

## Learning in Volatile Environments With the Bayes Factor Surprise

**Vasiliki Liakoni\***

*vasiliki.liakoni@epfl.ch*

**Alireza Modirshanechi\***

*alireza.modirshanechi@epfl.ch*

**Wulfram Gerstner**

*wulfram.gerstner@epfl.ch*

**Johanni Brea**

*johanni.brea@epfl.ch*

*École Polytechnique Fédérale de Lausanne, School of Computer and Communication Sciences and School of Life Sciences, 1015 Lausanne, Switzerland*

**Surprise-based learning allows agents to rapidly adapt to nonstationary stochastic environments characterized by sudden changes. We show that exact Bayesian inference in a hierarchical model gives rise to a surprise-modulated trade-off between forgetting old observations and integrating them with the new ones. The modulation depends on a probability ratio, which we call the Bayes Factor Surprise, that tests the prior belief against the current belief. We demonstrate that in several existing approximate algorithms, the Bayes Factor Surprise modulates the rate of adaptation to new observations. We derive three novel surprise-based algorithms, one in the family of particle filters, one in the family of variational learning, and one in the family of message passing, that have constant scaling in observation sequence length and particularly simple update dynamics for any distribution in the exponential family. Empirical results show that these surprise-based algorithms estimate parameters better than alternative approximate approaches and reach levels of performance comparable to computationally more expensive algorithms. The Bayes Factor Surprise is related to but different from the Shannon Surprise. In two hypothetical experiments, we make testable predictions for physiological indicators that dissociate the Bayes Factor Surprise from the Shannon Surprise. The theoretical insight of casting various approaches as surprise-based learning, as well as the proposed online algorithms, may be applied to the analysis of animal and human behavior and to reinforcement learning in nonstationary environments.**

---

\*V.L. and A.M. made equal contributions to this article and are co-corresponding authors.

## 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 leaves 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, t Hart, & Einhauser, 2011) and EEG signals (Mars et al., 2008; Modirshanechi, Kiani, & Aghajan, 2019; Ostwald et al., 2012), is believed to modulate learning, potentially through the release of specific neurotransmitters (Gerstner, Lehmann, Liakoni, Corneil, & Brea, 2018; Yu & 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, Woolrich, Walton, & Rushworth, 2007; Glaze, Kable, & Gold, 2015; Heilbron & Meyniel, 2019; Nassar et al., 2012; Nassar, Wilson, Heasly, & Gold, 2010; Yu & Dayan, 2005). The bulk of computational work on surprise-based learning can be separated into 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, Preuschoff, & Gerstner, 2018; Friston, 2010; Friston, FitzGerald, Rigoli, Schwartenbeck, & Pezzulo, 2017; Nassar et al., 2012; Nassar, Wilson, Heasly, & Gold, 2010; Ryali, Reddy, & Yu, 2018; Schwartenbeck, FitzGerald, Dolan, & Friston, 2013; Yu & Dayan, 2005), whereas exact and approximate Bayesian online methods (Adams & MacKay, 2007; Fearnhead & Liu, 2007) for change point detection and parameter estimation have been developed without any focus on biological plausibility (Aminikhanghahi & Cook, 2017; Cummings, Krehbiel, Mei, Tuo, & Zhang, 2018; Lin, Sharpnack, Rinaldo, & Tibshirani, 2017; Masegosa et al., 2017; Wilson, Nassar, & Gold, 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, Chopin, & Koechlin, 2019; Glaze et al., 2015; Heilbron & Meyniel, 2019; Nassar et al., 2012, 2010; Yu & 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 that is different from but closely related to the 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 this article 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 a high level of learning accuracy. 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 section 2, we introduce the generative model and 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., 2012, 2010; Ostwald et al., 2012). At the end of section 2, 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 section 3.

## 2 Results

---

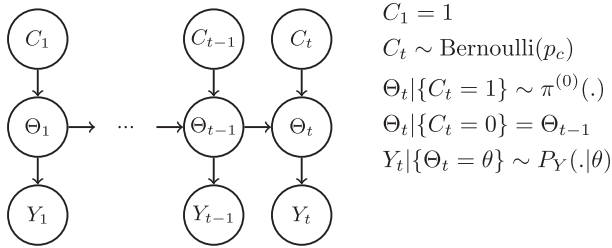
In order to study learning in an environment that exhibits occasional and abrupt changes, we consider a hierarchical generative model (see Figure 1A) in discrete time, similar to existing model environments (Behrens et al., 2007; Nassar et al., 2012, 2010; Yu & Dayan, 2005). At each time point  $t$ , 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  $\theta$  can be multidimensional. In general, we indicate random variables by capital letters and values by lowercase 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  $\theta_t$ . At every time  $t$ , there is a change probability  $p_c \in (0, 1)$  for the parameter  $\theta_t$  to be drawn from its prior distribution  $\pi^{(0)}$  independent of its previous value, and a probability  $1 - p_c$  to stay the same as  $\theta_{t-1}$ . 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  $\Theta_{1:T} \equiv (\Theta_1, \dots, \Theta_T)$ ,  $C_{1:T}$ , and  $Y_{1:T}$  as

$$\mathbf{P}(C_{1:T}, \theta_{1:T}, y_{1:T}) = \mathbf{P}(C_1)\mathbf{P}(\theta_1)\mathbf{P}(y_1|\theta_1) \prod_{t=2}^T \mathbf{P}(C_t)\mathbf{P}(\theta_t|C_t, \theta_{t-1})\mathbf{P}(y_t|\theta_t), \quad (2.1)$$

where  $\mathbf{P}(\theta_1) = \pi^{(0)}(\theta_1)$ ,  $\mathbf{P}(C_1) = \delta(C_1 - 1)$ , and

$$\mathbf{P}(C_t) = \text{Bernoulli}(C_t; p_c), \quad (2.2)$$

A



B

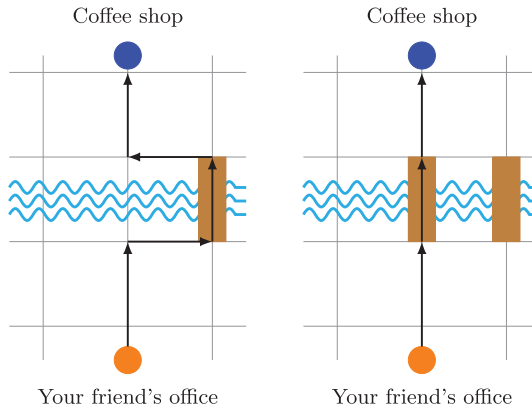


Figure 1: Nonstationary environment. (A) The generative model. At each time point  $t$  there is a probability  $p_c \in (0, 1)$  for a change in the environment. When there is a change in the environment, that is,  $C_t = 1$ , the parameter  $\Theta_t$  is drawn from its prior distribution  $\pi^{(0)}$ , independent of its previous value. Otherwise the value of  $\Theta_t$  retains its value from the previous time step  $t - 1$ . Given a parameter value  $\Theta_t = \theta$ , the observation  $Y_t = y$  is drawn from a probability distribution  $P_Y(y|\theta)$ . We indicate random variables by capital letters, and values by lowercase letters. (B) Example of a nonstationary environment. Your friend meets you every day at the coffee shop (blue dot) starting after work from her office (orange dot) and 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 ( $\theta$ ). If, however, a new bridge is opened ( $C_t = 1$  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.

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

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

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

Given a sequence of observations  $y_{1:t}$ , the agent's belief  $\pi^{(t)}(\theta)$  about the parameter  $\Theta_t$  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  $\pi^{(t)}(\theta)$  to the new belief  $\pi^{(t+1)}(\theta)$ , or an approximation thereof, upon observing  $y_{t+1}$ .

A simplified real-world example of such an environment is illustrated in Figure 1B. Imagine that every day, a friend meets you at the coffee shop, starting after work from her office (see Figure 1B, left). To do so, she needs to cross a river via a bridge. The time of arrival of your friend ( $y_t$ ) 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 ( $\theta_t$ ). However, if a new bridge is opened, your friend arrives earlier, since she no longer has to take a detour (see Figure 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  $\theta_t$  to  $\theta_{t+1}$ . 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.

**2.1 Online Bayesian Inference Modulated by Surprise.** According to the definition of the hierarchical generative model (see Figure 1A and equations 2.1 to 2.4), the value  $y_{t+1}$  of the observation at time  $t + 1$  depends only on the parameters  $\theta_{t+1}$ , and is (given  $\theta_{t+1}$ ) independent of earlier observations and earlier parameter values. We exploit this Markovian property and update, using Bayes' rule, the belief  $\pi^{(t)}(\theta) \equiv \mathbf{P}(\Theta_t = \theta | y_{1:t})$  at time  $t$  to the new belief at time  $t + 1$ :

$$\pi^{(t+1)}(\theta) = \frac{P_Y(y_{t+1} | \theta) \mathbf{P}(\Theta_{t+1} = \theta | y_{1:t})}{\mathbf{P}(y_{t+1} | y_{1:t})}. \quad (2.5)$$

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  $\Theta_{t+1}$  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  $\pi^{(t)}$  but  $\mathbf{P}(\Theta_{t+1} = \theta | y_{1:t}) = (1 - p_c) \pi^{(t)}(\theta) + p_c \pi^{(0)}(\theta)$ . 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  $\pi^{(t)}$  is denoted as

$$P(y; \pi^{(t)}) = \int P_Y(y|\theta)\pi^{(t)}(\theta)d\theta. \quad (2.6)$$

Note that if  $\pi$  is the exact Bayesian belief defined as above in equation 2.5, then  $P(y; \pi^{(t)}) = \mathbf{P}(Y_{t'+1} = y | y_{1:t'}, c_{t'+1} = 0)$ . In section 2.2 we will also use  $P(y; \hat{\pi}^{(t)})$  for an arbitrary  $\hat{\pi}^{(t)}$ . Two particularly interesting cases of equation 2.6 are  $P(y_{t+1}; \pi^{(t)})$ , the probability of a new observation  $y_{t+1}$  with the current belief  $\pi^{(t)}$ , and  $P(y_{t+1}; \pi^{(0)})$ , the probability of a new observation  $y_{t+1}$  with the prior belief  $\pi^{(0)}$ .

**Definition 2.** The Bayes Factor Surprise  $\mathbf{S}_{\text{BF}}$  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), that is,

$$\mathbf{S}_{\text{BF}}(y_{t+1}; \pi^{(t)}) = \frac{P(y_{t+1}; \pi^{(0)})}{P(y_{t+1}; \pi^{(t)})}. \quad (2.7)$$

This definition of surprise measures how much more probable the current observation is under the naive prior  $\pi^{(0)}$  relative to the current belief  $\pi^{(t)}$  (see section 3 for further interpretation). This probability ratio is the Bayes factor (Efron & Hastie, 2016; Kass & Raftery, 1995) that tests the prior belief  $\pi^{(0)}$  against the current belief  $\pi^{(t)}$ . 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  $\pi^{(t)}$  as prior, the exact Bayesian update for  $\pi^{(t+1)}$  is denoted as

$$\pi_B^{(t+1)}(\theta) = \frac{P_Y(y_{t+1}|\theta)\pi^{(t)}(\theta)}{P(y_{t+1}; \pi^{(t)})}. \quad (2.8)$$

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

**Definition 4.** *The surprise-modulated adaptation rate is a function  $\gamma : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow [0, 1]$  specified as*

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

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 Figure 1A and equations 2.1 to 2.4 the following proposition:

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

$$\begin{aligned} \pi^{(t+1)}(\theta) &= \left(1 - \gamma\left(\mathbf{S}_{\text{BF}}^{(t+1)}, \frac{p_c}{1 - p_c}\right)\right) \pi_B^{(t+1)}(\theta) \\ &\quad + \gamma\left(\mathbf{S}_{\text{BF}}^{(t+1)}, \frac{p_c}{1 - p_c}\right) P(\theta|y_{t+1}), \end{aligned} \quad (2.10)$$

where  $\mathbf{S}_{\text{BF}}^{(t+1)} = \mathbf{S}_{\text{BF}}(y_{t+1}; \pi^{(t)})$  is the Bayes Factor Surprise, and

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

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 (see Figure 1) leads to an explicit trade-off between (1) integrating a new observation  $y^{\text{new}}$  (corresponding to  $y_{t+1}$ ) with the old belief  $\pi^{\text{old}}$  (corresponding to  $\pi^{(t)}$ ) into a distribution  $\pi^{\text{integration}}$  (corresponding to  $\pi_B^{(t+1)}$ ) and (2) forgetting the past observations, so as to restart with the belief  $\pi^{\text{reset}}$  (corresponding to  $P(\theta|y_{t+1})$ ) which relies only on the new observation and the prior  $\pi^{(0)}$ :

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

This trade-off is governed by a surprise-modulated adaptation rate  $\gamma(S, m) \in [0, 1]$ , where  $S = \mathbf{S}_{\text{BF}} \geq 0$  (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  $\gamma$  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  $p_c \rightarrow 1$ , any surprise value leads to full forgetting,  $\gamma = 1$ .

As a conclusion, our first main result is that a split as in equation 2.12 with a weighting factor (“adaptation rate”  $\gamma$ ) 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 Approximate Algorithms Modulated by Surprise.** Despite the simplicity of the recursive formula in equation 2.10, the updated belief  $\pi^{(t+1)}$  is generally not in the same family of distributions as the previous belief  $\pi^{(t)}$ , for example, 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, say, some sufficient statistic. As a consequence, the memory demands for  $\pi^{(t+1)}$  scale linearly in time, and updating  $\pi^{(t+1)}$  using  $\pi^{(t)}$  needs  $\mathcal{O}(t)$  operations. In the following sections, we investigate three approximations, algorithms 1 to 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 our three approximate algorithms inherit the surprise-modulated adaptation rate from the exact Bayesian approach, equations 2.9 and 2.12. 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 as VarSMiLe (see algorithm 1). The second algorithm is based on message passing (Adams & MacKay, 2007) restricted to a finite number of messages  $N$ . We refer to this algorithm as MPN (see algorithm 2). The third algorithm uses the ideas of particle filtering (Gordon, Salmond, & Smith, 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 pfN (see algorithm 3). All algorithms are computationally efficient, have finite memory demands, and are biologically plausible; Particle Filtering has possible neuronal implementations (Huang & Rao, 2014; Kutschireiter, Surace, Sprekeler, & Pfister, 2017; Legenstein & Maass, 2014; Shi & Griffiths, 2009), MPN can be seen as a greedy version of pfN without sampling, and Variational SMiLe may be implemented by neo-Hebbian (Gerstner et al., 2018; Lisman, Grace, & Duzel, 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.

*2.2.1 Variational SMiLe Rule: Algorithm 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 (see equation 2.10) to the logarithm of the beliefs rather than the



**Algorithm 1:** Pseudocode for Variational SMiLe (Exponential Family).

---

```

1: Specify  $P_Y(y|\theta)$ ,  $\mathbf{P}_\pi(\Theta = \theta; \chi, \nu)$ , and  $\phi(y)$ 
   where  $P_Y \in \{\text{exponential family}\}$ ,  $\mathbf{P}_\pi \in \{\text{conjugate priors of } P_Y\}$ 
   parametrized by  $\chi$  and  $\nu$ , and  $\phi(y)$  is the sufficient statistic.
2: Specify  $m$ .
3: Initialize  $\chi^{(0)}$ ,  $\nu^{(0)}$ , and  $t \leftarrow 0$ .
4: while the sequence is not finished do
5:   Observe  $y_{t+1}$ 
   # Surprise
6:   Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)})$  using equation 4.56
   # Modulation factor
7:   Compute  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}), m)$ 
   # Updated belief
8:    $\chi^{(t+1)} \leftarrow (1 - \gamma_{t+1})\chi^{(t)} + \gamma_{t+1}\chi^{(0)} + \phi(y_{t+1})$ 
9:    $\nu^{(t+1)} \leftarrow (1 - \gamma_{t+1})\nu^{(t)} + \gamma_{t+1}\nu^{(0)} + 1$ 
10:   $\hat{\pi}^{(t+1)}(\theta) = \mathbf{P}_\pi(\Theta = \theta; \chi^{(t+1)}, \nu^{(t+1)})$ 
   # Iterate
11:   $t \leftarrow t + 1$ 

```

---

beliefs themselves:

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

where  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}), m)$  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 the initial belief  $\pi^{(0)}$  is its conjugate prior, then  $\hat{\pi}^{(t+1)}$  and  $\pi^{(0)}$  are members of the same family. In this particular case, we arrive at a simple update rule for the parameters of  $\hat{\pi}^{(t+1)}$  (see algorithm 1 for pseudocode and section 4 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  $\hat{\pi}^{(t+1)}$  is a variational approximation of the Bayesian update  $\hat{\pi}_B^{(t+1)}$  (see section 4)

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

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

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

where the bound  $B_{t+1} \in [0, \mathbf{D}_{\text{KL}}[\hat{\pi}_B^{(t+1)}(\theta)||P(\theta|y_{t+1})]]$  is a decreasing function of the Bayes Factor Surprise  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)})$  (see section 4 for proof), and  $P(\theta|y_{t+1})$  is given by equation 2.11.

Because of the similarity of the constraint optimization problem in equations 2.14 and 2.15 to the Surprise Minimization Learning rule, (SMiLe) (Faraji et al., 2018), we call this algorithm the Variational Surprise Minimization Learning rule, or for short, the Variational SMiLe rule. The differences between SMiLe and variational SMiLe are discussed in section 4.

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

*2.2.2 Message-Passing N: Algorithm 2.* For a hierarchical generative model similar to ours, a message-passing algorithm has been used to perform exact Bayesian inference (Adams & 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, for example,  $c_{1:t} = \{1, 0, 0, 1, 0, 1, 1\}$ , 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 and  $t$ . We can write the exact Bayesian expression for  $\pi^{(t)}(\theta)$  by marginalizing  $\mathbf{P}(\Theta_t = \theta, r_t | y_{1:t})$  over the  $t$  possible values of  $r_t$  in the following way:

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

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  $\pi_k^{(t)}(\theta) = \mathbf{P}(\Theta_t = \theta | R_t = t - k, y_{1:t})$  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  $t$ :

$$\pi^{(t)}(\theta) = \sum_{k=0}^{t-1} w_t^{(k)} \pi_k^{(t)}(\theta). \quad (2.17)$$

For each particle, the term  $\pi_k^{(t)}(\theta)$  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 (see 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  $P(\theta|y_{t+1})$  (that is,  $\pi^{reset}$ ), modeling the possibility of a change point occurring at  $t + 1$ . According to the proposition, the weight of the new particle ( $k = t$ ) is equal to

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

where  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \pi^{(t)}), \frac{p_c}{1-p_c})$  (cf. equation 2.9). The other  $t$  particles coming from  $\pi^{(t)}$  correspond to  $\pi_B^{(t+1)}$  (that is,  $\pi^{integration}$ ) in the proposition. The update rule (see section 4 for derivation) for the weights of these particles ( $0 \leq k \leq t - 1$ ) is

$$w_{t+1}^{(k)} = (1 - \gamma_{t+1})w_{B,t+1}^{(k)} = (1 - \gamma_{t+1})\frac{P(y_{t+1}; \pi_k^{(t)})}{P(y_{t+1}; \pi^{(t)})}w_t^{(k)}. \quad (2.19)$$

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 equations 2.18 and 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 equations 2.17 to 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 nonzero 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  $N$  (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 & 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 nonlinear 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 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  $\pi^{(0)}$  is its conjugate prior, the resulting algorithm of MPN has a simple update rule for the belief parameters (see algorithm 2 and section 4 for details). For 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, which we call ‘‘SORN.’’

