UNIVERSITE CATHOLIQUE DE LOUVAIN

Learning and making decisions in changing environments with Variational Inference

by

Vincent Moens

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Institute of Neuroscience
Cognition And Systems

November 2018

Declaration of Authorship

I, VINCENT MOENS, declare that this thesis titled, 'Learning and making decisions in changing environments with Variational Inference' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

"Why did I add to the infinite
series another symbol?"

Jorge Luis Borges

UNIVERSITE CATHOLIQUE DE LOUVAIN

Abstract

Institute of Neuroscience
Cognition And Systems

Doctor of Philosophy

Learning and making decisions in changing environments with Variational Inference

by Vincent Moens

Statistics, Cognition and Learning – either in machines or in animals – have two common features: first, they are aimed at formulating prediction of unseen data. Second, they require a model of the observations to express these predictions. When data are acquired sequentially, the temporal structure of the acquisition can be taken into account to predict with a better accuracy the distribution of the variable of interest: for instance, an animal can predict the future from its past experience, and an algorithm can use past correlation of between successive observations to predict upcoming events.

The present work will first focus on data-driven and informative algorithms aimed at predicting and explaining the behaviour of human subjects.

We will be interested in linking together two fields of cognitive neuroscience: reinforcement learning, with a special focus on the study of inflexible behaviours, and decision making through the popular sequential sampling models.

Both of these topics will be studied through the lens of the theories of the Bayesian brain, whom assume that the central nervous system performs some sort of Bayesian inference to predict and, sometimes, explain the observations that are provided to it.

We will base the models presented in this thesis on three pillars: first, reinforcement learning will constitute the core psychological and neurophysiological model of learning how to behave. Next, we will frame the process of action selection in the popular framework of the Sequential Sampling family of decision making models. Finally, in Bayesian statistics, we will focus on Variational approximations to perform inference about the world (at the computational and algorithmic level) and to fit the models we propose to behavioural data.

Our contribution to the field of cognitive neuroscience will be first methodological, as we will propose a novel Bayesian approach to fit known models of decision making. Next, we will introduce a new Bayesian conceptual framework around the concept of inflexibility.

In the context of the popular Drift Diffusion Model, we will show that the trial-wise posterior distribution of latent variables of the Drift-diffusion model can be inferred simply by assuming that the data and generative process are correlated in a forward manner. We will provide the important result that, if the customary *i.i.d.* assumption is relaxed, the fit of the DDM parameters can be remarkably improved.

Next, we will get interested in a crucial question for any agent interacting with the environment: in the presence of surprising events, when would one's model need to be changed? Inability to change the model may be a serious disadvantage in unsteady environment, but having a model that is too volatile may impair the agent ability to accurately predict future events, especially in the presence of noisy observations. This

second part of our work will aim at treating the specific problem of modelling the emergence of inflexible behaviours in a Bayesian framework. The algorithm we will propose relies on the principle that the more data are stored in memory, the more precise is the inference the agent can achieve in a stable environment. We will propose a scheme of learning where this agent learns the stability of the environment in a hierarchical manner, adapting its forgetting rate in function of the surrounding volatility, and adapting this measure of the volatility in a similar manner.

Finally, we will tie these two algorithms in a convenient and interpretable model of learning and decision-making.

Acknowledgements

I thank Dr. Alexandre Zénon, for having made this project possible. Working with him on these various projects has been an invaluable experience, and is certainly the aspect of the thesis I will miss the most.

I also thank Dr. Andrea Alamia, Oleg Solopchuk and Monika Gergelyfi, who did bring all the support that one could hope from his colleagues and friends.

I gratefully acknowledge the thesis committee, which put all the necessary involvement in following this project.

I thank Pr. Julie Duque, for having taken care of the supervision when it was needed, and for having been so supportive during the last two years of the thesis.

Furthermore, I want to acknowledge the work of the jury, who accepted to take care of reviewing thoroughly this manuscript and helped to significantly improve its quality.

Above everything, my gratitude goes to my wife, Caterina, to whom I owe and dedicate this entire work. She made a great amount of sacrifices in order to make this work possible. She was supportive when times were hard, and shared my joy for the many happy events.

This manuscript is in memory of late Pr. Etienne Olivier, former supervisor, advisor and mentor. Etienne gave me the guidance and mentorship that constitute the basis of this manuscript. We would have wished him to be with us up to the end of this journey.

Contents

List of Figures

List of Tables

Abbreviations

ABC	**A**pproximate **B**ayesian **C**omputation
AE	**A**auto **E**ncoder
AVI	**A**mortized **V**artiational **I**nference
BG	**B**asal **G**anglia
CDF	**C**umulative **D**ensity **F**unction
CPU	**C**entral **P**rocessing **U**nit
CS	**C**hi-**S**quared
CVMP	**C**onjugate **V**artiational **M**essage **P**assing
DAG	**D**irected **A**cyclic **G**raph
DDM	**D**rift **D**iffusion **M**odel
DP	**D**ynamic **P**programming
EF	**E**xponential **F**amily
ELBO	**E**vidence **L**ower **BO**und
EM	**E**xpectation **M**aximization
EP	**E**xpectation **P**ropagation
FAD	**F**orward **A**automatic **D**ifferentiation
FIVO	**Fi**ltering **V**ariational **O**jective
GAN	**G**enerative **A**dversarial **N**etwork
GP	**G**aussian **P**rocess **U**nit
GPU	**G**raphics **P**rocessing **U**nit
GRU	**G**ated **R**ecurrent **U**nit
HAFVF	**H**ierarchical **A**daptive **F**orgetting **V**ariational **F**ilter
HGF	**H**ierarchical **G**aussian **F**ilter
HMC	**H**amiltonian **M**onte **C**arlo
i.i.d.	independent and identically distributed

IN	**I**nference-**N**network
IS	**I**mportance-**S**ampling
IVI	**I**mplicit **V**artiational **I**nference
IWAE	**I**mportance **W**eighted **A**uto-**E**ncoder
KL	**K**ullback-**L**eibler
KS	**K**olmogorov-**S**mirnov
MAP	**M**aximum **A** **P**osteriori
MCMC	**M**arkov **C**hain **M**onte **C**arlo
MC-RL	**M**onte **C**arlo **R**einforcement **L**earning
MDP	**M**arkov **D**ecision **P**rocess
ML	**M**aximum **L**ikelihood
NCVMP	**N**on-**C**onjugate **V**artiational **M**essage **P**assing
NQ	**N**umerical **Q**uadrature
POMDP	**P**artially **O**bservable **M**arkov **D**ecision **P**rocess
PSIS	**P**areto **S**moothed **I**mportance **S**ampling
(R)QMC	(**R**andomized) **Q**uasi-**M**onte **C**arlo
QS	**Q**-value **S**ampling ($\equiv$ Thompson Sampling)
RAD	**R**everse **A**automatic **D**ifferentiation
RAE	**R**ecurrent **A**auto **E**ncoder
RL	**R**einforcement **L**earning
RNN	**R**ecurrent **N**eural **N**etwork
RPE	**R**ward **P**rediction **E**error
RT	**R**eaction **T**ime
SARSA	**S**tate-**A**ction-**R**eward-**S**tate-**A**ction
SGD	**S**tochastic **G**radient **D**escent
SGVB	**S**tochastic **G**radient **V**artiational **B**ayes
SPRT	**S**equential **P**robability **R**atio **T**est
SSM	**S**equential **S**ampling **M**odels
SVI	**S**tochastic **V**ariational **I**nference
TAFC	**T**wo-**A**lternative **F**orced **C**hoice
TD	**T**emporal-**D**ifference
VAE	**V**artiational **A**uto-**E**ncoder
VB (VI)	**V**artiational **B**ayes (**I**nference)

VD	**V**artiational **D**ropout
rhs	**r**ight **h**and **s**ide
lhs	**l**eft **h**and **s**ide
st	**s**uch **t**hat
wrt	**w**ith **r**espect **t**o

Symbols

General notation:

x_i		Random variable of dimension 1
$\mathbf{x}_i$		Random variable of dimension >1
$\mathbf{x}$		Set of variables $\{ \mathbf{x}_k \}_{k=1}^{K}$
$\mathbf{x}_{\leq T}$		$\{ x_{t=1} \}_{t=1=1}^{T}$
$\widehat{\mathbf{x}}$		True value of $\mathbf{x}$
$\widetilde{\mathbf{x}}$		Sample of $\mathbf{x} \sim p(\mathbf{x})$
n	$\in 1:N$	Subject index
j	$\in 1:J$	Trial index
k	$\in 1:K$	Sample index
$\mathbb{E}_{p(x)}[f(x)]$	$= \int p(x)f(x)dx$	Expectation of $f(x)$ under probability distribution $p(x)$
$\mathbf{T}(\mathbf{x})$		Summary statistics of $\mathbf{x}$
$x \sim p(x)$		x is distributed as $p(x)$
$\frac{\partial f(x)}{x}$		Partial derivative of $f(x)$ wrt. x
$\nabla_{\mathbf{x}} f(\mathbf{x})$		Gradient of $f(\mathbf{x})$ wrt. $\mathbf{x}$

DDM and RL Nomenclature:

ξ	$\in \mathbb{R}$	Drift rate	a.u.
ζ	$\in \mathbb{R}^+$	Threshold	a.u.
ν	$\in [0;1]$	Relative start point	Ratio
z_0	$\in [0;\zeta]$	Start point	a.u.
r	$\in \mathbb{R}$	Reward	a.u.
a	$\in \mathbb{N}$	Action (choices) index	
t	$\in \mathbb{R}$	Reaction time (or time from trial start)	seconds
o	$\in \mathbb{R}^d$	Observation	

$s \quad \in \mathbb{N}$	State index
$\gamma \quad \in [0, 1)$	Temporal discount factor
$A(\mathbf{y})$ (resp. $B(\mathbf{y})$)	Log partition function of a member of the EF (resp. conjugate prior)

Variational Bayes Terminology:

$p(\mathbf{z} \mid \boldsymbol{\vartheta}_0) \equiv p_{\boldsymbol{\vartheta}_0}(\mathbf{z})$	Probability distribution with parameter $\boldsymbol{\vartheta}_0$
$q(\mathbf{z} \mid \boldsymbol{\vartheta}_j) \equiv q_k(\mathbf{z} \mid \boldsymbol{\vartheta}) \equiv q_{\boldsymbol{\vartheta}_j^{\mathbf{z}}}(\mathbf{z})$	Approximate posterior (at trial j) with parameter $\boldsymbol{\vartheta}_j$
$p(\mathbf{y} \mid \mathbf{x})$	Probability distribution of $\mathbf{y}$ conditioned on $\mathbf{x}$
$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \equiv p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})$	Probability distribution of $\mathbf{y}$ conditioned on $\mathbf{x}$ with parameters $\boldsymbol{\theta}$
$\boldsymbol{\epsilon}$	Auxiliary random variable with arbitrary distribution $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$
$\boldsymbol{\rho}$	IN – or encoder / recognition model – weights and biases
$\boldsymbol{\phi}$	Generative model – or decoder – weights and biases
$\mathcal{L}\left(q(\mathbf{z} \mid \boldsymbol{\vartheta})\right) \equiv \mathcal{L}\left(\boldsymbol{\vartheta}\right)$	ELBO associated with $q_{\boldsymbol{\vartheta}}$
$\mathbf{z}$	Latent variable of a (V)AE
$\mathbf{h}$	Hidden variable of an RNN

HAFVF:

$\boldsymbol{\theta}_0$	Prior of the HAFVF at model level
$\{\boldsymbol{\mu}_0, \boldsymbol{\kappa}_0, \boldsymbol{\alpha}_0, \boldsymbol{\beta}_0\}$	$\mathcal{NG}^{-1}$ parameters
$\boldsymbol{\phi}_0 \equiv \{\alpha_0^{\phi}, \beta_0^{\phi}\}$	Beta Prior of the HAFVF at first memory level
$\boldsymbol{\beta}_0 \equiv \{\alpha_0^{\beta}, \beta_0^{\beta}\}$	Beta Prior of the HAFVF at second memory level

To new horizons...

Preface

Understanding the brain, and, broadly speaking, studying human nature, can be regarded as a nested problem: a group of of individuals of a kind try to model the specifics of this kind. This is a peculiar aspect of countless science areas, such as philosophy, social science, psychology, neuroscience, medicine and even economics. While looking at ourselves in the mirror of science, we, as a species, quickly realized that we were going to need to invent specific tools to account for the observations we were making: observing is nothing without a model. Indeed, observations are pointless if some sort of causality is not accounted for. We believe that this process of inference is not only characteristic to science, but also to how the brain works as a whole.

The auto-referential nature of the problem of understanding the self has the effect that, frequently, some of scientific tools emerge as algorithmic models of the observations that are made, and inversely. Language was probably the first tool that enjoyed this peculiar property: it is the most critical - but also probably most complex - scientific tool of all. But language quickly transcended its use as a tool to become a one of the models of human thinking: the theory of mind is one such example of thinking processes rooted on speech.

Today, in some areas, tools and models become nearly indistinguishable: this is particularly true for the field of cognitive neuroscience, where the statistical tools we use to study the brain by its outputs (the behaviours it produces or the signals that we can measure) are sometimes the models of the functioning of the brain itself. Bayesian statistics are such an example: they can be used both as a toolbox to find how relevant a model is to a particular problem, or to build the model itself. They provide the tools to select the best explanatory model for the researcher, and similar strategies may be used by the brain to prune its models to the most relevant ones.

It is particularly interesting to note the many bridges that have been built between cognitive neurosciences and the field of Machine Learning (ML). The past two decades have indeed been tremendously prolific in both these areas, and a large amount of theories, models and discoveries have been achieved in a collaborative manner. In many ways, Artificial Intelligence has been mapped onto the brain, as the brain has been mapped onto artificial learning networks.

The most representative example of these joints efforts is most probably reinforcement learning, which was initially – at least partly – developed as a model of learning in experimental psychology. Its conciseness and efficiency then inspired the ML community, and led to substantial improvements of the vanilla algorithms that were initially proposed. Today, the two fields participate with the same effort, albeit with two drastically

different purposes, to build robust yet simple algorithms to learn, predict, and decide in changing environments.

The present work fits into this way of thinking: on the bases of the mathematical tools we will develop to study neurophysiological phenomena, we will also try to build correspondent models of brain functioning, and vice-versa.

Chapter 1

Introduction

A great deal of studies, theoretical or experimental, have been focusing on building a formal framework of learning and decision making in humans. This thesis will propose a new Bayesian causal model accounting for the emergence of inflexible behaviours in human subjects following prolonged training in a stable environment. We will treat this topic from the perspective of learning and decision making.

We will first introduce the theoretical framework of the thesis: our purpose will be to present and distinguish some of the mathematical tools and models that have been used to climb up the chain of causality, in order to understand the dispositions (conscious, strategic, neuronal) that elicited a particular action.

Then (Chapter 2), we will focus on solving some issues when fitting behavioural data to a widely used model in neuroscience: the Drift Diffusion Model (DDM) [1], which will constitute the base of the actor-critic model we will present in the subsequent chapters. The DDM accounts for decisions in the broad experimental framework of Two-Alternative Forced Choice tasks (TAFC) [2], where subjects (animals [3] or humans [4]) are asked to choose one of the two actions that are proposed to them. Due to the richness and straightforward interpretability of its parameters, the DDM and related Sequential-Sampling Models (SSM) [5, 6] have received much attention in the past fifty years, to finally become conventional statistical tools and models of human behaviours. However, fit of the DDM – especially in its full version [7] – is hard to achieve, because it usually suffers from a high computational burden [8] for a quality of fit that is mostly poor [9, 10].

Surprisingly, most methods, if not all, that have been used to fit the DDM have relied on the assumption that the data were identically and independently distributed (*i.i.d.*) given the global, subject-wise parameters. At the most, authors used an informative

model where the parameters of the DDM were assumed to covary with some other parameters coming from an adjacent informative model: in RL, for instance, the DDM has been used as a choice rule instead of the more widespread softmax function [11, 12]. Some research groups, such as Turner and colleagues [13, 14, 15, 16] or Frank and colleagues [17] have suggested that inference about the DDM parameters could be improved by accounting for the cross-correlation between the DDM parameters and some neurophysiological measure.

We will therefore develop further this idea of relaxing the *i.i.d.* assumption about the distribution of the DDM parameters in a data-driven way. For this purpose, we will first frame the DDM in a Bayesian context, present a simple Variational implementation of this model, and finally provide a Recurrent Auto-Encoding algorithm (RAE-DDM) that takes advantage of the trial-to-trial cross-correlation of the behavioural observations and related DDM parameters. In this chapter, we will give a proof of concept of the efficiency of this method and discuss the advantages and disadvantages of this novel method.

In the following chapters, we will go on using the DDM in more informative models. The common thread of Chapters 3 and 4 will consist in deriving and studying a Machine Learning algorithm that will address the problem of change detection in a perspective inspired by the foremost experimental observations in the Reinforcement Learning literature, namely the emergence of inflexible behaviours following prolonged training in stable environments [18]. This model, the Hierarchical Adaptive Forgetting Variational Filter, will be first framed theoretically as a generic tool for change detection in various machine learning contexts Chapter 3: we will propose various applications of this algorithm in the fields of system identification and stochastic non-linear optimization.

Next, in Chapter 4, we will further detail how this algorithm can be used with the purpose of studying RL. This chapter will also be the occasion for us to tie together the learning algorithm and the DDM into a full Actor-Critic-HAFVF model of learning and decision. The *i.i.d.* variational optimization scheme of the DDM proposed in Chapter 2 will be used to optimize this model of decision making.

In Chapter 5, we will briefly discuss the results of the previous chapters and give some intuition of future works that might be conducted to refine the models proposed earlier.

To build a theoretical framework around our work, we will divide the introduction in three main parts. Because this work is aimed to be understandable by a wide and diverse audience, we will start each of these sections by giving a broad overview of the corresponding fields, before going into more details when necessary. As it is probably the oldest of all the ideas that will be used in this thesis, and because it will be a front and center concept herein, **Bayesian reasoning** will be introduced first in Section 1.1.

From there, we will pursue the same logical flow and introduce Approximate Inference (and more specifically Variational Bayes) as a family of algorithms that are aimed at using the Bayes rule in practice (Section 1.1.2). A brief introduction to the theory of the Bayesian brain and more specifically to Active Inference Section 1.1.3 will be given at the end of this section.

Having introduced these methods of inference, we will focus on the problem of decision making. Reinforcement Learning is the archetype of a machine learning algorithm put into action: Section 1.2 where it is described will give us the occasion to link inference methods, presented in the first section, to biologically-driven decision making algorithms, and specifically to Sequential Sampling Algorithms, presented in the last part (Section 1.3).

In the final section of this chapter, we will bring together these three fields to give an intuition on how each of them might help to solve the problem of predicting actions of animals evolving in a structured but volatile environments.

1.1 Bayesian Reasoning

Imagine that you are a young intern in an ER department in a big hospital. You have just started to practice a few weeks ago. The day has been long, and you are close to finish your shift, in which you saw 15 patients, 10 of which who came for a syndrome that you diagnosed as a virulent flu. It is indeed mid-February, and the epidemic is particularly strong this year. The last patient you will see today is in the waiting room. The nurse told you he came for some kind of cough, without further information. You might legitimately be inclined to bias already your diagnosis for a specific disease, similar to the ones you have seen before. Now consider you observe an unusual pattern of symptoms for a flu. Bayesian reasoning is most probably the technique you will use to tell if the patient has or does not have a flu: making a decision in such circumstances is – quite trivially – not identical to make an isolated diagnosis. Indeed, the observed frequency of the event "flu" in the days before will presumably bias your judgment towards a major propensity to classify respiratory syndromes as belonging to the same disease category. However, the large amount of observations at your disposal will also sharpen your classification acuity: you will be more able than before to classify an event as likely to co-occur with a flu or not.

This contrasts even more with the strategy that a well-documented but poorly trained intern would adopt: having been informed about the frequency (i.e. the likelihood) of the symptoms in patients with a flu would certainly be a valuable plus to diagnose the disease, which is framed as the opposite problem: estimating the likelihood of having the flu given the symptoms. What this poorly trained intern lacks is a *prior* knowledge of this probability of having the flu. This may seriously harm the discriminative power of this medical doctor.

We can easily rephrase all these concepts in Bayesian terms: let the probability of coughing given the disease be the *likelihood* of the symptom. Let the belief about the probability of having the flu before seeing the patient (i.e. depending on and only on previous observations and initial knowledge) being the *prior belief*. We call the *posterior probability* the probability of having a flu given that the patient is coughing. The Bayes theorem states that

$$\underbrace{p(\mathbf{y} \mid x)}_{\text{posterior probability}} = \frac{\overbrace{p(x \mid \mathbf{y})}^{\text{Likelihood}} \overbrace{p(\mathbf{y})}^{\text{Prior}}}{\underbrace{p(x)}_{\substack{\text{Model Evidence} \\ \text{or Marginal likelihood}}}} \tag{1.1}$$

where x is the symptom (the observation), and $\mathbf{y}$ is some specific disease, (the latent variable, which we model as the unknown cause of the observation). The Bayes Rule

in Equation (1.1) gives us a way to *invert the probability* of our initial knowledge about the likelihood of coughing when the patient has a flu to the probability of having a flu given that the patient is coughing. Crucially, this posterior probability depends linearly on both the likelihood and the prior.

An important term in Equation (1.1) is the denominator of the rhs: $p(x)$ ensures that the posterior probability is normalized (i.e. that it integrates to 1 over the space of $\mathbf{y}$). This quantity will be given various names in what follows, depending on the context: $p(x)$ is sometimes called the *marginal likelihood*, because it is equal to the joint probability marginalized over the space of $\mathbf{y}$:

$$p(x) = \int p(x \mid \mathbf{y})p(\mathbf{y})dy \tag{1.2}$$

$$= \int p(\mathbf{y} \mid x)p(x)dx \tag{1.3}$$

$$= \int p(x, \mathbf{y})dy \tag{1.4}$$

The marginal is sometimes called the model evidence, as it is equal to the probability of the data conditioned on the model and, importantly, not on any specific value of $\mathbf{y}$. Formulated as such, the model is a crucial quantity in Bayesian Statistics, as it can allow the statistician to compare models based on their capacity to predict data *irrespective of* $\mathbf{y}$. This contrasts with a common and intuitive interpretation of Maximum Likelihood methods and related information criteria, where the likelihood of a model being better than another is conditioned on a fit of $\mathbf{y}$ st $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(x \mid \mathbf{y})p(\mathbf{y})$, where $p(\mathbf{y})$ is a chosen prior distribution in the case of Maximum-A-Posteriori (MAP) estimation, or an unnormalized uniform prior in the case of a simple Maximum Likelihood (ML) approach. This justifies the reference sometimes made to ML methods as *point estimates*, in contrast with Bayesian methods which try to estimate a *distribution* rather than finding a point with a given property in this distribution. We will see in Chapter 2 that ML approaches to statistical inference can find a grounded Bayesian justification for large datasets with nice properties.

We note here an important, although mostly overlooked, weakness of the model evidence assessment for model comparison: although the value of $\mathbf{y}$ has been marginalized out of the likelihood function, comparing $p(x \mid M_1)$ and $p(x \mid M_2)$ for two models M_1 and M_2 still reduces to ML *at the model level*. In other words, a true Bayesian comparison of model evidence should be corrected over the space of the models, as we would be interested in the model posterior probability defined as

$$p(M_1 \mid x) = \frac{p(x \mid M_1)p(M_1)}{\sum_{m=1}^{\infty} p(x \mid M_m)p(M_m)},$$

where we have made explicit the fact that an infinite amount of models could be built to account for a specific dataset.

This problem is obviously unidentifiable, and we have no choice but to rely on maximum likelihood methods when comparing models. One should, however, keep in mind that comparing many models with a single dataset multiplies the chances of local maxima, where a model is suited for a specific dataset but not in general.

Several techniques have been developed to circumvent this problem: some models we will use will allow us to make inference about unseen data: then, the best model would be the one for which the probability of the new data given the model and the data that have been used to train the various models is maximized. The use of model evidence for model comparison is discussed in Box 1.2.9.

Another interesting feature of the Bayes formula is that, under the assumption that the data are independent and identically distributed (*i.i.d.*), the posterior probability can be expressed recursively:

$$p\left(\mathbf{y} \mid x_j, \mathbf{x}_{<j}\right) = \frac{p\left(x_j \mid \mathbf{y}\right) p\left(\mathbf{y} \mid \mathbf{x}_{<j}\right)}{p\left(x_j\right)} \tag{1.5}$$

where it appears that the prior distribution at time j is the posterior probability at time $j-1$.

1.1.1 Tractable Bayesian Inference: The Exponential Family

Evaluation the posterior probability for a given parameter $\mathbf{y}$ is not trivial for many models, as the integral in Equation (1.2) is most of the time intractable and/or expensive to obtain. Many members of the exponential family of distributions [19, 20] enjoy the convenient property that, if the prior probability is chosen to be *conjugate* to the likelihood, in a sense that will be made explicit short after, the posterior probability density of this model can be shown to have the same form as the prior.

A distribution is said to belong to the exponential family if its probability density can be formulated with the canonical form:

$$p\left(x \mid \mathbf{y}\right) = h(x) \exp\left\{\mathbf{y}^T \mathbf{T}(x) - A(\mathbf{y})\right\} \tag{1.6}$$

where $\mathbf{y}$ is said to be the *natural parameter* of the distribution, h(x) is the base measure, $\mathbf{T}(x)$ is the sufficient statistics of x and $-A(\mathbf{y})$ is a log-normalizer (or log-partition

function). For the sake of sparsity of the notation and without loss of generality[1], the base measure $h(x)$ will be mostly discarded from the following equations. Therefore, the reader should keep in mind that this slight abuse of notation has been adopted. Note that Equation (1.6) requires us to express the parameters of the distribution in the form of natural parameters. For instance, in the case of a Gaussian distribution, the sufficient statistics of x is $\mathbf{T}(x) = \{x, x^2\}$ and, therefore, $\mathbf{y} = \left\{\frac{\mu}{\sigma^2}, \frac{-1}{2\sigma^2}\right\}$.

