point, for each algorithm, computed over multiple changes, for two exemplar task settings. The Particle Filter with 20 particles (pf20), the VarSMiLe and the Nas12* have an overall performance very close to that of the Exact Bayes algorithm (i.e. $\mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}} | R_t = n]$), with much lower memory requirements. VarSMiLe sometimes slightly outperforms the other two early after an environmental change (Fig. 2.2B, right), but shows slightly higher error values at later phases. The MPN algorithm is the closest one to the optimal solution (i.e. Exact Bayes) for low σ (Fig. 2.2B, left), but its performance is much worse for the case of high σ and low p_c (Fig. 2.2B, right). For the Stratisfied Optimal Resampling (SORN) we observe a counter-intuitive behaviour in the regime of low σ; the inclusion of more particles leads to worse performance (Fig. 2.2B, left). At higher σ levels the performance of SOR20 is close to optimal and better than the MP20 in later time steps. This may be due to the fact that the MPN discards particles in a deterministic and greedy way (i.e. the one with the lowest weight), whereas for the SORN there is a component of randomness in the process of particle elimination, which may be important for environments with higher stochasticity.

For the Leaky Integrator we observe a trade-off between good performance in the transient phase and the stationary phase; a fixed leak value cannot fulfill both requirements. The SMiLe rule, by construction, never narrows its belief $\hat{\mathbb{b}}(\theta)$ below some minimal value, which allows it to have a low error immediately after a change, but leads later to high errors. Its performance deteriorates for higher σ (Fig. 2.2B, right). The Nas10* performs well for low, but not for higher values of σ. Despite the fact that a Particle Filter with 1 particle (pf1) is in expectation similar to Nas10* and Nas12* (see Methods), it performs worse than these two algorithms on trial-by-trial measures. Still, it performs better than the MP1 and identically to the SOR1.

In Fig. 2.3A, we have plotted the average of $\mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}]$ of the Exact Bayes algorithm over the whole simulation time for each of the considered σ and p_c levels. The difference between the other algorithms and this benchmark is called $\Delta\mathbf{MSE}[\hat{\Theta}_t]$ (see Methods) and is plotted in Fig. 2.3C–F. All algorithms except for the SOR20 have lower average error

values for low σ and low p_c, than high σ and high p_c. The Particle Filter pf20 and the Message Passing MP20 have the smallest difference from the optimal solution. The average error of MP20 is higher than that of pf20 for high σ and low p_c, whereas pf20 is more robust across levels of environmental parameters. The worst case performance for pf20 is $\mathbf{\Delta MSE}[\hat{\Theta}_t] = 0.033$ for $\sigma = 5$ and $p_c = 0.0001$, and for SOR20 it is $\mathbf{\Delta MSE}[\hat{\Theta}_t] = 0.061$ for $\sigma = 0.1$ and $p_c = 0.1$. The difference between these two worst case scenarios is significant (p-value $= 2.79 \times 10^{-6}$, two-sample t-test, 10 random seeds for each algorithm). Next in performance is the algorithm Nas12* and VarSMiLe. VarSMiLe exhibits its largest deviation from the optimal solution for high σ and low p_c, but is still more resilient compared to the MPN algorithms for this type of environments. Among the algorithms with only one unit of memory demands, i.e. pf1, MP1, SOR1, VarSMiLe, SMiLe, Leaky, Nas10* and Nas12*, the winners are VarSMiLe and Nas12*. The SOR20 has low error overall, but unexpectedly high error for environmental settings that are presumably more relevant for biological agents (intervals of low stochasticity marked by abrupt changes). The simple Leaky Integrator performs well at low σ and p_c but deviates more from the optimal solution as these parameters increase (Fig. 2.3F). The SMiLe rule performs best at lower σ, i.e. in more deterministic environments.

A summary graph, where we collect the $\mathbf{\Delta MSE}[\hat{\Theta}_t]$ across all levels of σ and p_c, is shown in Fig. 2.4. We can see that pf20, Nas12*, and VarSMiLe give the lowest worst case (lowest maximum value) $\mathbf{\Delta MSE}[\hat{\Theta}_t]$ and are statistically better than the other 8 algorithms (the errorbars indicate the standard error of the mean across the ten random task instances).

Categorical estimation task

The task is inspired by the experiments of Behrens et al. (2007); Maheu et al. (2019); Mars et al. (2008); Ostwald et al. (2012). The goal of the agent is to estimate the occurrence probability of five possible states. Each observation $y_{t+1} \in \{1, ..., 5\}$ is drawn from a categorical distribution with parameters $\theta_{t+1} = \boldsymbol{p}_{t+1}$, i.e. $y_{t+1}|\boldsymbol{p}_{t+1} \sim \text{Cat}(y_{t+1}; \boldsymbol{p}_{t+1})$. When there is a change $C_{t+1} = 1$ in the environment, the parameters $\boldsymbol{p}_{t+1}$ are drawn from a Dirichlet distribution $\text{Dir}(s \cdot \mathbf{1})$, where $s \in (0, \infty)$ is the stochasticity parameter. In relation to the generative model of Equation 2.2, Equation 2.3, and Equation 2.4 we, thus, have $\mathbb{b}^{(0)}(\boldsymbol{p}_t) = \text{Dir}(\boldsymbol{p}_t; s \cdot \mathbf{1})$ and $P_Y(y_t|\boldsymbol{p}_t) = \text{Cat}(y_t; \boldsymbol{p}_t)$. An illustration of this task is depicted in Fig. 2.5A.

We considered the combinations of stochasticity levels $s \in \{0.01, 0.1, 0.14, 0.25, 1, 2, 5\}$ and change probability levels $p_c \in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0001\}$. The algorithms of Nassar et al. (2010, 2012) were specifically developed for a Gaussian estimation task and cannot be applied here. All other algorithms were first optimized for each combination of environmental parameters before an experiment starts, and then evaluated on ten different random task instances, for 10^5 steps each for $p_c \in \{0.1, 0.05, 0.01, 0.005\}$ and 10^6 steps each for $p_c \in \{0.001, 0.0001\}$. The parameter s is not estimated and its actual

value is used by all algorithms except the Leaky Integrator.

The Particle Filter pf20, the MP20 and the SOR20 have a performance closest to that of Exact Bayes, i.e. the optimal solution (Fig. 2.5B). VarSMiLe is the next in the ranking, with a behaviour after a change similar to the Gaussian task. pf20 performs better for $s > 2$ and MP20 performs better for $s \leq 2$ (Fig. 2.6). For this task the biologically less plausible SOR20 is the winner in performance and it behaves most consistently across environmental parameters. Its worst case performance is $\Delta\mathbf{MSE}[\hat{\Theta}_t] = 8.16 \times 10^{-5}$ for $s = 2$ and $p_c = 0.01$, and the worst case performance for pf20 is $\Delta\mathbf{MSE}[\hat{\Theta}_t] = 0.0048$ for $s = 0.25$ and $p_c = 0.005$ (p-value $= 1.148 \times 10^{-12}$, two-sample t-test, 10 random seeds for each algorithm). For all the other algorithms, except for MP20, the highest deviations from the optimal solution are observed for medium stochasticity levels (Fig. 2.6B–F). When the environment is nearly deterministic (e.g. $s = 0.001$ so that the parameter vectors $\boldsymbol{p}_t$ have almost all mass concentrated in one component), or highly stochastic (e.g. $s > 1$ so that nearly uniform categorical distributions are likely to be sampled), these algorithms achieve higher performance, while the Particle Filter is the algorithm that is most resilient to extreme choices of the stochasticity parameter s. For VarSMiLe in particular, the lowest mean error is achieved for high s and high p_c or low s and low p_c.

A summary graph, with the $\Delta\mathbf{MSE}[\hat{\Theta}_t]$ across all levels of s and p_c, can be seen in Fig. 2.7. The algorithms with the lowest "worst case" are SOR20 and pf20. The top-4 algorithms SOR20, pf20, MP20 and VarSMiLe are significantly better than the others (the errorbars indicate the standard error of the mean across the ten random task instances), whereas MP1 and SMiLe have the largest error with a maximum at 0.53.

Summary of simulation results

In summary, our simulation results of the two tasks collectively suggest that our Particle Filtering (pf$N$) and Message Passing (MPN) algorithms achieve a high level of performance, very close to the one of biologically less plausible algorithms with higher (Exact Bayes) and same (SORN) memory demands. Moreover, their behaviour is more consistent across tasks. Finally, among the algorithms with memory demands of one unit, VarSMiLe performs best.

Robustness against suboptimal parameter choice

In all algorithms we considered, the environment's hyper-parameters are assumed to be known. We can distinguish between two types of hyper-parameters in our generative model: 1. the parameters of the likelihood function (e.g. σ in the Gaussian task), and 2. the p_c and the parameters of the conjugate prior (e.g. s in the Categorical task). Hyper-parameters of the first type can be added to the parameter vector θ and be inferred with the same algorithm. However, learning the second type of hyper-parameters is not

straightforward. By assuming that these hyper-parameters are learned more slowly than θ, one can fine-tune them after each n (e.g. 10) change points, while change points can be detected by looking at the particles for the Particle Filter and at the peaks of surprise values for VarSMiLe. Other approaches to hyper-parameter estimation can be found in Doucet and Tadić (2003); George and Doss (2017); Liu and West (2001); Wilson et al. (2010).

When the hyper-parameters are fixed, a mismatch between the assumed values and the true values is a possible source of errors. In this section, we investigate the robustness of the algorithms to a mismatch between the assumed and the actual probability of change points. To do so, we first tuned each algorithm's parameter for an environment with a change probability p_c, and then tested the algorithms in environments with different change probabilities, while keeping the parameter fixed. For each new environment with a different change probability, we calculated the difference between the MSE of these fixed parameters and the optimal MSE, i.e. the resulting MSE for the case that the Exact Bayes' parameter is tuned for the actual p_c.

More precisely, if we denote as $\mathbf{MSE}[\hat{\Theta}_t; p_c', p_c]$ the MSE of an algorithm with parameters tuned for an environment with p_c', applied in an environment with p_c, we calculated the mean regret, defined as $\mathbf{MSE}[\hat{\Theta}_t; p_c', p_c] - \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}, p_c]$ over time; note that the second term is equal to $\mathbf{MSE}[\hat{\Theta}_t; p_c, p_c]$ when the algorithm Exact Bayes is used for estimation. The lower the values and the flatter the curve of the mean regret, the better the performance and the robustness of the algorithm in the face of lacking knowledge of the environment. The slope of the curve indicates the degree of deviations of the performance as we move away from the optimally tuned parameter. We ran three random (and same for all algorithms) tasks initializations for each p_c level.

In Fig. 2.8 we plot the mean regret for each algorithm for the Gaussian task for four pairs of s and p_c' levels. For $\sigma = 0.1$ and $p_c' = 0.04$ (Fig. 2.8A) the Exact Bayes and the MP20 show the highest robustness (smallest regret) and are closely followed by the pf20, VarSMiLe, and Nas12* (note the regret's small range of values). The lower the actual p_c, the higher the regret, but still the changes are very small. The curves for the SMiLe and the Leaky Integrator are also relatively flat, but the mean regret is much higher. The SOR20 is the least robust algorithm.

Similar observations can be made for $\sigma = 0.1$ and $p_c' = 0.004$ (Fig. 2.8B). In this case, the performance of all algorithms deteriorates strongly when the actual p_c is higher than the assumed one.

However, for $\sigma = 5$ (Fig. 2.8C and Fig. 2.8D), the ranking of algorithms changes. The SOR20 is very robust for this level of stochasticity. The pf20 and MP20 perform similarly for $p_c = 0.04$, but for lower p_c' the pf20 is more robust and the MP20 exhibits high fluctuations in its performance. The Nas12* is quite robust at this σ level. Overall for

Exact Bayes, SOR20, pf20, VarSMiLe and Nas12*, a mismatch of the assumed p_c from the actual one does not deteriorate the performance dramatically for $\sigma = 5$, $p'_c = 0.004$ (Fig. 2.8D). The SMiLe and the Leaky Integrator outperform the other algorithms for higher p'_c if $p_c < p'_c$ (Fig. 2.8C). A potential reason is that the optimal behaviour for the Leaky Integrator (according to the tuned parameters) is to constantly integrate new observations into its belief (i.e. to act like a Perfect Integrator) regardless of the p'_c level. This feature makes it blind to the p_c and therefore very robust against the lack of knowledge of it (Fig. 2.8C).

In summary, most of the time, the mean regret for Exact Bayes and MP20 is less than the mean regret for pf20 and VarSMiLe. However, the variability in the mean regret for pf20 and VarSMiLe is smaller, and their curves are flatter across p_c levels, which makes their performance more predictable. The results for the Categorical estimation task are similar to those of the Gaussian task, with the difference that the SOR20 is very robust for this case (Fig. 2.9).

2.2.4 Experimental prediction

It has been experimentally shown that some important behavioural and physiological indicators statistically correlate with a measure of surprise or a prediction error. Examples of such indicators are the pupil diameter (Joshi and Gold, 2019; Nassar et al., 2012; Preuschoff et al., 2011), the amplitude of the P300, N400, and MMN components of EEG (Kopp and Lange, 2013; Lieder et al., 2013; Mars et al., 2008; Meyniel et al., 2016; Modirshanechi et al., 2019; Musiolek et al., 2019; Ostwald et al., 2012), the amplitude of MEG in specific time windows (Maheu et al., 2019), BOLD responses in fMRI (Konovalov and Krajbich, 2018; Loued-Khenissi et al., 2020), and reaction time (Huettel et al., 2002; Meyniel et al., 2016). The surprise measure is usually the negative log-probability of the observation, known as Shannon Surprise (Shannon, 1948), and denoted here as $\mathbf{S}_{\mathrm{Sh}}$. However, as we show in this section, as long as there is an uninformative prior over observations, Shannon Surprise $\mathbf{S}_{\mathrm{Sh}}$ is just an invertible function of our modulated adaptation rate γ and hence an invertible function of the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}$. Thus, based on the results of previous works (Meyniel et al., 2016; Modirshanechi et al., 2019; Nassar et al., 2010, 2012; Ostwald et al., 2012), that always used uninformative priors, one cannot determine whether the aforementioned physiological and behavioural indicators correlate with $\mathbf{S}_{\mathrm{Sh}}$ or $\mathbf{S}_{\mathrm{BF}}$.

In this section, we first investigate the theoretical differences between the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}$ and Shannon Surprise $\mathbf{S}_{\mathrm{Sh}}$. Then, based on their observed differences, we formulate two experimentally testable predictions, with a detailed experimental protocol. Our predictions make it possible to discriminate between the two measures of surprise, and to determine whether physiological or behavioural measurements are signatures of $\mathbf{S}_{\mathrm{BF}}$ or of $\mathbf{S}_{\mathrm{Sh}}$.

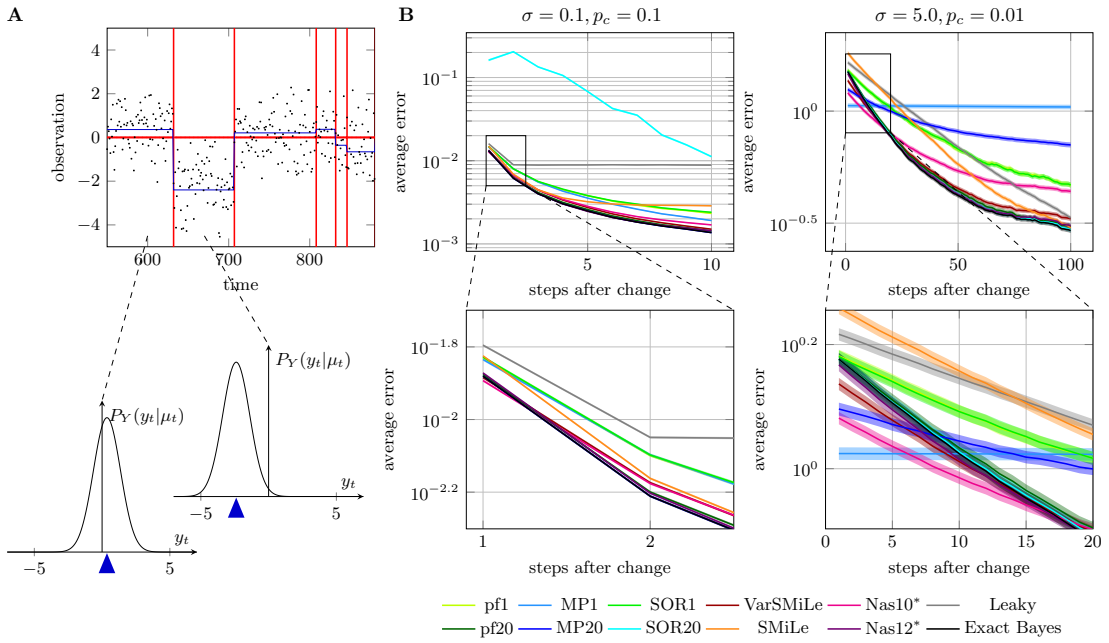


Figure 2.2 – **Gaussian estimation task: Transient performance after changes. A**. At each time step an observation (depicted as black dot) is drawn from a Gaussian distribution $\sim \exp(-(y_t - \mu_t)^2/2\sigma^2)$ with changing mean μ_t (marked in blue) and known variance σ^2 (lower left panels). At every change of the environment (marked with red lines) a new mean μ_t is drawn from a standard Gaussian distribution $\sim \exp(-\mu_t^2)$. In this example: $\sigma = 1$ and $p_c = 0.01$. **B**. Mean squared error for the estimation of μ_t at each time step n after an environmental change, i.e. the average of $\mathbf{MSE}[\hat{\Theta}_t|R_t = n]$ over time; $\sigma = 0.1$, $p_c = 0.1$ (left panel) and $\sigma = 5$, $p_c = 0.01$ (right panel). The shaded area corresponds to the standard error of the mean. *Abbreviations:* pf$N$: Particle Filtering with N particles, MPN: Message Passing with N particles, VarSMiLe: Variational SMiLe, SORN: Stratisfied Optimal Resampling with N particles (Fearnhead and Liu, 2007), SMiLe: Faraji et al. (2018), Nas10*, Nas12*: Variants of Nassar et al. (2010) and Nassar et al. (2012), respectively, Leaky: Leaky Integrator, Exact Bayes: Adams and MacKay (2007).

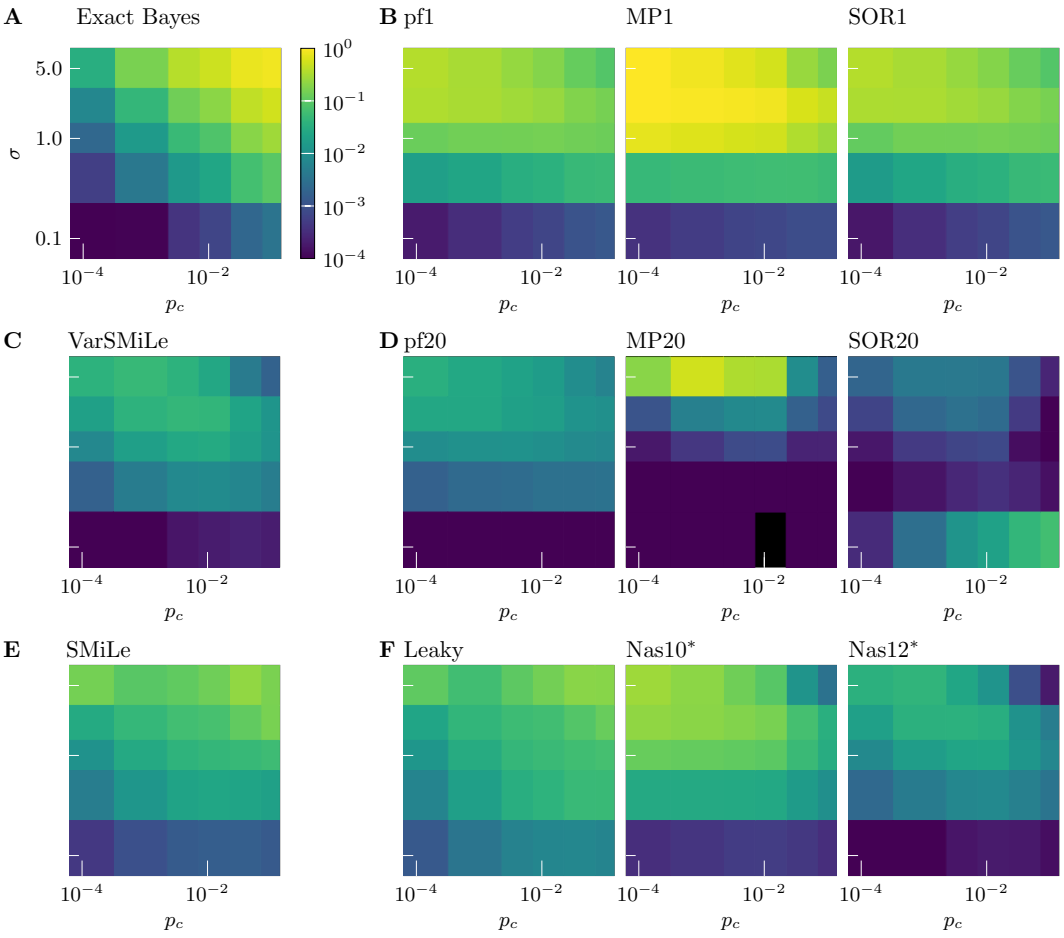


Figure 2.3 – **Gaussian estimation task: Steady-state performance.** **A**. Mean squared error of the Exact Bayes algorithm (i.e. optimal solution) for each combination of σ and p_c averaged over time. **B** – **F**. Difference between the mean squared error of each algorithm and the optimal solution (of panel A), i.e. the average of $\Delta\mathbf{MSE}[\hat{\Theta}_t]$ over time. The colorbar of panel A applies to these panels as well. Note that the black color for the MP20 indicates negative values, which are due to the finite sample size for the estimation of **MSE**. *Abbreviations:* See the caption of Fig. 2.2.

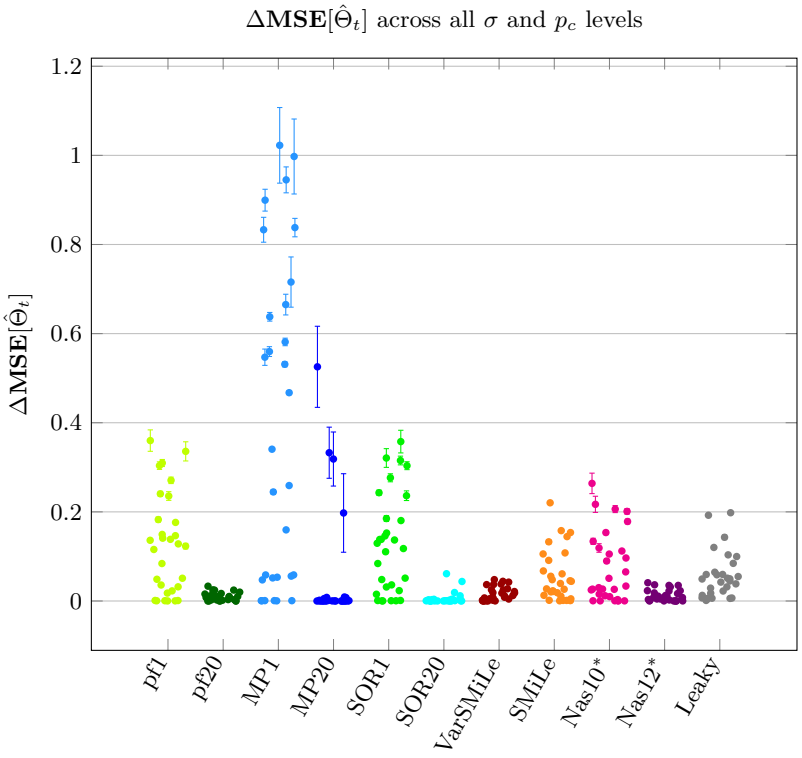


Figure 2.4 – **Gaussian estimation task: Steady-state performance summary.** Difference between the mean squared error of each algorithm and the optimal solution (Exact Bayes), i.e. the average of $\Delta\mathbf{MSE}[\hat{\Theta}_t]$ over time, for all combinations of σ and p_c together. For each algorithm we plot the 30 values (5 σ times 6 p_c values) of Fig. 2.3 with respect to randomly jittered values in the x-axis. The color coding is the same as in Fig. 2.2. The errorbars mark the standard error of the mean across 10 random task instances. The difference between the worst case of SOR20 and pf20 is significant (p-value $= 2.79 \times 10^{-6}$, two-sample t-test, 10 random seeds for each algorithm). *Abbreviations:* See the caption of Fig. 2.2.

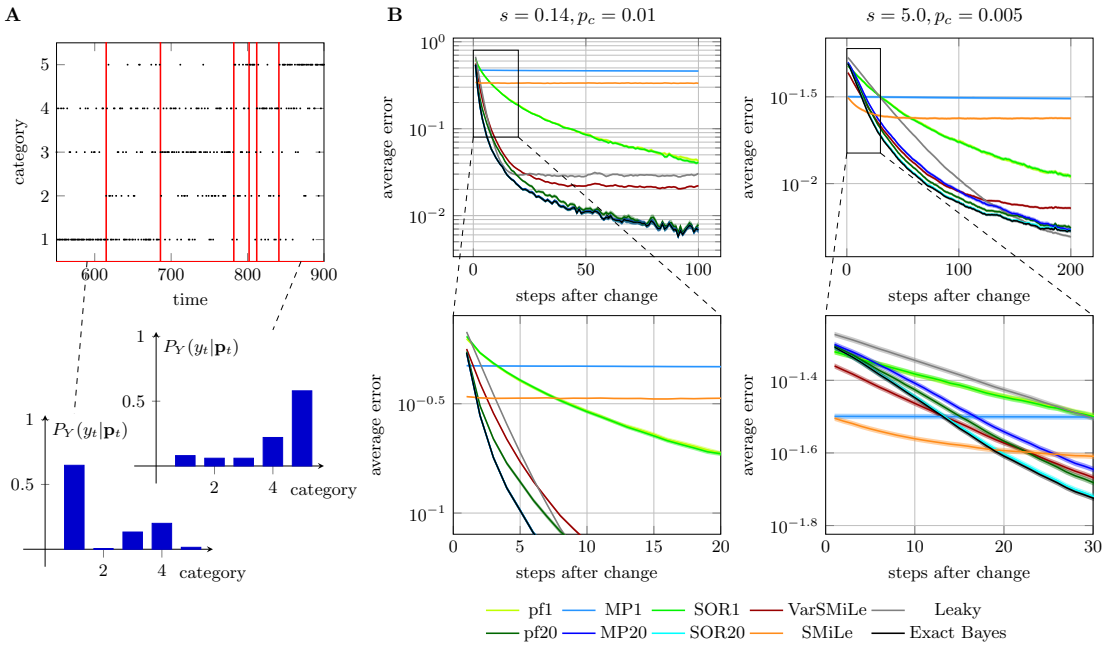


Figure 2.5 – **Categorical estimation task: Transient performance after changes.** **A**. At each time step the agent sees one out of 5 possible categories (black dots) drawn from a categorical distribution with parameters $\boldsymbol{p_t}$. Occasional abrupt changes happen with probability p_c and are marked with red lines. After each change a new $\boldsymbol{p_t}$ vector is drawn from a Dirichlet distribution with stochasticity parameter s. In this example: $s = 1$ and $p_c = 0.01$. **B**. Mean squared error for the estimation of $\boldsymbol{p_t}$ at each time step n after an environmental change, i.e. the average of $\mathbf{MSE}[\hat{\Theta}_t|R_t = n]$ over time; $s = 0.14$, $p_c = 0.01$ (left panel) and $s = 5$, $p_c = 0.005$ (right panel). The shaded area corresponds to the standard error of the mean. *Abbreviations:* pf$N$: Particle Filtering with N particles, MPN: Message Passing with N particles, VarSMiLe: Variational SMiLe, SORN: Stratisfied Optimal Resampling with N particles (Fearnhead and Liu, 2007), SMiLe: Faraji et al. (2018), Leaky: Leaky Integrator, Exact Bayes: Adams and MacKay (2007).

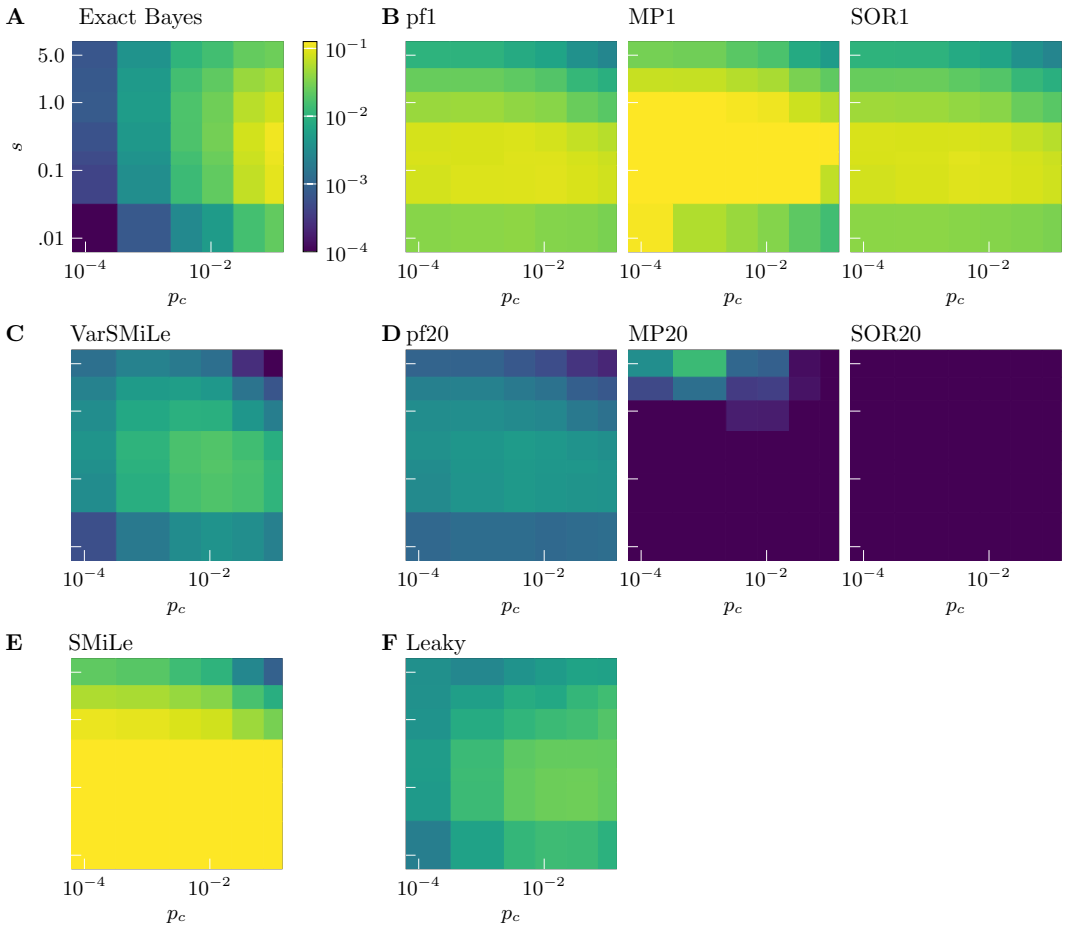


Figure 2.6 – **Categorical estimation task: Steady-state performance. A**. Mean squared error of the Exact Bayes algorithm (i.e. optimal solution) for each combination of environmental parameters s and p_c averaged over time. **B** – **F**. Difference between the mean squared error of each algorithm and the optimal solution (of panel A), i.e. the average of $\Delta\mathbf{MSE}[\hat{\Theta}_t]$ over time. The colorbar of panel A applies to these panels as well. *Abbreviations:* See the caption of Fig. 2.5.

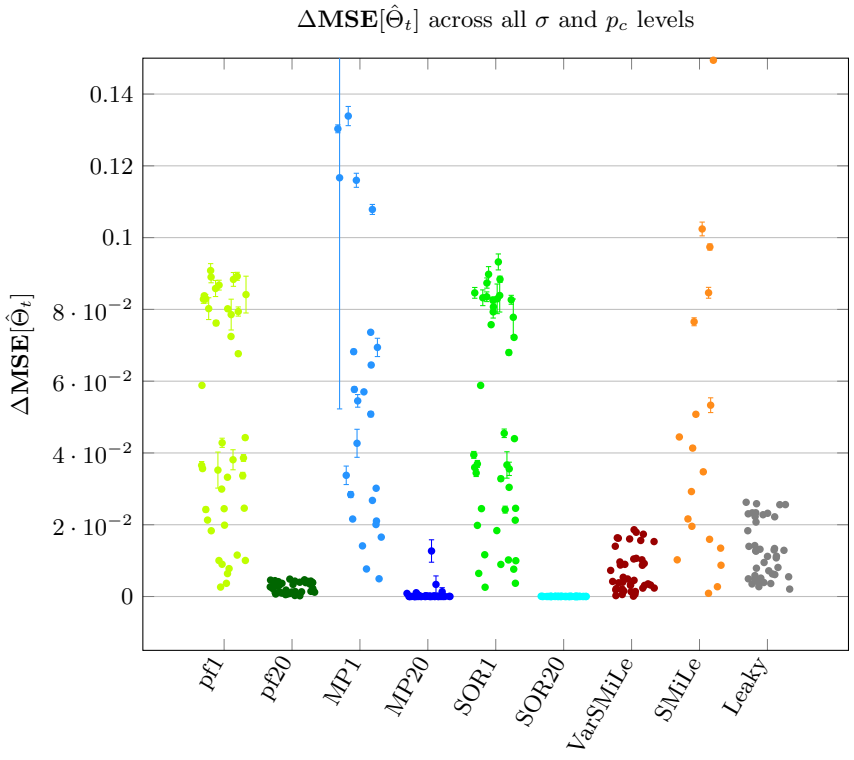


Figure 2.7 – **Categorical estimation task: Steady-state performance summary.** Difference between the mean squared error of each algorithm and the optimal solution (Exact Bayes), i.e. the average of $\Delta\mathbf{MSE}[\hat{\Theta}_t]$ over time, for all combinations of s and p_c together. For each algorithm we plot the 42 values (7 s times 6 p_c values) of Fig. 2.6 with respect to randomly jittered values in the x-axis. The color coding is the same as in Fig. 2.5. The errorbars mark the standard error of the mean across 10 random task instances. The difference between the worst case of SOR20 and pf20 is significant (p-value $= 1.148 \times 10^{-12}$, two-sample t-test, 10 random seeds for each algorithm). *Abbreviations:* See the caption of Fig. 2.5. Note that MP1 and SMiLe are out of bound with a maximum at 0.53.

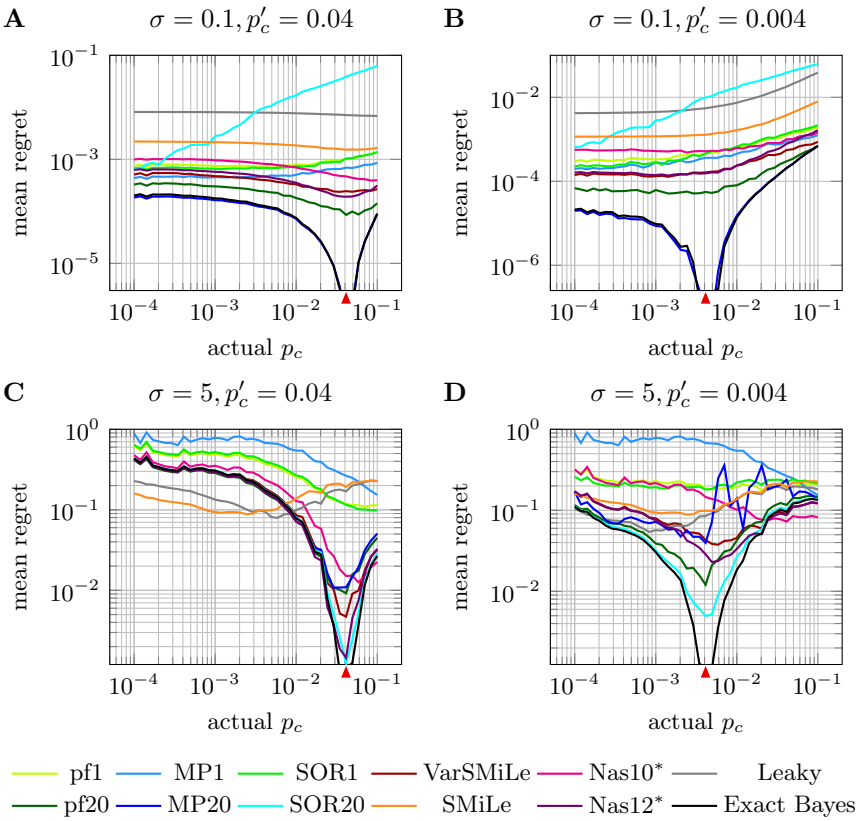


Figure 2.8 – **Robustness to mismatch between actual and assumed probability of changes for the Gaussian estimation task.** The mean regret is the mean squared error obtained with assumed change probability p'_c minus the mean squared error obtained with the optimal parameter choice of Exact Bayes for the given actual p_c, i.e. the average of the quantity $\mathbf{MSE}[\hat{\Theta}_t; p'_c, p_c] - \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}, p_c]$ over time versus. A red triangle marks the p'_c value each algorithm was tuned for. We plot the mean regret for the following parameter combinations: **A**. $\sigma = 0.1$ and $p'_c = 0.04$, **B**. $\sigma = 0.1$ and $p'_c = 0.004$, **C**. $\sigma = 5$ and $p'_c = 0.04$, **D**. $\sigma = 5$ and $p'_c = 0.004$. *Abbreviations:* See the caption of Fig. 2.2.

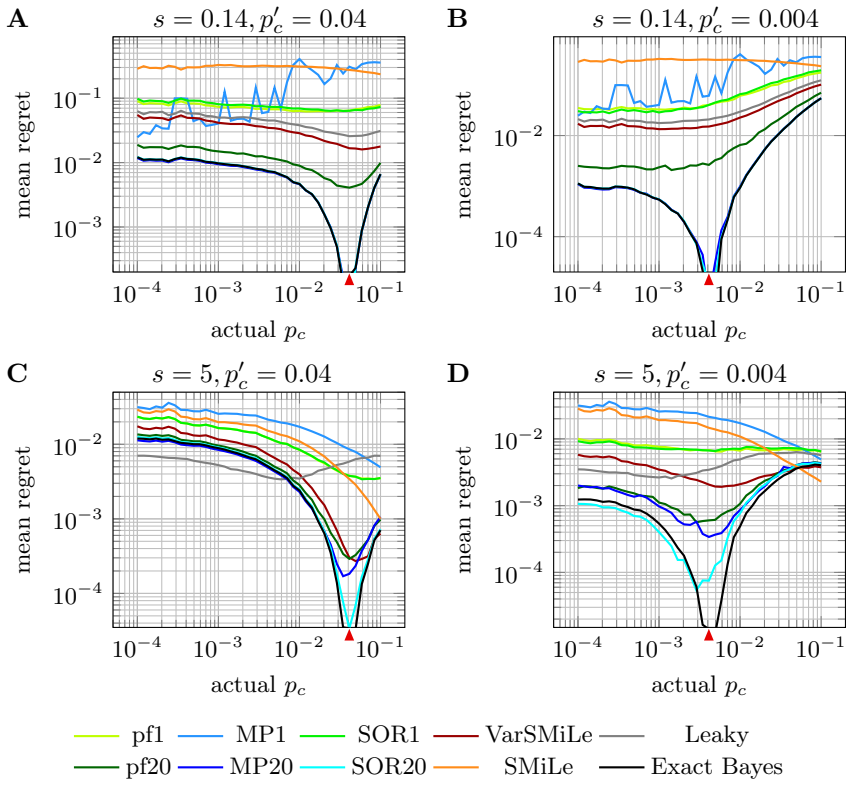


Figure 2.9 – **Robustness to mismatch between actual and assumed probability of changes for the Categorical estimation task.** The mean regret is the mean squared error obtained with assumed change probability p'_c minus the mean squared error obtained with the optimal parameter choice of Exact Bayes for the given actual p_c, i.e. the average of the quantity $\mathbf{MSE}[\hat{\Theta}_t; p'_c, p_c] - \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}, p_c]$ over time. A red triangle marks the p'_c value each algorithm was tuned for. We plot the mean regret for the following parameter combinations: **A**. $s = 0.14$ and $p'_c = 0.04$, **B**. $s = 0.14$ and $p'_c = 0.004$, **C**. $s = 5$ and $p'_c = 0.04$, **D**. $s = 5$ and $p'_c = 0.004$. *Abbreviations:* See the caption of Fig. 2.5

Theoretical difference between $\mathbf{S}_{\mathrm{BF}}$ and $\mathbf{S}_{\mathrm{Sh}}$

Shannon Surprise (Shannon, 1948) is defined as

$$\mathbf{S}_{\mathrm{Sh}}(y_{t+1};\mathbb{b}^{(t)}) = -\log\Big(\mathbf{P}(y_{t+1}|y_{1:t})\Big) \tag{2.24}$$

where for computing $\mathbf{P}(y_{t+1}|y_{1:t})$, one should know the structure of the generative model. For the generative model of Fig. 2.1A, we find $\mathbf{S}_{\mathrm{Sh}}(y_{t+1};\mathbb{b}^{(t)}) = -\log\Big((1-p_c)P(y_{t+1};\mathbb{b}^{(t)}) + p_c P(y_{t+1};\mathbb{b}^{(0)})\Big)$. While the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}$ depends on a ratio between the probability of the new observation under the prior and the current beliefs, Shannon Surprise depends on a weighted sum of these probabilities. Interestingly, it is possible to express (see Methods for derivation) the adaptation rate γ_{t+1} as a function of the "difference in Shannon Surprise"

$$\begin{aligned} &\gamma_{t+1} = p_c \exp\Big(\Delta\mathbf{S}_{\mathrm{Sh}}(y_{t+1};\mathbb{b}^{(t)},\mathbb{b}^{(0)})\Big), \\ &\text{where } \Delta\mathbf{S}_{\mathrm{Sh}}(y_{t+1};\mathbb{b}^{(t)},\mathbb{b}^{(0)}) = \mathbf{S}_{\mathrm{Sh}}(y_{t+1};\mathbb{b}^{(t)}) - \mathbf{S}_{\mathrm{Sh}}(y_{t+1};\mathbb{b}^{(0)}), \end{aligned} \tag{2.25}$$

where $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}^{(t+1)}, m\big)$ depends on the Bayes Factor Surprise and the saturation parameter m (cf. Equation 2.9). Equation 2.25 shows that the modulated adaptation rate is not just a function of Shannon Surprise upon observing y_{t+1}, but a function of the *difference* between the Shannon Surprise of this observation under the current and under the prior beliefs. In the next subsections, we exploit differences between $\mathbf{S}_{\mathrm{BF}}$ and $\mathbf{S}_{\mathrm{Sh}}$ to formulate our experimentally testable predictions.

Experimental protocol

Consider the variant of the Gaussian task of Nassar et al. (2010, 2012) which we used in our simulations, i.e. $P_Y(y|\theta) = \mathcal{N}(y;\theta,\sigma^2)$ and $\mathbb{b}^{(0)}(\theta) = \mathcal{N}(\theta;0,1)$. Human subjects are asked to predict the next observation y_{t+1} given what they have observed so far, i.e. $y_{1:t}$. The experimental procedure is as follows:

1. Fix the hyper parameters σ^2 and p_c.
2. At each time t, show the observation y_t (produced in the aforementioned way) to the subject, and measure a physiological or behavioural indicator M_t, e.g. pupil diameter (Nassar et al., 2010, 2012).
3. At each time t, after observing y_t, ask the subject to predict the next observation $\hat{y}_{t+1}$ and their confidence C_t about their prediction.

Note that the only difference between our task and the task of Nassar et al. (2010, 2012) is the choice of prior for θ (i.e. Gaussian instead of uniform). The assumption is that,

according to the previous studies, there is a *positive* correlation between M_t and a measure of surprise.

Prediction 1

Based on the results of Nassar et al. (2010, 2012), in such a Gaussian task, the best fit for subjects' prediction $\hat{y}_{t+1}$ is $\hat{\theta}_t$, and the confidence C_t is a monotonic function of $\hat{\sigma}_t$. In order to formalize our experimental prediction, we define, at time t, the prediction error as $\delta_t = y_t - \hat{y}_t$ and the "sign bias" as $s_t = \text{sign}(\delta_t \hat{y}_t)$. The variable s_t is a crucial variable for our analysis. It shows whether the prediction $\hat{y}_t$ is an overestimation in absolute value ($s_t = +1$) or an underestimation in absolute value ($s_t = -1$). Fig. 2.10A shows a schematic for the case that both the current and prior beliefs are Gaussian distributions. The two observations indicated by dashed lines have same absolute error $|\delta_t|$, but differ in the sign bias s.

Given an absolute prediction value $\hat{y} > 0$, an absolute prediction error $\delta > 0$, a confidence value $C > 0$, and a sign bias $s \in \{-1, 1\}$, we can compute the average of M_t over time for the time points with $|\hat{y}_t| \approx \hat{y}$, $|\delta_t| \approx \delta$, $C_t \approx C$, and $s_t = s$, which we denote as $\bar{M}_1(\hat{y}, \delta, s, C)$ – the index 1 stands for experimental prediction 1. The approximation notation $\approx$ is used for continuous variables instead of equality, due to practical limitations, i.e. for obtaining adequate number of samples for averaging. Note that for our theoretical proofs we use equality, but in our simulation we include the practical limitations of a real experiment, and hence, use an approximation. The formal definitions can be found in Methods. It is worth noting that the quantity $\bar{M}_1(\hat{y}, \delta, s, C)$ is model independent; its calculation does not require any assumption on the learning algorithm the subject may employ. Depending on whether the measurement $\bar{M}_1(\hat{y}, \delta, s, C)$ reflects $\mathbf{S}_{\text{Sh}}$ or $\mathbf{S}_{\text{BF}}$, its relationship to the defined four variables (i.e. $\hat{y}$, δ, s, C) is qualitatively and quantitatively different.

In order to prove and illustrate our prediction, let us consider each subject as an agent enabled with one of the learning algorithms that we discussed. Similar to above, given an absolute prediction $\hat{\theta} > 0$ (corresponding to the subjects' absolute prediction $\hat{y}$), an absolute prediction error $\delta > 0$, a standard deviation σ_C (corresponding to the subjects' confidence value C), and a sign bias $s \in \{-1, 1\}$, we can compute the average Shannon Surprise $\mathbf{S}_{\text{Sh}}(y_t; \hat{\mathbb{b}}^{(t-1)})$ and the average Bayes Factor Surprise $\mathbf{S}_{\text{BF}}(y_t; \hat{\mathbb{b}}^{(t-1)})$ over time, for the time points with $|\hat{\theta}_{t-1}| \approx \hat{\theta}$, $|\delta_t| \approx \delta$, $\hat{\sigma}_t \approx \sigma_C$, and $s_t = s$, which we denote as $\bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, s, \sigma_C)$ and $\bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s, \sigma_C)$ respectively. We can show theoretically (see Methods) and in simulations (see Fig. 2.10B and Methods) that for any value of $\hat{\theta}$, δ, and σ_C, we have $\bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, s = +1, \sigma_C) > \bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, s = -1, \sigma_C)$ for the Shannon Surprise, and exactly the opposite relation, i.e. $\bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C) < \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)$ for the Bayes Factor Surprise. Moreover, this effect increases with increasing δ.

Table 2.1 – **Experimental Hypotheses and Predictions 1.** $\Delta\bar{M}_1(\hat{\theta},\delta,C)$ stands for $\bar{M}_1(\hat{\theta},\delta,s=+1,C)-\bar{M}_1(\hat{\theta},\delta,s=-1,C)$

Hypothesis	**Prediction**
The indicator reflects $\mathbf{S}_{\mathrm{BF}}$	$\Delta\bar{M}_1(\hat{\theta},\delta,C)<0$ and $\frac{\partial\Delta\bar{M}_1(\hat{\theta},\delta,C)}{\partial\delta}<0$
The indicator reflects $\mathbf{S}_{\mathrm{Sh}}$	$\Delta\bar{M}_1(\hat{\theta},\delta,C)>0$ and $\frac{\partial\Delta\bar{M}_1(\hat{\theta},\delta,C)}{\partial\delta}>0$
The prior is not used for inference	$\Delta\bar{M}_1(\hat{\theta},\delta,C)=0$

It should be noted that such an effect is due to the essential difference of $\mathbf{S}_{\mathrm{Sh}}$ and $\mathbf{S}_{\mathrm{BF}}$ in using the prior belief $\mathbb{b}^{(0)}(\theta)$. Our experimental prediction is theoretically provable for the cases that each subject's belief $\hat{\mathbb{b}}^{(t)}$ is a Gaussian distribution, which is the case if they employ VarSMiLe, Nas10*, Nas12*, pf1, MP1, or Leaky Integrator as their learning rule (see Methods). For the cases that different learning rules (e.g. pf20) are used, where the posterior belief is a weighted sum of Gaussians, the theoretical analysis is more complicated, but our simulations show the same results (see Fig. 2.10B and Methods). Therefore, independent of the learning rule, we have the same experimental prediction on the manifestation of different surprise measures on physiological signals, such as pupil dilation. Our first experimental prediction can be summarized as a set of hypotheses shown in Table 2.1.

Prediction 2

Our second prediction follows the same experimental procedure as the one for the first prediction. The main difference is that for the second prediction we need to fit a model to the experimental data. Given one of the learning algorithms, the fitting procedure can be done by tuning the free parameters of the algorithm with the goal of minimizing the mean squared error between the model's prediction $\hat{\theta}_t$ and a subject's prediction $\hat{y}_{t+1}$ (similar to Nassar et al. (2010, 2012)) or with the goal of maximizing the likelihood of subject's prediction $\hat{\mathbb{b}}^{(t)}(\hat{y}_{t+1})$. Our prediction is independent of the learning algorithm, but in an actual experiment, we recommend to use model selection to find the model that fits the human data best.

Having a fitted model, we can compute the probabilities $P(y_{t+1};\hat{\mathbb{b}}^{(t)})$ and $P(y_{t+1};\hat{\mathbb{b}}^{(0)})$. For the case that these probabilities are equal, i.e. $P(y_{t+1};\hat{\mathbb{b}}^{(t)})=P(y_{t+1};\hat{\mathbb{b}}^{(0)})=p$, the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}$ is equal to 1, independent of the value of p (cf. Equation 2.7). However, the Shannon Surprise $\mathbf{S}_{\mathrm{Sh}}$ is equal to $-\log p$, and varies with p. Fig. 2.11A shows a schematic for the case that both current and prior beliefs are Gaussian distributions. Two cases for which we have $P(y_{t+1};\hat{\mathbb{b}}^{(t)})=P(y_{t+1};\hat{\mathbb{b}}^{(0)})=p$, for two different p values, are marked by black dots at the intersections of the curves.

Given a probability $p>0$, we can compute the average of M_t over time for the time points with $P(y_{t+1};\hat{\mathbb{b}}^{(t)})\approx p$ and $P(y_{t+1};\hat{\mathbb{b}}^{(0)})\approx p$, which we denote as $\bar{M}_2(p)$ – the index 2

Table 2.2 – **Experimental Hypotheses and Predictions 2.**

Hypothesis	**Prediction**
The indicator reflects $\mathbf{S}_{\mathrm{BF}}$	$\frac{\partial \bar{M}_2(p)}{\partial p} = 0$
The indicator reflects $\mathbf{S}_{\mathrm{Sh}}$	$\frac{\partial \bar{M}_2(p)}{\partial p} < 0$

stands for experimental prediction 2. Analogous to the first prediction, the approximation notation $\approx$ is used due to practical limitations. Then, if $\bar{M}_2(p)$ is independent of p, its behaviour is consistent with $\mathbf{S}_{\mathrm{BF}}$, whereas if it decreases by increasing p, it can be a signature of $\mathbf{S}_{\mathrm{Sh}}$. Our second experimental prediction can be summarized as two hypotheses shown in Table 2.2. Note that in contrast to our first prediction, with the assumption that the standard deviation of the prior belief is fitted using the behavioural data, we do not consider the hypothesis that the prior is not used for inference, because this is indistinguishable from a very large variance of the prior belief.

In order to illustrate the possible results and the feasibility of the experiment, we ran a simulation and computed $\bar{\mathbf{S}}_{\mathrm{BF}}(p)$ and $\bar{\mathbf{S}}_{\mathrm{Sh}}(p)$ for the time points with $P(y_{t+1}; \hat{\mathbb{b}}^{(t)}) \approx p$ and $P(y_{t+1}; \hat{\mathbb{b}}^{(0)}) \approx p$ (see Methods for details). The results of the simulation are shown in Fig. 2.11B.