*2.2.3 Particle Filtering: Algorithm 3.* Equation 2.17 demonstrates that the exact Bayesian belief  $\pi^{(t)}$  can be expressed 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  $c_{1:t}$ :

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

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

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

$$\hat{\pi}^{(t)}(\theta) = \sum_{i=1}^N w_i^{(i)} \hat{\pi}_i^{(t)}(\theta) = \sum_{i=1}^N w_i^{(i)} \mathbf{P}(\Theta_t = \theta | c_{1:t}^{(i)}, y_{1:t}), \quad (2.21)$$

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  $Q(c_{1:t} | y_{1:t})$ ,  $\{w_i^{(i)}\}_{i=1}^N$  are their

---

**Algorithm 2:** Pseudocode for MPN (Exponential Family).
 

---

- 1: Specify  $P_Y(y|\theta)$ ,  $\mathbf{P}_\pi(\Theta = \theta; \chi, \nu)$ , and  $\phi(y)$   
 where  $P_Y \in \{\text{exponential family}\}$ ,  $\mathbf{P}_\pi \in \{\text{conjugate priors of } P_Y\}$   
 parameterized by  $\chi$  and  $\nu$ , and  $\phi(y)$  is the sufficient statistic.
  - 2: Specify  $m = p_c/(1 - p_c)$ , and  $N$ .
  - 3: Initialize  $\chi_1^{(0)}$ ,  $\nu_1^{(0)}$ ,  $w_0^{(1)} = 1$  and  $t \leftarrow 0$ .
  - 4: Until  $N = t$ , do the exact message-passing algorithm of equations 2.19 and 2.18
  - 5: **while** the sequence is not finished and  $N < t$  **do**
  - 6:   Observe  $y_{t+1}$   
    # Surprise per particle  $i$
  - 7:   **for**  $i \in \{1, \dots, N\}$  **do**
  - 8:     Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}_i^{(t)})$  using equation 4.56 with  $\chi_i^{(t)}$ ,  $\nu_i^{(t)}$   
    # Global surprise
  - 9:     Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}) = [\sum_{i=1}^N w_i^{(i)} [\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}_i^{(t)})]^{-1}]^{-1}$   
    # Modulation factor
  - 10:    Compute  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}), m)$   
    # Weight per particle  $i$
  - 11:    **for**  $i \in \{1, \dots, N\}$  **do**
  - 12:     Compute the Bayesian weight  $w_{B,t+1}^{(i)}$  using equation 2.22
  - 13:     

$w_{t+1}^{(i)} \leftarrow (1 - \gamma_{t+1})w_{B,t+1}^{(i)}$

  
    # Weight for the new particle
  - 14:     

$w_{t+1}^{(N+1)} \leftarrow \gamma_{t+1}$

  
    # Updated belief per particle  $i$
  - 15:     **for**  $i \in \{1, \dots, N\}$  **do**
  - 16:        $\chi_i^{(t+1)} \leftarrow \chi_i^{(t)} + \phi(y_{t+1})$  and  $\nu_i^{(t+1)} \leftarrow \nu_i^{(t)} + 1$
  - 17:        $\chi_{N+1}^{(t+1)} \leftarrow \chi^{(0)} + \phi(y_{t+1})$  and  $\nu_{N+1}^{(t+1)} \leftarrow \nu^{(0)} + 1$   
       # Approximation
  - 18:     Keep the  $N$  particles with highest weights among  $w_{t+1}^{(1:N+1)}$ , rename, and  
    normalize their weights  
    # Updated belief
  - 19:      $\hat{\pi}^{(t+1)}(\theta) = \sum_{i=1}^N w_{t+1}^{(i)} \mathbf{P}_\pi(\Theta = \theta; \chi_i^{(t+1)}, \nu_i^{(t+1)})$   
    # Iterate
  - 20:     $t \leftarrow t + 1$
- 

corresponding weights at time  $t$ , and  $\hat{\pi}_i^{(t)}(\theta) = \mathbf{P}(\Theta_t = \theta | c_{1:t}^{(i)}, y_{1:t})$  is the approximate belief corresponding to particle  $i$ .

Upon observing  $y_{t+1}$ , the update procedure for the approximate belief  $\hat{\pi}^{(t+1)}$  of equation 2.21 includes two steps: (1) updating the weights and (2) sampling the new hidden state  $c_{t+1}$  for each particle. The two steps are coupled together through the choice of the proposal distribution  $Q$ , for which we choose the optimal proposal function (Doucet et al., 2000; see section 4). As a result, given this choice of proposal function, we show (see

section 4) that the first step amounts to

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

where  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}), m)$  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  $\hat{\pi}_B^{(t+1)}$  of equation 2.8 (see section 4). 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  $\hat{\pi}_i^{(t)}$  to  $\hat{\pi}_i^{(t+1)}$ . We sample the new element  $c_{t+1}^{(i)}$  from the optimal proposal distribution  $Q(c_{t+1}^{(i)} | c_{1:t}^{(i)}, y_{1:t+1})$  (Doucet et al., 2000), which is given by (see section 4)

$$Q(c_{t+1}^{(i)} = 1 | c_{1:t}^{(i)}, y_{1:t+1}) = \gamma \left( \mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}_i^{(t)}), \frac{p_c}{1-p_c} \right). \quad (2.23)$$

Interestingly, the above formulas entail the same surprise modulation and the same trade-off as proposed by the proposition's 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 (see 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  $\hat{\pi}_i^{(t+1)} = \mathbf{P}(\Theta_{t+1} = \theta | c_{t+1}^{(i)} = 1, c_{1:t}^{(i)}, y_{1:t+1})$  is equal to  $\mathbf{P}(\Theta_{t+1} = \theta | c_{t+1}^{(i)} = 1, y_{t+1}) = P(\theta | y_{t+1})$  (see Figure 1A), which is equivalent to a reset of the belief as in equation 2.12. In other words, while in MPN and the exact Bayesian inference in the proposition's equation 2.10, the trade-off between integration and reset is accomplished by adding at each time step a new particle with weight  $\gamma_{t+1}$ , in Particle Filtering, it is accomplished via sampling. As a conclusion, the above formulas are essentially the same as the update rules of MPN (see equations 2.19 and

---

**Algorithm 3:** Pseudocode for Particle Filtering (Exponential Family).
 

---

```

1: Specify  $P_Y(y|\theta)$ ,  $\mathbf{P}_\pi(\Theta = \theta; \chi, \nu)$ , and  $\phi(y)$ 
   where  $P_Y \in \{\text{exponential family}\}$ ,  $\mathbf{P}_\pi \in \{\text{conjugate priors of } P_Y\}$ 
   parameterized by  $\chi$  and  $\nu$ , and  $\phi(y)$  is the sufficient statistic.
2: Specify  $m = p_c/(1 - p_c)$ ,  $N$ , and  $N_{\text{thrs}}$ 
3: Initialize  $\chi^{(0)}$ ,  $\nu^{(0)}$ ,  $w_0^{(i)} \forall i \in \{1 \dots N\}$ , and  $t \leftarrow 0$ .
4: while the sequence is not finished do
5:   Observe  $y_{t+1}$ 
   # Surprise per particle  $i$ 
6:   for  $i \in \{1, \dots, N\}$  do
7:     Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}_i^{(t)})$  using equation 4.56 with  $\chi_i^{(t)}$ ,  $\nu_i^{(t)}$ 
   # Global surprise
8:   Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}) = [\sum_{i=1}^N w_i^{(i)} [\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}_i^{(t)})]^{-1}]^{-1}$ 
   # Modulation factor
9:   Compute  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}), m)$ 
   # Weight per particle  $i$ 
10:  for  $i \in \{1, \dots, N\}$  do
11:    Compute the Bayesian weight  $w_{B,t+1}^{(i)}$  using equation 2.22
12:     $w_{t+1}^{(i)} \leftarrow (1 - \gamma_{t+1})w_{B,t+1}^{(i)} + \gamma_{t+1}w_i^{(i)}$ 
   # Hidden state per particle  $i$ 
13:  for  $i \in \{1, \dots, N\}$  do
14:    Sample  $c_{t+1}^{(i)} \sim \text{Bernoulli}(\gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}_i^{(t)}), m))$ 
   # Resampling
15:   $N_{\text{eff}} \leftarrow (\sum_{i=1}^N w_{t+1}^{(i)2})^{-1}$ 
16:  If  $N_{\text{eff}} \leq N_{\text{thrs}}$ : resample
   # Updated belief per particle  $i$ 
17:  for  $i \in \{1, \dots, N\}$  do
18:    if  $c_{t+1}^{(i)} = 0$  then
19:       $\chi_i^{(t+1)} \leftarrow \chi_i^{(t)} + \phi(y_{t+1})$  and  $\nu_i^{(t+1)} \leftarrow \nu_i^{(t)} + 1$ 
20:    else
21:       $\chi_i^{(t+1)} \leftarrow \chi^{(0)} + \phi(y_{t+1})$  and  $\nu_i^{(t+1)} \leftarrow \nu^{(0)} + 1$ 
   # Updated (output) belief
22:   $\hat{\pi}^{(t+1)}(\theta) = \sum_{i=1}^N w_{t+1}^{(i)} \mathbf{P}_\pi(\Theta = \theta; \chi_i^{(t+1)}, \nu_i^{(t+1)})$ 
   # Iterate
23:   $t \leftarrow t + 1$ 

```

---

2.18) and have the same spirit as the recursive update of the proposition's equation 2.10.

Equations 2.21 and 2.23 can be applied to the case where the likelihood function  $P_Y(y|\theta)$  is in the exponential family and  $\pi^{(0)}$  is its conjugate prior. The resulting algorithm (algorithm 3) has a particularly simple update rule for the belief parameters (see section 4 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 & Steyvers, 2009) or more complex (Findling et al., 2019) forms have also been employed to explain human behavior. Here we derived a simple particle filter for the general case of generative models of equations 2.2 to 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.

*2.2.4 Surprise-Modulation as a Framework for Other Algorithms.* Other existing algorithms (Adams & MacKay, 2007; Faraji et al., 2018; Fearnhead & Liu, 2007; Nassar et al., 2012, 2010) can also be formulated in the surprise-modulation framework of equations 2.9 and 2.12 (see section 4). Moreover, in order to allow for a transparent discussion and for fair comparisons in simulations, we extended the algorithms of Nassar et al. (2012, 2010) to a more general setting. Here we give a brief summary of the algorithms we considered. A detailed analysis is provided in section 4.5.

The algorithms of Nassar et al. (2012, 2010) were originally designed for a gaussian estimation task (see section 2.3 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 their original versions see Figures 12 and 13). Both algorithms have the same surprise-modulation as in our proposition (see 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 section 4 and the appendix).

**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 estimating  $\Theta_t$  (see section 4), on two tasks inspired by and closely related to real experiments (Behrens et al., 2007; Maheu, Dehaene, & Meyniel, 2019; Mars et al., 2008; Nassar et al., 2012, 2010; Ostwald et al., 2012): a gaussian and a categorical estimation task.

We compared our three novel algorithms VarSMiLe, Particle Filtering (pfN, 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 & 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 & Liu, 2007) (SORN, where  $N$  is the number of particles), our variant of Nassar et al.



(2010) (Nas10\*) and 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 section 4).

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 pfN and MPN. The algorithms Nas10\*, Nas12\*, and SMiLe, on the other hand, come from the human learning literature and are more biologically oriented.

**2.3.1 Gaussian Estimation Task.** The task is a generalized version of the experiment of Nassar et al. (2012, 2010). 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  $\sigma^2$ :  $y_{t+1} | \mu_{t+1} \sim \mathcal{N}(\mu_{t+1}, \sigma^2)$ . The mean  $\mu_{t+1}$  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 equations 2.2 to 2.4, with  $\pi^{(0)}(\mu_t) = \mathcal{N}(\mu_t; 0, 1)$  and  $P_Y(y_t | \mu_t) = \mathcal{N}(y_t; \mu_t, \sigma^2)$ . An example of the task is shown in Figure 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  $\sigma$  and  $p_c$ , we first tuned the free parameter of each algorithm— $m$  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 (pfN), 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 10 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  $\sigma$  is not estimated, and its actual value is used by all algorithms except the Leaky Integrator.

In Figure 2B we show the  $\text{MSE}[\hat{\theta}_t | R_t = n]$  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 ( $\text{MSE}[\hat{\theta}_t^{\text{Opt}} | R_t = n]$ ), with much lower memory requirements. VarSMiLe sometimes slightly outperforms the other two early after an environmental change (see Figure 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  $\sigma$  (see Figure 2B, left), but its performance is much worse for the case of high  $\sigma$  and low  $p_c$  (see Figure 2B, right). For SORN we observe a counter intuitive behavior in the regime of low  $\sigma$ : the inclusion of more particles leads to worse performance (see Figure 2B, left). At higher  $\sigma$  levels, the performance of SOR20 is close to optimal and better than the MP20 in later time

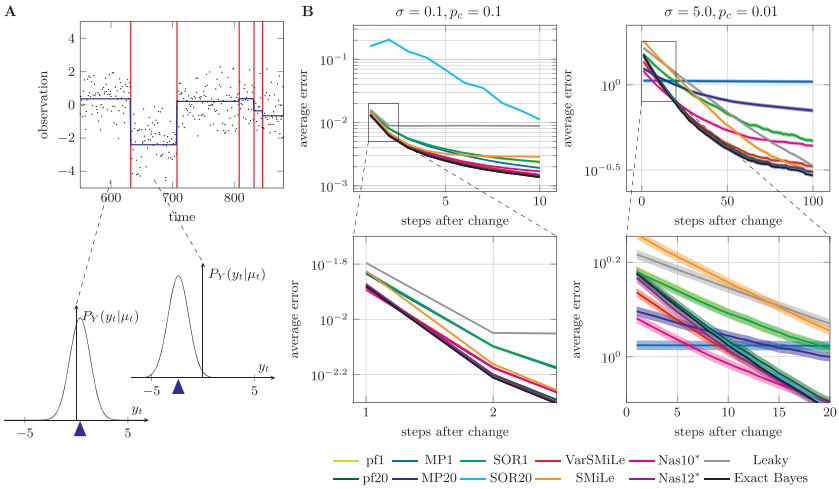


Figure 2: Gaussian estimation task: Transient performance after changes. (A) At each time step an observation (depicted as black dot) is drawn from a gaussian distribution  $\sim \exp(-(y_t - \mu_t)^2/2\sigma^2)$  with changing mean  $\mu_t$  (marked in blue) and known variance  $\sigma^2$  (lower left panels). At every change of the environment (marked with red lines), a new mean  $\mu_t$  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  $\mu_t$  at each time step  $n$  after an environmental change, that is, the average of  $\text{MSE}[\hat{\theta}_t | R_t = n]$  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: pfN: Particle Filtering with  $N$  particles; MPN: Message Passing with  $N$  particles; VarSMiLe: Variational SMiLe; SORN: Stratified Optimal Resampling with  $N$  particles (Fearnhead & 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).

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  $\hat{\pi}(\theta)$  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  $\sigma$  (see Figure 2B, right). The Nas10\*

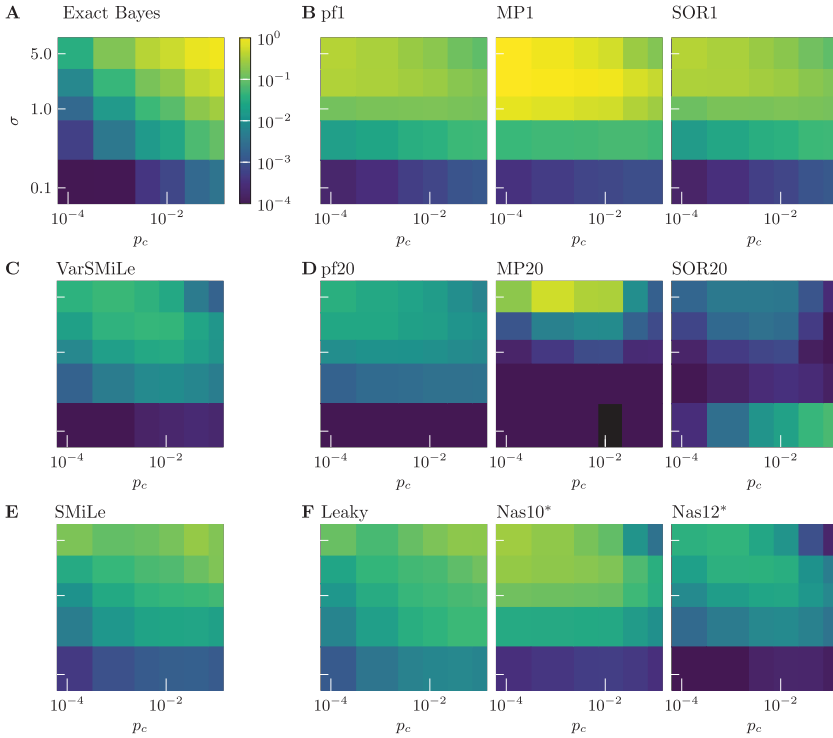


Figure 3: Gaussian estimation task: Steady-state performance. (A) Mean squared error of the Exact Bayes algorithm (i.e., optimal solution) for each combination of  $\sigma$  and  $p_c$  averaged over time. (B–F) Difference between the mean squared error of each algorithm and the optimal solution (of panel A), that is, the average of  $\Delta\text{MSE}[\hat{\theta}_t]$  over time. The color bar 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  $\text{MSE}$ . Abbreviations: See the Figure 2 caption.

performs well for low but not for higher values of  $\sigma$ . Despite the fact that a Particle Filter with one particle (pf1) is in expectation similar to Nas10\* and Nas12\* (see section 4), 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 Figure 3A, we have plotted the average of  $\text{MSE}[\hat{\theta}_t^{\text{Opt}}]$  of the Exact Bayes algorithm over the whole simulation time for each of the considered  $\sigma$  and  $p_c$  levels. The difference between the other algorithms and this benchmark is called  $\Delta\text{MSE}[\hat{\theta}_t]$  (see section 4) and is plotted in Figures 3C to 3F. All algorithms except for the SOR20 have lower average error

values for low  $\sigma$  and low  $p_c$  than high  $\sigma$  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  $\sigma$  and low  $p_c$ , whereas pf20 is more robust across levels of environmental parameters. The worst-case performance for pf20 is  $\Delta\text{MSE}[\hat{\theta}_t] = 0.033$  for  $\sigma = 5$  and  $p_c = 0.0001$ , and for SOR20, it is  $\Delta\text{MSE}[\hat{\theta}_t] = 0.061$  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 are the algorithms, Nas12\* and VarSMiLe. VarSMiLe exhibits its largest deviation from the optimal solution for high  $\sigma$  and low  $p_c$ , but is still more resilient compared to the MPN algorithms for this type of environment. Among the algorithms with only one unit of memory demands—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  $\sigma$  and  $p_c$  but deviates more from the optimal solution as these parameters increase (see Figure 3F). The SMiLe rule performs best at lower  $\sigma$ , that is, more deterministic environments.

A summary graph, where we collect the  $\Delta\text{MSE}[\hat{\theta}_t]$  across all levels of  $\sigma$  and  $p_c$ , is shown in Figure 4. We can see that pf20, Nas12\*, and VarSMiLe give the lowest worst case (lowest maximum value)  $\Delta\text{MSE}[\hat{\theta}_t]$  and are statistically better than the other eight algorithms (the error bars indicate the standard error of the mean across the ten random task instances).

**2.3.2 Categorical Estimation Task.** The task is inspired by the experiments of Behrens et al. (2007), Maheu et al. (2019), Mars et al. (2008), and Ostwald et al. (2012). The goal of the agent is to estimate the occurrence probability of five possible states. Each observation  $y_{t+1} \in \{1, \dots, 5\}$  is drawn from a categorical distribution with parameters  $\theta_{t+1} = \mathbf{p}_{t+1}$ , that is,  $y_{t+1} | \mathbf{p}_{t+1} \sim \text{Cat}(y_{t+1}; \mathbf{p}_{t+1})$ . When there is a change  $C_{t+1} = 1$  in the environment, the parameters  $\mathbf{p}_{t+1}$  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 equations 2.2 to 2.4, we thus have  $\pi^{(0)}(\mathbf{p}_t) = \text{Dir}(\mathbf{p}_t; s \cdot \mathbf{1})$  and  $P_Y(y_t | \mathbf{p}_t) = \text{Cat}(y_t; \mathbf{p}_t)$ . An illustration of this task is depicted in Figure 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. (2012, 2010) 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 10 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.