The exponential family has several interesting properties, some of which we will detail here: first, the product of distributions from the exponential family is still from the exponential family (closure under multiplication). Second, the expected value of the sufficient statistics has the convenient form [21]

$$\mathbb{E}_{p(x|\mathbf{y})}\left[\mathbf{T}(x)\right] = \nabla_{\mathbf{y}}\left\{A(\mathbf{y})\right\} \tag{1.7}$$

and, following this, the covariance of $\mathbf{T}(x)$ given by $p(x \mid \mathbf{y})$ is equal to:

$$\mathbb{E}_{p(x|\mathbf{y})}\left[\mathbf{T}(x)\mathbf{T}(x)^T\right] - \mathbb{E}_{p(x|\mathbf{y})}\left[\mathbf{T}(x)\right]\mathbb{E}_{p(x|\mathbf{y})}\left[\mathbf{T}(x)\right]^T = \nabla^2_{\mathbf{y}}\{A(\mathbf{y})\}\,.$$

Another interesting property of the exponential family is that some of its members enjoy a tractable conjugate prior also belonging to the exponential family:

$$\begin{aligned} p\left(\mathbf{y} \mid \boldsymbol{\theta}_0\right) &= \exp\left\{\mathbf{T}(\mathbf{y})^T\boldsymbol{\theta}_0 - B(\boldsymbol{\theta}_0)\right\} \\ \text{where } \mathbf{T}(\mathbf{y})^T\boldsymbol{\theta}_0 &= \mathbf{y}^T\boldsymbol{\theta}_0^{\boldsymbol{\xi}} - \theta_0^{\eta}A(\mathbf{y}) \\ \text{and } \boldsymbol{\theta}_0 &= \{\boldsymbol{\theta}_0^{\boldsymbol{\xi}}, \theta_0^{\eta}\}. \end{aligned} \tag{1.8}$$

One can easily check that Equation (1.8) has the required form to be a member of the exponential family. If such distribution exist, then the posterior probability of $\mathbf{y} \mid \mathbf{x}$ for some $\mathbf{x} = \{x_k\}_{k=1}^K$ can be expressed as:

$$\begin{aligned} p\left(\mathbf{y} \mid \mathbf{x}\right) &= \exp\left\{\mathbf{T}(\mathbf{y})^T\boldsymbol{\theta} - B(\boldsymbol{\theta})\right\} \\ \text{where } \mathbf{T}(\mathbf{y})^T\boldsymbol{\theta} &= \mathbf{y}^T\boldsymbol{\theta}^{\boldsymbol{\xi}} - \theta^{\eta}A(\mathbf{y}) \\ \text{and } \boldsymbol{\theta}^{\boldsymbol{\xi}} &= \boldsymbol{\theta}_0^{\boldsymbol{\xi}} + \sum_{j=1}^{J}\mathbf{T}(x_j) \\ \theta^{\eta} &= \theta_0^{\eta} + J. \end{aligned} \tag{1.9}$$

Note that the posterior probability shares the same form as the prior, with natural parameters being expressed as a function of the sufficient statistics of $\mathbf{x}$ and of the amount of data J. In line with the recursive posterior rule of Equation (1.5), a simple

[1] The two expressions are equivalent assuming that $y' \triangleq \begin{bmatrix} y \\ 1 \end{bmatrix}$ and that $\mathbf{T}'(x) \triangleq \begin{bmatrix} \mathbf{T}(x) \\ \log h(x) \end{bmatrix}$

interpretation of this equation is that $\theta_J^\eta = \theta_0^\eta + J$ is the memory of the system at time J, whereas $\boldsymbol{\theta}_J^{\boldsymbol{\xi}}$ determines the form of the posterior distribution.

The exponential family has many interesting members: in this thesis, we will mostly use the univariate Gaussian distribution with its Normal-Inverse-Gamma prior:

$$\mu \sim \mathcal{N}\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right) \qquad \sigma^2 \sim \mathcal{G}^{-1}\left(\alpha_0, \beta_0\right),$$

the multivariate Gaussian distribution with its Normal-Inverse-Wishart prior:

$$\boldsymbol{\mu} \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \frac{\Sigma}{\kappa_0}\right) \qquad \Sigma \sim \mathcal{W}_{\eta_0}^{-1}\left(\boldsymbol{\Lambda}_0\right),$$

and the multinomial distribution $t \sim \mathcal{M}(\mathbf{p})$ (where $t \in 1:d$ and $\mathbf{p} \in \mathbb{R}^d$) with its conjugate prior, the Dirichlet (or Beta if $d = 2$) distribution $\mathcal{D}(\boldsymbol{\pi})$.

1.1.2 (Deterministic) Approximate Inference

Most of the time, the posterior distribution of $\mathbf{y}$ does not enjoy the peculiar form detailed in the last paragraphs. Often, also, the computational cost of estimating the exact posterior probability is unreasonably high and precludes the use of simple update formulas that would require complex operations on lots of data. Hence, in many cases, one must rely on some approximation of $p\left(\mathbf{y} \mid x\right)$. Roughly, the field of Bayesian methods to estimate a posterior distribution can be divided in two sub-categories[2]: the first is *simulation*-based techniques[3], such as Markov-Chain Monte-Carlo (MCMC), Importance Sampling (IS) [23], Approximate Bayesian Computation [24], or Particle Filtering [25] use samples $f(\widetilde{\mathbf{y}})$ with $\widetilde{\mathbf{y}} \sim p\left(\mathbf{y}\right)$ to approximate some target value

$$\int f(\mathbf{y}) p(\mathbf{y}) d\mathbf{y} \approx \frac{1}{K} \sum_{k=1}^{K} f(\widetilde{\mathbf{y}}_k).$$

If one can simulate from the posterior distribution, then the posterior expectation of any function involving $\mathbf{y}$ can be approximated.

These techniques have numerous advantages: they are easily applicable to many problems, as the algorithms used to draw these samples are usually blind to the structure of the model. Also, they are asymptotically exact, and, the estimators they provide usually (but not always [26]) have a bounded variance. These qualities usually come with a high

[2] Here, we will use the nomenclature proposed by Bishop [22], where simulation-based techniques are opposed to approximate inference.

[3] We describe here general properties of simulation algorithms. Specific models may, however, differ from the description made here.

computational cost: for MCMC, for instance, several chains usually need to be drawn in order to make accurate predictions and to check the convergence of the algorithm [27]. Moreover, each chain may require a substantial amount of time, computational and storing resources. Finally, most algorithms are not scalable to unseen data, which makes their use for large datasets unattractive.

Box 1.1.1| Marginal likelihoods and Bayesian inference using MCMC

A ubiquitous problem in modern physics, statistics and Machine Learning is to compute marginal likelihoods of the form $p(x) = \int p(x \mid \mathbf{y})\, p(\mathbf{y})$. Computation (or approximation) of this quantity is required among other things to perform the estimation of the free energy in physics [28], for type II Maximum Likelihood (aka empirical Bayes) [29, 30], to compare models [31] (hence the alternative name *model evidence*), and to perform Bayesian Model averaging [32], posterior predictive checks [33, 34] etc.

Many techniques have been developed to this aim, and they can be mostly divided in three categories: (approximate) analytic solutions (such as the Laplace method ([35] and Paragraph 1.1.2.2.4) and related approximations [36], Expectation Propagation, Variational Inference etc.) ; numerical integration and Monte Carlo integration. Here, we give some perspective of the latter option. The most naive approximation of the marginal likelihood is given by its simple Monte Carlo estimator, which consists in sampling $\widetilde{\mathbf{y}}_k \sim p(\mathbf{y})$, and computing the average

$$p(x) \approx \frac{1}{K} \sum_{k=1}^{K} p(x \mid \widetilde{\mathbf{y}}_k)$$

but this estimator, even if unbiased, can still suffer from a high variance because the prior $p(\mathbf{y})$ can assign a high mass where $p(x \mid \mathbf{y})$ has a low mass, and inversely. As we will see later, IS [23] can give acceptable results for this kind of problems, especially when $\mathbf{y}$ is low-dimensional. In short, IS works by simulating variables from a proposal distribution $q(\mathbf{y})$ and weighting these samples with the ratio $\frac{p(\mathbf{y})}{q(\mathbf{y})}$. This introduces a second problem, which is to select the best proposal distribution for the integrands and data at hand. Intuitively, it makes sense to select a distribution $q(\mathbf{y})$ with heavier tails than $p(\mathbf{y})$, which will ensure that no subspace with a high mass is left unexplored, but we see that selection of the proposal distribution requires some initial knowledge about the integrants: this introduces an undesirable form of circularity in the approximation scheme.

These two approaches (MC and IS) both ignore some specific features of the problem, and the two following issues may be raised against the use of these methods to approximate an integral [37]. First, the shape of $p(\mathbf{y})$ (or worse,

$q(\mathbf{y})$) which is used to sample does not convey any information about $p\,(x \mid \mathbf{y})$. It is easy to see that, with IS, if two *different* densities $q_1(\mathbf{y})$ and $q_2(\mathbf{y})$ give (by chance) the *same* samples $\{\,\widetilde{\mathbf{y}}_k\,\}_{k=1}^{K}$ then they will give a different estimate of the integral. It can also be observed that with both methods, if $p_1(x \mid \mathbf{y})$ and $p_2(x \mid \mathbf{y})$ are two different distributions, the quality of the approximation will be different because the sampling scheme will not adapt to this difference. The second objection raised by [37] is that the sampling procedure does not convey information about the space between the samples. Note for instance that if two samples $\widetilde{\mathbf{y}}_1$ and $\widetilde{\mathbf{y}}_2$ are identical (or very close) the information brought by one of them is irrelevant, and the variance of the estimator will be underestimated in an IS scheme. These two objections essentially follow the observation that both MC and IS generally violate the likelihood principle [38].

When the number of dimensions of $\mathbf{y}$ grows, IS becomes more difficult to apply. In fact, the choice of $q(\mathbf{y})$ is crucial as this distribution should already be a good approximation of $p\,(\mathbf{y} \mid x)$ and cover the relevant regions of $p\,(x \mid \mathbf{y})$ in order to maximize the information brought by each sample [23]. This choice is therefore more difficult in high dimensions, as the chance that $q(\mathbf{y})$ is suboptimal increases. Annealed Importance Sampling [39] and quite similarly Bridge and Path sampling [28] provides solutions to this problem by using the Simulated Annealing optimization procedure in the context of simulation in statistics. Essentially, it uses a sequence of proposal distributions that starts from the prior and converges to the posterior distribution. Like Metropolis-Hastings [40] and Gibbs Sampling [41], these integration techniques use a set of samples with a high auto-correlation, which suggests that these techniques may only be valid for large sample size.

Rassmussen and Ghahramani [42] propose Bayesian Monte Carlo method (aka Bayesian Quadrature [43, 44]) where the first term of the integral $p\,(x \mid \mathbf{y})$ is considered as a random variable with a prior expressed as a Gaussian Process (GP). From there, a posterior of $p(x \mid \mathbf{y})$ is inferred from the samples given by $p\,(\mathbf{y})$. This gives us a posterior distribution over the values of the integral. In practice, this can help because the posterior distribution of $p\,(x \mid \mathbf{y})$ gives information about this pdf, in the sense that it can interpolate its behaviour in regions that have not been sampled. This comes however at the cost of storing and inverting a matrix of the size of the sample size.

Finally, it is possible to use a deterministic – i.e. non-(pseudo)-random – sequence to mimic randomness while ensuring that most of the space has been explored. This is the principle behind Quasi-Monte Carlo (QMC) methods [45, 46, 47], where a sequence of random variables $\mathbf{u} = \{\,u_k\,\}_{k=1}^{K}$ is mapped to the space

dictated by $p(\mathbf{y})$. This is not properly speaking a sampling technique (since there is no randomness involved) but it enjoys a theoretical and practical error which is usually much lower than the one of Monte-Carlo techniques for problems of low-dimension. Indeed, unlike for classical Monte-Carlo which has an error of order $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$, the QMC error has an upper bound of $\mathcal{O}\left(\frac{(\log K)^d}{K}\right)$ and therefore depends on the number of dimensions of the problem d.

The second sub-category of Bayesian inference algorithms is *Approximate inference*, which solves part or all of the issues of MCMC, yet at a certain cost. In the following paragraphs, we will detail the two most important family of approximate inference algorithms: Expectation propagation and Variational Inference.

Approximate inference works by optimizing a proxy to the true posterior distribution (which is unknown) in order to minimize some predefined measure of discrepancy between the two. The type of discrepancy that is used defines the family of methods. This leaves us with the problem of choosing (1) a measure of discrepancy, (2) a procedure to minimize the discrepancy and (3) a good candidate approximation for the posterior probability. The order of these challenges matter: the measure of discrepancy determines the methods than can be used, and the optimization procedure conditions the approximate posterior family that can be efficiently implemented. Each of these challenges will be discussed in the following.

Another important feature of deterministic approximate inference that justifies its choice for the models we will develop is that it usually provides a loss function that is not only differentiable wrt the approximate posterior parameters, but also wrt the prior parameters. The assumption (whether it is theoretically motivated or not) that an external agent makes decisions using such algorithms (internally) has the compelling advantage that we can quite easily perform inference about the parameters governing this process (externally), as the process is deterministic and sometimes differentiable Figure 1.1. This contrasts with simulation-based techniques, where differentiating the inference process can be quite hard, if not impossible. Besides, the so-called Free-Energy minimization, whose core justification lies precisely in variational inference, has now gather a fair amount of empirical [48] and theoretical [49, 50] justifications in neuroscience.

Approximate inference alleviates the need to explicitly estimate the true posterior distribution $p(\mathbf{y} \mid \mathbf{x})$ by recurring to an approximate posterior $q(\mathbf{y})$ which can take an arbitrary form. The form of this distribution is selected for mathematical convenience. Our aim will be to reduce a measure of discrepancy between the true (and unknown) posterior distribution and the approximate one. Classically, the discrepancy measure

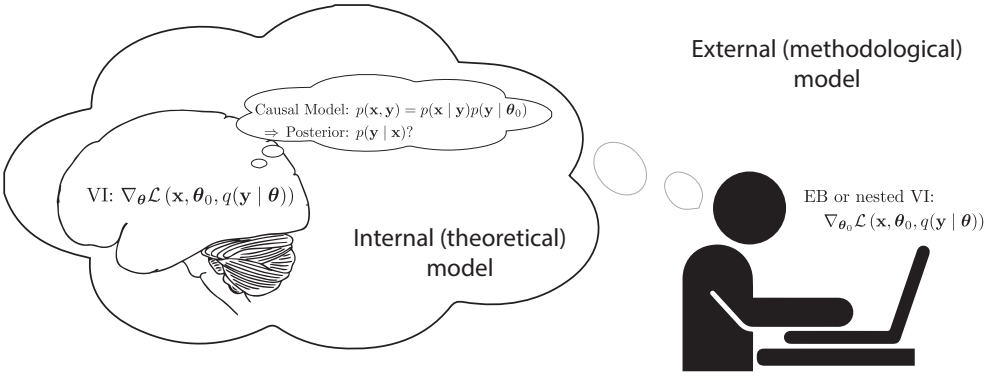


FIGURE 1.1: Active inference and other framework see the brain as a Bayesian machine, making inference about an internal causal model $p(\mathbf{x}, \mathbf{y})$ using approximate inference (e.g. Free Energy minimization). Besides their theoretical motivations, these models can have the advantage that the inference process can be differentiated wrt the subject (e.g. participant) prior $p(\mathbf{y} \mid \boldsymbol{\theta}_0)$: this gives us the possibility to readily infer the posterior probability of $\boldsymbol{\theta}_0$ given the data using some sort of nested inference (nested VI in the graph). At this level, differentiability of the external model is a desirable feature, because it allows us to use gradient descent and similar techniques to fit the model to the data.

used is the Kullback-Leibler (KL) divergence. The KL divergence is a statistical quantity that echos to many concepts in physics, neuroscience and machine learning. From an information theory perspective, $D_{KL}\left[f||g\right]$ can be intuitively understood as (a lower bound to) the number of bits (or nats in the continuous case) lost while coding f with g. From a theoretical physics point of view, the KL divergence is equal to the total energy of the system minus the amount of energy that can still be used in this system (the Helmoltz' *Free energy*): Minimizing the free energy maximizes the efficiency. Following this interpretation, we could compare the optimal distribution $q^* \triangleq \arg\min_q D_{KL}\left[q||p\right]$ to a engine that we would use to make the most efficient use of the energy available. It becomes then clear with this analogy that we will need to specify what form should q^* have in general, and how we can use it to consume most of the energy in the system. The implication of the KL divergence in computational neuroscience will be discussed in Section 1.1.3. This measure has several properties that one should keep in mind in the following paragraphs: first, as opposed to a *distance* measure, the KL *divergence* is not symmetric, meaning that for two measures f and g, $D_{KL}\left[f||g\right] \neq D_{KL}\left[g||f\right]$. Second, the KL divergence is strictly non-negative[4]. Third, it equals zero if and only if f and g match exactly.

The fact that the KL divergence is not symmetric suggests that it could be used in two different ways: the problem of matching q and p could be achieved by minimizing either $D_{KL}\left[q||p\right]$ or $D_{KL}\left[p||q\right]$, or a mixture of the two. The first approach is known

[4]we will disregard the cases where the KL divergence tends to infinity in what follows, which can occur for instance when the f and g have a different domain

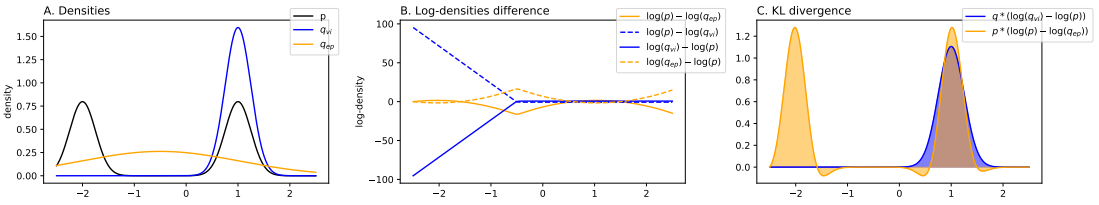


FIGURE 1.2: KL divergence for a one-dimensional bimodal posterior distribution for VI (blue) and EP (orange). **A.** Densities of target posterior p (black), and related approximate posteriors for VI and EP at optimality. VI clearly captures one mode only, whereas EP assigns a high density to a space where the posterior density is approximately null. **B.** (Negative) Difference of log-probabilities between true posterior and approximate posterior. **C.** The expected value of **B.** gives the KL divergence (filled areas), which in this specific case is lower for VI than EP (0.69 vs 1.11 respectively).

as Variational Inference (VI) (or Variational Bayes, VB[5]), the second as Expectation propagation (EP) (see Figure 1.2 [6]. For completeness, we will first briefly introduce the classical implementation of EP in order to give an intuition about the properties of this inference scheme. We will then detail the core principle of VI, and explore some of its recent developments. Note that, although only VI will be used in this thesis, we find it useful to give a broad overview of Approximate inference in general, hoping that it will provide the reader with the necessary tools to judge the advantages and disadvantages of each method

1.1.2.1 Expectation Propagation

We will describe EP in its most classical form [52, 22], where the following assumptions hold: first, we will assume that the approximate distribution $q(\mathbf{y})$ belongs to the exponential family:

$$q(\mathbf{y}) \triangleq \exp\left\{\mathbf{T}(\mathbf{y})^T \boldsymbol{\vartheta} - B(\boldsymbol{\vartheta})\right\}.$$

Most of the time, $q(\mathbf{y})$ is assumed to be Gaussian, which can be applied to almost any problem under suitable transformation of $\mathbf{y}$ [53].

Second, we will restrict ourselves to models that can be formulated as:

$$p(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^{J} p(x_j \mid \mathbf{y}) p(\mathbf{y}). \tag{1.10}$$

[5] In this thesis, we will use VB and VI interchangeably.

[6] We acknowledge that this nomenclature is quite arbitrary: some authors prefer to integrate both VI and EP under the umbrella of Variational inference [51]. We find it useful, however, to set the boundary between VI and EP as a function of the KL terms order, as this will prove to be a better shared nomenclature when we will refer to works achieved in Deep Learning and related fields.

Recall that our aim is to retrieve from this model and corresponding data two key quantities: first, an estimate of the posterior probability density function $p(\mathbf{y} \mid \mathbf{x})$ for inference and, second, an estimate of the marginal likelihood for model comparison $p(\mathbf{x})$. The model structure defined in Equation (1.10) belongs to the general family of models that enjoy the form

$$p(\mathbf{x}, \mathbf{y}) = \prod_{j=0}^{J} f_j(\mathbf{y}) \tag{1.11}$$

where we have implicitly used the equivalence $p(\mathbf{y}) \triangleq f_0(\mathbf{y})$. Similarly, the posterior probability can be rewritten as

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{x}) &= \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})} \\ &= \frac{1}{p(\mathbf{x})} \prod_{j=0}^{J} f_j(\mathbf{y}). \end{aligned}$$

Since we assume that the approximate posterior belongs to the exponential family too, we can formulate it as a product easily

$$q(\mathbf{y}) = \frac{1}{Z} \prod_{j=0}^{J} q_j(\mathbf{y})$$

which is equivalent to representing the natural parameters $\boldsymbol{\theta}$ of $q(\mathbf{y})$ as a sum of $J+1$ terms if we set

$$q_j(\mathbf{y}) = \exp\left\{\mathbf{T}(\mathbf{y})^T \boldsymbol{\theta}_j\right\}.$$

Obviously, given that we know the parameters of each *approximate local factor* $q_j(\mathbf{y})$ of $q(\mathbf{y})$, we can estimate the *cavity distribution* $q_{\neg j}(\mathbf{y}) = Z\frac{q(\mathbf{y})}{q_j(\mathbf{y})}$.

The core intuition behind EP is that, as an approximation to the problem of minimizing $D_{KL}\left[p(\mathbf{y})\middle\|q(\mathbf{y})\right]$, we can iteratively minimize the divergence between the *tilted distribution*

$$\widetilde{q}(\mathbf{y}) = \frac{1}{Z_j} f_j(\mathbf{y}) \prod_{i \neq j} q_i(\mathbf{y}) \tag{1.12}$$

and the approximate posterior $q(\mathbf{y} \mid \boldsymbol{\theta})$ by adapting $\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}^{\text{new}}$ globally using a moment matching procedure. This produces a newly defined approximate posterior with factors

$$q^{\text{new}}(\mathbf{y}) = \frac{1}{Z} q_j^{\text{new}}(\mathbf{y}) \prod_{i \neq j} q_i(\mathbf{y}) \tag{1.13}$$

where $\boldsymbol{\theta}_j$ is the only unknown, and has to be updated given $\boldsymbol{\theta}^{\text{new}}$.

It is interesting to consider the cavity distribution $q_{\neg j}(\mathbf{y})$ as a prior that carries information about other local factors, and constrains the update of the approximate posterior for the local likelihood $f_j(\mathbf{y})$. Let us emphasize that Z_j is the normalizer of the tilted distribution, whereas Z normalizes the approximate factorized posterior: the latter will give an approximation to the model evidence at optimality.

To see how EP works, we first introduce the following result: assuming that $q(\mathbf{y})$ belongs to the exponential family, the expression $D_{KL}\left[p(\mathbf{y})\middle||q(\mathbf{y})\right]$ simplifies as follows:

$$\begin{aligned} D_{KL}\left[p(\mathbf{y})\middle||q(\mathbf{y})\right] &= \mathbb{E}_{p(\mathbf{y})}\left[\log p(\mathbf{y}) - \log q(\mathbf{y})\right] \\ &= \mathbb{E}_{p(\mathbf{y})}\left[\log p(\mathbf{y}) - \left(\mathbf{T}(\mathbf{y})^T \boldsymbol{\theta} - B(\boldsymbol{\theta})\right)\right] \\ &= \mathbb{E}_{p(\mathbf{y})}\left[\log p(\mathbf{y}) - \mathbf{T}(\mathbf{y})^T \boldsymbol{\theta}\right] + B(\boldsymbol{\theta}) \end{aligned}$$

which has a null gradient wrt $\boldsymbol{\theta}$ when

$$\begin{aligned} \nabla_{\boldsymbol{\theta}}\left\{B(\boldsymbol{\theta})\right\} &= \mathbb{E}_{p(\mathbf{y})}\left[\mathbf{T}(\mathbf{y})\right] \\ \Leftrightarrow \mathbb{E}_{q(\mathbf{y})}\left[\mathbf{T}(\mathbf{y})\right] &= \mathbb{E}_{p(\mathbf{y})}\left[\mathbf{T}(\mathbf{y})\right] \text{ (see Equation (1.7))} \end{aligned}$$

In other words, the configuration of $q(\mathbf{y})$ that minimizes the KL divergence between p and q is the one that matches the moments of the two distributions. For example, for a Gaussian approximate posterior shape, we would find the mean and variance of $\mathbf{T}(\mathbf{y})$ under $p(\mathbf{y})$ and set the mean and variance of $\mathbf{T}(\mathbf{y})$ under $q(\mathbf{y})$ to be equal to those estimated from $p(\mathbf{y})$.