2.3 Discussion

We have shown that performing exact Bayesian inference on a generative world model naturally leads to a definition of surprise and a surprise-modulated adaptation rate. We have proposed three approximate algorithms (VarSMiLe, MPN, and pf$N$) for learning in non-stationary environments, which all exhibit the surprise-modulated adaptation rate of the exact Bayesian approach and are biologically plausible. Empirically we observed that our algorithms achieve levels of performance comparable to approximate Bayesian methods with higher memory demands (Adams and MacKay, 2007), and are more resilient across different environments compared to methods with similar memory demands (Faraji et al., 2018; Fearnhead and Liu, 2007; Nassar et al., 2010, 2012).

Learning in a volatile environment has been studied for a long time in the fields of Bayesian learning, neuroscience, and signal processing. In the following, we discuss the biological relevance of our work, and we briefly review some of the previously developed algorithms, with particular focus on the ones that have studied environments which can be modeled with a generative model similar to the one in Fig. 2.1. We then discuss further our results, and propose directions for future work on surprise-based learning.

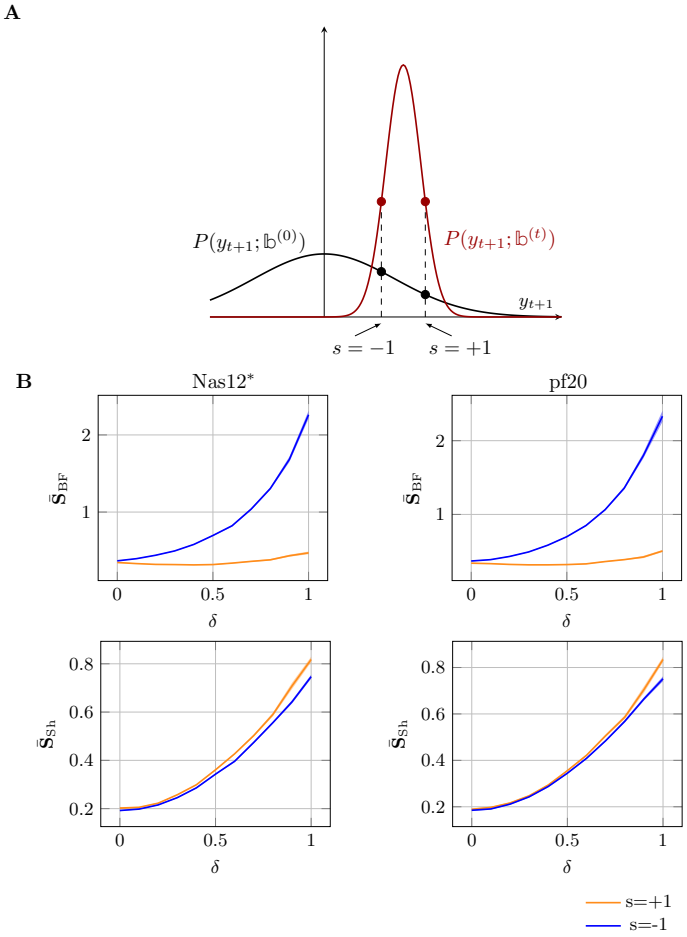


Figure 2.10 – **Experimental prediction 1. A.** Schematic of the task for the case of a Gaussian belief. The distribution of y_{t+1} under the prior belief $\mathbb{b}^{(0)}$ and the current belief $\mathbb{b}^{(t)}$ are shown by black and red curves, respectively. Two possible observations with equal absolute prediction error δ but opposite sign bias s are indicated by dashed lines. The two observations are equally probable under $\mathbb{b}^{(t)}$, but not under $\mathbb{b}^{(0)}$. $\mathbf{S}_{\mathrm{BF}}$ is computed as the ratio between the red and black dots for a given observation, whereas $\mathbf{S}_{\mathrm{Sh}}$ is a function of the weighted sum of the two. This phenomenon is the basis of our experimental prediction. **B.** The average surprise values $\bar{\mathbf{S}}_{\mathrm{Sh}}(\hat{\theta}=1,\delta,s=\pm1,\sigma_C=0.5)$ and $\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}=1,\delta,s=\pm1,\sigma_C=0.5)$ over 20 subjects (each with 500 observations) are shown for two different learning algorithms (Nas12* and pf20). The mean $\bar{\mathbf{S}}_{\mathrm{BF}}$ is higher for negative sign bias (marked in blue) than for positive sign bias (marked in orange). The opposite is observed for the mean $\bar{\mathbf{S}}_{\mathrm{Sh}}$. This effect increases with increasing values of prediction error δ. The shaded area corresponds to the standard error of the mean. The experimental task is the same as the Gaussian task we used in the previous section, with $\sigma=0.5$ and $p_c=0.1$ (see Methods for details).

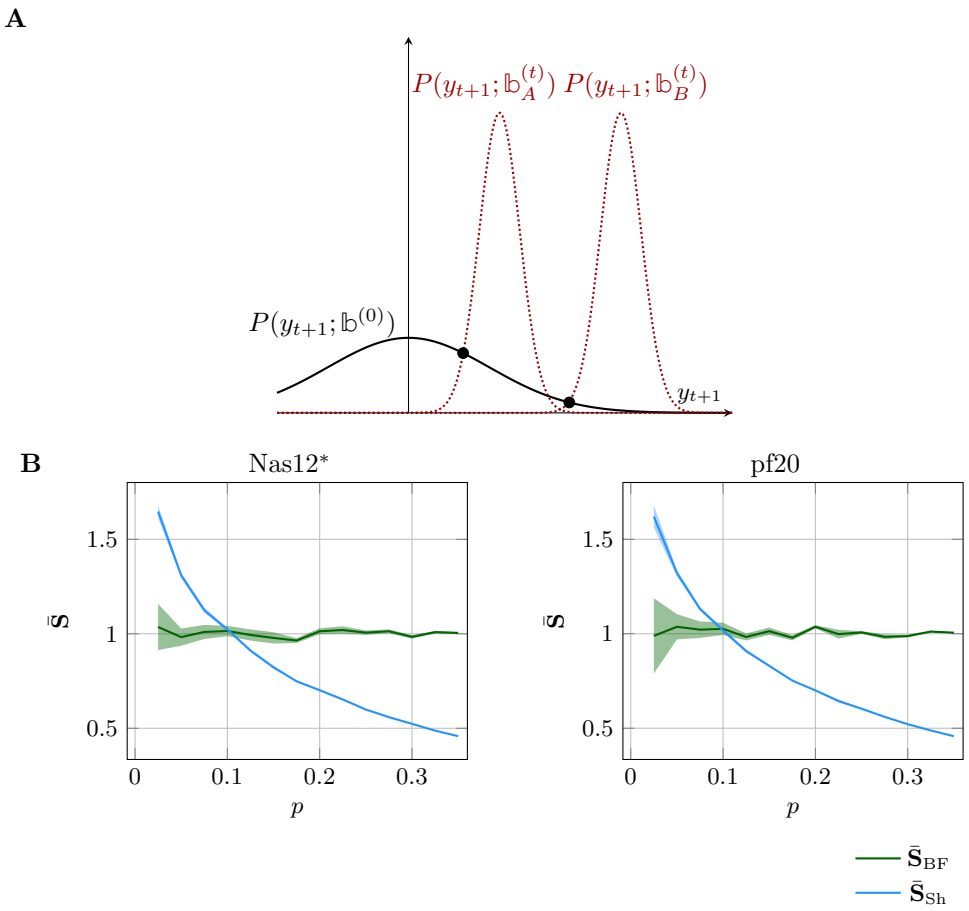


Figure 2.11 – **Experimental prediction 2. A.** Schematic of the task for the case of a Gaussian belief. The probability distribution of observations under the prior belief is shown by the solid black curve. Two different possible current beliefs (determined by the letters A and B) are shown by dashed red curves. The intersections of the dashed red curves with the prior belief determine observations whose $\mathbf{S}_{\mathrm{BF}}$ is same and equal to one, but their $\mathbf{S}_{\mathrm{Sh}}$ is a function of their probabilities under the prior belief p. **B.** The average surprise values $\bar{\mathbf{S}}_{\mathrm{Sh}}(p)$ and $\bar{\mathbf{S}}_{\mathrm{BF}}(p)$ over 20 subjects (each with 500 observations) are shown for two different learning algorithms (Nas12* and pf20). The mean $\mathbf{S}_{\mathrm{BF}}$ is constant (equal to 1) and independent of p, whereas the mean $\mathbf{S}_{\mathrm{Sh}}$ is a decreasing function of p. The shaded area corresponds to the standard error of the mean. The experimental task is the same as the Gaussian task we used in the previous section. Observations y_t are drawn from a Gaussian distribution with $\sigma = 0.5$, whose mean changes with change point probability $p_c = 0.1$ (see Methods for details).

2.3.1 Biological interpretation

Humans are able to quickly adapt to changes (Behrens et al., 2007; Nassar et al., 2010, 2012), but human behaviour is also often observed to be suboptimal, compared to the normative approach of exact Bayesian inference (Glaze et al., 2015; Mathys et al., 2011; Nassar et al., 2010; Prat-Carrabin et al., 2020; Wilson et al., 2013). In general, biological agents have limited resources and possibly inaccurate assumptions about hyper-parameters, yielding sub-optimal behaviour, as we also see with our algorithms whose accuracies degrade with a sub-optimal choice of hyper-parameters. Performance also deteriorates with a decreasing number of particles in the sampling-based algorithms, which might be another possible explanation of suboptimal human behaviour. Previously, Particle Filtering has been shown to explain the behaviour of human subjects in changing environments: Daw and Courville (2008) use a single particle, (Brown and Steyvers, 2009) use a simple heuristic form of particle filtering based on direct simulation, Findling et al. (2019) combine Particle Filtering with a noisy inference, and Prat-Carrabin et al. (2020) use it for a task with temporal structure.

At the level of neuronal implementation, we do not propose a specific suggestion. However, there are several hypotheses about neural implementations of related particle filters (Huang and Rao, 2014; Kutschireiter et al., 2017; Legenstein and Maass, 2014; Shi and Griffiths, 2009), on which, a neural model of pf$N$ and – its greedy version – MPN could be based. In a similar spirit, the updating scheme of Variational SMiLe may be implemented in biological neural networks (for distributions in the exponential family).

Our theoretical framework for modulation of learning by the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}$ is related to the body of literature on neo-Hebbian three-factor learning rules (Frémaux and Gerstner, 2016; Gerstner et al., 2018; Lisman et al., 2011), where a third factor indicating reward or surprise enables or modulates a synaptic change or a belief update (Angela, 2012; Yu and Dayan, 2005). We have shown how Bayesian or approximate Bayesian inference naturally leads to such a third factor that modulates learning via the surprise modulated adaptation rate $\gamma(\mathbf{S}_{\mathrm{BF}}, m)$. This may offer novel interpretations of behavioural and neurophysiological data, and help in understanding how three-factor learning computations may be implemented in the brain.

2.3.2 Related work

Exact Bayesian inference As already described in the "Message-Passing N" section of the Results, for the generative model in Fig. 2.1, it is possible to find an exact online Bayesian update of the belief using a message passing algorithm (Adams and MacKay, 2007). The space and time complexity of the algorithm increases linearly with t, which makes it unsuitable for an online learning setting. However, approximations like dropping messages below a certain threshold (Adams and MacKay, 2007) or stratified resampling

(Fearnhead and Liu, 2007) allow to reduce the computational complexity. The former has a variable number of particles in time, and the latter needs solving a complicated non-linear equation at each time step in order to reduce the number of particles to N (called SORN in the Results section).

Our message passing algorithm with finite number of particles (messages) N (MPN, Algo. 3) is closely related to these algorithms and can be seen as a biologically more plausible variant of the other two. All three algorithms have the same update rules given by Equation 2.19 and Equation 2.18. Hence the algorithms of both Adams and MacKay (2007) and Fearnhead and Liu (2007) have the same surprise modulation as our MPN. The difference lies in their approaches to eliminate less "important" particles.

In the literature of switching state-space models (Barber, 2012), the generative models of the kind in Fig. 2.1 are known as "reset models", and the message passing algorithm of Adams and MacKay (2007) is known to be the standard algorithm for inference over these models (Barber, 2012). See Barber (2006, 2012); Ghahramani and Hinton (2000) for other variations of switching state-space models and examples of approximate inference over them.

Leaky integration and variations of delta-rules In order to estimate some statistics, leaky integration of new observations is a particularly simple form of a trade-off between integrating and forgetting. After a transient phase, the update of a leaky integrator takes the form of a delta-rule that can be seen as an approximation of exact Bayesian updates (Heilbron and Meyniel, 2019; Meyniel et al., 2016; Ryali et al., 2018; Yu and Cohen, 2009). This update rule was found to be biologically plausible and consistent with human behavioural data (Meyniel et al., 2016; Yu and Cohen, 2009). However, Behrens et al. (2007) and Heilbron and Meyniel (2019) demonstrated that in some situations, the exact Bayesian model is significantly better than leaky integration in explaining human behaviour. The inflexibility of leaky integration with a single, constant leak parameter can be overcome by a weighted combination of multiple leaky integrators (Wilson et al., 2013), where the weights are updated in a similar fashion as in the exact online methods (Adams and MacKay, 2007; Fearnhead and Liu, 2007), or by considering an adaptive leak parameter (Nassar et al., 2010, 2012). We have shown that the two algorithms of Nassar et al. (2010, 2012) can be generalized to Gaussian prior beliefs (Nas10* and Nas12*). Our results show that these algorithms also inherit the surprise-modulation of the exact Bayesian inference. Our surprise-dependent adaptation rate γ can be interpreted as a surprise-modulated leak parameter.

Other approaches Learning in the presence of abrupt changes has also been considered without explicit assumptions about the underlying generative model. One approach uses a surprise-modulated adaptation rate (Faraji et al., 2018) similar to Equation 2.9. The

Surprise-Minimization Learning (SMiLe) algorithm of Faraji et al. (2018) has an updating rule similar to the one of VarSMiLe (Equation 2.14 and Equation 2.15). The adaptation rate modulation, however, is based on the Confidence Corrected Surprise (Faraji et al., 2018) rather than the Bayes Factor Surprise, and the trade-off in its update rule is between resetting and staying with the latest belief rather than between resetting and integrating (see Methods).

Other approaches use different generative models, such as conditional sampling of the parameters also when there is a change (Glaze et al., 2015; Yu and Dayan, 2005), a deeper hierarchy without fixed change probability p_c (Wilson et al., 2010), or drift in the parameters (Gershman et al., 2014b; Mathys et al., 2011). A recent work shows that inference on a simpler version of the generative model of Fig. 2.1, with no change points but with a noisy inference style, can explain human behaviour well even when the true generative model of the environment is different and more complicated (Findling et al., 2019). They develop a heuristic approach to add noise in the inference process of a Particle Filter. Their algorithm can be interpreted as a surprise-modulated Particle Filter, where the added noise scales with a measure of surprise (conceptually equivalent to Bayesian surprise (Itti and Baldi, 2006; Schmidhuber, 2010; Storck et al., 1995)). Moreover, another recent work (Prat-Carrabin et al., 2020) shows that approximate sampling algorithms (like Particle Filtering) can explain human behaviour better than their alternatives in tasks closely related to the generative model of Fig. 2.1. The signal processing literature provides further methods to address the problem of learning in non-stationary environments with abrupt changes; see Aminikhanghahi and Cook (2017) for a review, and Cummings et al. (2018); Lin et al. (2017); Masegosa et al. (2017); Özkan et al. (2013) for a few recent examples.

2.3.3 Surprise-modulation as a generic phenomenon

Learning rate modulation similar to the one in Equation 2.9 has been previously proposed in the neuroscience literature with either heuristic arguments (Faraji et al., 2018) or with Bayesian arguments for a particular experimental task, e.g. when samples are drawn from a Gaussian distribution (Nassar et al., 2010, 2012). The fact that the same form of modulation is at the heart of Bayesian inference for our relatively general generative model, that it is derived without any further assumptions, and is not a-priori defined is in our view an important contribution to the field of adaptive learning algorithms in computational neuroscience.

Furthermore, the results of our three approximate methods (Particle Filtering, Variational SMiLe, and Message Passing with fixed N number of messages) as well as some previously developed ones (Adams and MacKay, 2007; Fearnhead and Liu, 2007; Nassar et al., 2010, 2012) demonstrate that the surprise-based modulation of the learning rate is a generic phenomenon. Therefore, regardless of whether the brain uses Bayesian inference

or an approximate algorithm (Bogacz, 2017, 2019; Findling et al., 2019; Friston, 2010; Gershman, 2019; Gershman et al., 2014b; Mathys et al., 2011; Nassar et al., 2010, 2012; Prat-Carrabin et al., 2020), the notion of Bayes Factor Surprise and the way it modulates learning (i.e. Equation 4.6 and Equation 2.9) look generic.

The generality of the way surprise should modulate learning depends on an agent's inductive biases about its environment and is directly associated with the assumed generative model of the world. The generative model we considered in this work involves abrupt changes. However, one can think of other realistic examples, where an improbable observation does not indicate a persistent change, but a singular event or an outlier, similar to d'Acremont and Bossaerts (2016); Nassar et al. (2019). In such situations, the belief should not be changed and surprise should attenuate learning, rather than accelerate it. Interestingly, we can show that exact and approximate Bayesian inference on such a generative model naturally lead to a surprise-modulated adaptation rate $\gamma(\mathbf{S}_{\mathrm{BF}}, m)$, with the same definition of $\mathbf{S}_{\mathrm{BF}}$, where the trade-off is not between integrating and resetting, but between integrating and ignoring the new observation (see Chapter 4). This extends previous work on such environments (d'Acremont and Bossaerts, 2016; Nassar et al., 2019) to a general setting and highlights the general principle of surprise-based modulation, given the prior knowledge on the structure of the environment.

An aspect that the generative model we considered does not capture is the potential return to a previous state of the environment, rather than a change to a completely new situation. If in our example of Fig. 2.1B, the bridge with the shortest path is temporarily closed for repairs, your friend would again have to take the longer detour, therefore, her arrival times will return to their previous values, i.e. increase. In such cases, an agent should infer whether the surprising observation stems from a new hidden state or from an old state stored in memory. Relevant generative models have been studied in Collins and Koechlin (2012); Findling et al. (2019); Fox et al. (2011); Gershman et al. (2014b, 2017) and are out of the scope of our present work.

2.3.4 Bayes Factor Surprise as a novel measure of surprise

In view of a potential application in the neurosciences, a definition of surprise should exhibit two properties: (i) surprise should reflect how unexpected an event is, and, (ii) surprise should modulate learning. Surprising events indicate that our belief is far from the real world and suggest to update our model of the world, or, for large surprise, simply forget it. Forgetting is the same as returning to the prior belief. However, an observation y_{t+1} can be unexpected under both the prior $\mathbb{b}^{(0)}$ and the current beliefs $\mathbb{b}^{(t)}$. In these situations, it is not obvious whether forgetting helps. Therefore, the modulation between forgetting or not should be based on a comparison between the probability of an event under the current belief $P(y_{t+1}; \mathbb{b}^{(t)})$ and its probability under the prior belief $P(y_{t+1}; \mathbb{b}^{(0)})$.

The definition of the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}$ as the ratio of $P(y_{t+1}; \mathbb{b}^{(t)})$ and $P(y_{t+1}; \mathbb{b}^{(0)})$ exploits this insight. The Bayes Factor Surprise appears as a modulation factor in the recursive form of the exact Bayesian update rule for a hierarchical generative model of the environment. When two events are equally probable under the prior belief, the one which is less expected under the current belief is more surprising - satisfying the first property. At the same time, when two events are equally probable under the current belief, the one which is more expected under the prior belief is more surprising - signaling that forgetting may be beneficial.

$\mathbf{S}_{\mathrm{BF}}$ can be written (using Equation 2.6) in a more explicit way as

$$\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)}) = \frac{P(y_{t+1}; \mathbb{b}^{(0)})}{P(y_{t+1}; \mathbb{b}^{(t)})} = \frac{\mathbb{E}_{\mathbb{b}^{(0)}}\left[P_Y(y_{t+1}|\Theta)\right]}{\mathbb{E}_{\mathbb{b}^{(t)}}\left[P_Y(y_{t+1}|\Theta)\right]}. \tag{2.26}$$

Note that the definition by itself is independent of the specific form of the generative model. In other words, even in the cases where data is generated with another generative model (e.g. the real world), $\mathbf{S}_{\mathrm{BF}}$ could be a candidate surprise measure in order to interpret brain activity or pupil dilation.

We formally discussed the connections between the Bayes Factor Surprise and Shannon Surprise (Shannon, 1948), and showed that they are closely linked. We showed that the modulated adaptation rate (γ) used in (approximate) Bayesian inference is a function of the difference between the Shannon Surprise under the current and the prior beliefs, but cannot be expressed solely by the Shannon Surprise under the current one. Our formal comparisons between these two different measures of surprise lead to specific experimentally testable predictions.

The Bayesian Surprise $\mathbf{S}_{\mathrm{Ba}}$ (Itti and Baldi, 2006; Schmidhuber, 2010; Storck et al., 1995) and the Confidence Corrected Surprise $\mathbf{S}_{\mathrm{CC}}$ (Faraji et al., 2018) are two other measures of surprise in neuroscience. The learning modulation derived in our generative model cannot be expressed as a function of $\mathbf{S}_{\mathrm{Ba}}$ and $\mathbf{S}_{\mathrm{CC}}$. However, one can hypothesize that $\mathbf{S}_{\mathrm{Ba}}$ is computed after the update of the belief to measure the information gain of the observed event, and is therefore not a good candidate for online learning modulation. The Confidence Corrected surprise $\mathbf{S}_{\mathrm{CC}}$ takes into account the shape of the belief, and therefore includes the effects of confidence, but it does not consider any information about the prior belief. Hence, a result of $\bar{M}_1(\hat{\theta}, \delta, s = +1, C) = \bar{M}_1(\hat{\theta}, \delta, s = -1, C)$ in our first experimental prediction would be consistent with the corresponding behavioral or physiological indicator reflecting the $\mathbf{S}_{\mathrm{CC}}$.

2.3.5 Difference in Shannon Surprise, an alternative perspective

Following our formal comparison in the "Experimental prediction" section, $\mathbf{S}_{\mathrm{BF}}$ can be expressed as a deterministic function of the difference in Shannon Surprise as

$$\mathbf{S}_{\mathrm{BF}} = \frac{(1-p_c)e^{\Delta\mathbf{S}_{\mathrm{Sh}}}}{1-p_c e^{\Delta\mathbf{S}_{\mathrm{Sh}}}}. \tag{2.27}$$

All of our theoretical results can be rewritten by replacing $\mathbf{S}_{\mathrm{BF}}$ with this function of $\Delta\mathbf{S}_{\mathrm{Sh}}$. Moreover, because there is a 1-to-1 mapping between $\mathbf{S}_{\mathrm{BF}}$ and $\Delta\mathbf{S}_{\mathrm{Sh}}$, from a systemic point of view, it is not possible to specify whether the brain computes the former or the latter by analysis of behavioural data and biological signals. This suggests an alternative interpretation of surprise-modulated learning as an approximation of Bayesian inference: What the brain computes and perceives as surprise or prediction error may be Shannon Surprise, but the modulating factor in a three-factor synaptic plasticity rule (Frémaux and Gerstner, 2016; Gerstner et al., 2018; Lisman et al., 2011) may be implemented by comparing the Shannon Surprise values under the current and the prior beliefs.

2.3.6 Future directions

A natural continuation of our study is to test our experimental predictions in human behaviour and physiological signals, in order to investigate which measures of surprise are used by the brain. Along a similar direction, our approximate learning algorithms can be evaluated on human behavioural data from experiments that use a similar generative model (Behrens et al., 2007; Glaze et al., 2015; Heilbron and Meyniel, 2019; Nassar et al., 2010, 2012; Wilson et al., 2013; Yu and Dayan, 2005) in order to assess if our proposed algorithms achieve similar or better performance in explaining data.

Finally, our methods can potentially be applied to model-based reinforcement learning in non-stationary environments. In recent years, there has been a growing interest in adaptive or continually learning agents in changing environments in the form of Continual learning and Meta-learning (Lomonaco et al., 2019; Traoré et al., 2019). Many Continual learning model-based approaches make use of some procedure to detect changes (Lomonaco et al., 2019; Nagabandi et al., 2018). Integrating $\mathbf{S}_{\mathrm{BF}}$ and a learning rate $\gamma(\mathbf{S}_{\mathrm{BF}})$ into a reinforcement learning agent is a direction we explore in the next chapters.

2.4 Methods

2.4.1 Proof of the proposition

By definition

$$\mathbb{b}^{(t+1)}(\theta) \equiv \mathbf{P}(\Theta_{t+1} = \theta | y_{1:t+1}). \tag{2.28}$$

We exploit the Markov property of the generative model in Equation 2.2, Equation 2.3, and Equation 2.4, condition on the fixed past $y_{1:t}$ and rewrite

$$\mathbb{b}^{(t+1)}(\theta) = \frac{P_Y(y_{t+1}|\theta)\mathbf{P}(\Theta_{t+1} = \theta|y_{1:t})}{\mathbf{P}(y_{t+1}|y_{1:t})}. \tag{2.29}$$

By marginalization over the hidden state C_{t+1}, the second factor in the numerator of Equation 2.29 can be written as

$$\mathbf{P}(\Theta_{t+1} = \theta|y_{1:t}) = (1 - p_c)\mathbb{b}^{(t)}(\theta) + p_c\mathbb{b}^{(0)}(\theta). \tag{2.30}$$

The denominator in Equation 2.29 can be written as

$$\begin{aligned}\mathbf{P}(y_{t+1}|y_{1:t}) &= \int P_Y(y_{t+1}|\theta)\mathbf{P}(\Theta_{t+1} = \theta|y_{1:t})d\theta \\ &= (1 - p_c)\int P_Y(y_{t+1}|\theta)\mathbb{b}^{(t)}(\theta)d\theta + p_c\int P_Y(y_{t+1}|\theta)\mathbb{b}^{(0)}(\theta)d\theta \\ &= (1 - p_c)P(y_{t+1}; \mathbb{b}^{(t)}) + p_cP(y_{t+1}; \mathbb{b}^{(0)}).\end{aligned} \tag{2.31}$$

where we used the definition in Equation 2.6. Using these two expanded forms, Equation 2.29 can be rewritten

$$\mathbb{b}^{(t+1)}(\theta) = \frac{P_Y(y_{t+1}|\theta)\Big((1 - p_c)\mathbb{b}^{(t)}(\theta) + p_c\mathbb{b}^{(0)}(\theta)\Big)}{(1 - p_c)P(y_{t+1}; \mathbb{b}^{(t)}) + p_cP(y_{t+1}; \mathbb{b}^{(0)})}. \tag{2.32}$$

We define $P(\theta|y_{t+1})$ as the posterior given a change in the environment as

$$P(\theta|y_{t+1}) = \frac{P_Y(y_{t+1}|\theta)\mathbb{b}^{(0)}(\theta)}{P(y_{t+1}; \mathbb{b}^{(0)})}. \tag{2.33}$$

Then, we can write Equation 2.32 as

$$\begin{aligned}\mathbb{b}^{(t+1)}(\theta) &= \frac{(1 - p_c)P(y_{t+1}; \mathbb{b}^{(t)})\mathbb{b}_B^{(t+1)}(\theta) + p_cP(y_{t+1}; \mathbb{b}^{(0)})P(\theta|y_{t+1})}{(1 - p_c)P(y_{t+1}; \mathbb{b}^{(t)}) + p_cP(y_{t+1}; \mathbb{b}^{(0)})} \\ &= \frac{\mathbb{b}_B^{(t+1)}(\theta) + \frac{p_c}{1-p_c}\frac{P(y_{t+1};\mathbb{b}^{(0)})}{P(y_{t+1};\mathbb{b}^{(t)})}P(\theta|y_{t+1})}{1 + \frac{p_c}{1-p_c}\frac{P(y_{t+1};\mathbb{b}^{(0)})}{P(y_{t+1};\mathbb{b}^{(t)})}} \\ &= (1 - \gamma_{t+1})\mathbb{b}_B^{(t+1)}(\theta) + \gamma_{t+1}P(\theta|y_{t+1}),\end{aligned} \tag{2.34}$$

where $\mathbb{b}_B^{(t+1)}(\theta)$ is defined in Equation 2.8, and

$$\gamma_{t+1} = \gamma\Big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)}), \frac{p_c}{1 - p_c}\Big) \tag{2.35}$$

with $\mathbf{S}_{\mathrm{BF}}$ defined in Equation 2.7, and $\gamma(\mathrm{S}, m)$ defined in Equation 2.9. Thus our calculation yields a specific choice of surprise ($\mathrm{S} = \mathbf{S}_{\mathrm{BF}}$) and a specific value for the

saturation parameter $m = \frac{p_c}{1-p_c}$.

2.4.2 Derivation of the optimization-based formulation of VarSMiLe (Algo. 1)

To derive the optimization-based update rule for the Variational SMiLe rule and the relation of the bound B_{t+1} with surprise, we used the same approach used in Faraji et al. (2018).

Derivation of the update rule. Consider the general form of the following variational optimization problem:

$$\begin{aligned} q^*(\theta) = \text{argmin } & \mathbf{D}_{KL}\big[q(\theta)||p_1(\theta)\big] \\ & q(\theta) \text{ s.t. } \mathbf{D}_{KL}\big[q(\theta)||p_2(\theta)\big] < B \text{ and } \mathbb{E}_q[1] = 1, \end{aligned} \tag{2.36}$$

where $B \in \big[0, \mathbf{D}_{KL}[p_1(\theta)||p_2(\theta)]\big]$. On the extremes of B, we will have trivial solutions

$$q^*(\theta) = \begin{cases} p_2(\theta) & \text{if} \quad B = 0 \\ p_1(\theta) & \text{if} \quad B = \mathbf{D}_{KL}[p_1(\theta)||p_2(\theta)]. \end{cases} \tag{2.37}$$

Note that the Kullback–Leibler divergence is a convex function with respect to its first argument, i.e. q in our setting. Therefore, both the objective function and the constraints of the optimization problem in Equation 2.36 are convex. For convenience, we assume that the parameter space for θ is discrete, but the final results can be generalized also to the continuous case with some considerations - see Beal (2003) and Faraji et al. (2018). For the discrete setting, the optimization problem in Equation 2.36 can be rewritten as

$$\begin{aligned} q^*(\theta) = \text{argmin } & \sum_\theta q(\theta)\big(\log(q(\theta)) - \log(p_1(\theta))\big) \\ & q(\theta) \text{ s.t. } \sum_\theta q(\theta)\big(\log(q(\theta)) - \log(p_2(\theta)) < B \text{ and } \sum_\theta q(\theta) = 1. \end{aligned} \tag{2.38}$$

For solving the mentioned problem, one should find a q which satisfies the Karush–Kuhn–Tucker (KKT) conditions (Boyd and Vandenberghe, 2004) for

$$\mathcal{L} = \sum_\theta q(\theta)\log\Big(\frac{q(\theta)}{p_1(\theta)}\Big) + \lambda \sum_\theta q(\theta)\log\Big(\frac{q(\theta)}{p_2(\theta)}\Big) - \lambda B + \alpha - \alpha \sum_\theta q(\theta), \tag{2.39}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial q(\theta)} &= \log\Big(\frac{q(\theta)}{p_1(\theta)}\Big) + 1 + \lambda\log\Big(\frac{q(\theta)}{p_2(\theta)}\Big) + \lambda - \alpha \\ &= (1+\lambda)\log(q(\theta)) - \log(p_1(\theta)) - \lambda\log(p_2(\theta)) + 1 + \lambda - \alpha, \end{aligned} \tag{2.40}$$

where λ and α are the parameters of the dual problem. Defining $\gamma = \frac{\lambda}{1+\lambda}$, and considering the partial derivative to be zero, we have

$$\log(q^*(\theta)) = (1-\gamma)\log(p_1(\theta)) + \gamma\log(p_2(\theta)) + \text{Const}(\alpha,\gamma), \tag{2.41}$$

where α is always specified in a way to have $\text{Const}(\alpha,\gamma)$ as the normalization factor

$$\begin{aligned}&\text{Const}(\alpha,\gamma) = -\log(Z(\gamma))\\ &\qquad \text{where } Z(\gamma) = \sum_{\theta} p_1^{1-\gamma}(\theta)p_2^{\gamma}(\theta).\end{aligned} \tag{2.42}$$

According to the KKT conditions, $\lambda \geq 0$, and as a result $\gamma \in [0,1]$. Therefore, considering $p_1(\theta) = \hat{\mathbb{b}}_B^{(t+1)}(\theta)$ and $p_2(\theta) = P(\theta|y_{t+1})$, the solution to the optimization problem of Equation 2.14 and Equation 2.15 is Equation 2.13.

Proof of the claim that B is a decreasing function of Surprise. According to the KKT conditions

$$\lambda\Big(\mathbf{D}_{KL}[q^*(\theta)||p_2(\theta)] - B\Big) = 0. \tag{2.43}$$

For the case that $\lambda \neq 0$ (i.e. $\gamma \neq 0$), we have B as a function of γ

$$\begin{aligned}B(\gamma) &= \mathbf{D}_{KL}[q^*(\theta)||p_2(\theta)]\\ &= (1-\gamma)\mathbb{E}_{q^*}\Big[\log\big(\frac{p_1(\theta)}{p_2(\theta)}\big)\Big] - \log\big(Z(\gamma)\big).\end{aligned} \tag{2.44}$$

Now, we show that the derivate of $B(\gamma)$ with respect to γ is always non-positive. To do so, we first compute the derivative of $Z(\gamma)$ as

$$\begin{aligned}\frac{\partial\log(Z(\gamma))}{\partial\gamma} &= \frac{1}{Z(\gamma)}\frac{\partial}{\partial\gamma}\sum_{\theta} p_1^{1-\gamma}(\theta)p_2^{\gamma}(\theta)\\ &= \frac{1}{Z(\gamma)}\sum_{\theta} p_1^{1-\gamma}(\theta)p_2^{\gamma}(\theta)\log\big(\frac{p_2(\theta)}{p_1(\theta)}\big)\\ &= \mathbb{E}_{q^*}\Big[\log\big(\frac{p_2(\theta)}{p_1(\theta)}\big)\Big],\end{aligned} \tag{2.45}$$

and the derivate of $\mathbb{E}_{q^*}[h(\theta)]$ for an arbitrary $h(\theta)$ as

$$\begin{aligned}\frac{\partial \mathbb{E}_{q^*}[h(\theta)]}{\partial \gamma} &= \frac{\partial}{\partial \gamma}\sum_{\theta} q^*(\theta)h(\theta) \\ &= \sum_{\theta} q^*(\theta)h(\theta)\frac{\partial}{\partial \gamma}\log(q^*(\theta)) \\ &= \sum_{\theta} q^*(\theta)h(\theta)\frac{\partial}{\partial \gamma}\Big((1-\gamma)\log(p_1(\theta)) + \gamma\log(p_2(\theta)) - \log(Z(\gamma))\Big) \\ &= \mathbb{E}_{q^*}\Big[h(\theta)\log\big(\frac{p_2(\theta)}{p_1(\theta)}\big)\Big] - \mathbb{E}_{q^*}\big[h(\theta)\big]\mathbb{E}_{q^*}\Big[\log\big(\frac{p_2(\theta)}{p_1(\theta)}\big)\Big].\end{aligned} \tag{2.46}$$

Using the last three equations, we have

$$\begin{aligned}\frac{\partial B(\gamma)}{\partial \gamma} &= -(1-\gamma)\Big(\mathbb{E}_{q^*}\Big[\Big(\log\big(\frac{p_2(\theta)}{p_1(\theta)}\big)\Big)^2\Big] - \mathbb{E}_{q^*}\Big[\log\big(\frac{p_2(\theta)}{p_1(\theta)}\big)\Big]^2\Big) \\ &= -(1-\gamma)\mathbf{Var}_{q^*}\Big[\log\big(\frac{p_2(\theta)}{p_1(\theta)}\big)\Big] \leq 0,\end{aligned} \tag{2.47}$$

which means that B is a decreasing function of γ. Because γ is an increasing function of surprise, B is also a decreasing function of surprise.

2.4.3 Derivations of Message Passing N (Algo. 2)

For the sake of clarity and coherence, we repeat here some steps performed in the Results section.

Following the idea of Adams and MacKay (2007) let us first define the random variable $R_t = \min\{n \in \mathbb{N} : C_{t-n+1} = 1\}$. This is the time window from the last change point. Then the exact Bayesian form for $\mathbb{b}^{(t)}(\theta)$ can be written as

$$\begin{aligned}\mathbb{b}^{(t)}(\theta) &= \mathbf{P}(\Theta_{t+1} = \theta | y_{1:t}) \\ &= \sum_{r_t=1}^{t} \mathbf{P}(r_t|y_{1:t})\mathbf{P}(\Theta_{t+1} = \theta | r_t, y_{1:t}).\end{aligned} \tag{2.48}$$

To have a formulation similar to the one of Particle Filtering we rewrite the belief as

$$\mathbb{b}^{(t)}(\theta) = \sum_{k=0}^{t-1} w_t^{(k)}\mathbf{P}(\Theta_{t+1} = \theta | R_t = t-k, y_{1:t}) = \sum_{k=0}^{t-1} w_t^{(k)}\mathbb{b}_k^{(t)}(\theta), \tag{2.49}$$

where $\mathbb{b}_k^{(t)}(\theta) = \mathbf{P}(\Theta_t = \theta | R_t = t-k, y_{1:t})$ is the term corresponding to $R_t = t-k$, and $w_t^{(k)} = \mathbf{P}(R_t = t-k|y_{1:t})$ is its corresponding at time t.

To update the belief after observing y_{t+1}, one can use the exact Bayesian recursive formula Equation 2.10, for which one needs to compute $\mathbb{b}_B^{(t+1)}(\theta)$ as

$$\begin{aligned}\mathbb{b}_B^{(t+1)}(\theta) &= \frac{\mathbb{b}^{(t)}(\theta)P_Y(y_{t+1}|\theta)}{P(y_{t+1};\mathbb{b}^{(t)})} \\ &= \frac{P_Y(y_{t+1}|\theta)}{P(y_{t+1};\mathbb{b}^{(t)})}\sum_{k=0}^{t-1} w_t^{(k)}\mathbf{P}(\Theta_{t+1}=\theta|R_t=t-k,y_{1:t}).\end{aligned} \tag{2.50}$$

Using Bayes' rule and the conditional independence of observations, we have

$$\begin{aligned}\mathbb{b}_B^{(t+1)}(\theta) &= \frac{P_Y(y_{t+1}|\theta)}{P(y_{t+1};\mathbb{b}^{(t)})}\sum_{k=0}^{t-1} w_t^{(k)}\frac{\mathbf{P}(y_{k+1:t}|\Theta_{t+1}=\theta,R_t=t-k)\mathbb{b}^{(0)}(\theta)}{\mathbf{P}(y_{k+1:t}|R_t=t-k)} \\ &= \frac{1}{P(y_{t+1};\mathbb{b}^{(t)})}\sum_{k=0}^{t-1} w_t^{(k)}\frac{\prod_{i=k+1}^{t+1}P_Y(y_i|\theta)\mathbb{b}^{(0)}(\theta)}{\mathbf{P}(y_{k+1:t}|R_t=t-k)},\end{aligned} \tag{2.51}$$

and once again, by using the Bayes' rule and the conditional independence of observations, we find

$$\begin{aligned}\mathbb{b}_B^{(t+1)}(\theta) &= \frac{1}{P(y_{t+1};\mathbb{b}^{(t)})}\sum_{k=0}^{t-1} w_t^{(k)}\frac{\mathbf{P}(y_{k+1:t+1}|R_{t+1}=t-k+1)}{\mathbf{P}(y_{k+1:t}|R_t=t-k)}\times \\ &\qquad \mathbf{P}(\Theta_{t+1}=\theta|R_{t+1}=t-k+1,y_{k+1:t+1}) \\ &= \frac{1}{P(y_{t+1};\mathbb{b}^{(t)})}\sum_{k=0}^{t-1} w_t^{(k)}\mathbf{P}(y_{t+1}|R_{t+1}=t-k+1,y_{1:t})\times \\ &\qquad \mathbf{P}(\Theta_{t+1}=\theta|R_{t+1}=t-k+1,y_{1:t+1}).\end{aligned} \tag{2.52}$$

This gives us

$$\begin{aligned}\mathbb{b}_B^{(t+1)}(\theta) &= \sum_{k=0}^{t-1} w_t^{(k)}\frac{P(y_{t+1};\mathbb{b}_k^{(t)})}{P(y_{t+1};\mathbb{b}^{(t)})}\times \\ &\qquad \times\mathbf{P}(\Theta_{t+1}=\theta|R_{t+1}=t-k+1,Y_{1:t+1}=y_{1:t+1}),\end{aligned} \tag{2.53}$$

and finally

$$w_{B,t+1}^{(k)} = \frac{P(y_{t+1};\mathbb{b}_k^{(t)})}{P(y_{t+1};\mathbb{b}^{(t)})}w_t^{(k)}. \tag{2.54}$$

Using the recursive formula, the update rule for the weights for $0 \leq k \leq t-1$ is

$$w_{t+1}^{(k)} = (1-\gamma_{t+1})w_{B,t+1}^{(k)} = (1-\gamma_{t+1})\frac{P(y_{t+1};\mathbb{b}_k^{(t)})}{P(y_{t+1};\mathbb{b}^{(t)})}w_t^{(k)}, \tag{2.55}$$

and for the newly added particle t

$$w_{t+1}^{(t)} = \gamma_{t+1}, \tag{2.56}$$

where $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)}), m = \frac{p_c}{1-p_c}\big)$ of Equation 2.9.

The MPN algorithm uses Equation 2.49, Equation 2.55, and Equation 2.56 for computing the belief for $t \leq N$ - which is same as the exact Bayesian inference. For $t > N$, it first updates the weights in the same fashion as Equation 2.55 and Equation 2.56, keeps the greatest N weights, and sets the rest weights equal to 0. After normalizing the new weights, it uses Equation 2.49 (but only over the particles with non-zero weights) to compute the belief $\hat{\mathbb{b}}^{(t)}$. For the particular case of exponential family, see Algorithm 2 for the pseudocode.

2.4.4 Derivation of the weight update for Particle Filtering (Algo. 3)

We derive here the weight update for the particle filter. The difference in our formalism from a standard derivation (Särkkä, 2013) is the absence of the Markov property of conditional observations (i.e. $\mathbf{P}(y_{t+1}|c_{1:t+1}, y_{1:t}) \neq \mathbf{P}(y_{t+1}|c_{t+1})$). Our goal is to perform the approximation

$$P(c_{1:t+1}|y_{1:t+1}) \approx \sum_{i=1}^{N} w_{t+1}^{(i)} \delta(c_{1:t+1} - c_{1:t+1}^{(i)})\,. \tag{2.57}$$

Given a proposal sampling distribution Ψ, for the weight of particle i at time $t+1$ we have

$$\begin{aligned} w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(c_{1:t+1}^{(i)}|y_{1:t+1})}{\Psi(c_{1:t+1}^{(i)}|y_{1:t+1})} \propto \frac{\mathbf{P}(c_{1:t+1}^{(i)}, y_{t+1}|y_{1:t})}{\Psi(c_{1:t+1}^{(i)}|y_{1:t+1})} \\ w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(y_{t+1}, c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t})\mathbf{P}(c_{1:t}^{(i)}|y_{1:t})}{\Psi(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1})\Psi(c_{1:t}^{(i)}|y_{1:t})}\,, \end{aligned} \tag{2.58}$$

where the only assumption for the proposal distribution Ψ is that the previous hidden states $c_{1:t}^{(i)}$ are independent of the next observation y_{t+1}, which allows to keep the previous samples $c_{1:t}^{(i)}$ when going from $c_{1:t}^{(i)}$ to $c_{1:t+1}^{(i)}$ and to write the update of the weights in a recursive way (Särkkä, 2013).

Notice that $w_t^{(i)} \propto \frac{\mathbf{P}(c_{1:t}^{(i)}|y_{1:t})}{\Psi(c_{1:t}^{(i)}|y_{1:t})}$ are the weights calculated at the previous time step. Therefore

$$w_{t+1}^{(i)} \propto \frac{\mathbf{P}(y_{t+1}, c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t})}{\Psi(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1})} w_t^{(i)}\,. \tag{2.59}$$

For the choice of Ψ, we use the optimal proposal function in terms of variance of the weights (Doucet et al., 2000)

$$\Psi(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1}) = \mathbf{P}(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1})\,. \tag{2.60}$$

Using Bayes' rule and Equation 2.59 and Equation 2.60, after a few steps of algebra, we have

$$\begin{aligned} w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(y_{t+1}, c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t})}{\mathbf{P}(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1})} w_t^{(i)} = \mathbf{P}(y_{t+1}|c_{1:t}^{(i)}, y_{1:t}) w_t^{(i)} \\ &\propto \Big((1-p_c)\mathbf{P}(y_{t+1}|c_{1:t}^{(i)}, y_{1:t}, c_{t+1}^{(i)}=0) + p_c \mathbf{P}(y_{t+1}|c_{1:t}^{(i)}, y_{1:t}, c_{t+1}^{(i)}=1)\Big) w_t^{(i)}\,. \end{aligned} \tag{2.61}$$

Using the definition in Equation 2.6, we have $\mathbf{P}(y_{t+1}|c_{1:t}^{(i)}, y_{1:t}, c_{t+1}^{(i)}=0) = P(y_{t+1}; \hat{\mathbb{b}}_i^{(t)})$ and $\mathbf{P}(y_{t+1}|c_{1:t}^{(i)}, y_{1:t}, c_{t+1}^{(i)}=1) = P(y_{t+1}; \mathbb{b}^{(0)})$. Therefore, we have

$$w_{t+1}^{(i)} = \Big[(1-p_c)P(y_{t+1}; \hat{\mathbb{b}}_i^{(t)}) + p_c P(y_{t+1}; \mathbb{b}^{(0)})\Big] w_t^{(i)} / Z\,, \tag{2.62}$$

where Z is the normalization factor

$$Z = (1-p_c)P(y_{t+1}; \hat{\mathbb{b}}^{(t)}) + p_c P(y_{t+1}; \mathbb{b}^{(0)})\,, \tag{2.63}$$

where we have

$$P(y_{t+1}; \hat{\mathbb{b}}^{(t)}) = \sum_{i=1}^{N} w_t^{(i)} P(y_{t+1}; \hat{\mathbb{b}}_i^{(t)})\,. \tag{2.64}$$

We now compute the weights corresponding to $\mathbb{b}_B^{(t+1)}$ as defined in Equation 2.8

$$w_{B,t+1}^{(i)} = \frac{P(y_{t+1}; \hat{\mathbb{b}}_i^{(t)})}{P(y_{t+1}; \hat{\mathbb{b}}^{(t)})} w_t^{(i)}\,. \tag{2.65}$$

Combining Equation 2.62, Equation 2.63 and Equation 2.65 we can then re-write the weight update rule as

$$w_{t+1}^{(i)} = (1-\gamma_{t+1}) w_{B,t+1}^{(i)} + \gamma_{t+1} w_t^{(i)}\,, \tag{2.66}$$

where $\gamma_{t+1} = \gamma\Big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mathbb{b}}^{(t)}), \frac{p_c}{1-p_c}\Big)$ of Equation 2.9.

At every time step $t+1$ we sample each particle's hidden state c_{t+1} from the proposal

distribution. Using Equation 2.60, we have

$$\begin{aligned}\Psi(c_{t+1}^{(i)} = 1 | c_{1:t}^{(i)}, y_{1:t+1}) &= \frac{p_c P(y_{t+1}; \mathbb{b}^{(0)})}{(1-p_c) P(y_{t+1}; \hat{\mathbb{b}}_i^{(t)}) + p_c P(y_{t+1}; \mathbb{b}^{(0)})} \\ &= \gamma\Big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mathbb{b}}_i^{(t)}), \frac{p_c}{1-p_c}\Big)\,.\end{aligned} \tag{2.67}$$

We implemented the Sequential Importance Resampling algorithm (Doucet et al., 2000; Gordon et al., 1993), where the particles are resampled when their effective number falls below a threshold. The effective number of the particles is defined as (Doucet et al., 2000; Särkkä, 2013)

$$N_{\mathrm{eff}} \approx \frac{1}{\sum_{i=1}^{N} (w_t^{(i)})^2}\,. \tag{2.68}$$

When N_{eff} is below a critical threshold, the particles are resampled with replacement from the categorical distribution defined by their weights, and all their weights are set to $w_t^{(i)} = 1/N$. We did not optimize the parameter N_{eff}, and following Doucet and Johansen (2009), we performed resampling when $N_{\mathrm{eff}} \leq N/2$.

2.4.5 Surprise-modulation as a framework for other algorithms

SMiLe Rule

The Confidence Corrected Surprise (Faraji et al., 2018) is

$$\mathbf{S}_{CC}(y_{t+1}; \hat{\mathbb{b}}^{(t)}) = \mathbf{D}_{KL}\big[\hat{\mathbb{b}}^{(t)}(\theta) || \tilde{P}(\theta | y_{t+1})\big], \tag{2.69}$$

where $\tilde{P}(\theta|y_{t+1})$ is the scaled likelihood defined as

$$\tilde{P}(\theta | y_{t+1}) = \frac{P_Y(y_{t+1}|\theta)}{\int P_Y(y_{t+1}|\theta') d\theta'}. \tag{2.70}$$

Note that this becomes equal to $P(\theta|y_{t+1})$ if the prior belief $\mathbb{b}^{(0)}$ is a uniform distribution; cf. Equation 2.11.

With the aim of minimizing the Confidence Corrected Surprise by updating the belief during time, Faraji et al. (2018) suggested an update rule solving the optimization problem

$$\begin{aligned}\hat{\mathbb{b}}^{(t+1)}(\theta) = \arg\min_{q} \; & \mathbf{D}_{KL}\big[q(\theta) || \tilde{P}(\theta | y_{t+1})\big] \\ \text{s.t. } & \mathbf{D}_{KL}\big[q(\theta) || \hat{\mathbb{b}}^{(t)}(\theta)\big] \leq B_{t+1},\end{aligned} \tag{2.71}$$

where $B_{t+1} \in \left[0, \mathbf{D}_{KL}[P(\theta|y_{t+1})||\hat{\mathbb{b}}^{(t)}(\theta)]\right]$ is an arbitary bound. The authors showed that the solution to this optimization problem is

$$\log\big(\hat{\mathbb{b}}^{(t+1)}(\theta)\big) = (1-\gamma_{t+1})\,\log\big(\hat{\mathbb{b}}^{(t)}(\theta)\big) + \gamma_{t+1}\,\log\big(\tilde{P}(\theta|y_{t+1})\big) + \text{Const.}, \tag{2.72}$$

where $\gamma_{t+1} \in [0,1]$ is specified so that it satisfies the constraint in Equation 2.71.

Although Equation 2.72 looks very similar to Equation 2.13, it signifies a trade-off between the latest belief $\hat{\mathbb{b}}^{(t)}$ and the belief updated by only the most recent observation $\tilde{P}(\theta|y_{t+1})$, i.e. a trade-off between adherence to the current belief and reset. While SMiLe adheres to the current belief $\hat{\mathbb{b}}^{(t)}$, Variational SMiLe integrates the new observation with the current belief to get $\hat{\mathbb{b}}_B^{(t)}$, which leads to a trade-off similar to the one of the exact Bayesian inference (Equation 4.6 and Equation 2.10).

To modulate the learning rate by surprise, Faraji et al. (2018) considered the boundary B_{t+1} as a function of the Confidence Corrected Surprise, i.e.

$$\begin{aligned} B_{t+1} &= B_{\max}\gamma\Big(\mathbf{S}_{CC}(y_{t+1}), m\Big) \\ &\text{where } B_{\max} = \mathbf{D}_{KL}[P(\theta|y_{t+1})||\hat{\mathbb{b}}^{(t)}(\theta)]\,, \end{aligned} \tag{2.73}$$

where m is a free parameter. Then, γ_{t+1} is found by satisfying the constraint of the optimization problem in Equation 2.71 using Equation 2.72 and Equation 2.73.

Nassar's algorithm

For the particular case that observations are drawn from a Gaussian distribution with known variance and unknown mean, i.e. $y_{t+1}|\mu_{t+1} \sim \mathcal{N}(\mu_{t+1}, \sigma^2)$ and $\theta_t = \mu_t$, Nassar et al. (2010, 2012) considered the problem of estimating the expected μ_t and its variance rather than a probability distribution (i.e. belief) over it, implicitly assuming that the belief is always a Gaussian distribution. The algorithms of Nassar et al. (2010, 2012) were developed for the case that, whenever the environment changes, the mean μ_{t+1} is drawn from a uniform prior with a range of values much larger than the width of the Gaussian likelihood function. The authors showed that in this case, the expected μ_{t+1} (i.e. $\hat{\mu}_{t+1}$) estimated by the agent upon observing a new sample y_{t+1} is

$$\hat{\mu}_{t+1} = \hat{\mu}_t + \alpha_{t+1}(y_{t+1} - \hat{\mu}_t), \tag{2.74}$$

with α_{t+1} the adaptive learning rate given by

$$\alpha_{t+1} = \frac{1 + \Omega_{t+1}\hat{r}_t}{1 + \hat{r}_t}, \tag{2.75}$$

where $\hat{r}_t$ is the estimated time since the last change point (i.e. the estimated $R_t = \min\{n \in \mathbb{N} : C_{t-n+1} = 1)$ and $\Omega_{t+1} = \mathbf{P}(c_{t+1} = 1|y_{1:t+1})$ the probability of a change

given the observation. Note that this quantity, i.e. the posterior change point probability, is the same as our adaptation rate γ_{t+1} of Equation 2.9.