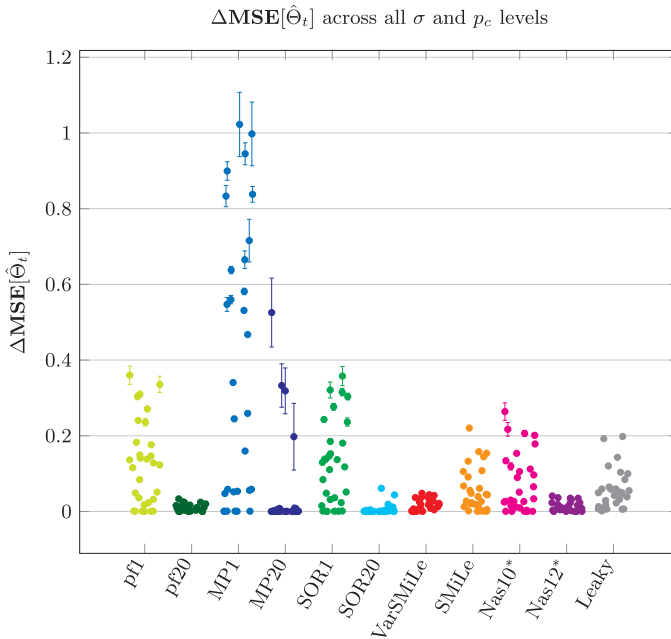


Figure 4: Gaussian estimation task: Steady-state performance summary. Difference between the mean squared error of each algorithm and the optimal solution (Exact Bayes), that is, the average of  $\Delta\text{MSE}[\hat{\theta}_t]$  over time, for all combinations of  $\sigma$  and  $p_c$  together. For each algorithm we plot the 30 values (5  $\sigma$  times 6  $p_c$  values) of Figure 3 with respect to randomly jittered values in the  $x$ -axis. The color coding is the same as in Figure 2. The error bars 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 Figure 2 caption.

The Particle Filter pf20, the MP20, and the SOR20 have a performance closest to that of Exact Bayes, the optimal solution (see Figure 5B). VarSMiLe is the next in the ranking, with a behavior after a change similar to the gaussian task. pf20 performs better for  $s > 2$ , and MP20 performs better for  $s \leq 2$  (see Figure 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  $\Delta\text{MSE}[\hat{\theta}_t] = 8.16 \times 10^{-5}$  for  $s = 2$  and  $p_c = 0.01$ , and the worst-case performance for pf20 is  $\Delta\text{MSE}[\hat{\theta}_t] = 0.0048$  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 MP20, the highest deviations from the optimal solution are observed for medium stochasticity levels (see Figures 6B to 6F). When the

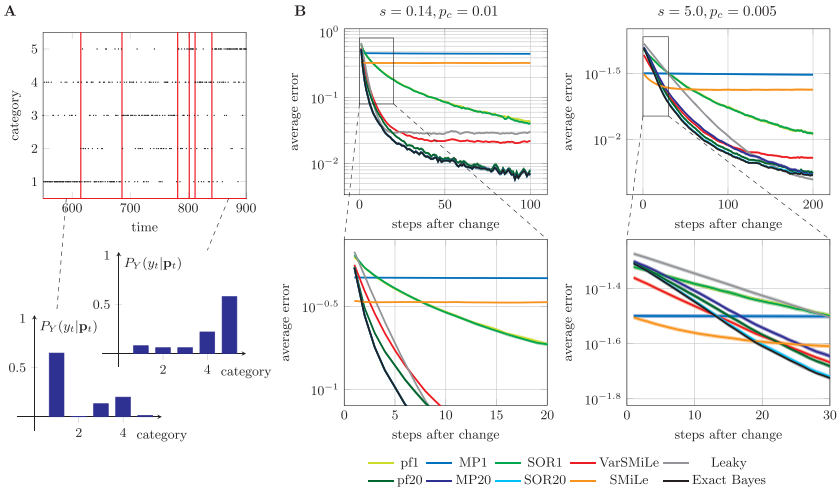


Figure 5: Categorical estimation task: Transient performance after changes. (A) At each time step, the agent sees one out of five possible categories (black dots) drawn from a categorical distribution with parameters  $\mathbf{p}_t$ . Occasional abrupt changes happen with probability  $p_c$  and are marked with red lines. After each change, a new  $\mathbf{p}_t$  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  $\mathbf{p}_t$  at each time step  $n$  after an environmental change, that is, the average of  $\text{MSE}[\hat{\theta}_t | R_t = n]$  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: pfN: Particle Filtering with  $N$  particles; MPN: Message Passing with  $N$  particles; VarSMiLe: Variational SMiLe; SORN: Stratified Optimal Resampling with  $N$  particles (Fearnhead & Liu, 2007); SMiLe: Faraji et al. (2018); Leaky: Leaky Integrator; Exact Bayes: Adams and MacKay (2007).

environment is nearly deterministic (e.g.,  $s = 0.001$  so that the parameter vectors  $\mathbf{p}_t$  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  $\Delta\text{MSE}[\hat{\theta}_t]$  across all levels of  $s$  and  $p_c$ , is in Figure 7. The algorithms with the lowest “worst case” are SOR20 and pf20. The top four algorithms SOR20, pf20, MP20, and VarSMiLe are significantly better than the others (the error bars indicate the standard error of the mean

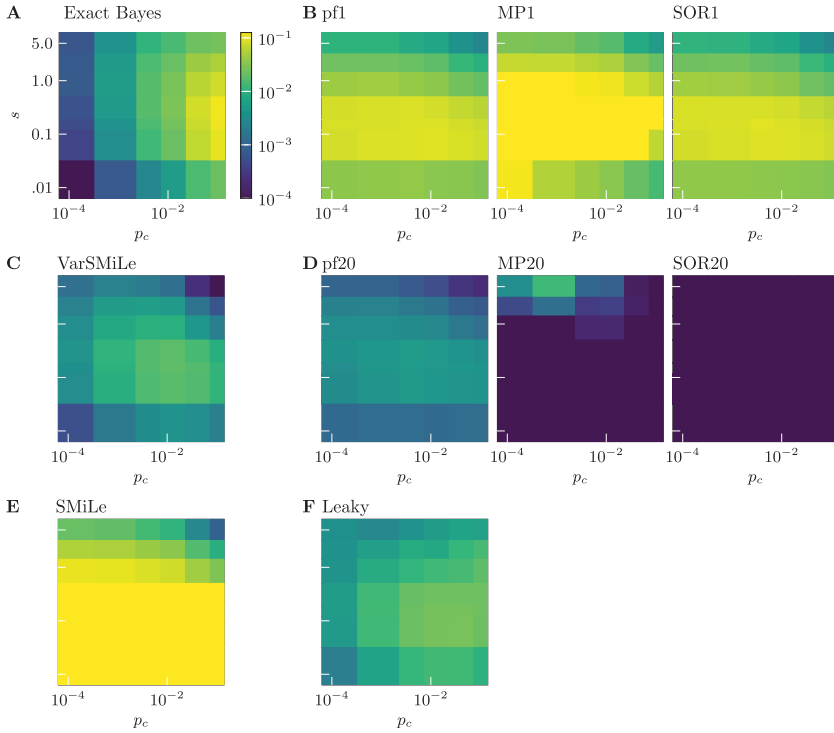


Figure 6: Categorical estimation task: Steady-state performance. (A) Mean squared error of the Exact Bayes algorithm (i.e., optimal solution) for each combination of environmental parameters  $s$  and  $p_c$  averaged over time. (B–F) Difference between the mean squared error of each algorithm and the optimal solution (of panel A), that is, the average of  $\Delta\text{MSE}[\hat{\Theta}_t]$  over time. The color bar of panel A applies to these panels as well. Abbreviations: See the Figure 5 caption.

across the 10 random task instances), whereas MP1 and SMiLe have the largest error with a maximum at 0.53.

**2.3.3 Summary of Simulation Results.** In summary, our simulation results of the two tasks collectively suggest that our Particle Filtering (pfN) 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 behavior is more consistent across tasks. Finally, among the algorithms with memory demands of one unit, VarSMiLe performs best.

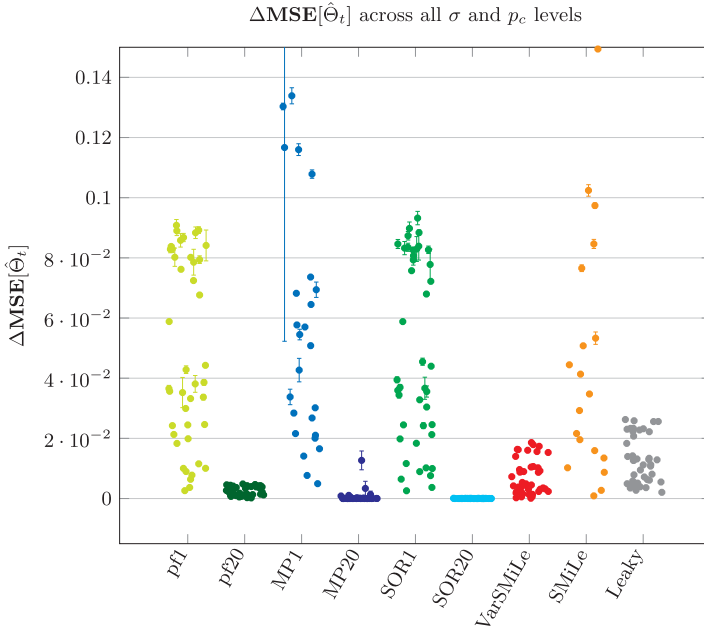


Figure 7: Categorical estimation task: Steady-state performance summary. Difference between the mean squared error of each algorithm and the optimal solution (Exact Bayes), that is, the average of  $\Delta\text{MSE}[\hat{\theta}_t]$  over time, for all combinations of  $s$  and  $p_c$  together. For each algorithm, we plot the 42 values (7  $s$  times 6  $p_c$  values) of Figure 6 with respect to randomly jittered values in the  $x$ -axis. The color coding is the same as in Figure 5. The error bars 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 Figure 5 caption. Note that MP1 and SMiLe are out of bound with a maximum at 0.53.

*2.3.4 Robustness against Suboptimal Parameter Choice.* In all algorithms we considered, the environment’s hyperparameters are assumed to be known. We can distinguish between two types of hyperparameters in our generative model: the parameters of the likelihood function (e.g.,  $\sigma$  in the gaussian task), and the parameters of the conjugate prior (e.g.,  $s$  in the categorical task). Hyperparameters of the first type can be added to the parameter vector  $\theta$  and be inferred with the same algorithm. However, learning the second type of hyperparameters is not straightforward. By assuming that these hyperparameters are learned more slowly than  $\theta$ , 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 hyperparameter estimation can be found in Doucet and Tadić (2003), George and Doss (2017), Liu and West (2001), and Wilson et al. (2010).

When the hyperparameters 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, that is, the resulting MSE for the case that the Exact Bayes' parameter is tuned for the actual  $p_c$ .

More precisely, if we denote as  $\text{MSE}[\hat{\Theta}_t; p'_c, p_c]$  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  $\text{MSE}[\hat{\Theta}_t; p'_c, p_c] - \text{MSE}[\hat{\Theta}_t^{\text{Opt}}, p_c]$ , over time; note that the second term is equal to  $\text{MSE}[\hat{\Theta}_t; p_c, p_c]$  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 Figure 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  $p'_c = 0.04$  (see Figure 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 the changes are still 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  $p'_c = 0.004$  (see Figure 8B). In this case, the performance of all algorithms deteriorates strongly when the actual  $p_c$  is higher than the assumed one.

However, for  $\sigma = 5$  (see Figures 8C and 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  $\sigma$  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  $\sigma = 5$ ,  $p'_c = 0.004$  (see Figure 8D). The SMiLe and the Leaky Integrator outperform the other algorithms for higher  $p'_c$  if  $p_c < p'_c$  (see Figure 8C). A potential reason is that

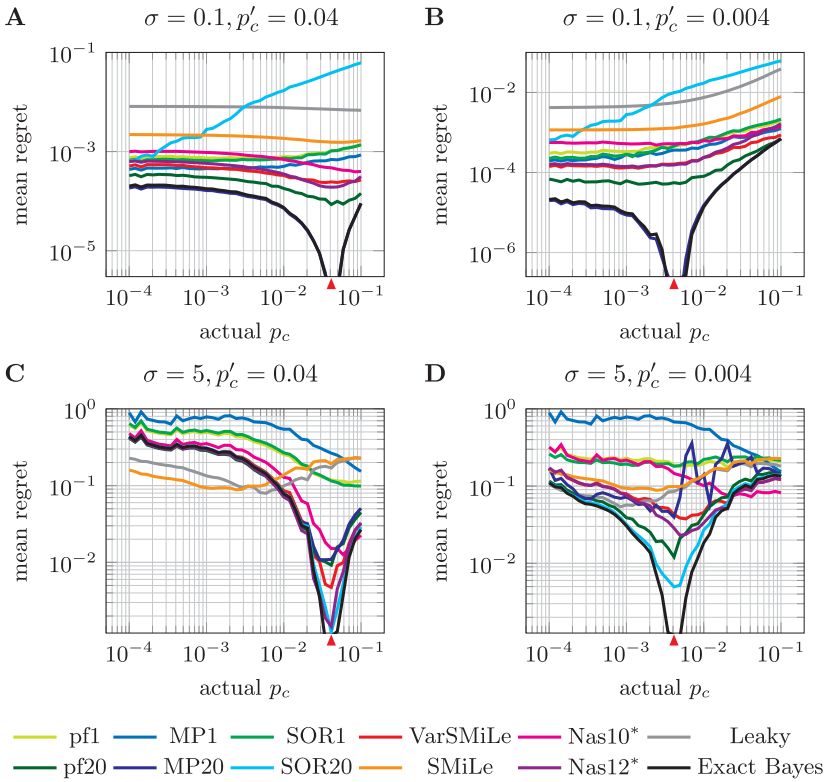


Figure 8: Robustness to mismatch between actual and assumed probability of changes for the gaussian estimation task. The mean regret is the mean squared error obtained with assumed change probability  $p'_c$  minus the mean squared error obtained with the optimal parameter choice of Exact Bayes for the given actual  $p_c$ , that is, the average of the quantity  $\text{MSE}[\hat{\Theta}_t; p'_c, p_c] - \text{MSE}[\hat{\Theta}_t^{\text{Opt}}, p_c]$  over time. A red triangle marks the  $p'_c$  value each algorithm with 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 Figure 2 caption.

the optimal behavior 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 (see Figure 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

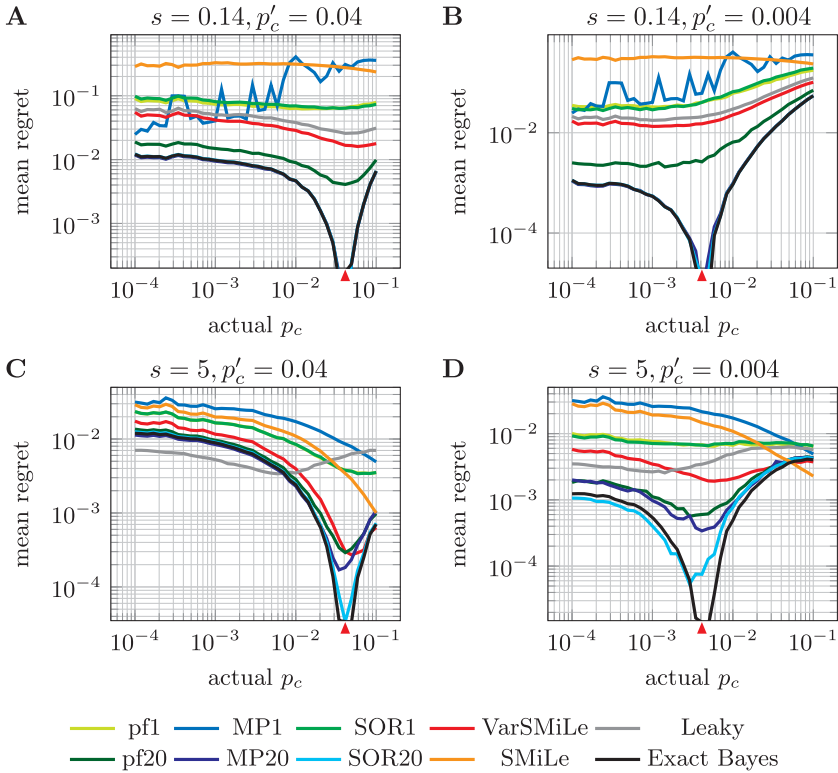


Figure 9: Robustness to mismatch between actual and assumed probability of changes for the categorical estimation task. The mean regret is the mean squared error obtained with assumed change probability  $p'_c$  minus the mean squared error obtained with the optimal parameter choice of Exact Bayes for the given actual  $p_c$ , that is, the average of the quantity  $\mathbf{MSE}[\hat{\theta}_t; p'_c, p_c] - \mathbf{MSE}[\hat{\theta}_t^{\text{Opt}}, p_c]$  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 Figure 5 caption.

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 (see Figure 9).

**2.4 Experimental Prediction.** It has been experimentally shown that some important behavioral and physiological indicators statistically

correlate with a measure of surprise or a prediction error. Examples of such indicators are the pupil diameter (Joshi & Gold, 2020; Nassar et al., 2012; Preuschoff et al., 2011), the amplitude of the P300, N400, and MMN components of EEG (Kopp & Lange, 2013; Lieder, Daunizeau, Garrido, Friston, & Stephan, 2013; Mars et al., 2008; Meyniel, Maheu, & Dehaene, 2016; Modirshanechi et al., 2019; Musiolek, Blankenburg, Ostwald, & Rabovsky, 2019; Ostwald et al., 2012), the amplitude of MEG in specific time windows (Maheu et al., 2019), BOLD responses in fMRI (Kononov & Krajbich, 2018; Loued-Khenissi, Pfeuffer, Einhäuser, & Preuschoff, 2020), and reaction time (Huettel, Mack, & McCarthy, 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  $S_{Sh}$ . However, as we show in this section, as long as there is an uninformative prior over observations, Shannon Surprise  $S_{Sh}$  is just an invertible function of our modulated adaptation rate  $\gamma$  and hence an invertible function of the Bayes Factor Surprise  $S_{BF}$ . Thus, based on the results of previous works (Meyniel et al., 2016; Modirshanechi et al., 2019; Nassar et al., 2012, 2010; Ostwald et al., 2012), which always used uninformative priors, one cannot determine whether the physiological and behavioral indicators we have noted correlate with  $S_{Sh}$  or  $S_{BF}$ .

In this section, we first investigate the theoretical differences between the Bayes Factor Surprise  $S_{BF}$  and Shannon Surprise  $S_{Sh}$ . 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 determine whether physiological or behavioral measurements are signatures of  $S_{BF}$  or of  $S_{Sh}$ .