Using this, we can now minimize

$$D_{KL}\left[\overbrace{\widetilde{q}(\mathbf{y})}^{(1.12)}\,||\,\overbrace{q^{\text{new}}(\mathbf{y})}^{(1.13)}\right] \tag{1.14}$$

by matching the moments of the two distributions. This defines the update of $q^{\text{new}}(\mathbf{y})$. The next step is to retrieve from there the new value of

$$q_j^{\text{new}}(\mathbf{y}) = Z_j \frac{q^{\text{new}}(\mathbf{y})}{q_{\neg j}(\mathbf{y})}$$

where $Z_j = \int q_j^{\text{new}}(\mathbf{y}) q_{\neg j}(\mathbf{y}) d\mathbf{y}$ is a normalizing constant (zeroth moment) of $q^{\text{new}}(\mathbf{y})$ which we impose to match $\int f_j(\mathbf{y}) q_{\neg j}(\mathbf{y}) d\mathbf{y}$.

Algorithm 1 summarizes the general procedure of EP. Initialization of this algorithm usually consists in setting all approximate local factors to an improper uniform distribution ($\Sigma = vI$ with $v \to \infty$).

Algorithm 1: Expectation Propagation

Data: Data $\mathbf{x}$

Result: Approximate posterior $q(\mathbf{y})$ and related approximate model evidence $p(\mathbf{x}) \approx \int \prod_{j=0}^{J} q_j(\mathbf{y}) d\mathbf{y}$

initialization;

repeat

  Select a component $q_j(\mathbf{y})$ to refine;

  Compute the cavity distribution $q_{\neg j}(\mathbf{y}) \propto \frac{q(\mathbf{y})}{q_j(\mathbf{y})}$;

  Minimize Equation (1.14) by matching moments of $\widetilde{q}(\mathbf{y}) \propto f_j(\mathbf{y}) q_{\neg j}(\mathbf{y})$ and $q^{\text{new}}(\mathbf{y})$ and get normalizing constant Z_j;

  Refine factor $q_j(\mathbf{y})$;

until *Some convergence criterion is reached*;

EP enjoys several advantages: first, each update of the global posterior over $\mathbf{y}$ is carried out based on a single observation j and not on the full likelihood over the J data-points. Importantly though, this update is constrained by other local factors: updates are achieved locally but not independently. This important feature makes the algorithm suitable for big problems, such as GP [54]. For instance, GP-based logistic regression can be shown to be a convex problem, which makes the use of EP compelling as there is no risk of overestimating the mass of parameter subspaces (see downsides of EP here below). Next, it can be parallelized, and each worker of the optimization process will require only a limited memory allocation [53].

However, the EP process outlined above is not guaranteed to converge: as opposed to Conjugate Variational Message Passing that we will see short after, in EP we do not minimize the target $D_{KL}\left[p\middle\|q\right]$ at each iteration. Indeed, convergence can be an issue, especially with poor parameter initialization [51]. Another disadvantage of EP is that, if the posterior distribution is multimodal, a Gaussian approximate posterior will tend to cover the whole posterior probability, possibly assigning a high density to parameter values that lie in the cavity separating a few posterior modes.

In the EP algorithm depicted above, each datapoint is assigned a local factor $f_j(\mathbf{y})$ and a corresponding local approximate site distribution $q_j(\mathbf{y})$. This might prove to be inefficient for large datasets. One way to overcome this problem is to impose the same form to each factor $\{\, q_j(\mathbf{y}) = q_{\text{gen}}(\mathbf{y}),\ j = 1, 2, \ldots, J \,\}$ [51] (in other words, we impose that $\boldsymbol{\theta}_j = \boldsymbol{\theta}\,/J$). Also, one can assign multiple points to a single approximate local factor [53].

Also, although we have assumed that the update of the approximate posterior through the computation of the tilted distribution has a closed form formula (see for instance [52, 54]), several models do not enjoy this property [53]. Several solutions exist in this case: the first one is to assume that the mode of the tilted distribution coincides with its

mean, and that the covariance matrix can be estimated as the negative inverse hessian at the mode. This is known as the Laplace approximation [55]. Laplace approximation can be viewed as an end-product of an iterative Newton method-guided optimization of the approximate posterior [51]. [53] suggests to use Laplace approximation as a first step before refining the approximate posterior update using simulation (typically IS such as in [56]). Otherwise, simulation can be used as a general method to match the moments of the tilted and approximate distribution.

1.1.2.2 Variational Inference

In the previous section, we have presented a popular method for the approximation of a posterior distribution p using an approximation q which relied on minimizing the KL divergence between these two distributions. We will now focus on the opposite problem, namely of minimizing the KL divergence between q and p, which we will group under the general name Variational Inference (VI, Figure 1.3). Unlike EP, VI can usually be framed as a standard optimization problem, where a loss function has to be minimized or maximized, providing most of the time a measure and a guarantee of convergence.

Although current VI techniques do not involve explicit calculus of variation, VI still performs the same optimization problem of the statistical physics field [57] where it finds its roots: namely minimizing a functional (i.e. a function of function) wrt a function. Specifically, in our case, we will try to minimize the KL divergence wrt q.

Because it is the core optimization algorithm of this thesis, we will expand the discussion around VI more than we did for expectation propagation. The discussion will be organized following a somehow historically-motivated discourse. First, we will present the most classical (but still widely used) mean-field Variational Message Passing algorithm for conjugate and non-conjugate models, and show how it can be approximated using Laplace approximation, or generalized to large problems using Stochastic Variational Inference.

This will lead us to present two VI techniques that rely on unbiased estimates of the (gradient of the) lower bound to the log-model evidence: Stochastic Gradient Variational Bayes methods and Black-Box Variational Inference. With these techniques, both the data and the parameters are sampled from the dataset and from the variational distribution, respectively, alleviating the need to compute an exact value for the lower bound at training and test time.

In its generic form, VI aims at minimizing the KL divergence between the approximate posterior and the true, unknown posterior distribution. This divergence can be expressed

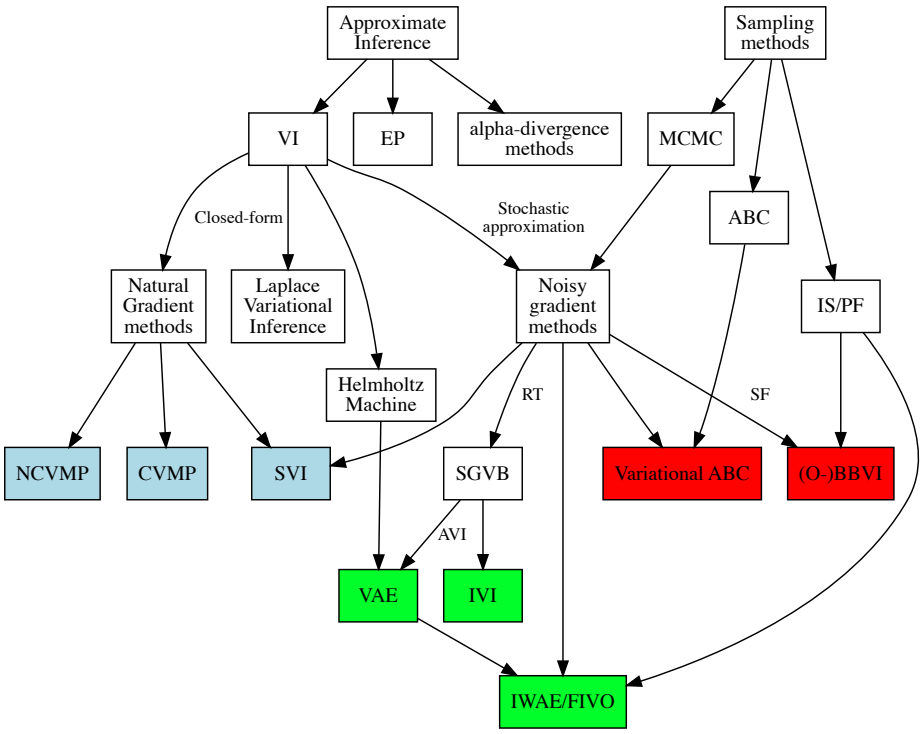


FIGURE 1.3: VI optimization algorithms detailed in this section. Light-blue boxes indicate algorithms for which fast inference can be achieved, at the cost of generalization capacity. Green boxes represent algorithms that have a fast convergence rate (low-variance noisy gradient) but that still require the model to have specific characteristics (reparameterizable gradient etc.) Red boxes represent black-box methods with a higher variance of the gradient estimate.

in several interesting ways:

$$
\begin{aligned}
D_{KL}\left[q(\mathbf{y} \mid \boldsymbol{\theta}) \middle\| p(\mathbf{y} \mid \mathbf{x})\right] &= \mathbb{E}_{q(\mathbf{y}|\boldsymbol{\theta})}\left[\log q(\mathbf{y} \mid \boldsymbol{\theta}) - \log p(\mathbf{y} \mid \mathbf{x})\right] \\
&= \mathbb{E}_{q(\mathbf{y}|\boldsymbol{\theta})}\left[\log q(\mathbf{y} \mid \boldsymbol{\theta}) - \log p(\mathbf{x}, \mathbf{y}) + \log p(\mathbf{x})\right] \\
&\quad \text{where } p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} \\
&= -\underbrace{\mathbb{E}_{q(\mathbf{y}|\boldsymbol{\theta})}\left[\log p(\mathbf{x}, \mathbf{y}) - \log q(\mathbf{y} \mid \boldsymbol{\theta})\right]}_{\mathcal{L}(\boldsymbol{\theta})} + \log p(\mathbf{x}).
\end{aligned}
\tag{1.15}
$$

Rearranging the above expression, and considering that the KL divergence is non-negative[7], we see that $\mathcal{L}(\boldsymbol{\theta})$[8] is a lower bound to the log-model evidence (log-Evidence Lower BOund, ELBO, or negative free energy), and matches exactly the log-model evidence when $q(\mathbf{y} \mid \boldsymbol{\theta}) = p(\mathbf{y} \mid \mathbf{x})$. Note that the same equivalence can be found by using the Jensen's inequality.

Box 1.1.2| The ELBO as an Occam's razor

A crucial problem when building a model is to find the best explanatory model $p(\mathbf{x} \mid \mathbf{y})$ that achieves the minimum complexity at the same time. We show how the ELBO achieves this: let us first reformulate it in a minimum description length [58] perspective:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{y})}\left[\log p(\mathbf{x}, \mathbf{y})\right] - \mathbb{E}_{q(\mathbf{y})}\left[\log q(\mathbf{y})\right] \\ &= \underbrace{\mathbb{E}_{q(\mathbf{y})}\left[\log p(\mathbf{x} \mid \mathbf{y})\right]}_{\text{Accuracy } \propto J} - \underbrace{D_{KL}\left[q(\mathbf{y}) \middle|\middle| p(\mathbf{y})\right]}_{\text{Complexity } \propto 1}.\end{aligned}$$

We see that the ELBO has two terms: the first one is the accuracy of the model, which scales linearly with the amount of data J. The second term is the discrepancy between the prior and the approximation, which does not depend on the data. We can interpret that as the complexity: a model is more complex if it differs from a given prior. For instance, in a linear regression or in an artificial neural network, it is legitimate to think that weights should not be very different of zero: consequently, it is common to use a Gaussian prior over the parameters (L2 regularization or ridge regression) or a Laplace distribution (L1 regularization or Lasso).

Let us assume that we have fitted $q(\mathbf{y})$ to $p(\mathbf{y} \mid \mathbf{x})$ and that the second argument is very low. Then, we will also need a very high first term (accuracy) to compensate for the mismatch between prior and posterior. In other words, a very complex model with a small dataset will be penalized by a posterior distant from the prior and a low accuracy. Adding more data may increase the accuracy, which will compensate for the complexity penalty.

[7] We provide a sketch of the proof of this assertion in the case of parametric distributions of continuous random variables: if $D_{KL}\left[q \middle|\middle| p\right] \geq 0$, it implies that $\mathbb{E}_q\left[\log q\right] \geq \mathbb{E}_q\left[\log p\right]$. This means that, of all probability distributions $\mathcal{P}$ with the same domain as q and p, q is the one with the maximum expected log value. Let us consider what function $p_\theta \in \mathcal{P}$ would maximize $\mathbb{E}_q\left[\log p_\theta\right]$:

$$\text{find } p_\theta^* \text{ s.t. } p_\theta^* = \arg\max_{p_\theta} \mathbb{E}_q\left[\log p_\theta\right].$$

We know that the expected value of the score function of a probability distribution is equal to 0: this implies that if $p_\theta^* = q$, then $\nabla_\theta \left\{\mathbb{E}_q\left[\log p^*\right]\right\} = 0$ and, as this expectation has generally no finite minimum, we can see that q is a maximum.

[8] The ELBO is sometimes written $\mathcal{L}(q(\mathbf{y} \mid \boldsymbol{\theta}))$ to highlight the fact that the loss is optimized wrt a distribution q, in contrast with ML methods.

We see that VI provides us with an objective function $\mathcal{L}(\boldsymbol{\theta})$ that we can maximize wrt $\boldsymbol{\theta}$ in order to match q and p as much as possible, given the restrictions we impose on q. The next sections will detail common approaches to solve the problem of maximizing the ELBO.

Box 1.1.3| Power EP and α-Divergence methods

VI and EP are based on two special cases of divergence measure between q and p. More generally, $D_{KL}\left[q||p\right]$ and $D_{KL}\left[p||q\right]$ find a common formulation under the α-divergence (or Rényi divergence) [59] which reads

$$D_{\alpha}\left[p(\mathbf{y})||q(\mathbf{y})\right] = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p(\mathbf{y})^{\alpha} q(\mathbf{y})^{1-\alpha} d\mathbf{y}\right).$$

When $\alpha \to 1$, we have $D_1\left[p(\mathbf{y})||q(\mathbf{y})\right] = D_{KL}\left[p||q\right]$ and we are in the EP case, on the contrary, when $\alpha \to 0$, $D_0\left[p(\mathbf{y})||q(\mathbf{y})\right] = D_{KL}\left[q||p\right]$ and the VI loss function can be derived. Finally, the α-divergence becomes a distance (specifically, the Hellinger distance) for $\alpha = 1/2$.

Power EP [60] is an algorithm that generalizes the vanilla message passing EP to cases where exponentiating the density of the model can improve the tractability of the message passing algorithm. For instance, a density $p(\mathbf{y}) = \frac{1}{\pi(\mathbf{y}^1+1)^2}$ is hard to use with EP, but exponentiating $p(\mathbf{y})$ by -1 makes the updates tractable. In this sense, Power EP is a general class of approximate inference algorithms from which EP is a special case. Minka [61] notes that the continuum of divergence measures that α-divergence provides allows us to divide the space of these divergences in an *inclusive* ($\alpha \geq 1$) subspace, where the algorithm will try to find a q that covers most of the posterior, or *exclusive* subspace ($\alpha < 1$) where the algorithm will focus on the main mode (or local modes if these modes have the same mass).

In its original flavor, Power EP is still a message-passing algorithm. Hernandez-Lobato and colleagues [62] propose a black-box α-divergence (BB-α) algorithm that generalizes Stochastic EP [63] and averaged EP [51] to the class of α-divergence minimization approximate inference algorithms. Crucially, this model does not rely on a message passing algorithm but approximates power EP with a simplified *objective* that depends on a pre-defined value of α. As a consequence, BB-α is guaranteed to converge for large datasets.

In the following paragraphs of this section, we will detail some of the algorithms proposed to optimize the ELBO: we will start with approaches that enjoy the property of having a fast convergence rate thanks to their use of natural gradient methods, but that require strong assumptions about the model and about the approximate posterior shapes that

are used. As the discussion goes on, we will progressively move to more general (and conversely slower) methods of optimization, that can endow more flexible approximate posteriors. This discussion is aimed at giving a broad overview of the current state-of-the-art, and to provide the necessary foundations for the numerous methods used or proposed in Chapters 2 to 4.

1.1.2.2.1 Mean-field Variational Inference The core idea of VI in most of its instances consists in making the strong but highly advantageous assumption that the approximate posterior probability we are interested in optimizing should factorize in some way [64, 65], namely:

$$q(\mathbf{y}) \triangleq \prod_{i=1}^{N} q(\mathbf{y}_i)$$

which is equivalent to say that the posterior distribution $\mathbf{y} = \{ \mathbf{y}_i \}_{i=1}^{N}$ are independent, or that their mutual information is equal to zero[9].

This assumption has also its roots in the statistical physics theory, and more specifically thermodynamics [66, 67]. In the Ising model, for instance, particles are typically assumed to be arranged on a lattice, which we can see as an instance of a Markov Random Field, or undirected graphical model. The Free Energy of such model is hardly described in systems with many (typically more than two) dimensions, and some approximations need to be made. The mean-field theory (or mean-field assumption) is aimed at describing such complex model through a simpler one, in which particles in the system interact by pairs: this effectively reduces the complexity of the problem, as the effect of distant particles in the lattice are concentrated on the neighbouring particles. Specifically, for the Ising model, the product of the fluctuation of spins around the mean of two neighbours is discarded from the computation of the Hamiltonian: for two particles with spin $s_{i,j} = m_{i,j} + \delta s_{i,j}$, the product of the spins is given by

$$\begin{aligned} s_i s_j &= m_i m_j + \delta s_i \, m_j + m_i \, \delta s_j + \delta s_i \, \delta s_j \\ &\overset{\text{MF}}{\approx} m_i m_j + \delta s_i \, m_j + m_i \, \delta s_j. \end{aligned}$$

In a graphical model, this is equivalent to say that the two adjacent nodes have a zero covariance. Interestingly, whereas the initial approaches to solve this problem have mainly relied on model-based approaches, a Mean-Field multi-agent RL methods has recently been proposed to solve the Ising model in a model-free manner [68].

[9] The notation should not confuse the reader: each of the q distributions in mean-field VI can have a different form, whereas each of the q_i local factors of the global approximate posterior in EP was an unnormalized distribution of the same family as the global one.

Framed in this way, we see that the mean-field assumption essentially amounts to a *message passing* framework. Using the mean-field assumption, we can focus on how a single node in the graphical model interacts with the adjacent nodes without considering their covariance by sending messages to them.

This greatly reduces the complexity of the VI framework: if we build a model (typically a directed acyclic graph) with nodes that are assumed to have an independent posterior distribution, we can estimate this posterior by sending messages from one node to the other by considering each single node's Markov blanket. In what follows, we will detail models where this deterministic update framework can be implemented, present its restrictions and the methods that can be used to circumvent them.

1.1.2.2.2 Conjugate Variational Message Passing Let us first focus on the utterly simple – although very general – case of conjugate models [69]. For sparsity of the notation, we will adopt the convention that a set of parameters $\mathbf{y}$ is mapped onto its natural space with a function $\psi_{\mathbf{y}}(\mathbf{y})$.

We will somehow give a rather informal, example guided explanation of the main principles that underlie Conjugate Variational Message Passing (CVMP) here, and we refer to the aforementioned paper and to [69, 22] for a theoretically detailed discussion of this method.

We show that the local q factor that maximizes the lower bound has an utterly simple general formula. Let us adopt the same general factorized q distribution from above $q(\mathbf{y}) = \prod_{i=1}^{N} q_i$, and denote by $q_{\neg i}$ the product of the factors that do not depend on $\mathbf{y}_i$. Let also $\log f(\mathbf{y})$ denote the log-joint probability given some (hidden in this formula) observed variables $\mathbf{x}$. We note that the lower bound can be written as:

$$\begin{aligned}\mathcal{L}\left(q(\mathbf{y})\right) &= \mathbb{E}_{q_1}\left[\underbrace{\mathbb{E}_{q_{\neg 1}}\left[\log f(\mathbf{y})\right]}_{\triangleq q_1^*}\right] + \mathbb{H}\left[q_1\right] + \mathbb{H}\left[q_{\neg 1}\right] \\ &= -D_{KL}\left[q_1 \middle|\middle| q_1^*\right] + \text{cst}\end{aligned}$$

where $\mathbb{H}[q]$ stands for the entropy of the distribution q. We see that the ELBO is maximized wrt q_1 when $q_1 = q_1^*$ (hence the notation) which can be achieved by setting

$$q_1 \propto \exp\left\{\mathbb{E}_{q_{\neg 1}}\left[\log f(\mathbf{y})\right]\right\}. \tag{1.16}$$

Equation (1.16) plays a central role in VI: it states that, under the mean-field assumption, if the expected log-joint probability wrt all but one (e.g. $\mathbf{y}_1$) latent variables has a

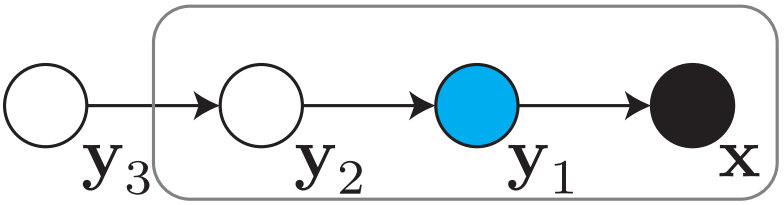


FIGURE 1.4: Directed Acyclic Graph of a simple hierarchical model. The blue node represent the variable about which inference is made. The gray box represent the Markov Blanket of this node.

closed-form expression, then this expression gives us the log of the approximate posterior probability of $\mathbf{y}_1$ given the current posterior belief of the other latent variables.

In conjugate models, one can easily see that the conjugate prior $p(\mathbf{y}_1)$ of some distribution $p(\mathbf{x} \mid \mathbf{y}_1)$ gives us the form of the optimal approximate posterior for $\mathbf{y}_1$ under the mean-field assumption. For instance, if we define the hierarchical model (Figure 1.4)

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x} \mid \mathbf{y}_1)\, p(\mathbf{y}_1 \mid \mathbf{y}_2)\, p(\mathbf{y}_2 \mid \mathbf{y}_3)\, p(\mathbf{y}_3)$$

where each prior distribution is conjugate to the level below, then the optimal approximate over some latent variable (e.g. $\mathbf{y}_1$) has the form of its prior:

$$\begin{aligned} q^*(\mathbf{y}_1) &\propto \exp\Big\{ \mathbb{E}_{q(\mathbf{y}_{\neg 1})}\big[\psi_{\mathbf{y}_1}(\mathbf{y}_1)^T \mathbf{T}(\mathbf{x}) - A_{\mathbf{y}_1}(\mathbf{y}_1) + \psi_{\mathbf{y}_1}(\mathbf{y}_1)^T \psi_{\mathbf{y}_2}(\mathbf{y}_2) - A_{\mathbf{y}_2}(\mathbf{y}_2) + \\ &\qquad \psi_{\mathbf{y}_3}(\mathbf{y}_3)^T \psi_{\mathbf{y}_2}(\mathbf{y}_2) - A_{\mathbf{y}_3}(\mathbf{y}_3) + \psi_{\mathbf{y}_3}(\mathbf{y}_3)^T \zeta - A_\zeta(\zeta)\big] \Big\} \\ &\propto \exp\left\{ \psi_{\mathbf{y}_1}(\mathbf{y}_1)^T \left(\underbrace{\mathbf{T}(\mathbf{x}) + \mathbb{E}_{q(\mathbf{y}_2)}\left[\psi_{\mathbf{y}_2}(\mathbf{y}_2)\right]}_{\triangleq \boldsymbol{\theta}_{\mathbf{y}_1}} \right) - A_{\mathbf{y}_1}(\mathbf{y}_1) \right\} \end{aligned} \tag{1.17}$$

where $A_{\mathbf{y},\zeta}$ are the log-partition functions of each of the prior distributions and ζ is the hyperprior, either defined beforehand or optimized in an empirical Bayes manner (e.g. Variational Expectation Maximization). Let us stress the importance of this result: in a conjugate model where we adequately use the mean-field assumption, the form of the prior distributions dictates the choice of the approximate posterior such that no other choice would have improved the ELBO at optimality. Of course, this does not assume anything about the accuracy of the mean-field assumption! However, this simple observation can greatly simplify inference for models with many levels and/or lots of observations. We also note that the update of the variational parameters $\boldsymbol{\theta}_1$ of

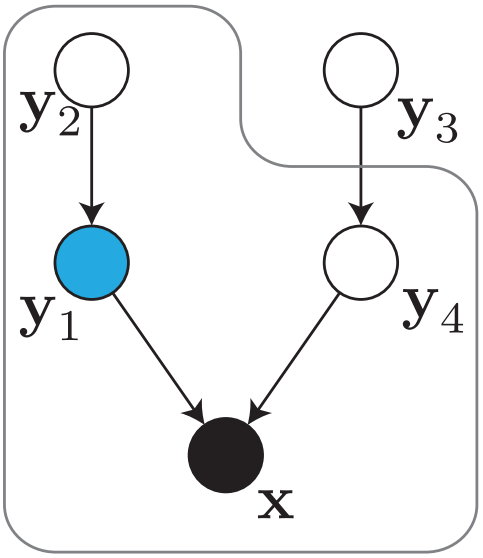


FIGURE 1.5: Directed Acyclic Graph of a two-branch hierarchical model.

this approximate posterior are updated with pieces of information (or better, *messages*) received from adjacent nodes.