In the next subsection, we extend their approach to a more general case where the prior is a Gaussian distribution with arbitrary variance, i.e. $\mu_{t+1} \sim \mathcal{N}(\mu_0, \sigma_0^2)$. We then discuss the relation of this method to Particle Filtering. A performance comparison between our extended algorithms Nas10* and Nas12* and their original versions Nas10 and Nas12 is depicted in Supplementary Fig. 2.12 and Supplementary Fig. 2.13.

Nas10* and Nas12* algorithms

Let us consider that $y_{1:t}$ are observed, the time since the last change point r_t is known, and the agent's current estimation of μ_t is $\hat{\mu}_t$. It can be shown (see Supplementary Material for the derivation) that the expected μ_{t+1} (i.e. $\hat{\mu}_{t+1}$) upon observing the new sample y_{t+1} is

$$\hat{\mu}_{t+1} = (1-\gamma_{t+1})\Big(\hat{\mu}_t + \frac{1}{\rho + r_t + 1}(y_{t+1} - \hat{\mu}_t)\Big) + \gamma_{t+1}\Big(\mu_0 + \frac{1}{\rho+1}(y_{t+1} - \mu_0)\Big)\,, \quad (2.76)$$

where $\rho = \frac{\sigma^2}{\sigma_0^2}$, μ_0 is the mean of the prior distribution and γ_{t+1} is the adaptation rate of Equation 2.9.

We can see that the updated mean is a weighted average, with surprise-modulated weights, between integrating the new observation with the current mean $\hat{\mu}_t$ and integrating it with the prior mean μ_0, in the same spirit as the other algorithms we considered here. Equation 2.76 can also be seen as a surprise-modulated weighted sum of two delta rules: one including a prediction error between the new observation and the current mean $(y_{t+1} - \hat{\mu}_t)$ and one including a prediction error between the observed sample and the prior mean $(y_{t+1} - \mu_0)$.

In order to obtain a form similar to the one of Nassar et al. (2010, 2012), we can rewrite the above formula as

$$\hat{\mu}_{t+1} = \frac{\rho}{\rho+1}\Big(\hat{\mu}_t + \gamma_{t+1}(\mu_0 - \hat{\mu}_t)\Big) + \frac{1}{\rho+1}\Big(\hat{\mu}_t + \alpha_{t+1}(y_{t+1} - \hat{\mu}_t)\Big)\,, \quad (2.77)$$

where we have defined $\alpha_{t+1} = \frac{\rho + \gamma_{t+1} r_t + 1}{\rho + r_t + 1}$. Hence the update rule takes the form of a weighted average, with fixed weights, between two delta rules: one including a prediction error between the prior mean and the current mean $(\mu_0 - \hat{\mu}_t)$ and one including a prediction error between the observed sample and the current mean $(y_{t+1} - \hat{\mu}_t)$, both with surprise-modulated learning rates.

In Nassar et al. (2010, 2012) the true new mean after a change point is drawn from a uniform distribution with a range of values much larger than the width of the Gaussian

likelihood. Their derivations implicitly approximate the uniform distribution with a Gaussian distribution with $\sigma_0 \gg \sigma$. Note that if $\sigma_0 \gg \sigma$ then $\rho \to 0$, so that the first term of Equation 2.77 disappears, and $\alpha_{t+1} = \frac{1+\gamma_{t+1} r_t}{1+r_t}$. This results in the delta-rule of the original algorithm in Equation 2.74 and Equation 2.75, with $\gamma_{t+1} = \Omega_{t+1}$.

All of the calculations so far were done by assuming that r_t is known. However, for the case of a non-stationary regime with a history of change points, the time interval r_t is not known. Nassar et al. (2010, 2012) used the expected time interval $\hat{r}_t$ as an estimate. We make a distinction here between Nassar et al. (2012) and Nassar et al. (2010):

In Nassar et al. (2010) $\hat{r}_t$ is calculated recursively on each trial in the same spirit as Equation 2.10: $\hat{r}_{t+1} = (1-\gamma_{t+1})(\hat{r}_t+1)+\gamma_{t+1}$, i.e., at each time step, there is a probability $(1-\gamma_{t+1})$ that $\hat{r}_t$ increments by 1 and a probability γ_{t+1} that it is reset to 1. So $\hat{r}_{t+1}$ is the weighted sum of these two outcomes. Hence, Equation 2.77 combined with the expected time interval $\hat{r}_t$ constitutes a generalization of the update rule of Nassar et al. (2010) for the case of Gaussian prior $\mathcal{N}(\mu_0, \sigma_0^2)$. We call this algorithm Nas10* (see Supplementary Material for the pseudocode).

In Nassar et al. (2012), the variance $\hat{\sigma}_{t+1}^2 = \mathrm{Var}[\mu_{t+1}|y_{1:t+1}]$ is estimated given $\hat{\mu}_t$, $\hat{r}_t$, and $\hat{\sigma}_t^2$. Based on this variance, $\hat{r}_{t+1} = \frac{\sigma^2}{\hat{\sigma}_{t+1}^2} - \frac{\sigma^2}{\sigma_0^2}$ is computed. The derivation of the recursive computation of $\hat{\sigma}_{t+1}^2$ for the case of Gaussian priors can be found in the Supplementary Material. We call the combination of Equation 2.77 with this way of computing the expected time interval $\hat{r}_t$ Nas12* (see Supplementary Material for the pseudocode). These two versions of calculating $\hat{r}_t$ in Nassar et al. (2010) and Nassar et al. (2012) give different results, and we compare our algorithms with both Nas10* and Nas12* in our simulations. Note that, as discussed in the section "Online Bayesian inference modulated by surprise" of the Results, the posterior belief at time $t+1$ does not generally belong to the same family of distributions as the belief of time t. However, we therefore approximate for both algorithms the posterior belief $P(\theta|y_{1:t+1})$ by a Gaussian.

Nassar's algorithm and Particle Filtering with one particle

In the case of Particle Filtering (cf. Equation 2.23) with only one particle, at each time step we sample the particle's hidden state with change probability $\Psi(c_{t+1}^{(1)} = 1|c_{1:t}^{(1)}, y_{1:t+1}) = \gamma_{t+1}$, generating a posterior belief that takes two possible values with probability (according to the proposal distribution)

$$\begin{aligned} \Psi\Big(\hat{\mathbb{b}}^{(t+1)}(\theta) = \hat{\mathbb{b}}_B^{(t+1)}(\theta)|\ y_{t+1}\Big) &= 1-\gamma_{t+1}, \\ \Psi\Big(\hat{\mathbb{b}}^{(t+1)}(\theta) = P(\theta|y_{t+1})|\ y_{t+1}\Big) &= \gamma_{t+1}. \end{aligned} \tag{2.78}$$

So, *in expectation*, the updated belief will be

$$\mathbb{E}_\Psi[\hat{\mathbb{b}}^{(t+1)}(\theta)] = (1-\gamma_{t+1})\hat{\mathbb{b}}_B^{(t+1)}(\theta) + \gamma_{t+1}P(\theta|y_{t+1}). \tag{2.79}$$

If we apply Equation 2.79 to $\hat{\mu}_{t+1}$, we find that $\mathbb{E}_\Psi[\hat{\mu}^{(t+1)}]$, is identical to the generalization of Nassar et al. (2010) (see Equation 2.76).

Moreover, in Particle Filtering with a single particle, we sample the particle's hidden state, which is equivalent to sampling the interval $\hat{R}_{t+1}$. Because $\hat{R}_{t+1}$ takes the value $\hat{r}_t + 1$ with $(1-\gamma_{t+1})$ and the value of 1 (=reset) with probability γ_{t+1}, the *expected value* of $\hat{R}_{t+1}$ is

$$\mathbb{E}_\Psi[\hat{R}_{t+1}] = (1-\gamma_{t+1})(\hat{r}_t + 1) + \gamma_{t+1}. \tag{2.80}$$

In other words, in Nassar et al. (2010), the belief is updated based on the *expected* $\hat{r}_t$, whereas in Particle Filtering with one particle, the belief is updated using the *sampled* $\hat{r}_t$.

In summary, the two methods will give different estimates on a trial-per-trial basis, but the same result in expectation. The pseudocode for Particle Filtering with one particle for the particular case of the Gaussian estimation task can be found in the Supplementary Material.

2.4.6 Application to the exponential family

For our all three algorithms Variational SMiLe, Message Passing with fixed number N of particles, and Particle Filtering, we derive compact update rules for $\hat{\mathbb{b}}^{(t+1)}(\theta)$ when the likelihood function $P_Y(y|\theta)$ is in the exponential family and $\mathbb{b}^{(0)}(\theta)$ is its conjugate prior. In that case, the likelihood function has the form

$$P_Y(y|\theta) = h(y)\exp\big(\theta^T\phi(y) - A(\theta)\big), \tag{2.81}$$

where θ is the vector of natural parameters, $h(y)$ is a positive function, $\phi(y)$ is the vector of sufficient statistics, and $A(\theta)$ is the normalization factor. Then, the conjugate prior $\mathbb{b}^{(0)}$ has the form

$$\begin{aligned}\mathbb{b}^{(0)}(\theta) &= \mathbf{P}_\mathbb{b}\big(\Theta = \theta; \chi^{(0)}, \nu^{(0)}\big) \\ &= \tilde{h}(\theta) f\big(\chi^{(0)}, \nu^{(0)}\big)\exp\big(\theta^T\chi^{(0)} - \nu^{(0)}A(\theta)\big)\end{aligned} \tag{2.82}$$

where $\chi^{(0)}$ and $\nu^{(0)}$ are the distribution parameters, $\tilde{h}(\theta)$ is a positive function, and $f\big(\chi^{(0)}, \nu^{(0)}\big)$ is the normalization factor. For this setting and while $\mathbb{b}^{(t)} = \mathbf{P}_\mathbb{b}\big(\Theta = \theta; \chi^{(t)}, \nu^{(t)}\big)$, the "Bayes Factor Surprise" has the compact form

$$\mathbf{S}_{\mathrm{BF}}\Big(y_{t+1}; \mathbf{P}_\mathbb{b}\big(\Theta = \theta; \chi^{(t)}, \nu^{(t)}\big)\Big) = \frac{f\big(\chi^{(t)} + \phi(y_{t+1}), \nu^{(t)} + 1\big)}{f\big(\chi^{(0)} + \phi(y_{t+1}), \nu^{(0)} + 1\big)} \frac{f\big(\chi^{(0)}, \nu^{(0)}\big)}{f\big(\chi^{(t)}, \nu^{(t)}\big)}. \tag{2.83}$$

The pseudocode for Variational SMiLe, MPN, and Particle Filtering can be seen in Algorithms 1, 2, and 3, respectively.

2.4.7 Simulation task

In this subsection, we first argue why the mean squared error is a proper measure for comparing different algorithms with each other, and then we explain the version of Leaky integrator which we used for simulations.

Mean squared error as an optimality measure

Consider the case that at each time point t, the goal of an agent is to have an estimation of the parameter Θ_t as a function of the observations $Y_{1:t}$, i.e. $\hat{\Theta}_t = f(Y_{1:t})$. The estimator which minimizes the mean squared error $\mathbf{MSE}[\hat{\Theta}_t] = \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)}\big[(\hat{\Theta}_t - \Theta_t)^2\big]$ is

$$\hat{\Theta}_t^{\mathrm{Opt}} = \mathbb{E}_{\mathbf{P}(\Theta_t|Y_{1:t})}\big[\Theta_t\big] = \mathbb{E}_{\mathbb{b}^{(t)}}\big[\Theta_t\big], \tag{2.84}$$

which is the expected value of Θ_t conditioned on the observations $Y_{1:t}$, or in other words under the Bayes-optimal current belief (see Papoulis and Saunders (1989) for a proof). The MSE for any other estimator $\hat{\Theta}_t$ can be written as (see below for the proof)

$$\begin{aligned} &\mathbf{MSE}[\hat{\Theta}_t] = \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}] + \Delta\mathbf{MSE}[\hat{\Theta}_t], \\ &\text{where } \Delta\mathbf{MSE}[\hat{\Theta}_t] = \mathbb{E}_{\mathbf{P}(Y_{1:t})}\Big[(\hat{\Theta}_t - \hat{\Theta}_t^{\mathrm{Opt}})^2\Big] \geq 0. \end{aligned} \tag{2.85}$$

This means that the MSE for any arbitrary estimator $\hat{\Theta}_t$ includes two terms: the optimal MSE and the mismatch of the actual estimator from the optimal estimator $\hat{\Theta}_t^{\mathrm{Opt}}$. As a result, if the estimator we are interested in is the expected value of Θ_t under the approximate belief $\hat{\mathbb{b}}^{(t)}$ computed by each of our algorithms (i.e. $\hat{\Theta}'_t = \mathbb{E}_{\hat{\mathbb{b}}^{(t)}}\big[\Theta_t\big]$), the second term in Equation 2.85, i.e. the deviation from optimality, is a measure of how good the approximation is.

Proof for the algorithms without sampling:

Consider $\hat{\Theta}_t^{\mathrm{Opt}} = f_{\mathrm{Opt}}(Y_{1:t})$. Then, for any other arbitrary estimator $\hat{\Theta}_t = f(Y_{1:t})$ (except for the ones with sampling), we have

$$\begin{aligned} \mathbf{MSE}[\hat{\Theta}_t] &= \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)}\big[(\hat{\Theta}_t - \Theta_t)^2\big] \\ &= \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)}\big[(f(Y_{1:t}) - \Theta_t)^2\big] \\ &= \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)}\Big[\big((f(Y_{1:t}) - f_{\mathrm{Opt}}(Y_{1:t})) + (f_{\mathrm{Opt}}(Y_{1:t}) - \Theta_t)\big)^2\Big]. \end{aligned} \tag{2.86}$$

The quadratic term in the last line can be expanded and written as

$$\begin{aligned}\mathbf{MSE}[\hat{\Theta}_t] =& \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)}\Big[(f(Y_{1:t}) - f_{\mathrm{Opt}}(Y_{1:t}))^2\Big] + \\ & \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)}\Big[(f_{\mathrm{Opt}}(Y_{1:t}) - \Theta_t)^2\Big] + \\ & 2\mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)}\Big[(f(Y_{1:t}) - f_{\mathrm{Opt}}(Y_{1:t}))(f_{\mathrm{Opt}}(Y_{1:t}) - \Theta_t)\Big].\end{aligned} \tag{2.87}$$

The random variables in the expected value of the first line are not dependent on Θ_t, so it can be computed over $Y_{1:t}$. The expected value of the second line is equal to $\mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}]$. It can also be shown that the expected value of the third line is equal to 0, i.e.

$$\begin{aligned}\text{3rd line } &= 2\mathbb{E}_{\mathbf{P}(Y_{1:t})}\Big[\mathbb{E}_{\mathbf{P}(\Theta_t|Y_{1:t})}\Big[(f(Y_{1:t}) - f_{\mathrm{Opt}}(Y_{1:t}))(f_{\mathrm{Opt}}(Y_{1:t}) - \Theta_t)\Big]\Big] \\ &= 2\mathbb{E}_{\mathbf{P}(Y_{1:t})}\Big[(f(Y_{1:t}) - f_{\mathrm{Opt}}(Y_{1:t}))(f_{\mathrm{Opt}}(Y_{1:t}) - \mathbb{E}_{\mathbf{P}(\Theta_t|Y_{1:t})}[\Theta_t])\Big] = 0,\end{aligned} \tag{2.88}$$

where in the last line we used the definition of the optimal estimator. All together, we have

$$\mathbf{MSE}[\hat{\Theta}_t] = \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}] + \mathbb{E}_{\mathbf{P}(Y_{1:t})}\Big[(\hat{\Theta}_t - \hat{\Theta}_t^{\mathrm{Opt}})^2\Big]. \tag{2.89}$$

Proof for the algorithms with sampling:

For particle filtering (and any kind of estimator with sampling), the estimator is not a deterministic function of observations $Y_{1:t}$. Rather, the estimator is a function of observations as well as a set of random variables (samples) which are drawn from a distribution which is also a function of observations $Y_{1:t}$. In our case, the samples are the sequence of hidden states $C_{1:t}$. The estimator can be written as

$$\hat{\Theta}_t^{\mathrm{PF}} = f(Y_{1:t}, C_{1:t}^{(1:N)}), \tag{2.90}$$

where $C_{1:t}^{(1:N)}$ are N iid samples drawn from the proposal distribution $\Psi(C_{1:t}|Y_{1:t})$. MSE for this estimator should also be averaged over the samples, which leads to

$$\begin{aligned}\mathbf{MSE}[\hat{\Theta}_t^{\mathrm{PF}}] &= \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)\Psi(C_{1:t}^{(1:N)}|Y_{1:t})}\big[(\hat{\Theta}_t^{\mathrm{PF}} - \Theta_t)^2\big] \\ &= \mathbb{E}_{\mathbf{P}(Y_{1:t},\Theta_t)\Psi(C_{1:t}^{(1:N)}|Y_{1:t})}\big[(f(Y_{1:t}, C_{1:t}^{(1:N)}) - \Theta_t)^2\big].\end{aligned} \tag{2.91}$$

Similar to what we did before, the MSE for particle filtering can be written as

$$\begin{aligned}\mathbf{MSE}[\hat{\Theta}_t^{\mathrm{PF}}] &= \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}] + \mathbb{E}_{\mathbf{P}(Y_{1:t})\Psi(C_{1:t}^{(1:N)}|Y_{1:t})}\Big[(\hat{\Theta}_t^{\mathrm{PF}} - \hat{\Theta}_t^{\mathrm{Opt}})^2\Big] \\ &= \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}] + \mathbb{E}_{\mathbf{P}(Y_{1:t})}\Big[\mathbb{E}_{\Psi(C_{1:t}^{(1:N)}|Y_{1:t})}\big[(\hat{\Theta}_t^{\mathrm{PF}} - \hat{\Theta}_t^{\mathrm{Opt}})^2\big]\Big]\end{aligned} \tag{2.92}$$

which can be written in terms of bias and variance over samples as

$$\begin{aligned}\mathbf{MSE}[\hat{\Theta}_t^{\mathrm{PF}}] &= \mathbf{MSE}[\hat{\Theta}_t^{\mathrm{Opt}}] \\ &+ \mathbb{E}_{\mathbf{P}(Y_{1:t})}\Big[\mathrm{Var}_{\Psi(C_{1:t}^{(1:N)}|Y_{1:t})}(\hat{\Theta}_t^{\mathrm{PF}}) + \mathrm{Bias}_{\Psi(C_{1:t}^{(1:N)}|Y_{1:t})}(\hat{\Theta}_t^{\mathrm{PF}}, \hat{\Theta}_t^{\mathrm{Opt}})^2\Big].\end{aligned} \tag{2.93}$$

Leaky integration

Gaussian task: The goal is to have an estimation of the mean of the Gaussian distribution at each time t, denoted by $\hat{\theta}_t$. Given a leak parameter $\omega \in (0, 1]$, the leaky integrator estimation is

$$\hat{\theta}_t = \frac{\sum_{k=1}^{t} \omega^{t-k} y_k}{\sum_{k=1}^{t} \omega^{t-k}}. \tag{2.94}$$

Categorical task: The goal is to have an estimation of the parameters of the categorical distribution at each time t, denoted by $\hat{\theta}_t = [\hat{\theta}_{i,t}]_{i=1}^{N}$ for the case that there are N categories. Given a leak parameter $\omega \in (0, 1]$, the leaky integrator estimation is

$$\hat{\theta}_{i,t} = \frac{\sum_{k=1}^{t} \omega^{t-k} \delta(y_k - i)}{\sum_{k=1}^{t} \omega^{t-k}}, \tag{2.95}$$

where δ is the Kronecker delta function.

2.4.8 Derivation of the Formula Relating Shannon Surprise to the Modulated Learning Rate

Given the defined generative model, the Shannon surprise upon observing y_{t+1} can be written as

$$\begin{aligned}\mathbf{S}_{\mathrm{Sh}}(y_{t+1}; \mathbb{b}^{(t)}) &= \log\Big(\frac{1}{\mathbf{P}(y_{t+1}|y_{1:t})}\Big) \\ &= \log\Big(\frac{1}{(1-p_c)P(y_{t+1}; \mathbb{b}^{(t)}) + p_c P(y_{t+1}; \mathbb{b}^{(0)})}\Big) \\ &= \log\Big(\frac{1}{P(y_{t+1}; \mathbb{b}^{(0)})}\Big) + \log\Big(\frac{1}{p_c}\frac{1}{1 + \frac{1}{m}\frac{1}{\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)})}}\Big) \\ &= \mathbf{S}_{\mathrm{Sh}}(y_{t+1}; \mathbb{b}^{(0)}) + \log\Big(\frac{\gamma_{t+1}}{p_c}\Big),\end{aligned} \tag{2.96}$$

where $\gamma_{t+1} = \gamma\Big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{(t)}), m = \frac{p_c}{1-p_c}\Big)$ of Equation 2.9. As a result, the modulated adaptation rate can be written as in Equation 2.25 and the Bayes Factor Surprise as in Equation 2.27.

2.4.9 Experimental predictions

Setting

Consider a Gaussian task where Y_t can take values in $\mathbb{R}$. The likelihood function $P_Y(y|\theta)$ is defined as

$$P_Y(y|\theta) = \mathcal{N}(y; \theta, \sigma^2), \tag{2.97}$$

where $\sigma \in \mathbb{R}^+$ is the standard deviation, and $\theta \in \mathbb{R}$ is the mean of the distribution, i.e. the parameter of the likelihood. Whenever there is a change in the environment (with probability $p_c \in (0, 1)$), the value θ is drawn from the prior distribution $\mathbb{b}^{(0)}(\theta) = \mathcal{N}(\theta; 0, 1)$.

Theoretical proofs for prediction 1

For our theoretical derivations for our first prediction, we consider the specific but relatively mild assumption that the subjects' belief $\mathbb{b}^{(t)}$ at each time is a Gaussian distribution

$$\mathbb{b}^{(t)}(\theta) = \mathcal{N}(\theta; \hat{\theta}_t, \hat{\sigma}_t^2), \tag{2.98}$$

where $\hat{\theta}_t$ and $\hat{\sigma}_t$ are determined by the learning algorithm and the sequence of observations $y_{1:t}$. This is the case when the subjects use either VarSMiLe, Nas10*, Nas12*, pf1, MP1, or Leaky Integration as their learning rule. With such assumptions, the inferred probability distribution $P(y; \mathbb{b}^{(t)})$ can be written as

$$P(y; \mathbb{b}^{(t)}) = \mathcal{N}(y; \hat{\theta}_t, \sigma^2 + \hat{\sigma}_t^2). \tag{2.99}$$

As mentioned in the Results section, we define, at time t, the prediction error as $\delta_{t+1} = y_{t+1} - \hat{\theta}_t$ and the "sign bias" as $s_{t+1} = \text{sign}(\delta_{t+1}\hat{\theta}_t)$. Then, given an absolute prediction $\hat{\theta} > 0$, an absolute prediction error $\delta > 0$, a standard deviation σ_C, and a sign bias $s \in \{-1, 1\}$, the average Bayes Factor Surprise is computed as

$$\begin{aligned} \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s, \sigma_C) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbf{S}_{\text{BF}}(y_t; \hat{\mathbb{b}}^{(t-1)}), \\ &\text{where } \mathcal{T} = \{t : |\hat{\theta}_{t-1}| = \hat{\theta}, |\delta_t| = \delta, \hat{\sigma}_t = \sigma_C, s_t = s\}. \end{aligned} \tag{2.100}$$

It can easily be shown that the value $\mathbf{S}_{\text{BF}}(y_t; \hat{\mathbb{b}}^{(t-1)})$ is same for all $t \in \mathcal{T}$, and hence the average surprise is same as the surprise for each time point. For example, the average surprise for $s = +1$ is equal to

$$\bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C) = \frac{\mathcal{N}(\hat{\theta} + \delta; 0, \sigma^2 + 1)}{\mathcal{N}(\delta; 0, \sigma^2 + \sigma_C^2)}. \tag{2.101}$$

Similar formulas can be computed for $\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)$. Then, the difference $\Delta\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, \sigma_C) = \bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C) - \bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)$ can be computed as

$$\Delta\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, \sigma_C) = \frac{\mathcal{N}(\hat{\theta} + \delta; 0, \sigma^2 + 1) - \mathcal{N}(\hat{\theta} - \delta; 0, \sigma^2 + 1)}{\mathcal{N}(\delta; 0, \sigma^2 + \sigma_C^2)}. \tag{2.102}$$

It can be shown that

$$\Delta\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, \sigma_C) < 0 \text{ and } \frac{\partial}{\partial\delta}\Delta\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, \sigma_C) < 0, \tag{2.103}$$

for all $\hat{\theta} > 0$, $\delta > 0$, and $\sigma_C > 0$. The first inequality is trivial, and the proof for the second inequality is given below.

The average Shannon Surprise can be computed in a similar way. For example, for $s = +1$, we have

$$\bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, s = +1, \sigma_C) = -\log\Big(p_c\mathcal{N}(\hat{\theta} + \delta; 0, \sigma^2 + 1) + (1 - p_c)\mathcal{N}(\delta; 0, \sigma^2 + \sigma_C^2)\Big), \tag{2.104}$$

and then the difference $\Delta\bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, \sigma_C) = \bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, s = +1, \sigma_C) - \bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, s = -1, \sigma_C)$ can be computed as

$$\Delta\bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, \sigma_C) = \log\Big(\frac{1 + m\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)}{1 + m\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C)}\Big), \tag{2.105}$$

where $m = \frac{p_c}{1-p_c}$. Then, using the results for the Bayes Factor Surprise, we have

$$\Delta\bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, \sigma_C) > 0 \text{ and } \frac{\partial}{\partial\delta}\Delta\bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, \sigma_C) > 0, \tag{2.106}$$

for all $\hat{\theta} > 0$, $\delta > 0$, and $\sigma_C > 0$. See below for the proof of the second inequality.

Proof of the 2nd inequality for $\mathbf{S}_{\mathrm{BF}}$: Let us define the variables

$$\sigma_d^2 = \sigma^2 + \sigma_C^2 \quad \sigma_n^2 = \sigma^2 + 1, \tag{2.107}$$

as well as the functions

$$\begin{aligned} f_1(\delta) &= \bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C) = \frac{\sigma_d}{\sigma_n}\exp\big(\frac{\delta^2}{2\sigma_d^2} - \frac{(\delta + \hat{\theta})^2}{2\sigma_n^2}\big) \\ f_2(\delta) &= \bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C) = \frac{\sigma_d}{\sigma_n}\exp\big(\frac{\delta^2}{2\sigma_d^2} - \frac{(\delta - \hat{\theta})^2}{2\sigma_n^2}\big) \\ f(\delta) &= \Delta\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, \sigma_C) = f_1(\delta) - f_2(\delta). \end{aligned} \tag{2.108}$$

The following inequalities hold true

$$\begin{aligned} f(\delta) < 0 \Rightarrow f_1(\delta) < f_2(\delta) \\ \sigma_C^2 < \sigma_0^2 = 1 \Rightarrow \sigma_d^2 < \sigma_n^2. \end{aligned} \tag{2.109}$$

Then, the derivative of $f(\delta)$ can be compute as

$$\begin{aligned} \frac{d}{d\delta} f(\delta) &= f_1(\delta)\big(\frac{\delta}{\sigma_d^2} - \frac{\delta + \hat{\theta}}{\sigma_n^2}\big) - f_2(\delta)\big(\frac{\delta}{\sigma_d^2} - \frac{\delta - \hat{\theta}}{\sigma_n^2}\big) \\ &= -\frac{\hat{\theta}}{\sigma_n^2}\big(f_1(\delta) + f_2(\delta)\big)\Big(1 - \frac{f_1(\delta) - f_2(\delta)}{f_1(\delta) + f_2(\delta)}\frac{\delta}{\hat{\theta}}\big(\frac{\sigma_n^2}{\sigma_d^2} - 1\big)\Big) \\ &< 0. \end{aligned} \tag{2.110}$$

Therefore we have $\frac{\partial}{\partial \delta}\Delta\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, \sigma_C) = \frac{d}{d\delta} f(\delta) < 0$.

Proof of the 2nd inequality for $\mathbf{S}_{\mathrm{Sh}}$: Using the functions we defined for the previous proof, and after computing the partial derivative of Equation 2.105, we have

$$\begin{aligned} \frac{\partial}{\partial \delta}\Delta\bar{\mathbf{S}}_{\mathrm{Sh}}(\hat{\theta}, \delta, \sigma_C) &= \frac{\partial}{\partial \delta}\log\Big(\frac{1 + m\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)}{1 + m\bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C)}\Big) \\ &= \frac{d}{d\delta}\log\Big(\frac{1 + mf_2(\delta)}{1 + mf_1(\delta)}\Big). \end{aligned} \tag{2.111}$$

The derivative of the last term can be written in terms of the derivates of f_1 and f_2, indicated by f_1' and f_2', respectively,

$$\begin{aligned} \frac{\partial}{\partial \delta}\Delta\bar{\mathbf{S}}_{\mathrm{Sh}}(\hat{\theta}, \delta, \sigma_C) &= \frac{mf_2'(\delta)}{1 + mf_2(\delta)} - \frac{mf_1'(\delta)}{1 + mf_1(\delta)} \\ &= \frac{-mf'(\delta)}{\big(1 + mf_1(\delta)\big)\big(1 + mf_2(\delta)\big)} + \frac{m^2(f_1f_2' - f_1'f_2)(\delta)}{\big(1 + mf_1(\delta)\big)\big(1 + mf_2(\delta)\big)}. \end{aligned} \tag{2.112}$$

The 1st term is always positive based on the proof for $\mathbf{S}_{\mathrm{BF}}$. The 2nd term is also always positive, because

$$\begin{aligned} (f_1f_2' - f_1'f_2)(\delta) &= f_1(\delta)f_2(\delta)\Big(\big(\frac{\delta}{\sigma_d^2} - \frac{\delta - \hat{\theta}}{\sigma_n^2}\big) - \big(\frac{\delta}{\sigma_d^2} - \frac{\delta + \hat{\theta}}{\sigma_n^2}\big)\Big) \\ &= f_1(\delta)f_2(\delta)\frac{2\hat{\theta}}{\sigma_n^2} > 0. \end{aligned} \tag{2.113}$$

As a result, we have $\frac{\partial}{\partial \delta}\Delta\bar{\mathbf{S}}_{\mathrm{Sh}}(\hat{\theta}, \delta, \sigma_C) > 0$.

Simulation procedure for prediction 1

In order to relax the main assumption of our theoretical proofs (i.e. the belief is always a Gaussian distribution), to include the practical difficulties of a real experiment (e.g. to use $|\delta_t| \approx \delta$ instead of $|\delta_t| = \delta$), and to have an estimation of the effect size, we also performed simulations for our first experimental prediction.

For each simulated subject, the procedure of our simulation was as follows:

1. We fixed the hyper parameters σ^2 and p_c for producing samples.

2. We selected a learning algorithm (e.g. pf20) and fixed its corresponding tuned parameters (based on our simulations in the Results section).

3. We applied the learning algorithm over a sequence of observations $y_{1:T}$. Note that in a real experiment, this step can be done through a few episodes, which makes it possible to have a long sequence of observations, i.e. large T.

4. At each time t, we saved the values y_t, $\hat{\theta}_t$, $\hat{\sigma}_t$, δ_t, s_t, $\mathbf{S}_{\mathrm{Sh}}(y_t; \hat{\mathbb{b}}^{(t-1)})$, and $\mathbf{S}_{\mathrm{BF}}(y_t; \hat{\mathbb{b}}^{(t-1)})$.

Then, given an absolute prediction $\hat{\theta} > 0$, an absolute prediction error $\delta > 0$, a standard deviation $\sigma_C > 0$, and a sign bias $s \in \{-1, 1\}$, we defined the set of time points

$$\mathcal{T} = \{1 < t \leq T : ||\hat{\theta}_{t-1}| - \hat{\theta}| < \Delta\theta, ||\delta_t| - \delta| < \Delta\delta, |\hat{\sigma}_t - \sigma_C| < \Delta\sigma_C, s_t = s\}, \quad (2.114)$$

where $||\hat{\theta}_{t-1}| - \hat{\theta}| < \Delta\theta$, $||\delta_t| - \delta| < \Delta\delta$, and $|\hat{\sigma}_t - \sigma_C| < \Delta\sigma_C$ are equivalent to $|\hat{\theta}_{t-1}| \approx \hat{\theta}$, $|\delta_t| \approx \delta$, and $\hat{\sigma}_t \approx \sigma_C$, respectively. $\Delta\theta$, $\Delta\delta$, and $\Delta\sigma_C$ are positive real values that should be determined based on practical limitations (mainly the length of the observation sequence T). We then computed the average surprise values as

$$\begin{aligned} \bar{\mathbf{S}}_{\mathrm{BF}}(\hat{\theta}, \delta, s, \sigma_C) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbf{S}_{\mathrm{BF}}(y_t; \hat{\mathbb{b}}^{(t-1)}) \\ \bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, s, \sigma_C) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbf{S}_{\mathrm{Sh}}(y_t; \hat{\mathbb{b}}^{(t-1)}). \end{aligned} \quad (2.115)$$

We repeated this procedure for N different simulated subjects (with different random seeds). The average of $\bar{\mathbf{S}}_{\mathrm{BF}}$ and $\bar{\mathbf{S}}_{Sh}$ over $N = 20$ subjects, for two learning algorithms (i.e. Nas12* and pf20), and for $T = 500$, $\hat{\theta} = 1$, $\sigma_C = 0.5$, $\Delta\theta = 0.25$, $\Delta\delta = 0.1$, and $\Delta\sigma_C = 1$ is shown in Fig. 2.10B. The results are the same as what was predicted by our theoretical analysis.

Simulation procedure for prediction 2

For our second prediction, the theoretical proof is trivial. However, in order to have a setting similar to a real experiment (e.g. to use $P(y_{t+1};\hat{\mathbb{b}}^{(t)}) \approx p$ instead of $P(y_{t+1};\hat{\mathbb{b}}^{(t)}) = p$), and to have an estimation of the effect size, we used simulations also for our second experimental predictions.

We followed the same procedure as the one for the simulation of the first prediction. For each simulated subject, and at each time t, we saved the quantities $P(y_{t+1};\hat{\mathbb{b}}^{(t)})$, $P(y_{t+1};\hat{\mathbb{b}}^{(0)})$, $\mathbf{S}_{\mathrm{Sh}}(y_t;\hat{\mathbb{b}}^{(t-1)})$, and $\mathbf{S}_{\mathrm{BF}}(y_t;\hat{\mathbb{b}}^{(t-1)})$. Then, for a given a probability value $p > 0$, we defined the set of time points

$$\mathcal{T} = \{0 \leq t \leq T : |P(y_{t+1};\hat{\mathbb{b}}^{(t)}) - p| < \Delta p, |P(y_{t+1};\hat{\mathbb{b}}^{(0)}) - p| < \Delta p\}, \tag{2.116}$$

where $|P(y_{t+1};\hat{\mathbb{b}}^{(t)})-p| < \Delta p$ and $|P(y_{t+1};\hat{\mathbb{b}}^{(0)})-p| < \Delta p$ are equivalent to $P(y_{t+1};\hat{\mathbb{b}}^{(t)}) \approx p$ and $P(y_{t+1};\hat{\mathbb{b}}^{(0)}) \approx p$, respectively. Δp is a positive real value that should be determined based on practical limitations (mainly the length of the observation sequence T). We then computed the average surprise $\bar{\mathbf{S}}_{\mathrm{BF}}(p)$ and $\bar{\mathbf{S}}_{\mathrm{Sh}}(p)$ over $\mathcal{T}$ for each value of p. We repeated this procedure for N different simulated subjects (with different random seeds). The average of $\bar{\mathbf{S}}_{\mathrm{BF}}$ and $\bar{\mathbf{S}}_{Sh}$ over $N = 20$ subjects, for two learning algorithms (i.e. Nas12* and pf20), and for $T = 500$ and $\Delta p = 0.0125$ is shown in Fig. 2.11B.

2.5 Supplementary Material

Modified algorithm of Nassar et al. (2010, 2012): Adaptation for Gaussian prior

Recursive update of the estimated mean for Gaussian prior

Let us first consider the case of a stationary regime (i.e. no change points) where observed samples are drawn from a Gaussian distribution with known variance, i.e. $y_{t+1}|\theta \sim \mathcal{N}(\theta, \sigma^2)$, and the parameter θ is also drawn from a Gaussian distribution $\theta \sim \mathcal{N}(\mu_0, \sigma_0^2)$. After having observed samples $y_1, ..., y_{t+1}$, it can be shown that, using Bayes' rule, the posterior distribution $P(\theta|y_{1:t+1}) = \mathbb{b}_B^{(t+1)}(\theta)$ is

$$P(\theta|y_{1:t+1}) = \mathcal{N}\Big(\theta; \mu_{B,t+1} = \frac{1}{\frac{1}{\sigma_0^2} + \frac{t+1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=1}^{t+1} y_i}{\sigma^2}\Big), \sigma_{B,t+1}^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{t+1}{\sigma^2}}\Big). \tag{S1}$$

An estimate of θ is its expected value $\mathbb{E}(\theta|y_{1:t+1}) = \mu_{B,t+1}$.

In a non-stationary regime where, after having observed $y_1, ..., y_t$ from the same hidden state, there is the possibility for a change point upon observing y_{t+1}, the posterior

distribution is

$$P(\theta|y_{1:t+1}) = (1-\gamma_{t+1})P(\theta|y_{1:t+1}, c_{t+1}=0) + \gamma_{t+1}P(\theta|y_{t+1}, c_{t+1}=1)\,. \tag{S2}$$

To facilitate notation in this subsection we denote $c_{t+1}=0$ as "stay" and $c_{t+1}=1$ as "change" so that

$$P(\theta|y_{1:t+1}) = (1-\gamma_{t+1})P(\theta|y_{1:t+1}, \text{stay}) + \gamma_{t+1}P(\theta|y_{t+1}, \text{change}) \tag{S3}$$

Note that the above is equivalent to Bayesian recursive formula (Equation 2.10) of the main text, where γ_{t+1} is the adaptation rate we saw in Equation 2.9 of the main text, and is essentially the probability to change given the new observation, i.e. $\mathbf{P}(c_{t+1}=1|y_{1:t+1})$. In Nassar et al. (2010) this quantity is denoted as Ω_{t+1}. Taking Equation S1 into account we have

$$\begin{aligned}
\mathbb{E}(\theta|y_{1:t+1}, \text{stay}) = \mu_{B,t+1} &= \frac{1}{\frac{1}{\sigma_0^2}+\frac{r_t+1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^{t+1} y_i}{\sigma^2}\Big)\,, \\
\mathbb{E}(\theta|y_{1:t+1}, \text{change}) &= \frac{1}{\frac{1}{\sigma_0^2}+\frac{1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2}\Big)\,,
\end{aligned} \tag{S4}$$

where r_t is the time interval of observations coming from the same hidden state, calculated at time t. Taking the expectation of Equation S3 the estimated mean upon observing the new sample y_{t+1} is

$$\hat{\mu}_{t+1} = (1-\gamma)\frac{1}{\frac{1}{\sigma_0^2}+\frac{r_t+1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^{t+1} y_i}{\sigma^2}\Big) + \gamma\frac{1}{\frac{1}{\sigma_0^2}+\frac{1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2}\Big)\,, \tag{S5}$$

where we dropped the subscript $t+1$ in γ to simplify notations. We have

$$\hat{\mu}_{t+1} = (1-\gamma)\frac{1}{\frac{1}{\sigma_0^2}+\frac{r_t+1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^{t} y_i}{\sigma^2} + \frac{y_{t+1}}{\sigma^2}\Big) + \gamma\frac{1}{\frac{1}{\sigma_0^2}+\frac{1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2}\Big)\,. \tag{S6}$$

Because $\hat{\mu}_t = \frac{1}{\frac{1}{\sigma_0^2}+\frac{r_t}{\sigma^2}}\Big(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^{t} y_i}{\sigma^2}\Big)$, after a few lines of algebra we have

$$\hat{\mu}_{t+1} = (1-\gamma)\hat{\mu}_t + \gamma\mu_0 + (1-\gamma)\frac{1}{\frac{\sigma^2}{\sigma_0^2}+r_t+1}(y_{t+1}-\hat{\mu}_t) + \gamma\frac{1}{\frac{\sigma^2}{\sigma_0^2}+1}(y_{t+1}-\mu_0). \tag{S7}$$

We now define $\rho = \frac{\sigma^2}{\sigma_0^2}$ and find

$$\hat{\mu}_{t+1} = (1-\gamma)\hat{\mu}_t + \gamma\mu_0 + (1-\gamma)\frac{1}{\rho+r_t+1}(y_{t+1}-\hat{\mu}_t) + \gamma\frac{1}{\rho+1}(y_{t+1}-\mu_0)\,. \tag{S8}$$

A rearrangement of the terms and inclusion of the dependency of γ on time yields

$$\hat{\mu}_{t+1} = (1-\gamma_{t+1})\Big(\hat{\mu}_t + \frac{1}{\rho + r_t + 1}(y_{t+1} - \hat{\mu}_t)\Big) + \gamma_{t+1}\Big(\mu_0 + \frac{1}{\rho+1}(y_{t+1} - \mu_0)\Big)\,. \quad \text{(S9)}$$

In order to obtain a form similar to the one of Nassar et al. (2010, 2012) we continue and we spell out the terms that include the quantities $\hat{\mu}_t, \mu_0$ and y_{t+1}

$$\begin{aligned}\hat{\mu}_{t+1} &= (1-\gamma)\hat{\mu}_t - (1-\gamma)\frac{1}{\rho + r_t + 1}\hat{\mu}_t \\ &+ \gamma\mu_0 - \gamma\frac{1}{\rho+1}\mu_0 \\ &+ (1-\gamma)\frac{1}{\rho + r_t + 1}y_{t+1} + \gamma\frac{1}{\rho+1}y_{t+1}\end{aligned} \quad \text{(S10)}$$

Using that $\frac{1}{\rho+r_t+1} = \frac{1}{\rho+1} - \frac{r_t}{(\rho+1)(\rho+r_t+1)}$ we have

$$\begin{aligned}\hat{\mu}_{t+1} &= (1-\gamma)\hat{\mu}_t - (1-\gamma)\frac{1}{\rho+1}\hat{\mu}_t + (1-\gamma)\frac{r_t}{(\rho+1)(\rho+r_t+1)}\hat{\mu}_t \\ &+ \gamma\mu_0 - \gamma\frac{1}{\rho+1}\mu_0 \\ &+ (1-\gamma)\frac{1}{\rho+1}y_{t+1} - (1-\gamma)\frac{r_t}{(\rho+1)(\rho+r_t+1)}y_{t+1} + \gamma\frac{1}{\rho+1}y_{t+1}.\end{aligned} \quad \text{(S11)}$$

After a further step of algebra we arrive at

$$\hat{\mu}_{t+1} = \frac{\rho}{\rho+1}\Big((1-\gamma)\hat{\mu}_t + \gamma\mu_0\Big) + \frac{1}{\rho+1}\Big((1-\gamma)\frac{r_t}{\rho+r_t+1}(\hat{\mu}_t - y_{t+1}) + y_{t+1}\Big)\,. \quad \text{(S12)}$$

If we define $1-\alpha = (1-\gamma)\frac{r_t}{\rho+r_t+1} \Rightarrow \alpha = 1-(1-\gamma)\frac{r_t}{\rho+r_t+1} \Rightarrow \alpha = \frac{\rho+\gamma r_t+1}{\rho+r_t+1}$ and rearrange the terms, we have

$$\begin{aligned}\hat{\mu}_{t+1} &= \frac{\rho}{\rho+1}\Big((1-\gamma)\hat{\mu}_t + \gamma\mu_0\Big) + \frac{1}{\rho+1}\Big((1-\alpha)\hat{\mu}_t + \alpha y_{t+1}\Big) \\ \hat{\mu}_{t+1} &= \frac{\rho}{\rho+1}\Big(\hat{\mu}_t + \gamma(\mu_0 - \hat{\mu}_t)\Big) + \frac{1}{\rho+1}\Big(\hat{\mu}_t + \alpha(y_{t+1} - \hat{\mu}_t)\Big)\,.\end{aligned} \quad \text{(S13)}$$

Adding back the dependency of γ and α on time we finally have

$$\hat{\mu}_{t+1} = \frac{\rho}{\rho+1}\Big(\hat{\mu}_t + \gamma_{t+1}(\mu_0 - \hat{\mu}_t)\Big) + \frac{1}{\rho+1}\Big(\hat{\mu}_t + \alpha_{t+1}(y_{t+1} - \hat{\mu}_t)\Big)\,. \quad \text{(S14)}$$

Recursive update of the the Estimated Variance for Gaussian Prior

In Nassar et al. (2012) the authors calculate first the variance $\hat{\sigma}^2_{t+1} = \mathrm{Var}(\theta|y_{1:t+1})$ and based on this compute then $\hat{r}_{t+1}$. We derive here these calculations for the case of

Gaussian prior. We remind once again that

$$P(\theta|y_{1:t+1}) = (1-\gamma_{t+1})P(\theta|y_{1:t+1}, \text{stay}) + \gamma_{t+1}P(\theta|y_{t+1}, \text{change}) \tag{S15}$$

Then for the variance $\hat{\sigma}^2_{t+1} = \text{Var}(\theta|y_{1:t+1})$ we have

$$\begin{aligned}\hat{\sigma}^2_{t+1} &= (1-\gamma)\sigma^2_{stay} + \gamma\sigma^2_{change} + (1-\gamma)\gamma(\mu_{stay} - \mu_{change})^2 \\ &= (1-\gamma)\sigma^2_{B,t+1} + \gamma\sigma^2_{change} + (1-\gamma)\gamma(\mu_{B,t+1} - \mu_{change})^2\end{aligned} \tag{S16}$$

where $\sigma^2_{B,t+1} = \frac{1}{\frac{1}{\sigma^2_0} + \frac{r_{t+1}}{\sigma^2}}$ and $\sigma^2_{change} = \frac{1}{\frac{1}{\sigma^2_0} + \frac{1}{\sigma^2}}$.

We have defined earlier $\rho = \frac{\sigma^2}{\sigma^2_0}$ so that

$$A = (1-\gamma)\sigma^2_{B,t+1} + \gamma\sigma^2_{change} = (1-\gamma)\frac{\sigma^2}{\rho + r_t + 1} + \gamma\frac{\sigma^2}{\rho+1} \tag{S17}$$

Using, as before, that $\frac{1}{\rho+r_t+1} = \frac{1}{\rho+1} - \frac{r_t}{(\rho+1)(\rho+r_t+1)}$ we have

$$A = \frac{\sigma^2}{\rho+1}\Big(1 - (1-\gamma)\frac{1}{\rho + r_t + 1}\Big). \tag{S18}$$

We have defined earlier the learning rate $\alpha = 1 - (1-\gamma)\frac{r_t}{\rho+r_t+1}$, so we can write

$$A = \frac{\sigma^2}{\rho+1}\alpha \tag{S19}$$

Note that $\mu_t = \frac{1}{\frac{1}{\sigma^2_0} + \frac{r_t}{\sigma^2}}\Big(\frac{\mu_0}{\sigma^2_0} + \frac{\sum_{i=t+1-r_t}^{t} y_i}{\sigma^2}\Big)$ so for the calculation of the last term we have

$$\begin{aligned}B &= \mu_{B,t+1} - \mu_{change} \\ &= \frac{1}{\frac{1}{\sigma^2_0} + \frac{r_{t+1}}{\sigma^2}}\Big(\frac{\mu_0}{\sigma^2_0} + \frac{\sum_{i=t+1-r_t}^{t+1} y_i}{\sigma^2}\Big) - \frac{1}{\frac{1}{\sigma^2_0} + \frac{1}{\sigma^2}}\Big(\frac{\mu_0}{\sigma^2_0} + \frac{y_{t+1}}{\sigma^2}\Big).\end{aligned} \tag{S20}$$

We now rearrange terms

$$B = \mu_t + (\frac{1}{\rho+1} - \frac{r_t}{(\rho+1)(\rho+r_t+1)})(y_{t+1} - \mu_t) - \mu_0 - \frac{1}{\rho+1}(y_{t+1} - \mu_0), \tag{S21}$$

and finally we have

$$\hat{\sigma}^2_{t+1} = \frac{\sigma^2}{\rho+1}\alpha + (1-\gamma)\gamma B^2. \tag{S22}$$

Implementation of Nas10*, Nas12* and Particle Filtering with 1 particle for the Gaussian estimation task

We provide here the pseudocode for the algorithms Nas10*, Nas12* and Particle Filtering with 1 particle for the Gaussian estimation task. Observations are drawn from a Gaussian distribution with known variance and unknown mean, i.e. $y_{t+1}|\mu_{t+1} \sim \mathcal{N}(\mu_{t+1}, \sigma^2)$ and $\theta_t = \mu_t$. When there is a change, the parameter μ is also drawn from a Gaussian distribution $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$. All three algorithms estimate the expected μ_{t+1} (i.e. $\hat{\mu}_{t+1}$) upon observing a new sample y_{t+1}.

After re-writing Equation 2.83 we have

$$\mathbf{S}_{\text{BF}}\Big(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t\Big) = \frac{\sigma^2 + \hat{\sigma}_t^2}{\sigma^2 + \sigma_0^2} \exp\Big[-\frac{\mu_0^2}{2\sigma_0^2} - \frac{\hat{\mu}_t^2}{2\hat{\sigma}_t^2} + \frac{\frac{\hat{\mu}_t}{\hat{\sigma}_t^2} + \frac{y_{t+1}}{\sigma^2}}{2(\frac{\sigma^2}{\hat{\sigma}_t^2} + 1)} + \frac{\frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2}}{2(\frac{\sigma^2}{\sigma_0^2} + 1)} \Big] \tag{S23}$$

Note that the pseudocode for pf1 provided here, is a translation of Algorithm 3 to the case of a single particle (where there are no weights to calculate) and for the Gaussian distribution as a particular instance of the exponential family.