*2.4.1 Theoretical Difference between  $S_{BF}$  and  $S_{Sh}$ .* Shannon Surprise (Shannon, 1948) is defined as

$$S_{Sh}(y_{t+1}; \pi^{(t)}) = -\log(\mathbf{P}(y_{t+1}|y_{1:t})), \quad (2.24)$$

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 Figure 1A, we find  $S_{Sh}(y_{t+1}; \pi^{(t)}) = -\log((1 - p_c)P(y_{t+1}; \pi^{(t)}) + p_cP(y_{t+1}; \pi^{(0)}))$ . While the Bayes Factor Surprise  $S_{BF}$  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 section 4 for derivation) the adaptation rate  $\gamma_{t+1}$  as a function of the “difference in Shannon Surprise,”

$$\gamma_{t+1} = p_c \exp(\Delta S_{Sh}(y_{t+1}; \pi^{(t)}, \pi^{(0)})),$$

$$\text{where } \Delta S_{Sh}(y_{t+1}; \pi^{(t)}, \pi^{(0)}) = S_{Sh}(y_{t+1}; \pi^{(t)}) - S_{Sh}(y_{t+1}; \pi^{(0)}), \quad (2.25)$$

where  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}^{(t+1)}, m)$  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 the prior beliefs. Note that if the prior is uninformative then the second term in equation 2.25 is a constant with respect to observations, and  $\gamma_{t+1}$  is an invertible function of  $\mathbf{S}_{\text{Sh}}$ , thus indistinguishable from  $\mathbf{S}_{\text{Sh}}$ . We next exploit differences between  $\mathbf{S}_{\text{BF}}$  and  $\mathbf{S}_{\text{Sh}}$  to formulate our experimentally testable predictions.

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

1. Fix the hyperparameters  $\sigma^2$  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 behavioral indicator  $M_t$ , for example, pupil diameter (Nassar et al., 2012, 2010).
3. At each time  $t$ , after observing  $y_t$ , ask the subject to predict the next observation  $\hat{y}_{t+1}$  and his or her confidence  $C_t$  about that prediction.

Note that the only difference between our task and the task of Nassar et al. (2012, 2010) is the choice of prior for  $\theta$  (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.

**2.4.3 Prediction 1.** Based on the results of Nassar et al. (2012, 2010), 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 the standard deviation of  $\hat{\theta}_t$ , denoted as  $\hat{\sigma}_t$ . 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 ( $s_t = -1$ ). Figure 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 the 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

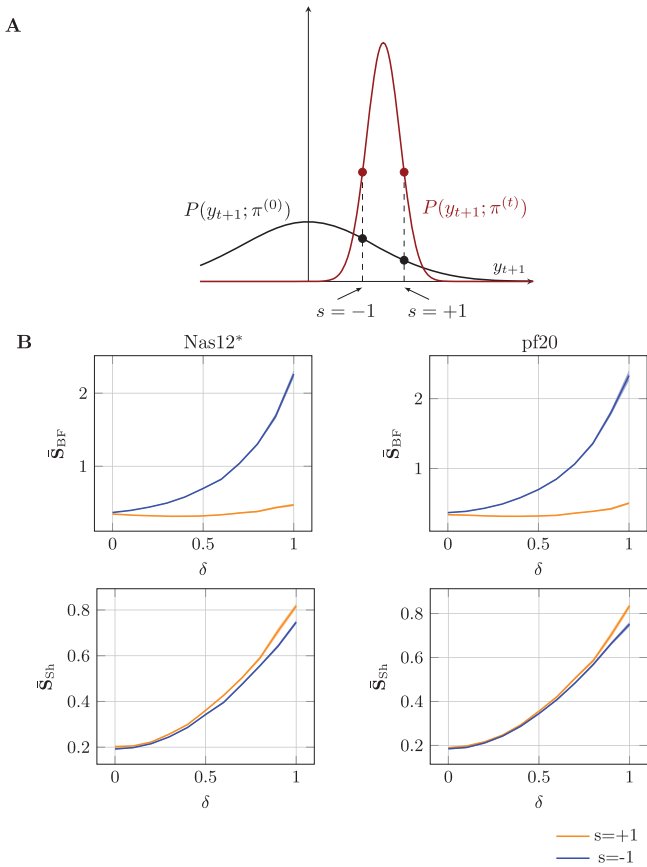


Figure 10: Experimental prediction 1. (A) Schematic of the task for the case of a gaussian belief. The distributions of  $y_{t+1}$  under the prior belief  $\pi^{(0)}$  and the current belief  $\pi^{(t)}$  are shown by black and red curves, respectively. Two possible observations with equal absolute prediction error  $\delta$  but opposite sign bias  $s$  are indicated by dashed lines. The two observations are equally probable under  $\pi^{(t)}$  but not under  $\pi^{(0)}$ .  $\bar{S}_{BF}$  is computed as the ratio between the red and black dots for a given observation, whereas  $\bar{S}_{Sh}$  is a function of the weighted sum of the two. This phenomenon is the basis of our experimental prediction. (B) The average surprise values  $\bar{S}_{Sh}(\hat{\theta} = 1, \delta, s = \pm 1, \sigma_c = 0.5)$  and  $\bar{S}_{BF}(\hat{\theta} = 1, \delta, s = \pm 1, \sigma_c = 0.5)$  over 20 subjects (each with 500 observations) are shown for two different learning algorithms (Nas12\* and pf20). The mean  $\bar{S}_{BF}$  is higher for negative sign bias (marked in blue) than for positive sign bias (marked in orange). The opposite is observed for the mean  $\bar{S}_{Sh}$ . This effect increases with increasing values of prediction error  $\delta$ . 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 section 4 for details).

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 section 4. 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  $(\hat{y}, \delta, s, C)$  is qualitatively and quantitatively different.

In order to prove and illustrate our prediction, we consider each subject as an agent enabled with one of the learning algorithms that we discussed. Similar to the 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  $\sigma_C$  (corresponding to the subjects' confidence value  $C$ ), and a sign bias  $s \in \{-1, 1\}$ , we can compute the average Shannon Surprise  $\mathbf{S}_{\text{Sh}}(y_t; \hat{\pi}^{(t-1)})$  and the average Bayes Factor Surprise  $\mathbf{S}_{\text{BF}}(y_t; \hat{\pi}^{(t-1)})$  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 section 4) and in simulations (see Figure 10B and section 4) that for any value of  $\hat{\theta}$ ,  $\delta$ , and  $\sigma_C$ , 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,  $\bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C) < \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)$ , for the Bayes Factor Surprise. Moreover, this effect increases with increasing  $\delta$ .

It should be noted that such an effect is due to the essential difference of  $\mathbf{S}_{\text{Sh}}$  and  $\mathbf{S}_{\text{BF}}$  in using the prior belief  $\pi^{(0)}(\theta)$ . Our experimental prediction is theoretically provable for the cases that each subject's belief  $\hat{\pi}^{(t)}$  is a gaussian distribution, which is the case if they employ VarSMiLe, Nas10\*, Nas12\*, pf1, MP1, or Leaky Integrator as their learning rule (see section 4). 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 Figure 10B and section 4). 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 the set of hypotheses shown in Table 1.

*2.4.4 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., 2012, 2010) or with the goal of

Table 1: Experimental Hypotheses and Predictions 1.

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

Note:  $\Delta\bar{M}_1(\hat{\theta}, \delta, C)$  stands for  $\bar{M}_1(\hat{\theta}, \delta, s = +1, C) - \bar{M}_1(\hat{\theta}, \delta, s = -1, C)$ .

maximizing the likelihood of subject’s prediction  $\hat{\pi}^{(t)}(\hat{y}_{t+1})$ . Our prediction is independent of the learning algorithm, but in an actual experiment, we recommend using model selection to find the model that fits the human data best.

Having a fitted model, we can compute the probabilities  $P(y_{t+1}; \hat{\pi}^{(t)})$  and  $P(y_{t+1}; \hat{\pi}^{(0)})$ . For the case that these probabilities are equal,  $P(y_{t+1}; \hat{\pi}^{(t)}) = P(y_{t+1}; \hat{\pi}^{(0)}) = p$ , the Bayes Factor Surprise  $\mathbf{S}_{\text{BF}}$  is equal to 1, independent of the value of  $p$  (see equation 2.7). However, the Shannon Surprise  $\mathbf{S}_{\text{Sh}}$  is equal to  $-\log p$  and varies with  $p$ . Figure 11A shows a schematic for the case that both current and prior beliefs are gaussian distributions. Two cases for which we have  $P(y_{t+1}; \hat{\pi}^{(t)}) = P(y_{t+1}; \hat{\pi}^{(0)}) = p$ , 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  $P(y_{t+1}; \hat{\pi}^{(t)}) \approx p$  and  $P(y_{t+1}; \hat{\pi}^{(0)}) \approx p$ , which we denote as  $\bar{M}_2(p)$ ; the index 2 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 behavior is consistent with  $\mathbf{S}_{\text{BF}}$ , whereas if it decreases by increasing  $p$ , it can be a signature of  $\mathbf{S}_{\text{Sh}}$ . Our second experimental prediction can be summarized as the two hypotheses shown in Table 2. Note that in contrast to our first prediction, with the assumption that the standard deviation of the prior belief is fitted using the behavioral 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  $\bar{\mathbf{S}}_{\text{BF}}(p)$  and  $\bar{\mathbf{S}}_{\text{Sh}}(p)$  for the time points with  $P(y_{t+1}; \hat{\pi}^{(t)}) \approx p$  and  $P(y_{t+1}; \hat{\pi}^{(0)}) \approx p$  (see section 4 for details). The results of the simulation are shown in Figure 11B.

### 3 Discussion

We have shown that performing exact Bayesian inference on a generative world model naturally leads to a definition of surprise and a



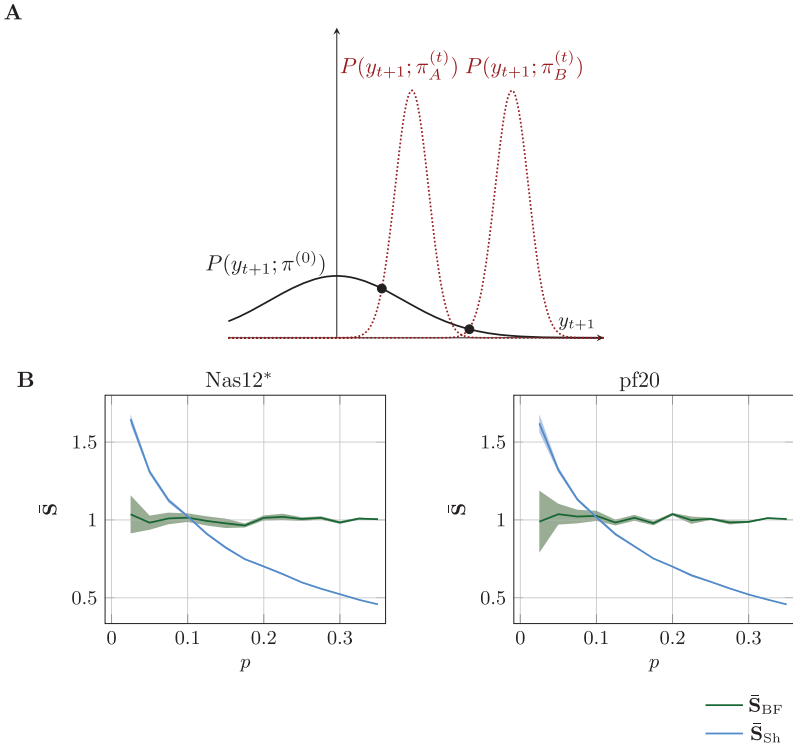


Figure 11: Experimental prediction 2. (A) Schematic of the task for the case of a gaussian belief. The probability distribution of observations under the prior belief is shown by the solid black curve. Two different possible current beliefs (determined by the letters *A* and *B*) are shown by dashed red curves. The intersections of the dashed red curves with the prior belief determine observations whose  $\mathcal{S}_{BF}$  is same and equal to one, but their  $\mathcal{S}_{Sh}$  is a function of their probabilities under the prior belief  $p$ . (B) The average surprise values  $\bar{\mathcal{S}}_{Sh}(p)$  and  $\bar{\mathcal{S}}_{BF}(p)$  over 20 subjects (each with 500 observations) are shown for two different learning algorithms (Nas12\* and pf20). The mean  $\mathcal{S}_{BF}$  is constant (equal to 1) and independent of  $p$ , whereas the mean  $\mathcal{S}_{Sh}$  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 section 4 for details).

surprise-modulated adaptation rate. We have proposed three approximate algorithms (VarSMiLe, MPN, and pfN) for learning in nonstationary environments, which all exhibit the surprise-modulated adaptation rate of the

Table 2: Experimental Hypotheses and Predictions 2.

Hypothesis	Prediction
The indicator reflects $S_{BF}$	$\frac{\partial \bar{M}_2(p)}{\partial p} = 0$
The indicator reflects $S_{Sh}$	$\frac{\partial \bar{M}_2(p)}{\partial p} < 0$

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 & MacKay, 2007) and are more resilient across different environments compared to methods with similar memory demands (Faraji et al., 2018; Fearnhead & Liu, 2007; Nassar et al., 2012, 2010).

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 that can be modeled with a generative model similar to the one in Figure 1. We then discuss further our results and propose directions for future work on surprise-based learning.

**3.1 Biological Interpretation.** Humans are able to quickly adapt to changes (Behrens et al., 2007; Nassar et al., 2012, 2010), but human behavior is also often observed to be suboptimal, compared to the normative approach of exact Bayesian inference (Glaze et al., 2015; Mathys, Daunizeau, Friston, & Stephan, 2011; Nassar et al., 2010; Prat-Carrabin, Wilson, Cohen, & Da Silveira, 2020; Wilson, Nassar, & Gold, 2013). In general, biological agents have limited resources and possibly inaccurate assumptions about hyperparameters, yielding suboptimal behavior, as we also see with our algorithms whose accuracies degrade with a suboptimal choice of hyperparameters. Performance also deteriorates with a decreasing number of particles in the sampling-based algorithms, which might be another possible explanation of suboptimal human behavior. Previously, Particle Filtering has been shown to explain the behavior 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 & Rao, 2014; Kutschireiter et al., 2017; Legenstein & Maass, 2014; Shi & Griffiths, 2009) on which, a neural

model of pfn 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  $\mathbf{S}_{\text{BF}}$  is related to the body of literature on neo-Hebbian three-factor learning rules (Frémaux & 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 (Yu, 2012; Yu & 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  $\gamma(\mathbf{S}_{\text{BF}}, m)$ . This may offer novel interpretations of behavioral and neurophysiological data and help in understanding how three-factor learning computations may be implemented in the brain.

### 3.2 Related Work.

*3.2.1 Exact Bayesian Inference.* As already described in section 2.2.2, for the generative model in Figure 1, it is possible to find an exact online Bayesian update of the belief using a message-passing algorithm (Adams & 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 & MacKay, 2007) or stratified resampling (Fearnhead & Liu, 2007) allow reducing the computational complexity. The former has a variable number of particles in time, and the latter needs solving a complicated nonlinear equation at each time step in order to reduce the number of particles to  $N$  (called SORN in section 2).

Our message-passing algorithm with a finite number of particles (messages)  $N$  (MPN; see algorithm 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 equations 2.19 and 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 Figure 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), and Ghahramani & Hinton (2000) for other variations of switching state-space models and examples of approximate inference over them.

*3.2.2 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 & Meyniel, 2019; Meyniel et al., 2016; Ryalı et al., 2018; Yu & Cohen, 2009). This update rule was found to be biologically plausible and consistent with human behavioral data (Meyniel et al., 2016; Yu & 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 behavior. 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 & MacKay, 2007; Fearnhead & Liu, 2007), or by considering an adaptive leak parameter (Nassar et al., 2012, 2010). We have shown that the two algorithms of Nassar et al. (2012, 2010) 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  $\gamma$  can be interpreted as a surprise-modulated leak parameter.

*3.2.3 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 (see equations 2.14 and 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 section 4).

Other approaches use different generative models, such as conditional sampling of the parameters also when there is a change (Glaze et al., 2015; Yu & Dayan, 2005), a deeper hierarchy without fixed change probability  $p_c$  (Wilson et al., 2010), or drift in the parameters (Gershman, Radulescu, Norman, & Niv, 2014; Mathys et al., 2011). A recent work shows that inference on a simpler version the generative model of Figure 1, with no change points but with a noisy inference style, can explain human behavior 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 & Baldi, 2006; Schmidhuber, 2010; Storck, Hochreiter, & Schmidhuber, 1995). Moreover, another recent work (Prat-Carrabin et al., 2020) shows that approximate sampling algorithms (like Particle Filtering) can

explain human behavior better than their alternatives in tasks closely related to the generative model of Figure 1. The signal processing literature provides further methods to address the problem of learning in nonstationary 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), and Özkan et al. (2013) for a few recent examples.

**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 Bayesian arguments for a particular experimental task, for example, when samples are drawn from a gaussian distribution (Nassar et al., 2012, 2010). 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 a fixed  $N$  number of messages) as well as some previously developed ones (Adams & MacKay, 2007; Fearnhead & Liu, 2007; Nassar et al., 2012, 2010) 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., 2014; Mathys et al., 2011; Nassar et al., 2012, 2010; Prat-Carrabin et al., 2020), the notion of Bayes Factor Surprise and the way it modulates learning (see equations 2.12 and 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) and Nassar, Bruckner, and Frank (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 leads to a surprise-modulated adaptation rate  $\gamma(\mathbf{S}_{\text{BF}}, m)$ , with the same definition of  $\mathbf{S}_{\text{BF}}$ , where the trade-off is not between integrating and resetting, but between integrating and ignoring the new observation.<sup>1</sup>

---

<sup>1</sup>Liakoni, V., Lehmann, M. P., Modirshanechi, A., Brea, J., Herzog, M. H., Gerstner, W., & Preuschoff, K. *Dissociating brain regions encoding reward prediction error and surprise*. Manuscript in preparation.

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 Figure 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 Fox, Sudderth, Jordan, and Willsky (2011), Gershman, Monfils, Norman, and Niv (2017), Gershman et al. (2014) and are out of the scope of this article.

**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 reflect how unexpected an event is and should modulate learning. Surprising events indicate that our belief is far from the real world and suggest updating our model of the world, or, for a large surprise, simply forgetting it. Forgetting is the same as returning to the prior belief. However, an observation  $y_{t+1}$  can be unexpected under both the prior  $\pi^{(0)}$  and the current beliefs  $\pi^{(t)}$ . 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  $P(y_{t+1}; \pi^{(t)})$  and its probability under the prior belief  $P(y_{t+1}; \pi^{(0)})$ .

The definition of the Bayes Factor Surprise  $\mathbf{S}_{\text{BF}}$  as the ratio of  $P(y_{t+1}; \pi^{(t)})$  and  $P(y_{t+1}; \pi^{(0)})$  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 that 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 that is more expected under the prior belief is more surprising, signaling that forgetting may be beneficial.

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

$$\mathbf{S}_{\text{BF}}(y_{t+1}; \pi^{(t)}) = \frac{P(y_{t+1}; \pi^{(0)})}{P(y_{t+1}; \pi^{(t)})} = \frac{\mathbb{E}_{\pi^{(0)}}[P_Y(y_{t+1}|\Theta)]}{\mathbb{E}_{\pi^{(t)}}[P_Y(y_{t+1}|\Theta)]}. \quad (3.1)$$

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 are generated with another generative model (e.g., the real world),  $\mathbf{S}_{\text{BF}}$  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 ( $\gamma$ ) 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  $S_{Ba}$  (Itti & Baldi, 2006; Schmidhuber, 2010; Storck et al., 1995) and the Confidence Corrected Surprise  $S_{CC}$  (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  $S_{Ba}$  and  $S_{CC}$ . However, one can hypothesize that  $S_{Ba}$  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  $S_{CC}$  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  $M_1(\hat{\theta}, \delta, s = +1, C) = \bar{M}_1(\hat{\theta}, \delta, s = -1, C)$  in our first experimental prediction would be consistent with the corresponding behavioral or physiological indicator reflecting the  $S_{CC}$ .

**3.5 Difference in Shannon Surprise, an Alternative Perspective.** Following our formal comparison in section 2.4,  $S_{BF}$  can be expressed as a deterministic function of the difference in Shannon Surprise as

$$S_{BF} = \frac{(1 - p_c)e^{\Delta S_{Sh}}}{1 - p_c e^{\Delta S_{Sh}}}. \quad (3.2)$$

All of our theoretical results can be rewritten by replacing  $S_{BF}$  with this function of  $\Delta S_{Sh}$ . Moreover, because there is a one-to-one mapping between  $S_{BF}$  and  $\Delta S_{Sh}$ , from a systemic point of view, it is not possible to specify whether the brain computes the former or the latter by analysis of behavioral 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 & 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.