The example in Equation (1.17) is probably too simplistic to be of any help when dealing with large and/or complex hierarchical models, and one has to consider directed graphical models where a single variable has many parents. A more general model is outlined in Figure 1.5. In this model, it appears that the terms in the expected log-joint probability that involves $\mathbf{y}_1$ include terms that depend upon its prior $\mathbf{y}_2$, its child $\mathbf{x}$, and its coparents $\mathbf{y}_4$. Therefore, the optimal approximate posterior is given by

$$q^*(\mathbf{y}_1) \propto \exp\left\{\mathbb{E}_{q(\mathbf{y}_{\neg 1})}\left[\psi_{1,4}(\mathbf{y}_{1,4})\mathbf{T}(\mathbf{x}) - A_{1,4}(\mathbf{y}_{1,4}) + \psi_1(\mathbf{y}_1)\psi_2(\mathbf{y}_2) - A_2(\mathbf{y}_2)\right]\right\}$$

Conjugacy of the model ensures that the above expression can be rewritten as

$$\begin{aligned} q^*(\mathbf{y}_1) &\propto \exp\left\{\mathbb{E}_{q(\mathbf{y})}\left[\psi_{\mathbf{x},2}(\mathbf{x},\mathbf{y}_2)^T\psi_1(\mathbf{y}) - A_1(\mathbf{y}_1) + \psi_1(\mathbf{y}_1)^T\psi_2(\mathbf{y}_2)\right]\right\} \\ &\propto \exp\left\{\mathbb{E}_{q(\mathbf{y}_{\neg 1})}\left[\psi_1(\mathbf{y}_1)^T\left(\psi_{\mathbf{x},4}(x,\mathbf{y}_4) + \psi_2(\mathbf{y}_2)\right)\right]\right\} \end{aligned} \tag{1.18}$$

which, again, has the same form as the prior of $\mathbf{y}_1$, up to a normalizing constant. Equation (1.18) shows that the update of q_1 receives messages from each of the nodes of the Markov Blanket of $\mathbf{y}_1$: $m_{\{\mathbf{x},\mathbf{y}_4\}\to\mathbf{y}_1} = \psi_{\mathbf{x},4}(\mathbf{x},\mathbf{y}_4)$ and $m_{\mathbf{y}_2\to\mathbf{y}_1} = \psi_2(\mathbf{y}_2)$.

1.1.2.2.3 Non-Conjugate Variational Message Passing Updating approximate posteriors of models that are not conjugate is a frequent problem. We will distinguish cases where the local approximate posteriors are from the exponential family and where the log-joint probability has a closed form and is once differentiable, from cases where at least one of these conditions is not met. Non-Conjugate Variational Message

Passing (NCVMP) [70] provides an update scheme for models of the first category. Interestingly, NCVMP shares many similarities with the EP framework outlined above, and we will make these explicit shortly.

Let $p(\mathbf{x}, \mathbf{y})$ be the log-joint probability of some model, and $q(\mathbf{y}_1)$ be the approximate posterior of $\mathbf{y}_1$ under the mean-field assumption. NCVMP states that the update of the natural parameters $\boldsymbol{\theta}_1$ of $q(\mathbf{y}_1 \mid \boldsymbol{\theta}_1)$ takes the form

$$\boldsymbol{\theta}_1^* = C(\boldsymbol{\theta}_1)^{-1} \frac{\partial \mathbb{E}_{q(\mathbf{y})}\left[p(\mathbf{x}, \mathbf{y})\right]}{\partial \boldsymbol{\theta}_1} \tag{1.19}$$

where $C(\boldsymbol{\theta}_1)$ is the covariance of $\mathbf{T}(\mathbf{y})$ given $q(\mathbf{y}_1 \mid \boldsymbol{\theta}_1)$ (which is identical to the second derivative of the log-partition $\frac{\partial^2 A(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1^2} = \frac{\partial \mathbb{E}_{q_1}[\mathbf{T}(\mathbf{y}_1)]}{\partial \boldsymbol{\theta}_1}$). Of course, this update depends upon and only upon the nodes that belong to the Markov Blanket of $\mathbf{y}_1$, which means that the above expression can also be expressed as a sum of the messages sent by the neighbouring nodes of $\mathbf{y}_1$.

It can be seen from Equation (1.19) that, if the model is conjugate, this equation simplifies to CVMP: indeed, in this equation, $C(\boldsymbol{\theta}_1)^{-1}$ would simplify with $\frac{\partial \mathbb{E}_{q(\mathbf{y})}[p(\mathbf{x},\mathbf{y})]}{\partial \boldsymbol{\theta}_1}$, as one of the terms of this expression can be formulated as $\frac{\partial \mathbb{E}_{q(\mathbf{y}_1)}[\mathbf{T}(\mathbf{y}_1)]}{\partial \boldsymbol{\theta}_1} = C(\boldsymbol{\theta}_1)$.

However, the only guarantee we can have about NCVMP is that if a stationary point is reached, then this point is guaranteed to be a maximum of the ELBO wrt $\boldsymbol{\theta}_1$. Consequently, there is neither a guarantee that these single updates will decrease the KL divergence nor that the algorithm will converge. These problems are not to be ignored in practical implementations. Many convergence issues can be prevented by damping successive updates in the following manner:

$$\boldsymbol{\theta}_1 = \boldsymbol{\theta}_1^{*a} \boldsymbol{\theta}_1^{1-a}$$

where $a \in]0, 1)$ is a damping factor.

As mentioned above, NCVMP shares several similarities with EP: both use small messages from adjacent nodes to update the parameters of a specific node. None of them is guaranteed to converge. Also, both rely on the assumption that the approximate posterior comes from the exponential family (which, for EP, is most of the time a Gaussian distribution).

1.1.2.2.4 Laplace Approximation We now tackle the problem of updating the variational parameters for models where one assumes that (1) the approximate posterior for a given variable (e.g. $\mathbf{y}$) is Gaussian, (2) the expectation of log-joint probability

wrt to all latent variables but $\mathbf{y}$ can be maximized wrt $\mathbf{y}$, and (3) this expected log-joint probability function is twice differentiable wrt $\mathbf{y}$. In general, Laplace Variational Inference [71, 72, 73] can be applied to a larger family of models than NCVMP.

Let us first briefly present the Laplace approximation [22, 74]. Let us consider a posterior probability $p(\mathbf{y} \mid \mathbf{x})$. We can take the second order Taylor series of the logarithm of this distribution around the mode (assuming that it exists), which gives:

$$\begin{aligned}\log p(\mathbf{y} \mid \mathbf{x}) \propto \log p(\mathbf{x}, \mathbf{y}^*) + \\ (\mathbf{y}^* - \mathbf{y}) \nabla_{\mathbf{y}^*} \{\log p(\mathbf{x}, \mathbf{y}^*)\} + \\ \frac{1}{2} (\mathbf{y}^* - \mathbf{y})^T \nabla^2_{\mathbf{y}^*} \{\log p(\mathbf{x}, \mathbf{y}^*)\} (\mathbf{y}^* - \mathbf{y}).\end{aligned}$$

By definition, $\nabla_{\mathbf{y}^*} \{\log p(\mathbf{x}, \mathbf{y}^*)\} = 0$ and the above expression takes the form of a Gaussian distribution with parameters

$$p(\mathbf{y} \mid \mathbf{x}) \approx \mathcal{N}\left(\mathbf{y}^*, -\nabla^2_{\mathbf{y}^*} \{\log p(\mathbf{x} \mid \mathbf{y}^*)\}^{-1}\right).$$

A well known property of likelihood functions that follows from there is that, if some some rules are respected (continuity and non-boundary location of the maximum parameter value) the posterior probability of such models tends to look like a Gaussian distribution around the mode as the dataset grows. The Laplace refers to the use of the same technique to solve integrals of the form

$$\int \exp\{M f(\mathbf{y})\} \, d\mathbf{y}$$

for large M. Note that, under these circumstances, and also under relatively mild assumptions [75], the prior distribution of $\mathbf{y}$ can be also discarded. This justifies the quadratic approximation of the log-likelihood function around the mode in these cases. Applied to the Variational update problem we face, we can also use the same idea to derive useful update rules for a Gaussian approximation to a factorized posterior.

Let us take back the logarithm of the expression in Equation (1.16):

$$\begin{aligned}\log q_1^* &= \mathbb{E}_{q_{\neg 1}} [\log p(\mathbf{x}, \mathbf{y})] + cst \\ &\approx \mathbb{E}_{q_{\neg 1}} \Big[\log p(\mathbf{x}, \mathbf{y}_1^*, \mathbf{y}_{\neg 1}) + (\mathbf{y}_1^* - \mathbf{y}_1) \nabla_{\mathbf{y}_1^*} \{\log p(\mathbf{x}, \mathbf{y}_1^*, \mathbf{y}_{\neg 1})\} \\ &\quad + \frac{1}{2} (\mathbf{y}_1^* - \mathbf{y}_1)^T \nabla^2_{\mathbf{y}^*} \{\log p(\mathbf{x}, \mathbf{y}_1^*, \mathbf{y}_{\neg 1})\} (\mathbf{y}_1^* - \mathbf{y}_1) \Big] + cst\end{aligned}$$

for some $\mathbf{y}_1$ close to $\mathbf{y}_1^*$. This expression, again, is quadratic in $\mathbf{y}_1$ and simplifies to a Gaussian distribution with parameters

$$q^*(\mathbf{y}_1) \approx \mathcal{N}\left(\mathbf{y}_1^*, -\nabla^2_{\mathbf{y}_1^*}\{\log p(\mathbf{x}, \mathbf{y}_1^*, \mathbf{y}_{\neg 1})\}^{-1}\right).$$

where the value of $\log p(\mathbf{x}, \mathbf{y}_1^*, \mathbf{y}_{\neg 1})$ plays the role of a log-base function. In models where one tries to achieve closed-form updates of the variational parameters, the use of the Laplace Approximation has to be considered in cases where the ELBO does not enjoy a closed-form formula, but can be optimized and differentiated twice wrt $\mathbf{y}_1$.

The Laplace approximation will *not* work, for instance, if the ELBO has an infinite maximum in $\mathbf{y}_1^*$. For instance, this situation might arise in the Full Drift-Diffusion-Model, where the local approximate posteriors (at the trial level) can have multiple maximum (and, actually, infinite) densities. In general, the Laplace approximation works well when the ratio of data per number of dimension is high. In these cases, other approaches have to be considered.

1.1.2.2.5 Stochastic Variational Inference Even though Stochastic Variational Inference (SVI) [76] in its vanilla form still mainly relies upon conjugate models, it will be useful for us to flow from closed-form update methods to stochastic approximations to the VI problem.

Hoffman and colleagues [76] note that the updates achieved by CVMP (and, actually, also by NCVMP) can be derived by computing the natural gradient [77] of the ELBO wrt the variational parameters. In short, a gradient is said to be natural if it is invariant to bijective transformations of the parameterization: in other words, natural gradients ensure that the arbitrary choice of the parameterization does not impact the convergence of the optimization algorithm. Indeed, many formulations of VB updates somehow recur to this natural (Reimannian) gradient [78, 79, 80, 81, 82]. For a certain node $\mathbf{y}$, each of the steps of the gradient descent can be framed as a sum of *messages* received from the nodes belonging to the Markov blanket of $\mathbf{y}$. For large models, this is an interesting property, as it ensures that the messages of each local node to a global parent can be progressively retrieved while iterating through the dataset (instead of moving once along the sum of *all* the children). This approach contrasts mainly with what we have presented before as it does not require us to iterate through the whole dataset before achieving a single update of the global parameters. Instead, we only gather small pieces of information from each local node to update the global node.

Here, we detail Stochastic Gradient Descent, as it is the general family of algorithm upon which SVI relies. Let us first note that, if $\mathbf{x} = \{ x_i \}_{i=1}^N$ are a local nodes where

$p(x_i \mid \mathbf{z}_i)p(\mathbf{z}_i \mid \mathbf{y})$ are the local factors, and if $\mathbf{y}$ is the global node, then

$$\begin{aligned}\widehat{\nabla}_{\boldsymbol{\theta}_\mathbf{y}}\left(\mathcal{L}(\boldsymbol{\theta}_\mathbf{z}, \boldsymbol{\theta}_\mathbf{y})\right) &\approx \frac{N}{M}\sum_{i=1}^{M}\widehat{\nabla}_{\boldsymbol{\theta}_\mathbf{y}}\left\{\mathbb{E}_{q(\mathbf{z}_i,\mathbf{y})}\left[\log p(x_i \mid \mathbf{z}_i) + \log p(\mathbf{z}_i \mid \mathbf{y})\right]\right\} + \widehat{\nabla}_{\boldsymbol{\theta}_\mathbf{y}}\left\{\mathbb{E}_{q(\mathbf{y})}\left[\log p(\mathbf{y})\right]\right\} \\ &\triangleq \widetilde{\nabla}_{\boldsymbol{\theta}_\mathbf{y}}\left(\mathcal{L}(\boldsymbol{\theta}_\mathbf{z}, \boldsymbol{\theta}_\mathbf{y})\right) \\ &\text{with } M < N\end{aligned}$$

gives an unbiased estimate of the natural gradient (denoted by $\widehat{\nabla}$) of the ELBO wrt $\boldsymbol{\theta}_\mathbf{y}$, which by the strong law of large numbers converges with probability one to the true natural gradient. In short, this expression ensures that a subsample of $M < N$ points is, on average, equal to the true natural gradient, and can therefore be used for optimization.

Let α_t be a learning rate that satisfies some asymptotic conditions (sometimes referred to as Robbins-Monro conditions [83]):

$$\begin{aligned}\lim t \to \infty \alpha_t &= 0 \\ \lim_{N\to\infty}\sum_{t=1}^{N}\alpha_t &= \infty \\ \lim_{N\to\infty}\sum_{t=1}^{N}\alpha_t^2 &< \infty.\end{aligned}$$

Then, it can be shown that under mild assumptions [84] the SGD update approach outlined in Algorithm 2 converges to the true maximum of the ELBO wrt $\{\boldsymbol{\theta}_\mathbf{z}, \boldsymbol{\theta}_\mathbf{y}\}$.

Algorithm 2: SVI. The local parameters are updated one at a time, whereas the global parameters are progressively updated given the previously $\mathcal{N}_i$ is the set of the neighbours of node i, and $m_{j\to i}$ is the message sent from node j to i.

initialization;
t=0;
repeat
    t++;
    Select a local node $i \sim \mathcal{U}(1, N)$;
    Update $\boldsymbol{\theta}_{\mathbf{z}i} = \sum_{j\in\mathcal{N}_i} m_{j\to i}$;
    Update $\boldsymbol{\theta}_\mathbf{y} = (1-\alpha_t)\,\boldsymbol{\theta}_\mathbf{y} + \alpha_t N * m_{\mathbf{z}_i\to\mathbf{y}}$;
until *Some convergence criterion is reached*;

VI scales well with large datasets if they meet the requirements of CVMP and NCVMP (e.g. tractability of the gradient of the ELBO). The vanilla version sketched here can clearly be greatly improved: the use of minibatches of data can lower the variance of the updates achieved at each time step for a computational cost still far below the cost of

the whole-dataset based approach. Learning rate adaptation [85], automated mini-batch selection [86] and, even more importantly, (semi-)amortized inference [87] can also boost the convergence rate of this algorithm.

1.1.2.2.6 (Doubly) Stochastic Variational Inference methods The previous method focused on specific models where the ELBO had a tractable form. Let us rephrase the maximum likelihood estimate of a model with a loss function (i.e. negative log-likelihood) $\ell(\cdot;\boldsymbol{\theta}) \triangleq \log p(\cdot \mid \boldsymbol{\theta})$ fed with a set of N datapoints $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$: $\ell(\mathbf{X};\boldsymbol{\theta}) = \sum_{i=1}^N \ell(\mathbf{x}_i;\boldsymbol{\theta})$. If we take the average of the $\ell(\mathbf{x}_i;\boldsymbol{\theta})$ instead of the sum, and if we make some usual assumptions about the domain of $\mathbf{X}$, namely $\mathcal{X}$ (typically *i.i.d.* of the samples), we see that this model loss-function takes the form of a Monte-Carlo estimate of the expected likelihood function given a random sample of datapoints:

$$\ell(\boldsymbol{\theta}) \triangleq \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{x}_i;\boldsymbol{\theta}).$$

This fairly simple expression should not surprise the reader, as its only aim is to attract the attention of the fact that, when collecting data and fitting a model to it, what we are trying to do is to evaluate the model likelihood *while the specific data sample has been marginalized out*. More precisely, if the data has been generated with a probability distribution $p_{\text{data}}(\mathbf{x})$, then the ML approach minimizes the cross entropy $H(p_{\text{data}}, p)$ between the true data distribution and the model distribution:

$$\begin{aligned} H(p_{\text{data}}, p) &= \mathbb{H}\left[p_{\text{data}}\right] + D_{KL}\left[p_{\text{data}} \middle|\middle| p\right] \\ &= \int_{\mathcal{X}} p_{\text{data}}(\mathbf{x})\ell(\mathbf{x};\boldsymbol{\theta})d\mathbf{x} \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{x}_i \mid \boldsymbol{\theta}) \quad \text{with} \quad \mathbf{x}_i \sim p_{\text{data}}(\mathbf{x}) \end{aligned} \tag{1.20}$$

Because the (unknown) data distribution is fixed, the ML of $\boldsymbol{\theta}$ minimizes the KL divergence between p_{data} and p as $N \to \infty$.

SGD optimization approaches further simplify this problem for large N when resources are limited by subdividing randomly the dataset in small, non-exclusive subsets that will dictate the training procedure of the model. An interesting feature of SGD is that, besides its computational efficiency, it can also somehow avoid local minima during optimization (although this is not guaranteed). Paragraph 1.1.2.2.6 gives an intuition of why it is the case.

Example 1.1.1| SGD and local minima

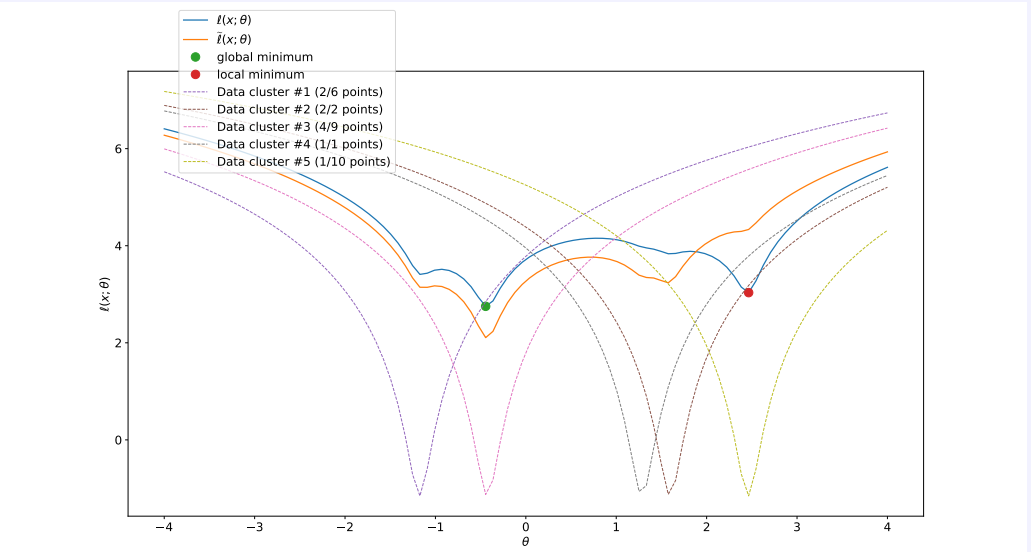


FIGURE 1.6: Local vs Global data-dependent minima of a loss function $\ell(\mathbf{x};\boldsymbol{\theta})$. We artificially generated a loss function that, when applied to clusters of data, produced a quite different loss curve for various values of the parameter $\boldsymbol{\theta}$ (in this case, the loss function was the log-pdf of a Cauchy distribution). We can see that the global loss function (blue) was attracted by each component (dashed lines) local minimum. Therefore, any optimization procedure starting around the local minimum marked with a red point would have wrongly stopped after reaching this point. SGD can alleviate this problem by subsampling datapoints unevenly from the various unknown clusters (orange curve). On average, the algorithm will ignore most of the local minima and focus mainly on the global minimum of the function (green point). The fraction of datapoints in parenthesis represents the ratio of points sampled in for the orange trace wrt the total number of points in this cluster. Each component log-pdf in displayed without having been scaled to its importance.

Consider now the case where $\ell(\mathbf{x}_i \mid \boldsymbol{\theta})$ is itself intractable. For instance, a common case that we will often encounter is when $\int p(\mathbf{y} \mid \boldsymbol{\theta})\ell(\mathbf{x}_i \mid \mathbf{y})d\mathbf{y}$ is either intractable, and/or heavy to compute/approximate. Then we can apply the exact same approach and use a Monte Carlo estimate of this quantity for the SGD optimization. More specifically, if the ELBO of some model is expressed as

$$\mathcal{L}(\boldsymbol{\theta}) = \int q(\mathbf{y} \mid \boldsymbol{\theta}) \left(\log p\left(\mathbf{x}, \mathbf{y}\right) - \log q(\mathbf{y} \mid \boldsymbol{\theta})\right) d\mathbf{y}$$

where $\mathbf{x} = \{\, \mathbf{x}_i \,\}_{i=1}^{N}$, then this expression has an unbiased estimator given by

$$\mathcal{L}(\boldsymbol{\theta}) \approx \frac{1}{MK} \sum_{i=1,k=1}^{M,K} \log p\left(\mathbf{x}_i, \widetilde{\mathbf{y}}_k\right) - \log q(\widetilde{\mathbf{y}}_k \mid \boldsymbol{\theta})$$

$$\text{where } \widetilde{\mathbf{y}}_k \sim q(\mathbf{y} \mid \boldsymbol{\theta})$$

for some large K and some $M \leq N$. An unbiased estimator of the ELBO can be computed in a similar manner. This is known as Stochastic Gradient Variational Bayes (SGVB).

1.1.2.2.7 The reparameterization trick In most instances, gradient computation in SGVB is implemented with the use of what is known as the *reparameterization trick* [88, 89], pathwise gradient [90, 91] or Infinitesimal Perturbation Analysis estimator [92, 93, 94]. For clarity, we will derive it here: let us assume that the approximate posterior we have chosen $q(\mathbf{y} \mid \boldsymbol{\theta})$ enjoys the nice property that, given an *i.i.d.* distributed random variable $\epsilon \sim p(\epsilon)$, we can sample from $q(\mathbf{y} \mid \boldsymbol{\theta})$ using a differentiable function $g(\epsilon; \phi)$ (for instance, ϵ can follow a Gaussian distribution $\mathcal{N}(0, 1)$ or a uniform distribution $\mathcal{U}(0, 1)$). Our aim is to approximate the gradient of the integral

$$\nabla_\phi \{\mathcal{L}(\boldsymbol{\theta})\} = \nabla_{\boldsymbol{\theta}} \left\{ \int q(\mathbf{y} \mid \boldsymbol{\theta}) \left(\log p(\mathbf{x}, \mathbf{y}) - \log q(\mathbf{y} \mid \boldsymbol{\theta})\right) d\mathbf{y} \right\}.$$

For this aim, we will restric ourselves to cases where the inverse cumulative density function (CDF) of q is tractable and differentiable. We define $g(\epsilon; \boldsymbol{\theta}) \triangleq Q^{-1}(\epsilon; \boldsymbol{\theta})$ where Q^{-1} is the inverse CDF of q. Using the rule of inverse function differentiation, we have

$$\begin{aligned} \nabla_\epsilon \{g(\epsilon; \boldsymbol{\theta})\} = \nabla_\epsilon \left\{Q^{-1}(\epsilon; \boldsymbol{\theta})\right\} &= \frac{1}{\nabla_{\mathbf{y}} \{Q(\mathbf{y}; \boldsymbol{\theta})\}} \qquad \text{with } \mathbf{y} = \left(Q^{-1}(\epsilon; \boldsymbol{\theta})\right) \\ &= \frac{1}{q(\mathbf{y} \mid \boldsymbol{\theta})} \qquad \text{given that } \nabla_{\mathbf{y}} \{Q(\mathbf{y}; \boldsymbol{\theta})\} = q(\mathbf{y} \mid \boldsymbol{\theta}). \end{aligned} \tag{1.21}$$

If we define f to be the inner part of the ELBO integral,

$$f(\mathbf{y}) \triangleq q(\mathbf{y} \mid \boldsymbol{\theta}) \left(\log p(\mathbf{x}, \mathbf{y}) - \log q(\mathbf{y} \mid \boldsymbol{\theta})\right)$$

and we substitute $\mathbf{y}$ by $g(\epsilon; \boldsymbol{\theta})$ in the above equation, then we have:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \left\{ \int f(\mathbf{y}) d\mathbf{y} \right\} &= \nabla_{\boldsymbol{\theta}} \left\{ \int f(g(\epsilon; \boldsymbol{\theta})) \nabla_\epsilon \{g(\epsilon; \boldsymbol{\theta})\} \, d\epsilon \right\} \\ &\qquad \text{using integration by substitution:} \\ \Leftrightarrow \nabla_{\boldsymbol{\theta}} \{\mathcal{L}(\boldsymbol{\theta})\} &= \int_0^1 p(\epsilon) \nabla_{\boldsymbol{\theta}} \left\{\log p\left(\mathbf{x}, g(\epsilon; \boldsymbol{\theta})\right) - \log q\left(g(\epsilon; \boldsymbol{\theta}) \mid \boldsymbol{\theta}\right)\right\} d\epsilon \end{aligned} \tag{1.22}$$

given that $p(\epsilon) = \mathcal{U}(0, 1)$ and $q\left(g(\epsilon; \boldsymbol{\theta}) \mid \boldsymbol{\theta}\right) \nabla_\epsilon \{g(\epsilon; \boldsymbol{\theta})\} = 1$ (see Equation (1.21)). Equation (1.22) is of particular interest to us as it shows that if we want to sample gradients from the ELBO, we can do so by first sampling some ϵ and then computing the gradient of the log-ratio between $p(\mathbf{x}, g(\epsilon; \boldsymbol{\theta}))$ and $q(g(\epsilon; \boldsymbol{\theta}) \mid \boldsymbol{\theta})$ wrt $\boldsymbol{\theta}$, which has an analytical form and can be readily computed using automatic differentiation. Similar, although

slightly more tedious, derivations could be achieved for ϵ that would not be uniformly distributed.

A fair amount of research has been conducted on reparameterized gradients in order to decrease their variance, for instance by the use of control variates [95].