Algorithm S1 Pseudocode for Nas10* for the Gaussian estimation task

Specify $m = p_c/(1-p_c)$, μ_0, σ_0, σ and $\rho = \sigma^2/\sigma_0^2$.
Initialize $\hat{\mu}_0$, $\hat{\sigma}_0$, $\hat{r}_0$ and $t \leftarrow 0$.
while the sequence is not finished **do**
    Observe y_{t+1}
    # Surprise
    Compute $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t)$ using Equation S23
    # Modulation factor
    Compute $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t), m\big)$ as in Equation 2.9
    # Expected mean
    Compute $\hat{\mu}_{t+1}$ using Equation 2.76
    # Expected time interval
    Compute $\hat{r}_{t+1} = (1 - \gamma_{t+1})(\hat{r}_t + 1) + \gamma_{t+1}$
    # Expected variance
    Compute $\hat{\sigma}_{t+1} = \big[\frac{1}{\sigma_0^2} + \frac{\hat{r}_{t+1}}{\sigma^2}\big]^{-1}$
    # Iterate
    $t \leftarrow t + 1$

Algorithm S2 Pseudocode for Nas12* for the Gaussian estimation task

Specify $m = p_c/(1-p_c)$, μ_0, σ_0, σ and $\rho = \sigma^2/\sigma_0^2$.
Initialize $\hat{\mu}_0$, $\hat{\sigma}_0$, $\hat{r}_0$ and $t \leftarrow 0$.
while the sequence is not finished **do**
    Observe y_{t+1}
    # Surprise
    Compute $\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t)$ using Equation S23
    # Modulation factor
    Compute $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t), m\big)$ as in Equation 2.9
    # Expected mean
    Compute $\hat{\mu}_{t+1}$ using Equation 2.76
    # Expected variance
    Compute the expected variance $\hat{\sigma}_{t+1}$ using Equation S22
    # Expected time interval
    Compute the expected time interval $\hat{r}_{t+1} = \frac{\sigma^2}{\hat{\sigma}_{t+1}^2} - \frac{\sigma^2}{\sigma_0^2}$
    # Iterate
    $t \leftarrow t+1$

Algorithm S3 Pseudocode for Particle Filtering with 1 particle for the Gaussian estimation task

Specify $m = p_c/(1-p_c)$, μ_0, σ_0, σ and $\rho = \sigma^2/\sigma_0^2$.
Initialize $\hat{\mu}_0$, $\hat{\sigma}_0$, $\hat{r}_0$ and $t \leftarrow 0$.
while the sequence is not finished **do**
    Observe y_{t+1}
    # Surprise
    Compute $\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t)$ using Equation S23
    # Modulation factor
    Compute $\gamma_{t+1} = \gamma\big(\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t), m\big)$ as in Equation 2.9
    # Hidden state of particle
    Sample $c_{t+1}^{(1)} \sim \mathrm{Bernoulli}\big(\gamma_{t+1}\big)$
    # Expected mean
    if $c_{t+1}^{(1)} = 0$ **then**
        $\hat{\mu}_{t+1} \leftarrow \hat{\mu}_t + \frac{1}{\rho + \hat{r}_t + 1}(y_{t+1} - \hat{\mu}_t)$ and $\hat{r}_{t+1} \leftarrow \hat{r}_t + 1$
    else
        $\hat{\mu}_{t+1} \leftarrow \mu_0 + \frac{1}{\rho+1}(y_{t+1} - \mu_0)$ and $\hat{r}_{t+1} \leftarrow 1$
    # Expected variance
    Compute the expected variance $\hat{\sigma}_{t+1} = \frac{\sigma^2}{\hat{r}_{t+1} + \rho}$
    # Iterate
    $t \leftarrow t+1$

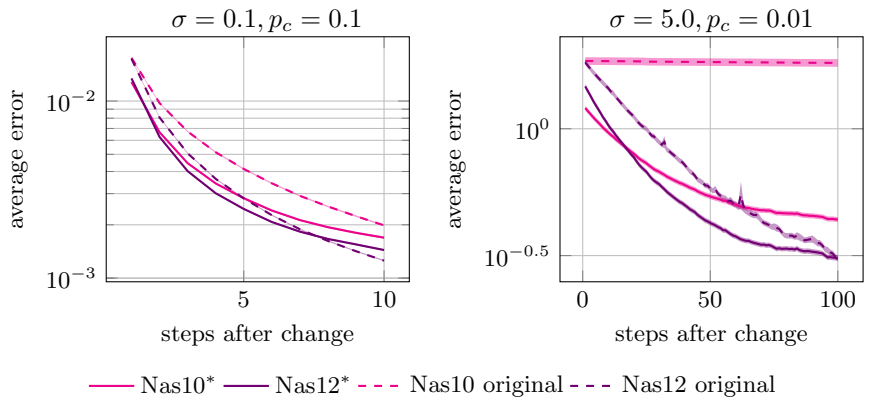


Figure 2.12 – **Gaussian estimation task: Transient performance after changes for original algorithms of Nassar et al. (2010) and Nassar et al. (2012).** Mean squared error for the estimation of μ_t at each time step n after an environmental change, i.e. the average of $\mathbf{MSE}[\hat{\Theta}_t | R_t = n]$ over time; $\sigma = 0.1$, $p_c = 0.1$ (left panel) and $\sigma = 5$, $p_c = 0.01$ (right panel). The shaded area corresponds to the standard error of the mean. *Abbreviations:* Nas10*, Nas12*: Variants of Nassar et al. (2010) and Nassar et al. (2012) respectively, Nas10 Original, Nas12 Original: Original algorithms of Nassar et al. (2010) and Nassar et al. (2012) respectively.

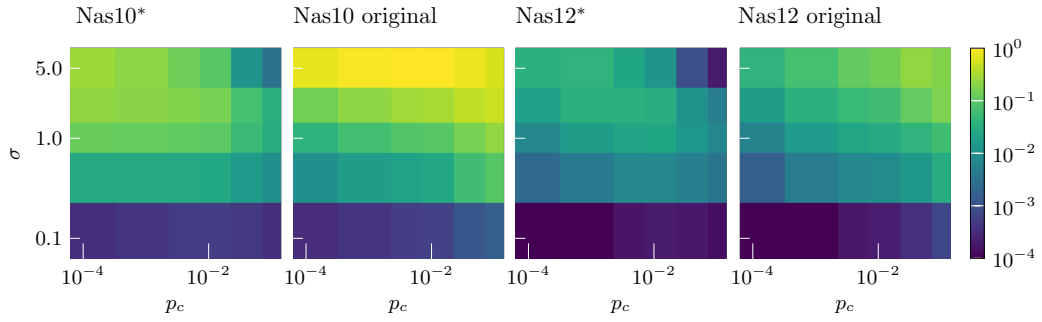


Figure 2.13 – **Gaussian estimation task: Steady-state performance for original algorithms of Nassar et al. (2010) and Nassar et al. (2012).** Difference between the mean squared error of each algorithm and the optimal solution (Exact Bayes), i.e. the average $\mathbf{\Delta MSE}[\hat{\Theta}_t]$ over time for each combination of environmental parameters σ and p_c. *Abbreviations:* Nas10*, Nas12*: Variants of Nassar et al. (2010) and Nassar et al. (2012) respectively, Nas10 Original, Nas12 Original: Original algorithms of Nassar et al. (2010) and Nassar et al. (2012) respectively.

2.6 Contributions

VL, AM and JB conceived and designed the project.
AM defined the Bayes Factor Surprise and worked out the surprise-based interpretation of the exact Bayesian inference, with the help of VL and JB.
VL and AM worked out the surprise-based interpretation of the algorithms, with the help of JB.
VL developed the Particle Filtering algorithm, with the help of AM and JB.
AM developed the Variational SMiLe algorithm.
JB developed the Message-Passing N algorithm.
AM conceived and worked out the experimental prediction 1.
WG conceived the experimental prediction 2.
VL wrote the code for the algorithms, the simulations, and the experimental predictions, with the help and feedback of AM and JB.
VL analyzed the simulation results, with the help of AM and JB.
VL made the figures, with the help of JB.
VL, AM, JB and WG interpreted the results.
VL, AM, JB and WG wrote the manuscript.

3 Surprise is (not) important: model estimation in non-stationary reinforcement learning

This chapter presents research performed in collaboration with Alireza Modirshanechi, Dr. Johanni Brea and Prof. Wulfram Gerstner.

3.1 Introduction

Model-based reinforcement learning agents learn by building an explicit model of the world and present many advantages over model-free learning, such as flexibility to changes. Still, this flexibility is often compromised when abrupt changes occur long after convergence or in cases of unseen situations (Sutton and Barto, 2018). A sudden blockage or opening of a path in a maze, a shift in the reward characteristics, a failure of an agent's system, or a perturbation in the agent-environment interaction are examples of changes likely to be encountered at any time in real-world tasks.

In the recent years the focus is increasingly turning towards this more challenging problem of building agents that are able to quickly adapt in the face of non-stationarity, rather than converge to a sole stationary solution. Depending on the aim, the problem statement and the type of environmental changes, this interest is addressed by approaches such as meta-RL and continual RL (Lomonaco et al., 2019). On the other hand, change-point detection and surprise-based model learning are active fields of research that offer a repertoire of online algorithms for adaptive parameter estimation (Adams and MacKay, 2007). Many recent RL algorithms for non-stationary learning do employ some procedure or measure to detect a change, but transfer of knowledge from the field of model learning to RL seems to be rare. It is thus unclear in which non-stationary learning situations RL could use and profit from advances in the change-point detection field.

In this work, we identify conditions under which surprise, generated by change-point detection, is or is not important for non-stationary RL. Our overarching goal is to

investigate in which non-stationary learning scenarios an artificial or biological agent needs to invest significant resources in rigorous model estimation and in which scenarios approximate and inaccurate models can safely be employed.

To this end, we consider model learning procedures of varying complexity and accuracy and apply them on non-stationary tasks with local – and possibly more difficult to detect – changes, rather than global ones. Previously, we have developed an approximate Bayesian algorithm (Particle Filtering) featuring surprise-based adaptation to changes (Chapter 2). Here, we extend it in order to incorporate the treatment of possible environmental changes happening in the background, in the absence of direct experience, and allow for informed exploration. We couple this approach, as well as simpler Leaky Integration approaches, with the Prioritized Sweeping algorithm from model-based RL (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013) and evaluate their performance. This chapter builds on the work presented in Chapter 2 and applies model and surprise-based learning approaches developed there to reinforcement learning purposes. Our results show that in a task where adapting to abrupt changes is crucial for performance, a surprise-based method, such as Particle Filtering, performs better. In cases of distal changes, where exploration is desired, as well as in cases of higher stochasticity, a simple Leaky Integrator with prior knowledge on the environment is sufficient to achieve high levels of performance.

In the following section, we first recall the general RL framework and notations, and introduce our model learning approaches for non-stationary reward-based tasks. We then present their evaluation on three simulated tasks with different characteristics. Finally, we briefly review related work and discuss possible future directions.

3.2 Learning in non-stationary environments

3.2.1 Reinforcement Learning preliminaries

We employ the widely used Markov Decision Process (MDP) formulation for reinforcement learning problems (Sutton and Barto, 2018). At each time step t the agent observes a state $S_t = s_t \in \mathcal{S}$, where $\mathcal{S} = \{s^{(1)}, s^{(2)}, ..., s^{(n)}\}$ is the set of n possible states, and chooses an action $A_t = a_t \in \mathcal{A}$ from the set of possible actions $\mathcal{A}$. This causes a transition to a state $S_{t+1} = s_{t+1} \in \mathcal{S}$ and the observation of some reward $R_{t+1} = r_{t+1} \in \mathcal{R} \subset \mathbb{R}$. As in Chapter 2, we indicate random variables by capital letters, and values by small letters, and we omit the explicit indication of random variables whenever there is no ambiguity. The transition probability from $S_t = s_t$ to $S_{t+1} = s_{t+1}$ when taking action $A_t = a_t$ is denoted as $T_t(s_t, a_t, s_{t+1}) = \mathbf{P}(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t) \in [0, 1]$, where, as in Chapter 2, $\mathbf{P}$ stands for probability mass function (for the discrete variables, which is the case here), or for probability density function (for continuous variables). The expected reward when taking action $A_t = a$ at $S_t = s$ is $\bar{R}_t(s_t, a_t, s_{t+1}) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s_{t+1}] \in \mathbb{R}$. These quantities are the same as the ones we introduced in Chapter

1 (cf. Equation 1.2 and Equation 1.3), but, in the non-stationary setting we are interested in, they are now also time-dependent. The goal of an RL agent is the maximization of the expected cumulative discounted sum of rewards $\mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}]$, where $\gamma \in [0, 1]$ is a parameter that discounts the importance of distant rewards. In other words, the agent seeks the optimal policy π^*, i.e. the mapping from states to actions or to action selection probabilities $\pi(s_t, a_t) = \mathbf{P}(A_t = a_t | S_t = s_t)$ that maximizes this expected discounted sum.

One way to do so is through the calculation of values, which quantify the "goodness" of selecting certain actions from certain states. The value of selecting action a_t in state s_t and following the policy π thereafter is defined as $Q_t^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s_t, A_t = a_t]$, for all $s \in \mathcal{S}, a \in \mathcal{A}$. A model-based RL agent estimates the Q values of the optimal policy π^*, i.e. the optimal Q^* values, by explicitly learning the model of the world, i.e. the quantities T_t and $\bar{R}_t$, and directly estimating the (Bellman) equation $Q_t(s_t, a_t) = \sum_{s'} \hat{T}_t(s_t, a_t, s') \big(\hat{\bar{R}}_t(s_t, a_t, s') + \gamma V_t(s') \big)$, where $V_t(s') = \max_{a'} Q_t(s', a')$, and $\hat{T}_t$ and $\hat{\bar{R}}_t$ are the estimations of the true T_t and $\bar{R}_t$ by the agent. This can be done through value iteration or through other approximate methods, such as Prioritized Sweeping (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013).

In this work, we focus on the estimation of non-stationary transition probabilities T_t. The framework and the model learning modules we use build on the framework and the algorithms developed in Chapter 2, and we briefly describe them in the following section.

3.2.2 Learning the model of the world

The Generative Model

In Chapter 2 we have considered a hierarchical generative model in discrete time (Equation 2.2 - Equation 2.4 and Fig. 2.1) and studied surprise-based learning in non-stationary tasks. An environment that exhibits sudden changes in the transition probabilities can be modelled as a set of generative models of the same type, one for each state-action pair (s, a). We represent the vector of transition probabilities for a single state-action pair to all states in the environment as the random variable $P_t^{sa} = p_t^{sa} \in [0, 1]^{|\mathcal{S}|}$. To avoid confusion, we recall that $T_t \in [0, 1]$ is the true value of the time-dependent transition probability from a state and an action to another state, and $\hat{T}_t \in [0, 1]$ is the agent's point estimate of T_t. In other words, $P_t^{sa} = \Big(T_t(s, a, s^{(j)}) \Big), j \in \{1, ..., |\mathcal{S}|\}$.

From each state s and action a the agent observes at the next time step a state $s' \in \mathcal{S}$ which is drawn from a Categorical distribution with parameters $P_{t+1}^{sa} = p_{t+1}^{sa}$, i.e. $s' | p_{t+1}^{sa} \sim \text{Cat}(s'; p_{t+1}^{sa})$, where $\dim(p_{t+1}^{sa}) = |\mathcal{S}|$. We consider an environment where changes in the transition probabilities can occur at any time t at any state-action pair, not only at the currently experienced one, and independently of the other pairs, with change probability

$p_c \in (0, 1)$. When there is an environmental change at a state-action pair (s, a), indicated by the event $C^{sa}_{t+1} = 1$, the parameters p^{sa}_{t+1} are drawn from a prior Dirichlet distribution $p^{sa}_{t+1} \sim \mathrm{Dir}(\sigma \cdot \mathbf{1})$, where $\sigma \in (0, \infty)$ is a stochasticity parameter. The next observed state s' at time step $t + 1$ is, however, drawn from the Categorical distribution corresponding only to the currently experienced state-action pair at time t. Thus, the time index t refers to real time, and we define the variable $\mathcal{T}(s, a)$ as the set of timepoints in $[1, t]$ that a particular (s, a) pair is visited.

In other words, the generative model for *each state* (s, a) *pair* independently is

$$\mathbf{P}(c^{sa}_t) = \mathrm{Bernoulli}(c^{sa}_t; p_c)\,, \tag{3.1}$$

$$\mathbf{P}(p^{sa}_t | c^{sa}_t, p^{sa}_{t-1}) = \begin{cases} \delta(p^{sa}_t - p^{sa}_{t-1}) & \text{if} \quad c^{sa}_t = 0\,, \\ \mathrm{Dir}(p^{sa}_t; \sigma \cdot 1) & \text{if} \quad c^{sa}_t = 1\,, \end{cases} \tag{3.2}$$

$$\mathbf{P}(s' | p^{sa}_t) = \mathrm{Cat}(s' | p^{sa}_t) \quad \text{if} \quad t - 1 \in \mathcal{T}(s, a)\,, \tag{3.3}$$

where δ is the Dirac delta distribution. In general, the superscript sa serves as a reminder that all the corresponding quantities refer to a single state-action pair, i.e. they are functions with arguments (s, a). Sometimes, we skip the explicit mention of these arguments to simplify notation.

Here, we focus on non-stationarity in the transition matrix and not in the reward locations or values. However, our framework can easily be applied to other sources of non-stationarity. For reward locations, we can consider a binary indicator random variable I^s_t associated with the occurrence of reward at a certain state at time t, so that $\mathbf{P}(i^s_t) = \mathrm{Bernoulli}(i^s_t; p^s_t)$, where p^s_t is the probability of obtaining reward at a certain state at time t. For the case of non-stationary reward values, we can consider that they are drawn from a Gaussian distribution, for example, with state-dependent mean μ^s_t and variance σ^2, i.e. $r_t | \mu^s_t \sim \mathcal{N}(\mu^s_t, \sigma^2)$, and then the goal of the agent is the estimation of the changing μ^s_t for all $s \in \mathcal{S}$, similar to the Gaussian estimation task of the previous chapter.

It is worth noting that the generative model describes how the agent views and models the world, which, later in our simulations, may or may not be the same as how the world really functions.

Model Learning approaches

We briefly describe here the approaches we use in order to learn the environment's transition probabilities: an adaptive surprise-modulated Particle Filtering algorithm, introduced in Chapter 2 (section 2.2), as well as simple Leaky Integrators with and without prior knowledge.

Particle Filtering. The goal of our particle filter is to approximate the belief

$$\mathbb{b}^{sa,(t)}(p_t^{sa}) = \mathbf{P}(P_t^{sa} = p_t^{sa} | s_{1:t}) \tag{3.4}$$

for each (s, a) pair and recursively update it upon each new observation s_{t+1}.

As we showed in the previous chapter, $\mathbb{b}^{sa,(t)}$ can be approximated as

$$\hat{\mathbb{b}}^{sa,(t)}(p_t^{sa}) = \sum_{i=1}^{N} w_t^{sa,(i)} \hat{\mathbb{b}}_i^{sa,(t)}(p_t^{sa}) = \sum_{i=1}^{N} w_t^{sa,(i)} \mathbf{P}(p_t^{sa} | c_{1:t}^{sa,(i)}, s_{1:t}) \,, \tag{3.5}$$

where $\{c_{1:t}^{sa,(i)}\}_{i=1}^{N}$ is a set of N realization (or samples) of $c_{1:t}^{sa}$ (i.e. N particles) drawn from a proposal distribution $\Psi(c_{1:t}^{sa} | s_{1:t})$, $\{w_t^{sa,(i)}\}_{i=1}^{N}$ are their corresponding weights at time t, and $\hat{\mathbb{b}}_i^{sa,(t)}(p_t^{sa})$ is the approximated belief corresponding to particle i.

In an environment where a change can happen at any time t at any (s, a) pair (and not only at the currently experienced one), performing inference means performing additional background updates in all (s, a) pairs not currently visited, in order to take into account the possibility of changes in some other part of the environment. We therefore make a distinction between the particles' update process for the currently experienced state-action pair, i.e. $t \in \mathcal{T}(s_t, a_t)$, with $S_t = s_t$ and $A_t = a_t$, and for other the state-action pairs (s'', a'') not currently experienced, i.e. $t \notin \mathcal{T}(s_t, a_t)$, which implies $(s'', a'') \neq (s_t, a_t)$.

For the first case, the update rule for the approximated belief over the transition probability vector p_{t+1}^{sa} to all other states, upon the observation of the next state $S_{t+1} = s_{t+1}$ follows Equation 2.22 and Equation 2.23 of Chapter 2 and is the same as for the Categorical estimation task in the Simulations section of Chapter 2 (subsection 2.2.3). After applying the general treatment for the exponential family for this task, it can be shown that the update of the unnormalized weights *for the currently experienced state and action pair* $S_t = s_t$ *and* $A_t = a_t$ takes the simple form

$$\tilde{w}_{t+1}^{sa,(i)} = \Big((1 - p_c) \frac{\alpha_t^{(i)}(s_t, a_t, s_{t+1})}{\sum_{j=1}^{|\mathcal{S}|} \alpha_t^{(i)}(s_t, a_t, s^{(j)})} + p_c \frac{1}{|\mathcal{S}|} \Big) \tilde{w}_t^{sa,(i)}, \tag{3.6}$$

where $\alpha_t^{(i)}(s_t, a_t, s_{t+1}) = \sigma + N_t^{(i)}(s_t, a_t, s_{t+1})$.
Here, $N_t^{(i)}(s_t, a_t, s_{t+1}) = \sum_{t' \in r_t^{sa,(i)}} [S_{t'} = s_{t+1}]$ denotes the actual counts of observing s_{t+1} from (s_t, a_t) within the time interval $r_t^{sa,(i)} = \min\{n \in \mathbb{N} : c_{t-n+1}^{sa,(i)} = 1, s = s_t, a = a_t\}$, i.e. since last change point of the ith particle for this state-action pair. [.] denotes the Iverson bracket (and equals to 1 if the condition within the bracket is fulfilled, 0 otherwise).

Intuitively, as we have shown in detail in the previous chapter, the above equation entails a weighted average between the expected value of the transition probability as calculated under the particle's current belief, and under the uniform prior.

At each time step, we first update the weights according to Equation 3.6 and normalize them. Second, we sample each particle's hidden state $c_{t+1}^{sa,(i)}$ from the proposal distribution with change probability

$$\Psi\big(c_{t+1}^{sa,(i)} = 1 | c_{1:t}^{sa,(i)}, s_{1:t+1}\big) = \gamma\Big(\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \hat{\mathbb{b}}_i^{sa,(t)}), \frac{p_c}{1-p_c}\Big), \tag{3.7}$$

where γ is the adaptation rate and $\mathbf{S}_{\mathrm{BF}}$ the Bayes Factor Surprise we have seen in Chapter 2 (Equation 2.9 and Equation 2.7, respectively). Following the general formulation of the exponential family, it can be shown that the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \hat{\mathbb{b}}_i^{sa,(t)})$ for particle i in this application is given by

$$\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \hat{\mathbb{b}}_i^{sa,(t)}) = \frac{1/|\mathcal{S}|}{\alpha_t^{(i)}(s_t, a_t, s_{t+1}) / \sum_{j=1}^{|\mathcal{S}|} \alpha_t^{(i)}(s_t, a_t, s^{(j)})}. \tag{3.8}$$

For *all other pairs* $(s'', a'') \neq (s_t, a_t)$, there is also a possibility for an independently occurring change. We sample at each time step the hidden state of their particles from the proposal distribution

$$\Psi\big(c_{t+1}^{sa,(i)} = 1 | c_{1:t}^{sa,(i)}, s_{1:t}\big) = P\big(c_{t+1}^{sa,(i)} = 1\big) = p_c. \tag{3.9}$$

This means that at each time step particles change their states "in the background", in the absence of observation, with probability p_c. It can be shown that the values of the particles' weights for the pairs $(s'', a'') \neq (s_t, a_t)$ do not change (See Supplementary Material 3.5.3). Interestingly, this optimal – in the Bayesian sense – updating of the background (s, a) pairs naturally leads to exploratory behavior (see also (Dayan and Sejnowski, 1996)). If an agent has some prior about how the world works and forgets experiences at a rate consistent with the rate of changes in the world, the agent is more likely to explore and discover new ways to reach rewards. In this way, if, for example, the passage from a certain state to a reward was once experienced to be blocked, taking into account that the block may not be permanent influences the estimated Q values and encourages an agent to re-visit the passage after some time.

The implementation details of our particle filtering (e.g. the resampling procedure) are the same as described in the previous chapter (See section 2.4 and Algorithm 3 for the pseudocode).

Finally, the (model-based RL) agent's point estimate $\hat{T}_t$ of the true transition probabilities T_t from any (s, a) pair to any state s' given is calculated from the weighted mean of the parameter P_t^{sa} averaged over the particles, i.e.

$$\hat{T}_t(s, a, s') = \mathbb{E}_{\mathbb{b}^{sa,(t)}}[P_t^{sa}] = \sum_{i=1}^{N} w_t^{sa,(i)} \frac{\alpha_t^{(i)}(s, a, s')}{\sum_{j=1}^{|\mathcal{S}|} \alpha_t^{(i)}(s_t, a_t, s^{(j)})}. \tag{3.10}$$

Leaky Integration with prior knowledge. We employed a simple Leaky Integrator that includes as a free parameter the environment's stochasticity σ. In our simulations we denote this agent as "Leaky prior". For the current state-action pair (s_t, a_t) the outgoing counts to all states $s^{(j)}, j \in \{1, ..., |\mathcal{S}|\}$, upon the observation of the state s_{t+1} are updated as

$$\tilde{N}_{t+1}(s_t, a_t, s^{(j)}) = \eta \tilde{N}_t(s_t, a_t, s^{(j)}) + [s^{(j)} = s_{t+1}], \tag{3.11}$$

where η is a constant leak parameter, [.] is the Iverson bracket, and $\tilde{N}_t(s_t, a_t, s^{(j)})$ are the (leaky) counts of observing $s^{(j)}, j \in \{1, ..., |\mathcal{S}|\}$ from (s, a), which are initialized to zero.

For all background $(s'', a'') \neq (s_t, a_t)$ pairs the counts are leaked as

$$\tilde{N}_{t+1}(s'', a'', s^{(j)}) = \eta \tilde{N}_t(s'', a'', s^{(j)}), \tag{3.12}$$

for all $s'' \in \mathcal{S}$, $a'' \in \mathcal{A}$ and $s^{(j)}, j \in \{1, ..., |\mathcal{S}|\}$.

Concerning the operations on the background (s'', a'') pairs, there is an equivalence between the Particle Filter and the Leaky prior, roughly of the form $\eta \propto (1 - p_c)$, but the exact relationship would also depend on the number of particles. Moreover, in the extreme cases of $\eta = 1$ and $p_c = 0$, the Leaky prior is equivalent to Particle Filtering with one particle.

The estimated transition probabilities from any (s, a) pair to any state s' are then

$$\hat{T}_t(s, a, s') = \mathbb{E}_{\mathbb{b}^{sa,(t)}}[P_t^{sa}] = \frac{\tilde{N}_t(s, a, s') + \sigma}{\sum_{j=1}^{|\mathcal{S}|} \left(\tilde{N}_t(s, a, s^{(j)}) + \sigma\right)}. \tag{3.13}$$

Thus the estimated transition probabilities for all state-action pairs not visited are gradually in time leaked ("forgotten") towards the uniform prior. Similar to the particle filter this background forgetting can promote exploration. This operation of background leakage to the prior is similar to the one performed in the algorithm of Dayan and Sejnowski (1996) (see "Related work" in the Discussion section).

Leaky Integration. Finally, we used a simple Leaky Integrator ("Leaky" in our simulations) that has no knowledge of the stochasticity σ of the environment. For the current state-action pair (s_t, a_t) the outgoing counts to all states $s^{(j)}, j \in \{1, ..., |\mathcal{S}|\}$, upon the observation of the state s_{t+1} undergo the same update as in Equation 3.11, with a free leak parameter η.

For all background $(s'', a'') \neq (s_t, a_t)$ the counts are leaked as

$$\tilde{N}_{t+1}(s'', a'', s^{(j)}) = \eta_{\text{bckgrd}} \tilde{N}_t(s'', a'', s^{(j)}), \tag{3.14}$$

for all $s'' \in \mathcal{S}$, $a'' \in \mathcal{A}$ and $s^{(j)}, j \in \{1, ..., |\mathcal{S}|\}$, where η_{bckgrd} is a constant leak parameter.

The estimated transition probabilities from any (s, a) pair to any state s' are then

$$\hat{T}_t(s, a, s') = \frac{\tilde{N}_t(s, a, s')}{\sum_{j=1}^{|\mathcal{S}|} \tilde{N}_t(s, a, s^{(j)})}. \tag{3.15}$$

In this case, the background leakage does not lead to forgetting, but affects only the confidence of the estimation. In other words, the estimated $\hat{T}_t$ of a certain (s, a) pair will stay the same, no matter how much time has passed since the last visit. However, the longer the time since the last visit, the more the $\hat{T}_t$ would potential be changed upon the next visit of the (s, a) pair. Note that, Perfect Integration, typically used in a standard implementation of Prioritized Sweeping (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013), is a special case of Leaky Integration without prior knowledge, with $\eta = 1$ and $\eta_{\text{bckgrd}} = 1$.

In summary, the three model learning procedures we consider exhibit different features, whose importance we later test by means of simulations. The Particle Filter implements surprise-based adaptation through Equation 3.7, as well as exploratory behaviour through prior knowledge about the environment and background updating in Equation 3.9. The Leaky Integrator with prior knowledge is not surprise-based, but has the capability to explore. Finally, the Leaky Integrator without prior knowledge has neither of the two features. We used these three model learning alternatives in conjunction with Prioritized Sweeping, a tabular model-based RL algorithm, which we describe in the next section.

3.2.3 Using the learned model for Reinforcement Learning

Prioritized Sweeping (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013) is an efficient model-based RL algorithm that instead of updating at each time step the whole state-action space, until the Bellman equations reach an equilibrium (i.e. value iteration), it updates only the state-action pairs that would change "a lot". More specifically, after an observed transition from a state-action pair (s, a) to a state s_{t+1} the estimate $\hat{T}_t$ is updated, and, thus, the $Q_t(s, a)$ value needs to be updated so that $Q_{t+1}(s, a) = \sum_{s'} \hat{T}_{t+1}(s, a, s')\big(\hat{\hat{R}}_{t+1}(s, a, s') + \gamma V_t(s')\big)$. We can measure the change in the value of state s caused by this update as $\Delta V = V_{t+1}(s) - V_t(s) = \max_{a'} Q_{t+1}(s, a') - \max_{a'} Q_t(s, a')$. If we propagate the updated $Q_{t+1}(s, a)$ to other state-action pairs, in order to restore the Bellman equation everywhere, then, intuitively, a large value change of a state, will give rise to a large value change at the states that lead to it. Prioritized Sweeping keeps track of these ΔV changes, called *priorities*, in a list, called *priority queue*, and at every time step, updates, in a number of update cycles, only those states that precede the ones with the largest priorities ΔV. As the number of update cycles goes to infinity Prioritized Sweeping becomes identical to value iteration, where the whole state-action

space is updated at each time step recursively until convergence.

There are two key differences between the three model learners we consider and the standard Perfect Integration typically used in Prioritized Sweeping, which have direct implementational consequences. First, a transition from $(s, a) \rightarrow s_{t+1}$ changes the estimation not only of this transition, but also of all the transitions $(s, a) \rightarrow s'$, where $s' \neq s_{t+1}$, i.e. of the whole transition probability vector from (s, a). This poses a difficulty in using the efficient "small backup" update method of Van Seijen and Sutton (2013). Instead, we use traditional ("full") backups, i.e. $Q_t(s, a) = \sum_{s'} \hat{T}_t(s, a, s')\big(\hat{\bar{R}}_t(s, a, s') + \gamma V(s')\big]$, for all $s' \in \mathcal{S}$. Second, at every step the estimated transition probabilities of the whole state-action space change through background updating (forgetting). Thus, at every step, before going through the priority queue and reverse backups, we perform one round of full backups for all states and actions in the environment, in randomized order. Even though this may appear computationally expensive, it is still linear in the number of states and actions. Efficiency could be improved by applying theses backups, that aim at incorporating the background forgetting, only every few time steps, instead of every step.

Furthermore, we initialize the estimated $\hat{T}_t$ uniformly to $1/|S|$ for all model learners. In the standard implementation of Prioritized Sweeping with Perfect Integration the knowledge of a possibility to transition from any state to any other state is not provided to the agent. The rewarded (goal/terminal) states, once discovered by the agent, are set to uniform outgoing transitions to all other states, and stay fixed thereafter (no updates are performed).

The reward function in our tasks is a function of the landing states only, that is we can write $\bar{R}_t(s, a, s') = \bar{R}_t(s')$. For its estimation we use Perfect Integration for all our simulated agents, since in all our tasks the reward locations and values are stationary, i.e.

$$\begin{aligned} N_{t+1}(s) &= N_t(s) + 1, \\ R^{sum}_{t+1}(s) &= R^{sum}_t(s) + r, \\ \hat{\bar{R}}_{t+1}(s) &= R^{sum}_{t+1}(s)/N_{t+1}(s), \end{aligned} \tag{3.16}$$

where $N_t(s)$ is the number of visits of a state s and r the immediate reward experienced. We use optimistic initialization, i.e. we initialize $\hat{\bar{R}}_t$ to the maximum possible immediate reward value and the Q values as $\hat{\bar{R}}_t/(1-\gamma)$, so that the Bellman equation is fulfilled. Once a state s' is visited, the reward function $\hat{\bar{R}}$ is overwritten by the result of the updates in Equation 3.16. The pseudocode of our implementation is provided in the Supplementary Material (Algorithm 4).

3.3 Simulations

We evaluate the three model learning modules – Particle Filtering with 20 particles ("pf20"), Leaky Integration with prior knowledge ("Leaky prior") and Leaky Integration without prior knowledge ("Leaky") – combined with Prioritized Sweeping on three RL tasks: (i) non-stationary random MDPs, which correspond to the generative model of Equation 3.1 - Equation 3.3; (ii) a task similar to Sutton and Barto (2018) which we denote as "Simple Maze"; and (iii) a task inspired by Sutton et al. (1999) and Bacon et al. (2017) which we call the "Four Rooms Maze" task. These tasks allow us to test different facets of learning and assess the impact of the three alternative model learners. We first describe the procedure we follow in order to optimize the algorithms and evaluate their performance.

Optimization and evaluation procedures. For all agents and all tasks we use a discount factor $\gamma = 0.9$, ϵ-greedy policy with $\epsilon = 0.01$, and 100 priority queue update cycles, and we do not optimise these parameters. The simulation time for each instance of a task is 10^5 steps. Each of the model learners has two free parameters; (p_c, σ) in the Particle Filter, (η, σ) in the Leaky prior and $(\eta, \eta_{\text{bckgrd}})$ in the Leaky Integrator. For the case of non-stationary MDPs, in order to reduce the computing time of the optimization procedure, we set the stochasticity parameter σ of the Particle Filter and the Leaky prior to the true stochasticity of the environment and do not optimize it.

We first tune each learner's free parameters using a number of random seeds: 9 random seeds for the non-stationary random MDPs, and 6 random seeds for each of the Simple Maze and Four Rooms Maze tasks. The seeds control the following sources of randomness: the environmental probabilistic changes, the true transition probabilities drawn randomly after each environmental change, the agent's ϵ-greedy policy, and the sampling procedure in the Particle Filter. For a given learner, for each task instance (i.e. random seed) we perform a gridsearch over the two- (or one-) dimensional parameter space. For each parameter values in the gridsearch, we record the total reward obtained on each random seed, and, then, average the total reward across seeds. The optimal parameter values are the ones that yield the maximum mean reward across the (training) random seeds. However, we have empirically seen that using as a criterion the maximum mean reward across the training seeds was often not the most fair choice. In some cases, the parameters associated to the maximum mean reward also exhibit very high variance (across seeds). For a more fair comparison, we retain for each agent the parameter values of the maximum mean reward, as well as all the parameter values of a mean reward within one standard deviation (over training seeds) of the maximum. We then evaluate all these "winning" parameter values on 10 different (testing) random seeds.

To motivate this approach, let us denote as $R(\theta, k)$ the total reward obtained using the parameter values θ on a random seed k (the seed includes all sources of stochasticity, so that $R(\theta, k)$ is deterministic). For example, for the Particle Filter θ is a tuple of

values for (p_c, σ). We are seeking the optimal parameters $\theta^{opt} = \text{argmax}_{\theta \in \mathbb{R}^d} \mathbb{E}_k[R(\theta, \cdot)]$, i.e. the parameter values that are optimal, averaged over all possible random seeds. We estimate this quantity via the empirical mean, and at the limit of infinite random seeds, the empirical mean should converge to the true expectation. We are however restricted to employ a finite and small number of seeds. We thus have a noisy estimation and we can not be sure whether we are close to the true expectation. Choosing this set of "winning" parameter values increases our estimation precision and we can consider them as a range of values within which the true values should fall. Another motivation for this procedure is that the single objective of maximum mean reward may not be the only desired feature, and the robustness, i.e. variance across seeds, should also be taken into account in the evaluation criteria. Overall, this treatment of retaining a set of best possible parameter values rather than a single value, gives us a better idea about the best possible performance of the algorithms.

3.3.1 Non-stationary Random MDPs

We simulated our algorithms on tabular non-stationary random MDPs that correspond to the generative model we described in Equation 3.1 - Equation 3.3. The environment consists of 100 states, 4 actions and 4 reward locations of reward $r = 1$. In the beginning and after each environmental change, the transition probabilities from each (s, a) pair to all states are randomly drawn from a Dirichlet distribution, i.e. $p_t^{sa} \sim \text{Dir}(\sigma \cdot \mathbf{1})$. Note that, for a single (s, a) pair, this task is similar to the Categorical task of the previous chapter (subsection 2.2.3). The only difference is the occasional absence of observation, since the agent is on different (s, a) pairs at different time steps, and changes can happen even in the absence of observations. The four reward locations are randomly selected and stay fixed throughout the duration of the task. After reaching the reward, the agent is placed on some randomly chosen state among all the available states, excluding the rewarding ones.

We simulated all combinations of stochasticity levels $\sigma \in \{0.01, 0.1\}$ and change probability levels $p_c \in \{10^{-5}, 10^{-4}, 10^{-3}\}$, each for 10^5 steps. Each p_c level translates to $|S| \times |A| \cdot p_c$ state-action pairs changing on average at each time step.

Fig. 3.1 shows the mean reward for all algorithms and winning parameter values across the 10 task instances, and Fig. 3.2 the total reward for the best ones among those of Fig. 3.1. For $p_c = 10^{-5}$ (Fig. 3.1A and B, left panels) all algorithms achieve similar performance levels, and even the simple Leaky Integrator performs well. Since we did not hand-craft the environments, and changes happen randomly, it might be that they rarely lead to a need for a change of the agent's policy. Moreover, the σ levels we use, combined with the large number of states, give rise to fairly stochastic environments (see Supplementary Material Fig. 3.11). It is likely that the agents do not have enough time to encounter enough re-occurrences of states before a change happens and to build strong beliefs (i.e.

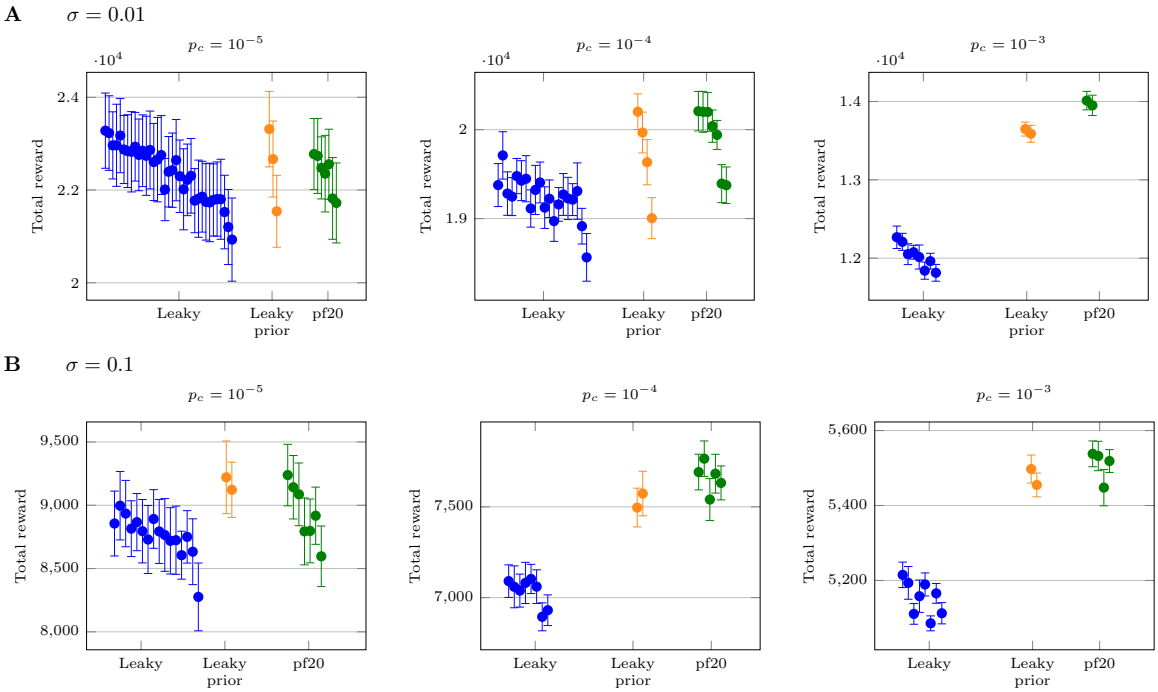


Figure 3.1 – **Non-stationary Random MDPs: Mean total reward.** Mean reward (across the testing random seeds) obtained by each algorithm for **A.** $\sigma = 0.01$, and **B.** $\sigma = 0.1$, for different p_c levels (increasing from left column to right column). Each circle corresponds to a different set of winning parameter values for each algorithm, chosen as described in the "Optimization and evaluation procedures" paragraph. The order of the circles is according to their performance during the optimization (training seeds) in terms of maximum mean reward (from larger to smaller). The error bars indicate the standard error of the mean across the 10 testing seeds. Leaky prior achieves high levels of performance compared to pf20. The only case where the Particle Filter performs significantly better than the Leaky prior is for $p_c = 10^{-3}$ and $\sigma = 0.01$.

move away from the prior), thus, surprise is less effective. For $p_c = 10^{-4}$ (Fig. 3.1A and B, middle panels) the Leaky prior and the Particle Filter perform better than the Leaky, and more so for $\sigma = 0.1$ (Fig. 3.1B, middle panel). A similar observation can be made for the case of $p_c = 10^{-3}$ (Fig. 3.1B, right panel). The only case where the Particle Filter performs significantly better (Wilcoxon rank-sum test, p-value $= 0.05$; unequal variance two-sample t-test, p-value $= 0.02$) than the Leaky prior is for the relatively low, but not extremely low, level of $p_c = 10^{-3}$ and low stochasticity $\sigma = 0.01$ (Fig. 3.2A and B, upper right panels). A closer look at the best possible performance of all algorithms is provided in Fig. 3.2. We empirically found that for even higher p_c and σ values, the three algorithms perform very similarly.

In summary, in the general case of random independent abrupt changes, simple Leaky Integrators, in some cases even without prior knowledge, are sufficient to achieve high levels of performance. The Particle Filter performed significantly better only at a specific

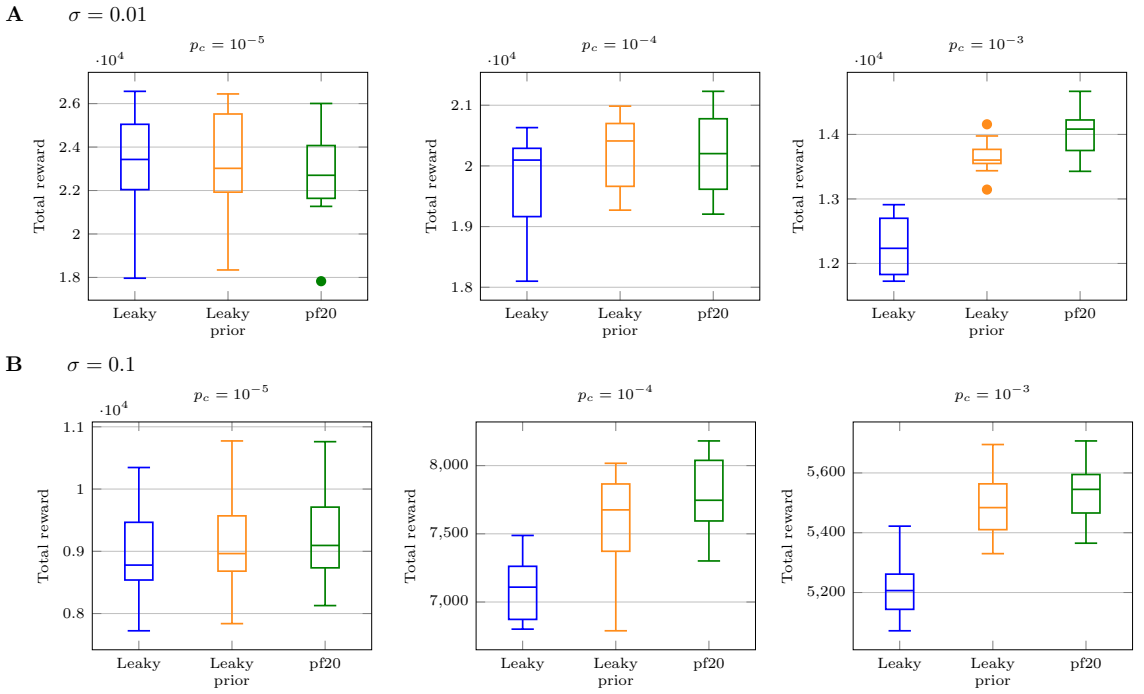


Figure 3.2 – **Non-stationary Random MDPs: Total reward.** Total reward obtained by the best performing parameter set of the two algorithms, across the 10 task instances, for **A.** $\sigma = 0.01$, and **B.** $\sigma = 0.1$, , for different p_c levels (increasing from left to right). The boxplots are made with 10 values, corresponding to the 10 random seeds.

regime of change probability and stochasticity. It is, however, difficult to quantify the extend to which a change in these environments requires a change in the agent's policy. Using rewards of different values spread across the environment or lower stochasticity levels, could potentially bring larger differences among the algorithms. The next two tasks include abrupt environmental changes that directly affect the optimal policy.

3.3.2 Simple Maze

We implemented the tabular task that appears in (Sutton and Barto, 2018), in Chapter 8 ("When the model is wrong"). An illustration of the task can be seen in Fig. 3.3. Each cell in the environment is a state and there are four available actions (up, down, left, and right). Each action leads deterministically to a transition in the corresponding direction, unless it is towards a wall, in which case the agent stays at the same state. There is one starting state (marked in blue) and a single goal (terminal) state with reward value $r = 1$ (marked in red). Once the goal state is reached the agent is placed back to the starting state. Initially there is a path from the starting state to the reward at the right side of the maze (Fig. 3.3, Left). After $N = 5 \cdot 10^4$ time steps, i.e. in the middle of the simulation time, this passage is blocked and a new passage opens on the left side of the

maze. Thus, this task, with its sudden blocking of a learned path that directly affects the optimal policy, assesses the capability of the agent to adapt to a sudden environmental change.

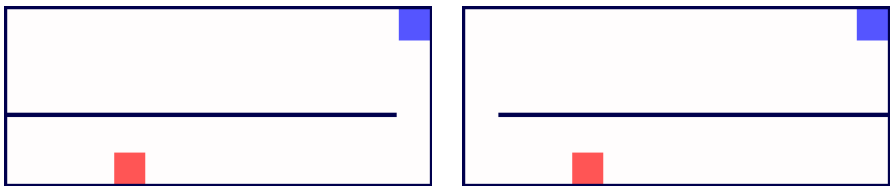


Figure 3.3 – **Simple Maze task.** The path from the starting state (marked in blue) to the goal (marked in red) passes initially through the right side of the environment (Left figure). After N time steps this path is blocked and a door opens at the right side of the environment (Right figure).

We emphasize that this task is not a faithful implementation of the generative model we saw in Equation 3.1 - Equation 3.3 for the following reasons: 1. From each state-action pair there is a possibility to transition to at most 4 states, and for all the other states the probability is 0. In other words, the transition probabilities are not drawn from a Dirichlet distribution with equal stochasticity σ for all states. Humans and animals may have additional priors about the relative topology of states, possibly formed through hippocampal place cells and grid cells, that allow for more informed transition probabilities priors. We do not provide such capability to the agents here. 2. Changes do not occur strictly independently across state-action pairs; opening a door in the maze implies changes in the transition probabilities of two state-action pairs (corresponding to the two states adjacent to the wall). This again relates to a topological dependence between the generative models of different state-action pairs. 3. Changes are not generated probabilistically, but at a predetermined moment (at least from an omniscient observer's point of view).

We evaluated the mean reward across the 10 random seeds, for all algorithms and winning parameter values. The Leaky Integrator exhibited very low performance and most often did not discover the new path to the goal (best case: 3 out of 10 seeds). Overall, the Particle Filtering obtains more reward than the Leaky prior (Fig. 3.4). All algorithms' instances found the new path from the goal, except for one seed for the Particle Filter (first green circle in Fig. 3.4A). For this particular seed, the failure occurred because – by chance – no particle sampled a change-point when experiencing the closed passage. Therefore, the new observation was integrated with the old belief, and beyond this point it became even more unlikely for a switch to happen in the particles' state. Note that this would most likely have been prevented with the use of more particles, and it did not

occur with different parameter values (rest of green circles).

Fig. 3.4B shows the reward obtained by the best performing parameter set for the two algorithms (1st circle of Fig. 3.4A for Leaky prior, and 16th for the Particle Filter). The best parameter set for the Particle Filter is the one that exhibited the minimum variance on the training seeds. The Particle Filter obtains significantly higher reward (Wilcoxon rank-sum test, p-value $< 10^{-4}$).

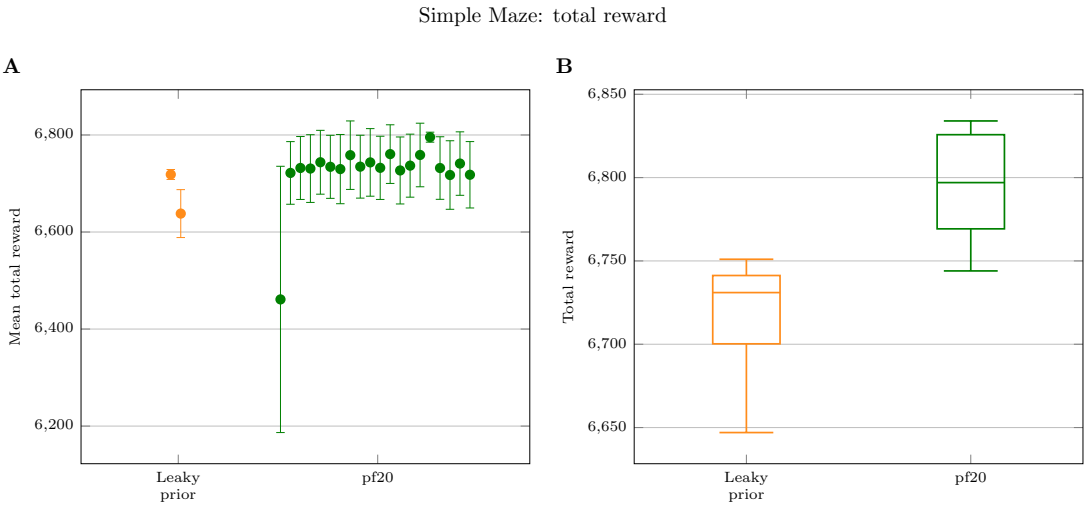


Figure 3.4 – **Simple Maze task: Total reward. A.** Mean reward (across the random seeds) obtained by each algorithm. Each circle corresponds to a different set of winning parameter values for each algorithm, chosen as described in the "Optimization and evaluation procedures" paragraph. The order of the circles is according to their performance during the optimization (training seeds) in terms of maximum mean reward (from larger to smaller). The error bars indicate the standard error of the mean across the 10 testing seeds. For better visualization we show here the performance of only the Leaky prior and the Particle Filter (see Supplementary Material Fig. 3.9 for the figure that includes the Leaky Integrator). **B.** Total reward obtained by the best performing parameter set for the two algorithms (1st circle of **A** for Leaky prior, and 16th for the Particle Filter). The boxplots are made with 10 values, corresponding to the 10 testing random seeds. The Particle Filter obtains more reward than the Leaky prior.

In order to investigate the transient performance of the algorithms, we checked how much time it takes for each agent to reach the goal state after experiencing the change in the environment, i.e. after attempting to move down from the state adjacent to the newly closed passage. Then, we also calculated the time between this 1st visit to the goal following the new path and the 2nd visit. These quantities indicate how fast the algorithms adapt to the change in the environment.

Fig. 3.5A and B show the mean time until 1st visit after experiencing the blocked passage, and the difference between the 2nd and the 1st visit, respectively. When the new path was not found, we set the calculated time to the duration of the simulation after the

change (i.e. $5 \cdot 10^4$). Apart from the one case where the new path was never found, the surprise-based Particle Filter adapts faster than the Leaky prior (Fig. 3.5A and B). Fig. 3.5C and D show the same quantities for the parameter sets that led to the highest reward (1st circle for Leaky prior, and 16th for the Particle Filter). The Particle Filter reaches the reward faster after the change than the Leaky prior (Wilcoxon rank-sum test, p-value = 0.03). For one seed among the 10 seeds we used, Particle Filter exhibits an outlier behaviour, where it reaches the reward much later (1140 time steps) than in the rest seeds (Fig. 3.5C, green outlier), and than the Leaky prior at the same seed (627 time steps). Excluding this seed from both learners increases even more the significance in favour of the Particle Filter (Wilcoxon rank-sum test, p-value = 0.008).

3.3.3 Four Rooms Maze

We implemented a tabular task inspired by Sutton et al. (1999) and Bacon et al. (2017). The environment consists of four rooms as shown in Fig. 3.6. As in the Simple Maze task, each cell in the environment is a state, and there are four available actions. Transitioning towards a wall causes the agent to stay at the same state. There are two fixed reward locations (goals) G_1 and G_2 (marked in red in Fig. 3.6) of values $r_1 = 1$ and $r_2 = 12$, respectively. G_1 is located in the bottom left room and G_2 in the bottom right room. Initially G_2 is surrounded by walls and there is no access to it (Fig. 3.6, Left). One door opens and closes stochastically, with probability $p_c' = 10^{-4}$, and can thus provide access to G_2 (Fig. 3.6, right). Note that p_c' is different from the parameter p_c of the generative model of Equation 3.1 - Equation 3.3, which refers to the change probability of each (s, a) pair. After reaching either of the two goals the agent is moved to the single initial state, at the top and left corner of the environment (marked in blue). We implemented two versions of this task: one with deterministic transitions, and one where the agent moves to its selected direction with a probability $2/3$ and to one of the other three directions with probability $1/9$ each. We indicate the deterministic version with $w_a = 1$, where w_a stands for weight of action, and the second version with $w_a = 2/3$.

Similar to the Simple Maze, this task does also not faithfully correspond to the generative model we described in Equation 3.1 - Equation 3.3. Unlike the Simple Maze, this task features a distal change that leads to a more rewarding option, and thus assesses the capability of the algorithms to explore.