**3.6 Future Directions.** A natural continuation of our study is to test our experimental predictions in human behavior and physiological signals in order to investigate which measures of surprise are used by the brain. In a similar direction, our approximate learning algorithms can be evaluated on human behavioral data from experiments that use a similar generative

model (Behrens et al., 2007; Glaze et al., 2015; Heilbron & Meyniel, 2019; Nassar et al., 2012, 2010; Wilson et al., 2013; Yu & 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 nonstationary environments. In recent years, there has been growing interest in adaptive or continually learning agents in changing environments in the form of continual learning and meta-learning (Lomonaco, Desai, Culurciello, & Maltoni, 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  $S_{BF}$  and a learning rate  $\gamma(S_{BF})$  into a reinforcement learning agent would be an interesting future direction.

## 4 Methods

---

### 4.1 Proof of the proposition. By definition,

$$\pi^{(t+1)}(\theta) \equiv \mathbf{P}(\Theta_{t+1} = \theta | y_{1:t+1}). \quad (4.1)$$

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

$$\pi^{(t+1)}(\theta) = \frac{P_Y(y_{t+1}|\theta)\mathbf{P}(\Theta_{t+1} = \theta | y_{1:t})}{\mathbf{P}(y_{t+1}|y_{1:t})}. \quad (4.2)$$

By marginalization over the hidden state  $C_{t+1}$ , the second factor in the numerator of equation 4.2 can be written as

$$\mathbf{P}(\Theta_{t+1} = \theta | y_{1:t}) = (1 - p_c)\pi^{(t)}(\theta) + p_c\pi^{(0)}(\theta). \quad (4.3)$$

The denominator in equation 4.2 can be written as

$$\begin{aligned} \mathbf{P}(y_{t+1}|y_{1:t}) &= \int P_Y(y_{t+1}|\theta)\mathbf{P}(\Theta_{t+1} = \theta | y_{1:t})d\theta \\ &= (1 - p_c) \int P_Y(y_{t+1}|\theta)\pi^{(t)}(\theta)d\theta + p_c \int P_Y(y_{t+1}|\theta)\pi^{(0)}(\theta)d\theta \\ &= (1 - p_c)P(y_{t+1}; \pi^{(t)}) + p_cP(y_{t+1}; \pi^{(0)}), \end{aligned} \quad (4.4)$$



where we used the definition in equation 2.6. Using these two expanded forms, equation 4.2 can be rewritten as

$$\pi^{(t+1)}(\theta) = \frac{P_Y(y_{t+1}|\theta)\left((1-p_c)\pi^{(t)}(\theta) + p_c\pi^{(0)}(\theta)\right)}{(1-p_c)P(y_{t+1}; \pi^{(t)}) + p_cP(y_{t+1}; \pi^{(0)})}. \quad (4.5)$$

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

$$P(\theta|y_{t+1}) = \frac{P_Y(y_{t+1}|\theta)\pi^{(0)}(\theta)}{P(y_{t+1}; \pi^{(0)})}. \quad (4.6)$$

Then, we can write equation 4.5 as

$$\begin{aligned} \pi^{(t+1)}(\theta) &= \frac{(1-p_c)P(y_{t+1}; \pi^{(t)})\pi_B^{(t+1)}(\theta) + p_cP(y_{t+1}; \pi^{(0)})P(\theta|y_{t+1})}{(1-p_c)P(y_{t+1}; \pi^{(t)}) + p_cP(y_{t+1}; \pi^{(0)})} \\ &= \frac{\pi_B^{(t+1)}(\theta) + \frac{p_c}{1-p_c} \frac{P(y_{t+1}; \pi^{(0)})}{P(y_{t+1}; \pi^{(t)})} P(\theta|y_{t+1})}{1 + \frac{p_c}{1-p_c} \frac{P(y_{t+1}; \pi^{(0)})}{P(y_{t+1}; \pi^{(t)})}} \\ &= (1-\gamma_{t+1})\pi_B^{(t+1)}(\theta) + \gamma_{t+1}P(\theta|y_{t+1}), \end{aligned} \quad (4.7)$$

where  $\pi_B^{(t+1)}(\theta)$  is defined in equation 2.8, and

$$\gamma_{t+1} = \gamma\left(\mathbf{S}_{\text{BF}}(y_{t+1}; \pi^{(t)}), \frac{p_c}{1-p_c}\right) \quad (4.8)$$

with  $\mathbf{S}_{\text{BF}}$  defined in equation 2.7, and  $\gamma(\mathbf{S}, m)$  defined in equation 2.9. Thus, our calculation yields a specific choice of surprise ( $\mathbf{S} = \mathbf{S}_{\text{BF}}$ ) and a specific value for the saturation parameter  $m = \frac{p_c}{1-p_c}$ .

## 4.2 Derivation of the Optimization-Based Formulation of VarSMiLe:

**Algorithm 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).

*4.2.1 Derivation of the Update Rule.* Consider the general form of the following variational optimization problem:

$$\begin{aligned} q^*(\theta) &= \operatorname{argmin} \mathbf{D}_{\text{KL}}[q(\theta)||p_1(\theta)] \\ q(\theta) \text{ s.t. } &\mathbf{D}_{\text{KL}}[q(\theta)||p_2(\theta)] < B \text{ and } \mathbb{E}_q[1] = 1, \end{aligned} \quad (4.9)$$

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

$$q^*(\theta) = \begin{cases} p_2(\theta) & \text{if } B = 0 \\ p_1(\theta) & \text{if } B = \mathbf{D}_{KL}[p_1(\theta)||p_2(\theta)]. \end{cases} \quad (4.10)$$

Note that the Kullback–Leibler divergence is a convex function with respect to its first argument— $q$  in our setting. Therefore, both the objective function and the constraints of the optimization problem in equation 4.9 are convex. For convenience, we assume that the parameter space for  $\theta$  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 4.9 can be rewritten as

$$\begin{aligned} q^*(\theta) &= \operatorname{argmin}_{\theta} \sum_{\theta} q(\theta)(\log(q(\theta)) - \log(p_1(\theta))) \\ q(\theta) &\text{ s.t. } \sum_{\theta} q(\theta)(\log(q(\theta)) - \log(p_2(\theta))) < B \text{ and } \sum_{\theta} q(\theta) = 1. \end{aligned} \quad (4.11)$$

For solving the mentioned problem, one should find a  $q$  that satisfies the Karush-Kuhn-Tucker (KKT) conditions (Boyd & Vandenberghe, 2004) for

$$\mathcal{L} = \sum_{\theta} q(\theta)\log\left(\frac{q(\theta)}{p_1(\theta)}\right) + \lambda \sum_{\theta} q(\theta)\log\left(\frac{q(\theta)}{p_2(\theta)}\right) - \lambda B + \alpha - \alpha \sum_{\theta} q(\theta), \quad (4.12)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial q(\theta)} &= \log\left(\frac{q(\theta)}{p_1(\theta)}\right) + 1 + \lambda \log\left(\frac{q(\theta)}{p_2(\theta)}\right) + \lambda - \alpha \\ &= (1 + \lambda)\log(q(\theta)) - \log(p_1(\theta)) - \lambda \log(p_2(\theta)) + 1 + \lambda - \alpha, \end{aligned} \quad (4.13)$$

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

$$\log(q^*(\theta)) = (1 - \gamma)\log(p_1(\theta)) + \gamma\log(p_2(\theta)) + \operatorname{Const}(\alpha, \gamma), \quad (4.14)$$

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

$$\begin{aligned} \operatorname{Const}(\alpha, \gamma) &= -\log(Z(\gamma)) \\ \text{where } Z(\gamma) &= \sum_{\theta} p_1^{1-\gamma}(\theta)p_2^{\gamma}(\theta). \end{aligned} \quad (4.15)$$

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

4.2.2 *Proof of the Claim That B Is a Decreasing Function of Surprise.* According to the KKT conditions,

$$\lambda(\mathbf{D}_{KL}[q^*(\theta)||p_2(\theta)] - B) = 0. \quad (4.16)$$

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

$$\begin{aligned} B(\gamma) &= \mathbf{D}_{KL}[q^*(\theta)||p_2(\theta)] \\ &= (1 - \gamma)\mathbb{E}_{q^*} \left[ \log \left( \frac{p_1(\theta)}{p_2(\theta)} \right) \right] - \log(Z(\gamma)). \end{aligned} \quad (4.17)$$

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

$$\begin{aligned} \frac{\partial \log(Z(\gamma))}{\partial \gamma} &= \frac{1}{Z(\gamma)} \frac{\partial}{\partial \gamma} \sum_{\theta} p_1^{1-\gamma}(\theta) p_2^{\gamma}(\theta) \\ &= \frac{1}{Z(\gamma)} \sum_{\theta} p_1^{1-\gamma}(\theta) p_2^{\gamma}(\theta) \log \left( \frac{p_2(\theta)}{p_1(\theta)} \right) \\ &= \mathbb{E}_{q^*} \left[ \log \left( \frac{p_2(\theta)}{p_1(\theta)} \right) \right], \end{aligned} \quad (4.18)$$

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

$$\begin{aligned} \frac{\partial \mathbb{E}_{q^*}[h(\theta)]}{\partial \gamma} &= \frac{\partial}{\partial \gamma} \sum_{\theta} q^*(\theta) h(\theta) \\ &= \sum_{\theta} q^*(\theta) h(\theta) \frac{\partial}{\partial \gamma} \log(q^*(\theta)) \\ &= \sum_{\theta} q^*(\theta) h(\theta) \frac{\partial}{\partial \gamma} ((1 - \gamma)\log(p_1(\theta)) + \gamma\log(p_2(\theta)) - \log(Z(\gamma))) \\ &= \mathbb{E}_{q^*} \left[ h(\theta) \log \left( \frac{p_2(\theta)}{p_1(\theta)} \right) \right] - \mathbb{E}_{q^*}[h(\theta)] \mathbb{E}_{q^*} \left[ \log \left( \frac{p_2(\theta)}{p_1(\theta)} \right) \right]. \end{aligned} \quad (4.19)$$

Using the last previous equations, we have

$$\begin{aligned} \frac{\partial B(\gamma)}{\partial \gamma} &= -(1 - \gamma) \left( \mathbb{E}_{q^*} \left[ \left( \log \left( \frac{p_2(\theta)}{p_1(\theta)} \right) \right)^2 \right] - \mathbb{E}_{q^*} \left[ \log \left( \frac{p_2(\theta)}{p_1(\theta)} \right) \right]^2 \right) \\ &= -(1 - \gamma) \mathbf{Var}_{q^*} \left[ \log \left( \frac{p_2(\theta)}{p_1(\theta)} \right) \right] \leq 0, \end{aligned} \quad (4.20)$$

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

**4.3 Derivations of Message Passing  $N$ : Algorithm 2.** For the sake of clarity and coherence, we repeat here some steps performed in section 2.

Following the idea of Adams and MacKay (2007) we 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  $\pi^{(t)}(\theta)$  can be written as

$$\begin{aligned} \pi^{(t)}(\theta) &= \mathbf{P}(\Theta_{t+1} = \theta | y_{1:t}) \\ &= \sum_{r=1}^t \mathbf{P}(r_t | y_{1:t}) \mathbf{P}(\Theta_{t+1} = \theta | r_t, y_{1:t}). \end{aligned} \quad (4.21)$$

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

$$\pi^{(t)}(\theta) = \sum_{k=0}^{t-1} w_t^{(k)} \mathbf{P}(\Theta_{t+1} = \theta | R_t = t - k, y_{1:t}) = \sum_{k=0}^{t-1} w_t^{(k)} \pi_k^{(t)}(\theta), \quad (4.22)$$

where  $\pi_k^{(t)}(\theta) = \mathbf{P}(\Theta_t = \theta | R_t = t - k, y_{1:t})$  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 weight 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  $\pi_B^{(t+1)}(\theta)$  as

$$\begin{aligned} \pi_B^{(t+1)}(\theta) &= \frac{\pi^{(t)}(\theta) P_Y(y_{t+1} | \theta)}{P(y_{t+1}; \pi^{(t)})} \\ &= \frac{P_Y(y_{t+1} | \theta)}{P(y_{t+1}; \pi^{(t)})} \sum_{k=0}^{t-1} w_t^{(k)} \mathbf{P}(\Theta_{t+1} = \theta | R_t = t - k, y_{1:t}). \end{aligned} \quad (4.23)$$

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

$$\begin{aligned}\pi_B^{(t+1)}(\theta) &= \frac{P_Y(y_{t+1}|\theta)}{P(y_{t+1}; \pi^{(t)})} \sum_{k=0}^{t-1} w_t^{(k)} \frac{\mathbf{P}(y_{k+1:t}|\Theta_{t+1} = \theta, R_t = t - k)\pi^{(0)}(\theta)}{\mathbf{P}(y_{k+1:t}|R_t = t - k)} \\ &= \frac{1}{P(y_{t+1}; \pi^{(t)})} \sum_{k=0}^{t-1} w_t^{(k)} \frac{\prod_{i=k+1}^{t+1} P_Y(y_i|\theta)\pi^{(0)}(\theta)}{\mathbf{P}(y_{k+1:t}|R_t = t - k)},\end{aligned}\quad (4.24)$$

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

$$\begin{aligned}\pi_B^{(t+1)}(\theta) &= \frac{1}{P(y_{t+1}; \pi^{(t)})} \sum_{k=0}^{t-1} w_t^{(k)} \frac{\mathbf{P}(y_{k+1:t+1}|R_{t+1} = t - k + 1)}{\mathbf{P}(y_{k+1:t}|R_t = t - k)} \\ &\quad \times \mathbf{P}(\Theta_{t+1} = \theta|R_{t+1} = t - k + 1, y_{k+1:t+1}) \\ &= \frac{1}{P(y_{t+1}; \pi^{(t)})} \sum_{k=0}^{t-1} w_t^{(k)} \mathbf{P}(y_{t+1}|R_{t+1} = t - k + 1, y_{1:t}) \\ &\quad \times \mathbf{P}(\Theta_{t+1} = \theta|R_{t+1} = t - k + 1, y_{1:t+1}).\end{aligned}\quad (4.25)$$

This gives us

$$\begin{aligned}\pi_B^{(t+1)}(\theta) &= \sum_{k=0}^{t-1} w_t^{(k)} \frac{P(y_{t+1}; \pi_k^{(t)})}{P(y_{t+1}; \pi^{(t)})} \times \\ &\quad \times \mathbf{P}(\Theta_{t+1} = \theta|R_{t+1} = t - k + 1, y_{1:t+1}),\end{aligned}\quad (4.26)$$

and finally,

$$w_{B,t+1}^{(k)} = \frac{P(y_{t+1}; \pi_k^{(t)})}{P(y_{t+1}; \pi^{(t)})} w_t^{(k)}.\quad (4.27)$$

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

$$w_{t+1}^{(k)} = (1 - \gamma_{t+1})w_{B,t+1}^{(k)} = (1 - \gamma_{t+1}) \frac{P(y_{t+1}; \pi_k^{(t)})}{P(y_{t+1}; \pi^{(t)})} w_t^{(k)},\quad (4.28)$$

and for the newly added particle  $t$ ,

$$w_{t+1}^{(t)} = \gamma_{t+1},\quad (4.29)$$

where  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \pi^{(t)}), m = \frac{p_c}{1-p_c})$  of equation 2.9.

The MPN algorithm uses equations 4.22, 4.28, and 4.29 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 equations 4.28 and 4.29, keeps the greatest  $N$  weights, and sets the rest weights equal to 0. After normalizing the new weights, it uses equation 4.22 (but only over the particles with nonzero weights) to compute the belief  $\hat{\pi}^{(t)}$ . (For the particular case of exponential family, see algorithm 2 for the pseudocode.)

**4.4 Derivation of the Weight Update for Particle Filtering: Algorithm 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

$$P(c_{1:t+1}|y_{1:t+1}) \approx \sum_{i=1}^N w_{t+1}^{(i)} \delta(c_{1:t+1} - c_{1:t+1}^{(i)}). \quad (4.30)$$

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

$$\begin{aligned} w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(c_{1:t+1}^{(i)}|y_{1:t+1})}{Q(c_{1:t+1}^{(i)}|y_{1:t+1})} \propto \frac{\mathbf{P}(c_{1:t+1}^{(i)}, y_{t+1}|y_{1:t})}{Q(c_{1:t+1}^{(i)}|y_{1:t+1})}, \\ w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(y_{t+1}, c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t})\mathbf{P}(c_{1:t}^{(i)}|y_{1:t})}{Q(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1})Q(c_{1:t}^{(i)}|y_{1:t})}, \end{aligned} \quad (4.31)$$

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

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

$$w_{t+1}^{(i)} \propto \frac{\mathbf{P}(y_{t+1}, c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t})}{Q(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1})} w_t^{(i)}. \quad (4.32)$$

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

$$Q(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1}) = \mathbf{P}(c_{t+1}^{(i)}|c_{1:t}^{(i)}, y_{1:t+1}). \quad (4.33)$$

Using Bayes' rule and equations 4.32 and 4.33, after a few steps of algebra, we have

$$\begin{aligned} w_{t+1}^{(i)} &\propto \frac{\mathbf{P}(y_{t+1}, c_{t+1}^{(i)} | c_{1:t}^{(i)}, y_{1:t})}{\mathbf{P}(c_{t+1}^{(i)} | c_{1:t}^{(i)}, y_{1:t+1})} w_t^{(i)} = \mathbf{P}(y_{t+1} | c_{1:t}^{(i)}, y_{1:t}) w_t^{(i)} \\ &\propto \left( (1 - p_c) \mathbf{P}(y_{t+1} | c_{1:t}^{(i)}, y_{1:t}, c_{t+1}^{(i)} = 0) + p_c \mathbf{P}(y_{t+1} | c_{1:t}^{(i)}, y_{1:t}, c_{t+1}^{(i)} = 1) \right) w_t^{(i)}. \end{aligned} \quad (4.34)$$

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

$$w_{t+1}^{(i)} = \left[ (1 - p_c) P(y_{t+1}; \hat{\pi}_i^{(t)}) + p_c P(y_{t+1}; \pi^{(0)}) \right] w_t^{(i)} / Z, \quad (4.35)$$

where  $Z$  is the normalization factor,

$$Z = (1 - p_c) P(y_{t+1}; \hat{\pi}^{(t)}) + p_c P(y_{t+1}; \pi^{(0)}), \quad (4.36)$$

where we have

$$P(y_{t+1}; \hat{\pi}^{(t)}) = \sum_{i=1}^N w_i^{(i)} P(y_{t+1}; \hat{\pi}_i^{(t)}). \quad (4.37)$$

We now compute the weights corresponding to  $\pi_B^{(t+1)}$  as defined in equation 2.8

$$w_{B,t+1}^{(i)} = \frac{P(y_{t+1}; \hat{\pi}_i^{(t)})}{P(y_{t+1}; \hat{\pi}^{(t)})} w_i^{(i)}. \quad (4.38)$$

Combining equations 4.35, 4.36, and 4.38, we can then rewrite the weight update rule as

$$w_{t+1}^{(i)} = (1 - \gamma_{t+1}) w_{B,t+1}^{(i)} + \gamma_{t+1} w_t^{(i)}, \quad (4.39)$$

where  $\gamma_{t+1} = \gamma \left( \mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}^{(t)}), \frac{p_c}{1-p_c} \right)$  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 4.33, we have

$$\begin{aligned}
Q(c_{t+1}^{(i)} = 1 | c_{1:t}^{(i)}, y_{1:t+1}) &= \frac{p_c P(y_{t+1}; \pi^{(0)})}{(1 - p_c) P(y_{t+1}; \hat{\pi}_i^{(t)}) + p_c P(y_{t+1}; \pi^{(0)})} \\
&= \gamma \left( \mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\pi}_i^{(t)}), \frac{p_c}{1 - p_c} \right). \tag{4.40}
\end{aligned}$$

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)

$$N_{\text{eff}} \approx \frac{1}{\sum_{i=1}^N (w_t^{(i)})^2}. \tag{4.41}$$

When  $N_{\text{eff}}$  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  $N_{\text{eff}}$ , and following Doucet and Johansen (2009), we performed resampling when  $N_{\text{eff}} \leq N/2$ .