Box 1.1.4| Reparameterization trick for discrete variables and acceptance-rejection generated latent variables

The form of reparameterization that we have exposed assumes that the CDF of q is invertible and that we can derive a differentiable function $\mathcal{T}(\epsilon; \boldsymbol{\theta})$ that constitutes the end point of the propagation of the gradient.
These restrictions preclude the use of this technique in two important cases: when discrete variables are generated (in which case g is not differentiable) and for random variables generated according to an acceptance rejection algorithm, mainly the Gamma, Inverse-Gamma, Beta, Dirichlet and many truncated distributions[a]. Note that being able to reparameterize the Gamma distribution automatically leads to tractable gradients for the Inverse-Gamma, Beta and Dirichlet distribution. Coupling the two techniques sketched here allows us to compute a tractable gradient for the Dirichlet-multinomial compound distribution.

1.1.2.2.7.1 Discrete variables The solution for discrete variables is rather simple and holds in a very neat and short algorithm: the core idea is just to marginalize over ϵ to retrieve the gradient. This is the approach proposed by Tokui and Sato [96], and the following working example gives the intuition of the method they propose. Let

$$p(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2, y) = p(\mathbf{z}_1)p(\mathbf{z}_2)p(y) \times \begin{cases} p(\mathbf{x} \mid \mathbf{z}_1) & \text{if } y = 1 \\ p(\mathbf{x} \mid \mathbf{z}_2) & \text{otherwise} \end{cases}$$

be the joint probability of our model. We want to derive an approximate posterior over y, for instance a binomial distribution with parameters $q(y|\omega_y) = \sigma(\omega_y)$ for some $\sigma : \mathbb{R} \mapsto (0, 1)$ and $\omega \in \mathbb{R}$. We can sample from $q(y \mid \omega_y)$ by applying the following rule

$$\widetilde{y} = \begin{cases} 1 & \text{if } \epsilon < \sigma(\omega_y) \\ 0 & \text{otherwise} \end{cases} \qquad \text{for some } \epsilon \sim \mathcal{U}(0, 1)$$

which is not differentiable. To optimize ω_y, we can marginalize over ϵ, to get a tractable gradient:

$$\begin{aligned}\frac{\partial}{\partial \omega_y}\mathbb{E}_{q(y)}\left[\log p(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2, y)\right] &= \sum_{y_i=\{0,1\}} \frac{\partial q(y_i \mid \omega_y)}{\partial \omega_y} \log p(\mathbf{x}, \widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2, y_i) \\ &= \sum_{y_i=\{0,1\}} \frac{\partial \sigma(s\omega_y)}{\partial \omega_y} \log p(\mathbf{x}, \widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2, y_i) \\ &\text{where } s = \begin{cases} 1 & \text{if } y = 1 \\ -1 & \text{otherwise} \end{cases}\end{aligned}$$

when σ is symmetric around $\omega = 0$.

1.1.2.2.7.2 Acceptance-rejection algorithms Two main techniques can be distinguished in the case of acceptance-rejection simulated variables: The Generalized reparameterization Gradient [97] works by generating a sample $\mathbf{y} \sim q(\mathbf{y} \mid \boldsymbol{\theta})$, then by mapping $\mathbf{y}$ to ϵ by some transformation $\epsilon = \mathcal{T}^{-1}(\mathbf{y}; \boldsymbol{\theta})$ which ensures that the first moment of ϵ (but not necessarily higher moments) is independent of $\boldsymbol{\theta}$[b]. Then, we have a distribution $q(\epsilon)$ of $q(\mathbf{y})$ with the form $\epsilon \sim q(\epsilon \mid \boldsymbol{\theta})$, and the reparameterization trick can be applied to this sampling scheme.

Rejection Sampling Variational Inference [98] differs from the former approach by correcting the gradient by the probability that a given sample $\widetilde{\mathbf{y}} = \mathcal{T}(\epsilon; \boldsymbol{\theta})$ is accepted given a proposal distribution $r(\mathbf{y} \mid \boldsymbol{\theta})$. The two approaches share many similarities: in both, the noisy gradient of the expected log joint probability can be decomposed into the sum of a term representing a standard reparameterized gradient and a term correcting for the dependency of $q(\epsilon)$ on $\boldsymbol{\theta}$, for the former approach, and a term correcting for the acceptance probability for the latter. Note that the approach developed in [98] exhibits a lower variance than the other.

As a final observation, one can see that the reparameterized gradient can be simply computed if the CDF of the proposal distribution can be differentiated wrt its parameters [99, 91]. Although it is not the case for the Gamma and related distributions, it can lead to efficient gradient approximations that have a low variance.

[a] Fortunately, the form of the CDF of the one-dimensional truncated Gaussian distribution allows us to derive a convenient gradient, easily used with Automatic Differentiation. This is shown in the supplementary materials of Chapter 2.

[b] Recall that the reparameterization trick requires ϵ to be independent of $\boldsymbol{\theta}$, which is not the case here.

In practice, sampled gradients exhibit a lower variance with this technique than with the

score-function one that we will describe in the next section Paragraph 1.1.2.2.8. This method can be extended to distributions where an inverse CDF or a differentiable random sample generator does not exist Box 1.1.4. Also, parameters can be mapped onto a different space by using non-linear transforms such as logistic transforms, exponentiation etc., the only requirement being that the probability density has to be corrected by the Jacobian of the inverse transform. This is the kind of approach we will use with Normalizing Flows in Chapter 2.

Doubly Stochastic Variational Inference (DSVI) [100] uses a similar idea as the reparameterized gradient, but generalizes it to any standard distribution $r(\epsilon)$ from which samples can be drawn: if we define $q(\mathbf{y} \mid \boldsymbol{\theta})$ to be equal to

$$\begin{aligned} q(\mathbf{y} \mid \boldsymbol{\theta}) &= \frac{1}{\mid \Sigma^{1/2} \mid} r\left(\Sigma^{-1/2}(\mathbf{y} - \mu)\right) \\ \text{where } \boldsymbol{\theta} &= \left\{\mu, \Sigma^{1/2}\right\} \end{aligned} \tag{1.23}$$

This form of approximate posterior makes the derivation of a differentiable auxiliary function $\mathcal{T}(\epsilon; \boldsymbol{\theta})$ straightforward.

Example 1.1.2| Example of DSVI approximate posteriors

Let us define $r(\epsilon)$ as an Inverse-Gamma distribution with the form $\epsilon \sim \mathcal{G}^{-1}(\alpha, \beta)$, where $\{\alpha, \beta\}$ are chosen arbitrarily. We can define an approximate posterior according to Equation (1.23) with the formula:

$$q(y \mid \boldsymbol{\theta}) = \frac{\sigma\left(\frac{y-m}{s}\right)}{s} r\left(\texttt{softplus}\left((y-m)/s\right)\right) \tag{1.24}$$

where $\sigma(\cdot)$ is the logistic function, and where the first term is the Jacobian mass correction of the distribution. The differentiable auxiliary function $\mathcal{T}(\epsilon; \boldsymbol{\theta})$ with $\boldsymbol{\theta} = \{m, s\}$ is differentiable and can therefore be used to compute the stochastic gradient of the ELBO.

This distribution has the property of having a right heavy tail.

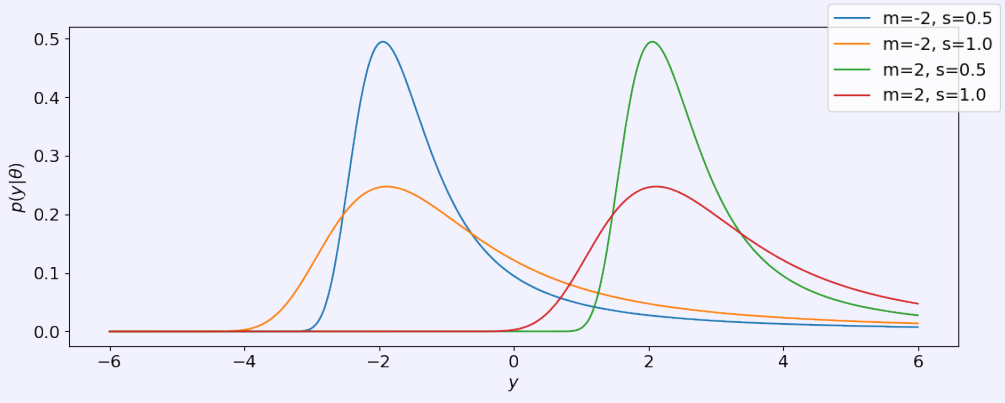


FIGURE 1.7: Various configurations of the approximate posterior are shown in the above figure. The mean plays the role of a location parameter, the standard deviation the role of the scale parameter. We can sample $y \sim q(y \mid \boldsymbol{\theta})$ by sampling $\epsilon \sim \mathcal{G}^{-1}(\alpha, \beta)$ and then mapping ϵ to y by inverting the transformation in Equation (1.24):

$$\widetilde{y} = s \log\left(\exp\{\epsilon\} - 1\right) + m.$$

1.1.2.2.8 Black-Box Variational Inference The reparameterization trick allows gradients to be estimated stochastically when the ELBO is too expensive to be computed exactly. One of its main advantages is that, for many models, the gradient can be easily computed with automatic differentiation toolboxes: for instance, if a Gaussian sample is computed using $\mu + \Sigma^{1/2}\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$, then the Jacobian of this expression with respect to μ and $\Sigma^{1/2}$ can be used to compute a sampled gradient of the ELBO. However, it is not highly generalizable: the restrictions we have put on $\mathcal{T}$ make it hard for many models to use SGVB in a blind manner. Here we present Black-Box Variational Inference (BBVI) [101] as a method to generalize stochastic approximations of the ELBO and its gradient to a wide family of variational models.

Our aim will be to derive another stochastic gradient for the ELBO:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}\{\mathcal{L}(\boldsymbol{\theta})\} &= \nabla_{\boldsymbol{\theta}}\left\{\int q(\mathbf{y} \mid \boldsymbol{\theta})\left(\log p(\mathbf{x}, \mathbf{y}) - \log q(\mathbf{y} \mid \boldsymbol{\theta})\right) d\boldsymbol{\theta}\right\} \\
&= \int \nabla_{\boldsymbol{\theta}}\{q(\mathbf{y} \mid \boldsymbol{\theta})\}\left(\log p(\mathbf{x}, \mathbf{y}) - \log q(\mathbf{y} \mid \boldsymbol{\theta})\right) d\boldsymbol{\theta} \\
&\quad \underbrace{- \int q(\mathbf{y} \mid \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}\{\log q(\mathbf{y} \mid \boldsymbol{\theta})\} d\boldsymbol{\theta}}_{=0}
\end{aligned} \tag{1.25}$$

and given that $\nabla_{\boldsymbol{\theta}}\{q(\mathbf{y} \mid \boldsymbol{\theta})\} = q(\mathbf{y} \mid \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}\{\log q(\mathbf{y} \mid \boldsymbol{\theta})\}$,

$$\begin{aligned}
&= \int q(\mathbf{y} \mid \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}\{\log q(\mathbf{y} \mid \boldsymbol{\theta})\}\left(\log p(\mathbf{x}, \mathbf{y}) - \log q(\mathbf{y} \mid \boldsymbol{\theta})\right) d\boldsymbol{\theta} \\
&\approx \frac{1}{K} \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}}\{\log q(\widetilde{\mathbf{y}}_k \mid \boldsymbol{\theta})\}\left(\log p(\mathbf{x}, \widetilde{\mathbf{y}}_k) - \log q(\widetilde{\mathbf{y}}_k \mid \boldsymbol{\theta})\right).
\end{aligned}$$

Note that this estimator does *not* require to compute the gradient of the log-joint probability, neither does it necessitate to compute the inverse CDF of q.

Because unlike the reparameterized gradient, the *score function* (or *likelihood ratio* and *REINFORCE* in the RL literature, see Section 1.2.2) method does not use the gradient of the log-joint probability, it can be applied to models where the likelihood is not differentiable. However, this estimator has unfortunately a high variance. Several tools

can be used to improve it: we will quickly present two of such techniques in the following paragraphs.

1.1.2.2.8.1 Control variates The method of control variates works by observing that we can reduce the estimate variance of some random variable $f(y)$ by introducing an auxiliary variable $g(y)$ from which we know the expected value under the same process $p(y)$. Let us first observe that the estimator

$$\begin{aligned} f^*(y) &= f(y) + c * (g(y) - \mu_g(y)) \\ &\text{with } \mu_g(y) = \mathbb{E}_p\left[g(y)\right] \end{aligned} \tag{1.26}$$

has the same expectation but a lower variance than $f(y)$ under specific conditions that we will detail now. Let the variance of $f^*(y)$ be

$$\mathbb{V}ar_{p(y)}\left[f^*(y)\right] = \mathbb{V}ar_{p(y)}\left[f(y)\right] + c^2\mathbb{V}ar_{p(y)}\left[g(y)\right] + 2\mathbb{C}ov_{p(y)}\left[f(y), g(y)\right]$$

which can be obtained from the variance-covariance formula and is minimized wrt c when

$$c = -\frac{\mathbb{C}ov_{p(y)}\left[f(y), g(y)\right]}{\mathbb{V}ar_{p(y)}\left[g(y)\right]}$$

This means that, if we choose $g(y)$ carefully enough to be (positively or negatively) correlated with $f(y)$, we can minimize the variance of the gradient estimator just by substituting $f(\widetilde{y})$ by $f^*(\widetilde{y})$ from Equation (1.26). In BBVI, this is typically done by choosing $g(\mathbf{y})$ to be the score function $\nabla_\phi \{\log q(\mathbf{y} \mid \boldsymbol{\theta})\}$, which has a expected value of 0 under q.

Of course, there is no reason to limit the use of control variates to BBVI, and it has been proposed that a similar approach could be used to reduce the variance of the reparameterization gradient [95].

1.1.2.2.8.2 Rao-Blackwellisation Consider a model $p(\mathbf{x}, \mathbf{y})$ where $\mathbf{y}$ can be decomposed into $\mathbf{y} = \{\mathbf{y}_j\}_{j=1}^L$ and where we use the mean-field assumption, so that the posterior probability density has the form $q(\mathbf{y}) = \prod_{j=1}^L q(\mathbf{y}_j \mid \boldsymbol{\theta}_j)$. When sampling gradients from the ELBO according to Equation (1.25), we can observe that a naive version of the BBVI algorithm will produce samples of $\nabla_{\boldsymbol{\theta}_j}\{\mathcal{L}(\boldsymbol{\theta})\}$ with a high variance, which will be partly increased by the randomness in the concurrent sampling of other variables $\mathbf{y}_{\neg j}$. More precisely, let us consider the expectation $\mathbb{E}_{q(\mathbf{y})}\left[f(\mathbf{y}_j, \mathbf{y}_{\neg j})\right]$. We observe that

this expectation can be re-written as:

$$\mathbb{E}_{q(\mathbf{y}_j,\mathbf{y}_{\neg j})}\left[f(\mathbf{y}_j,\mathbf{y}_{\neg j})\right] = \int q(\mathbf{y}_j) \underbrace{\int q(\mathbf{y}_{\neg j}) f(\mathbf{y}_j,\mathbf{y}_{\neg j}) d\mathbf{y}_{\neg j}}_{\triangleq \widehat{f}(\mathbf{y}_j)=\mathbb{E}_{q(\mathbf{y}_{\neg j})}[f(\mathbf{y}_j,\mathbf{y}_{\neg j})|\mathbf{y}_j]} d\mathbf{y}_j.$$

We see that $\widehat{f}(\mathbf{y}_j)$ and $f(\mathbf{y}_j,\mathbf{y}_{\neg j})$ share the same expected value. However, $\widehat{f}(\mathbf{y}_j)$ can be shown to have a lower variance than f. This suggests that, to estimate the gradient $\nabla_{\boldsymbol{\theta}_j}\{\mathcal{L}(\boldsymbol{\theta})\}$, if we can solve part of the integral and not apply the estimator in a brute force way, we will reduce the variance of the estimator. This is indeed the case in the computation of the gradient of the ELBO. If we can decompose the log-joint probability $\log p(\mathbf{x},\mathbf{y}) = f(\mathbf{y}_j) + f(\mathbf{y}_{\neg j})$, where $f(\mathbf{y}_j)$ groups all the terms of the Markov Blanket of $\mathbf{y}_j$, we have

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_j}\{\mathcal{L}(\boldsymbol{\theta})\} = \mathbb{E}_{q(\mathbf{y}_j)}\Big[\mathbb{E}_{q(\mathbf{y}_{\neg j})}\Big[\nabla_{\boldsymbol{\theta}_j}\{\log q(\mathbf{y}_j \mid \boldsymbol{\theta}_j)\}\left(f(\mathbf{y}_j) + f(\mathbf{y}_{\neg j})\right.\\ \left.- \log q(\mathbf{y}_j \mid \boldsymbol{\theta}_j) - \log q(\mathbf{y}_{\neg j} \mid \boldsymbol{\theta}_{\neg j})\right)\Big]\Big].\end{aligned}$$

where we can use Fubini's theorem to isolate and simplify

$$\mathbb{E}_{q(\mathbf{y}_j)}\left[\mathbb{E}_{q(\mathbf{y}_{\neg j})}\left[\nabla_{\boldsymbol{\theta}_j}\{\log q(\mathbf{y}_j \mid \boldsymbol{\theta}_j)\}\underbrace{\left(f(\mathbf{y}_{\neg j}) - \log q(\mathbf{y}_{\neg j} \mid \boldsymbol{\theta}_{\neg j})\right)}_{=C(\mathbf{y}_{\neg j})}\right]\right] =$$

$$\mathbb{E}_{q(\mathbf{y}_{\neg j})}\left[\underbrace{\mathbb{E}_{q(\mathbf{y}_j)}\left[\nabla_{\boldsymbol{\theta}_j}\{\log q(\mathbf{y}_j \mid \boldsymbol{\theta}_j)\}\right]}_{=0} C(\mathbf{y}_{\neg j})\right] = 0$$

from this formula, giving the simplifies Rao-Blackwellized form of the gradient

$$\nabla_{\boldsymbol{\theta}_j}\{\mathcal{L}(\boldsymbol{\theta})\} = \mathbb{E}_{q(\mathbf{y}_j)}\left[\nabla_{\boldsymbol{\theta}_i}\{\log q(\mathbf{y}_j \mid \boldsymbol{\theta}_j)\}\left(\log f(\mathbf{y}_j) - \log q(\mathbf{y}_i \mid \boldsymbol{\theta}_j)\right)\right]. \tag{1.27}$$

Intuitively, Rae-Blackwellisation is quite easy to understand in this context: just by removing from the gradient sample each and every part that is not strictly dependent upon $\boldsymbol{\theta}_j$ and, consequently, dependent of $\mathbf{y}_j$ too (i.e. that does not belong to the Markov Blanket of $\mathbf{y}_j$), we ensure that the gradient estimate has a lower variance while keeping the same expectation. Once more, this can be viewed as a form of message passing algorithm, where the update of $\boldsymbol{\theta}_j$ is equal to the sum of the messages sent by the neighbouring nodes.

1.1.2.2.8.3 Overdispersed BBVI As for SGVB, IS can be used to improve the estimator in Equation (1.25). We first note that the optimal proposal distribution (i.e. the proposal distribution that minimizes the variance of the estimator) in IS is *not* the

probability distribution according to which the random variable is distributed [23], but has the following form instead:

$$r^*(y) = \frac{\mid f(y) \mid q(y)}{\int \mid f(y) \mid q(y) dx}$$

for some function $f(y)$ and some distribution over y, $q(y)$. Overdispersed BBVI (O-BBVI) [102] builds on this observation, and replace the sampling scheme outlined above by an IS estimator. If we restrict ourselves to cases where $q(\mathbf{y})$ belongs to the exponential family and has a form $\mathbf{h}(\mathbf{y}) \exp\left\{\boldsymbol{\theta}^T \mathbf{T}(\mathbf{y}) - A(\boldsymbol{\theta})\right\}$, we can construct and overdispersed distribution $r(\mathbf{y} \mid \boldsymbol{\theta}, \tau)$ with the form

$$r(\mathbf{y} \mid \boldsymbol{\theta}, \tau) = h(\mathbf{y}, \tau) \exp\left\{\frac{\boldsymbol{\theta}^T \mathbf{T}(\mathbf{y}) - A(\boldsymbol{\theta})}{\tau}\right\}$$

where $\tau \geq 1$ is a dispersion coefficient optimized online in order to minimize the variance of the estimator. O-BBVI achieves a much better performance than BBVI both in terms of ELBO maximization and variance reduction.

Box 1.1.5| Highly flexible approximate posteriors

The use of stochastic gradients alleviates the need to use approximate posteriors of specific forms (exponential family etc.) Many publications have focused on deriving highly flexible distributions for this purpose. Importantly, there is no risk of overfitting by increasing the flexibility q: the more flexible q is, the closer the approximate posterior will be from the true posterior. Also, an important restriction to keep in mind is that, for conjugate models where the Mean-field assumption is used, the conjugate prior of each distribution constitutes the best choice of approximate posterior in this setting.

We will give some examples of these highly flexible distributions here to illustrate the many potentialities of these techniques. Normalizing flows [103] work by applying multiple successive transformations (scaling, rotation) to an approximate posterior of a simple form. At each transformation, the probability density has to be rescaled by the determinant of the Jacobian of the inverse transformed in order for the distribution to keep the same volume. For a sequence of transformations $f_n \circ f_{n-1} \circ ... \circ f_1$ of an initial distribution $p_0(z_0)$, the log-density of $\mathbf{y}_n$ has the form:

$$\log p_n(\mathbf{y}_n) = \log p_0(\mathbf{y}_0) - \sum_{i=1}^{n} \left| \det \frac{\partial f_i(\mathbf{y}_{i-1})}{\partial \mathbf{y}_{i-1}} \right|$$

$$\text{where } \mathbf{y}_n = f_n \circ ... \circ f_1(\mathbf{y}_0).$$

A similar approach is used with Automatic Differentiation Variational Inference [104], where a normal distribution is used as a generic approximate posterior, and then transformed to match the desired target space.

Rezende and Mohamed [103] have proposed mainly two forms of finite flows (planar and radial), but these transformations are not easily parallelizable and do not scale well to

large models. Inverse Autoregressive flows [105] have been proposed as an alternative to these transformations in order to solve the two aforementioned problems: their peculiar architecture greatly simplifies both the transformation step:

$$\mathbf{y}_n = \sigma \odot \mathbf{y}_{n-1} + (1 - \sigma) \odot m$$

for some $m, \sigma = f(\mathbf{y}_{n-1})$ and the mass correction step

$$\log p_n(\mathbf{y}_n) = \log p_{n-1}(\mathbf{y}_{n-1}) - \text{sum}(\log \sigma).$$

The Householder flow [106] builds on a somehow similar idea, where the transformation applied to $\mathbf{y}_{n-1}$ is restricted to a very specific form that keeps the volume of the initial transformation, and does not require any mass correction.
Finally, we should mention that substantial efforts have also been made in order to account for covariance between latent variables. An idea is to build hierarchical Variational Models [107] where the latent parameter is marginalized over its own prior distribution, and only this prior is optimized. Variational GP [108] is a special case of this framework, that enjoys the property of being able to model any desired posterior shape.

1.1.2.2.9 Amortized Variational Inference and Variational Auto-Encoders

In many models, our aim will be to compute the marginal likelihood of the data given some hyperparameters while integrating out some latent local parameters. In many cases where Variational Inference is used, the amount of datapoints where the local approximate posterior has to be fitted is considerable, and optimizing each one independently is computationally demanding, both in term of computation time, computational burden and memory allocation. SVI already provided an incomplete solution to this problem, as it still necessitated to optimize the fit the local approximate posteriors for each observation.

Amortized[10] Variational Inference (AVI) [88, 109] (aka Neural Variational Inference and Learning [110]) works by alleviating the need to compute each local approximate posterior parameters independently. First, note that in a one-level hierarchical model of the form $p(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \mathbf{y}_i)p(\mathbf{y}_i)$, each local posterior distribution is entirely determined by the datapoint $\mathbf{x}_i$. Therefore, there must be a mapping from $\mathbf{x}_i$ to $q(\mathbf{y}_i \mid \boldsymbol{\theta}_i)$, namely $q_{\boldsymbol{\rho}}(\mathbf{y}_i \mid \mathbf{x}_i)$ that can amortize the cost estimating $\boldsymbol{\theta}_i$ for each point.

[10] We distinguish three understandings of the term "amortized inference" in this manuscript: the first is the one used in this paragraph, which is the idea that instead of finding the per-observation local approximate posterior, we map the value of the observation $\mathbf{x} \in \mathcal{X}$ for a given prior to some variational parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ using a flexible function $f_{\boldsymbol{\rho}} : \mathcal{X} \mapsto \boldsymbol{\theta}$. The second refers to the idea that, in RL, given a model of the state-action to state transition function, one wastes resources if she achieves the same simulation each time the same state is encountered. In other words, using model-based RL in a stable environment requires some technique of storing previous simulations for future inference. The last use of the term amortized inference relates significantly to the one just described. It consists in re-using the simulation achieved at the preceding trial in model-based RL to make inference at the current trial. In any game, for instance, inference is not conducted from scratch at each step, but the model is refined each time a new state is encountered.

As our notation suggests, AVI uses this observation to optimize auxiliary parameters $\boldsymbol{\rho}$ of a transformation $\boldsymbol{\theta}_i = f(\mathbf{x}_i; \boldsymbol{\rho})$. This non-linear transformation has usually the form of an artificial neural network, and the transformation $f(\cdot)$ is usually referred to as the inference network (or encoder in the case of variational auto-encoders).

Not only AVI lowers the cost in terms of storage, but it also makes the algorithm generalizable to unseen datapoints: any point $\mathbf{x}_j \notin \mathbf{x}$ can be mapped onto its correspondent $\boldsymbol{\theta}_j$ using the same transformation as the one derived using $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^N$, and for a sufficiently large and representative N, we can assume that $\boldsymbol{\theta}_j = f(\mathbf{x}_j; \boldsymbol{\rho})$ is close to its true optimum $\boldsymbol{\theta}_j^*$.