Fig. 3.7A shows the mean reward across the 10 random seeds of the Four Rooms Maze task with $w_a = 1$, for all winning parameter values of all algorithms, and Fig. 3.7B shows the total reward for the best ones among them (1st circle for Leaky, 18th circle for Leaky prior, and 6th for the Particle Filter) in boxplots. The simple Leaky Integrator is not equipped with background forgetting and therefore never reached the reward $r_2 = 12$. In this task the Leaky Prior and the Particle Filter achieve similar levels of performance (Wilcoxon rank-sum test, p-value = 0.7).

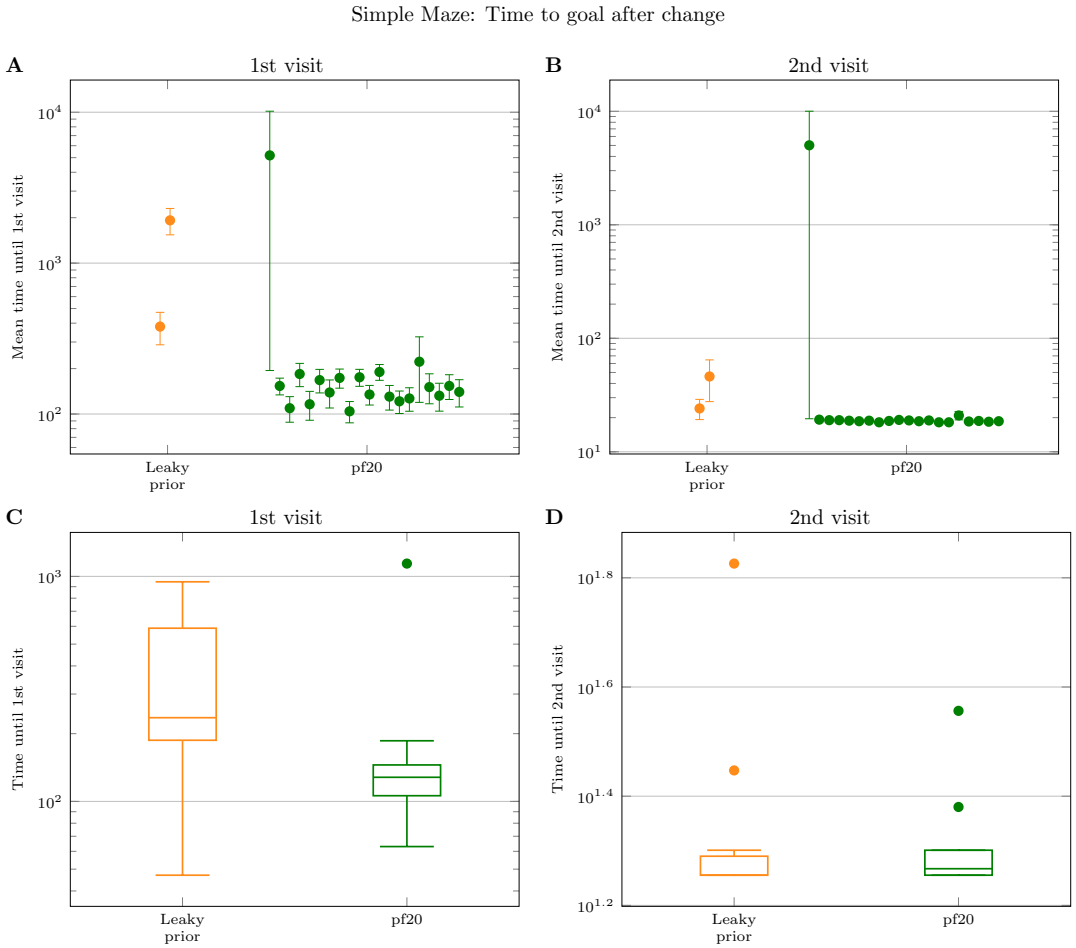


Figure 3.5 – **Simple Maze task: Transient performance. A.** Mean time (across the random seeds) until the 1st visit to the goal after the blockage of the passage, and **B.** Mean time between the 1st and the 2nd visit to the goal. Each circle corresponds to a different set of best parameter values for each algorithm. When the new path was not found, we set the calculated time to the duration of the simulation after the change (i.e. $5 \cdot 10^4$). The error bars indicate the standard error of the mean across the 10 testing seeds. **C.** Total time until the 1st visit and, **D.** between 1st and 2nd visit, for the parameter sets that yielded highest reward (1st circle of **A** for Leaky prior, and 16th for the Particle Filter). The boxplots are made of 10 values, corresponding to the 10 testing random seeds. The minimum possible number of steps from the starting state to the reward, after the passage is blocked is 18 steps. The Particle Filter reaches the goal faster than the Leaky prior after the environmental change.

In order to examine the transient performance of the algorithms, we recorded the time it took to reach the reward $r_2 = 12$ from the moment the door opens, as well as the time lapsed between this 1st and the 2nd visit to $r_2 = 12$. The first quantity reflects the exploratory behaviour of the algorithms, whereas the second one assesses their capability for fast updating. Cases where the door closed before the agent reached the high reward,

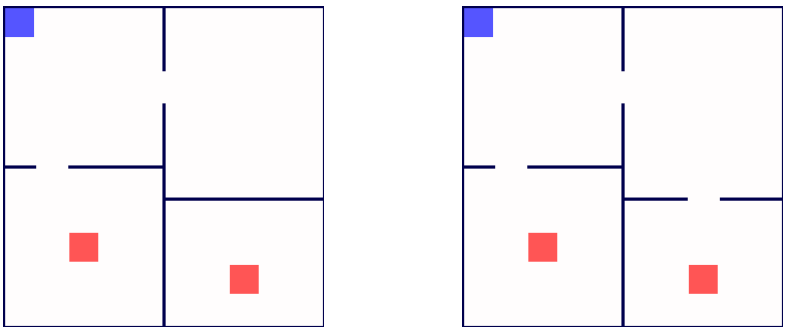


Figure 3.6 – **Four Rooms Maze task.** The starting state is marked in blue and the goal locations (terminal states) in red. The goal location G_1 (bottom left room) is associated with an immediate reward value $r_1 = 1$ and the goal location G_2 (bottom right room) with $r_2 = 12$. Initially there is no access to G_2 (Left figure). The door that gives access to G_2 opens (Right figure) and closes probabilistically.

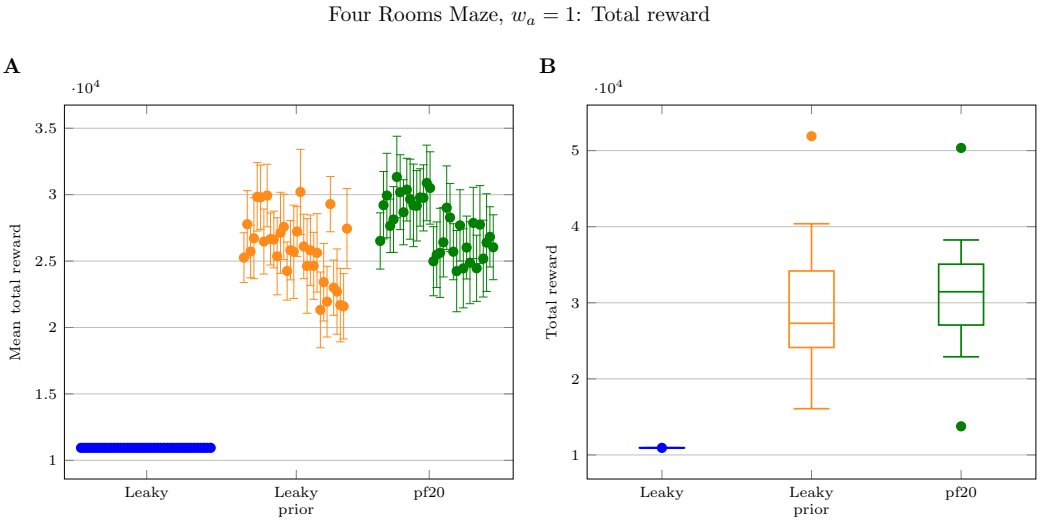


Figure 3.7 – **Four Rooms Maze task, $w_a = 1$: Total reward. A.** Mean reward (across the random seeds) obtained by each algorithm. Each circle corresponds to a different set of winning parameter values for each algorithm, chosen as described in the "Optimization and evaluation procedures" paragraph. The order of the circles is according to their performance during the optimization (training seeds) in terms of maximum mean reward (from larger to smaller). The error bars indicate the standard error of the mean across the 10 testing seeds. **B.** Total reward obtained by the best performing parameter set for the two algorithms (1st circle of **A** for Leaky, 18th circle for Leaky prior, and 6th for the Particle Filter). The boxplots are made with 10 values, corresponding to the 10 testing random seeds. The Leaky Prior and the Particle Filter achieve similar levels of performance.

were excluded from our counts, in order to get a cleaner estimation.

Fig. 3.8A and B show the average of the above two quantities across switches for the

winning parameter values for the Leaky prior and the Particle Filter, for the deterministic environment ($w_a = 1$). The Leaky Integrator did not reach the high reward at any point, we thus exclude it from this analysis. Fig. 3.8C and D depict the boxplots for the best performing (in terms of reward) parameter sets (18th circle for Leaky prior, and 6th for the Particle Filter). From the total 53 times of door openings during the 10 task instances, the Leaky prior reached the high reward state 44 times (83%), and the Particle Filter 45 times (85%) – these are the numbers of data points we used to create the boxplots. Among the times that $r_2 = 12$ was reached, it was always reached a 2nd time, by both algorithms. We observe that the time until 1st visit is not significantly different for Leaky prior and Particle Filter (Wilcoxon rank-sum test, p-value $= 0.9$) indicating that both algorithms are capable to re-explore previously visited states (Fig. 3.8A and C). On the other hand, the 2nd consecutive time the high reward state is reached is significantly faster for the Particle Filtering (Wilcoxon rank-sum test, p-value $= 10^{-4}$) (Fig. 3.8D).

In this task, where exploration and strategic forgetting are presumably more important for good performance, a Leaky Integrator with prior knowledge suffices. The Particle Filter has better transient performance, since it seems to adapt faster at the detected change. However, this ability does not seem to impact much the final performance and is overshadowed by the ability to explore. We obtained similar results for the case of $w_a = 2/3$. In this case of higher stochasticity the two algorithms become even more indistinguishable in their performance (see Supplementary Material Fig. 3.10).

3.4 Discussion

We have developed a model based RL agent that features fast surprise-based updates and exploration. Both of these features stem from a Bayesian treatment of environments exhibiting abrupt changes. We have tested model estimation methods of varying sophistication in a number of environments with different characteristics. Our surprise-modulated RL agent adapts rapidly to sudden immediately experienced changes and achieves high performance. In environments, however, with distal changes or with higher stochasticity, simpler methods of Leaky Integration perform equally well. Furthermore, maintaining a prior on the stochasticity of the environment appears crucial in coping with non-stationarity. Interestingly, in some scenarios, learning an accurate model of the world is not needed for reinforcement learning. In this section we briefly review previous related work, we discuss our findings and possible future directions.

3.4.1 Related Work

Exploration-Exploitation. Upon its first mention, the problem of sudden environmental changes in RL was addressed as part of the exploration-exploitation dilemma (Sutton, 1990; Sutton and Barto, 2018); an agent that is encouraged to explore is more

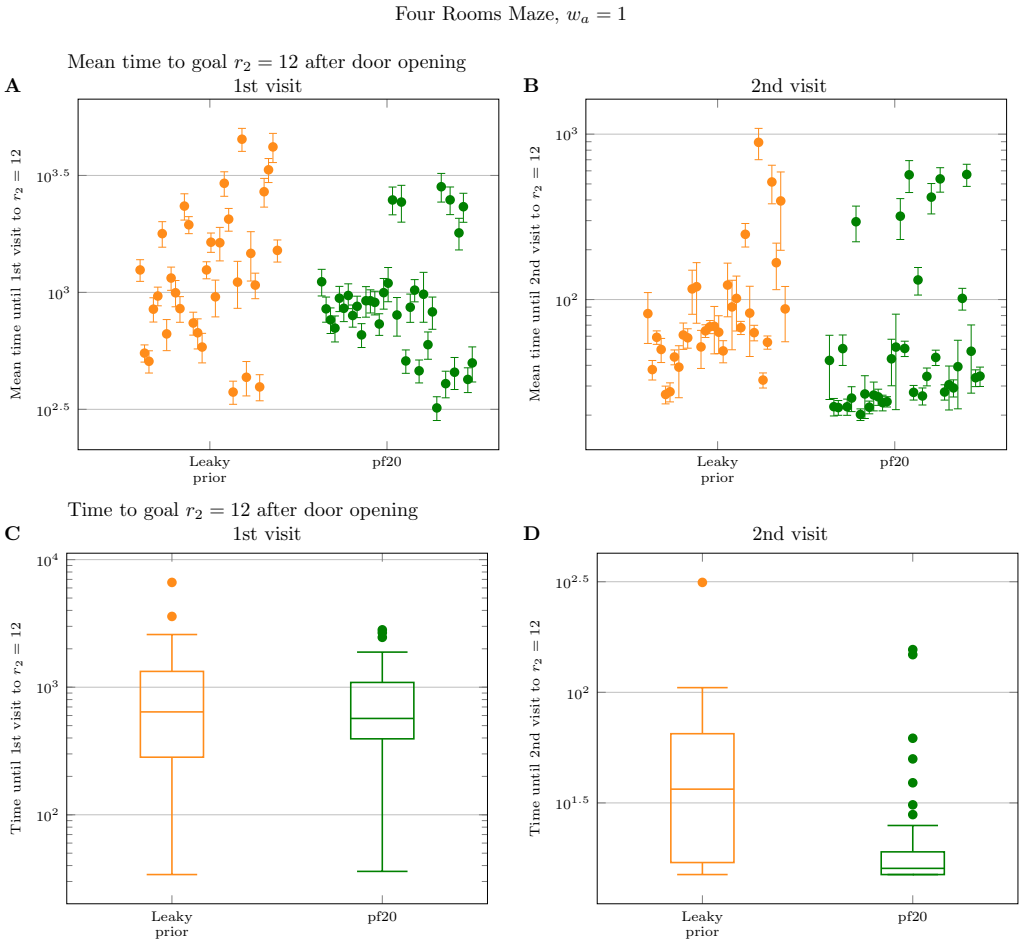


Figure 3.8 – **Four Rooms Maze task, $w_a = 1$: Transient performance. A.** Mean time (across switches) until the 1st visit to the reward $r_2 = 12$ after the door opening, and **B.** Mean time between the 1st and the 2nd visit to the reward $r_2 = 12$. Each circle corresponds to a different set of best parameter values for each algorithm. The error bars indicate the standard error of the mean across switches. We excluded the cases where the door closed before the agent reached the high reward. **C.** - **D.** Boxplots corresponding to the quantities of **A** and **B**, respectively, for the parameter sets that yielded highest reward (18th circle for Leaky prior, and 6th for the Particle Filter). The boxplots are made with 44 values for the Leaky prior and 45 values for the Particle Filter (i.e. number of times the reward $r_2 = 12$ was reached out of the total 53 times the door opened). The minimum possible number of steps from the starting state to the reward $r_2 = 12$ is 15 steps. The two algorithms reach the high reward state equally fast after the door opens. The Particle Filter re-visits it faster for a 2nd time, i.e. after having experienced the change.

likely to cope with a change and discover a new path. A simple illustration of a sudden environmental change leading to suboptimal performance is the blocking maze in Sutton and Barto (2018), similar to the Simple maze task (Fig. 3.3). Depending on the time point

in the agents' lifetime at which this change happens, the agent may get completely stuck. Sutton (1990) added to each state-action pair's value an exploration bonus, according to the time since it was last encountered. This enables re-testing actions that were found unsuccessful in the past and recovers the performance. A more systematic approach to exploration bonuses was employed by Dayan and Sejnowski (1996), starting from the argument that exploration should be driven by the agent's uncertainty about the world. Since their method is the closest to our work, we describe it here in more detail.

Dayan and Sejnowski (1996) considered deterministic mazes, where changes can happen between episodes (i.e. after the agent has reached a terminal state). Whenever there is an environmental change, the effectiveness of an action to advance the agent to some other state is changed, i.e. the path from an (s, a) pair to another state is blocked or opened. There are no rewards, each action costs a certain amount and the goal of the agent is to minimize the expected discounted cost. The agent is equipped with an abstract model of how often the world changes, and knows the change probability p_c and the probability ϕ for an action to be effective, as well as the cost of each action and the locations of the terminal states. The agent estimates the transition probabilities as follows: Within an episode the agent's estimated probability for a transition (or for the efficacy of an action) is reset to whatever happened, i.e. is set to 1 or 0, if the action was effective (successfully led to some other state, instead of the same one) or ineffective, respectively. At the end of an episode, estimated transition probabilities of state-action pairs that were not visited are leaked at a rate given by p_c, and relax in time to the prior ϕ, similarly to the Leaky with prior in our simulations. This forgetting gives naturally rise to an exploration bonus. For the state-action pairs that were visited during the episode, the estimated transition probability for the transitions that were permitted – which during the episode were set to 1 – is now reduced by the amount $p_c \cdot (1 - \phi)$. The estimated transition probability for the transitions that were not permitted is set to the value $p_c \cdot \phi$. These last two operations implement the agent's knowledge that between two episodes a change might occur. This approach begins with solid Bayesian arguments and is intuitive and elegant, but is still a heuristic approximation. Our particle filter improves upon this approach and represents a more principled and generic way of learning in non-stationary environments.

Context Detection. Other approaches tackled non-stationarity using context or "hidden mode" detection. Choi et al. (2000) assume that there is a small number of possible stationary "modes" that the environment can be in. The agent knows the number of modes and model learning is done with a variant of the Baum-Welch algorithm. This requirement of an a priori knowledge on the number of possible environments is removed in (Da Silva et al., 2006) and (Hadoux et al., 2014). There, a quality measure is calculated to estimate which of the so far learned modes is currently the most likely one. If this measure is below some threshold for all possible modes, then a new uniformly initialized mode is appended and the agent starts learning the model from scratch. The possible drawbacks of these approaches are high memory demands, especially for high-dimensional spaces, and possible failure for similar modes. All the above methods can be seen as

"reactive" approaches to non-stationarity (Papoudakis et al., 2019), in the sense that they develop agents that quickly modify their behaviour after a change has been detected. Our approach also falls in this category.

Meta- and Continual Learning. Recently there has been a renewed interest in non-stationary learning in the form of *meta-learning* and *continual learning*, and particularly in the domain of deep RL. In contrast to the reactive view, the focus of *meta-learning* is on building agents that are prepared for changes, rather than agents that learn how to react to them (Papoudakis et al., 2019). These agents perform better in changed situations, because they were exposed beforehand to various perturbations and task variants. A key assumption is that all situations come from the same distribution and the aim is to find the (initial) parameters so that the algorithm has learned how to learn, namely knows how to generalize and adjust with minimal or no updates, and is more efficient than starting from scratch. *Continual learning* is a notion closely linked to meta-learning and often interchangeably used. Sometimes continual learning is defined as the ability to adapt and generalize in the light of new observations, while resorting as little as possible to storage and re-processing of past observations (Lomonaco et al., 2019). Elsewhere, continual learning is formulated as the ability to learn new tasks without forgetting previous ones. This can be a different aim as it may imply maintaining some memory of previous tasks and being able to recognize the current task, in order to select which policy to use (Traoré et al., 2019). In that sense, continual learning is similar to multi-task learning, but with tasks being experienced sequentially. The main approaches to continual learning are: rehearsal, regularization, dynamic network architecture and generative replay (see Traoré et al. (2019) for more details). At the same time, some continual learning approaches do essentially employ some change-detection procedure and "reactive" handling, for example, comparing the average obtained reward within a short-term time window to a long-term average reward in order to control weight regularization (Lomonaco et al., 2019), or computing the likelihood of the next few data points under a predictive model with parameters that were updated based on the past few data points (Nagabandi et al., 2018).

Model-free methods. An approach to non-stationarity on the model-free side has been developed by Kearney et al. (2018) and Young et al. (2018), where the learning rate is adapted proportionally to the correlation between the current weight update and a memory trace of past updates, analogous to momentum. Other model-free meta-RL approaches with function approximation have been done by Duan et al. (2016); Finn et al. (2017); Wang et al. (2018). One possible disadvantage of these approaches, is that they are often sample inefficient, even more so than standard model-free RL algorithms.

Distributional RL. In our work we perform approximate Bayesian inference via particle filtering, but the model-based RL agent eventually uses the expected value of the transition probabilities over its belief and not the full distribution. One can think of extensions of our work where the full distribution is taken into account, and thus the Q values are also distributions, instead of scalar values. Examples of such approaches can

be seen in Bellemare et al. (2017); Dearden et al. (2013); Osband et al. (2013), but have not been applied to non-stationary environments.

3.4.2 When do we need change-point detection for non-stationary RL?

In our simulations we found that a simple Leaky Integrator with prior knowledge can achieve high levels of performance in seemingly complex environments. An approach of this type, where an agent is equipped with a model of how transitions can change over time, was previously developed in (Dayan and Sejnowski, 1996) and was found to be successful in non-stationary mazes. Our results are also consistent with the work of (Ryali and Yu, 2016), who showed that in non-stationary categorical tasks an appropriately adjusted Leaky Integrator with constant leak parameter can reach near-Bayes-optimal performance in data prediction. Other studies have also shown that, after an initial phase, the updates of a Leaky Integrator form a delta-rule that can approximate exact Bayesian updates (Heilbron and Meyniel, 2019; Yu and Cohen, 2009), as well as that such an updating scheme is consistent with human behaviour (Gijsen et al., 2020; Heilbron and Meyniel, 2019; Meyniel et al., 2016). Along similar lines, Findling et al. (2019) showed that an inferential procedure on a simpler generative model than the true one, that assumes stationarity and is equipped with noise, can lead to near-optimal adaptation in volatile environments, and matches observed human data in adaptive bandit tasks.

The aforementioned findings combined with our results indicate that elaborate and computationally more expensive change-point detection methods may often not be needed by an artificial or a biological agent to achieve high performance. "Bounded rationality" (Simon, 1957) is a widely used term to express the idea that biological organisms have constrained resources, which prevent them from being optimal in the often intractable real-world problems. These results, however, add another facet to this notion; that organisms often do not *need* to invest resources in order to behave near-optimally. This line of thought also echoes ideas from the science of heuristics; heuristic inference can often be equally or more accurate than complex formal methods, given the agent's uncertainty (Belousov et al., 2016; Gigerenzer and Gaissmaier, 2011).

3.4.3 Future directions

All the above findings, as well as ours, refer to tasks where the observed data are of categorical nature, which is the case for a wide range of real-world situations. Complex and accurate change-point detection are likely to be more beneficial in continuous or tracking tasks, and this would be interesting to test in future work. Along the same lines, the evaluation of these algorithms in more complicated tasks, such as mazes with multiple doors changing their status in individually different degrees of volatility, would be very informative.

Additionally, one parameter that we did not investigate systematically is the number of update cycles of the Prioritized Sweeping. This parameter dictates how far back the agent backpropagates its estimations after a new information. We empirically saw that in some cases, counterintuitively, increasing the number of update cycles does not necessarily lead to better performance. The reason is that when the model estimation is suboptimal, deeper backpropagation of the information throughout the value landscape can be even harmful. This adds another aspect to take into account when assessing the necessity of accurate model estimation, and it would be very interesting to be quantified systematically.

For all our algorithms we first fine-tuned their parameters with respect to each task. It would be insightful to investigate the robustness of the algorithms under a mismatch between the assumed and the true environmental parameters, similar to what we did in Chapter 2. This procedure, on the other hand, could entail some arbitrariness in the range and the resolution of the parameter values considered.

An exciting continuation of our work would be the online estimation of the environment's hyper-parameters, with the ultimate goal of building a "generic" learner. We briefly mentioned in Chapter 2 some methods that could handle this challenging problem. We hypothesize that in tasks with distal changes, like the Four Rooms maze, a prior on how the world functions or a smart initialisation of the parameters might still be needed for a good estimate of the environment's volatility.

Finally, it would be interesting to combine these model learning approaches with function approximation (deep RL) methods, and in particular with deep RL agents with tabular abstraction (Corneil et al., 2018; Sutton et al., 1999) that allow the use of efficient tabular model-based methods. Overall, our results may guide building efficient adaptive RL agents and can provide insights in the type of estimations that biological organisms may adopt.

3.5 Supplementary Material

3.5.1 Supplementary figures

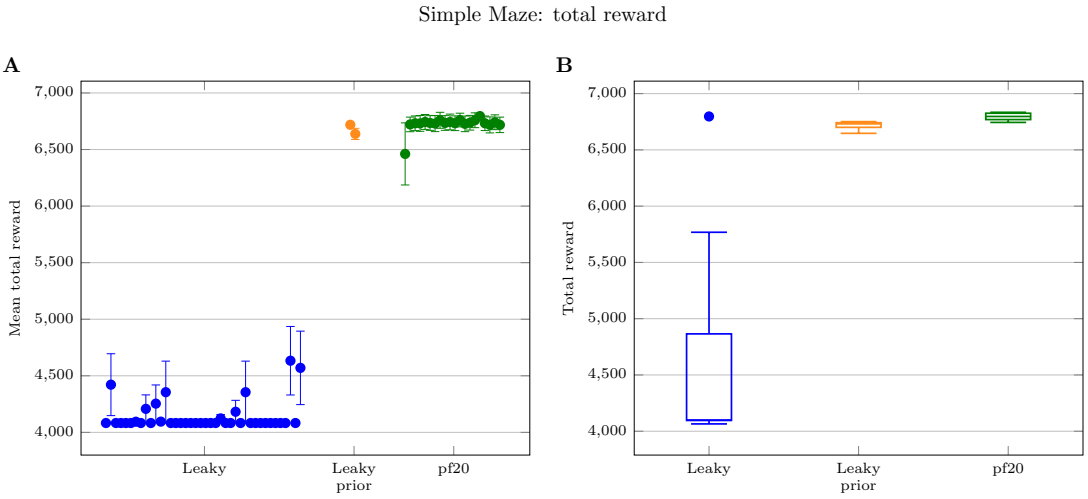


Figure 3.9 – **Simple Maze task: Total reward - Supplementary figure. A.** Mean reward (across the random seeds) obtained by each algorithm. Each circle corresponds to a different set of winning parameter values for each algorithm, chosen as described in the "Optimization and evaluation procedures" paragraph. The order of the circles is according to their performance during the optimization (training seeds) in terms of maximum mean reward (from larger to smaller). The error bars indicate the standard error of the mean across the 10 testing seeds. The Leaky Integrator exhibited very low performance and most often did not discover the new path to the goal (best case: 3 out of 10 seeds). **B.** Total reward obtained by the best performing parameter set for the two algorithms (38th circle of **A** for Leaky, 1st circle for Leaky prior, and 16th for the Particle Filter). The boxplots are made with 10 values, corresponding to the 10 testing random seeds.

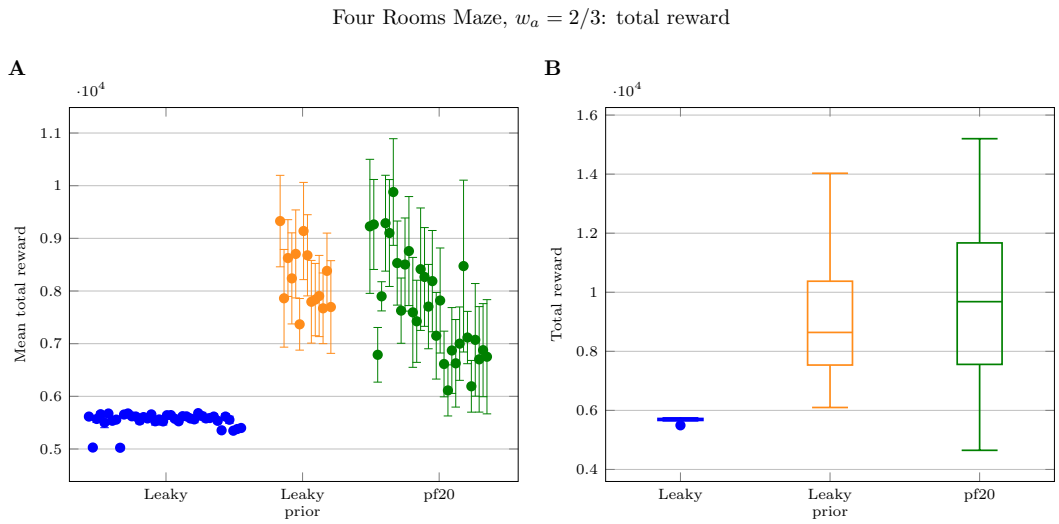


Figure 3.10 – **Four Rooms Maze task, $w_a = 2/3$: Total reward - Supplementary figure. A.** Mean reward (across the random seeds) obtained by each algorithm. Each circle corresponds to a different set of winning parameter values for each algorithm, chosen as described in the "Optimization and evaluation procedures" paragraph. The order of the circles is according to their performance during the optimization (training seeds) in terms of maximum mean reward (from larger to smaller). The error bars indicate the standard error of the mean across the 10 testing seeds. **B.** Total reward obtained by the best performing parameter set for the two algorithms (29th circle of **A** for Leaky, 1st circle for Leaky prior, and 7th for the Particle Filter). The boxplots are made with 10 values, corresponding to the 10 testing random seeds. The Leaky Prior and the Particle Filter achieve similar levels of performance.

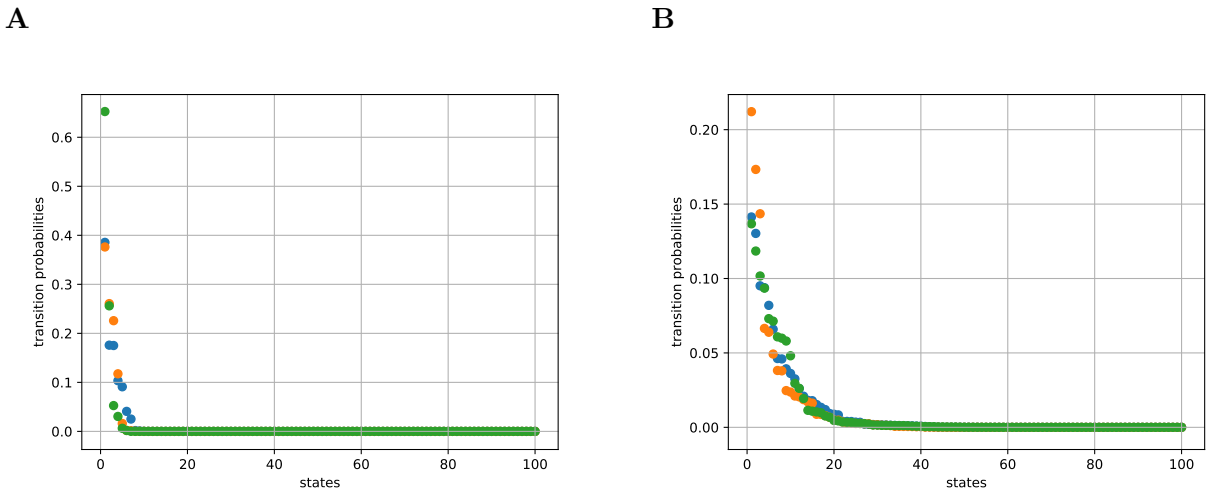


Figure 3.11 – **Sampled transition probability vectors.** Examples of transition probability vectors of 100 elements (states) drawn from a Dirichlet distribution with stochasticity parameter **A.** σ=0.01 and **B.** σ= 0.1, for the non-stationary random MDPs. Each figure shows 3 random draws of $p_t^{sa} \sim \text{Dir}(\sigma \cdot \mathbf{1})$, marked with 3 different colors.

3.5.2 Weight update for the background (s, a) pairs in Particle Filtering

The weight of a particle i for a background pair $(s'', a'') \neq (s_t, a_t)$ at time $t+1$, given a proposal function Ψ and the absence of an observation coming from this (s'', a'') pair at $t+1$, can be calculated as

$$w_{t+1}^{sa,(i)} \propto \frac{\mathbf{P}(c_{1:t+1}^{sa,(i)}|s_{1:t})}{\Psi(c_{1:t+1}^{sa,(i)}|s_{1:t})} = \frac{\mathbf{P}(c_{t+1}^{sa,(i)}|c_{1:t}^{sa,(i)}, s_{1:t})\mathbf{P}(c_{1:t}^{sa,(i)}|s_{1:t})}{\Psi(c_{t+1}^{sa,(i)}|c_{1:t}^{sa,(i)}, s_{1:t})\Psi(c_{1:t}^{sa,(i)}|s_{1:t})}. \tag{3.17}$$

Note that $w_t^{sa,(i)} \propto \frac{\mathbf{P}(c_{1:t}^{sa,(i)}|s_{1:t})}{\Psi(c_{1:t}^{sa,(i)}|s_{1:t})}$ are the weights calculated at the previous time step. Moreover, $\Psi(c_{t+1}^{sa,(i)}|c_{1:t}^{sa,(i)}, s_{1:t+1}) = \mathbf{P}(c_{t+1}^{sa,(i)}|c_{1:t}^{sa,(i)}, s_{1:t+1})$.

Hence, we have that $w_{t+1}^{sa,(i)} = w_t^{sa,(i)}$.

3.5.3 Prioritized Sweeping algorithm

Algorithm 4 Pseudocode for Prioritized Sweeping

Specify the discount rate γ, Ncycles number of update cycles, $|\mathcal{S}|$ number of states, $|\mathcal{A}|$ number of actions, p_{min} minimum priority value.
Initialize $\hat{T}_0(s,a,s') = 1/|\mathcal{S}|$, $\hat{\bar{R}}_0(s) = r_{\max}$, $Q_0(s,a) = V_0(s) = U_0(s) = \hat{\bar{R}}_0/(1-\gamma)$, for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $s' \in \mathcal{S}$.
Initialize empty priority queue PQ.
Observe the state s_0 and select the action a_0.
while the sequence is not finished **do**
    Observe the s_{t+1} and the reward r_{t+1}
    # Update the estimated transition probabilities $\hat{T}$
    Compute $\hat{T}_{t+1}(s,a,s')$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $s' \in \mathcal{S}$ according to one of the model learners (Equation 3.10, Equation 4.15, or Equation 3.15).
    # Update the estimated reward function $\hat{\bar{R}}$
    Compute $\hat{\bar{R}}_{t+1}(s_{t+1})$ according to Equation 3.16.
    # Perform one back-up to incorporate the background updates of $\hat{T}$ into Q
    for $s \in |\mathcal{S}|$ in randomized order **do**
        for $a \in |\mathcal{A}|$ in randomized order **do**
            $Q_{t+1}(s,a) = \sum_{s'} \hat{T}_{t+1}(s,a,s')\big(\hat{\bar{R}}_{t+1}(s,a,s') + \gamma V(s')\big]$, for all $s' \in \mathcal{S}$.
        $V_{t+1}(s) = \max_{a'} Q_{t+1}(s,a')$.
    # Calculate priority
        $p = |V(s) - U(s)|$
        if $p > p_{min}$ **then**
            Add or update the state s in PQ with priority p.
    # Process priority queue
    for Ncycles **do**
        Remove the state s'' from PQ that has the largest priority p.
        Compute $\Delta V = V(s'') - U(s'')$
        Set $U(s'') = V(s'')$
        for $s \in |\mathcal{S}|$ **do**
            for $a \in |\mathcal{A}|$ **do**
                Update $Q_{t+1}(s,a) \leftarrow Q_{t+1}(s,a) + \gamma \hat{T}_{t+1}(s,a,s'')\Delta V$
            $V_{t+1}(s) = \max_{a'} Q_{t+1}(s,a')$.
    # Calculate priority
            $p = |V(s) - U(s)|$
            if $p > p_{min}$ **then**
                Add or update the state s in PQ with priority p.
    # Iterate
    $t \leftarrow t + 1$

3.6 Contributions

VL and JB conceived and designed the project.
VL developed the algorithms, with the help and feedback of AM and JB.
VL wrote the code for the algorithms and the simulations, with the help and feedback of AM and JB, and using as a starting point a julia package [1] written by JB.
VL analysed the results and made the figures, with the feedback of AM and JB.
VL, AM, JB and WG interpreted the results.
VL, JB and WG wrote the manuscript.

[1] https://github.com/JuliaReinforcementLearning/ReinforcementLearning.jl

4 Dissociating human brain regions encoding reward prediction error and surprise

This chapter presents research performed in collaboration with Dr. Marco Lehmann, Alireza Modirshanechi, Dr. Johanni Brea, Prof. Michael Herzog, Prof. Wulfram Gerstner, and Prof. Kerstin Preuschoff.

4.1 Introduction

When learning to ride a bike, many falls and adjustments are needed until we experience some encouraging sign of maintaining our balance for a few meters. Through this experience, we then, slowly, and maybe subconsciously, learn which positions and movements are successful and which make us lose balance. At the same time, while navigating in a part of the town we have never been, we are often able to quickly form a mental map of the place, so that when we finally find this new recommended restaurant, we can easily visit it again in the future.

In the field of reinforcement learning (RL) (Sutton and Barto, 1998), these two types of learning have been mathematically formalised as model-free (MF) and model-based (MB) learning, respectively (Daw et al., 2005). In model-free RL, artificial or biological agents incrementally update their values or their policies via trial-and-error interaction with the world. In value-based model-free algorithms (Sutton and Barto, 1998), the values, i.e. the "goodness" of being in certain states and taking certain actions, are learned via the reward prediction error (RPE), namely the discrepancy between the value that was expected and the one that is perceived given the new actual experience. The estimated values are then used to guide the agent's policy. Policy gradient model-free algorithms (Peters, 2010; Schulman et al., 2015; Sutton and Barto, 1998; Williams, 1992) feature reward triggered changes directly on the agent's policy, i.e. the mapping from states to favorable actions, typically using eligibility traces.

Model-free learning is a simple way to reinforce repetition of rewarded actions and has successfully explained many aspects of reward-based learning in animals and humans (Matsumoto et al., 2007; Niv and Schoenbaum, 2008; O'Doherty et al., 2004; Roesch et al., 2007; Schultz et al., 1997; Tobler et al., 2003). However, it is slow and inflexible and fails to explain other aspects of animal behaviour, such as quick adaptation to changes or learning of associations in the absence of direct reward (Balleine and Dickinson, 1998; Daw et al., 2005; Dayan, 2012; Doll et al., 2012; Foster and Wilson, 2006; Pfeiffer and Foster, 2013; Tanaka et al., 2015). Model-based RL algorithms, on the other hand, exhibit these capacities. In model-based RL, agents learn a model of the environment, i.e. how states are connected, and can flexibly update values through mental simulation, at the expense of higher computational costs. Learning the model of the world is mediated by a state prediction error (SPE) or by surprise, which express the discrepancy between the expected and the experienced state in the world.

In the field of neuroscience, it soon became apparent that this binary segregation between model-free and model-based may be oversimplifying and there may not be a clear separating line (Collins and Cockburn, 2020; Daw, 2015, 2018; Langdon et al., 2018); humans and animals exhibit behaviours consistent with both or a mixture of the two types of learning in different circumstances (Daw et al., 2005), and in the brain, the neural substrates of the two strategies are often shared (Doll et al., 2012; Gremel and Costa, 2013; Langdon et al., 2018; Tanaka et al., 2015). For example, dopaminergic neurons have been traditionally thought to convey a model-free RPE, but recent results have shown that they are also sensitive to sensory prediction errors (Howard and Kahnt, 2018; Takahashi et al., 2017). Thus, despite the wealth of studies on human learning (Anggraini et al., 2018; Cushman and Morris, 2015; Daw et al., 2011a; Deserno et al., 2015; Dezfouli et al., 2014; Doll et al., 2015a,b; Economides et al., 2015; Fermin et al., 2016; Gershman et al., 2014a; Gläscher et al., 2010; Huys et al., 2012; Kroemer et al., 2019; Lee et al., 2014; Otto et al., 2013a,b; Simon and Daw, 2011; Wimmer and Shohamy, 2012; Wunderlich et al., 2012a,b) it is still an open question which strategies best describe human behavior in what type of situations, as well as how strategies are implemented and combined in the brain (Daw, 2018; Huang et al., 2020).

More specifically, on the experimental side, research trying to address the above questions has mostly employed the now classic two-stage task (Daw et al., 2011a; Gläscher et al., 2010) or variations thereof (Cushman and Morris, 2015; Deserno et al., 2015; Dezfouli et al., 2014; Doll et al., 2015a,b; Economides et al., 2015; Kroemer et al., 2019; Otto et al., 2013a,b; Wunderlich et al., 2012b), where the temporal credit assignment problem is less pronounced and the computational cost of MB learning becomes minor. Scaling up the task complexity in brain imaging experiments causes challenges for dissociating different entangled learning signals (Daw, 2018; Fouragnan et al., 2018; Pernet, 2014). Moreover, results seem to largely depend on task details, or even task instructions (da Silva and Hare, 2020; Tanaka et al., 2015). On the theory side, the current RL account seems to be missing various aspects of human learning both in terms of efficiency (Gershman and Daw,

2017; Lake et al., 2017; Lengyel and Dayan, 2007), and of suboptimality (da Silva and Hare, 2020; Findling et al., 2019; Mathys et al., 2011; Prat-Carrabin et al., 2020). The space of possible algorithms describing human learning is yet to be determined (da Silva and Hare, 2020; Daw, 2018) and the number of possible competing models considered in studies is usually quite limited. In particular, policy gradient methods have received relatively less attention in human experimental studies (Coddington and Dudman, 2019; Li and Daw, 2011; O'Doherty et al., 2004).

In this work, we aim at identifying which classes of algorithms – model-free value-based, model-free policy gradient, and model-based value-based – or combinations thereof describe human learning behaviour and brain activity in a multi-step task with larger state-space. We started from the broadly accepted standpoint that both model-free and model-based computations are implemented in the human brain (Daw et al., 2011a; Gläscher et al., 2010; Lee et al., 2014). We designed a novel multi-step decision making task with an experimental manipulation that disentangles different learning signals at the level of BOLD brain responses. A Bayesian treatment of our task gives rise to an outlier detection algorithm featuring a surprise-modulated update. Contrary to the algorithms we saw in the previous chapters, this algorithm entails a trade-off between integrating a new observation and ignoring it, where the role of surprise is to attenuate learning instead of accelerating it. We use this surprise modulation in a model-based algorithm and in novel hybrid algorithms, where model-based surprise influences the model-free learning rate. We report evidence for an actor-critic framework with possible model-based influences as a likely model for behaviour, and we find neural signatures for both model-free and model-based prediction errors. Our results extend previous fMRI findings to a multi-step scenario and support the existence of multiple parallel learning systems in the brain.

4.2 Results

4.2.1 Experimental design to separate RPE from SPE

Twenty-three participants were recruited to perform our multi-step decision making task, in a state-space with 7 circularly arranged fractal images (states) and 2 possible actions at each state (apart from the goal which is a terminal state) (Fig. 4.1A). At the beginning of each episode participants are shown an initial state, randomly chosen among two possible initial states. As soon as a participant chooses an action, a different image is shown. Participants continue to choose actions until they reach the goal, which completes an episode (Fig. 4.1B). Their task is to reach the goal in the smallest number of actions. The image of the goal state is visually distinguishable from all other states and known to participants beforehand. State transitions are deterministic, with occasional "surprise trials", explained further below. In what follows, we will use the terms "state transition" and "trial" interchangeably.

During the experiment we run in the background the model-free SARSA-λ algorithm (Sutton and Barto, 1998) and the model-based Forward Learner (Gläscher et al., 2010) with the participant's choices as inputs. This gives us an online estimate of the reward prediction error (RPE) and the state prediction error (SPE) used in the two algorithms, respectively. The SARSA-λ uses the RPE to approximate the $Q(s, a)$ values, i.e. the expected sum of future discounted rewards starting from state s and action a. The Forward Learner updates the transition probabilities via the SPE and uses them to directly compute the $Q(s, a)$ values via the Bellman equation. More details on the algorithms are provided in the Methods (subsection 4.4.4) and in Chapter 1. For the online RPE and SPE computations, the choice of the parameters (e.g. learning rates α) was based on pilot experiments performed prior to this study. On a "surprise trial" participants transit to a state s'' other than the one they have learned to expect as the outcome of action a from state s (Fig. 4.1C). We have two types of surprise trials: (i) purely random transitions, (ii) transitions that meet a threshold criterion on V values, explained in the following. If a participant expects to transit from s to s', the other state s'' is chosen such that s' and s'' have similar V values, i.e. $|V(s') - V(s'')| \leq \Delta V$ where $V(s) = \max_a Q(s, a)$ and ΔV is a small threshold. This manipulation does not affect the MF system, since the experienced RPE stays the same. In learned transitions, in particular, the RPE will take low values. At the same time, the experienced MB SPE will be high, since the learned transition has been violated (Fig. 4.1C). As planned, this novel experimental manipulation with online monitoring enables us to decorrelate the RPE from the SPE (see Methods and Fig. 4.4 for more details).

4.2.2 Behavioral results

Participants were able to learn the task during a block of 20 minutes and reached the goal in 3.5 actions on average (minimum possible is 2) already at the 4th episode (Fig. 4.1D). We introduced surprise trials from the 5th episode onwards, which results in an increased average number of steps that participants took to reach the goal (Fig. 4.1D). After this point, the episode length gradually decreases again, indicating that participants were able to learn how to act, even in the presence of surprise trials.

Fig. 4.1E depicts the percentage of correct actions in time averaged across all participants, for 4 representative states. The task structure allows different paths to the goal state and therefore not every participant visits the same states in each episode. In order to make learning comparable across participants, the horizontal axis of Fig. 4.1E does not index the episode number, but the n-th visit of that state. We define a "correct action" as the one that brings the participant closer (in terms of number of actions) to the goal state. Starting from a given state the two actions lead to new states at different distances from the goal. For example, from the state $s5$ of Fig. 4.1A one action (the correct one) leads to the goal (0 actions from goal), whereas the other action leads to a state located 2 actions away from the goal. We therefore denote state $s5$ as "0-or-2". States that have

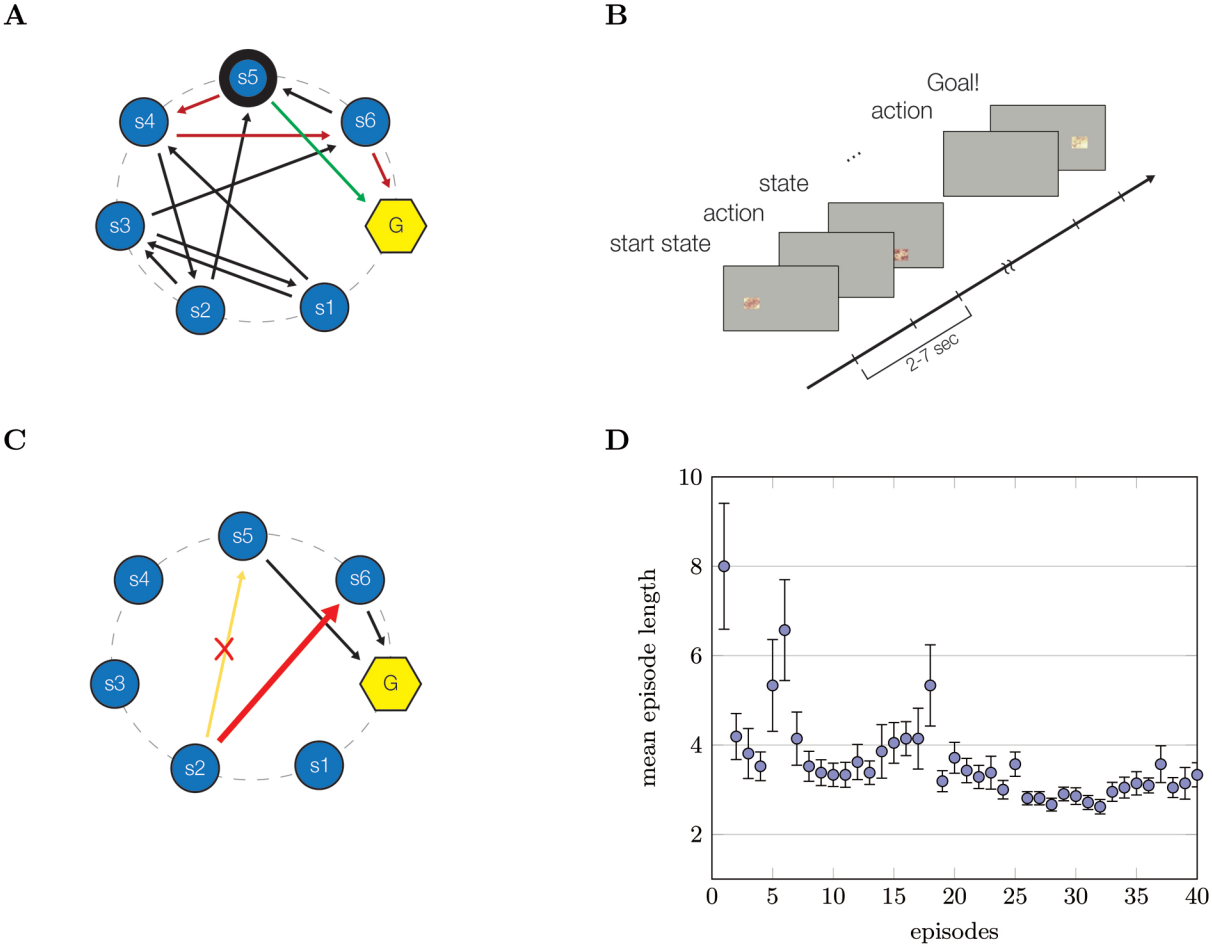

A
s5
s4
s6
s3
G
s2
s1
B
Goal!
action
state
action
start state
2-7 sec
C
s5
s4
s6
s3
G
s2
s1
D
mean episode length
episodes
E
0-or-3
1-or-2
0-or-1
2-or-2
% correct
% choice
visits
B
A

Figure 4.1 *(previous page)* – **Multi-step learning task de-correlates learning signals.** **A**. The hidden graph of the task. There are seven states and two possible actions at each state. Black arrows mark the possible deterministic transitions between states. The goal state G is highlighted in yellow. For most states there is a "correct" action that brings participants closer to the goal and a "wrong" one that brings them to a state away from the goal. For the exemplar state $s5$ the correct action leads directly to the goal (green arrow - 0 actions from G), whereas the wrong action leads to state $s6$ which is 2 actions away from the goal (red arrows - 2 actions from G). We denote this state as "0-or-2". **B**. Schematic of the timeline of a full episode. Participants view one state (specific fractal at a specific location) on the screen, choose one action that moves them to the next state, and continue until they eventually reach the goal state. **C**. Example of a surprise trial. The expected transition for the action chosen by the participant is $s2 \rightarrow s5$, but the next image (and corresponding location) is the one of $s6$, which has a model-free V value approximately equal to the one of $s5$. This results in a high State Prediction Error (SPE) and a low Reward Prediction Error (RPE). **D**. Mean length of each episode. The circles represent the number of actions per episode (from start to goal state) averaged across participants. The error bars mark the standard error of the mean. Already at the 4th episode, participants reach the goal within 3.5 state visits on average (minimum possible is 2). From the 5th episode onwards, we introduce surprise trials and the average episode length increases at this point. Participants learn nevertheless how to act despite the presence of surprise trials, indicated by the decrease in episode length thereafter. **E**. Percentage of selecting the "correct" action at states whose distance from goal is "0-or-3", "0-or-1", "1-or-2", and "2-or-2" actions, respectively, as a function of the number n of state visits. The vertical position of a green circle indicates the fraction of participants that selected the "correct" action, while the circle size represents the number of participants that visited this state n times. Only a few participants (small circles) have visited a state more than 20 times. The average learning curve (red line) is obtained by fitting a weighted exponential to the dots. The vertical dashed blue line indicates the time when the red learning curve reaches the 80% performance level. These graphs provide qualitative evidence that participants learn to choose the "correct" action faster for states that are closer to the goal and for which the "wrong" action has more negative consequences.

equal minimum distance from goal may differ in the resulting distance from goal when the "wrong" action is chosen, or, in other words, in their "difference in correctness" between the two available actions. For example, for two states "0-or-3" and "0-or-1" choosing the correct action brings participants to the goal, but choosing the wrong one is more detrimental for the "0-or-3" state (see Supplementary Material Table 4.2 for distance to goal and action "correctness" across states). We find that participants' speed of learning at each state is, qualitatively, related to the distance from the goal and to the "correctness difference" of the available actions (Fig. 4.1E).