## 4.5 Surprise-Modulation as a Framework for Other Algorithms.

*4.5.1 SMiLe Rule.* The Confidence Corrected Surprise (Faraji et al., 2018) is

$$\mathbf{S}_{\text{CC}}(y_{t+1}; \hat{\pi}^{(t)}) = \mathbf{D}_{\text{KL}}[\hat{\pi}^{(t)}(\theta) || \tilde{P}(\theta | y_{t+1})], \tag{4.42}$$

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

$$\tilde{P}(\theta | y_{t+1}) = \frac{P_Y(y_{t+1} | \theta)}{\int P_Y(y_{t+1} | \theta') d\theta'}. \tag{4.43}$$

Note that this becomes equal to  $P(\theta | y_{t+1})$  if the prior belief  $\pi^{(0)}$  is a uniform distribution (see 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,

$$\begin{aligned}
\hat{\pi}^{(t+1)}(\theta) &= \arg \min_q \mathbf{D}_{\text{KL}}[q(\theta) || \tilde{P}(\theta | y_{t+1})] \\
\text{s.t. } \mathbf{D}_{\text{KL}}[q(\theta) || \hat{\pi}^{(t)}(\theta)] &\leq B_{t+1}, \tag{4.44}
\end{aligned}$$



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

$$\log(\hat{\pi}^{(t+1)}(\theta)) = (1 - \gamma_{t+1}) \log(\hat{\pi}^{(t)}(\theta)) + \gamma_{t+1} \log(\tilde{P}(\theta|y_{t+1})) + \text{Const.}, \quad (4.45)$$

where  $\gamma_{t+1} \in [0, 1]$  is specified so that it satisfies the constraint in equation 4.44.

Although equation 4.45 looks very similar to equation 2.13, it signifies a trade-off between the latest belief  $\hat{\pi}^{(t)}$  and the belief updated by only the most recent observation  $\tilde{P}(\theta|y_{t+1})$ , that is, a trade-off between adherence to the current belief and reset. While SMiLe adheres to the current belief  $\hat{\pi}^{(t)}$ , Variational SMiLe integrates the new observation with the current belief to get  $\hat{\pi}_B^{(t)}$ , which leads to a trade-off similar to the one of the exact Bayesian inference (see equations 2.12 and 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,

$$B_{t+1} = B_{\max} \gamma(\mathbf{S}_{CC}(y_{t+1}), m) \quad \text{where } B_{\max} = \mathbf{D}_{KL}[P(\theta|y_{t+1})||\hat{\pi}^{(t)}(\theta)], \quad (4.46)$$

where  $m$  is a free parameter. Then,  $\gamma_{t+1}$  is found by satisfying the constraint of the optimization problem in equation 4.44 using equations 4.45 and 4.46.

*4.5.2 Nassar's Algorithm.* For the particular case that observations are drawn from a gaussian distribution with known variance and unknown mean,  $y_{t+1}|\mu_{t+1} \sim \mathcal{N}(\mu_{t+1}, \sigma^2)$  and  $\theta_t = \mu_t$ , Nassar et al. (2012, 2010) considered the problem of estimating the expected  $\mu_t$  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. (2012, 2010) were developed for the case that whenever the environment changes, the mean  $\mu_{t+1}$  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  $\mu_{t+1}$  (i.e.,  $\hat{\mu}_{t+1}$ ) estimated by the agent upon observing a new sample  $y_{t+1}$  is

$$\hat{\mu}_{t+1} = \hat{\mu}_t + \alpha_{t+1}(y_{t+1} - \hat{\mu}_t), \quad (4.47)$$

with  $\alpha_{t+1}$  the adaptive learning rate given by

$$\alpha_{t+1} = \frac{1 + \Omega_{t+1} \hat{r}_t}{1 + \hat{r}_t}, \quad (4.48)$$

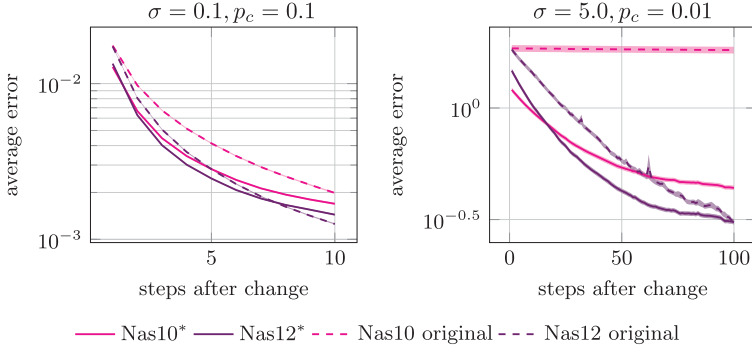


Figure 12: Gaussian estimation task: Transient performance after changes for original algorithms of Nassar et al. (2010) and Nassar et al. (2012). Mean squared error for the estimation of  $\mu_t$  at each time step  $n$  after an environmental change, that is, the average of  $\mathbf{MSE}[\hat{\theta}_t | R_t = n]$  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. 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.

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  $\gamma_{t+1}$  of equation 2.9.

We next extend their approach to a more general case where the prior is a gaussian distribution with arbitrary variance,  $\mu_{t+1} \sim \mathcal{N}(\mu_0, \sigma_0^2)$ . 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 Figures 12 and 13.

**4.5.3 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  $\mu_t$  is  $\hat{\mu}_t$ . It can be shown (see the appendix for the derivation) that the expected  $\mu_{t+1}$  (i.e.,  $\hat{\mu}_{t+1}$ ) upon observing the new sample  $y_{t+1}$  is

$$\begin{aligned} \hat{\mu}_{t+1} = & (1 - \gamma_{t+1}) \left( \hat{\mu}_t + \frac{1}{\rho + r_t + 1} (y_{t+1} - \hat{\mu}_t) \right) \\ & + \gamma_{t+1} \left( \mu_0 + \frac{1}{\rho + 1} (y_{t+1} - \mu_0) \right), \end{aligned} \quad (4.49)$$

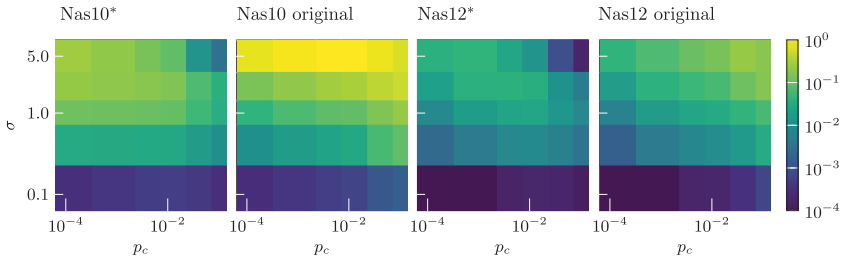


Figure 13: Gaussian estimation task: Steady-state performance for original algorithms of Nassar et al. (2010) and Nassar et al. (2012). Difference between the mean squared error of each algorithm and the optimal solution (Exact Bayes), that is, the average  $\Delta\text{MSE}[\hat{\phi}_t]$  over time for each combination of environmental parameters  $\sigma$  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.

where  $\rho = \frac{\sigma^2}{\sigma_0^2}$ ,  $\mu_0$  is the mean of the prior distribution, and  $\gamma_{t+1}$  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  $\mu_0$ , in the same spirit as the other algorithms we considered here. Equation 4.49 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. (2012, 2010), we can rewrite the above formula as

$$\hat{\mu}_{t+1} = \frac{\rho}{\rho + 1} \left( \hat{\mu}_t + \gamma_{t+1} (\mu_0 - \hat{\mu}_t) \right) + \frac{1}{\rho + 1} \left( \hat{\mu}_t + \alpha_{t+1} (y_{t+1} - \hat{\mu}_t) \right), \quad (4.50)$$

where we have defined  $\alpha_{t+1} = \frac{\rho + \gamma_{t+1} \rho + 1}{\rho + \gamma_{t+1} \rho + 1}$ . 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. (2012, 2010), 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  $\rho \rightarrow 0$ , so that the first term of equation 4.50 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 equations 4.47 and 4.48, 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 nonstationary regime with a history of change points, the time interval  $r_t$  is not known. Nassar et al. (2012, 2010) 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:  $\hat{r}_{t+1} = (1 - \gamma_{t+1})(\hat{r}_t + 1) + \gamma_{t+1}$ , that is, at each time step, there is a probability  $(1 - \gamma_{t+1})$  that  $\hat{r}_t$  increments by 1 and a probability  $\gamma_{t+1}$  that it is reset to 1. So  $\hat{r}_{t+1}$  is the weighted sum of these two outcomes. Hence, equation 4.50 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 the appendix for pseudocode).

In Nassar et al. (2012), the variance  $\hat{\sigma}_{t+1}^2 = \text{Var}[\mu_{t+1}|y_{1:t+1}]$  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 appendix. We call the combination of equation 4.50 with this way of computing the expected time interval  $\hat{r}_t$  Nas12\* (see the appendix for 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 section 2.1, 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.

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

$$\begin{aligned} Q\left(\hat{\pi}^{(t+1)}(\theta) = \hat{\pi}_B^{(t+1)}(\theta) | y_{t+1}\right) &= 1 - \gamma_{t+1}, \\ Q\left(\hat{\pi}^{(t+1)}(\theta) = P(\theta|y_{t+1}) | y_{t+1}\right) &= \gamma_{t+1}. \end{aligned} \quad (4.51)$$

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

$$\mathbb{E}_Q[\hat{\pi}^{(t+1)}(\theta)] = (1 - \gamma_{t+1})\hat{\pi}_B^{(t+1)}(\theta) + \gamma_{t+1}P(\theta|y_{t+1}). \quad (4.52)$$

If we apply equation 4.52 to  $\hat{\mu}_{t+1}$ , we find that  $\mathbb{E}_Q[\hat{\mu}^{(t+1)}]$ , is identical to the generalization of Nassar et al. (2010) (see equation 4.49).

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  $\gamma_{t+1}$ , the *expected value* of  $\hat{R}_{t+1}$  is

$$\mathbb{E}_Q[\hat{R}_{t+1}] = (1 - \gamma_{t+1})(\hat{r}_t + 1) + \gamma_{t+1}. \quad (4.53)$$

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 is in the appendix.

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

$$P_Y(y|\theta) = h(y)\exp(\theta^T \phi(y) - A(\theta)), \quad (4.54)$$

where  $\theta$  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  $\pi^{(0)}$  has the form

$$\begin{aligned} \pi^{(0)}(\theta) &= \mathbf{P}_\pi(\Theta = \theta; \chi^{(0)}, \nu^{(0)}) \\ &= \tilde{h}(\theta)f(\chi^{(0)}, \nu^{(0)})\exp(\theta^T \chi^{(0)} - \nu^{(0)}A(\theta)), \end{aligned} \quad (4.55)$$

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

$$\mathbf{S}_{\text{BF}}\left(y_{t+1}; \mathbf{P}_\pi(\Theta = \theta; \chi^{(t)}, \nu^{(t)})\right) = \frac{f(\chi^{(t)} + \phi(y_{t+1}), \nu^{(t)} + 1)}{f(\chi^{(0)} + \phi(y_{t+1}), \nu^{(0)} + 1)} \frac{f(\chi^{(0)}, \nu^{(0)})}{f(\chi^{(t)}, \nu^{(t)})}. \quad (4.56)$$

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

**4.7 Simulation Task.** In this section, 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 that we used for simulations.

*4.7.1 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  $\Theta_t$  as a function of the observations  $Y_{1:t}$ , that is,  $\hat{\Theta}_t = f(Y_{1:t})$ . The estimator that minimizes the mean squared error  $\text{MSE}[\hat{\Theta}_t] = \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t)}[(\hat{\Theta}_t - \Theta_t)^2]$  is

$$\hat{\Theta}_t^{\text{Opt}} = \mathbb{E}_{\mathbf{P}(\Theta_t|Y_{1:t})}[\Theta_t] = \mathbb{E}_{\pi^{(t)}}[\Theta_t], \quad (4.57)$$

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

$$\begin{aligned} \text{MSE}[\hat{\Theta}_t] &= \text{MSE}[\hat{\Theta}_t^{\text{Opt}}] + \Delta\text{MSE}[\hat{\Theta}_t], \\ \text{where } \Delta\text{MSE}[\hat{\Theta}_t] &= \mathbb{E}_{\mathbf{P}(Y_{1:t})}[(\hat{\Theta}_t - \hat{\Theta}_t^{\text{Opt}})^2] \geq 0. \end{aligned} \quad (4.58)$$

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  $\hat{\Theta}_t^{\text{Opt}}$ . As a result, if the estimator we are interested in is the expected value of  $\Theta_t$  under the approximate belief  $\hat{\pi}^{(t)}$  computed by each of our algorithms (i.e.,  $\hat{\Theta}_t = \mathbb{E}_{\hat{\pi}^{(t)}}[\Theta_t]$ ), the second term in equation 4.58, that is, the deviation from optimality, is a measure of how good the approximation is.

**Proof for the Algorithms without Sampling.** Consider  $\hat{\Theta}_t^{\text{Opt}} = f_{\text{Opt}}(Y_{1:t})$ . Then, for any other arbitrary estimator  $\hat{\Theta}_t = f(Y_{1:t})$  (except for the ones with sampling), we have

$$\begin{aligned} \text{MSE}[\hat{\Theta}_t] &= \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t)}[(\hat{\Theta}_t - \Theta_t)^2] \\ &= \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t)}[(f(Y_{1:t}) - \Theta_t)^2] \\ &= \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t)}\left[\left((f(Y_{1:t}) - f_{\text{Opt}}(Y_{1:t})) + (f_{\text{Opt}}(Y_{1:t}) - \Theta_t)\right)^2\right]. \end{aligned} \quad (4.59)$$

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

$$\begin{aligned} \mathbf{MSE}[\hat{\Theta}_t] &= \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t)} \left[ (f(Y_{1:t}) - f_{\text{Opt}}(Y_{1:t}))^2 \right] \\ &\quad + \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t)} \left[ (f_{\text{Opt}}(Y_{1:t}) - \Theta_t)^2 \right] \\ &\quad + 2\mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t)} \left[ (f(Y_{1:t}) - f_{\text{Opt}}(Y_{1:t})) (f_{\text{Opt}}(Y_{1:t}) - \Theta_t) \right]. \end{aligned} \quad (4.60)$$

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

$$\begin{aligned} \text{3rd line} &= 2\mathbb{E}_{\mathbf{P}(Y_{1:t})} \left[ \mathbb{E}_{\mathbf{P}(\Theta_t | Y_{1:t})} \left[ (f(Y_{1:t}) - f_{\text{Opt}}(Y_{1:t})) (f_{\text{Opt}}(Y_{1:t}) - \Theta_t) \right] \right] \\ &= 2\mathbb{E}_{\mathbf{P}(Y_{1:t})} \left[ (f(Y_{1:t}) - f_{\text{Opt}}(Y_{1:t})) (f_{\text{Opt}}(Y_{1:t}) - \mathbb{E}_{\mathbf{P}(\Theta_t | Y_{1:t})}[\Theta_t]) \right] = 0, \end{aligned} \quad (4.61)$$

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

$$\mathbf{MSE}[\hat{\Theta}_t] = \mathbf{MSE}[\hat{\Theta}_t^{\text{Opt}}] + \mathbb{E}_{\mathbf{P}(Y_{1:t})} \left[ (\hat{\Theta}_t - \hat{\Theta}_t^{\text{Opt}})^2 \right]. \quad (4.62)$$

□

**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) drawn from a distribution that 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

$$\hat{\Theta}_t^{\text{PF}} = f(Y_{1:t}, C_{1:t}^{(1:N)}), \quad (4.63)$$

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

$$\begin{aligned} \mathbf{MSE}[\hat{\Theta}_t^{\text{PF}}] &= \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t) Q(C_{1:t}^{(1:N)} | Y_{1:t})} \left[ (\hat{\Theta}_t^{\text{PF}} - \Theta_t)^2 \right] \\ &= \mathbb{E}_{\mathbf{P}(Y_{1:t}, \Theta_t) Q(C_{1:t}^{(1:N)} | Y_{1:t})} \left[ (f(Y_{1:t}, C_{1:t}^{(1:N)}) - \Theta_t)^2 \right]. \end{aligned} \quad (4.64)$$

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

$$\begin{aligned} \mathbf{MSE}[\hat{\Theta}_t^{\text{PF}}] &= \mathbf{MSE}[\hat{\Theta}_t^{\text{Opt}}] + \mathbb{E}_{\mathbf{P}(Y_{1:t})Q(C_{1:t}^{(1:N)}|Y_{1:t})} [(\hat{\Theta}_t^{\text{PF}} - \hat{\Theta}_t^{\text{Opt}})^2] \\ &= \mathbf{MSE}[\hat{\Theta}_t^{\text{Opt}}] + \mathbb{E}_{\mathbf{P}(Y_{1:t})} \left[ \mathbb{E}_{Q(C_{1:t}^{(1:N)}|Y_{1:t})} [(\hat{\Theta}_t^{\text{PF}} - \hat{\Theta}_t^{\text{Opt}})^2] \right], \end{aligned} \quad (4.65)$$

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

$$\begin{aligned} \mathbf{MSE}[\hat{\Theta}_t^{\text{PF}}] &= \mathbf{MSE}[\hat{\Theta}_t^{\text{Opt}}] + \mathbb{E}_{\mathbf{P}(Y_{1:t})} \left[ \text{Var}_{Q(C_{1:t}^{(1:N)}|Y_{1:t})} (\hat{\Theta}_t^{\text{PF}}) \right. \\ &\quad \left. + \text{Bias}_{Q(C_{1:t}^{(1:N)}|Y_{1:t})} (\hat{\Theta}_t^{\text{PF}}, \hat{\Theta}_t^{\text{Opt}})^2 \right]. \end{aligned} \quad (4.66)$$

□

**4.7.2 Leaky Integration.** For the 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

$$\hat{\theta}_t = \frac{\sum_{k=1}^t \omega^{t-k} y_k}{\sum_{k=1}^t \omega^{t-k}}. \quad (4.67)$$

For the 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

$$\hat{\theta}_{i,t} = \frac{\sum_{k=1}^t \omega^{t-k} \delta(y_k - i)}{\sum_{k=1}^t \omega^{t-k}}, \quad (4.68)$$

where  $\delta$  is the Kronecker delta function.

**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

$$\begin{aligned} \mathbf{S}_{\text{Sh}}(y_{t+1}; \pi^{(t)}) &= \log \left( \frac{1}{\mathbf{P}(y_{t+1}|y_{1:t})} \right) \\ &= \log \left( \frac{1}{(1 - p_c)P(y_{t+1}; \pi^{(t)}) + p_c P(y_{t+1}; \pi^{(0)})} \right) \end{aligned}$$



$$\begin{aligned}
&= \log\left(\frac{1}{P(y_{t+1}; \pi^{(0)})}\right) + \log\left(\frac{1}{p_c} \frac{1}{1 + \frac{1}{m} \frac{1}{S_{\text{BF}}(y_{t+1}; \pi^{(t)})}}\right) \\
&= S_{\text{Sh}}(y_{t+1}; \pi^{(0)}) + \log\left(\frac{\gamma_{t+1}}{p_c}\right), \tag{4.69}
\end{aligned}$$

where  $\gamma_{t+1} = \gamma\left(S_{\text{BF}}(y_{t+1}; \pi^{(t)}), m = \frac{p_c}{1-p_c}\right)$  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 3.2.