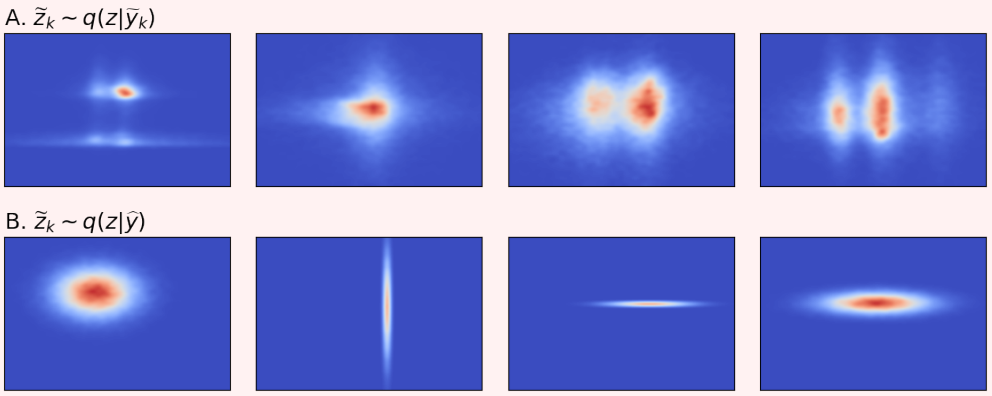
Observing that such scheme can be slightly too restrictive, especially in the case where $\boldsymbol{\theta}_i$ can be efficiently optimized using SVI, [87] propose a mixed approach where a first approximation of $\boldsymbol{\theta}_{i0}$ is generated with an inference network before being refined using another differentiable local SVI algorithm to produce a final, better estimate approximate posterior with parameters $\boldsymbol{\theta}_{in}$. This is motivated by the observation that AVI adds an *amortization gap* to the *approximation gap* that is inherent to VI: it might be shown that the former can exceed the latter in many instances [111]. During training, the final gradient $\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is corrected by backpropagating each further refinement of $\boldsymbol{\theta}_{in}$ back to the original gradient.

Box 1.1.6| AVI for hierarchical models

AVI is mostly used for models such as Variational Auto-encoders, where the latent variable prior is known in advance and fixed to a simple value (such as a unit-variance Gaussian distribution). However, more hierarchical models may be worth studying. If a local latent variable $\mathbf{z}_i$ is itself distributed according to a prior distribution $p(\mathbf{z}_i \mid \mathbf{y})$ for an unknown $\mathbf{y} \sim p(\mathbf{y})$, then a simple approach to estimate $q(\mathbf{z}_i \mid \boldsymbol{\theta}_i)$ is to use a form where the inference network receives inputs from both $\mathbf{x}_i$ and $\mathbf{y}$, so that the approximate posterior of $\mathbf{z}_i$ now reads $\mathbf{z}_i \sim q(\mathbf{z}_i \mid \boldsymbol{\theta}_i; \mathbf{z}_i, \mathbf{y})$. In practice, the sampling procedure of $\mathbf{z}_i$ will somehow reflect the hierarchy in the model: using the reparameterization trick, we can sample $\widetilde{\mathbf{y}} \sim q(\mathbf{y})$ before sampling $\mathbf{z}_i \sim q(\mathbf{z}_i \mid \boldsymbol{\theta}_i; \mathbf{x}_i, \widetilde{\mathbf{y}})$, and the approximate posterior has a form similar to the model structure.
With such model, multimodal posteriors of $\mathbf{z}_i$ can be accounted, if this multimodality is related to the prior probability of $\mathbf{z}_i$, and so without any need for normalization of the approximate over $\mathbf{z}_i$, as

$$\begin{aligned}\iint q(\mathbf{z} \mid \boldsymbol{\theta}, \mathbf{y}) q(\mathbf{y}) d\,\mathbf{z}\, d\mathbf{y} &= \int q(\mathbf{y}) \int q(\mathbf{z} \mid \boldsymbol{\theta}, \mathbf{y}) d\,\mathbf{z}\; d\mathbf{y} \\ &= \int q(\mathbf{y}) d\mathbf{y} \\ &= 1.\end{aligned}$$

The image below gives an intuition of how multilevel AVI works: by first sampling some $\widetilde{\mathbf{y}}_k$ and then sampling $\widetilde{\mathbf{z}}_k$ conditioned on $\widetilde{\mathbf{y}}_k$, we can create a complex and multimodal approximate posterior space for $\mathbf{z} \mid \mathbf{y}$.



The above graph shows some approximate posterior density of $\mathbf{z}_i$ using an inference network with input $\{\mathbf{x}_i, \widetilde{\mathbf{y}}_k\}$. The second row shows the density (Gaussian in this case) of $\mathbf{z}_i$ conditioned on the most likely value of $\mathbf{y}$ alone, $\widehat{\mathbf{y}}$.

1.1.2.2.9.1 Variational Auto-Encoders Having introduces AVI as a method to estimate local approximate posteriors for non-conjugate models, we can now introduce one of the most successful inference algorithms for generative models, namely variational auto-encoder (VAE) [88, 89]. At its core, VAE is a Bayesian dimensionality reduction algorithm: given a local set of observations $\mathbf{x}_i$ where $\mathbf{x}_i \in \mathbb{R}^d$ with $d \gg 0$, auto-encoders (AE) are constituted of an *encoder* f trained to map this data to a space of reduced dimensionality $\mathbf{z} = f_{\boldsymbol{\rho}}(\mathbf{x})$ with $\mathbf{z} \in \mathbb{R}^m$, $m < d$ and prior density $\mathbf{z} \sim \mathcal{N}(0, 1)$, and a decoder $f_{\boldsymbol{\phi}}$ that translates $\mathbf{z}$ back to the space of $\mathbf{x}$. VAE differs from AE by their capacity to account for the uncertainty in the value of $\mathbf{z}$: now, $\mathbf{z}$ is a considered as a latent variable with prior distribution $\mathcal{N}(0, 1)$ and approximate posterior $q(\mathbf{z} \mid \boldsymbol{\theta})$. The encoder does not encode $\mathbf{z}$ directly anymore, but rather produce for a given sample a set of approximate posterior parameters $\boldsymbol{\theta} \triangleq f_{\boldsymbol{\rho}}(\mathbf{x})$. VAE and extensions show remarkable performances in generating speeches [112], music samples [113, 114], faces [115], landscapes and animals [116] or molecule configurations [117].

In its original flavour, the VAE paradigm can simply be implemented as a non-linear mapping from $\mathbf{z}$ to $\boldsymbol{\delta}$ where $\boldsymbol{\delta}$ are the parameters of the distribution of $\mathbf{x}$: $\mathbf{x} \sim p(\mathbf{x} \mid f_{\boldsymbol{\phi}}(\mathbf{z}))$. Training of the encoder (which is aimed at making inference about $\mathbf{z}$) and training of the decoder are usually conducted simultaneously by combining AVI, the reparameterization gradients and SGVB.

Box 1.1.7| Importance Sampling as an alternative to SGVB

Variational Auto-Encoders [88] have constituted the first and probably mostly used case of SGVB-based Variational models. The use of a variational bound has however the undesirable property that the model is penalized by the assumptions made about the approximate posterior shape (distribution family, mean-field assumption). This small issue can be easily alleviated. It has been proposed that, in order to reduce the variance of this estimator, importance sampling could be used [118]. The idea is quite simple: the approximate posterior q can be used as a proposal distribution instead of an approximate

posterior. In VAE, the variational lower bound was expressed as

$$\log p(\mathbf{x}) \geq \mathcal{L}(\boldsymbol{\rho}) = \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right] \approx \frac{1}{K}\sum_{k=1}^{K} \log \frac{p(\mathbf{x},\widetilde{\mathbf{z}}_k)}{q(\widetilde{\mathbf{z}}_k)}.$$

In many instances, the value of K is set to 1, as the convergence improvement induced by increasing it has been empirically shown to be marginal. Now consider what would happen if we were to move the sum operator *inside* the logarithm:

$$\mathcal{L}_K(\boldsymbol{\rho}) = \mathbb{E}_{q(\mathbf{z})}\left[\log \sum_{k=1}^{K} \frac{p(\mathbf{x},\widetilde{\mathbf{z}}_k)}{q(\widetilde{\mathbf{z}}_k)}\right].$$

We would then keep the lower-bound property of our estimator: using Jensens' inequality, we can easily see that

$$\mathbb{E}_{q(\mathbf{z})}\left[\log \sum_{k=1}^{K} \frac{p(\mathbf{x},\widetilde{\mathbf{z}}_k)}{q(\widetilde{\mathbf{z}}_k)}\right] \leq \log \mathbb{E}_{q(\mathbf{z})}\left[\sum_{k=1}^{K} \frac{p(\mathbf{x},\widetilde{\mathbf{z}}_k)}{q(\widetilde{\mathbf{z}}_k)}\right] = \log p(\mathbf{x})$$

with the only difference that (for bounded value of the ratio $p(\mathbf{x},\mathbf{z})/q(\mathbf{z})$) its expected value for $K \to \infty$ would be equal to the one of $\log p(\mathbf{z})$.
For auto-encoders, this approach is known as Importance Weighted Auto-Encoders. The fact that sampling is achieved inside the log operator (and not outside) should not be overlooked as a mere implementation detail, and, in general, one might gain in accuracy by sampling both inside and outside the logarithm [119] (in fact, Rainforth and colleagues [120] suggest that, for nested simulated integrals $\mathbb{E}_{q_1}[f(\mathbb{E}_{q_2}[\mathbf{y}])]$, the sample size of the inner integral should be quadratic in the number of the outer integral).

VAE in its original form has lead to many controversies and improvement. To see why, note that Equation (1.15) can be rearranged as

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{y})}[p(\mathbf{x},\mathbf{y})] - D_{KL}\left[q(\mathbf{y})\middle\|p(\mathbf{y})\right]$$

and, therefore, it could be important to check how each of these two terms are optimized during the training of the model: there might be for instance two different solutions with a similar ELBO, one with a high KL divergence but a high generative accuracy, and another with a low KL but low generative power. When looking at this, a common observation has been that the model tends to reduce as much as possible the KL divergence between $q(\mathbf{y})$ and $p(\mathbf{y})$, thereby breaking the link between the encoder and the decoder. In other words, training a naive version of the VAE tends to make the encoder irrelevant to generate new data: at some point, the decoder can generate new samples with a low variability and discards what the latent space instructs him. Several solutions have been proposed to this problem,, which we do not detail here [121, 122, 123, 124].

1.1.2.2.10 Implicit Variational Inference As a final example of the use of stochastic variational models, we will introduce an interesting family of algorithms that makes VI applicable to problems where either the likelihood is intractable (i.e. when we can sample from it but do not know its precise formula), or when we want to make inference with a highly flexible approximate posterior such as a Bayesian neural network, which makes the computation of the posterior log-density impractical to retrieve for a limited and rational computational budget.

Despite the obvious challenges that follow from their use, implicit models constitute a rich class of generative models, for which many solutions can be developed [125].

Tran and colleagues [126] propose an algorithm for inference about likelihoods with an intractable density that can have an unbiased Monte-Carlo estimate. This include, for instance, marginal models such as Generalized Linear Mixed Models, where the random effects have been integrated out: in this case, the likelihood is generally intractable, but the integrands can be estimated easily and, therefore, an IS estimator can be retrieved. Moreno et al. [127] propose a method based on ABC [24], named Automatic Variational ABC, aimed at solving the problem of inference with intractable likelihood. ABC works by approximating an intractable distribution $p(x \mid \mathbf{y})$ by

$$\begin{aligned} p(x \mid \mathbf{y}) &= \int p(x \mid u, \epsilon) p(u \mid \mathbf{y}) du \\ &\approx \frac{1}{K} \sum_{k=1}^{K} p(x \mid \widetilde{u}, \epsilon) \qquad \text{where } \widetilde{u} \sim p(u \mid \mathbf{y}), \end{aligned}$$

$p(x \mid \widetilde{u}, \epsilon)$ can be, for instance, a Gaussian with mean $\widetilde{u}$ and standard deviation ϵ and $p(u \mid \mathbf{y})$ had the same form as $p(x \mid \mathbf{y})$. The reparameterization trick can be used if there exists a differentiable generator $\mathcal{T}(\epsilon; \mathbf{y})$ for $p(u \mid \mathbf{y})$, which is not always the case.

The core idea of the model proposed in [128] we will detail now relates to a generative framework, known as Generative Adversial Network (GAN), that somehow competes with Variational Auto-Encoders. Whereas VAE requires a likelihood function for the observations, GAN do not necessitate such explicit distributions. However, GAN are not inference models: there is no way to map a given sample to its latent space. GAN are trained by concurrently optimizing a generative model $G_{\boldsymbol{\phi}} : \mathbb{R}^d \mapsto \mathcal{X}$ with $\mathbf{z} \in \mathbb{R}^d$, $\mathbf{z} \sim q(\mathbf{z})$ to create samples $\widetilde{\mathbf{x}}$ similar to the actual data $\mathbf{x} = \{ \mathbf{x}_i \}_{i=1}^N \subseteq \mathcal{X}$, and a discriminator $D_{\boldsymbol{\lambda}} : \mathcal{X} \mapsto [0, 1]$ that returns the probability $p(\widetilde{\mathbf{x}} \subseteq \mathcal{X})$. With GANs, the following loss is optimized:

$$\{\boldsymbol{\phi}^*, \boldsymbol{\lambda}^*\} = \arg\min_{\boldsymbol{\phi}} \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_{p(\mathbf{x})} \left[\log D_{\boldsymbol{\lambda}}(\mathbf{x})\right] + \mathbb{E}_{q(\mathbf{z})} \left[\log(1 - D_{\boldsymbol{\lambda}}(G_{\boldsymbol{\phi}}(\mathbf{z})))\right]. \tag{1.28}$$

The process of training a GAN can be seen as a competition between a discriminator, which has to tell apart true and false samples, and a generator that tries to fool the discriminator. The better the generator, the better the discriminator must be to discriminate true and false samples. The better the discriminator, the more realistic must be a fake sample to fool it.

We can use a somehow similar idea to train implicit models [128] in an approach named Implicit Variational Inference (IVI): Let $\prod_{i=1}^N p(\mathbf{x}_i \mid \mathbf{z}_i) p(\mathbf{z}_i \mid \mathbf{y}) p(\mathbf{y})$ be the model from which we wish to estimate the posterior probability. We first re-write down the form of the ELBO in the following

form:

$$\mathcal{L}\left(q(\mathbf{y},\mathbf{z})\right) = \mathbb{E}_{q(\mathbf{y},\mathbf{z})}\left[\log p(\mathbf{y}) - \log q(\mathbf{y}) + \sum_{i=1}^{N} \log \frac{p(\mathbf{x}_i,\mathbf{z}_i \mid \mathbf{y})}{q(\mathbf{x}_i,\mathbf{z}_i \mid \mathbf{y})}\right] + \sum_{i=1}^{N} \log p_{\text{data}}(\mathbf{x}_i)$$

where we have used $q(\mathbf{x}_i,\mathbf{z}_i \mid \mathbf{y}) = p_{\text{data}}(\mathbf{x}_i)q(\mathbf{z}_i \mid \mathbf{x}_i,\mathbf{y})$. Note that, according to what we have seen earlier Paragraph 1.1.2.2.9, the inference network over $\mathbf{z}$ is conditioned on both $\mathbf{x}$ and $\mathbf{y}$.

Now if we could substitute $\log \frac{p(\mathbf{x}_i,\mathbf{z}_i|\mathbf{y})}{q(\mathbf{x}_i,\mathbf{z}_i|\mathbf{y})}$, which is by definition intractable, by a tractable log-ratio estimator $r(\mathbf{x},\mathbf{z})$, then we could estimate the value of the ELBO. Fortunately, we can sample from both distributions: to simulate from the numerator, we simply need to sample some $\widetilde{\mathbf{y}} \sim p(\mathbf{y})$, $\widetilde{\mathbf{z}}_i \sim p(\mathbf{z}_i \mid \mathbf{y})$ and then simulate $\widetilde{\mathbf{x}}_i \sim p(\mathbf{x}_i \mid \widetilde{\mathbf{y}}_i)$. For the numerator, we just need to randomly select some $\mathbf{x}_i$ from the dataset and then gather a sample from $\widetilde{\mathbf{z}}_i \sim q(\mathbf{z}_i \mid \mathbf{x}_i,\widetilde{\mathbf{y}})$.

IVI works by training some function r to discriminate samples drawn from $p(\mathbf{x}_i,\mathbf{z}_i \mid \mathbf{y})$ from samples drawn from $q(\mathbf{x}_i,\mathbf{z}_i \mid \mathbf{y})$, which will provide us an estimate of the ELBO at optimality. The value of r that best approximate the above fraction is given by the proper scoring rule [129]:

$$r^*(\mathbf{x},\mathbf{z}) = \arg\max_r \mathbb{E}_{p(\mathbf{x}_i,\mathbf{z}_i|\mathbf{y})}\left[\log \sigma(r(\mathbf{x}_i,\mathbf{z}_i,\mathbf{y}))\right] + \mathbb{E}_{q(\mathbf{x}_i,\mathbf{z}_i|\mathbf{y})}\left[\log 1 - \sigma(r(\mathbf{x}_i,\mathbf{z}_i,\mathbf{y}))\right] \tag{1.29}$$

where $\sigma : \mathbb{R} \mapsto (0,1)$ is the standard logistic function. To see how IVI works, consider for instance the case where $\sigma(r(\mathbf{x},\mathbf{z}))$ returns a value close to 1 (e.g. $r(\mathbf{x},\mathbf{z}) \to \infty$): in this case, the discriminator r estimates that it is infinitely more likely that $\widetilde{\mathbf{x}}_i,\widetilde{\mathbf{z}}_i \sim p(\mathbf{x}_i,\mathbf{z}_i \mid \mathbf{y})$ rather than $\widetilde{\mathbf{x}}_i,\widetilde{\mathbf{z}}_i \sim q(\mathbf{x}_i,\mathbf{z}_i \mid \mathbf{y})$. On the other hand, if $\sigma(r(\mathbf{x},\mathbf{z}))$ is close to 0, then the sample can be categorized as belonging to q rather than p. By training r to maximize the proper scoring rule, which both rely on non-linear transforms such as neural networks, we ensure that r gets as close at it can from the expected ratio between p and q. We can then concomitantly train q to maximize the ELBO: in some way, this approach is comparable to GAN, in that we train r to be as high as possible (thereby maximizing its power to discriminate between p and q) and train simultaneously q to gets as close as we can from the posterior distribution.

1.1.3 VI and the human brain: a short journey in Active Inference

The fact that VI has had a great success in neuroscience should not be surprising. We can individuate two main reasons for this: the first and most obvious is that VI allows researchers to fit complex neurophysiological models in a Bayesian way to the data they have acquired, thereupon retrieving from their models a measure of the *model evidence*, which can guide the researcher in making enlightened judgment about the causes of the observed behavior [130, 131, 73, 132, 133, 134, 135, 136, 137, 138, 139, 72, 140, 141].

The same principled method of inferring causes in the laboratory can be mapped onto a general model of brain functioning [142]: the brain can make models of the surrounding environments (i.e. make hypothesis), test, fit and confront these hypotheses against each other and against the real world [50]. The term *Bayesian Brain* encompasses many observations and theories covering the behavior of single neurons [143], population of neurons [144, 145, 146], brain areas [147]

and brain networks [148, 149] that have in common the fact that they suppose that the brain models are generative [150]. In other words, they assume that cerebral structures encode for distributions of events rather than events themselves: this allows the brain to perform causal inference about the surrounding world.

We start by describing the foundation of the lower bound maximization hypothesis of the brain inference process. The Helmoltz machine [151] can be viewed as the ancestor of the VAE outlined above. As for the VAE, this model is made of an encoder, which generates a simplified model of the observations (the q distribution) across several layers of a neural network. Then, from this top-layer where the explanations are inferred, a generative model is also trained to produce samples that mimic the real world. Each coder can be used to train the other in a scheme that is known as the wake-sleep algorithm [152]: during the sleep phase, the decoder is blocked and generates (or fantasizes) observations with a known cause. The encoder is then trained to recognize these observations, and map them to their latent cause space while minimizing the error between the true, known generative cause and its estimate. During the wake phase, the encoder is blocked and the decoder is trained to produce samples that resemble to the truth.

The interesting feature of the Helmoltz machine is that the principle of Free Energy minimization was already present in its initial formulation, even before VI was developed as we know it today: this theory of brain functioning assumed that the work of inference achieved by the execution of neural codes was aimed at minimizing the loss of information while modelling p with q. At its core, the negative Free Energy is the sum of two terms: a term reflecting the model fitness and a term reflecting the model entropy (i.e. generalization capacity). Formulated as such, it is not surprising that this quantity can bridge thermodynamical laws with other fields such as evolutionary biology [153, 154], where the genetic variance determines the fitness of the organism in its environment, or theories of self-organization and life formation [50].

Box 1.1.8| Reformulating the ELBO

The ELBO has many other interpretations which are summarized here:

$$\mathcal{L}(\phi) = \underbrace{\mathbb{E}_{q(x|\phi)}\left[\log p(a,x)\right]}_{\substack{\text{Approximate model fitness}\\ \text{Energy}}} + \underbrace{\mathbb{H}\left[q(x \mid \phi)\right]}_{\text{Entropy}} \tag{1.30}$$

$$= \underbrace{\mathbb{E}_{q(x|\phi)}\left[\log p(a \mid x)\right]}_{\text{Accuracy}} - \underbrace{D_{KL}\left[q(x \mid \phi)\middle\|p(x)\right]}_{\text{Model complexity}} \tag{1.31}$$

$$= \underbrace{\log p(a)}_{\substack{\text{Marginal likelihood}\\ \text{Log-model evidence}\\ \text{Negative surprise}}} \underbrace{- D_{KL}\left[q(x \mid \phi)\middle\|p(x \mid a)\right]}_{\text{Information loss}} \tag{1.32}$$

The *Approximate model fitness* reflects how well a model configuration is accurate on average, given a prior and a set of observations. The entropy reflects how generalizable is your approximation. The *model accuracy* ignores the prior likelihood, as the *model complexity* measures how much your posterior belief differs from your prior: if the complexity is high, it might be worth considering updating your prior or changing your approximate posterior. The *marginal likelihood* is an unknown quantity reflecting how good would be

our model if our approximations matched perfectly the reality. The *information loss* is equal to the amount of information that would be lost while coding q with p. Each of these formulations has its own interest.

Active inference generalizes this Free Energy minimization long-term goal that we framed in term of observation to the combine triad of action, learning and observation [49]. In this context, an agent policy, memory and perceptual state are all tuned to minimize the surprise in the environment [155]: this assumption is intended to act both as a normative value (what should be ideally be achieved) and as categorical imperative (what the agent is physiologically constrained to do). Note that here, surprise is intended as the negative log model evidence: minimizing surprise is equivalent to find the best model for the observations that are shown to the agent. Loosely speaking, such theoretical framework should entail any behavior, even reward-seeking policies. As phrased by Karl Friston and colleagues [155]:

> When friends and colleagues first come across this conclusion, they invariably respond with; "but that means I should just close my eyes or head for a dark room and stay there". In one sense this is absolutely right; and is a nice description of going to bed. However, this can only be sustained for a limited amount of time, because the world does not support, in the language of dynamical systems, stable fixed-point attractors. At some point you will experience surprising states (e.g., dehydration or hypoglycaemia).

One way to see how we can map the concept of rewards in terms of active inference, one can consider that a rewarded inferred state is a state that has a high prior probability. For instance, satiation and other homeostasis states can be considered as having a high prior probability. If the posterior does not match this prior, then the state is surprising, and the agent will act in such a way that the homeostasis is restored. In summary, active inference unifies the perception, learning (i.e. the critic) and the policy algorithms together under the umbrella of surprise minimization.

Box 1.1.9| Unfalsifiability of the Active Inference framework

The predictive brain is a very general theory, that actually goes way beyond the human brain: biological self-organization, evolution, homeostasis and many other biological phenomena can be interpreted in this way. One may therefore legitimately ask whether the active inference framework is not too general, as it seems to be unfalsifiable.

Indeed, we (and actually Karl Friston himself) can safely claim that active inference cannot be logically negated. This does not mean that this theory is not useful, and should not be endorsed: mathematics and logic, for instance, are constituted by a set of tautologies [156], and are therefore unfalsifiable.

The theory of biological self-organization [50] follows the same principle: it is

grounded on the following tautology (which is the basic understanding of the law of natural selection): an agent is likely to survive if he is able to adapt (i.e. change its internal state in response to changes in the surrounding environment). Following a precise mathematical formulation, Friston shows that this concept can be framed in a Bayesian (and more specifically in a Variational) manner. This is highly tautological: there is not falsifiable assumption, no testable concept, just a set of equivalencies that flow naturally and logically from the initial premises.

We made the explicit choice of not framing our work in an active inference perspective, in order to keep it as general and understandable as possible for an audience familiar with classical or Bayesian RL. In Section 5.2, we will attempt to map our theoretical contributions to the domain of active inference.

1.2 Reinforcement Learning

So far, we have introduced Bayesian Inference as general philosophy of model learning that appears to be sound both on the theoretical and biological point of view. In this section, we will introduce a family of machine learning algorithms that achieve the goal of maximizing a reward associated with certain states and actions. We will describe the various existing branches of Reinforcement Learning (RL), link them to the Bayesian theory presented before, and show how these algorithms find elegant analogs in animal behaviour.