More specifically, participants reach higher performance levels much earlier for the state that is "0-or-3" actions away from the goal, compared to the "0-or-1" and "1-or-2" states (Fig. 4.1E, upper left, bottom left and upper right panels, respectively). In the state "0-or-3" (Fig. 4.1E, upper left) the 80% performance level (vertical dashed blue line) is reached after only 5 visits, whereas for the state "0-or-1" (Fig. 4.1E, bottom left) after approximately 14 visits. This is likely due to the fact that, even though the correct action is still only one step away from the goal, the "wrong" action has less negative effects. At the state with distance index "1-or-2" (Fig. 4.1E, upper right) learning to a performance level of 80% takes 27 visits. On the other hand, for the state of Fig. 4.1E bottom right, where any action brings the participant to states 2 actions away from the goal we do not observe a clear preference between the two. Collectively, this tendency to choose the correct action when in closer proximity to the goal, can be interpreted as a sign of reward information backpropagating to actions that led to it.

4.2.3 Model-free algorithms explain behaviour best

We consider several possible strategies that a participant may follow to accomplish the task: purely model-free, purely model-based, model-free with surprise modulation, and a hybrid combination of model-free and model-based (Fig. 4.2).

A MF strategy with RPE-mediated updates does not take into account the information about the existence of surprise trials, and caches the values of surprising transitions together with non-surprising ones. By construction of the task, this will most of the time not affect substantially the value estimation, since the landing state of a surprise trial is chosen so that it has a value similar to the expected state (apart from the purely random transitions). In the family of purely MF algorithms we considered the TD algorithm SARSA-λ (Sutton and Barto, 1998), the policy gradient REINFORCE (Williams, 1992), and the Actor-critic (Sutton and Barto, 1998). SARSA-λ estimates the $Q(s, a)$ values with RPE-mediated updates, and the REINFORCE algorithm estimates the policy parameters of all the preceding within-episode decisions directly with gradient ascent using the return. The Actor-critic involves both value and policy parameter learning. The V values are estimated by the critic and an RPE is fed into the actor to modify the policy parameters. All three algorithms use eligibility traces to backpropagate updates to preceding actions.

On the other hand, a MB strategy attempts to estimate a model of the world, summarized by the transition matrix $T(s_t, a_t, s_{t+1}) = \mathbf{P}(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t)$ from a state $S_t = s_t$ to a state $S_{t+1} = s_{t+1}$ when selecting action $A_t = a_t$, and the reward function $\bar{R}(s_t, a_t, s_{t+1}) = \mathbb{E}[R_{t+1} = r_{t+1}|S_t = s_t, A_t = a_t, S_{t+1} = s_{t+1}]$. It then uses its estimated model to compute the values of states and action. Analogous to Gläscher et al. (2010), we assume that the reward function is known to participants through the instructions and familiarization with the task; i.e. only one of the images is rewarding. We consider two ways of estimating the transition matrix of the task, a sub-optimal and an optimal way.

First, a traditional approach for learning the transition matrix employs a delta-rule based on the SPE (Forward Learner: Daw et al. (2011a); Gläscher et al. (2010)), that incorporates new information with a constant leak term. In this case, surprising trials are essentially treated as stochasticity in the environment.

Second, the transition matrix can be estimated optimally in a Bayesian fashion, assuming prior knowledge on the structure of the task (generative model), formed by the instructions we gave to participants (i.e. that there will be occasional unexpected transitions, but the task graph does not change). For this, we developed an approximate Bayesian model learning algorithm – a particle filter – that estimates the transition matrix accurately, while maintaining constant computational complexity in time (see Methods - subsection 4.4.3 - for details). The derived update rule involves a surprise-modulated adaptation rate $\gamma_{\mathbf{S}_{\mathrm{BF}}}$, where $\mathbf{S}_{\mathrm{BF}}$ stands for the "Bayes Factor Surprise" we have seen in Chapter 2 (Equation 2.7 and Equation 2.9). Our algorithm essentially implements outlier detection; high values of surprise in this task, signal a surprising transition that should be ignored, since the underlying graph connectivity does not change. Hence, in contrast to Chapter 2, surprise slows down learning, instead of accelerating it.

Either of the two ways to estimate the model of the world can be coupled to a reinforcement learning procedure that estimates the Q values through value iteration (e.g. the Forward Learner (Daw et al., 2011a; Gläscher et al., 2010)) or an approximation thereof (e.g. Prioritized Sweeping (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013)), leading to a total of four MB algorithms. We considered two of these: the Forward Learner that estimates the transition probabilities via an SPE (FWD), and a particle filter (with 20 particles) for the model estimation combined with Prioritized Sweeping (PS with pf), similarly to Chapter 3.

We also considered strategies that are a mixture of MF and MB. A first possibility is that model learning provides an additional teaching signal to the model-free system and affects the value or policy computations implicitly. For example, a high value of $\gamma_{\mathbf{S}_{\mathrm{BF}}}$ or a detection of a surprise trial by the particle filter may dampen ("continuous modulation") or shut off ("binary modulation") the update of the model-free values or the policy parameters of a particular transition (see Methods - subsection 4.4.4 - for details). We indicate methods with binary modulation with the suffix "binary" and

with continuous modulation with "continuous". We introduce the Surprise REINFORCE binary, the Surprise Actor-critic binary and the Surprise SARSA-λ binary, as well as their corresponding continuous versions.

A second possibility is a hybrid strategy that involves a weighted average of MF and MB computations, similar to Daw et al. (2011a); Gläscher et al. (2010); Lee et al. (2014). In the category of hybrid strategies we include the Hybrid Learner-0 (Gläscher et al., 2010), which is a mixture of FWD and SARSA-0, and the Hybrid Learner-λ (Daw et al., 2011a), which is a mixture of FWD and SARSA-λ. Moreover we introduce the following hybrid algorithms: (1) Hybrid-λ-PS-pf, that combines a particle filter (pf) with Prioritized Sweeping (PS) and SARSA-λ, (2) the Surprise Hybrid-λ-PS-pf binary, that combines pf with PS and Surprise SARSA-λ binary, (3) the Surprise Hybrid-λ-PS-pf continuous, that combines pf with PS and Surprise SARSA-λ continuous, and (4) the Hybrid Actor-critic, that is a mixture of the Actor-critic and the FWD. Finally, we also included a random walk with a bias term as a null model. More details on each algorithm can be found in the Methods (subsection 4.4.4).

After the experiment we fit the above algorithms to behaviour, i.e. to participant's actions, using the Metropolis-Hasting Markov Chain Monte Carlo (MCMC) method (Hastings, 1970), similar to Lehmann et al. (2019). In order to perform model comparison we approximated each model's log-evidence using cross-validation. This method is similar to approaches used in statistics and economics (Berger and Pericchi, 1996; Fong and Holmes, 2020; Rust and Schmittlein, 1985; Wang and Pericchi, 2020) and is often considered a more robust method for model comparison than Akaike's Information Criterion (AIC) and Bayesian information criterion (BIC) (Ito and Doya, 2011). We repeated 5 times a 3-fold cross-validation optimization procedure, starting from different random locations in the parameter space for each run and each fold. Each cross-validation run gives us the sum of the estimated maximum negative log-likelihood (LL) across the 3 test folds. We compute the mean of this quantity over the 5 optimization runs and report this as log-evidence for each algorithm. More details on the model fitting procedure can be found in the Methods (subsection 4.4.5).

Fig. 4.2A and Fig. 4.2B depict the results of the model fitting procedure in terms of negative log-evidence. Fig. 4.2A shows all algorithms we considered and Fig. 4.2B is a zoomed version of only the leading algorithms. For simplicity, we omit here the "continuous" versions of the algorithms, which led to similar results as their "binary" versions. We also omit the Surprise Hybrid Learner-λ-PS-pf binary and the Surprise Hybrid Learner-λ-PS-pf continuous, which led to similar or worst performance than the Hybrid-λ-PS-pf, and provide them in the Supplementary Material (Fig. 4.6).

The algorithms that are the most likely models of behaviour are the Actor-critic, the Surprise Actor-critic, the Hybrid Actor-critic and the REINFORCE (Fig. 4.2B). The Actor-critic is weakly significantly better than the other three, with difference in log-

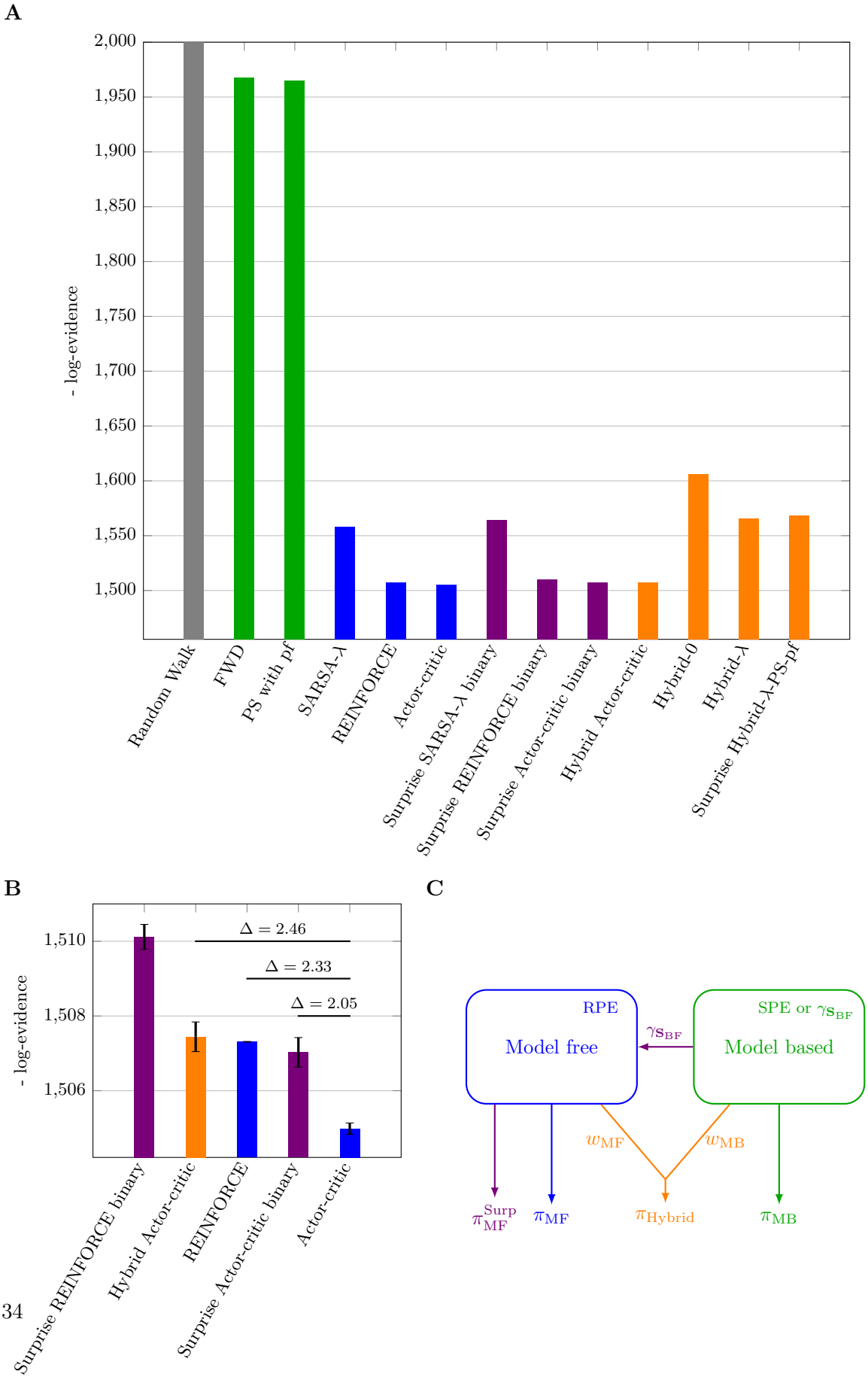

A
2,000
1,950
1,900
1,850
1,800
1,750
1,700
1,650
1,600
1,550
1,500
- log-evidence
Random Walk
FWD
PS with pf
SARSA-λ
REINFORCE
Actor-critic
Surprise SARSA-λ binary
Surprise REINFORCE binary
Surprise Actor-critic binary
Hybrid Actor-critic
Hybrid-0
Hybrid-λ
Surprise Hybrid-λ-PS-pf
B
1,510
1,508
1,506
- log-evidence
Δ = 2.46
Δ = 2.33
Δ = 2.05
Surprise REINFORCE binary
Hybrid Actor-critic
REINFORCE
Surprise Actor-critic binary
Actor-critic
C
RPE
Model free
SPE or γ_SBF
Model based
γ_SBF
w_MF
w_MB
π^Surp_MF
π_MF
π_Hybrid
π_MB

Figure 4.2 *(previous page)* – **Algorithm fit to behaviour. A**. Negative model log-evidence for all algorithms. Smaller values indicate better performance. The most likely models for behavior are the Actor-critic, the Surprise Actor-critic binary, the Hybrid Actor-critic and the REINFORCE. **B**. Negative model log-evidence for only the best algorithms, sorted with increasing performance. The error bars (not marked in A because they were not visible) indicate the standard error of the mean across 5 runs of a 3-fold cross-validation procedure. The log-evidence differences Δ between the Actor-critic and the rest closest winning algorithms are noted on the graph. Δ larger than 3 is considered significant, and larger than 10 strongly significant, but sometimes the significance threshold is placed at the value of 2 (Efron and Hastie, 2016; Held and Ott, 2018; Neath and Cavanaugh, 2012). We thus consider the Actor-critic only weakly significantly better than the other winning algorithms. The Surprise Actor-critic, the Hybrid Actor-critic and the REINFORCE are essentially indistinguishable from each other ($\Delta < 1$). **C**. Schematic illustration of the families of algorithms considered. The participants' policy π might be formed in a pure model free manner (blue), possibly mediated by an RPE, or in a purely model based manner (green), possibly mediated by an SPE or a surprise-modulated rate $\gamma_{\mathbf{S}_{\mathrm{BF}}}$. Alternatively participants may exhibit hybrid policies. One possibility is the modulation of the model free system by a model based surprise-related signal (magenta), either in a continuous or in a binary way (i.e. on/off manner). Another possibility is a weighted sum of a model free (pure or modulated) and a model based strategy (orange). *Abbreviations:* PS: Prioritized Sweeping, pf: Particle Filtering.

evidence (Δ) 2.05, 2.33, and 2.46 from the Surprise Actor-critic, the REINFORCE and the Hybrid Actor-critic, respectively (Fig. 4.2B). Differences larger than 3 are usually considered significant, and larger than 10 strongly significant, but some authors place the significance threshold at the value of 2 instead of 3 (Efron and Hastie, 2016; Held and Ott, 2018; Neath and Cavanaugh, 2012). The Surprise Actor-critic, the Hybrid Actor-critic and the REINFORCE are essentially indistinguishable from each other ($\Delta < 1$). We interpret these results as evidence for the family of algorithms with policy learning and Actor-critic architecture as the most likely model of behaviour, and as not strong evidence for selecting one among them compared to the rest in this family.

The purely MB algorithms seem not appropriate for explaining behaviour in this task. Among the hybrid algorithms, the one that achieves the highest log-evidence is the Hybrid Actor-critic. Overall our results suggest that behaviour is more consistent with model-free learning. The Surprise Actor-critic and the Hybrid Actor-critic do employ a model learning procedure, each one with a different approach. The first one learns to ignore the surprise trials and the second one averages them together in the transition probabilities estimation. Nevertheless, in terms of behavioural fitting, they are indistinguishable from each other and from a purely model-free Actor-critic.

Our behavioural fit favors an estimation of policy parameters rather than Q values, since

overall the frameworks of Actor-critic and REINFORCE are the winning ones. It is worth mentioning that in the Actor-critic the resulting fitted learning rate of the critic was smaller than 0.0001, making it therefore very similar to the REINFORCE algorithm. The exact values of the log-evidence for each algorithm, and its standard error across the optimization rounds, can be found in Table 4.1 of the Methods.

4.2.4 Neural signatures of learning signals

So far we saw that the model-free Actor-critic is sufficient for explaining behavior and weakly significantly better than the Hybrid Actor-critic and the Surprise Actor-critic. At the same time, however, the introduction of surprise trials from the 5th episode onwards causes an increase in the mean path length of participants (Fig. 4.1D). We also found that the participants' reaction times were significantly longer on surprise trials (see Supplementary Material Fig. 4.7). Moreover, at the end of the experiment, participants reported that they were able to notice the occurrence of surprising transitions. We, thus, hypothesized that fMRI brain activity nevertheless correlates with a surprise signal, that is presumably relevant for action selection rather than model updating. We use two different algorithms to evaluate this, the Hybrid Actor-critic and the Surprise Actor-critic. Both rely on RPE and a surprise signal in an Actor-critic architecture. For each of the two algorithms we build one General Linear Model (GLM), where we included their RPE and their respective model learning signals (SPE and $\gamma_{\mathbf{S}_{\mathrm{BF}}}$ for the Hybrid Actor-critic and Surprise Actor-critic, respectively). Signals were time-locked to the occurrence of the states and were othogonalized with respect to the states regressor, but not with respect to each other (see Methods for details on the fMRI data acquisition, preprocessing and statistical analysis).

Using the Hybrid Actor-critic, we find significant correlation of the RPE in the ventromedial prefrontal cortex (vmPFC), in the anterior cingulate gyrus, the posterior orbital gyrus, the parahippocampal gyrus, the inferior occipital gyrus (Fig. 4.3A), as well as in the middle temporal lobe and the fusiform area (not shown). As mentioned earlier, the learning rate of the critic was very low, meaning that there is a very small update of the critic's V values on a trial-per-trial basis. Thus, the RPE takes most of the time (in non-goal states) very small values and higher values at the goal state. Hence, the neural correlates we find largely include regions that have been associated with reward delivery and values, e.g. vmPFC and orbitofrontal cortex (OFC) (Behrens et al., 2008; Chase et al., 2015; Hare et al., 2008; Stalnaker et al., 2018; Wunderlich et al., 2012a), rather than subcortical regions usually reported in the literature for RPEs.

For the MB SPE we find correlated activity in the supplementary motor area (SMA), the anterior mid-cingulate cortex (mACC), the middle frontal gyrus, the angular gyrus, the supramarginal gyrus, the superior parietal lobule and the superior frontal gyrus (Fig. 4.3B). While some of these regions, namely the regions located around the intraparietal sulcus

(angular gyrus, supramarginal gyrus and superior parietal lobule) and regions in prefrontal cortex were also reported in the two-step task of Gläscher et al. (2010), the prefrontal regions we find are shifted to more middle and superior locations than the ones found in Gläscher et al. (2010). All aforementioned regions we find overlap with components of the "salience network" (Seeley et al., 2007), previously associated with the detection of salient or novel stimuli and with error monitoring in order to guide actions. Interestingly, we do not find correlates of the SPE in subcortical structures. The exact coordinates and p-values of the locations showing significant (peak) correlation are provided in Table 4.3.

Using the Surprise Actor-critic, we next correlate BOLD responses with the RPE and the $\gamma_{\mathbf{S}_{\mathrm{BF}}}$ signals, in a separate linear model. For the RPE we overall find the same regions as with the Hybrid Actor-critic. For the $\gamma_{\mathbf{S}_{\mathrm{BF}}}$ we find activation of a small extend in a subset of the regions we reported for the SPE and unilaterally (right middle frontal gyrus, right angular gyrus, left supramarginal gyrus, right superior parietal lobule, right superior frontal gyrus). Empirically, there seems to be an (approximately) one-to-one non-linear relationship between the SPE and the $\gamma_{\mathbf{S}_{\mathrm{BF}}}$ (see Supplementary Fig. 4.8). At least in a linear model, however, the SPE seems to lead to more regions of significant correlation with brain activity (see Supplementary Material subsection 4.5.5 for more details).

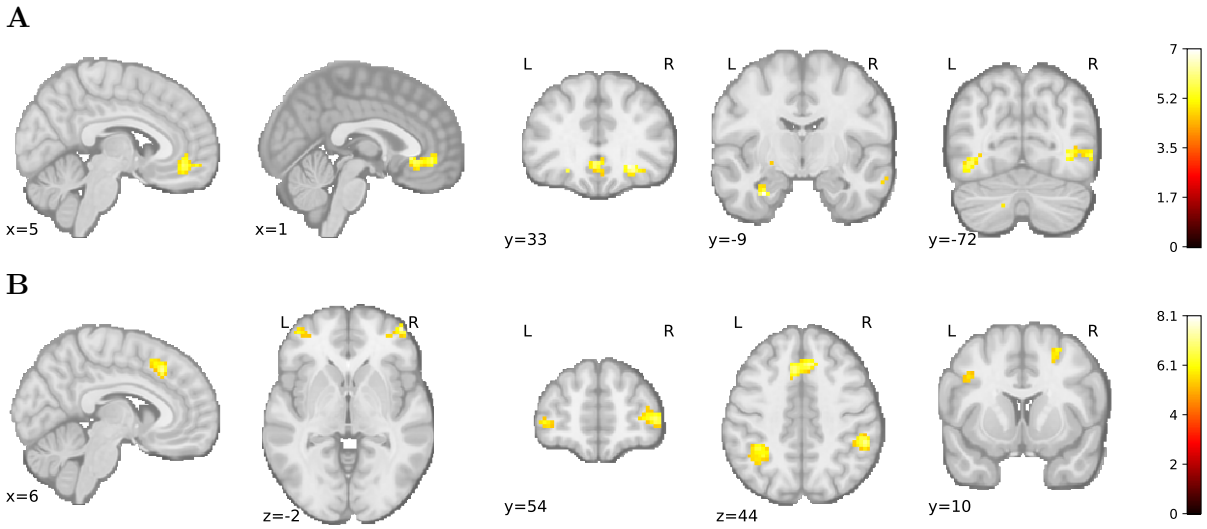


Figure 4.3 – **Neural correlates of model-free and model-based prediction errors.** T-statistic maps of **A**. RPE in ventromedial prefrontal cortex, anterior cingulate gyrus, posterior orbital gyrus, parahippocampal gyrus and inferior orbital gyrus, **B**. SPE in supplementary motor area, middle frontal gyrus, angular gyrus, supramarginal gyrus and in the superior frontal gyrus (21 subjects, random effects analysis, whole brain family-wise error (FWE) correction $p<0.05$, nonparametric permutation test with maximum statistic approach).

4.3 Discussion

We have introduced a novel multi-step decision making task that allows the disentanglement of model-free and model-based prediction errors in human BOLD signals. In our analysis we considered various existing and novel RL algorithms, as well as different ways to estimate the model of the task. We have developed a novel surprise-based particle filtering approach for outlier detection, and hybrid RL algorithms, where the learning rate of the model-free system is modulated by surprise. We have found that human behaviour is best explained by the actor-critic/policy gradient framework. Contributions from the model-based learning system are not detectable in behaviour, but we did find representations of model-based prediction errors in neural signals. We found signatures of RPE in vmPFC and OFC, whereas SPE correlated with activity in the intraparietal sulcus, SMA, and middle frontal gyrus. Our results extend previous fMRI results in a multi-step scenario, support the existence of parallel learning systems in the brain, and add to the collection of learning tasks towards gaining a better understanding of the various aspects of human learning.

In this section we discuss our experiment and its connections to other experiments in the field of animal and human learning. We then discuss further our results on behaviour and neural signatures, and provide a few future directions.

4.3.1 A multi-step decision making task with surprising transitions

Our experiment allows the detection of model-free and model-based brain signatures in a multi-step scenario. As a starting point for the task design, we had the algorithms SARSA-λ (Sutton and Barto, 1998), Forward Learner and their hybrid combination, which have been shown to explain human behaviour in numerous studies (Daw et al., 2011a; Doll et al., 2015a; Economides et al., 2015; Gläscher et al., 2010; Lee et al., 2014; Otto et al., 2013a). However, the idea we follow for the decorrelation of MF and MB prediction errors stays quite generic (but see subsection 4.4.2 in the Methods section for a situation that can reduce the efficiency). It is worth noting that our task does not aim at de-correlating or distinguishing algorithms at a behavioural level, but, given the view that humans implement both MF and MB strategies, de-correlating prediction errors at the level of brain signals. Moreover, our task does not seek to dissociate different possible MB signals from each other (i.e. SPE from other surprise signals), but MF signals from model learning ones.

We have shown that Bayesian inference on the generative model of our task leads to a surprise signal that inhibits learning, rather than accelerating it (see Methods section 4.4). The concept of surprise having different effect on learning depending on the statistical context has been previously proposed and developed for tasks involving tracking of targets in Gaussian settings (d'Acremont and Bossaerts, 2016; Nassar et al., 2019). Here, we

start from a general generative model describing the occurrence of outliers and develop an approximate Bayesian algorithm for a general case. Previous work has sought to differentiate behavioural and brain responses for the case that learning should increase (in a change-point setting) versus when learning should decrease (in a outlier occurrence setting) (d'Acremont and Bossaerts, 2016; Nassar et al., 2019). In our work, we focused on dissociating signals related to reward from those related to model learning, and the use of outliers served as a handle towards this goal.

We see connections between our task and tasks developed recently for studying the role of dopamine in learning (Langdon et al., 2018). For example, Uchida and colleagues employed a virtual navigation task where mice were teleported to different tracks with same distance to goal. They found that the ramping activity of dopamine neurons codes for an RPE and not sensory surprise (Mikhael et al. (2019) and unpublished data). In another study, Takahashi et al. (2017) administered reward of the same value but different identity (flavor) to rats. This increased the firing rate of some dopamine neurons, suggesting that they respond to errors in reward identity, and not only to reward quantity, and that dopamine may relay a multi-dimensional prediction error (Stalnaker et al., 2019; Takahashi et al., 2017). Similarly, in an fMRI study (Howard and Kahnt, 2018), the identity of an unexpected odor with same pleasantness could be decoded from midbrain BOLD signals (Howard and Kahnt, 2018; Stalnaker et al., 2019). The focus, the tasks or the nature of the data of the above studies differ from ours, but the common line is the introduction of a (sensory) change, while keeping the value similar. Here, we did not find striatal activation uniquely explained by SPE or surprise. However, dopaminergic neurons are known to project to many other regions in the brain apart from the striatum, such as the prefrontal cortex, thus it is hard to tell from our data if dopamine is or is not involved in the SPE.

4.3.2 Behaviour is best explained by model-free learning

Participants learn fast and all the models that are the most likely descriptions of behaviour make use of eligibility traces, consistent with findings in Lehmann et al. (2019). The winning algorithms come from the family of policy gradient methods, with possible contributions of RPE from a critic in the Actor-Critic architecture and from a surprise signal. Policy learning has received relatively less attention in human studies, with few exceptions (Ito and Doya, 2011; Li and Daw, 2011; O'Doherty et al., 2004). More attention is likely to be drawn, as recent studies indicate that the activity of midbrain dopamine neurons seems to be closely related to the initiation of actions, and that policy learning is a likely framework to reconcile these observations (Coddington and Dudman, 2019).

The reasons why these algorithms fit better with behaviour in our task are not clear. One possibility is that the REINFORCE and the Actor-critic are more flexible and allow the policy to turn into deterministic behaviour (for constant temperature τ), so they

may be able to capture some gradual changes in the exploration strategy of participants that SARSA-λ cannot capture. Another possibility is that, similarly to Li and Daw (2011) our task has a symmetric structure for the two actions, i.e. most of the time one action is better than the other. REINFORCE allows for a simultaneous decrease in the selection probability of the alternative action when the chosen one proved to be successful. Further analysis, for example including the symmetric structure of the task in the other algorithms, may help clarify the underlying reasons behind our behavioural fitting results.

From a pure model building perspective, our task can be seen as an outlier detection task, where the optimal behaviour is to ignore the improbable observations. However, the hybrid algorithm that includes an outlier detector modulated by surprise performed similarly to algorithms without it, in terms of behavioural fit. Moreover, the resulting values for the fitted stochasticity level (parameter σ of the particle filter – see Methods subsection 4.4.3 for details on the algorithm) were high. Taken together, these suggest that participants may have not perceived the task as a purely deterministic one with outliers, despite being instructed that the underlying graph does not change.

It is worth pointing out that our behavioural analysis carries the assumption that participants use the same learning algorithm in this task, possibly with different parameter values (e.g. learning rate). However, learning processes may differ across participants. Since our task is not designed to make clear distinctions between learning algorithms at the behavioural level, we did not consider this possibility here. Nevertheless, it may still be interesting to investigate this possibility using a group-level analysis method (Rigoux et al., 2014; Stephan et al., 2009).

4.3.3 Model-free and model-based neural signatures

Overall, learning the model of the task does not increase significantly the fit of the respective hybrid algorithms and is not manifested clearly in behaviour. Nevertheless, we find correlates of a MB SPE and weaker, but largely overlapping, correlates of $\gamma_{\mathbf{S}_{\mathrm{BF}}}$. This may suggest that a MB learning system is active, possibly building an internal model of the task and performing "latent learning" (Bast et al., 2009; Tolman, 1948), but it is not (yet) in control. Such an interpretation is consistent with the idea that a "mixture of experts" co-exist and run in parallel in the brain (Daw et al., 2005; Lee et al., 2014; O'Doherty et al., 2020), and the control of the behaviour is delegated among them depending on the circumstances and on multiple factors such as the uncertainty of each expert and time constraints.

More specifically, concerning the neural representation of SPE, we find regions belonging to the salience network (Seeley et al., 2007). The intraparietal sulcus has been found to correlate with SPE and surprise signals in previous studies (Gläscher et al., 2010; Lee et al., 2014; Schad et al., 2020), as well as regions in the lateral prefrontal and

orbitofrontal cortex (Doll et al., 2012; Gläscher et al., 2010; O'Doherty et al., 2015; Simon and Daw, 2011). Moreover, the mACC and the SMA have been found to be components of the network related to surprise (Fouragnan et al., 2018). Surprise and its network have been viewed to comprise two roles. One role is the encoding of saliency or how much an observation protrudes among others, driving an attentional mechanism that helps in guiding actions. A second role is the implementation of a learning signal that mediates the updating of beliefs and better future predictions (Fouragnan et al., 2018). Concerning the first role, representations of surprise signals have been found in lateral parietal cortex and the SMA, whereas the second role has been additionally associated with other brain structures, such the insula and the striatum (Fouragnan et al., 2018). Our results are consistent with the first role of surprise, also referred to as puzzlement surprise (Faraji et al., 2018).

For the model-free RPE we found signatures in vmPFC, the anterior cingulate gyrus and the posterior orbital gyrus. The vmPFC has been previously reported to correlate with reward prediction errors, and more commonly with reward expectation and reward receipt (Behrens et al., 2008; Chase et al., 2015; Daw et al., 2011a; Hare et al., 2008; Stalnaker et al., 2018; Wunderlich et al., 2012a). The anterior cingulate gyrus has also been reported to be active with expected values and with the assessment of outcomes (Chase et al., 2015; Kolling et al., 2016). We did not find correlates of RPE in subcortical regions, such as the ventral striatum. This can be explained by the fact that given our fitted parameters the RPE timeline follows closely the timeline of reward receipt (higher values at the goal state, lower elsewhere). Moreover, we did not perform a region-of-interest analysis focused on specific regions, often done in the literature. At the same time, we found activation in the putamen to be correlating with action selection, which speaks in favour of an actor-based algorithm. Putamen is part of the dorsal striatum, and is known to receive dopaminergic input from the substantia nigra and to be involved in motor planning and execution (Takahashi et al., 2008).

The RPE timelines of the leading hybrid algorithms Hybrid Actor-critic and Surprise Actor-critic give rise to similar brain regions with significant activations. Even more interestingly, the timelines of SPE and $\gamma_{\mathbf{S}_{\mathrm{BF}}}$, that stem from the different update rules of the corresponding models, also give rise to similar brain regions with significant activations. Thus, the observed neural representations seem to be robust and our results point to regions involved in this type of computations, beyond the specific details of each signal and each algorithm.

4.3.4 Future directions

We provide a method for decorrelating model-free and model-based prediction errors, that can be used in different settings and can be combined with other experimental manipulations. Our analysis of the behaviour and of the fMRI data indicates that

behaviour in this task is explained best by model-free RL, that surprise trials were perceived, and that model-based signals were calculated in the brain. Combining our task with, for example, a change of goal location at a later stage would allow to assess whether participants built indeed an internal model of a task.

In addition to finding neural correlates of model-based prediction errors, we found that participants may have perceived the task as stochastic, rather than deterministic with occasional outliers, despite having been informed that the graph does not change. A recent study (da Silva and Hare, 2020) on the two-stage task (Daw et al., 2011a) pointed out the impact of participants' understanding of the task on behaviour. The authors also showed that if a simulated agent is model-based but is using a "wrong" model of the task structure, then the apparent best fit for behaviour can be a hybrid mixture of model-free and model-based. Under the assumption that there were model-based contributions in our task, the question is then, what is the model structure that human subjects used? Does behaviour appear model-free because we yet do not know the "imperfect" model and updating scheme that humans function with? These are in our view central questions towards understanding human learning behaviour and more theoretical as well as experimental work are needed to address them.

4.4 Methods

4.4.1 Participants and experiment details

Twenty-three healthy adults (average age 23.8 years old, right-handed, 10 female) were recruited to participate in our experiment. All participants provided written informed consent, and the experiment was conducted in accordance with the ethics commission of the Canton de Vaud, Switzerland. Participants performed the task in a 3T Siemens Prisma MRI Scanner at the Laboratoire de recherche en neuroimagerie (LREN) at the Centre hospitalier universitaire vaudois (CHUV). Prior to the experiment, participants were informed about the number of states and possible actions, and got familiar with the task outside and inside the scanner during short sessions of two episodes each, with different images and transitions than the ones used during the experiment. Furthermore, participants were beforehand informed on the existence of surprise trials and on the fact that the underlying transition matrix does not change.

We excluded two participants from both our behavioural and fMRI analysis: One participant performed less than half of the average number of episodes that the rest of the participants and likely did not understand the task, and another participant was falling asleep and his brain images exhibited a high degree of movement artifacts. The remaining 21 participants performed on average 54 full episodes (std: 5.24), and 188 actions (std: 8.33). From these, approximately 17% (std: 1.5 %) were surprise trials. Fractal images, their locations on the screen, and the assignment of transitions to left or right action

presses were randomized across participants. We employed two different underlying transition matrices also in a randomized way across participants. See Supplementary Material Table 4.2 for the distance of each state from goal and the action "correctness". Participants were compensated with a fixed monetary amount for their participation, plus a small extra performance-based amount.

After each action taken by a participant, from a state s to (an expected state) s', we checked whether the following conditions were fulfilled: (i) The transition from s to s' is learned according to the Forward Learner we ran online, (ii) there is a state s'' so that $|V(s') - V(s'')| \leq \Delta V$ where $V(s) = \max_a Q(s, a)$ and ΔV is a small threshold, according to the SARSA-λ we ran online, and (iii) more than 3 trials have occurred since the last surprise trial. If these conditions were fulfilled, the participant transited to s'', and this constituted a surprise trial/surprising transition. Moreover, if during 8 consecutive trials no surprise trial had occurred, i.e. the above conditions were not fulfilled, a randomly chosen unexpected transition was enforced in order to ensure some variability – excluding the goal state and the current state from possible landing states. Thus we have two types of surprise trials: those that meet the threshold criterion on V values and purely random transitions. We did not perform surprise trials during the first 4 episodes of the experiment.

4.4.2 Post-hoc analysis on the RPE/SPE decorrelation

After the experiment we fitted the SARSA-λ (Sutton and Barto, 1998) and the Forward Learner (Daw et al., 2011a; Gläscher et al., 2010) to participants' behaviour, obtained their corresponding RPE and SPE values, and validated the original purpose of our experimental design. Fig. 4.4A depicts the values of the RPE and the SPE for one representative participant. Values that correspond to surprise trials are marked in red, and non-surprise trials in blue. The Pearson correlation coefficient of RPE and SPE without surprise trials (blue values) is $r = +0.49$. Adding controlled surprise trials to the experiment yields an effective de-correlation $r = +0.011$ for this participant.

The mean absolute RPE and SPE correlation across participants is 0.105 ± 0.063 (mean ± std, 21 participants). The maximum correlation observed was 0.193 and the minimum -0.151. Thus, our experimental design successfully breaks the correlation of the two prediction errors over the course of learning. Fig. 4.4B and C show the histograms of the RPE and the SPE, respectively, at surprise trials for all participants. The distribution of the SPE is shifted towards higher values, whereas the one of RPE is centered around 0.

Fig. 4.5 depicts the same quantities for the Hybrid Actor-critic algorithm, which was among the winning algorithms, with parameters fitted to the participants' data. In this case the RPE and SPE are moderately anti-correlated. The mean correlation of RPE and SPE across participants is -0.503 ± 0.122 (mean ± std, 21 participants), with a

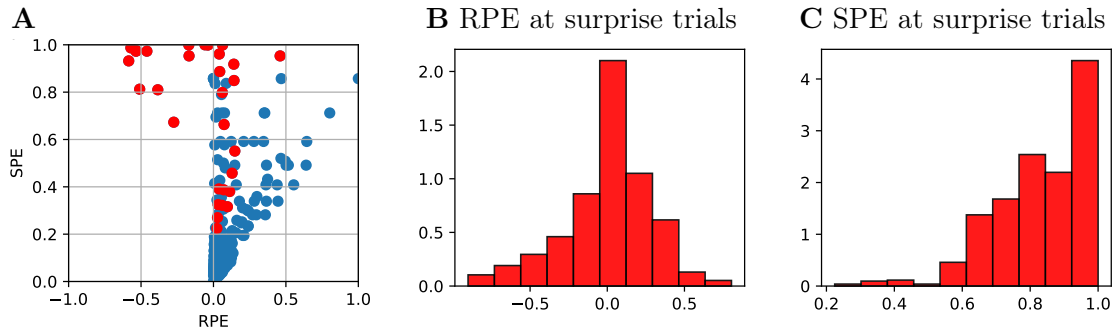


Figure 4.4 – **Post-hoc validation of RPE/SPE de-correlation.** **A**. Each circle corresponds to a joint RPE/SPE value (of SARSA-λ and Forward Learner respectively) for each of the 192 actions of one participant. Surprise trials are indicated in red and non-surprise ones in blue. Without surprise trials (blue values only), RPE and SPE have a Pearson Correlation Coefficient $r = +0.49$. The addition of surprise trials to the experiment successfully decreased the correlation of the two signals ($r = +0.011$). **B**. Histogram of RPE values at surprise trials and of **C**. SPE values at surprise trials, for all 675 surprise trials across all participants. We observe that surprise trials have overall low RPE and high SPE values.

maximum correlation of -0.258 and a minimum of -0.684. Because the fitted learning rate of the critic is very low, the V values are effectively not updated, and the RPE clusters around zero and one: the RPE assumes very high values at the goal state and very low values otherwise. The SPE at the goal state has lower values because it is a well-learned transition. Nevertheless, the use of surprise trials successfully adds variability to the SPE (Fig. 4.5C). The surprise trials push its distribution to higher values (Fig. 4.5C, red) and make the SPE detectable in fMRI.

4.4.3 An approximate Bayesian algorithm for outlier detection

In the section we describe the approximate Bayesian model learning algorithm we developed that estimates the transition matrix of our task.

The generative model of the task

At each time step (trial) t the participant is at a state $S_t = s \in \mathcal{S}$, where $\mathcal{S} = \{1, ..., 7\}$ the set of possible states. Upon selection of an action $A_t = a \in \mathcal{A}$, where $|\mathcal{A}| = 2$ the participant observes at the next time step $t+1$ a state $S_{t+1} = s' \in \mathcal{S}$ which is drawn from a *fixed in time* probability vector $P^{sa} = p^{sa} \in [0,1]^{|\mathcal{S}|}$. Or equivalently, the next state s' is drawn from a distribution with parameters p^{sa}, i.e. $s'|p^{sa} \sim P(s'; p^{sa})$. However, at every time step t, there is a *jump probability* $p_j \in (0,1)$ for a surprising transition to take place. The occurrence of a surprise trial is indicated by the event $Z_t^{sa} = 1$; otherwise

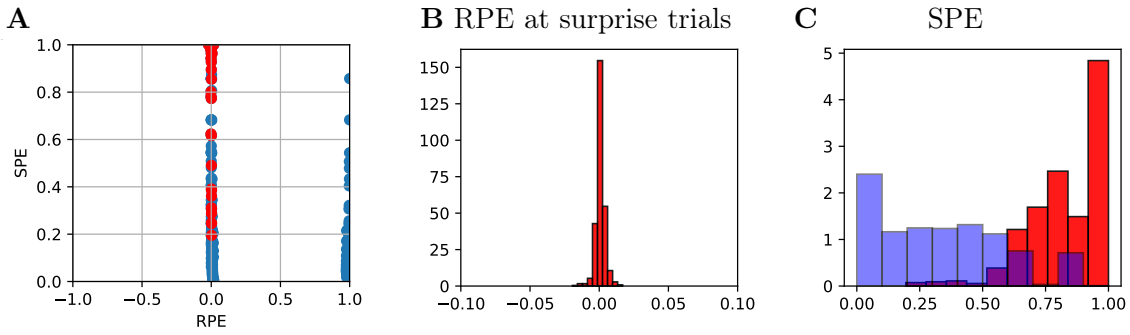


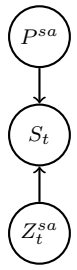
Figure 4.5 – **RPE/SPE correlation in Hybrid Actor-critic. A**. Each circle corresponds to a joint RPE/SPE value of the fitted Hybrid Actor-critic model for each of the 192 actions of one participant. Surprise trials are indicated in red and normal trials in blue. Surprise trials add variability to the SPE, but the RPE and SPE values of Hybrid Actor-critic are anti-correlated (Pearson Correlation Coefficient $r = -0.3$ for this participant). This is because RPE is high at the goal states, where SPE is low since it is a well-learned transition. **B**. Histogram of RPE values at surprise trials for all 675 surprise trials across all participants. **C**. SPE values at surprise trials in red (675 trials) and at non-surprise trials in blue (3278 trials) . Surprise trials shift the distribution of SPE to higher values, compared to the non-surprising trials.

$Z_t^{sa} = 0$. On a surprise trial the next state s' is uniformly selected from a pool of available states, i.e. is drawn from a known (uniform) distribution, i.e. $s'| \sim U(s')$. Note that our criterion for the landing state upon a surprise trial, i.e. having a V value close to the state that was expected (see subsection 4.4.1 for details) is neglected here, since the participants are agnostic to this fact.

The next observed state s' at time step $t+1$ is drawn from the distribution corresponding to the currently experienced state-action pair at time t. Thus, as in Chapter 3, the time index t refers to real time (discrete time steps), and we define the variable $\mathcal{T}(s,a)$ as the set of timepoints in $[1,t]$ that a particular (s,a) pair is visited.

Our task can be formalized as a set of generative models of the following form, *for each* (s,a) *pair*

$$
\begin{aligned}
&\mathbf{P}(p^{sa}) = \mathbb{b}^{sa,(0)}(.)\,, \\
&\mathbf{P}(z_t^{sa}) = \text{Bernoulli}(p_j)\,, \\
&\mathbf{P}(s'|p^{sa}, z_t^{sa}) = \begin{cases} P(s'|p^{sa}) & \text{if} \quad z_t^{sa} = 0 \quad \text{and} \quad t-1 \in \mathcal{T}(s,a)\,, \\ U(s') & \text{if} \quad z_t^{sa} = 1 \quad \text{and} \quad t-1 \in \mathcal{T}(s,a)\,. \end{cases}
\end{aligned} \tag{4.1}
$$



We recall that random variables are indicated by capital letters, and values by small

letters, and we often drop the indication of the random variables to ease notation. $\mathbf{P}$ indicates either a probability density function (for the continuous variables) or a probability mass function (for the discrete variables). $\mathbb{b}^{sa,(0)}$ is a prior distribution from which the parameters p^{sa} are drawn at the beginning and stay fixed throughout the task. P is the time-invariant likelihood function and U can be any distribution.

The goal of a participant or an agent learning the model of the task is the estimation of the parameters p^{sa}, for each (s, a) pair. Given a sequence of observed states $s_{1:t}$, the participant may maintain a *belief* $\mathbb{b}^{sa,(t)}(p^{sa})$ about the parameter p^{sa} at time t, which, as we saw in the previous chapters, is defined as the posterior probability distribution $\mathbf{P}(P^{sa} = p^{sa}|s_{1:t})$. The participant's goal is to update the belief $\mathbb{b}^{sa,(t)}(p^{sa})$ to the new belief $\mathbb{b}^{sa,(t+1)}(p^{sa}) = \mathbf{P}(P^{sa} = p^{sa}|s_{1:t+1})$, or an approximation thereof, upon observing s_{t+1}.

It can be shown that the updated belief $\mathbb{b}^{sa,(t+1)}(p^{sa})$ after observing s_{t+1} is

$$\mathbb{b}^{sa,(t+1)}(p^{sa}) = (1 - \gamma^{(t+1)}_{\mathbf{S}_{\mathrm{BF}}})\mathbb{b}^{sa,(t+1)}_{B}(p^{sa}) + \gamma^{(t+1)}_{\mathbf{S}_{\mathrm{BF}}}\mathbb{b}^{sa,(t)}(p^{sa}), \tag{4.2}$$

The derivation follows the same steps as in Chapter 2 (section 2.4) for the proof of the Proposition (Equation 2.10) and is also provided in the Supplementary Material of this chapter for completeness.

We briefly explain the terms appearing in Equation 4.2.

The surprise-modulated adaptation rate $\gamma^{(t+1)}_{\mathbf{S}_{\mathrm{BF}}}$ is

$$\gamma^{(t+1)}_{\mathbf{S}_{\mathrm{BF}}} = \frac{m\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \mathbb{b}^{sa,(t)})}{1 + m\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \mathbb{b}^{sa,(t)})}, \tag{4.3}$$

where $m = \frac{p_j}{1-p_j}$. The $\mathbf{S}_{\mathrm{BF}}$ that modulates the adaptation rate is the "Bayes Factor Surprise" of the observation $S_{t+1} = s_{t+1}$, which is defined as the ratio of the probability of observing $S_{t+1} = s_{t+1}$ given $Z^{sa}_{t+1} = 1$ (i.e. given a surprise trial), to the probability of observing $S_{t+1} = s_{t+1}$ given $Z^{sa}_{t+1} = 0$ (i.e. given a non-surprise trial)

$$\mathbf{S}_{\mathrm{BF}}(y_{t+1}; \mathbb{b}^{sa,(t)}) = \frac{U(s_{t+1})}{P(s_{t+1}; \mathbb{b}^{sa,(t)})}\,. \tag{4.4}$$

This definition is conceptually the same as the definition of $\mathbf{S}_{\mathrm{BF}}$ we saw in Chapter 2 (Equation 2.7) for this generative model, for the case of jumps instead of change points (see Supplementary Material).

$\mathbb{b}^{sa,(t+1)}_{B}(p^{sa})$ is the updated belief corresponding to a simple Bayesian update, i.e. to the

incorporation of the new observation to the current belief using Bayes' rule,

$$\mathbb{b}_B^{sa,(t+1)}(p^{sa}) = \frac{P(s_{t+1}|p^{sa})\mathbb{b}^{sa,(t)}(p^{sa})}{P(s_{t+1};\mathbb{b}^{sa,(t)})}. \tag{4.5}$$

We can observe that the resulting belief $\mathbb{b}^{sa,(t+1)}(p^{sa})$ of Equation 4.2 is a weighted average between integrating the new observation to the current belief and maintaining the current belief (i.e. ignoring the new observation). That is

$$\mathbb{b}^{\text{new}}(p^{sa}) = (1-\gamma_{\mathbf{S}_{\text{BF}}})\,\mathbb{b}^{\text{integration}}(p^{sa}|s^{\text{new}},\mathbb{b}^{\text{old}}) + \gamma_{\mathbf{S}_{\text{BF}}}\,\mathbb{b}^{\text{old}}(p^{sa}). \tag{4.6}$$

Bayesian inference on this generative model leads to an updating scheme that performs outlier detection, similar to d'Acremont and Bossaerts (2016); Nassar et al. (2019). Interestingly, in this setting surprise *modulates* learning, but *does not accelerate* it; on the contrary high surprise reduces the influence of the new observation on the update.

Particle Filtering

Computing the updated belief of Equation 4.2 at each time step is computational intensive. We describe here a way to approximate the belief with Particle Filtering.

One way to compute the belief at time $t+1$ is through marginalization over the hidden variables $z_{1:t+1}^{sa}$, i.e.

$$\mathbb{b}^{sa,(t+1)}(p^{sa}) = \sum_{z_{1:t+1}} \mathbf{P}(p^{sa}|z_{1:t+1}^{sa}, s_{1:t+1})\mathbf{P}(z_{1:t+1}^{sa}|s_{1:t+1}). \tag{4.7}$$

The first term $\mathbf{P}(p^{sa}|z_{1:t+1}^{sa}, s_{1:t+1})$ in the sum can be easily computed; given the knowledge of the occurrences of surprise trials, we simply gather together all the non-surprising observations and use them to calculate the distribution of p^{sa}.

The term $\mathbf{P}(z_{1:t+1}^{sa}|s_{1:t+1})$ is however difficult to compute and the summation over all possible sequences of surprise trial occurrences is computationally expensive. We approximate this term via particle filtering (Gordon et al., 1993), i.e.

$$\mathbf{P}(z_{1:t+1}^{sa}|s_{1:t+1}) \approx \sum_{i=1}^{N} w_{t+1}^{sa,(i)}\delta(z_{1:t+1}^{sa} - z_{1:t+1}^{sa,(i)}), \tag{4.8}$$

where $\{z_{1:t+1}^{sa,(i)}\}_{i=1}^N$ is a set of N realizations (particles) of $z_{1:t}^{sa}$ drawn from a proposal distribution $\Psi(z_{1:t+1}^{sa}|s_{1:t+1})$, and $\{w_{t+1}^{sa,(i)}\}_{i=1}^N$ are their corresponding weights at time $t+1$.

Therefore the approximated belief is

$$\hat{\mathbb{b}}^{sa,(t+1)}(p^{sa}) = \sum_{i=1}^{N} w_{t+1}^{(i)} \hat{\mathbb{b}}_i^{sa,(t+1)}(p^{sa}) = \sum_{i=1}^{N} w_{t+1}^{sa,(i)} \mathbf{P}(p^{sa}|z_{1:t+1}^{sa,(i)}, s_{1:t+1}) \,, \tag{4.9}$$

where $\hat{\mathbb{b}}_i^{sa,(t+1)}(p^{sa})$ is the approximated belief of each particle i. At each time step we (i) update the weights $\{w_{t+1}^{sa,(i)}\}_{i=1}^{N}$, and (ii) sample the new state $z_{t+1}^{sa,(i)}$ for all N particles.

It can be shown (the derivation follows the same steps as in Chapter 2 (section 2.4) and is also provided in the Supplementary Material of this chapter) that the weight update of the particles is

$$w_{t+1}^{sa,(i)} = (1 - \gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)}) w_{B,t+1}^{sa,(i)} + \gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)} w_t^{sa,(i)} \,, \tag{4.10}$$

where $\gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)} = \frac{m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{sa,(t)})}{1+m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{sa,(t)})}$ the surprise-modulated adaptation rate of Equation 4.3, $m = \frac{p_j}{1-p_j}$, and $\mathbf{S}_{\mathrm{BF}}$ of Equation 4.4. Also

$$w_{B,t+1}^{sa,(i)} = \frac{P(s_{t+1}; \hat{\mathbb{b}}_i^{sa,(t)})}{P(s_{t+1}; \hat{\mathbb{b}}^{sa,(t)})} w_t^{sa,(i)} \tag{4.11}$$

are the weights that correspond to the incorporation of the new observation, i.e. to the Bayesian update $\mathbb{b}_B^{sa,(t+1)}$.

We can see that the weight update rule exhibits the same surprise-modulated trade-off as Equation 4.2.

At each time step we sample the new state $z_{t+1}^{sa,(i)}$ of each particle i from a proposal distribution $\Psi(z_{t+1}^{sa,(i)}|z_{1:t}^{sa,(i)}, s_{1:t+1})$. The probability for a particle i to interpret the observed transition as a surprise trial is (see Supplementary Material)

$$\Psi(z_{t+1}^{sa,(i)} = 1|z_{1:t}^{sa,(i)}, s_{1:t+1}) = \gamma_{\mathbf{S}_{\mathrm{BF}}}\Big(\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \hat{\mathbb{b}}_i^{sa,(t)}), \frac{p_c}{1-p_c}\Big) \,. \tag{4.12}$$

Since $\gamma_{\mathbf{S}_{\mathrm{BF}}}$ is a monotonous function of surprise, the more surprising the observed state s_{t+1} is, the more likely it will be for the particle's hidden state to take the value $z_{t+1}^{sa,(i)} = 1$.

Finally the approximated belief is

$$\hat{\mathbb{b}}^{sa,(t+1)}(p^{sa}) = \sum_{i=1}^{N} w_{t+1}^{sa,(i)} \mathbf{P}(p^{sa}|z_{1:t+1}^{sa,(i)}, s_{1:t+1}) \quad = \sum_{i=1}^{N} w_{t+1}^{sa,(i)} \mathbf{P}(p^{sa}|s_{m_{[1:t+1]}^{sa,(i)}}), \tag{4.13}$$

where we denote $m_{[1:t+1]}^{sa,(i)}$ the time points of non-surprise trials within the time window $t' = 1$ to $t' = t+1$ for particle i for a state-action pair, i.e. $m_{[1:t]}^{sa,(i)} = \{k \in \{1, ..., t\} | z_k^{sa,(i)} = 0, s = s_k, a = a_k\}$.