#### 4.9 Experimental Predictions.

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

$$P_Y(y|\theta) = \mathcal{N}(y; \theta, \sigma^2), \tag{4.70}$$

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

*4.9.2 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  $\pi^{(t)}$  at each time is a gaussian distribution,

$$\pi^{(t)}(\theta) = \mathcal{N}(\theta; \hat{\theta}_t, \hat{\sigma}_t^2), \tag{4.71}$$

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  $P(y; \pi^{(t)})$  can be written as

$$P(y; \pi^{(t)}) = \mathcal{N}(y; \hat{\theta}_t, \sigma^2 + \hat{\sigma}_t^2). \tag{4.72}$$

As mentioned in section 2, 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  $\sigma_C$ , and a sign bias  $s \in \{-1, 1\}$ , the average Bayes Factor Surprise is computed as

$$\bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s, \sigma_C) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbf{S}_{\text{BF}}(y_t; \hat{\pi}^{(t-1)}),$$

where  $\mathcal{T} = \{t : |\hat{\theta}_{t-1}| = \hat{\theta}, |\delta_t| = \delta, \hat{\sigma}_t = \sigma_C, s_t = s\}$ . (4.73)

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

$$\bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C) = \frac{\mathcal{N}(\hat{\theta} + \delta; 0, \sigma^2 + 1)}{\mathcal{N}(\delta; 0, \sigma^2 + \sigma_C^2)}. \quad (4.74)$$

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

$$\Delta \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, \sigma_C) = \frac{\mathcal{N}(\hat{\theta} + \delta; 0, \sigma^2 + 1) - \mathcal{N}(\hat{\theta} - \delta; 0, \sigma^2 + 1)}{\mathcal{N}(\delta; 0, \sigma^2 + \sigma_C^2)}. \quad (4.75)$$

It can be shown that

$$\Delta \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, \sigma_C) < 0 \text{ and } \frac{\partial}{\partial \delta} \Delta \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, \sigma_C) < 0 \quad (4.76)$$

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

$$\begin{aligned} & \bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, s = +1, \sigma_C) \\ &= -\log \left( p_c \mathcal{N}(\hat{\theta} + \delta; 0, \sigma^2 + 1) + (1 - p_c) \mathcal{N}(\delta; 0, \sigma^2 + \sigma_C^2) \right), \end{aligned} \quad (4.77)$$

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

$$\Delta \bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, \sigma_C) = \log \left( \frac{1 + m \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)}{1 + m \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C)} \right), \quad (4.78)$$

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

$$\Delta \bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, \sigma_C) > 0 \text{ and } \frac{\partial}{\partial \delta} \Delta \bar{\mathbf{S}}_{Sh}(\hat{\theta}, \delta, \sigma_C) > 0, \quad (4.79)$$

for all  $\hat{\theta} > 0$ ,  $\delta > 0$ , and  $\sigma_C > 0$ .

**Proof of the Second Inequality for  $\mathbf{S}_{BF}$ .** Let us define the variables

$$\begin{aligned} \sigma_d^2 &= \sigma^2 + \sigma_C^2, \\ \sigma_n^2 &= \sigma^2 + 1, \end{aligned} \quad (4.80)$$

as well as the functions

$$\begin{aligned} f_1(\delta) &= \bar{\mathbf{S}}_{BF}(\hat{\theta}, \delta, s = +1, \sigma_C) = \frac{\sigma_d}{\sigma_n} \exp\left(\frac{\delta^2}{2\sigma_d^2} - \frac{(\delta + \hat{\theta})^2}{2\sigma_n^2}\right), \\ f_2(\delta) &= \bar{\mathbf{S}}_{BF}(\hat{\theta}, \delta, s = -1, \sigma_C) = \frac{\sigma_d}{\sigma_n} \exp\left(\frac{\delta^2}{2\sigma_d^2} - \frac{(\delta - \hat{\theta})^2}{2\sigma_n^2}\right), \\ f(\delta) &= \Delta \bar{\mathbf{S}}_{BF}(\hat{\theta}, \delta, \sigma_C) = f_1(\delta) - f_2(\delta). \end{aligned} \quad (4.81)$$

The following inequalities hold true:

$$\begin{aligned} f(\delta) < 0 &\Rightarrow f_1(\delta) < f_2(\delta), \\ \sigma_C^2 < \sigma_0^2 = 1 &\Rightarrow \sigma_d^2 < \sigma_n^2. \end{aligned} \quad (4.82)$$

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

$$\begin{aligned} \frac{d}{d\delta} f(\delta) &= f_1(\delta) \left( \frac{\delta}{\sigma_d^2} - \frac{\delta + \hat{\theta}}{\sigma_n^2} \right) - f_2(\delta) \left( \frac{\delta}{\sigma_d^2} - \frac{\delta - \hat{\theta}}{\sigma_n^2} \right) \\ &= -\frac{\hat{\theta}}{\sigma_n^2} (f_1(\delta) + f_2(\delta)) \left( 1 - \frac{f_1(\delta) - f_2(\delta)}{f_1(\delta) + f_2(\delta)} \frac{\delta}{\hat{\theta}} \left( \frac{\sigma_n^2}{\sigma_d^2} - 1 \right) \right) \\ &< 0. \end{aligned} \quad (4.83)$$

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

**Proof of the Second Inequality for  $\mathbf{S}_{Sh}$ .** Using the functions we defined for the previous proof and after computing the partial derivative of equation 4.78, we have

$$\begin{aligned}
\frac{\partial}{\partial \delta} \Delta \bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, \sigma_C) &= \frac{\partial}{\partial \delta} \log \left( \frac{1 + m \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = -1, \sigma_C)}{1 + m \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s = +1, \sigma_C)} \right) \\
&= \frac{d}{d\delta} \log \left( \frac{1 + m f_2(\delta)}{1 + m f_1(\delta)} \right). \tag{4.84}
\end{aligned}$$

The derivative of the last term can be written in terms of the derivatives of  $f_1$  and  $f_2$ , indicated by  $f'_1$  and  $f'_2$ , respectively,

$$\begin{aligned}
\frac{\partial}{\partial \delta} \Delta \bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, \sigma_C) &= \frac{m f'_2(\delta)}{1 + m f_2(\delta)} - \frac{m f'_1(\delta)}{1 + m f_1(\delta)} \\
&= \frac{-m f'(\delta)}{(1 + m f_1(\delta))(1 + m f_2(\delta))} + \frac{m^2 (f_1 f'_2 - f'_1 f_2)(\delta)}{(1 + m f_1(\delta))(1 + m f_2(\delta))}. \tag{4.85}
\end{aligned}$$

The first term is always positive based on the proof for  $\mathbf{S}_{\text{BF}}$ . The second term is also always positive because

$$\begin{aligned}
(f_1 f'_2 - f'_1 f_2)(\delta) &= f_1(\delta) f_2(\delta) \left( \left( \frac{\delta}{\sigma_d^2} - \frac{\delta - \hat{\theta}}{\sigma_n^2} \right) - \left( \frac{\delta}{\sigma_d^2} - \frac{\delta + \hat{\theta}}{\sigma_n^2} \right) \right) \\
&= f_1(\delta) f_2(\delta) \frac{2\hat{\theta}}{\sigma_n^2} > 0. \tag{4.86}
\end{aligned}$$

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

**4.9.3 Simulation Procedure for Prediction 1.** In order to relax the main assumption of our theoretical proofs (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 hyperparameters  $\sigma^2$  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 section 2).
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  $y_t, \hat{\theta}_t, \hat{\sigma}_t, \delta_t, s_t, \mathbf{S}_{\text{Sh}}(y_t; \hat{\pi}^{(t-1)})$ , and  $\mathbf{S}_{\text{BF}}(y_t; \hat{\pi}^{(t-1)})$ .

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,

$$\mathcal{T} = \{1 < t \leq T : |\hat{\theta}_{t-1}| - \hat{\theta} < \Delta\theta, |\delta_t| - \delta < \Delta\delta, |\hat{\sigma}_t - \sigma_C| < \Delta\sigma_C, s_t = s\}, \quad (4.87)$$

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

$$\begin{aligned} \bar{\mathbf{S}}_{\text{BF}}(\hat{\theta}, \delta, s, \sigma_C) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbf{S}_{\text{BF}}(y_t; \hat{\pi}^{(t-1)}), \\ \bar{\mathbf{S}}_{\text{Sh}}(\hat{\theta}, \delta, s, \sigma_C) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbf{S}_{\text{Sh}}(y_t; \hat{\pi}^{(t-1)}). \end{aligned} \quad (4.88)$$

We repeated this procedure for  $N$  different simulated subjects (with different random seeds). The average of  $\bar{\mathbf{S}}_{\text{BF}}$  and  $\bar{\mathbf{S}}_{\text{Sh}}$  over  $N = 20$  subjects, for two learning algorithms (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 Figure 10B. The results are the same as what our theoretical analysis predicted.

*4.9.4 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  $P(y_{t+1}; \hat{\pi}^{(t)}) \approx p$  instead of  $P(y_{t+1}; \hat{\pi}^{(t)}) = p$ ), 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  $P(y_{t+1}; \hat{\pi}^{(t)})$ ,  $P(y_{t+1}; \hat{\pi}^{(0)})$ ,  $\mathbf{S}_{\text{Sh}}(y_t; \hat{\pi}^{(t-1)})$ , and  $\mathbf{S}_{\text{BF}}(y_t; \hat{\pi}^{(t-1)})$ . Then, for a given a probability value  $p > 0$ , we defined the set of time points,

$$\mathcal{T} = \{0 \leq t \leq T : |P(y_{t+1}; \hat{\pi}^{(t)}) - p| < \Delta p, |P(y_{t+1}; \hat{\pi}^{(0)}) - p| < \Delta p\}, \quad (4.89)$$

where  $|P(y_{t+1}; \hat{\pi}^{(t)}) - p| < \Delta p$  and  $|P(y_{t+1}; \hat{\pi}^{(0)}) - p| < \Delta p$  are equivalent to  $P(y_{t+1}; \hat{\pi}^{(t)}) \approx p$  and  $P(y_{t+1}; \hat{\pi}^{(0)}) \approx p$ , respectively.  $\Delta p$  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  $\bar{\mathbf{S}}_{\text{BF}}(p)$  and  $\bar{\mathbf{S}}_{\text{Sh}}(p)$  over  $\mathcal{T}$  for each value of  $p$ . We repeated this procedure for  $N$  different simulated subjects (with different random seeds). The average of  $\bar{\mathbf{S}}_{\text{BF}}$  and  $\bar{\mathbf{S}}_{\text{Sh}}$  over  $N = 20$  subjects, for two learning

algorithms (Nas12\* and pf20), and for  $T = 500$  and  $\Delta p = 0.0125$  is shown in Figure 11B.

## Appendix

---

### A.1 Modified Algorithm of Nassar et al. (2012, 2010): Adaptation for Gaussian Prior.

*A.1.1 Recursive Update of the Estimated Mean for Gaussian Prior.* Let us first consider the case of a stationary regime (no change points) where observed samples are drawn from a gaussian distribution with known variance,  $y_{t+1}|\theta \sim \mathcal{N}(\theta, \sigma^2)$ , and the parameter  $\theta$  is also drawn from a gaussian distribution  $\theta \sim \mathcal{N}(\mu_0, \sigma_0^2)$ . After having observed samples  $y_1, \dots, y_{t+1}$ , it can be shown that, using Bayes' rule, the posterior distribution  $P(\theta|y_{1:t+1}) = \pi_B^{(t+1)}(\theta)$  is

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

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

In a nonstationary regime where, after having observed  $y_1, \dots, y_t$  from the same hidden state, there is the possibility for a change point upon observing  $y_{t+1}$ , the posterior distribution is

$$P(\theta|y_{1:t+1}) = (1 - \gamma_{t+1})P(\theta|y_{1:t+1}, c_{t+1} = 0) + \gamma_{t+1}P(\theta|y_{t+1}, c_{t+1} = 1). \quad (\text{A.2})$$

To facilitate notation, we denote  $c_{t+1} = 0$  as ‘‘stay’’ and  $c_{t+1} = 1$  as ‘‘change’’ so that

$$P(\theta|y_{1:t+1}) = (1 - \gamma_{t+1})P(\theta|y_{1:t+1}, \text{stay}) + \gamma_{t+1}P(\theta|y_{t+1}, \text{change}). \quad (\text{A.3})$$

Note that the above is equivalent to the Bayesian recursive formula, equation 2.10 of the main text, where  $\gamma_{t+1}$  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,  $P(c_{t+1} = 1|y_{1:t+1})$ . In Nassar et al. (2010) this quantity is denoted as  $\Omega_{t+1}$ . Taking equation A.1 into account, we have

$$\begin{aligned} \mathbb{E}(\theta|y_{1:t+1}, \text{stay}) &= \mu_{B,t+1} = \frac{1}{\frac{1}{\sigma_0^2} + \frac{t+1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r}^{t+1} y_i}{\sigma^2} \right), \\ \mathbb{E}(\theta|y_{1:t+1}, \text{change}) &= \frac{1}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2} \right), \end{aligned} \quad (\text{A.4})$$

where  $r_t$  is the time interval of observations coming from the same hidden state, calculated at time  $t$ . Taking the expectation of equation A.3 the estimated mean upon observing the new sample  $y_{t+1}$  is

$$\hat{\mu}_{t+1} = (1 - \gamma) \frac{1}{\frac{1}{\sigma_0^2} + \frac{r_t+1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^{t+1} y_i}{\sigma^2} \right) + \gamma \frac{1}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2} \right), \quad (\text{A.5})$$

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

$$\begin{aligned} \hat{\mu}_{t+1} &= (1 - \gamma) \frac{1}{\frac{1}{\sigma_0^2} + \frac{r_t+1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^t y_i}{\sigma^2} + \frac{y_{t+1}}{\sigma^2} \right) \\ &\quad + \gamma \frac{1}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2} \right). \end{aligned} \quad (\text{A.6})$$

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

$$\begin{aligned} \hat{\mu}_{t+1} &= (1 - \gamma) \hat{\mu}_t + \gamma \mu_0 + (1 - \gamma) \frac{1}{\frac{\sigma^2}{\sigma_0^2} + r_t + 1} (y_{t+1} - \hat{\mu}_t) \\ &\quad + \gamma \frac{1}{\frac{\sigma^2}{\sigma_0^2} + 1} (y_{t+1} - \mu_0). \end{aligned} \quad (\text{A.7})$$

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

$$\begin{aligned} \hat{\mu}_{t+1} &= (1 - \gamma) \hat{\mu}_t + \gamma \mu_0 + (1 - \gamma) \frac{1}{\rho + r_t + 1} (y_{t+1} - \hat{\mu}_t) \\ &\quad + \gamma \frac{1}{\rho + 1} (y_{t+1} - \mu_0). \end{aligned} \quad (\text{A.8})$$

A rearrangement of the terms and inclusion of the dependency of  $\gamma$  on time yields

$$\begin{aligned} \hat{\mu}_{t+1} &= (1 - \gamma_{t+1}) \left( \hat{\mu}_t + \frac{1}{\rho + r_t + 1} (y_{t+1} - \hat{\mu}_t) \right) \\ &\quad + \gamma_{t+1} \left( \mu_0 + \frac{1}{\rho + 1} (y_{t+1} - \mu_0) \right). \end{aligned} \quad (\text{A.9})$$

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

$$\begin{aligned}\hat{\mu}_{t+1} &= (1 - \gamma)\hat{\mu}_t - (1 - \gamma)\frac{1}{\rho + r_t + 1}\hat{\mu}_t \\ &+ \gamma\mu_0 - \gamma\frac{1}{\rho + 1}\mu_0 \\ &+ (1 - \gamma)\frac{1}{\rho + r_t + 1}y_{t+1} + \gamma\frac{1}{\rho + 1}y_{t+1}.\end{aligned}\quad (\text{A.10})$$

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

$$\begin{aligned}\hat{\mu}_{t+1} &= (1 - \gamma)\hat{\mu}_t - (1 - \gamma)\frac{1}{\rho + 1}\hat{\mu}_t + (1 - \gamma)\frac{r_t}{(\rho + 1)(\rho + r_t + 1)}\hat{\mu}_t \\ &+ \gamma\mu_0 - \gamma\frac{1}{\rho + 1}\mu_0 \\ &+ (1 - \gamma)\frac{1}{\rho + 1}y_{t+1} - (1 - \gamma)\frac{r_t}{(\rho + 1)(\rho + r_t + 1)}y_{t+1} \\ &+ \gamma\frac{1}{\rho + 1}y_{t+1}.\end{aligned}\quad (\text{A.11})$$

After a further step of algebra, we arrive at

$$\begin{aligned}\hat{\mu}_{t+1} &= \frac{\rho}{\rho + 1}\left((1 - \gamma)\hat{\mu}_t + \gamma\mu_0\right) \\ &+ \frac{1}{\rho + 1}\left((1 - \gamma)\frac{r_t}{\rho + r_t + 1}(\hat{\mu}_t - y_{t+1}) + y_{t+1}\right).\end{aligned}\quad (\text{A.12})$$

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

$$\begin{aligned}\hat{\mu}_{t+1} &= \frac{\rho}{\rho + 1}\left((1 - \gamma)\hat{\mu}_t + \gamma\mu_0\right) + \frac{1}{\rho + 1}\left((1 - \alpha)\hat{\mu}_t + \alpha y_{t+1}\right) \\ \hat{\mu}_{t+1} &= \frac{\rho}{\rho + 1}\left(\hat{\mu}_t + \gamma(\mu_0 - \hat{\mu}_t)\right) + \frac{1}{\rho + 1}\left(\hat{\mu}_t + \alpha(y_{t+1} - \hat{\mu}_t)\right).\end{aligned}\quad (\text{A.13})$$



Adding back the dependency of  $\gamma$  and  $\alpha$  on time, we finally have

$$\hat{\mu}_{t+1} = \frac{\rho}{\rho+1} \left( \hat{\mu}_t + \gamma_{t+1} (\mu_0 - \hat{\mu}_t) \right) + \frac{1}{\rho+1} \left( \hat{\mu}_t + \alpha_{t+1} (y_{t+1} - \hat{\mu}_t) \right). \quad (\text{A.14})$$

*A.1.2 Recursive Update of the Estimated Variance for Gaussian Prior.* Nassar et al. (2012) first calculate the variance  $\hat{\sigma}_{t+1}^2 = \text{Var}(\theta|y_{1:t+1})$  and then, based on this, compute  $\hat{r}_{t+1}$ . We derive here these calculations for the case of gaussian prior. We note again that

$$P(\theta|y_{1:t+1}) = (1 - \gamma_{t+1})P(\theta|y_{1:t+1}, \text{stay}) + \gamma_{t+1}P(\theta|y_{t+1}, \text{change}). \quad (\text{A.15})$$

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

$$\begin{aligned} \hat{\sigma}_{t+1}^2 &= (1 - \gamma)\sigma_{\text{stay}}^2 + \gamma\sigma_{\text{change}}^2 + (1 - \gamma)\gamma(\mu_{\text{stay}} - \mu_{\text{change}})^2 \\ &= (1 - \gamma)\sigma_{B,t+1}^2 + \gamma\sigma_{\text{change}}^2 + (1 - \gamma)\gamma(\mu_{B,t+1} - \mu_{\text{change}})^2, \end{aligned} \quad (\text{A.16})$$

where  $\sigma_{B,t+1}^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{r_{t+1}}{\sigma^2}}$  and  $\sigma_{\text{change}}^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}}$ .