RL is a family of ML algorithms and, loosely speaking, cognitive neuroscience theories that has the distinctive feature of tying together behaviour and learning. As such, RL constitutes an attractive tool for the study – and implementation – of intelligent animals and systems, respectively. Classically, RL assumes that an *agent* can gather *rewards* from the environment where it evolves by taking pre-defined *actions* in observed *states*. These actions or states can be discrete or continuous. It is usually assumed that the environment is *Markovian*: the Markov property ensures that the probability distribution of an observation x_t conditioned on all preceding observations $x_{<t}$ equals the marginal probability of this observation given the last one x_{t-1}. Formally, we have

$$p(x_t \mid x_{<t}) = p(x_t \mid x_{t-1}). \tag{1.33}$$

The set $\{ x_t \}_{t=1}^{T}$ that respects the condition in Equation (1.33) is called a first-order Markov chain. Other equivalent ways to express the Markov property are to say that the future and the past are conditionally independent given the present, or that the future depends on the past only through the present [157]. Many situations can be tricked to respect the Markov property: take for instance the second-order Markov Chain where we have $p\left(x_t \mid x_{t-1}\right) \neq p\left(x_t \mid x_{t-1}, x_{t-2}\right)$. If we define $y_s = \{x_{2s-1}, x_{2s}\}$, the chain of $Y = \{y_s, s = 0, 1, ..., N\}$ is a first order Markov chain. This feature is important to notice, as it means that problems that can only be solved by recurring to past history (i.e. that require a *memory* of the past) can be reformulated as a first-order Markov chain. For this reason, RL algorithms are designed to evolve in environments that respect this assumption, namely Markov Decision Processes (MDP).

Classically, an MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma, \mathcal{D})$. The set $\mathcal{S} = \{ s_i \}_{i=1}^{S}$ is the set of states and $\mathcal{A} = \{ a_i \}_{i=1}^{A}$ is the set of actions (which may or may not be identical in each state). The random variable $\mathcal{R}$ is the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, or alternatively a reward distribution with space $(\Omega, \mathcal{F}, \mathcal{R}(s, a))$ where the random variable is virtually always a real number $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \Omega \mapsto \mathbb{R}$. The random variable $\mathcal{T}$ is a Markovian transition process, or transition probability distribution s.t. it defines a

probability space $(\Omega, \mathcal{F}, \mathcal{T}(s,a))$ where the random variable is a state, i.e. $\mathcal{X} : \Omega \mapsto \mathcal{S}$. The real number $\gamma \in (0,1]$ is a geometric decay of future rewards and $\mathcal{D} \in \mathcal{S}$ is the set of initial possible states.

Box 1.2.1| Non-Markovian models

Several models do not have the Markov property: first of all comes Partially Observable MDP (POMDP). In a POMDP, at each state, one or more observations o are delivered to the agent which has to infer in which state it is based on a known model $p(o \mid s)$. This obviously requires it to use Bayes rule to inverse the probability distribution: in the naive approach, at each state the agent carries on a belief vector of the length of the state space where each element reflects the probability that this agent is in the corresponding state. When an action is taken, an observation is provided, and using a known (or estimated) transition matrix, a new belief vector can be generated. A POMDP should not be considered as Markovian, because the state in unsecured: the agent needs memory to know where it stands and what to do. One may think that the introduction of a belief vector relaxes this assumption, as this variable encodes whole that there is to know about past history in a local, time-contingent variable. However, there are two important differences that will still hold between MDPs and POMDPs: first, the agent may still need memory in a POMDP [158]. The reason for that is that the observation at $t-1$ may be important to interpret the observation at t. For instance, two states s_1 and s_2 might provide the same observation, but if the agent knows that it cannot stay in s_1 for more than one trial, then it needs memory to behave optimally in s_2. The second difference, which follows closely the first one, is that in a POMDP, *the optimal policy might be stochastic* (which contrasts with the Bellman optimality equation, see below).

Another example is a Markov (or Stochastic) Game [159], in which the agent is opposed to one or more agent with a different goal and which interacts also with the same environment. From a global point of view, this can still be modelled as a MDP, but from a local point of view (i.e. of a single agent), the state is not fully observable and therefore the system is not Markovian.

Finally, in Semi-Markovian Decision Processes (Semi-MDP) [160] the transition of state s to s' with the action a occur after some delay $d \sim p(d \mid s,a)$. As such, Semi-MDPs generalize MDPs by framing it in a continuous time domain.

All of these three models reflect some truth about the surrounding world of animals, and therefore they should not be overlooked. First, animals do not have a full and immediate access to the necessary and sufficient information to determine the state they are in. This information requires time to be collected and

integrated, and this process is almost never complete. Second, animals interplay with other agents that try to maximize their own gain, and therefore affect the world in which they evolve. Finally, actions may have an effect that has some long term effect whose timing cannot be predicted accurately.

The learner (which may happen to be also an agent) has to solve the problem of maximizing the return by selecting the best *policy* in the environment. This policy can be deterministic or stochastic. As the former case is just a degenerate case of the latter, we will adopt the nomenclature $a \sim \pi(a \mid s)$

This learner may or may not know the perfect structure of the MDP. If the knowledge is perfect, Dynamic Programming (Section 1.2.1.2) can be used to compute *exact value functions* of the policy which will dictate the behaviour (i.e. the policy) of the agent. In most useful circumstances, the knowledge of the environment is incomplete (some elements of the MDP tuple are not directly accessible), and the learner will need to rely on some strategy to find the way to the maximum return.

Box 1.2.2| RL vs (un)supervised learning

At the very base of RL lies the assumption that rewards will drive the learning process. In the words of Richard Sutton [161]:

> The use of a reward signal to formalize the idea of a goal is one of the most distinctive features of reinforcement learning.

This feature distinguishes RL from supervised learning, as there is no label provided by the user to compute some kind of error of prediction. Specifically, in RL, what the correct answer is is not only hidden, but also unknown. This feature might recall unsupervised learning: however, it is not a form of unsupervised learning either, as we do not want to generate patterns or cluster unlabelled data following some assumption, but we want to maximize an clearly defined objective which is always ultimately expressible in terms of reward.

Our agenda will be the following: We will distinguish value based learning methods (Section 1.2.1) (either Model-Free or Model-Based) from policy-based learning methods (Section 1.2.2). In the former category, we will first introduce some key concepts of RL from the machine learning point of view, starting from offline value-based learning methods (Dynamic Programming, Monte Carlo methods, Section 1.2.1.2), that we will contrast with online methods of learning (Section 1.2.1.3). These last category will be central for us, as a crucial aspect of animal behaviour is their capacity to learn while performing a task. Finally, we will shortly introduce Bayesian RL (Section 1.2.3) before

giving a small introduction to the use and theories of RL in psychology and animal behaviours (Section 1.2.4).

1.2.1 Value-Based RL

One strategy to learn how to maximize the reward in a task is to learn a probability distribution of the long term returns (conditioned on γ) for any initial state. In bandit tasks, an equal knowledge of the posterior probability of each value distribution can provide an asymptotically optimal policy to the agent [162, 163, 164]. For simplicity, early approaches of value-based RL focused mainly on learning the first moment (average) of the reward for the events of interest[11]. This value is conditioned on the current policy that the agent has: it is therefore not difficult to see that such measure provides the agent with an objective to optimize wrt the policy in order to maximize the long-term return. In other words, if the policy value is v_π (with π conventionally indicating the policy of interest), then the optimal policy is given by $\pi^* = \arg\max_\pi v_\pi$. It should be clear in the formulation we have chosen that what the agent is actually trying to do is to *approximate* some value function (e.g. v_{π^*}), assumed to exist in the external environment, with another learned value function which is learned from experience (e.g. V_π).

There are essentially two types of values that can be computed: state values (conventionally indicated by $v_\pi(s)$), which can be interpreted as the distribution of rewards that follow the event of reaching some state of the environment, and action value ($q_\pi(s, a)$), which decouple the states and the actions available. We will see that the state value is simply the marginal probability of the action value conditioned on the policy.

1.2.1.1 The Bellman equation

We now detail how a *state value* can be computed in a discrete MDP given a perfect knowledge of this process. Let $s \in \mathcal{S}$ be a state of the process, and $a \in \mathcal{A}_s$ be an action available in this state. We fist detail the intuition behind the use of a discount factor γ.

1.2.1.1.1 Discounted return Imagine an MDP where the reward function of each state-action pair is non-negative and where no absorbing state exist. In this case, the long-term return associated with each state (and each action) is infinite, and therefore state-action values will be hard to compare. Another issue is that there is no guarantee

[11] This can be viewed as a frequentist approach, which would contrasts with the Bayesian approach where we would be more interested in the distribution (and therefore in the uncertainty) of the long run return of some policy

that the environment will remain stable during the transitions from state to state: the MDP might suddenly or progressively change in the future. Therefore, an agent might wrongly discard a reward situated in a near future to favor a distant and higher reward that will not be delivered because of a contingency change. A final issue is that a reward might be more beneficial in a near future: for instance, an animal cannot remain in a state of hunger for too long even if this strategy has a high long-term pay off. Therefore, the weight of distant rewards has to be inferior than close rewards. To reflect this, the decay factor γ is usually introduced in the value function:

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right]$$
$$q_\pi(s,a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right] \qquad \text{with } r_{t+k+1} = \mathcal{R}(s_{t+k+1}, a_{t+k+1}).$$

It is easy to see that, as mentioned before, $v_\pi(s) = \mathbb{E}_\pi \left[q_\pi(s,a) \right]$. Importantly, $\gamma = 1$ can only be encountered in episodic tasks, as otherwise the value may be infinite. On the other extreme, a value of $\gamma = 0$ will make the agent myopic to future rewards.

1.2.1.1.2 The Bellman equation for state value gives an analytical form to the above expression:

$$v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left(r(s,a) + \gamma v_\pi(s') \right). \tag{1.34}$$

The Bellman equation requires us to have a perfect knowledge of the environment: is is assumed that the agent has access to the true state values, transition matrix, reward functions and policy. Obviously, most of the time none of these quantities is known exactly, and our aim will be to investigate how we can efficiently approximate this quantity with a limited cost. The main advantage of the Bellman equation is that it expresses how the current state value is related to the next; in this way, we can recursively estimate value functions.

Interestingly, from there we can derive a rule that dictates the best policy given a complete knowledge of the environment.

1.2.1.1.3 The Bellman optimality equation is given by the reformulation of the optimal policy:

$$\begin{aligned} v_*(s) &= \max_{\pi} v_\pi(s) \\ &= \max_{a} q_*(s,a) \\ &= \max_{a} \sum_{r,s'} p(s', r \mid s, a) \left(r(s,a) + \gamma v_*(s') \right). \end{aligned} \tag{1.35}$$

Most of the value-based RL literature gravitates around the Bellman equations. In the next sections we will first explore how this equation can be used to select the best policy offline. Then, we will see how value function and policy can be improved together in an online fashion.

1.2.1.2 Offline methods: Dynamic programming and Monte Carlo methods

Consider an agent that knows both the MDP transition matrix and how much each state-action pair is rewarded in a stationary environment. Its aim is to learn the optimal policy, which requires the knowledge of the state value functions (Equation (1.35)). These two values depend on each other: in value-based RL, the policy requires by definition the knowledge of the value (or a proxy to the true value) of each state under this policy. This might seem like an unsolvable problem: selecting the best policy requires the knowledge of the best state-value function associated with it, which is by definition unknown if the best policy is itself unknown. Fortunately, there is a way to sequentially improve one's policy given a specific state value function: this is known as Dynamic Programming (DP).

1.2.1.2.1 Policy-improvement theorem A common operation in RL is to assess how do two policies compare. It can be shown that if for all $s \in \mathcal{S}$,

$$v_\pi(s) \leq q_\pi(s, \pi'(s)) \tag{1.36}$$

for two policies π and π', then π' is at least as good as π. This tells us that for a given set of value functions of a policy π, we can test whether π' outperforms π by comparing the state-values with the state-action values under the alternative policy π'. The advantage of this formula is that it shows us that we can use a suboptimal policy to evaluate an alternative policy: therefore, we could iteratively learn v_π, and then improve $\pi \rightarrow \pi'$ given v_π. This is the principle under the policy-iteration algorithm [165] in dynamic programming. This algorithm alternates between policy evaluation and policy improvement on the basis of the policy-improvement theorem. Value iteration packs

together these two steps and updates each state-value independently and sequentially by refining both the state value and the decision rule for this state. Algorithms 3 and 4 compare these two approaches. Note that both of them are guaranteed to converge with a worst case time complexity that is exponentially lower than linear search.

Box 1.2.3| DP policy learning

Algorithm 3: Policy iteration.

Data: States $s \in \mathcal{S}$, Actions $a \in \mathcal{A}$
input: Transitions $\mathcal{T}(s'|s,a)$, Reward function $\mathcal{R}(s,a)$
Result: Optimal policy estimate $\pi^* \approx \pi$
Initialize $\pi, v_\pi(s)$;
repeat
  Update $V(s)$ for all $s \in \mathcal{S}$ given π;
  Compute improved policy π' with policy-improvement theorem Equation (1.36);
  Set $\pi \leftarrow \pi'$;
until *Some convergence criterion is met*;

Algorithm 4: Value iteration

Data: States $s \in \mathcal{S}$, Actions $a \in \mathcal{A}$
input: Transitions $\mathcal{T}(s'|s,a)$, Reward function $\mathcal{R}(s,a)$
Result: Optimal policy estimate $\pi^* \approx \pi$
Initialize $\pi, v_\pi(s)$;
repeat
  Update $V(s)$ for all $s \in \mathcal{S}$ st $V(s) = \max_a p(s', r|s,a)\left(r(s,a) + \gamma v_\pi(s')\right)$;
until *Some convergence criterion is met*;

Both algorithms enjoy the same asymptotic convergence property, namely that:

$$\lim_{i\to\infty} \pi = \pi^* \qquad \lim_{i\to\infty} V(s) = v_{\pi^*}(s).$$

If the requirements stated above are met (namely perfect knowledge of the MDP and stationarity), DP is an efficient strategy to solve the problem of finding the optimal policy. An example of problem where DP could be used is for instance to find the best road (where "best" could be defined in term of speed, distance, cost etc.) from A to B given that we have complete knowledge of the environment (of the routes that could be taken from A to B). From a biological point of view, DP makes little sense: we can hardly think of a situation where an animal has a complete knowledge of its surrounding environment before having explored it.

Monte Carlo (MC) methods tackle the problem of optimizing one's policy and value functions given an incomplete knowledge of the environment, i.e. when neither the model of the environment nor the reward functions are known in advance. Consequently, MC requires us to be able to generate sequences of events in the environment: these sequences can be either simulated or generated according to a real experience: for instance, a robot might use MC to learn the optimal policy by evolving in the environment or by using simulations of what would happen if it were evolving in this environment. However,

we still say that this method works offline because inside an episode, no update of the policy is done.

In MC learning, some K episodes are generated with a policy π_0, then the returns of each state-action are averaged and the assigned to the q_{π_1}-values, then a second set of episodes is generated with a new policy π_1 which follows the values of q_{π_1}. After that, the process is repeated with the computation of q_{π_2} to computer π_2 etc. until some convergence criterion is reached.

Roughly speaking, MC methods can be of two types: the first type is called *on-policy* learning, and it consists in applying the same policy to learn and to maximize the reward in the environment. As we will see with online methods, on-policy has the advantage that we are maximizing rewards at the same time that we are learning the policy. *Off-policy* methods contrast with these by their division between a target policy, which is the policy that will be used once learning is achieved, and the behavior policy, which is the policy used to explore the environment. The advantage here is that we can optimize exploration first, and once a sufficiently good knowledge of the optimal policy and behavior is reached, we can apply the target policy that was learned in the background.

In brief, on-policy MC control works by repetitively generating episodes with a given policy, then averaging and using the long-term returns for each state-action pair as a proxy to the true value of these states under the policy of interest, before updating the policy based on these values. Then, the process starts over with a new episode etc. Importantly, the policy has to be *soft*: given that we use the policy both for exploration and exploitation, we will need to explore apparently suboptimal choices from time to time in order to assess their long-term efficiency. Therefore, each action needs to have a non-null probability of being chosen.

In off-policy MC control, the policy used for learning and the target policy are uncoupled. This scheme recalls the importance sampling simulation algorithm: a proposal distribution (the behavior policy) $b(a \mid s)$ is used to generate samples whose importance is corrected based on the ratio between the likelihood of these samples given the target distribution (the target policy) $\pi(a \mid s)$ and the likelihood of these samples given the proposal. Given this observation, we can implement a policy learning algorithm that repetitively generates episodes according to b, then updates the values with an importance sampling scheme, starting at the end of the episode and going backward to the first state. When convergence is reached, the optimal policy is simply given by selecting the action with the highest value in each state.

Box 1.2.4| On the advantages of bootstrapping

Aside from their requirements, the most crucial difference between DP and MC is the fact that the first uses an estimate of the successor state value to improve the current state value function estimate, whereas the second does not. This feature of improving an estimate using an estimate (learning a guess from a guess) is usually referred to as *bootstrapping*. To see why this is a good idea, we can see that, without bootstrapping, a whole episode is needed to compute the state(-action) value functions under the current policy. This means that the agent does not need a full episode to learn from experience: seeing that an action leads to a suboptimal state is an information *per se*, and going through the whole episode might not be necessary to learn from this action. Also, in certain situations, a final absorbing episode is rare or even not defined. In this case, the notion of episode itself is not well defined. Finally, the cost of storing all the events in an episode can be high, and the computation peak after the episode may be prohibitively expensive. Bootstrapping dispatches the computation time to all events in an episode, which makes better use of the resources. Usually, bootstrapping methods are empirically found to converge faster than Monte-Carlo methods, although there is no known theoretical proof of this observation. In the context of biological agents, it makes much more sense to think about bootstrapping as a way to improve (or repair) knowledge of the environment, as it is unlikely that an animal will wait until the end of an episode – whatever its definition – to correct a suboptimal policy.

1.2.1.3 Online methods

In the last section, we have sketched some ideas of learning an optimal policy in stable environments where the agent had either (a) a complete knowledge of the environment and where no experience was required to learn the policy (DP), or (b) an incomplete knowledge of the environment and where learning was achieved by generating episodes of events with some policy, averaging the values of each state-action pair during the episodes and learning the optimal policy from these (MC). In many cases, it is more beneficial to learn during the evolution in the environment: for instance, if an initial policy is highly damaging, the agent has a great advantage in using it as little as possible. We will detail two approaches that will be of crucial importance to us: *model-free RL*, in which the agent discards the transition model and focuses in learning the value function

of each action, and *model-based RL*, in which the agent uses a knowledge (either true or approximate) of the transition table to reach the best policy[12].

1.2.1.3.1 Model-Free RL: Temporal-Difference learning In MC-RL, the value function had to be approximated given the samples that were drawn for a given policy. We first look at the way an agent can evaluate its policy online. The Bellman equation for state value Equation (1.34) gave us a closed-form formula to compute the value of each state given the reward received in this state and the next state value (weighted by the probability of selecting each action and the probability that this action would lead in this state). Here, we are interested in deriving an update equation of the state value given a single sample of the reward and an imprecise estimate of the next state value. Moreover, we do not have access to the state-to-state transition probability. Temporal difference (TD) [166] learning algorithms provide a way to learn an approximate state value $V(s_t)$ at each step by averaging the return $r(s,a) + \gamma V(s_{t+1})$ of each state-action pair:

$$V(s_t) = V(s_t) + \alpha \left(r(s,a) + \gamma V(s_{t+1}) - V(s_t) \right) \tag{1.37}$$

where $0 < \alpha \leq 1$ is a learning rate. Given that actions are selected following the policy π, if neither the policy nor the environment change and if α satisfies the Robbins-Monro conditions, it can be shown using the law of large numbers that the values computed according to Equation (1.37) will approach their true value for $t \to \infty$. It is interesting to contrast TD learning methods to MC methods: MC required the episode to be finished to learn from them, whereas TD can learn at each time step (it bootstraps). The corollary of this is also that TD does not require us to store the entire episode to perform updates, whereas MC does and, moreover, it has a high peak computation that TD methods do not have.

Box 1.2.5| Eligibility trace

In the TD equation presented above, only the last state value is updated according to the current observation of the reward. However, it might be useful to iterate through each state and update its value based on the current estimate if this state preceded the occurrence of this reward. TD(λ) achieves this by keeping track of an eligibility trace $\boldsymbol{\epsilon} = \{ \epsilon_i \}_{i=1}^{S}$ for each state with a decay λ, and updating the

[12] Strictly speaking, in the previous section, DP was a form of model-based RL and MC was a form of model-free RL, but we decided to arbitrarily restrict this denomination to online methods, as it is the use that is made in cognitive neuroscience.

state-values as follows:

Algorithm 5: TD(λ)

input: Initial state s_0, γ, λ

Result: Approximate state value V_π

initialization;
Set $\boldsymbol{\epsilon} = 0$;
Set $V(s) = 0$ for all s;
Set $k = 0$;
repeat
  Select $a_k \in \mathcal{A}$ with policy π;
  Observe $r_{k+1}(s, a)$ and s_{k+1};
  ϵ_{s_k} += 1;
  for $s \in \mathcal{S}$ **do**
    $V(s) = V(s) + \alpha \left(r(s_k, a_k) + \gamma V(s_{k+1}) - V(s_k)\right) \epsilon_s$;
  end
  $\boldsymbol{\epsilon} = \lambda\gamma\boldsymbol{\epsilon}$;
  k += 1;
until *ever*;

In this way, each observation not only impacts the last state value, but also the whole set of previous states encountered. Note that the TD framework outlined before is a special case (namely TD(0)) of this algorithm. Similarly, MC control is equivalent to an offline version of the TD(1) algorithm.

The next stage is to understand how we can use Equation (1.37) to not only evaluate, but also improve the policy online. We have seen in Section 1.2.1.2 that policy-iteration provides an asymptotically accurate policy. Recall that this algorithms alternated between policy evaluation and policy improvement: TD algorithms for policy learning build on a similar idea. As we have seen for MC-RL methods, this can be done in an *on-policy* or *off-policy* manner. Here, instead of learning state value functions, we are interested in learning state-action value functions, as we would like to be able to guide the policy towards actions that lead to a greater return.

The *on-policy* TD learning is known as **SARSA** [167], for the State-Action-Reward-State-Action tuple appearing in the update equation:

$$Q(s, a) = Q(s, a) + \alpha \underbrace{\left(r(s, a) + \gamma Q(s', a') - Q(s, a)\right)}_{\text{RPE}}.$$

The Q-learning Reward Prediction Error (RPE) can be understood as the signal that

drives learning, similar to an unbiased sample of the value function gradient giving the update direction to follow to reach optimality. We see that, here, learning and control are coupled: if a policy is highly exploratory for some reason, then the updates will be negatively impacted with respect to an off-policy strategy. This is not the case of the **Q-learning** algorithm [168], which is the prototype of an *off-policy* algorithm:

$$Q(s,a) = Q(s,a) + \alpha \left(r(s,a) + \gamma \max_a Q(s',a') - Q(s,a) \right).$$

It is easy to see that the above update is done without recurring to the current policy of the agent. As such, Q-learning tends to over-estimate the return of each state-action pair, as an uncertain (low-sample) Q value in the next state will positively impact the current update, whereas SARSA is more conservative. Consider for instance an agent improving her policy by trial and error: a good strategy would probably be to favor exploration over exploitation in the early stages of learning. Q-learning will then take each early optimal estimate at their face value, which may wrongly guide the agent to consistently select a suboptimal action in the future. SARSA is more likely to avoid this problem: exploration-based updates will prevent this kind of phenomenon.

Finally, assuming that an agent has a direct access to her policy, the update of the SARSA algorithm can be improved by considering the weighted average of the next return:

$$Q(s,a) = Q(s,a) + \alpha \left(r(s,a) + \gamma \sum_{a'} \pi(a' \mid s') Q(s',a') - Q(s,a) \right).$$

which is known as the Expected SARSA algorithm.

1.2.1.3.2 Model-Based RL The idea behind model-based RL is that planning ahead the impact of one's actions on the environment might help to improve her policy. The idea here is that there is one aspect of the environment which is marginalized by TD methods, which is the transition from state to state that follows each action (this transition is intended as a "change" in the environment, not in the sense that it modifies the underlying structure but in the sense that it produces a transition to another state that can be predicted).

We distinguish two ways to use planning in RL: in the first, planning is used directly for control. At the decision step, the agent simulates the evolution of the environment in the next steps and selects the action leading to the maximum reward. In the second, planning is used to look ahead during learning. This is the idea of the Dyna family of algorithms [169].

1.2.1.3.2.1 Planning at decision time If the MDP is very large, and if the transitions can be efficiently predicted, then it might be difficult to learn each state action value functions. Visiting of a state may occur very rarely, and using a separate cache for each of them can constitute a loss of memory. In these cases, simulating the model behaviour at decision time is likely to be beneficial. For instance, most board games have a very large number of possible configuration possible, and the chance that one configuration will be encountered more than once can be quite unlikely. In their original flavour, algorithms that use planning at decision time do not store state-(action) value functions, but they approximate it online. As a consequence, they only require the agent to learn a transition table and a reward function.

Once more, tree search can be used in two distinct ways: first, a brute force exploration of the possible outcomes of each action can be achieved by simulating randomly sequences of actions up to a predefined final state. At each branch of the decision tree, all the actions are explored many times, and the action leading to the maximum average return is selected. Finally, the action with the maximum value is selected.

Another strategy, named *rollout*, consists in simulating multiple sequences of events with an alternative policy (the rollout policy). This policy can be a fixed, naive policy, or the previous policy of the agent. In this case, the policy improvement theorem states that the new policy π' adopted by using simulations with the previous policy π will be at least equally good, if not better than π.