The implementation details of our particle filtering (e.g. the resampling procedure) are the same as described in Chapter 2 (See section 2.4 and Algorithm 3 for the pseudocode).

Approximating the belief via Equation 4.13 may give the impression that the whole history of observations $s_{1:t+1}$ and of particles' hidden states $z_{1:t+1}^{sa,(i)}$ have to be kept in memory. But in many tasks, and in particular in our task, this is not needed and all information about the previous observation can be summarized in counts, similar to the previous chapters.

Without loss of generality, we can formulate our task using distributions from the exponential family. Then the parameters p^{sa} are drawn from a Dirichlet distribution $\mathrm{Dir}(\sigma \cdot \mathbf{1})$, where $\sigma \in (0, \infty)$ is the stochasticity parameter. On a non-surprise trial the next state $s_{t+1} \in \{s^{(1)}, ..., s^{(7)}\}$ is drawn from a categorical distribution with parameters p^{sa}, i.e. $s_{t+1}|p^{sa} \sim \mathrm{Cat}(s_{t+1}; p^{sa})$. As in Chapter 3, following the general formulation of the exponential family of distributions (given in Chapter 2) it can then be shown that for the estimation of the transition probabilities p^{sa} to all other states, for the currently experienced state and action pair (s, a) the update of the unnormalized weights $\tilde{w}_{t+1}^{sa,(i)}$ upon the observation of the next state $s_{t+1} = s'$ is

$$\tilde{w}_{t+1}^{sa,(i)} = \Big((1 - p_j)\frac{N_t^{(i)}(s, a, s_{t+1}) + \sigma}{\sum_{k=1}^{|S|}\big(N_t^{(i)}(s, a, s^{(k)}) + \sigma\big)} + p_j \frac{1}{|S|}\Big)\tilde{w}_t^{sa,(i)}, \tag{4.14}$$

where $N_t^{(i)}(s, a, s_{t+1}) = \sum_{t'=m_{[1:t]}^{sa,(i)}}^{t}[S_{t'} = s_{t+1}]$ are the counts for the occurrence of a state s_{t+1} from (s, a) at all non-surprising trials, as counted by the ith particle ([.] denotes the Iverson bracket, and equals to 1 if the condition within the bracket is fulfilled, 0 otherwise). Intuitively, the above equation entails a weighted average between the expected value of the transition vector p^{sa} as calculated under the particle's current hidden state, and a uniform reset.

Finally, the estimated transitions from the particle filter from any (s, a) pair to any state s' are

$$\hat{T}_t(s, a, s') = \mathbb{E}_{\mathbb{b}^{sa,(t)}}[P^{sa}] = \sum_{i=1}^{N} w_t^{sa,(i)} \frac{N_t^{(i)}(s, a, s') + \sigma}{\sum_{k=1}^{|\mathcal{S}|}\big(N_t^{(i)}(s, a, s^{(k)}) + \sigma\big)}. \tag{4.15}$$

This estimation is then used in combination with Prioritized Sweeping, similar to Chapter 3.

4.4.4 Reinforcement learning algorithms

We briefly describe here the algorithms we considered or introduced. More details on RL algorithms can be found in Chapter 1. At a time step t, the agent is at a state $s_t \in \mathcal{S}$, selects and action $a_t \in \mathcal{A}$, which results in a transition to a state $s_{t+1} \in \mathcal{S}$ and

the observation of the reward $r_{t+1} \in \mathcal{R}$ at the next time step $t+1$, upon which the agent updates its estimations. In all following algorithms, $\alpha \in (0, 1]$ denotes the learning rate and $\gamma \in [0, 1]$ the discount factor.

SARSA-λ. SARSA-λ (Sutton and Barto, 1998) estimates its (model-free) Q values Q_{MF} via the reward prediction error RPE in the following way

$$\begin{aligned} &RPE_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \\ &Q(s,a) \leftarrow Q(s,a) + \alpha\, RPE_t\, e_t(s,a), \quad \forall\, s \in \mathcal{S}, a \in \mathcal{A} \\ &e_t(s,a) = \begin{cases} 1, & \text{if } s_t = s, a_t = a \\ \gamma\lambda e_{t-1}(s,a), & \text{otherwise,} \end{cases} \end{aligned} \tag{4.16}$$

where the immediate reward r_{t+1} is 1 for the goal state, and 0 for all other states), and $e_t(s,a)$ are the exponentially decaying eligibility traces with decay factor $\lambda \in [0, 1]$, initialized to zero. Note that this is implementation of eligibility traces is called "replacing eligibility traces" (Sutton and Barto, 1998).

Forward Learner. The Forward Learner (Daw et al., 2011a; Gläscher et al., 2010) estimates the true transition matrix T via the state prediction error SPE. It then estimates its (model-based) Q values Q_{MB} through value iteration.

$$\begin{aligned} &SPE_t = 1 - \hat{T}(s_t, a_t, s_{t+1}) \\ &\hat{T}(s_t, a_t, s') \leftarrow \hat{T}(s_t, a_t, s') + \alpha\, SPE_t, \quad \text{if } s_{t+1} = s' \\ &\hat{T}(s_t, a_t, s') \leftarrow \hat{T}(s_t, a_t, s') - \alpha\, \hat{T}(s_t, a_t, s'), \quad \text{otherwise} \\ &Q(s,a) = \sum_{s'} \hat{T}(s,a,s') \big[\bar{R}(s,a,s') + \gamma V^{\pi}(s')\big], \quad \forall\, s \in \mathcal{S}, a \in \mathcal{A}, s \in \mathcal{S}, \end{aligned} \tag{4.17}$$

where $\hat{T}$ the estimated transition matrix. Analogous to Gläscher et al. (2010) we assume the reward function $\bar{R}$ to be known to participants through the instructions and familiarization with the task; i.e. only one of the images is rewarding.

Hybrid Learner. The Hybrid Learner (Daw et al., 2011a; Gläscher et al., 2010) is a (time-dependent) weighted average of SARSA-λ and Forward Learner.

$$\begin{aligned} &Q(s,a) = w_t\, Q_{MB}(s,a) + (1 - w_t)\, Q_{MF}(s,a), \quad \forall\, s \in \mathcal{S}, a \in \mathcal{A} \\ &w_t = w_0\, e^{-kt}, \end{aligned} \tag{4.18}$$

where Q_{MB} and Q_{MF} the values of the model-based Forward Learner and of the model-free SARSA-λ respectively. The weight between the two systems can change in time in an exponential fashion, controlled by an offset w_0 and a decay slope k.

Actor-critic. The Actor-critic (Sutton and Barto, 1998) is a model-free algorithm, where the V values are estimated by the critic and a RPE is fed into the actor to modify

the policy parameters.

$$
\begin{aligned}
&RPE_t = r_{t+1} + \gamma\, V(s_{t+1}) - V(s_t) \\
&V(s) \leftarrow V(s) + \alpha_c\, RPE_t\, e_t^c(s) \quad \forall\, s \in \mathcal{S} \\
&e_t^c(s) = \begin{cases} 1, \text{ if } s_t = s \\ \gamma\lambda e_{t-1}^c(s), \text{ otherwise} \end{cases} \\
&p(s,a) \leftarrow p(s,a) + \alpha_a\, RPE_t\, e_t^a(s,a) \quad \forall\, s \in \mathcal{S}, a \in \mathcal{A} \\
&e_t^a(s,a) = \begin{cases} 1 - \pi(s,a), \text{ if } s_t = s, a_t = a \\ \gamma\lambda e_{t-1}^a(s,a), \quad \text{otherwise,} \end{cases}
\end{aligned}
\tag{4.19}
$$

where $e_t^c(s)$ and $e_t^a(s,a)$ are exponentially decaying eligibility traces for the critic and the actor respectively, initialized to zero, α_c and α_a are the learning rates of critic and actor respectively, and $p(s,a)$ signifies the preference for action a when in state s.

REINFORCE. The REINFORCE algorithm (Williams, 1992) estimates the policy parameters of all the preceding within-episode decisions directly with gradient ascent using the return, in a model-free manner.

$$
\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\, \nabla \log \pi(s_t, a)\, G_t, \quad \forall\, a \in \mathcal{A}, \tag{4.20}
$$

where $\boldsymbol{\theta}$ is the policy parameter vector, G_t is the (episodic) return $G_t = \sum_{k=1}^{K-t} \gamma^{k-1} r_{t+k}$, with K the total episode length, and $\pi(s,a)$ is the policy of a state-action pair (s,a). We consider a softmax policy $\pi(s,a) = e^{\boldsymbol{\phi}(s,a)'\boldsymbol{\theta}/\tau} / \sum_b e^{\boldsymbol{\phi}(s,b)'\boldsymbol{\theta}(s,b)/\tau}$, with temperature parameter τ, where $\boldsymbol{\phi}(s,a)$ is the feature vector for the state action pair (s,a) and has the same dimensionality as the vector $\boldsymbol{\theta}$, and $'$ stands for transpose. In our tabular setting, the dimension of $\boldsymbol{\theta}$ equals the number of states and actions and $\boldsymbol{\phi}(s,a)$ is a one-hot feature vector (1 for current (s,a), 0 otherwise).

Expanding the previous formula we have

$$
\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \left(\boldsymbol{\phi}(s_t,a)' - \sum_b \pi(s_t,b)\boldsymbol{\phi}(s_t,b)'\right) G_t, \quad \forall a \in \mathcal{A}. \tag{4.21}
$$

In our task, there are two possible actions at each state. If we denote the selected action from a state s_t as $a^{(sel)}$ and the alternative non-selected as $a^{(alt)}$, the update takes the following form

$$
\begin{aligned}
&\boldsymbol{\theta}(s_t, a^{(sel)}) \leftarrow \boldsymbol{\theta}(s_t, a^{(sel)}) + \alpha \left(1 - \pi(s_t, a^{(sel)})\right) G_t, \\
&\boldsymbol{\theta}(s_t, a^{(alt)}) \leftarrow \boldsymbol{\theta}(s_t, a^{(alt)}) - \alpha \left(1 - \pi(s_t, a^{(sel)})\right) G_t.
\end{aligned}
\tag{4.22}
$$

That is, at each step both the selected and the non-selected action are updated. The above update occurs at the end of each episode backwards in time, for all $t = 1, ..., K-1$ steps within the episode.

Surprise SARSA-λ continuous, Surprise Actor-critic continuous, Surprise REINFORCE continuous. We introduce surprise modulation of the model-free learning rate. The algorithms we denote with the suffix "continuous" are same as their original version, but the learning rate α is replaced by the surprise-modulated learning rate $\alpha_{\mathbf{S}_{\mathrm{BF}}}$ at each time point t as follows

$$\alpha^{(t)}_{\mathbf{S}_{\mathrm{BF}}} = \alpha \cdot (1 - \gamma^{(t)}_{\mathbf{S}_{\mathrm{BF}}}), \tag{4.23}$$

where $\gamma^{(t)}_{\mathbf{S}_{\mathrm{BF}}}$ the surprise-modulated adaptation rate of Equation 4.3. This means that whenever a transition is surprising and $\gamma^{(t)}_{\mathbf{S}_{\mathrm{BF}}}$ gets high values, then its effect on the (model-free) value update is reduced. For Surprise Actor-critic continuous this modulation is applied to the learning rate of the actor, i.e. α_a. For REINFORCE this modulation is applied on the backward update steps upon the receipt of the reward at the end of the episode.

Surprise SARSA-λ binary, Surprise Actor-critic binary, Surprise REINFORCE binary. The algorithms we denote with the suffix "binary" are same as their original version, but the learning rate α is replaced by the surprise-modulated learning rate $\alpha_{\mathbf{S}_{\mathrm{BF}}}$ at each time point t as follows

$$\alpha^{(t)}_{\mathbf{S}_{\mathrm{BF}}} = \alpha \cdot (1 - \lfloor \mathbb{E}_{\hat{\mathbb{b}}^{(t)}}[Z^{sa}_t] \rceil), \tag{4.24}$$

where $\lfloor \cdot \rceil$ stands for rounding to the nearest integer and $\mathbb{E}_{\hat{\mathbb{b}}^{(t)}}[Z^{sa}_t] = \sum_{i=1}^{N} w^{sa,(i)}_t z^{sa,(i)}_t$ is the estimated hidden state by the particle filter. This means that whenever a transition is estimated to correspond to a surprise trial, it is completely omitted from the (model-free) value update. As for the continuous version, for Surprise Actor-critic binary this modulation is applied to the learning rate of the actor, i.e. α_a, and for REINFORCE it is applied at the backward step of policy updates.

Hybrid Actor-critic. We implemented a weighted sum of Actor-critic and Forward Learner as follows

$$Q(s,a) = w_{MB}\, Q_{MB}(s,a) + w_{MF}\, p_{MF}(s,a), \quad \forall\, s \in \mathcal{S}, a \in \mathcal{A}, \tag{4.25}$$

where Q_{MB} are the values of the model-based Forward Learner and p_{MF} are the policy parameters of the model-free Actor-critic. Since the policy parameters can take values larger than 1, we use separate weights w_{MB} and w_{MF} instead of the convex sum of the Hybrid Learner of (Daw et al., 2011a; Gläscher et al., 2010). We did not include time dependency of the weights, in order to keep the algorithms simpler, in particular since in the resulting fit of Hybrid Learner we observed that the contribution of the decay slope was very small.

Prioritized Sweeping with particle filtering. Prioritized Sweeping was described

in Chapter 3 (subsection 3.2.3). Briefly, instead of computing the Bellman equation $Q_{MB}(s,a) = \sum_{s'} \hat{T}(s,a,s')\big[\bar{R}(s,a,s') + \gamma V^{\pi}(s')\big],\ \forall\ s \in \mathcal{S}, a \in \mathcal{A}, s' \in \mathcal{S}$ recursively until equilibrium (i.e. value iteration), Prioritized Sweeping (Moore and Atkeson, 1993; Van Seijen and Sutton, 2013) performs a number of update cycles, where only the predecessors of states that change "a lot" are updated. Here, we estimate the transition matrix T with particle filtering, instead of Perfect Integration (which is typically used in standard Prioritized Sweeping). Our implementation is similar to the one described in Chapter 3 (see Algorithm 4 for pseudocode). The only difference is in the update rule of the particle filtering, that involves a trade-off between integrating and ignoring (see subsection 4.4.3), rather than integrating and resetting (see subsection 3.2.2).

Hybrid-λ-PS-pf, Surprise Hybrid-λ-PS-pf continuous and Surprise Hybrid-λ-PS-pf binary.
These three algorithms implement a weighted sum of the following form

$$Q(s,a) = w_{MB}\, Q_{MB}(s,a) + w_{MF}\, Q_{MF}(s,a), \quad \forall\ s \in \mathcal{S}, a \in \mathcal{A}, \tag{4.26}$$

where Q_{MB} are the values of the model-based Prioritized Sweeping with particle filtering. Q_{MF} are the values of SARSA-λ for Hybrid-λ-PS-pf, of Surprise SARSA-λ continuous for Surprise Hybrid-λ-PS-pf continuous, and of Surprise SARSA-λ binary in Surprise Hybrid-λ-PS-pf binary. Thus, in the last two algorithms surprise modulation acts both on the model-free and on the model-based system.

Policy. All algorithms were used in combination with a softmax action selection policy $\pi(s,a) = e^{f(s,a)/\tau} / \sum_b e^{f(s,b)/\tau}$, with temperature parameter τ, where $f(s,a) \in \{Q(s,a),\ p(s,a),\ \boldsymbol{\phi}(s,a)'\boldsymbol{\theta}(s,a)\}$ depending on the algorithm. $\boldsymbol{\phi}$ is a one-hot feature vector (1 for current (s,a), 0 otherwise) used at the REINFORCE algorithm. For the hybrid algorithms that exhibit a non-convex sum, to avoid over-parametrization we set $\tau = 1$ and the effect of the temperature is included in the hybrid weights, which can take any possible value and are not restricted in $[0,1]$.

In Table 4.1 we provide for each algorithm the number of free parameters that are fitted to data and the obtained log-evidence (mean cross-validated maximum log-likelihood), and the standard error of the estimated log-evidence across the 5 optimization rounds.

4.4.5 Parameter fit and model selection

Each of the algorithms we consider includes a set of free parameters Θ. In order to find the parameter values that explain the participant's actions best, we use the Metropolis-Hastings Markov Chain Monte Carlo (MCMC) algorithm (Hastings, 1970) to approximate the posterior distribution $\mathbf{P}(\Theta|D)$, where D the data of all participants, similar to Lehmann et al. (2019).

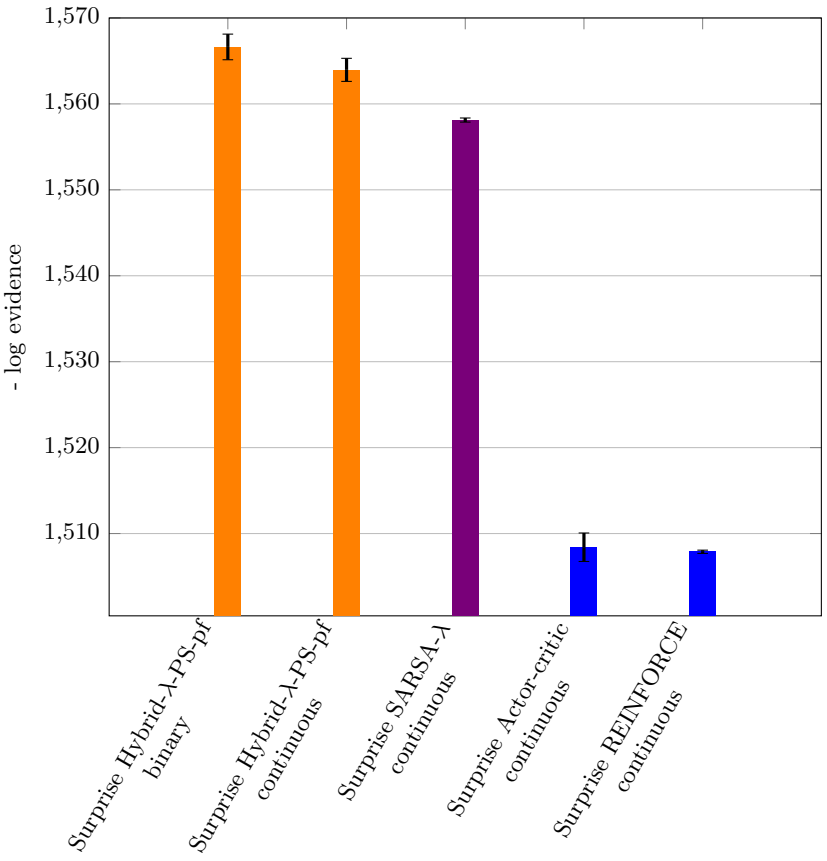


Figure 4.6 – **Algorithm fit to behaviour - Supplementary Figure.** Negative model log-evidence for additional algorithms we considered. Smaller values indicate better performance. The Surprise Actor-critic continuous, and the Surprise REINFORCE continuous are likely models for behavior and reach similar fit levels to the rest policy learning algorithms we considered (see Fig. 4.2B). *Abbreviations:* PS: Prioritized Sweeping, pf: Particle Filtering.

Algorithm	**#Param**	**Cross-validated LL mean**	**Cross-validated LL std error**
Actor-critic	6	−1,504.98	0.15
Surprise Actor-critic binary	9	−1,507.03	0.4
REINFORCE	3	−1,507.31	$8.77 \cdot 10^{-3}$
Hybrid Actor-critic	8	−1,507.44	0.4
Surprise REINFORCE continuous	6	−1,507.88	0.19
Surprise Actor-critic continuous	9	−1,508.41	1.66
Surprise REINFORCE binary	6	−1,510.12	0.34
SARSA-λ	4	−1,557.97	$9.18 \cdot 10^{-2}$
Surprise SARSA-λ continuous	7	−1,558.12	0.24
Surprise Hybrid-λ-PS-pf continuous	9	−1,563.96	1.34
Surprise SARSA-λ binary	7	−1,564.45	0.94
Hybrid Learner - λ	7	−1,565.28	1.1
Surprise Hybrid-λ-PS-pf binary	9	−1,566.63	1.49
Hybrid-λ-PS-pf	9	−1,568.49	1.25
Hybrid Learner - 0	6	−1,605.96	0.15
Prioritized Sweeping with particle filtering	6	−1,965.18	1.93
Forward Learner	3	−1,967.76	0.14
Biased Random Walk	1	−2,485.64	0

Table 4.1 – **Learning algorithms and their corresponding performance in explaining the behavioral data.** Abbreviations and notations: #Param: number of parameters; Cross-validated LL mean: Cross-validated maximum log-likelihood, averaged across 5 runs of a 3-fold cross-validation procedure. We use the mean cross-validated LL as an approximation of the model log-evidence; Cross-validated LL std error: Standard error of the mean for the cross-validated maximum log-likelihood across 5 runs of a 3-fold cross-validation procedure.

We have

$$\mathbf{P}(\Theta|D) = \mathbf{P}(D|\Theta)\mathbf{P}(\Theta)/\mathbf{P}(D). \tag{4.27}$$

We use uniform prior $\mathbf{P}(\Theta)$ and we sample from the likelihood $\mathbf{P}(D|\Theta)$. The likelihood is the joint probability of all participants' actions for a set of parameter values $\Theta = \theta$, i.e. for the log-likelihood (LL) of the parameters θ we have

$$LL(\theta; D) = \log[\mathbf{P}(D|\theta)] = \sum_{n=1}^{N}\sum_{t=1}^{T} log[\pi(s_{n,t}, a_{n,t})], \tag{4.28}$$

where $n = \{1, ..., N\}$ the participant id, and $t = \{1, ..., T\}$ the trials (time steps) performed by each participant.

Within a single MCMC run we perform 50 repetitions (i.e. 50 random starting points in

the parameter space). For each repetition we collect 100,000 parameter samples, with a burn-in (i.e. discarding) of the first 1500 samples and via keeping only every 10th sample. At the end of the run we register the parameter values θ^* that maximize the LL and the corresponding LL value.

In order to select the model that explains the data best, we performed cross-validation in 3 folds. At each fold we leave 7 participants out of the fitting procedure and we estimate the algorithms' parameters on the data of the remaining participants (i.e. at each fold we perform one MCMC run as described above). With the obtained parameter values we then assess the goodness of fit on the left-out participants, by calculating the LL on these unseen participants. At the end of a 3-fold cross-validation round, we sum, for each algorithm, the out-of-sample LL of the 3 folds.

The MCMC procedure includes some randomness, due to random staring points and random moves in the parameter space. In order to deal with this source of noise and to make more informed conclusions about model selection, we repeated the cross-validation rounds 5 times, for each algorithm. At the end of this procedure we obtain the mean sum of out-of-sample LL and its standard error across the 5 rounds. We consider the mean sum of out-of-sample LL as an approximation of the log-evidence. The penalty for high complexity comes naturally through cross-validation and the algorithm with the highest log-evidence is the winning model. This procedure for model selection is similar to methods used in (Berger and Pericchi, 1996; Fong and Holmes, 2020; Rust and Schmittlein, 1985; Wang and Pericchi, 2020). Furthermore, its theoretical foundations, as well as extensions thereof, are ongoing work led by Alireza Modirshanechi (LCN, EPFL).

4.4.6 fMRI data acquisition and preprocessing

We acquired functional data of 23 participants (11 female) on a 3T Siemens Prisma MRI Scanner, using a T2*-weighted 2D echo planar imaging (EPI) sequence (442 volumes, 34 slices/volume, slice thickness of 2.5 mm, 20% interslice gap, repetition time 2720 ms, flip angle 90°, matrix size 64x64, field of view 192 mm^2). We acquired three echo images following each radio-frequency excitation (echo times = 17.4 s, 35.2 s, 53 s) in order to achieve optimal BOLD sensitivity in all brain regions (Poser et al., 2006). Slices were tilted by -20° off the line connecting the anterior-posterior commissure. Brain coverage included the orbitofrontal cortex and subcortical structures and excluded some posterior-superior frontal and parietal regions. We acquired structural T1-weighted MPRAGE images for co-registration of the fMRI data. We used B0-field maps, obtained from double-echo FLASH acquisitions (64 slices; matrix size 64×64, spatial resolution 3 mm; short echo time 10 ms, long echo time 12.46 ms; repetition time 1020 ms) to correct the EPI images for distortions along the phase-encode direction (Hutton et al., 2002).

For the analysis of the fMRI images we used the SPM12 software (preprocessing and

and first level statistical analysis), the SnPM13 (second level statistical analysis) and the Nilearn software (plotting utilities). The three echo images were added to form the final functional images. Then the images were realigned, spatially normalized to standard Montreal Neurological Institute coordinates and smoothed with a Gaussian kernel of 8 mm (Full width at half maximum - FWHM). Two participants were excluded from the analysis, one due to high degree of movement artifacts in his brain images and one due to performance (completion of less than half of the number of episodes that other participants performed on average).

4.4.7 fMRI data statistical analysis

After fitting the algorithms to the behavioral data we compute for each participants trial-by-trial learning signals and use them as regressors against the fMRI data in a general linear model (GLM). For this step, we used the population parameters resulting from fitting all participants together, as it is usually done and recommended in the analysis of fMRI data (Daw et al., 2011b). We included four regressors in the model: (i) one regressor for the intervals during which a state was on the screen (boxcars events), (ii) the SPE calculated with the Hybrid Actor-critic, (iii) the RPE calculated with the Hybrid Actor-critic, and (iv) one regressor for participants' actions (zero-duration events). The SPE and RPE regressors were placed at the time of the states (as their parametric modulators). They were othogonalized with respect to the states, but not with respect to each other, to ensure that any shared variance is not assigned to one or the other. All regressors were convolved with the canonical hemodynamic response function (HRF) and its derivatives, apart from the action regressor which was included to control for button presses, rather than brain activity (note that the states stay on the screen until the participant presses a button to select an action). To control for remaining motion artifacts the six rigid-body realignment motion parameters were included in the model. The estimated regression coefficients for each of the regressors from each participant were taken to random effects group level analysis (one-sample t-test). For the statistical analysis at the group level we perform nonparametric permutation testing and we controlled for multiple comparisons (whole brain family-wise error rate FWER ≤ 0.05) using the maximum statistic (Nichols and Holmes, 2002). We included the whole brain in our analysis and we did not focus on a-priori selected regions of interest.

4.5 Supplementary Material

4.5.1 Reaction times are longer for surprise trials

The participants' reaction time serves as a behavioural signature of surprise (Huettel et al., 2002; Meyniel et al., 2016). Fig. 4.7 shows the average reaction time across participants for surprise and non-surprise trials. The reaction time following a surprising transition

State id	Action id	
	1	**2**
1	-	-
2	2	1
3	1	3
4	3	0
5	1	1
6	0	2
7	2	2

State id	Action id	
	1	**2**
1	-	-
2	0	1
3	0	2
4	2	1
5	3	1
6	2	1
7	2	2

Table 4.2 – **Distance to goal at task graphs.** We used two underlying task graphs, randomized across participants. Starting from a given state the two actions lead to new states at different distances from the goal. These distances are noted in the tables, for each of the two graphs. For example, from the state with id 4 of the first task graph, one action leads to a state that is 3 steps away from the goal and the other state leads directly to the goal ("0"). The state with id 1 is the goal state. The locations of the states on the screen, their associated images, and the assignment of action ids to left or right were randomized across participants.

is significantly higher (unequal variance two-sample t-test, p=0.03), indicating that participants do notice the unexpected transitions. Following a surprise trial, participants need to leverage their knowledge and choose their next action from the unexpected landing state. This cognitive process is presumably reflected in longer reaction times after a surprising transition.

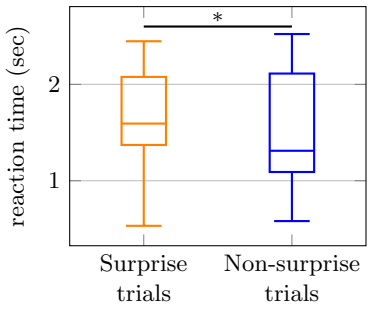


Figure 4.7 – **Reaction time on surprise trials.** Mean reaction times on surprise trials (orange) and on non-surprise trials (blue). The boxplots are made by the mean reaction times of participants across trials of each type (21 values each). The reaction time on surprise trials is significantly higher than on non-surprise trials (unequal variance, two-sample t-test, p=0.03).

4.5.2 Brain activation statistical results

State Prediction Error (SPE)			
Brain region	x, y, z	t	p
left supramarginal gyrus	-36, -46, 38	7.80	0.0005
right middle frontal gyrus	45, 56, 2	8.01	0.0001
left middle frontal gyrus	-42, 53, -1	6.19	0.01
superior frontal gyrus medial segment	6, 26, 44	7.26	0.0013
left Supplementary motor area (SMA)	-6, 20, 44	7.04	0.0019
right angular gyrus	30, -67, 35	5.8	0.0199
Reward Prediction Error (RPE)			
Brain region	x, y, z	t	p
right anterior cingulate gyrus	-6, 26, -7	6.83	0.0007
superior frontal gyrus medial segment	3, 44, -10	6.16	0.0022
left parahippocampal gyrus	-30, -10, -31	6.86	0.0007
left occipital gyrus	-42, -70, -7	6.40	0.0014
right occipital gyrus	45, -79, -10	4.99	0.0269

Table 4.3 – **Brain activation statistical results.** Brain x, y, z coordinates in Montreal Neurological Institute (MNI) coordinate space of the peak activations, with their corresponding t- and p-values. We performed random effects group level analysis (21 participants, one-sample t-test) with nonparametric permutation testing (SnPM13) and we controlled for multiple comparisons for whole brain family-wise error rate FWER $\leq$ 0.05 using the maximum statistic approach (Nichols and Holmes, 2002).

4.5.3 Derivation of Bayesian inference on the generative model

We derive here the recursive formula to update the belief $\mathbb{b}^{sa,(t)}(p^{sa}) \equiv \mathbf{P}(P^{sa} = p^{sa}|s_{1:t})$ to the new belief $\mathbb{b}^{sa,(t+1)}(p^{sa})$. To simplify notation we will skip the superscript sa in this section, apart from the parameters $P^{sa} = p^{sa}$.

Exploiting the Markov property of the generative model we have

$$\mathbb{b}^{(t+1)}(p^{sa}) = \frac{\mathbf{P}(s_{t+1}|p^{sa})\mathbf{P}(P^{sa} = p^{sa}|s_{1:t})}{\mathbf{P}(s_{t+1}|s_{1:t})}\,. \tag{4.29}$$

The second factor in the numerator is by definition equal to $\mathbb{b}^{(t)}(p^{sa})$.

For the first factor in the numerator we have

$$\begin{aligned}\mathbf{P}(s_{t+1}|p^{sa}) &= \sum_{k\in\{0,1\}} \mathbf{P}(s_{t+1}|p^{sa}, Z_{t+1} = k)\mathbf{P}(Z_{t+1} = k|p^{sa}) \\ &= (1 - p_j)P(s_{t+1}|p^{sa}) + p_j U(s_{t+1}).\end{aligned} \tag{4.30}$$

The denominator is equal to the marginalization of the numerator over P^{sa}, thus

$$\begin{aligned}\mathbf{P}(s_{t+1}|s_{1:t}) &= \int \mathbf{P}(s_{t+1}|p^{sa})\mathbb{b}^{(t)}(p^{sa})dp^{sa} \\ &= (1-p_j)\int P(s_{t+1}|p^{sa})\mathbb{b}^{(t)}(p^{sa})dp^{sa} + p_j\int U(s_{t+1})\mathbb{b}^{(t)}(p^{sa})dp^{sa} \\ &= (1-p_j)P(s_{t+1};\mathbb{b}^{(t)}) + p_jU(s_{t+1}),\end{aligned} \tag{4.31}$$

where $P(s_{t+1};\mathbb{b}^{(t)}) = \int P(s_{t+1}|p^{sa})\mathbb{b}^{(t)}(p^{sa})dp^{sa}$ is the probability of the observation $S_{t+1} = s_{t+1}$ under the current belief $\mathbb{b}^{(t)}(p^{sa})$, as defined in Equation 2.6.

Combining all the above, we have for the updated belief

$$\begin{aligned}\mathbb{b}^{(t+1)}(p^{sa}) &= \frac{\Big((1-p_j)P(s_{t+1}|p^{sa}) + p_jU(s_{t+1})\Big)\mathbb{b}^{(t)}(p^{sa})}{(1-p_j)P(s_{t+1};\mathbb{b}^{(t)}) + p_jU(s_{t+1})} \\ &= \frac{\frac{P(s_{t+1}|p^{sa})\mathbb{b}^{(t)}(p^{sa})}{P(s_{t+1};\mathbb{b}^{(t)})} + \frac{p_j}{1-p_j}\frac{U(s_{t+1})}{P(s_{t+1};\mathbb{b}^{(t)})}\mathbb{b}^{(t)}(p^{sa})}{1+\frac{p_j}{1-p_j}\frac{U(s_{t+1})}{P(s_{t+1};\mathbb{b}^{(t)})}}.\end{aligned} \tag{4.32}$$

We define

$$\mathbb{b}_B^{(t+1)}(p^{sa}) = \frac{P(s_{t+1}|p^{sa})\mathbb{b}^{(t)}(p^{sa})}{P(s_{t+1};\mathbb{b}^{(t)})} \tag{4.33}$$

the belief after an exact Bayesian update, i.e. the incorporation of the new observation to the current belief using Bayes' rule, as in Equation 2.8.

We define as surprise $\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{(t)})$

$$\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{(t)}) = \frac{U(s_{t+1})}{P(s_{t+1};\mathbb{b}^{(t)})}, \tag{4.34}$$

i.e. the ratio of the uniform likelihood relative to the likelihood of the observation under the current belief. Then we have for the updated belief

$$\begin{aligned}\mathbb{b}^{(t+1)}(p^{sa}) &= \frac{\mathbb{b}_B^{(t+1)}(p^{sa}) + \frac{p_j}{1-p_j}\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{(t)})\mathbb{b}^{(t)}(p^{sa})}{1+\frac{p_j}{1-p_j}\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{(t)})} \\ &= (1-\gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)})\mathbb{b}_B^{(t+1)}(p^{sa}) + \gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)}\mathbb{b}^{(t)}(p^{sa}),\end{aligned} \tag{4.35}$$

with $m = \frac{p_j}{1-p_j}$ and

$$\gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)} = \frac{m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{(t)})}{1+m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\mathbb{b}^{(t)})} \tag{4.36}$$

the surprise-modulated adaptation rate. The resulting belief $\mathbb{b}^{(t+1)}(p^{sa})$ is a weighted

average between integrating the new observation to the current belief and maintaining the current belief (i.e. ignoring the new observation).

4.5.4 Derivation of Particle Filtering

The difference here from a standard derivation (Särkkä, 2013) is the absence of the Markov property of conditional observations (i.e. $P(s_{t+1}|z_{1:t+1}, s_{1:t}) \neq P(s_{t+1}|z_{t+1})$). Our goal is to approximate

$$\mathbf{P}(z_{1:t+1}|s_{1:t+1}) \approx \sum_{i=1}^{N} w_{t+1}^{(i)} \delta(z_{1:t+1} - z_{1:t+1}^{(i)}) \,, \tag{4.37}$$

where we skip the subscripts sa in the section as well, for simplicity.

Using a proposal distribution Ψ for particle i at time $t+1$ we have

$$\begin{aligned} w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(z_{1:t+1}^{(i)}|s_{1:t+1})}{\Psi(z_{1:t+1}^{(i)}|s_{1:t+1})} \propto \frac{\mathbf{P}(z_{1:t+1}^{(i)}, s_{t+1}|s_{1:t})}{\Psi(z_{1:t+1}^{(i)}|s_{1:t+1})} \\ w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(s_{t+1}, z_{t+1}^{(i)}|z_{1:t+1}^{(i)}, s_{1:t})\mathbf{P}(z_{1:t}^{(i)}|s_{1:t})}{\Psi(z_{t+1}^{(i)}|z_{1:t}^{(i)}, s_{1:t+1})\Psi(z_{1:t}^{(i)}|s_{1:t})} \,. \end{aligned} \tag{4.38}$$

Note that $w_t^{(i)} \propto \frac{\mathbf{P}(z_{1:t}^{(i)}|s_{1:t})}{\Psi(z_{1:t}^{(i)}|s_{1:t})}$ are the weights calculated at the previous time step.

Therefore

$$w_{t+1}^{(i)} \propto \frac{\mathbf{P}(s_{t+1}, z_{t+1}^{(i)}|z_{1:t+1}^{(i)}, s_{1:t})}{\Psi(z_{t+1}^{(i)}|z_{1:t}^{(i)}, s_{1:t+1})} w_t^{(i)} \,. \tag{4.39}$$

We use the optimal proposal function in terms of variance of the weights (Doucet et al., 2000)

$$\Psi(z_{t+1}^{(i)}|z_{1:t}^{(i)}, s_{1:t+1}) = \mathbf{P}(z_{t+1}^{(i)}|z_{1:t}^{(i)}, s_{1:t+1}) \,. \tag{4.40}$$

Using Bayes's rule we, thus, have

$$\begin{aligned} w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(s_{t+1}, z_{t+1}^{(i)}|z_{1:t+1}^{(i)}, s_{1:t})}{\mathbf{P}(z_{t+1}^{(i)}|z_{1:t}^{(i)}, s_{1:t+1})} w_t^{(i)} = \mathbf{P}(s_{t+1}|z_{1:t}^{(i)}, s_{1:t}) w_t^{(i)} \\ &\propto \Big((1-p_j)\mathbf{P}(s_{t+1}|z_{1:t}^{(i)}, s_{1:t}, z_{t+1}^{(i)} = 0) + p_j \mathbf{P}(s_{t+1}|z_{1:t}^{(i)}, s_{1:t}, z_{t+1}^{(i)} = 1)\Big) w_t^{(i)}. \end{aligned} \tag{4.41}$$

We have $\mathbf{P}(s_{t+1}|z_{1:t}^{(i)}, s_{1:t}, z_{t+1}^{(i)} = 0) = P(s_{t+1}; \hat{\mathbb{b}}_i^{(t)})$ (see definition of Equation 2.6), and

$\mathbf{P}(s_{t+1}|z_{1:t}^{(i)}, s_{1:t}, z_{t+1}^{(i)} = 1) = U(s_{t+1})$.

Therefore the normalized weights are

$$w_{t+1}^{(i)}\Big((1-p_j)P(s_{t+1};\hat{\mathbb{b}}_i^{(t)}) + p_j U(s_{t+1})\Big)w_t^{(i)}/Z, \tag{4.42}$$

where Z the normalization factor

$$\begin{aligned} Z &= \sum_{i=1}^{N}\left[(1-p_j)P(s_{t+1};\hat{\mathbb{b}}_i^{(t)}) + p_j U(s_{t+1})\right]w_t^{(i)} \\ &= (1-p_j)\sum_{i=1}^{N} w_t^{(i)} P(s_{t+1};\hat{\mathbb{b}}_i^{(t)}) + p_j U(s_{t+1})\sum_{i=1}^{N} w_t^{(i)}. \end{aligned} \tag{4.43}$$

The weights from the previous time step are normalized, i.e. $\sum_{i=1}^{N} w_t^{(i)} = 1$, and $\sum_{i=1}^{N} w_t^{(i)} P(s_{t+1};\hat{\mathbb{b}}_i^{(t)}) = P(s_{t+1};\hat{\mathbb{b}}^{(t)})$, thus

$$Z = (1-p_j)P(s_{t+1};\hat{\mathbb{b}}^{(t)}) + p_j U(s_{t+1}). \tag{4.44}$$

We define as

$$w_{B,t+1}^{(i)} = \frac{P(s_{t+1};\hat{\mathbb{b}}_i^{(t)})}{P(s_{t+1};\hat{\mathbb{b}}^{(t)})} w_t^{(i)} \tag{4.45}$$

the weights that correspond to the incorporation of the new observation, i.e. to a Bayesian update $\mathbb{b}_B^{(t+1)}$ (Equation 2.8). With $m = \frac{p_j}{1-p_j}$ and $\mathbf{S}_{\mathrm{BF}}(s_{t+1};\hat{\mathbb{b}}^{(t)}) = \frac{U(s_{t+1})}{P(s_{t+1};\hat{\mathbb{b}}^{(t)})}$ we can find

$$w_{t+1}^{(i)} = (1-\gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)})w_{B,t+1}^{(i)} + \gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)} w_t^{(i)}\,, \tag{4.46}$$

where $\gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1)} = \frac{m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\hat{\mathbb{b}}^{(t)})}{1+m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\hat{\mathbb{b}}^{(t)})}$ the surprise-modulated adaptation rate, same as we saw in Equation 2.9.

We can see that the weight update rule exhibits the same surprise-modulated trade-off between incorporating the new observation to the belief and ignoring it.

At each time step we sample the new state $z_{t+1}^{(i)}$ of each particle i from the proposal

distribution. After a few steps, it can be shown that the probability for a surprise trial is

$$\begin{aligned}\Psi(z_{t+1}^{(i)}=1|z_{1:t}^{(i)},s_{1:t+1}) &= \frac{p_j U(s_{t+1})}{(1-p_j)P(s_{t+1};\hat{\mathbb{b}}_i^{(t)})+p_j U(s_{t+1})} \\ &= \frac{m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\hat{\mathbb{b}}_i^{(t)})}{1+m\mathbf{S}_{\mathrm{BF}}(s_{t+1};\hat{\mathbb{b}}_i^{(t)})} \\ &= \gamma_{\mathbf{S}_{\mathrm{BF}}}^{(t+1),(i)},\end{aligned} \tag{4.47}$$

where $\mathbf{S}_{\mathrm{BF}}(s_{t+1};\hat{\mathbb{b}}_i^{(t)}) = \frac{U(s_{t+1})}{P(s_{t+1};\hat{\mathbb{b}}_i^{(t)})}$ the surprise of observing s_{t+1} under the belief $\hat{\mathbb{b}}_i^{(t)}$ of particle i. This means that the more surprising the observation s_{t+1} the more likely it will be for the particle's hidden state to get the value $z_{t+1}^{(i)}=1$, i.e. to signal the occurrence of a surprise trial.

Finally, the approximated belief of a particle i is

$$\hat{\mathbb{b}}_i^{(t+1)}(p^{sa}) = \sum_{i=1}^{N} w_{t+1}^{(i)}\mathbf{P}(p^{sa}|z_{1:t+1}^{(i)},s_{1:t+1}) \quad = \sum_{i=1}^{N} w_{t+1}^{(i)}\mathbf{P}(p^{sa}|s_{m_{1:t+1}^{(i)}}), \tag{4.48}$$

where we define as $m_{[1:t+1]}^{(i)}$ the time points of non-surprise trials within the time window $t'=1$ to $t'=t+1$ for particle i, i.e $m_{[1:t]}^{(i)} = \{j \in \{1,...,t\}|z_j^{(i)}=0\}$.

In order to avoid the problem of degeneracy of the weights we implemented the Sequential Importance Resampling algorithm (Doucet et al., 2000; Gordon et al., 1993), where the particles are resampled when their effective number falls below a threshold. The effective number of the particles can be computed as in (Doucet et al., 2000), (Särkkä, 2013)

$$N_{\mathrm{eff}} \approx \frac{1}{\sum_{i=1}^{N}(w_t^{(i)})^2}\,. \tag{4.49}$$

When N_{eff} is below a critical threshold, the particles are resampled with replacement from the Categorical distribution defined by their weights, and then all the weights are set to $w_t^{(i)} = 1/N$. We performed resampling when $N_{\mathrm{eff}} \leq N/2$, following Doucet and Johansen (2009), and we did not optimize the parameter N_{eff}.

4.5.5 Relationship between SPE and $\mathbf{S}_{\mathrm{BF}}$

The SPE (Daw et al., 2011a; Gläscher et al., 2010) for an observed state s_{t+1} from a state-action pair (s,a) at time $t+1$ is defined as

$$SPE_{t+1} = 1 - P(s_{t+1};\mathbb{b}^{sa,(t)}(p^{sa})), \tag{4.50}$$

where $\mathbb{b}^{sa,(t)}(p^{sa}) = \mathbf{P}(p^{sa}|s_{1:t})$ is the participant's belief at time t.

On the other hand, the Bayes Factor Surprise $\mathbf{S}_{\mathrm{BF}}$ is defined as

$$\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \mathbb{b}^{sa,(t)}(p^{sa})) = \frac{U(s_{t+1})}{P(s_{t+1}; \mathbb{b}^{sa,(t)}(p^{sa}))}. \tag{4.51}$$

If the expected transition probabilities under U are uniform (i.e. same σ for all elements of the transition probability vector), then $P(s_{t+1}; \mathbb{b}^{sa,(0)}(p^{sa}))$ is a constant C. If the *same belief* is used for computation of both $\mathbf{S}_{\mathrm{BF}}$ and SPE, then

$$SPE_{t+1} = 1 - \frac{C}{\mathbf{S}_{\mathrm{BF}}(s_{t+1}; \mathbb{b}^{sa,(t)}(p^{sa}))}. \tag{4.52}$$

However, two different learning algorithms lead to different beliefs at each time step, and hence, there is no clear connection between the SPE of an algorithm and the $\mathbf{S}_{\mathrm{BF}}$ (or even the SPE) of another.

Another important influence on how much the timelines of SPE and $\mathbf{S}_{\mathrm{BF}}$ differ, comes from the assumed prior stochasticity σ of the environment. The lower the σ, the more "spiky" $\mathbf{S}_{\mathrm{BF}}$ will be. In other words, the more deterministic the environment is assumed to be, the more surprising an observation that lies far from current belief will be. For the SPE, we followed the implementation of Daw et al. (2011a); Gläscher et al. (2010) - see Supplementary Material subsection 4.4.4 for the algorithm - where the SPE mediates a simple delta-rule and there is no free parameter for the stochasticity of the environment. The SPE-mediated algorithms can be interpreted as maintaining the assumption that the environment is fairly stochastic, i.e. $\sigma = 1$, and approximating the transition probabilities with the mode of the posterior belief. We empirically found that the closer to 1 the σ is, the more similar the SPE and $\mathbf{S}_{\mathrm{BF}}$ signals are.

Fig. 4.8 depicts the SPE from Hybrid Actor-critic and the $\mathbf{S}_{\mathrm{BF}}$ from Surprise Actor-critic binary for two representative participants, in time and with respect to each other. Note, that the resulting value for the fitted stochasticity σ for Surprise Actor-critic binary was 2.35, meaning that the environment was perceived as stochastic by most subjects. We can empirically see that $\mathbf{S}_{\mathrm{BF}}$ is approximately an increasing function of the SPE. Hybrid Actor-critic integrates all observations, whereas Surprise Actor-critic binary ignores the more surprising ones. Therefore, as time passes by, the beliefs of the two learning algorithms grow different, which leads to larger differences between the two learning signals towards the end of the experiment.

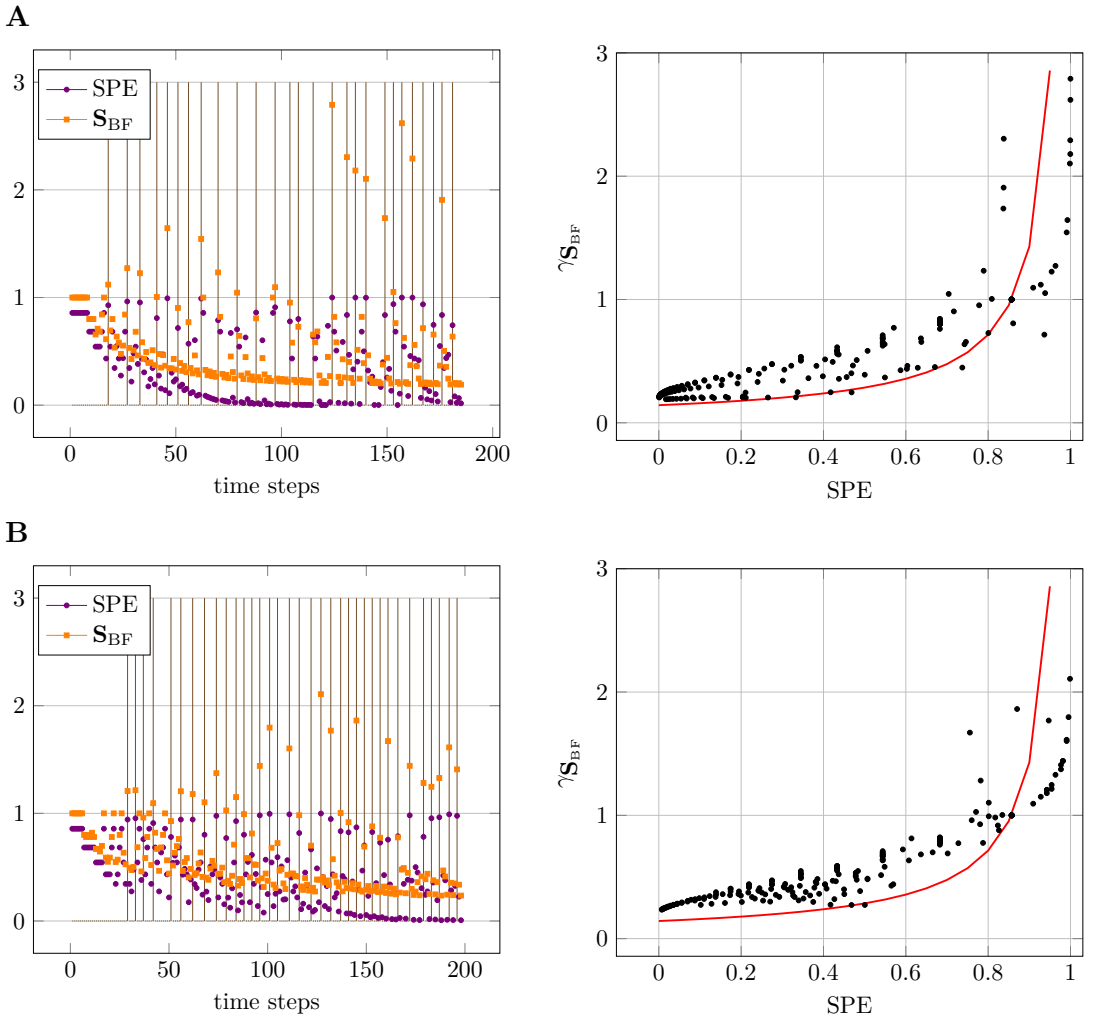


Figure 4.8 – **SPE versus $\mathbf{S}_{\mathrm{BF}}$.** SPE and $\mathbf{S}_{\mathrm{BF}}$ signals of Hybrid Actor-critic and Surprise Actor-critic binary respectively, for two representative participants (**A** and **B**). The graphs on the left column depict the evolution of the two signals in time. The graphs on the right side show the two signals with respect to each other for any given time point. The theoretical curve corresponding to Equation 4.52, with $C = 1/|\mathcal{S}|$, where $|\mathcal{S}| = 7$ the number of states in our task, is plotted in red. We can empirically see that $\mathbf{S}_{\mathrm{BF}}$ is approximately an increasing function of the SPE.

4.6 Acknowledgements

The experiments were performed at the Laboratoire de recherche en neuroimagerie (LREN) at the Centre hospitalier universitaire vaudois (CHUV), Lausanne, Switzerland. We would like to thank Dr. Bogdan Draganski for his support and Dr. Antoine Lutti for the fMRI acquisition sequence and his invaluable help. We also thank Dr. Leyla Loued-Khenissi, Remi Castella, and Estelle Dupuis for their help with scanning, and Maya Jastrzebowska for useful discussions.

4.7 Contributions

VL, ML, KP, WG and MH conceived the project.
VL, ML, and KP designed the experiment.
VL and ML wrote the code for the experiment and analyzed the behavioural data. VL ran the experiments, analyzed the fMRI data. VL developed the Bayesian and approximate Bayesian treatment of the task and the surprise-modulated reinforcement learning algorithms, with the feedback of AM.
VL, ML and WG interpreted the results.
JB and AM provided invaluable insights on the analysis of the data, the modelling of the behaviour, and on the interpretation of the results.
VL, ML and WG wrote the manuscript, which was reviewed and improved by JB, AM and KP.

5 Contributions

My thesis lies in the intersection of theory, computational modelling and experiments, in order to study model estimation via surprise and its contributions to reinforcement learning.
I developed a surprise-based adaptive algorithm (particle filter) and by means of simulations I showed that it performs better than alternative approximate approaches and more robustly across tasks. Together with Alireza Modirshanechi, I showed that many existing model learning approaches exhibit the same surprise modulation in their update rules (Chapter 2). I, then, coupled the particle filtering model learning approach with a reinforcement learning agent, and I investigated via simulations the scenarios in which surprise adaptation can be beneficial for reward-based tasks (Chapter 3). Next, I sought to detect signals of model learning and model-free reward-based learning in human brain signals and behaviour. To this end, I designed and implemented, together with Dr. Marco Lehmann, a multi-step task that dissociates reward learning signals from model learning ones at the level of brain BOLD responses. I conducted the fMRI experiments, analyzed the data, and developed an approximate Bayesian surprise-based algorithm that implements outlier detection, as well as hybrid model-free and model-based algorithms with surprise modulation. I extended previous fMRI findings on model-free and model-based learning in a multi-step scenario and reported evidence for a model-free Actor-critic architecture as the most likely model of human behaviour, with possible contributions from model estimation (Chapter 4). In a separate project, I designed and implemented, together with Dr. Marco Lehmann, a behavioural and pupillometry experiment that studies fast learning in humans via eligibility traces (Appendix).