We have already defined  $\rho = \frac{\sigma^2}{\sigma_0^2}$  so that

$$A = (1 - \gamma)\sigma_{B,t+1}^2 + \gamma\sigma_{\text{change}}^2 = (1 - \gamma)\frac{\sigma^2}{\rho + r_t + 1} + \gamma\frac{\sigma^2}{\rho + 1}, \quad (\text{A.17})$$

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

$$A = \frac{\sigma^2}{\rho+1} \left( 1 - (1 - \gamma)\frac{1}{\rho + r_t + 1} \right). \quad (\text{A.18})$$

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

$$A = \frac{\sigma^2}{\rho+1} \alpha. \quad (\text{A.19})$$

Note that  $\mu_t = \frac{1}{\frac{1}{\sigma_0^2} + \frac{r_t}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^t y_i}{\sigma^2} \right)$  so for the calculation of the last term, we have

$$\begin{aligned} B &= \mu_{B,t+1} - \mu_{\text{change}} \\ &= \frac{1}{\frac{1}{\sigma_0^2} + \frac{r_t+1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=t+1-r_t}^{t+1} y_i}{\sigma^2} \right) - \frac{1}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}} \left( \frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2} \right). \end{aligned} \quad (\text{A.20})$$

---

**Algorithm A1:** Pseudocode for Nas10\* for the Gaussian Estimation Task.

---

- 1: Specify  $m = p_c/(1 - p_c)$ ,  $\mu_0$ ,  $\sigma_0$ ,  $\sigma$  and  $\rho = \sigma^2/\sigma_0^2$ .
  - 2: Initialize  $\hat{\mu}_0$ ,  $\hat{\sigma}_0$ ,  $\hat{r}_0$  and  $t \leftarrow 0$ .
  - 3: **while** the sequence is not finished **do**
  - 4:   Observe  $y_{t+1}$   
       # Surprise
  - 5:   Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t)$  using equation A.23  
       # Modulation factor
  - 6:   Compute  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t), m)$  as in equation 2.9  
       # Expected mean
  - 7:   Compute  $\hat{\mu}_{t+1}$  using equation 4.49  
       # Expected time interval
  - 8:   Compute  $\hat{r}_{t+1} = (1 - \gamma_{t+1})(\hat{r}_t + 1) + \gamma_{t+1}$   
       # Expected variance
  - 9:   Compute  $\hat{\sigma}_{t+1} = \left[\frac{1}{\sigma_0^2} + \frac{\hat{r}_{t+1}}{\sigma^2}\right]^{-1}$   
       # Iterate
  - 10:    $t \leftarrow t + 1$
- 

We now rearrange the terms

$$B = \mu_t + \left( \frac{1}{\rho + 1} - \frac{r_t}{(\rho + 1)(\rho + r_t + 1)} \right) (y_{t+1} - \mu_t) - \mu_0 - \frac{1}{\rho + 1} (y_{t+1} - \mu_0), \quad (\text{A.21})$$

and finally we have

$$\hat{\sigma}_{t+1}^2 = \frac{\sigma^2}{\rho + 1} \alpha + (1 - \gamma)\gamma B^2. \quad (\text{A.22})$$

**A.2 Implementation of Nas10\*, Nas12\*, and Particle Filtering with One Particle for the Gaussian Estimation Task.** We provide here the pseudocode for the algorithms Nas10\*, Nas12\*, and Particle Filtering with one particle for the gaussian estimation task (see algorithms A1, A2, and A3). Observations are drawn from a gaussian distribution with known variance and unknown mean:  $y_{t+1} | \mu_{t+1} \sim \mathcal{N}(\mu_{t+1}, \sigma^2)$  and  $\theta_t = \mu_t$ . When there is a change, the parameter  $\mu$  is also drawn from a gaussian distribution,  $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$ . All three algorithms estimate the expected  $\mu_{t+1}$  (i.e.,  $\hat{\mu}_{t+1}$ ) upon observing a new sample  $y_{t+1}$ .

After rewriting equation 4.56, we have

$$\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t) = \frac{\sigma^2 + \hat{\sigma}_t^2}{\sigma^2 + \sigma_0^2} \exp \left[ -\frac{\mu_0^2}{2\sigma_0^2} - \frac{\hat{\mu}_t^2}{2\hat{\sigma}_t^2} + \frac{\frac{\hat{\mu}_t}{\hat{\sigma}_t^2} + \frac{y_{t+1}}{\sigma^2}}{2(\frac{\sigma^2}{\hat{\sigma}_t^2} + 1)} + \frac{\frac{\mu_0}{\sigma_0^2} + \frac{y_{t+1}}{\sigma^2}}{2(\frac{\sigma^2}{\sigma_0^2} + 1)} \right]. \quad (\text{A.23})$$

**Algorithm A2:** Pseudocode for Nas12\* for the Gaussian Estimation Task.

- 
- 1: Specify  $m = p_c/(1 - p_c)$ ,  $\mu_0$ ,  $\sigma_0$ ,  $\sigma$  and  $\rho = \sigma^2/\sigma_0^2$ .
  - 2: Initialize  $\hat{\mu}_0$ ,  $\hat{\sigma}_0$ ,  $\hat{r}_0$  and  $t \leftarrow 0$ .
  - 3: **while** the sequence is not finished **do**
  - 4:   Observe  $y_{t+1}$   
    # Surprise
  - 5:   Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t)$  using equation A.23  
    # Modulation factor
  - 6:   Compute  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t), m)$  as in equation 2.9  
    # Expected mean
  - 7:   Compute  $\hat{\mu}_{t+1}$  using equation 4.49  
    # Expected variance
  - 8:   Compute the expected variance  $\hat{\sigma}_{t+1}$  using equation A.22  
    # Expected time interval
  - 9:   Compute the expected time interval  $\hat{r}_{t+1} = \frac{\sigma^2}{\hat{\sigma}_{t+1}^2} - \frac{\sigma^2}{\sigma_0^2}$   
    # Iterate
  - 10:  $t \leftarrow t + 1$
- 

**Algorithm A3:** Pseudocode for Particle Filtering with One Particle for the Gaussian Estimation Task.

- 
- 1: Specify  $m = p_c/(1 - p_c)$ ,  $\mu_0$ ,  $\sigma_0$ ,  $\sigma$  and  $\rho = \sigma^2/\sigma_0^2$ .
  - 2: Initialize  $\hat{\mu}_0$ ,  $\hat{\sigma}_0$ ,  $\hat{r}_0$  and  $t \leftarrow 0$ .
  - 3: **while** the sequence is not finished **do**
  - 4:   Observe  $y_{t+1}$   
    # Surprise
  - 5:   Compute  $\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t)$  using equation A.23  
    # Modulation factor
  - 6:   Compute  $\gamma_{t+1} = \gamma(\mathbf{S}_{\text{BF}}(y_{t+1}; \hat{\mu}_t, \hat{\sigma}_t), m)$  as in equation 2.9  
    # Hidden state of particle
  - 7:   Sample  $c_{t+1}^{(1)} \sim \text{Bernoulli}(\gamma_{t+1})$   
    # Expected mean
  - 8:   **if**  $c_{t+1}^{(1)} = 0$  **then**
  - 9:      $\hat{\mu}_{t+1} \leftarrow \hat{\mu}_t + \frac{1}{\rho + \hat{r}_t + 1}(y_{t+1} - \hat{\mu}_t)$  and  $\hat{r}_{t+1} \leftarrow \hat{r}_t + 1$
  - 10:   **else**
  - 11:      $\hat{\mu}_{t+1} \leftarrow \mu_0 + \frac{1}{\rho + 1}(y_{t+1} - \mu_0)$  and  $\hat{r}_{t+1} \leftarrow 1$   
    # Expected variance
  - 12:   Compute the expected variance  $\hat{\sigma}_{t+1} = \frac{\sigma^2}{\hat{r}_{t+1} + \rho}$   
    # Iterate
  - 13:  $t \leftarrow t + 1$
- 

Note that the pseudocode for pfl 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.

## Acknowledgments

---

We thank both reviewers for their constructive and helpful comments. This research was supported by the Swiss National Science Foundation (200020 184615) and the European Union Horizon 2020 Framework Program under grant agreement 785907 (Human Brain Project, SGA2).

## References

---

- Adams, R. P., & MacKay, D. J. (2007). *Bayesian online changepoint detection*. arXiv:0710.3742.
- Aminikhanghahi, S., & Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2), 339–367.
- Barber, D. (2006). Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning Research*, 7, 2515–2540.
- Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge: Cambridge University Press.
- Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. Ph.D. diss., University College London.
- Behrens, T. E., Woolrich, M. W., Walton, M. E., & Rushworth, M. F. (2007). Learning the value of information in an uncertain world. *Nature Neuroscience*, 10(9), 1214.
- Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76, 198–211.
- Bogacz, R. (2019). *Dopamine role in learning and action inference*. bioRxiv:837641.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.
- Brown, S. D., & Steyvers, M. (2009). Detecting and predicting changes. *Cognitive Psychology*, 5(1), 49–67.
- Cummings, R., Krehbiel, S., Mei, Y., Tuo, R., & Zhang, W. (2018). Differentially private change-point detection. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*, 31 (pp. 10825–10834). Red Hook, NY: Curran.
- d’Acremont, M., & Bossaerts, P. (2016). Neural mechanisms behind identification of leptokurtic noise and adaptive behavioral response. *Cerebral Cortex*, 26(4), 1818–1830.
- Daw, N., & Courville, A. (2008). The pigeon as particle filter. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *Advances in neural information processing systems*, 20 (pp. 369–376). Red Hook, NY: Curran.
- Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3), 197–208.
- Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan & B. Rozovskii (Eds.), *The Oxford handbook of nonlinear filtering* (pp. 656–704), 3. Oxford: Oxford University Press.
- Doucet, A., & Tadić, V. B. (2003). Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics*, 55(2), 409–422.

- Efron, B., & Hastie, T. (2016). *Computer age statistical inference*. Cambridge: Cambridge University Press.
- Faraji, M., Preuschoff, K., & Gerstner, W. (2018). Balancing new against old information: The role of puzzlement surprise in learning. *Neural Computation*, 30(1), 34–83.
- Fearnhead, P., & Liu, Z. (2007). On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4), 589–605.
- Findling, C., Chopin, N., & Koechlin, E. (2019). *Imprecise neural computations as source of human adaptive behavior in volatile environments*. bioRxiv:799239.
- Fox, E., Sudderth, E. B., Jordan, M. I., & Willsky, A. S. (2011). Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4), 1569–1585.
- Frémaux, N., & Gerstner, W. (2016). Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in Neural Circuits*, 9, 85.
- Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 1(2), 127.
- Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., & Pezzulo, G. (2017). Active inference: A process theory. *Neural Computation*, 29(1), 1–49.
- George, C. P., & Doss, H. (2017). Principled selection of hyperparameters in the latent Dirichlet allocation model. *Journal of Machine Learning Research*, 18, 1–38.
- Gershman, S. J. (2019). *What does the free energy principle tell us about the brain?* arXiv:1901/07945.
- Gershman, S. J., Monfils, M.-H., Norman, K. A., & Niv, Y. (2017). The computational nature of memory modification. *Elife*, 6, e23763.
- Gershman, S. J., Radulescu, A., Norman, K. A., & Niv, Y. (2014). Statistical computations underlying the dynamics of memory updating. *PLOS Computational Biology*, 10(11), e1003939.
- Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D., & Brea, J. (2018). Eligibility traces and plasticity on behavioral time scales: Experimental support of neo-Hebbian three-factor learning rules. *Frontiers in Neural Circuits*, 12.
- Ghahramani, Z., & Hinton, G. E. (2000). Variational learning for switching state-space models. *Neural Computation*, 12(4), 831–864.
- Glaze, C. M., Kable, J. W., & Gold, J. I. (2015). Normative evidence accumulation in unpredictable environments. *Elife*, 4, e08825.
- Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian Bayesian state estimation. In *IEE Proceedings (Radar and Signal Processing)*, 140, 107–113.
- Heilbron, M., & Meyniel, F. (2019). Confidence resets reveal hierarchical adaptive learning in humans. *PLOS Computational Biology*, 15(4), e1006972.
- Huang, Y., & Rao, R. P. (2014). Neurons as Monte Carlo samplers: Bayesian inference and learning in spiking networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, 27 (pp. 1943–1951). Red Hook, NY: Curran.
- Huettel, S. A., Mack, P. B., & McCarthy, G. (2002). Perceiving patterns in random series: Dynamic processing of sequence in prefrontal cortex. *Nature Neuroscience*, 5(5), 485–490.

- Itti, L., & Baldi, P. F. (2006). Bayesian surprise attracts human attention. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems*, 18 (pp. 547–554). Cambridge, MA: MIT Press.
- Joshi, S., & Gold, J. I. (2020). Pupil size as a window on neural substrates of cognition. *Trends in Cognitive Sciences*, 24.
- Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 9(430), 773–795.
- Konovalov, A., & Krajbich, I. (2018). Neurocomputational dynamics of sequence learning. *Neuron*, 98(6), 1282–1293.
- Kopp, B., & Lange, F. (2013). Electrophysiological indicators of surprise and entropy in dynamic task-switching environments. *Frontiers in Human Neuroscience*, 7, 300.
- Kutschireiter, A., Surace, S. C., Sprekeler, H., & Pfister, J.-P. (2017). Nonlinear Bayesian filtering and learning: A neuronal dynamics for perception. *Scientific Reports*, 7(1), 8722.
- Legenstein, R., & Maass, W. (2014). Ensembles of spiking neurons with noise support optimal probabilistic inference in a dynamically changing environment. *PLoS Computational Biology*, 10(10).
- Lieder, F., Daunizeau, J., Garrido, M. I., Friston, K. J., & Stephan, K. E. (2013). Modelling trial-by-trial changes in the mismatch negativity. *PLoS Computational Biology*, 9(2).
- Lin, K., Sharpnack, J. L., Rinaldo, A., & Tibshirani, R. J. (2017). A sharp error analysis for the fused lasso, with application to approximate changepoint screening. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*, 30 (pp. 6884–6893). Red Hook, NY: Curran.
- Lisman, J., Grace, A. A., & Duzel, E. (2011). A neohebbian framework for episodic memory; role of dopamine-dependent late LTP. *Trends in Neurosciences*, 34(10), 536–547.
- Liu, J., & West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas, & N. Gordon (Eds.), *Sequential Monte Carlo methods in practice* (pp. 197–223). Berlin: Springer.
- Lomonaco, V., Desai, K., Culurciello, E., & Maltoni, D. (2019). *Continual reinforcement learning in 3D non-stationary environments*. arXiv:1905.1011.
- Loued-Khenissi, L., Pfeuffer, A., Einhäuser, W., & Preusschoff, K. (2020). Anterior insula reflects surprise in value-based decision-making and perception. *NeuroImage*, 116549.
- Maheu, M., Dehaene, S., & Meyniel, F. (2019). Brain signatures of a multiscale process of sequence learning in humans. *Elife*, 8, e41541.
- Mars, R. B., Debener, S., Gladwin, T. E., Harrison, L. M., Haggard, P., Rothwell, J. C., & Bestmann, S. (2008). Trial-by-trial fluctuations in the event-related electroencephalogram reflect dynamic changes in the degree of surprise. *Journal of Neuroscience*, 28(47), 12539–12545.
- Masegosa, A., Nielsen, T. D., Langseth, H., Ramos-López, D., Salmerón, A., & Madsen, A. L. (2017). Bayesian models of data streams with hierarchical power priors. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 (pp. 2334–2343). PMLR.

- Mathys, C., Daunizeau, J., Friston, K. J., & Stephan, K. E. (2011). A Bayesian foundation for individual learning under uncertainty. *Frontiers in Human Neuroscience*, 5, 39.
- Meyniel, F., Maheu, M., & Dehaene, S. (2016). Human inferences about sequences: A minimal transition probability model. *PLOS Computational Biology*, 12(12), e1005260.
- Modirshanechi, A., Kiani, M. M., & Aghajan, H. (2019). Trial-by-trial surprise-decoding model for visual and auditory binary oddball tasks. *NeuroImage*, 196, 302–317.
- Musiolek, L., Blankenburg, F., Ostwald, D., & Rabovsky, M. (2019). Modeling the n400 brain potential as semantic Bayesian surprise. In *Proceedings of the 2019 Conference on Cognitive Computational Neuroscience*. <https://ccneuro.org/2019/Papers/AcceptedPapers.asp>
- Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., & Finn, C. (2018). *Learning to adapt in dynamic, real-world environments through meta-reinforcement learning*. arXiv:1803.11347.
- Nassar, M. R., Bruckner, R., & Frank, M. J. (2019). Statistical context dictates the relationship between feedback-related EEG signals and learning. *Elife*, 8, e46975.
- Nassar, M. R., Rumsey, K. M., Wilson, R. C., Parikh, K., Heasley, B., & Gold, J. I. (2012). Rational regulation of learning dynamics by pupil-linked arousal systems. *Nature Neuroscience*, 15(7), 1040–1046.
- Nassar, M. R., Wilson, R. C., Heasley, B., & Gold, J. I. (2010). An approximately Bayesian delta-rule model explains the dynamics of belief updating in a changing environment. *Journal of Neuroscience*, 30(37), 12366–12378.
- Ostwald, D., Spitzer, B., Guggenmos, M., Schmidt, T. T., Kiebel, S. J., & Blankenburg, F. (2012). Evidence for neural encoding of Bayesian surprise in human somatosensation. *NeuroImage*, 62(1), 177–188.
- Özkan, E., Šmídl, V., Saha, S., Lundquist, C., & Gustafsson, F. (2013). Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters. *Automatica*, 49(6), 1566–1575.
- Papoulis, A., & Saunders, H. (1989). *Probability, random variables and stochastic processes*. American Society of Mechanical Engineers Digital Collection.
- Prat-Carrabin, A., Wilson, R. C., Cohen, J. D., & Da Silveira, R. A. (2020). *Human inference in changing environments with temporal structure*. bioRxiv:720516.
- Preuschoff, K., Hart, B. M., & Einhauser, W. (2011). Pupil dilation signals surprise: Evidence for noradrenaline's role in decision making. *Frontiers in Neuroscience*, 5, 115.
- Ryali, C., Reddy, G., & Yu, A. J. (2018). Demystifying excessively volatile human learning: A Bayesian persistent prior and a neural approximation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett, *Advances in neural information processing systems*, 31 (pp. 2781–2790). Red Hook, NY: Curran.
- Särkkä, S. (2013). *Bayesian filtering and smoothing* (Vol. 3). Cambridge: Cambridge University Press.
- Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3), 230–247.
- Schwartenbeck, P., FitzGerald, T., Dolan, R., & Friston, K. (2013). Exploration, novelty, surprise, and free energy minimization. *Frontiers in Psychology*, 4, 710.

- Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 20, 623–656; 27, 379–423.
- Shi, L., & Griffiths, T. L. (2009). Neural implementation of hierarchical Bayesian inference by importance sampling. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems*, 22 (pp. 1669–1677). Red Hook, NY: Curran.
- Storck, J., Hochreiter, S., & Schmidhuber, J. (1995). Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks* (Vol. 2, pp. 159–164). Piscataway, NJ: IEEE.
- Traoré, R., Caselles-Dupré, H., Lesort, T., Sun, T., Cai, G., Díaz-Rodríguez, N., & Filliat, D. (2019). *Discorl: Continual reinforcement learning via policy distillation*. arXiv:1907.05855.
- Wilson, R. C., Nassar, M. R., & Gold, J. I. (2010). Bayesian online learning of the hazard rate in change-point problems. *Neural Computation*, 22(9), 2452–2476.
- Wilson, R. C., Nassar, M. R., & Gold, J. I. (2013). A mixture of delta-rules approximation to Bayesian inference in change-point problems. *PLOS Computational Biology*, 9(7), e1003150.
- Yu, A. J. (2012). Change is in the eye of the beholder. *Nature Neuroscience*, 15(7), 933.
- Yu, A. J., & Cohen, J. D. (2009). Sequential effects: Superstition or rational behavior? In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems*, 21 (pp. 1873–1880). Red Hook, NY: Curran.
- Yu, A. J., & Dayan, P. (2005). Uncertainty, neuromodulation, and attention. *Neuron*, 46(4), 681–692.

---

Received April 27, 2020; accepted September 8, 2020.