6 Conclusion

In this thesis, we went from adaptive model estimation in theory and simulations, through model estimation applied to reinforcement learning agents, to reward learning with and without model estimation in humans. We have already discussed our results, as well as possible next steps, for each project individually. Here, we summarize and discuss the findings of this thesis collectively and provide some ideas and speculations for future research.

Starting from theoretical arguments, we have shown that Bayesian inference entails a surprise measure which has an intuitive interpretation and is a common feature of many learning methods. Our work brings under the same umbrella different approaches of varying accuracy, computational complexity and biological plausibility. Next, we studied model estimation contributions in reinforcement learning tasks, in artificial agents and in humans. In the first case, we used a similar generative model as for the pure model learners and investigated the benefits of accurate model estimation in reward performance. In the second case, we were interested in dissociating brain signatures of model-free and model-based learning. We, thus, used a different set-up where surprising events were a handle to differentiate the dynamics of learning signals. We showed that Bayesian inference leads in this case to the same surprise measure we found earlier and an outlier detection algorithm. We developed hybrid reinforcement learning algorithms that use this outlier detection mechanisms in their update rules. In BOLD responses, we found signatures of model-free learning, as well as model estimation signals.

The common line in this thesis has been how surprise and model estimation manifest in learning performance and in the brain. When the goal is estimation accuracy per se, we have seen that surprise-based learning can give rise to higher performance. When the goal is to obtain reward, however, the role of surprise and accurate model estimation depends more on the situation. In our simulations and experimental results, we have detected influences of surprise in performance and in brain signals, respectively, in environments involving sudden changes that directly affect the selection of the next action: the blocking

of a passage in the case of simulations and a jump to some other state in the experiment. Surprise implies "unexpectedness", which in turn implies the existence of an expectation; something is perceived as surprising when an agent has formed a belief about the world. In an environment exhibiting deterministic periods, marked by sudden changes, such the real world, surprise has a stronger and important effect.

We see interesting extensions of our work, both in terms of theory and experiments. On the theory side, it would be an exciting continuation to develop algorithms that utilize multiple types of surprise, at multiple levels, i.e. for both change-point and outlier detection, and investigate how these signals may interact. Equally exciting would be the simultaneous online estimation of the environment's hyper-parameters and the study of the inductive biases that may be necessary to this end. On the experimental side, it would be fascinating to test experimentally if the "Bayes Factor Surprise" is present in the brain and to design experiments that could differentiate among the surprise-based algorithms we considered, both in behaviour and in neural signals. For model estimation in reward-based tasks, the investigation of more scenarios would be insightful, such as continuous changes in the reward values, and more complex environments with different degrees of volatility in different parts. Our motivation with our fMRI experiment has been to study human learning and brain signals in a multi-step, more complex and presumably more realistic scenario. Our task is, however, still far from realistic situations encountered by biological agents. The design of experiments involving tasks that are closer to real life will be a crucial step in understanding the learning schemes that animals and humans may employ.

A Appendix

A.1 One-shot learning and behavioral eligibility traces in sequential decision making

Marco Lehmann, He Xu, Vasiliki Liakoni, Michael Herzog, Wulfram Gerstner, and Kerstin Preuschoff.

Published in: Elife, 2019.
(Lehmann et al., 2019), doi: 10.7554/eLife.47463

Abstract
In many daily tasks, we make multiple decisions before reaching a goal. In order to learn such sequences of decisions, a mechanism to link earlier actions to later reward is necessary. Reinforcement learning (RL) theory suggests two classes of algorithms solving this credit assignment problem: In classic temporal-difference learning, earlier actions receive reward information only after multiple repetitions of the task, whereas models with eligibility traces reinforce entire sequences of actions from a single experience (one-shot). Here, we show one-shot learning of sequences. We developed a novel paradigm to directly observe which actions and states along a multi-step sequence are reinforced after a single reward. By focusing our analysis on those states for which RL with and without eligibility trace make qualitatively distinct predictions, we find direct behavioral (choice probability) and physiological (pupil dilation) signatures of reinforcement learning with eligibility trace across multiple sensory modalities.

Author contributions
MP, VL, MH, KP and WG conceived the project and designed the experiment.
MP and VL implemented the experiment.
MP and VL ran the pupillometry experiments.
HX and ML ran the EEG experiments.
ML analyzed the behavioral and pupil data.

HX analyzed the EEG data.
MP, HX, VL, MH, KP and WG discussed and interpreted the results.
ML, HX, KP and WG wrote the manuscript.

Bibliography

Christopher D Adams and Anthony Dickinson. Instrumental responding following reinforcer devaluation. *The Quarterly Journal of Experimental Psychology Section B*, 33 (2b):109–121, 1981.

Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

William H Alexander and Joshua W Brown. The role of the anterior cingulate cortex in prediction error and signaling surprise. *Topics in cognitive science*, 11(1):119–135, 2019.

Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

J Yu Angela. Change is in the eye of the beholder. *Nature neuroscience*, 15(7):933, 2012.

Dian Anggraini, Stefan Glasauer, and Klaus Wunderlich. Neural signatures of reinforcement learning correlate with strategy adoption during spatial navigation. *Scientific reports*, 8(1):1–14, 2018.

G Aston-Jones, J Rajkowski, and P Kubiak. Conditioned responses of monkey locus coeruleus neurons anticipate acquisition of discriminative behavior in a vigilance task. *Neuroscience*, 80(3):697–715, 1997.

Gary Aston-Jones and Jonathan D Cohen. An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance. *Annu. Rev. Neurosci.*, 28:403–450, 2005.

Gary Aston-Jones, Janusz Rajkowski, Piotr Kubiak, and Tatiana Alexinsky. Locus coeruleus neurons in monkey are selectively activated by attended cues in a vigilance task. *Journal of Neuroscience*, 14(7):4467–4480, 1994.

Michael C Avery and Jeffrey L Krichmar. Neuromodulatory systems and their interactions: a review of models, theories, and experiments. *Frontiers in Neural Circuits*, 11:108, 2017.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Bernard W Balleine. Neural bases of food-seeking: affect, arousal and reward in corticostriatolimbic circuits. *Physiology & behavior*, 86(5):717–730, 2005.

Bernard W Balleine and Anthony Dickinson. Goal-directed instrumental action: contingency and incentive learning and their cortical substrates. *Neuropharmacology*, 37(4-5): 407–419, 1998.

David Barber. Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning Research*, 7(Nov):2515–2540, 2006.

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Andrew G Barto. Learning by statistical cooperation of self-interested neuron-like computing elements. *Human Neurobiology*, 4(4):229–256, 1985.

Tobias Bast, Iain A Wilson, Menno P Witter, and Richard GM Morris. From rapid place learning to behavioral performance: a key role for the intermediate hippocampus. *PLoS Biol*, 7(4):e1000089, 2009.

Matthew James Beal. *Variational algorithms for approximate Bayesian inference*. Ph.D. thesis, University College London, 2003.

Timothy EJ Behrens, Mark W Woolrich, Mark E Walton, and Matthew FS Rushworth. Learning the value of information in an uncertain world. *Nature neuroscience*, 10(9): 1214, 2007.

Timothy EJ Behrens, Laurence T Hunt, Mark W Woolrich, and Matthew FS Rushworth. Associative learning of social value. *Nature*, 456(7219):245–249, 2008.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv preprint arXiv:1707.06887*, 2017.

Boris Belousov, Gerhard Neumann, Constantin A Rothkopf, and Jan R Peters. Catching heuristics are optimal control policies. In *Advances in neural information processing systems*, pages 1426–1434, 2016.

James O Berger and Luis R Pericchi. The intrinsic bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996.

Gregory B Bissonette and Matthew R Roesch. Neurophysiology of rule switching in the corticostriatal circuit. *Neuroscience*, 345:64–76, 2017.

Hugh Carlton Blodgett. The effect of the introduction of reward upon the maze performance of rats. *University of California publications in psychology*, 1929.

Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017.

Rafal Bogacz. Dopamine role in learning and action inference. *BioRxiv*, page 837641, 2019.

Erie D Boorman, Timothy EJ Behrens, Mark W Woolrich, and Matthew FS Rushworth. How green is the grass on the other side? frontopolar cortex and the evidence in favor of alternative courses of action. *Neuron*, 62(5):733–743, 2009.

Sebastien Bouret and Susan J Sara. Network reset: a simplified overarching theory of locus coeruleus noradrenaline function. *Trends in neurosciences*, 28(11):574–582, 2005.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Scott D Brown and Mark Steyvers. Detecting and predicting changes. *Cognitive psychology*, 58(1):49–67, 2009.

Henry W Chase, Poornima Kumar, Simon B Eickhoff, and Alexandre Y Dombrovski. Reinforcement learning models and their neural correlates: An activation likelihood estimation meta-analysis. *Cognitive, affective, & behavioral neuroscience*, 15(2):435–459, 2015.

Samuel PM Choi, Dit-Yan Yeung, and Nevin Lianwen Zhang. An environment model for nonstationary reinforcement learning. In *Advances in neural information processing systems*, pages 987–993, 2000.

Luke T Coddington and Joshua T Dudman. Learning from action: Reconsidering movement signaling in midbrain dopamine neuron activity. *Neuron*, 104(1):63–77, 2019.

Anne Collins and Etienne Koechlin. Reasoning, learning, and creativity: frontal lobe function and human decision-making. *PLoS Biol*, 10(3):e1001293, 2012.

Anne GE Collins and Jeffrey Cockburn. Beyond dichotomies in reinforcement learning. *Nature Reviews Neuroscience*, pages 1–11, 2020.

Dane Corneil, Wulfram Gerstner, and Johanni Brea. Efficient model-based deep reinforcement learning with variational state tabulation. *arXiv preprint arXiv:1802.04325*, 2018.

Etienne Coutureau and Shauna L Parkes. Cortical determinants of goal-directed behavior. In *Goal-Directed Decision Making*, pages 179–197. Elsevier, 2018.

Rachel Cummings, Sara Krehbiel, Yajun Mei, Rui Tuo, and Wanrong Zhang. Differentially private change-point detection. In *Advances in Neural Information Processing Systems*, pages 10825–10834, 2018.

Fiery Cushman and Adam Morris. Habitual control of goal selection in humans. *Proceedings of the National Academy of Sciences*, 112(45):13817–13822, 2015.

Bruno C Da Silva, Eduardo W Basso, Ana LC Bazzan, and Paulo M Engel. Dealing with non-stationary environments using context detection. In *Proceedings of the 23rd international conference on Machine learning*, pages 217–224. ACM, 2006.

Carolina Feher da Silva and Todd A Hare. Humans primarily use model-based inference in the two-stage task. *Nature Human Behaviour*, pages 1–14, 2020.

Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. A distributional code for value in dopamine-based reinforcement learning. *Nature*, 577(7792):671–675, 2020.

Mathieu d'Acremont and Peter Bossaerts. Neural mechanisms behind identification of leptokurtic noise and adaptive behavioral response. *Cerebral Cortex*, 26(4):1818–1830, 2016.

Mathieu d'Acremont, Wolfram Schultz, and Peter Bossaerts. The human brain encodes event frequencies while forming subjective beliefs. *Journal of Neuroscience*, 33(26): 10887–10897, 2013.

Nathaniel Daw and Aaron Courville. The pigeon as particle filter. *Advances in neural information processing systems*, 20:369–376, 2008.

Nathaniel D Daw. Of goals and habits. *Proceedings of the National Academy of Sciences*, 112(45):13749–13750, 2015.

Nathaniel D Daw. Are we of two minds? *Nature neuroscience*, 21(11):1497–1499, 2018.

Nathaniel D Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12):1704–1711, 2005.

Nathaniel D Daw, John P O'Doherty, Peter Dayan, Ben Seymour, and Raymond J Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095):876–879, 2006.

Nathaniel D Daw, Samuel J Gershman, Ben Seymour, Peter Dayan, and Raymond J Dolan. Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69 (6):1204–1215, 2011a.

Nathaniel D Daw et al. Trial-by-trial data analysis using computational models. *Decision making, affect, and learning: Attention and performance XXIII*, 23(1), 2011b.

Peter Dayan. How to set the switches on this thing. *Current opinion in neurobiology*, 22 (6):1068–1074, 2012.

Peter Dayan and Terrence J Sejnowski. Exploration bonuses and dual control. *Machine Learning*, 25(1):5–22, 1996.

Richard Dearden, Nir Friedman, and David Andre. Model-based bayesian exploration. *arXiv preprint arXiv:1301.6690*, 2013.

Lorenz Deserno, Quentin JM Huys, Rebecca Boehme, Ralph Buchert, Hans-Jochen Heinze, Anthony A Grace, Raymond J Dolan, Andreas Heinz, and Florian Schlagenhauf. Ventral striatal dopamine reflects behavioral and neural signatures of model-based control during sequential decision making. *Proceedings of the National Academy of Sciences*, 112(5): 1595–1600, 2015.

Amir Dezfouli and Bernard W Balleine. Actions, action sequences and habits: evidence that goal-directed and habitual action control are hierarchically organized. *PLoS Comput Biol*, 9(12):e1003364, 2013.

Amir Dezfouli, Nura W Lingawi, and Bernard W Balleine. Habits as action sequences: hierarchical action control and changes in outcome value. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 369(1655):20130482, 2014.

Ray J Dolan and Peter Dayan. Goals and habits in the brain. *Neuron*, 80(2):312–325, 2013.

Bradley B Doll, Dylan A Simon, and Nathaniel D Daw. The ubiquity of model-based reinforcement learning. *Current opinion in neurobiology*, 22(6):1075–1081, 2012.

Bradley B Doll, Katherine D Duncan, Dylan A Simon, Daphna Shohamy, and Nathaniel D Daw. Model-based choices involve prospective neural activity. *Nature neuroscience*, 18 (5):767, 2015a.

Bradley B Doll, Daphna Shohamy, and Nathaniel D Daw. Multiple memory systems as substrates for multiple decision systems. *Neurobiology of learning and memory*, 117: 4–13, 2015b.

Philippe Domenech and Etienne Koechlin. Executive control and decision-making in the prefrontal cortex. *Current opinion in behavioral sciences*, 1:101–106, 2015.

Maël Donoso, Anne GE Collins, and Etienne Koechlin. Foundations of human reasoning in the prefrontal cortex. *Science*, 344(6191):1481–1486, 2014.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

Arnaud Doucet and Vladislav B Tadić. Parameter estimation in general state-space models using particle methods. *Annals of the institute of Statistical Mathematics*, 55 (2):409–422, 2003.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

Kenji Doya. Modulators of decision making. *Nature neuroscience*, 11(4):410–416, 2008.

Kenji Doya, Shin Ishii, Alexandre Pouget, and Rajesh PN Rao. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Marcos Economides, Zeb Kurth-Nelson, Annika Lübbert, Marc Guitart-Masip, and Raymond J Dolan. Model-based reasoning in humans becomes automatic with training. *PLoS Comput Biol*, 11(9):e1004463, 2015.

Bradley Efron and Trevor Hastie. *Computer age statistical inference*, volume 5. Cambridge University Press, 2016.

Neir Eshel, Michael Bukwich, Vinod Rao, Vivian Hemmelder, Ju Tian, and Naoshige Uchida. Arithmetic and local circuitry underlying dopamine prediction errors. *Nature*, 525(7568):243–246, 2015.

David R Euston, Aaron J Gruber, and Bruce L McNaughton. The role of medial prefrontal cortex in memory and decision making. *Neuron*, 76(6):1057–1070, 2012.

Mohammadjavad Faraji, Kerstin Preuschoff, and Wulfram Gerstner. Balancing new against old information: the role of puzzlement surprise in learning. *Neural computation*, 30(1):34–83, 2018.

Paul Fearnhead and Zhen Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4): 589–605, 2007.

Alan SR Fermin, Takehiko Yoshida, Junichiro Yoshimoto, Makoto Ito, Saori C Tanaka, and Kenji Doya. Model-based action planning involves cortico-cerebellar and basal ganglia networks. *Scientific reports*, 6:31378, 2016.

Charles Findling, Nicolas Chopin, and Etienne Koechlin. Imprecise neural computations as source of human adaptive behavior in volatile environments. *bioRxiv*, page 799239, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Edwin Fong and CC Holmes. On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496, 2020.

David J Foster and Matthew A Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, 2006.

Elsa Fouragnan, Chris Retzler, and Marios G Philiastides. Separate neural representations of prediction error valence and surprise: Evidence from an fmri meta-analysis. *Human brain mapping*, 39(7):2887–2906, 2018.

Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4):1569–1585, 2011.

Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.

Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127, 2010.

Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo. Active inference: a process theory. *Neural computation*, 29(1):1–49, 2017.

Matthew PH Gardner, Jessica S Conroy, Michael H Shaham, Clay V Styer, and Geoffrey Schoenbaum. Lateral orbitofrontal inactivation dissociates devaluation-sensitive behavior and economic choice. *Neuron*, 96(5):1192–1203, 2017.

Clint P George and Hani Doss. Principled selection of hyperparameters in the latent dirichlet allocation model. *Journal of Machine Learning Research*, 18:162–1, 2017.

Samuel J Gershman. What does the free energy principle tell us about the brain? *arXiv preprint arXiv:1901.07945*, 2019.

Samuel J Gershman and Nathaniel D Daw. Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annual Review of Psychology*, 68: 101–128, 2017.

Samuel J Gershman and Yael Niv. Learning latent structure: carving nature at its joints. *Current opinion in neurobiology*, 20(2):251–256, 2010.

Samuel J Gershman, Arthur B Markman, and A Ross Otto. Retrospective revaluation in sequential decision making: A tale of two systems. *Journal of Experimental Psychology: General*, 143(1):182, 2014a.

Samuel J Gershman, Angela Radulescu, Kenneth A Norman, and Yael Niv. Statistical computations underlying the dynamics of memory updating. *PLoS computational biology*, 10(11):e1003939, 2014b.

Samuel J Gershman, Marie-H Monfils, Kenneth A Norman, and Yael Niv. The computational nature of memory modification. *Elife*, 6:e23763, 2017.

Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12, 2018.

Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

Gerd Gigerenzer and Wolfgang Gaissmaier. Heuristic decision making. *Annual review of psychology*, 62:451–482, 2011.

Sam Gijsen, Miro Grundei, Robert T Lange, Dirk Ostwald, and Felix Blankenburg. Neural surprise in somatosensory bayesian learning. *BioRxiv*, 2020.

Jan Gläscher, Nathaniel Daw, Peter Dayan, and John P O'Doherty. States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron*, 66(4):585–595, 2010.

Christopher M Glaze, Joseph W Kable, and Joshua I Gold. Normative evidence accumulation in unpredictable environments. *Elife*, 4:e08825, 2015.

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

Christina M Gremel and Rui M Costa. Orbitofrontal and striatal circuits dynamically encode the shift between goal-directed and habitual actions. *Nature communications*, 4, 2013.

Suzanne N Haber. Corticostriatal circuitry. *Dialogues in clinical neuroscience*, 18(1):7, 2016.

Emmanuel Hadoux, Aurelie Beynier, and Paul Weng. Sequential Decision-Making under Non-stationary Environments via Sequential Change-point Detection. page 10, 2014.

Todd A Hare, John O'doherty, Colin F Camerer, Wolfram Schultz, and Antonio Rangel. Dissociating the role of the orbitofrontal cortex and the striatum in the computation of goal values and prediction errors. *Journal of neuroscience*, 28(22):5623–5630, 2008.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Benjamin Y Hayden, Sarah R Heilbronner, John M Pearson, and Michael L Platt. Surprise signals in anterior cingulate cortex: neuronal encoding of unsigned reward prediction errors driving adjustment in behavior. *Journal of Neuroscience*, 31(11):4178–4187, 2011.

Donald Olding Hebb. *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall, 1949.

Micha Heilbron and Florent Meyniel. Confidence resets reveal hierarchical adaptive learning in humans. *PLoS computational biology*, 15(4):e1006972, 2019.

Leonhard Held and Manuela Ott. On p-values and bayes factors. 2018.

Hermann von Helmholtz. Concerning the perceptions in general, 1867. 1948.

James D Howard and Thorsten Kahnt. Identity prediction errors in the human midbrain update reward-identity expectations in the orbitofrontal cortex. *Nature communications*, 9(1):1–11, 2018.

Yanping Huang and Rajesh PN Rao. Neurons as monte carlo samplers: Bayesian inference and learning in spiking networks. In *Advances in neural information processing systems*, pages 1943–1951, 2014.

Yi Huang, Zachary A Yaple, and Rongjun Yu. Goal-oriented and habitual decisions: Neural signatures of model-based and model-free learning. *NeuroImage*, page 116834, 2020.

Scott A Huettel, Peter B Mack, and Gregory McCarthy. Perceiving patterns in random series: dynamic processing of sequence in prefrontal cortex. *Nature neuroscience*, 5(5): 485–490, 2002.

Chloe Hutton, Andreas Bork, Oliver Josephs, Ralf Deichmann, John Ashburner, and Robert Turner. Image distortion correction in fmri: a quantitative evaluation. *Neuroimage*, 16(1):217–240, 2002.

QJ Huys, Neir Eshel, Elizabeth O'Nions, Luke Sheridan, Peter Dayan, and Jonathan P Roiser. Bonsai trees in your head: how the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS Comput. Biol*, 8(3):e1002410, 2012.

Makoto Ito and Kenji Doya. Multiple representations and algorithms for reinforcement learning in the cortico-basal ganglia circuit. *Current Opinion in Neurobiology*, 21(3): 368–373, June 2011.

Laurent Itti and Pierre F Baldi. Bayesian surprise attracts human attention. In *Advances in neural information processing systems*, pages 547–554, 2006.

Xin Jin and Rui M Costa. Start/stop signals emerge in nigrostriatal circuits during sequence learning. *Nature*, 466(7305):457–462, 2010.

Daphna Joel, Yael Niv, and Eytan Ruppin. Actor–critic models of the basal ganglia: New anatomical and computational perspectives. *Neural networks*, 15(4-6):535–547, 2002.

Adam Johnson and A David Redish. Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point. *Journal of Neuroscience*, 27(45):12176–12189, 2007.

Siddhartha Joshi and Joshua I Gold. Pupil size as a window on neural substrates of cognition. *PsyArXiv*, 2019.

Keno Juechems and Christopher Summerfield. Where does value come from? *Trends in cognitive sciences*, 23(10):836–850, 2019.

Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

Alex Kearney, Vivek Veeriah, Jaden B Travnik, Richard S Sutton, and Patrick M Pilarski. Tidbd: Adapting temporal-difference step-sizes through stochastic meta-descent. *arXiv preprint arXiv:1804.03334*, 2018.

Etienne Koechlin. Prefrontal executive function and adaptive behavior in complex environments. *Current Opinion in Neurobiology*, 37:1–6, 2016.

Etienne Koechlin and Alexandre Hyafil. Anterior prefrontal function and the limits of human decision-making. *Science*, 318(5850):594–598, 2007.

Nils Kolling, Marco K Wittmann, Tim EJ Behrens, Erie D Boorman, Rogier B Mars, and Matthew FS Rushworth. Value, search, persistence and model updating in anterior cingulate cortex. *Nature neuroscience*, 19(10):1280–1285, 2016.

Antonio Kolossa, Tim Fingscheidt, Karl Wessel, and Bruno Kopp. A model-based approach to trial-by-trial p300 amplitude fluctuations. *Frontiers in human neuroscience*, 6:359, 2013.

Arkady Konovalov and Ian Krajbich. Neurocomputational dynamics of sequence learning. *Neuron*, 98(6):1282–1293, 2018.

Bruno Kopp and Florian Lange. Electrophysiological indicators of surprise and entropy in dynamic task-switching environments. *Frontiers in human neuroscience*, 7:300, 2013.

Konrad P Körding and Daniel M Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244–247, 2004.

Nils B Kroemer, Ying Lee, Shakoor Pooseh, Ben Eppinger, Thomas Goschke, and Michael N Smolka. L-dopa reduces model-free control of behavior by attenuating the transfer of value to action. *Neuroimage*, 186:113–125, 2019.

Lea K Krugel, Guido Biele, Peter NC Mohr, Shu-Chen Li, and Hauke R Heekeren. Genetic variation in dopaminergic neuromodulation influences the ability to rapidly and flexibly adapt decisions. *Proceedings of the National Academy of Sciences*, 106(42):17951–17956, 2009.

Anna Kutschireiter, Simone Carlo Surace, Henning Sprekeler, and Jean-Pascal Pfister. Nonlinear bayesian filtering and learning: a neuronal dynamics for perception. *Scientific reports*, 7(1):8722, 2017.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Angela J Langdon, Melissa J Sharpe, Geoffrey Schoenbaum, and Yael Niv. Model-based predictions for dopamine. *Current Opinion in Neurobiology*, 49:1–7, 2018.

Claudio Lavín, René San Martín, and Eduardo Rosales Jubal. Pupil dilation signals uncertainty and surprise in a learning gambling task. *Frontiers in Behavioral Neuroscience*, 7:218, 2014.

Sang Wan Lee, Shinsuke Shimojo, and John P O'Doherty. Neural computations underlying arbitration between model-based and model-free learning. *Neuron*, 81(3):687–699, 2014.

Robert Legenstein and Wolfgang Maass. Ensembles of spiking neurons with noise support optimal probabilistic inference in a dynamically changing environment. *PLoS computational biology*, 10(10), 2014.

Marco P Lehmann, He A Xu, Vasiliki Liakoni, Michael H Herzog, Wulfram Gerstner, and Kerstin Preuschoff. One-shot learning and behavioral eligibility traces in sequential decision making. *Elife*, 8:e47463, 2019.

Máté Lengyel and Peter Dayan. Hippocampal contributions to control: The third way. In *NIPS*, volume 20, pages 889–896, 2007.

Jian Li and Nathaniel D Daw. Signals in human striatum are appropriate for policy update rather than value prediction. *Journal of Neuroscience*, 31(14):5504–5511, 2011.

Falk Lieder, Jean Daunizeau, Marta I Garrido, Karl J Friston, and Klaas E Stephan. Modelling trial-by-trial changes in the mismatch negativity. *PLoS computational biology*, 9(2), 2013.

Kevin Lin, James L Sharpnack, Alessandro Rinaldo, and Ryan J Tibshirani. A sharp error analysis for the fused lasso, with application to approximate changepoint screening. In *Advances in Neural Information Processing Systems*, pages 6884–6893, 2017.

John Lisman, Anthony A Grace, and Emrah Duzel. A neohebbian framework for episodic memory; role of dopamine-dependent late ltp. *Trends in neurosciences*, 34(10):536–547, 2011.

Jane Liu and Mike West. Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer, 2001.

Vincenzo Lomonaco, Karan Desai, Eugenio Culurciello, and Davide Maltoni. Continual reinforcement learning in 3d non-stationary environments. *arXiv preprint arXiv:1905.10112*, 2019.

Leyla Loued-Khenissi, Adrien Pfeuffer, Wolfgang Einhäuser, and Kerstin Preuschoff. Anterior insula reflects surprise in value-based decision-making and perception. *NeuroImage*, page 116549, 2020.

Maxime Maheu, Stanislas Dehaene, and Florent Meyniel. Brain signatures of a multiscale process of sequence learning in humans. *Elife*, 8:e41541, 2019.

Rogier B Mars, Stefan Debener, Thomas E Gladwin, Lee M Harrison, Patrick Haggard, John C Rothwell, and Sven Bestmann. Trial-by-trial fluctuations in the event-related electroencephalogram reflect dynamic changes in the degree of surprise. *Journal of Neuroscience*, 28(47):12539–12545, 2008.

Stephen J Martin, Paul D Grimwood, and Richard GM Morris. Synaptic plasticity and memory: an evaluation of the hypothesis. *Annual review of neuroscience*, 23(1):649–711, 2000.

Andrés Masegosa, Thomas D Nielsen, Helge Langseth, Darío Ramos-López, Antonio Salmerón, and Anders L Madsen. Bayesian models of data streams with hierarchical power priors. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2334–2343. JMLR. org, 2017.

Alexander Mathis, Andreas VM Herz, and Martin Stemmler. Optimal population codes for space: grid cells outperform place cells. *Neural computation*, 24(9):2280–2317, 2012.

Christoph Mathys, Jean Daunizeau, Karl J Friston, and Klaas Enno Stephan. A bayesian foundation for individual learning under uncertainty. *Frontiers in human neuroscience*, 5:39, 2011.

Madoka Matsumoto, Kenji Matsumoto, Hiroshi Abe, and Keiji Tanaka. Medial prefrontal cell activity signaling prediction errors of action values. *Nature neuroscience*, 10(5): 647–656, 2007.

Michael A McDannald, Federica Lucantonio, Kathryn A Burke, Yael Niv, and Geoffrey Schoenbaum. Ventral striatum and orbitofrontal cortex are both required for model-based, but not model-free, reinforcement learning. *Journal of Neuroscience*, 31(7): 2700–2705, 2011.

William Menegas, Korleki Akiti, Ryunosuke Amo, Naoshige Uchida, and Mitsuko Watabe-Uchida. Dopamine neurons projecting to the posterior striatum reinforce avoidance of threatening stimuli. *Nature neuroscience*, 21(10):1421–1430, 2018.

Florent Meyniel, Maxime Maheu, and Stanislas Dehaene. Human inferences about sequences: A minimal transition probability model. *PLoS computational biology*, 12 (12):e1005260, 2016.

John G Mikhael, HyungGoo R Kim, Naoshige Uchida, and Samuel J Gershman. Ramping and state uncertainty in the dopamine signal. *bioRxiv*, page 805366, 2019.

Kevin Miller and Sarah Jo Venditto. Multi-step planning in the brain. 2020.

Alireza Modirshanechi, Mohammad Mahdi Kiani, and Hamid Aghajan. Trial-by-trial surprise-decoding model for visual and auditory binary oddball tasks. *NeuroImage*, 196:302–317, 2019.

P Read Montague, Peter Dayan, and Terrence J Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of neuroscience*, 16 (5):1936–1947, 1996.

Andrew W Moore and Christopher G Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine learning*, 13(1):103–130, 1993.

Joachim Morrens, Çağatay Aydin, Aliza Janse van Rensburg, José Esquivelzeta Rabell, and Sebastian Haesler. Cue-evoked dopamine promotes conditioned responding during learning. *Neuron*, 2020.

Elisabeth A Murray and Peter H Rudebeck. Specializations for reward-guided decision-making in the primate ventral prefrontal cortex. *Nature Reviews Neuroscience*, 19(7): 404–417, 2018.

Lea Musiolek, Felix Blankenburg, Dirk Ostwald, and Milena Rabovsky. Modeling the n400 brain potential as semantic bayesian surprise. In *2019 Conference on Cognitive Computational Neuroscience*, 2019.

Blake Myers-Schulz and Michael Koenigs. Functional anatomy of ventromedial prefrontal cortex: implications for mood and anxiety disorders. *Molecular psychiatry*, 17(2): 132–141, 2012.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

Matthew R Nassar, Robert C Wilson, Benjamin Heasly, and Joshua I Gold. An approximately bayesian delta-rule model explains the dynamics of belief updating in a changing environment. *Journal of Neuroscience*, 30(37):12366–12378, 2010.

Matthew R Nassar, Katherine M Rumsey, Robert C Wilson, Kinjan Parikh, Benjamin Heasly, and Joshua I Gold. Rational regulation of learning dynamics by pupil-linked arousal systems. *Nature neuroscience*, 15(7):1040, 2012.

Matthew R Nassar, Rasmus Bruckner, and Michael J Frank. Statistical context dictates the relationship between feedback-related eeg signals and learning. *Elife*, 8:e46975, 2019.

Andrew A Neath and Joseph E Cavanaugh. The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.

Thomas E Nichols and Andrew P Holmes. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping*, 15(1):1–25, 2002.

Yael Niv. Learning task-state representations. *Nature neuroscience*, 22(10):1544–1553, 2019.

Yael Niv and Geoffrey Schoenbaum. Dialogues on prediction errors. *Trends in cognitive sciences*, 12(7):265–272, 2008.

John O'Doherty, Peter Dayan, Johannes Schultz, Ralf Deichmann, Karl Friston, and Raymond J Dolan. Dissociable roles of ventral and dorsal striatum in instrumental conditioning. *science*, 304(5669):452–454, 2004.

John P O'Doherty, Peter Dayan, Karl Friston, Hugo Critchley, and Raymond J Dolan. Temporal difference models and reward-related learning in the human brain. *Neuron*, 38(2):329–337, 2003.

John P O'Doherty, Sang Wan Lee, and Daniel McNamee. The structure of reinforcement-learning mechanisms in the human brain. *Current Opinion in Behavioral Sciences*, 1: 94–100, 2015.

John Philip O'Doherty, Sangwan Lee, Reza Tadayonnejad, Jeff Cockburn, Kiyohito Iigaya, and Caroline J Charpentier. Why and how the brain weights contributions from a mixture of experts. 2020.

John O'Keefe and Lynn Nadel. *The hippocampus as a cognitive map*. Oxford: Clarendon Press, 1978.

Jill X O'Reilly, Urs Schüffelgen, Steven F Cuell, Timothy EJ Behrens, Rogier B Mars, and Matthew FS Rushworth. Dissociable effects of surprise and model update in parietal and anterior cingulate cortex. *Proceedings of the National Academy of Sciences*, 110 (38):E3660–E3669, 2013.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

Dirk Ostwald, Bernhard Spitzer, Matthias Guggenmos, Timo T Schmidt, Stefan J Kiebel, and Felix Blankenburg. Evidence for neural encoding of bayesian surprise in human somatosensation. *NeuroImage*, 62(1):177–188, 2012.

A Ross Otto, Samuel J Gershman, Arthur B Markman, and Nathaniel D Daw. The curse of planning dissecting multiple reinforcement-learning systems by taxing the central executive. *Psychological science*, 24:751–761, 2013a.

A Ross Otto, Candace M Raio, Alice Chiang, Elizabeth A Phelps, and Nathaniel D Daw. Working-memory capacity protects model-based learning from stress. *Proceedings of the National Academy of Sciences*, 110(52):20941–20946, 2013b.

Emre Özkan, Václav Šmídl, Saikat Saha, Christian Lundquist, and Fredrik Gustafsson. Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters. *Automatica*, 49(6):1566–1575, 2013.

Camillo Padoa-Schioppa and John A Assad. Neurons in the orbitofrontal cortex encode economic value. *Nature*, 441(7090):223–226, 2006.

Xiaochuan Pan, Hongwei Fan, Kosuke Sawa, Ichiro Tsuda, Minoru Tsukada, and Masamichi Sakagami. Reward inference by primate prefrontal and striatal neurons. *The Journal of Neuroscience*, 34(4):1380–1396, 2014.

Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*, 2019.

Athanasios Papoulis and H Saunders. *Probability, random variables and stochastic processes*. American Society of Mechanical Engineers Digital Collection, 1989.

Ivan P Pavlov. Conditioned reflexes, translated by gv anrep. *London: Oxford*, 1927.

Elise Payzan-LeNestour, Simon Dunne, Peter Bossaerts, and John P O'Doherty. The neural representation of unexpected uncertainty during value-based decision making. *Neuron*, 79(1):191–201, 2013.

John M Pearce and Geoffrey Hall. A model for pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological review*, 87 (6):532, 1980.

Jing Peng and Ronald J Williams. Efficient learning and planning within the dyna framework. *Adaptive behavior*, 1(4):437–454, 1993.

Cyril R Pernet. Misconceptions in the use of the general linear model applied to functional mri: a tutorial for junior neuro-imagers. *Frontiers in neuroscience*, 8:1, 2014.

J. Peters. Policy gradient methods. *Scholarpedia*, 5(11):3698, 2010. . revision #137199.

Brad E Pfeiffer and David J Foster. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74–79, 2013.

Benedikt A Poser, Maarten J Versluis, Johannes M Hoogduin, and David G Norris. Bold contrast sensitivity enhancement and artifact reduction with multiecho epi: parallel-acquired inhomogeneity-desensitized fmri. *Magnetic Resonance in Medicine*, 55(6): 1227–1235, 2006.

Arthur Prat-Carrabin, Robert C Wilson, Jonathan D Cohen, and Rava Azeredo Da Silveira. Human inference in changing environments with temporal structure. *BioRxiv*, page 720516, 2020.

Kerstin Preuschoff, Bernard Marius t Hart, and Wolfgang Einhäuser. Pupil dilation signals surprise: evidence for noradrenaline's role in decision making. *Front Neurosci*, 5:115, 2011.

D Purves, GJ Augustine, D Fitzpatrick, WC Hall, AS LaMantia, JO McNamara, and SM Williams. Neuroscience. 3rd. *Massachusetts: Sinauer Associates Inc Publishers*, 2004.

Lionel Rigoux, Klaas Enno Stephan, Karl J Friston, and Jean Daunizeau. Bayesian model selection for group studies - revisited. *Neuroimage*, 84:971–985, 2014.

Matthew R Roesch, Donna J Calu, and Geoffrey Schoenbaum. Dopamine neurons encode the better option in rats deciding between differently delayed or sized rewards. *Nature neuroscience*, 10(12):1615–1624, 2007.

Marion Rouault, Jan Drugowitsch, and Etienne Koechlin. Prefrontal mechanisms combining rewards and beliefs in human decision-making. *Nature communications*, 10(1): 1–16, 2019.

Nina Rouhani, Kenneth A Norman, Yael Niv, and Aaron M Bornstein. Reward prediction errors create event boundaries in memory. *Cognition*, 203:104269, 2020.

Peter H Rudebeck and Erin L Rich. Orbitofrontal cortex. *Current Biology*, 28(18): R1083–R1088, 2018.

Matthew FS Rushworth, MaryAnn P Noonan, Erie D Boorman, Mark E Walton, and Timothy E Behrens. Frontal cortex and reward-guided learning and decision-making. *Neuron*, 70(6):1054–1069, 2011.

Roland T Rust and David C Schmittlein. A bayesian cross-validated likelihood method for comparing alternative specifications of quantitative models. *Marketing Science*, 4 (1):20–40, 1985.

Chaitanya Ryali, Gautam Reddy, and Angela J Yu. Demystifying excessively volatile human learning: A bayesian persistent prior and a neural approximation. In *Advances in Neural Information Processing Systems*, pages 2781–2790, 2018.

Chaitanya K Ryali and Angela J Yu. Change-point detection without needing to detect change-points? *bioRxiv*, page 077719, 2016.

Susan J Sara, Carole Dyon-Laurent, and Anne Hervé. Novelty seeking behavior in the rat is dependent upon the integrity of the noradrenergic system. *Cognitive Brain Research*, 2(3):181–187, 1995.

Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

Daniel J Schad, Michael A Rapp, Maria Garbusow, Stephan Nebe, Miriam Sebold, Elisabeth Obst, Christian Sommer, Lorenz Deserno, Milena Rabovsky, Eva Friedel, et al. Dissociating neural learning signals in human sign-and goal-trackers. *Nature Human Behaviour*, 4(2):201–214, 2020.

Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pages 1458–1463, 1991.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Geoffrey Schoenbaum, Matthew R Roesch, Thomas A Stalnaker, and Yuji K Takahashi. A new perspective on the role of the orbitofrontal cortex in adaptive behaviour. *Nature Reviews Neuroscience*, 10(12):885–892, 2009.

Nicolas W Schuck, Robert Wilson, and Yael Niv. A state representation for reinforcement learning and decision-making in the orbitofrontal cortex. In *Goal-directed decision making*, pages 259–278. Elsevier, 2018.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

Philipp Schwartenbeck, Thomas FitzGerald, Ray Dolan, and Karl Friston. Exploration, novelty, surprise, and free energy minimization. *Frontiers in psychology*, 4:710, 2013.

Philipp Schwartenbeck, Thomas HB FitzGerald, and Ray Dolan. Neural signals encoding shifts in beliefs. *Neuroimage*, 125:578–586, 2016.

William W Seeley, Vinod Menon, Alan F Schatzberg, Jennifer Keller, Gary H Glover, Heather Kenna, Allan L Reiss, and Michael D Greicius. Dissociable intrinsic connectivity networks for salience processing and executive control. *Journal of Neuroscience*, 27(9): 2349–2356, 2007.

C Shannon. A mathematical theory of communication. *Bell System Technical Journal 27: 379-423 and 623–656*, 20, 1948.

Melissa J Sharpe, Thomas Stalnaker, Nicolas W Schuck, Simon Killcross, Geoffrey Schoenbaum, and Yael Niv. An integrated model of action selection: distinct modes of cortical control of striatal decision making. *Annual review of psychology*, 2019.

Lei Shi and Thomas L Griffiths. Neural implementation of hierarchical bayesian inference by importance sampling. In *Advances in neural information processing systems*, pages 1669–1677, 2009.

Dylan Alexander Simon and Nathaniel D Daw. Neural correlates of forward planning in a spatial decision task in humans. *The Journal of Neuroscience*, 31(14):5526–5539, 2011.

Herbert A Simon. Models of man; social and rational. 1957.

Alyssa H Sinclair and Morgan D Barense. Surprise and destabilize: prediction error influences episodic memory reconsolidation. *Learning & Memory*, 25(8):369–381, 2018.

Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3):287–308, 2000.

Kenneth C Squires, Christopher Wickens, Nancy K Squires, and Emanuel Donchin. The effect of stimulus sequence on the waveform of the cortical event-related potential. *Science*, 193(4258):1142–1146, 1976.

Thomas A Stalnaker, Nisha K Cooch, and Geoffrey Schoenbaum. What the orbitofrontal cortex does not do. *Nature neuroscience*, 18(5):620, 2015.

Thomas A Stalnaker, Tzu-Lan Liu, Yuji K Takahashi, and Geoffrey Schoenbaum. Orbitofrontal neurons signal reward predictions, not reward prediction errors. *Neurobiology of learning and memory*, 153:137–143, 2018.

Thomas A Stalnaker, James D Howard, Yuji K Takahashi, Samuel J Gershman, Thorsten Kahnt, and Geoffrey Schoenbaum. Dopamine neuron ensembles signal the content of sensory prediction errors. *Elife*, 8, 2019.

Klaas Enno Stephan, Will D Penny, Jean Daunizeau, Rosalyn J Moran, and Karl J Friston. Bayesian model selection for group studies. *Neuroimage*, 46(4):1004–1017, 2009.

Jan Storck, Sepp Hochreiter, and Jürgen Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, volume 2, pages 159–164. Citeseer, 1995.

Christopher Summerfield and Konstantinos Tsetsos. Do humans make good decisions? *Trends in cognitive sciences*, 19(1):27–34, 2015.

Yi Sun, Faustino Gomez, and Jürgen Schmidhuber. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *International Conference on Artificial General Intelligence*, pages 41–51. Springer, 2011.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. A Bradford book. Bradford Book, 1998.

Yuji Takahashi, Geoffrey Schoenbaum, and Yael Niv. Silencing the critics: understanding the effects of cocaine sensitization on dorsolateral and ventral striatum in the context of an actor/critic model. *Frontiers in neuroscience*, 2:14, 2008.

Yuji K Takahashi, Hannah M Batchelor, Bing Liu, Akash Khanna, Marisela Morales, and Geoffrey Schoenbaum. Dopamine neurons respond to errors in the prediction of sensory features of expected rewards. *Neuron*, 95(6):1395–1405, 2017.

Shingo Tanaka, Xiaochuan Pan, Mineki Oguchi, Jessica E Taylor, and Masamichi Sakagami. Dissociable functions of reward inference in the lateral prefrontal cortex and the striatum. *Frontiers in psychology*, 6, 2015.

Donald Thistlethwaite. A critical review of latent learning and related experiments. *Psychological bulletin*, 48(2):97, 1951.

EL Thorndike. Animal intelligence. darien. *CT, Hafner*, 1911.

Philippe N Tobler, Anthony Dickinson, and Wolfram Schultz. Coding of predicted reward omission by dopamine neurons in a conditioned inhibition paradigm. *Journal of Neuroscience*, 23(32):10402–10410, 2003.

Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

René Traoré, Hugo Caselles-Dupré, Timothée Lesort, Te Sun, Guanghang Cai, Natalia Díaz-Rodríguez, and David Filliat. Discorl: Continual reinforcement learning via policy distillation. *arXiv preprint arXiv:1907.05855*, 2019.

Harm Van Seijen and Richard S Sutton. Efficient planning in mdps by small backups. In *Proc. 30th Int. Conf. Mach. Learn.*, pages 1–3, 2013.

Andrey Vankov, Anne Hervé-Minvielle, and Susan J Sara. Response to novelty and its rapid habituation in locus coeruleus neurons of the freely exploring rat. *European Journal of Neuroscience*, 7(6):1180–1187, 1995.

Eliana Vassena, James Deraeve, and William H Alexander. Surprise, value and control in anterior cingulate cortex during speeded decision-making. *Nature Human Behaviour*, 4 (4):412–422, 2020.

Antonino Visalli, Mariagrazia Capizzi, Ettore Ambrosini, Ilaria Mazzonetto, and Antonino Vallesi. Bayesian modeling of temporal expectations in the human brain. *NeuroImage*, 202:116097, 2019.

Simone Vossel, Christoph Mathys, Jean Daunizeau, Markus Bauer, Jon Driver, Karl J Friston, and Klaas E Stephan. Spatial attention, precision, and bayesian inference: a study of saccadic response speed. *Cerebral cortex*, 24(6):1436–1450, 2014.

Jonathan D Wallis. Cross-species studies of orbitofrontal cortex and value-based decision-making. *Nature neuroscience*, 15(1):13–19, 2012.

Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, 21(6):860, 2018.

Yekun Wang and Luis Pericchi. A bridge between cross-validation bayes factors and geometric intrinsic bayes factors. *arXiv preprint arXiv:2006.06495*, 2020.

Mitsuko Watabe-Uchida and Naoshige Uchida. Multiple dopamine systems: Weal and woe of dopamine. In *Cold Spring Harbor symposia on quantitative biology*, volume 83, pages 83–95. Cold Spring Harbor Laboratory Press, 2018.

Mitsuko Watabe-Uchida, Neir Eshel, and Naoshige Uchida. Neural circuitry of reward prediction error. *Annual review of neuroscience*, 40:373–394, 2017.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Paul N Wilson, Patrick Boumphrey, and John M Pearce. Restoration of the orienting response to a light by a change in its predictive accuracy. *The Quarterly Journal of Experimental Psychology Section B*, 44(1b):17–36, 1992.

Robert C Wilson, Matthew R Nassar, and Joshua I Gold. Bayesian online learning of the hazard rate in change-point problems. *Neural computation*, 22(9):2452–2476, 2010.

Robert C Wilson, Matthew R Nassar, and Joshua I Gold. A mixture of delta-rules approximation to bayesian inference in change-point problems. *PLoS computational biology*, 9(7):e1003150, 2013.

Robert C Wilson, Yuji K Takahashi, Geoffrey Schoenbaum, and Yael Niv. Orbitofrontal cortex as a cognitive map of task space. *Neuron*, 81(2):267–279, 2014.

G Elliott Wimmer and Daphna Shohamy. Preference by association: how memory mechanisms in the hippocampus bias decisions. *Science*, 338(6104):270–273, 2012.

Klaus Wunderlich, Peter Dayan, and Raymond J Dolan. Mapping value based planning and extensively trained choice in the human brain. *Nature neuroscience*, 15(5):786–791, 2012a.

Klaus Wunderlich, Peter Smittenaar, and Raymond J Dolan. Dopamine enhances model-based over model-free choice behavior. *Neuron*, 75(3):418–424, 2012b.

Kenny Young, Baoxiang Wang, and Matthew E Taylor. Metatrace: Online step-size tuning by meta-gradient descent for reinforcement learning control. *arXiv preprint arXiv:1805.04514*, 2018.

Angela J Yu and Jonathan D Cohen. Sequential effects: superstition or rational behavior? In *Advances in neural information processing systems*, pages 1873–1880, 2009.

Angela J Yu and Peter Dayan. Uncertainty, neuromodulation, and attention. *Neuron*, 46 (4):681–692, 2005.

Vasiliki Liakoni

vasiliki.liakoni@epfl.ch

EDUCATION

PhD candidate in Neuroscience *2015 - 2020*
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Laboratory of Computational Neuroscience (LCN)

Master in Life Sciences and Technology *2011 - 2014*
EPFL, Switzerland
Major: Neuroscience (Grade: 5.45/6)

5-year Diploma in Electrical and Computer Engineering *2005 - 2010*
National Technical University of Athens, Greece
Major: Energy, Minor: Bioengineering (Grade: 7.88/10)

Senior High School Diploma *2002 - 2005*
Athens, Greece (Grade: 19.3/20)
Consecutive awards and honours from the Greek Ministry of Education for achieving high school GPA above 19/20 *1999 - 2005*

INTERESTS

Reinforcement learning, Machine learning, data analysis, decision making, computational and behavioural neuroscience, mathematical modelling

JOURNAL PAPERS / CONFERENCE PRESENTATIONS

V. Liakoni*, A. Modirshanechi*, W. Gerstner, J. Brea. Learning in Volatile Environments with the Bayes Factor Surprise, Journal paper, to appear in *Neural Computation, 2020.*

M. Lehmann, H. Xu, V. Liakoni, M. Herzog, W. Gerstner, K. Preuschoff. One-shot learning and behavioral eligibility traces in sequential decision making. Journal paper, ELife, 2019

V. Liakoni, M. Lehmann, J. Brea, W. Gerstner, K. Preuschoff. Identifying distinct learning strategies in humans during a complex task. Poster presentation, Reinforcement Learning and Decision Making (RLDM), 2017

V. Liakoni, M. Lehmann, W. Gerstner, K. Preuschoff. Human learning in complex environments: episodic memory challenges the model-free – model-based realm. Poster presentation, Computational and Systems Neuroscience (*Cosyne)*, 2017.

V. Liakoni, M. Lehmann, W. Gerstner, K. Preuschoff. Model-free, model-based and episodic memory contributions during learning in complex environments. Oral presentation, *Alpine Brain Imaging Meeting (ABIM)*, 2017.

RESEARCH / PROFESSIONAL EXPERIENCE

PhD Thesis (ongoing) *2015 - 2020*
Laboratory of Computational Neuroscience, EPFL, Switzerland
Prof. Wulfram Gerstner and Prof. Kerstin Preuschoff

- Thesis topic: Surprise-based model estimation in reinforcement learning: algorithms and brain signatures
- Developed algorithms that can quickly adapt to changes in the environment for both purely model learning and reinforcement learning purposes.
- Designed and performed an experiment to dissociate learning strategies of human participants via computational modelling and brain imaging.

Teaching Assistant - *EPFL, Switzerland*
- Biological Modelling of Neural Networks (Prof. Wulfram Gerstner) *2016 - 2018*
- Linear Algebra (Prof. Kathryn Hess) *2015*

Research engineer position & Master Thesis *2013 - 2014*
MindMaze S.A, Switzerland
Chair in Non-Invasive Brain Machine Interfaces, EPFL, Switzerland
- Title: Neural correlates of mirrored visual feedback in a virtual environment designed for neurorehabilitation. (Grade: 6/6)
- Successfully validated neurohabilitation technology on healthy participants via EEG neural markers.
- Completed a feasibility study on stroke patients (acute neurological rehabilitation unit, CHUV, Lausanne) via EEG and motion data analysis, which allows the conduction of further clinical trials.

Lab Internship *2012 (2 months)*
Chair in Non-Invasive Brain Machine Interfaces, EPFL, Switzerland
- Title: Modulation of motor-related brain activity by transcranial direct current stimulation (tDCS): Effect on healthy subjects and spinal cord injury patients.
- Developed a graphical user interface for EEG artifact detection and handling.
- Identified the machine learning algorithms for best performance in this application.

Diploma Thesis *2010 (6 months)*
National Technical University of Athens, Greece
- Title: Risk factors identification and risk estimation for the development of diabetic retinopathy using artificial intelligence algorithms. (Grade: 10/10)
- Successfully developed a hybrid artificial intelligence algorithm and evaluated it on diabetic patients' data (Diabetes center, Ippokrateio hospital, Athens).

Private tutor - *Greece* *2007 - 2008*
Mathematics and Physics (high-school level)

COMPUTER SKILLS

Programming	Julia, Python, Matlab, C
Miscellaneous	Git, Latex, Illustrator, Microsoft Office Suite

LANGUAGES

Greek	Native language	
English	Proficient user (C1)	Cambridge Proficiency in English (C2), *2003*
German	Basic user (B1) *	Kleines Deutsches Sprachdiplom (C2), *2004* **
French	Basic user (A2/B1)	
Spanish	Beginner (A1)	

* current level ** past acquired level