1.2.1.3.2.2 Planning as a surrogate to real world experience Both model-free-RL and model-based RL have their virtue, and several algorithms have been developed to use the best of both worlds. The Dyna family we have mentioned above is an example of such strategies [169]. With Dyna-Q, the agent still learns a state-action value, but each time its experience is enriched by the new observations (transition and reward), the Q-value table is updated in two successive steps: first, the Q-value is updated as in classical Q-learning, and the model of the environment $M(s, a)$ (which is aimed at learning the transition matrix $\mathcal{T}$) is updated with the observed transition $t_{s' \leftarrow s,a} \sim \mathcal{T}(s' \mid s, a)$. Then, in a second step, the Q-values of the entire process are updated many times by simulating the behaviour of the environment given the new observations using simulation. In this way, the agent can account for rare events (or contingency changes) that are distant for a specific state, because this state will be explored using simulation between two consecutive steps.

Performing these updates on each and every state-action pair can be prohibitively expensive. Moreover, the vanilla Dyna-Q algorithm does not account for the fact that a new observation (either transition or reward) will have an unequal impact on the rest

of the MDP. More specifically, a large update will especially impact the states that are likely to precede the current one. Following this, it becomes clear that working backward from the current state will lead the agent to update preferentially the sates that might be significantly impacted because of the new observation.

Prioritized sweeping [170] implements this strategy. After gathering a new observation, the agent creates a queue of the state-action pairs that were more impacted by the new observation. If a state-action of the queue is processed, then its value is updated. In this queue, priority is given to pairs that were the most impacted: this implies that, for a finite number of iterations, only the most highly ranked state-actions will be simulated. The corollary is obviously that, if the observation has a mitigated effect on some state-action pair, then this node and the preceding ones will be ignored by the algorithm.

Of course, learning the model of the environment comes with a cost: first, it increases the degree of freedom of the model. Whereas the model-free RL algorithms outlined above learn only the state-action values, the model-based RL aims at learning the transitions too, which can be either very numerous and unpredictable. In the latter case (e.g. $\mathcal{T}(s,a) \approx \mathcal{M}(1/S)$ for all or some $s \in \mathcal{S}$), some action may have a higher value than other, but learning the transition will simply be a waste of resources, or worse, will mislead the agent for some actions if the sample size is low and/or the number of actions available A is large: in other words, a model-based model will overfit. The other issue with model-based RL is its computational cost: planning can be expensive, and in cases where learning has to be achieved online, the learning or decision step (depending on where planning is used) may require a consequent amount of time and resources to produce a reliable action selection policy.

However, model-based RL has also many advantages: if the agent learns that there is a change in either the reward function of a distant state-action pair, or if a transition has changed, then she will be able to select the actions accordingly without needing to go through each chain of event that may lead to this change of contingency.

Dyna-Q has the advantage that it mitigate the cost of the decision step by using simulation in the background: decision is still based on the learned action value and the action value only. However, planning is still used to learn the action value behind the scenes. As such, Dyna-Q reduces the computational cost of the decision process and reduce the reward cost of Model-Free decision making.

Box 1.2.6| Intermediate model: the Successor Representation

The successor representation [171] is a clever mixture of model-free and model-based RL. Consider an agent that learns each average state-action reward function (and not value), and concomitantly learns the long-term probability of visiting all

the other states of the environment using a TD algorithm. The idea here is that the evaluation of a state-value function and transition matrix can be alleviated if we can learn the probability of visiting each of the future states. Let us expand and rearrange the Bellman equation for state value:

$$
\begin{aligned}
q_\pi(s,a) &= \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{T}(s' \mid s,a) \times \\
&\qquad \sum_{a'} \pi(a' \mid s') \left(\mathcal{R}(s',a') + \gamma \sum_{s''} \mathcal{T}(s'' \mid s',a') \times \dots \right) \\
&= \mathcal{R}(s,a) + \sum_{i=1}^{S} \sum_{j=1}^{A} c_\pi^{s,a}(s_i,a_j) \mathcal{R}(s_i,a_j) \\
\text{where } c_\pi^{s,a}(s_i,a_j) &= \gamma \Bigg(\mathcal{T}(s_i \mid s,a)\pi(a_j \mid s_i) + \\
&\qquad \gamma \left(\sum_{s'} \mathcal{T}(s' \mid s,a) \sum_{a'} \pi(a' \mid s') \mathcal{T}(s_i \mid s',a') \pi(a_j \mid s_j) + \dots \right) \Bigg)
\end{aligned}
$$

is the geometric series of the probabilities of selecting a_j in s_j in the future. Of course, this expression can be marginalized in form of state values. As noted above, c can be computed using a TD algorithm, which leaves $\mathcal{R}$ to be the only remaining quantity to estimate.

With classical TD learning, we have seen that either a change in the reward function or the transition table was hard to cope with, because changing the value function of a state-action upstream of the change requires the whole sequence of events leading to it to be explored. On the contrary, with model-based RL, either changes can be coped with easily. With the successor representation, a change in the reward function can be learned readily, but a change in transition will require $c_\pi^{s,a}$ to be adapted, and, again, this will require a re-exploration of the events upstream. Therefore, from the point of view of an observer, these three strategies can be discriminated by testing the sensitivity of a learner to these various changes. This is the approach was proposed as a putative mechanism of learning and decision making by [172][a] and adopted by [174] to investigate each strategy in humans. As expected, these authors show that participants indeed demonstrated a lower sensitivity in changes of reward functions than changes in the transition table, which suggests that an intermediate strategy between model-free and model-based RL might be used by human subjects.

[a] Stachenfeld [173] also shows that the hippocampus is well suited for the encoding of the successor representation.

1.2.2 Policy gradient methods

Another sounded method for solving an MDP would be to optimize directly the policy, without recurring the problem of optimizing an approximation to the Bellman optimality equation.

Ideally, given a parametric policy $\pi_\theta(a \mid s)$, we would like to be able to derive a function $J(\theta)$ that would measure the performance of our model wrt θ and that we could optimize without needing to optimize the value function $q_{\pi_\theta}(s, a)$. Fortunately, this can be achieved: the Policy gradient theorem [175] states that the gradient of the performance measure function $J(\theta) \triangleq \mathbb{E}_{s_0 \sim \mathcal{D}}\left[v_{\pi_\theta}(s_0)\right]$ enjoys the property that

$$\nabla_\theta \left\{J(\theta)\right\} = \sum_{s \in \mathcal{S}} d_{\pi_\theta}(s) \sum_{a \in \mathcal{A}} \nabla_\theta \left\{\pi_\theta(a \mid s)\right\} q_{\pi_\theta}(s, a)$$

where $d_{\pi_\theta}(s) = \lim_{t \to \infty} p(s_t = s \mid \mathcal{D}, \pi_\theta)$ is the sum of the probabilities of encountering the state s in an infinite episode. As required, this expression does not require us to compute the state-action value function q but only to observe returns $G_t = \sum_{k=0}^{\infty} \gamma^{k+t} \mathcal{R}_{k+t}(s_{k+t}, a_{k+t})$, because, observing that $\mathbb{E}_{\pi_\theta}\left[G_t \mid s_t, a_t\right] = q_{\pi_\theta}(s_t, a_t)$ and therefore, after some manipulations, we have

$$\nabla_\theta \left\{J(\theta)\right\} \approx G_t \nabla_\theta \left\{\log \pi_\theta(a_t | s_t)\right\} \tag{1.38}$$

which is an unbiased estimate of the policy gradient. This equation has an intuitive interpretation: during optimization, we should move in directions of θ where the product of the decision gradient and the return is high. If the decision gradient is positive (negative) and the return negative (positive), then there is a mismatch between the two and we should go away of this solution.

REINFORCE is the archetype of policy gradient method [176]. The main shortcoming of this technique is the large variance of the unbiased gradient proposed, which can seriously threaten the convergence of the algorithm. The introduction of an action-independent baseline $b(s_t)$ can help to reduce this variance [176] (REINFORCE with baseline). This value can, for instance, be an estimate of the state value: we can expect in this case that each action return will be situated above or below this value, thereby reducing the variance of the gradient estimate without introducing any bias, but in practice it has been shown that the choice of the average long-term reward was not the best choice in term of variance reduction [177, 178, 179]. Also, as we have seen in the section treating of VI Section 1.1.2.2, the use of a natural [180] (or reparameterized [181]) gradient can greatly improve the convergence of the SGD optimization procedure. The use of Bayesian Monte Carlo introduced in Box 1.1.1 can be used to this aim [182], which

has the double advantage of providing a more accurate estimate of the gradient than the simple Monte Carlo method and giving a free estimate of the Fisher Information matrix for the computation of the natural gradient.

Policy-gradient methods such as the one used in REINFORCE require full episodes to be optimized. As we have seen in Box 1.2.4, this may not be an optimal way to behave, in the sense that it is in general more convenient to use bootstrapping methods (which do not require complete episodes). The next section couples online value learning with policy gradient methods to solve this issue.

1.2.2.1 Actor-Critic methods

The introduction of policy gradient methods is mainly for us the occasion to introduce the Actor-Critic framework [183, 166, 175, 184, 185] as an intermediate method between value based and policy based methods. These approaches are particularity useful when the state-action space is large, and when tabular functions are unreliable. In this case, function approximations (where $v_\pi(s)$ is replaced by a parametric function $v_\pi(s;\phi)$) are a logical choice, but TD learning algorithms often fail to converge in such settings.

The central idea of actor-critic algorithms is quite simple given what we have introduced above: a bootstrapping method, such as TD(λ) is used to compute a state value for the current state visited and reduce the variance of the policy gradient method. The parametric policy is the *actor*, whose role is to achieve an *control*, and the TD learning (or more generally, the algorithm used to learn from the environment) is the *critic*, whose role is to guide the actor by *predicting* future outcomes. In a TD(0) scheme for instance, the value of G_t is then estimated according to

$$G_t = \mathcal{R}_t(s_t, a_t) + \gamma v_{\pi_\theta}(s_{t+1}) - v_{\pi_\theta}(s)$$

which can be efficiently plugged in the gradient estimate given by Equation (1.38).

In Chapter 4, we will study a particular form of actor-critic, where the actor uses some adaptive parametric policy function that takes as input the state-action values that are provided to it by the critic. We adopted the terminology of actor-critic as we disentangle clearly a model of *learning* and a model of *agency* in this framework, although this could be arguably considered as a misnomer given the definition of actor-critic methods provided in this section.

1.2.3 Bayesian Reinforcement Learning

So far, we have focused on learning a point estimate of the policy or MDP structure in order to maximize the return of an episode in the environment. This is analogous to the frequentist approach in statistics, and this observation motivate us to consider a Bayesian alternative to this scheme. As always, a Bayesian approach would provide us a full estimate of the posterior probability of some specific MDP configuration under the observations made so far and the prior knowledge or belief that we had about the environment. A crucial feature here is that the knowledge of the uncertainty of the environment can enforce an optimal exploration/exploitation balance: as opposed to a frequentist agent, a Bayesian agent will know how much it knows, how secure it is about its model.

1.2.3.1 Model-Free Bayesian Reinforcement Learning

Once more, we must distinguish the cases where the agent learns a model of the environment from those in which it does not: with model-free Bayesian RL, the agent keeps track of the distribution of the state value functions during learning, and makes decisions based on these. Application of this learning scheme requires us to make some assumptions about the environment. More concretely, we need to specify the distribution of the reward or the return (which is unknown) and put a prior on this distribution. The learning problem then becomes to infer the corresponding posterior distribution. We can consider that the agent assumes that the reward is normally distributed, and uses a conjugate Normal-Inverse-Gamma prior over this distribution. For binary rewards in Bandit tasks, a Beta prior over a Bernoulli distribution can be used.

In [162], it is shown that in the case of a simple bandit task the posterior updates have a closed-form formula when the reward distributions are independent. In the case of MDPs, posterior computation is more complex as the state-action posterior values are correlated: the return of a policy in the Bellman equation involves the value of the next state, and therefore the posteriors of the state value function are not independent. For simplicity, in [162] this correlation is broken (which recalls somehow the mean-field assumption seen above). This ensure that the posterior can be computed or approximated with a distribution of the form of the conjugate prior of the distribution of $\mathcal{R}$. As we have seen, the Mean-Field assumption tends to underestimate the posterior variance, and the same feature can be observed in our context: through learning, the agent becomes confident too quickly in the posterior distribution of the value functions, and

therefore misses opportunities to explore the environment (see Thompson sampling below). Dearden [162] suggests that exponential forgetting could be used to prevent this phenomenon: this is also the strategy we will be using in Chapter 4.

1.2.3.1.1 Thompson Sampling Fortunately, when we have an equal knowledge of each action in a certain state, the posterior probability of the Q-values can dictate a policy that will balance exploration and exploitation. A commonly used method is simply to choose actions with the posterior predictive probability that these actions are better than others. This strategy, sometimes referred to as probability matching [186], Q-sampling [162], or Thompson sampling [187, 164] algorithm, is guaranteed to converge but is not, generally speaking, optimal.

Thompson sampling in RL can be formally expressed as

$$\begin{aligned} p(a_{t+k}) &= p\left(a_{t+k} = \arg\max_a \sum_{k=0}^{K} \mathbb{E}\left[\mathcal{R}_{t+k}(s_{t+k}, a)\right]\right) \\ &= p\left(\mathbb{E}\left[q(s,a)\right] > \mathbb{E}\left[q(s,a')\right], \forall a' \neq a\right). \end{aligned} \tag{1.39}$$

This strategy is compelling in many cases, and it has successfully been used in investment decisions, marketing [188], game control [189], advertising [190] and it is a standard decision making policy in almost every modern high-tech company [191, 192, 193].

To see why this may not be optimal, we will consider two cases: first, consider the case where there may be a very informative action to sample from in a near future. This will not appear in the action selection policy dictated by Equation (1.39), and therefore the agent might miss this opportunity. In other words, the basic version of Thompson sampling without planning outlined here can give poor results in terms of exploration in MDPs [163] (i.e. unlike VPI, Thompson sampling is myopic [194, 195]). Therefore, it is more suited for multi-armed Bandit tasks.

It should also be emphasized that Thompson sampling will only work when we can consider that the environment is stable: if the environment is changing, or when playing against an adversary, other strategies have to be considered (e.g. [196]).

Another feature that this algorithm does not take into account is the fact that the amount of information for the various available actions might not be identical: the agent would then gain in computing how much information it could get from selecting one action, and optimize the sum of this action and the information expected when selecting it. This is the approach developed in [162] and [197], and detailed in Section 4.5.5. Another approach could consist in selecting the action that minimizes the entropy of

the Bayesian policy [186]. Finally, if the aim is to accumulate the maximum reward in a definite duration, exploration could harm the overall performance of the agent: this problem of optimizing the reward rate is explored in [198]. In practice, it has been found that Thompson sampling tends to over-explore [194].

Box 1.2.7| Probability matching

In psychology, Thompson sampling recalls a phenomenon known as *probability matching* [199, 200, 201, 202] which is essentially the observation that even after an exploration phase, participants in an experiment involving a probabilistic outcome tend to select the best option with a probability that match the probability that this option will be rewarded. This contrasts with the best policy dictated by the Bellman Optimality equation which states that the best policy is to select the most rewarded action. Many authors have been surprised by this observation, that somehow demonstrates that choices in stochastic environment seem to be irrational. Shanks and colleagues [201] show that this behaviour, which may seem apparently maladaptive when seen from a frequentist point of view, is actually rational when some factors are taken in consideration, such as training length, sufficiently high financial incentives etc.: in practice, the optimal action tends to be selected more and more often if certain experimental criteria are met. In this perspective, probability matching can be seen as an instance of Thompson sampling. This is an extremely interesting feature of animal behavior that suggest once more that the human brain makes inference in a Bayesian way.

1.2.3.1.2 Amortized inference using GP: from TD to policy gradient methods Recent advances in Bayesian model-free RL have focused on the problem of extending this framework to large problems. The approach proposed above can be generalized to large or infinite state-spaces by the use of a non-parametric model, the Gaussian Process Temporal Difference learning (GPTD) [203], in which a Gaussian Process (GP) is used as a prior for the value functions. Because of their computational and space cost of $\mathcal{O}(T^3)$ and $\mathcal{O}(T^2)$ respectively, Engel and colleagues note that the use of GP might seem ill-fitted for RL problems, and they propose to sparsify the problem by recurring to the basis function (or feature vector) product $\phi(x)\phi(x')^T$ rather than to the kernel $k(x, x')$ [204, 22] (where x is some continuous state), which can provide accurate online approximations of the true posterior covariance matrix. This framework is then extended to a parametric setting in [205].

Note that policy-gradient algorithms too can be used in a Bayesian model-free setting [182] and generalized to Actor-Critic methods [206, 207].

Box 1.2.8| Model-Based Bayesian Reinforcement Learning

Learning a model in a Bayesian RL setting is *per se* not a difficult task, but the use of this information for control is extremely complex. For this reason, and because it is beyond the scope of this thesis, we only briefly outline the problem that Bayesian model-based RL represents. First, in the continuation of their development of the Bayesian Q-Learning algorithm, Dearden and colleagues [208] propose a series of solutions to account for the uncertainty about the model when solving an MDP online. Let us assume that an agent is learning some distribution over the state-action values and transitions given a policy. The distribution of the former depends tightly on the distribution of the latter, and the entire distribution of the q-values will need to be updated following each observation of a transition. This problem motivates these authors to adopt the following general framework: K samples are drawn from the reward and transition distributions, and each of these MDPs is solved independently. It is easy to see that this approach provides a posterior sample of the MDP distribution with a high computational cost. Some solutions to this problem are proposed: for instance, each sample can be corrected following a new observation using Prioritized Sweeping (see above).

Duff [209] develops another method still in vogue today [210] that consists in considering that the couple of the known state and the unknown transition matrix defines a "hyper-state" which can only be partially observed. This recalls the problem of maximizing the return in a POMDP, and can be solved as such. This is a topic we do not consider fully in this thesis.

Another interesting treatment of model-based Bayesian RL is given by Furmston and Barber [211], who use VI and EP to solve an MDP in an original manner. Briefly, for observed transitions $\mathcal{D}$ and latent transition distribution parameters $\boldsymbol{\theta}$, their approach is the following: by restricting the reward considered to be non-negative, these authors define a *reward-weighted path distribution*

$$\widehat{p}(s_{1:t}, a_{1:t}, t, \boldsymbol{\theta} \mid \pi) = \frac{\mathcal{R}_t(s_t, a_t) p(s_{1:t}, a_{1:t} \mid \boldsymbol{\theta}, \pi) p(\boldsymbol{\theta} \mid \mathcal{D})}{G(\pi \mid \mathcal{D})}$$

$$\text{where } G(\pi \mid \mathcal{D}) = \int G(\pi \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) \text{ is the policy utility under the observations}$$

$$\text{and } G(\pi \mid \boldsymbol{\theta}) = \sum_{t=1}^{T} \sum_{s_t, a_t} \mathcal{R}_t(s_t, a_t) p(s_t, a_t \mid \pi) \text{ is the policy utility under } \boldsymbol{\theta}.$$

As always, it is evident that the aim of the agent is to maximize $G(\pi \mid \mathcal{D})$, and VI can be used to obtain a lower-bound on the policy utility given an approximate posterior reward-weighted path distribution. Specifically, we can define an

approximate posterior

$$\begin{aligned}\widehat{p}(s_{1:t}, a_{1:t}, t, \boldsymbol{\theta} \mid \pi) &\approx q(s_{1:t}, a_{1:t}, t, \boldsymbol{\theta}) \\ &\overset{\text{MF}}{\approx} q(s_{1:t}, a_{1:t}, t) q(\boldsymbol{\theta})\end{aligned}$$

where MF stands for the mean-field assumption. Note that $q(s_{1:t}, a_{1:t}, t)$ indicates an approximation of the posterior reward-weighted path, and not a state-action value like before. This distribution could directly dictate the policy using Thompson sampling. The authors found that the mean field assumption in this framework impaired learning by overweighting initial time steps, a flaw that was not shared by its relative EP framework.

1.2.4 Reinforcement Learning in the Animal Brain

The history of the research in the psychology of learning and RL are so close that it can be difficult to separate them clearly. Sutton and Barto [212] identify two main threads in the developments of RL: optimal control and Dynamic Programming, which followed the works of Bellman [213, 214] and conditioning on trial-and-error. In psychology, this is known as *instrumental learning*, by opposition to Pavlovian conditioning in which the animal learns contingencies without interacting with the environment, although considering these two learning schemes as distinct is too restrictive to explain several experimental observations [215].

The term "reinforcement" was originally used to refer to a phenomenon previously known as Thorndike's "Law of effect" [216]. It is indeed a quite old observation that animals trained to obtain a pleasant outcome from an action tend to learn this association, and the strength of this learning is directly *proportional to the training length*[13]. This can be easily tested by removing the reward after the training, and testing how long it takes to the animal to adapt to the new contingency Section 1.2.4.

This basic observation is at the very center of this thesis. From a frequentist (point-estimate) point of view, it makes little sense that when the behaviour of the animal is stable (which is supposed to happens after a few trials) the behaviour keeps being reinforced. The Q-learning update rules we have seen above do not predict a different learning rate during training and extinction, even though this is what is observed in experimental settings. In Chapters 3 and 4, we will try to provide a new Bayesian learning framework where an agent learns the confidence it has in the stability of its

[13] Note that other factors such as the interval between the executions of actions [217] can predict the sensitivity to contingency changes.

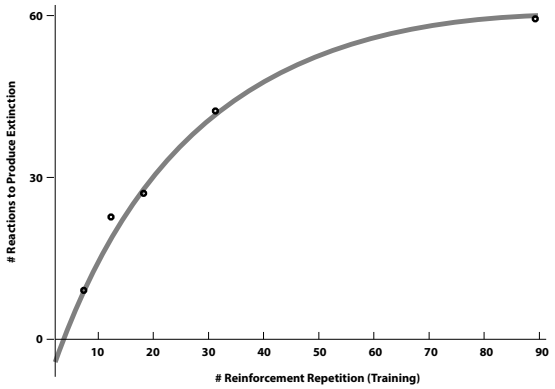


FIGURE 1.8: Empirical learning curve plotted in terms of resistance to experimental extinction (i.e. reward retrieval). The circles represent the mean number of unreinforced bar-pressing reactions evoked by the conditioned stimulus in different groups of hungry albino rats after varying numbers of food reinforcements. The curved line represents a positive growth function which has been fitted to the data represented by the circles. (Adapted from Hull 1943 [18])

observations (in terms of change of contingency): this will allow it to adapt its forgetting and learning rate to the likelihood that a change of contingency has occurred.

In the following paragraphs, we will introduce RL in the context of animal behaviour, relate it to the artificial intelligence/machine learning perspective that we have given to the problem so far and to the theories of the Bayesian Brain outlined in the first section of this introduction.

Although Artificial intelligence is now well separated from cognitive neuroscience, the work of mapping animal behaviour to the construction of artificial learning algorithms (e.g. [218]) is now sometimes inverted (e.g. [219]) or replaced by a joint effort in building bridges between the two (e.g. [220, 221, 198, 222]).

1.2.4.1 The early days: Model-Free in the brain

The early attempts to explain animal behaviour using Reinforcement learning were made in a Model-Free setting. The study of habits finds its roots in the works of Dickinson [223, 224, 225] and Adams [226, 217] who showed that a behaviour which is initially *goal-directed* becomes with training autonomous: it is not triggered by a need or will to reach a profitable goal but rather by a habitual, somehow procedural action selection pattern that ignores the consequences of the actions that are undertaken. These works are seminal, even if they were merely based on behavioural observations of rat behaviour in various specific contingencies, and it brings a qualitative (and not only quantitative) distinction between goal-directed and habitual behaviour, which is not only slower but also more myopic to the outcome of actions.

One possible explanation for the switch between goal-directed to habitual behaviour is that animals keep a cognitive map of the environment [227] – putatively thought as relying on prefrontal regions of the cortex [228] – and then learn to act in a procedural way with a basal ganglia (BG) driven behaviour [229, 230]. This view of a transfer of agency from cortical to subcortical structures is, however, too simplistic: BG are necessary for learning and memory [231, 232] even in early stages of skill acquisition, and cortical structures are highly involved in the execution of automatic behaviours [233]. It seems therefore more reasonable to see the striatum as a structure mediating the learning (rather than the execution) of both goal-directed (response-outcome) and habitual (stimulus-response) behaviours, which would sequentially involve the caudate (or associative striatum), and then putamen (or sensorimotor striatum) respectively [229]. This mirrors the idea that learning switches from response-outcome to stimulus-response learning which both are driven by specific regions of the BG [234].

Several experimental observations confirm this view: for instance, it has been observed that dopamine has a diminishing influence on the *expression* of repetitive behaviours through learning. Moreover, execution of habitual behaviours seem also to rely on cortical regions [233] (sensorimotor [235] motor [236, 237, 238] or infralimbic cortex [239]). An alternative and more extreme view would be that the BG support and train cortico-cortical learning of automatic behaviours [240].

We have seen that learning model-free policies can be achieved through TD learning (Paragraph 1.2.1.3.1). In practice, it is observed that if a reward is delivered following an unconditioned stimulus (UCS) that follows a conditioned stimulus (CS), then the reinforced signal of the UCS *transfers* to the CS through learning. Importantly, this observation of the transfer of activity is noted to be mediated at least partially by the ventral striatum [241, 242], to which projects dopamine neurons from the ventral tegmental area and substentia nigra which are thought to report the RPE induced by the received reward or punishment [243, 244, 245].

1.2.4.1.1 Exploration/Exploitation in a model-free setting Another question to be answered is to understand how the brain does trade exploration and exploitation. Actor-critic methods offer a compelling answer to this question, as they implement a stochastic policy which is refined through learning [246]. The assumption that the brain uses a ramp-to-threshold mechanism to make decisions provides the opportunity to adjust the level of stochasticity of the action selection process with a certain decision threshold that determines the amount of evidence necessary to secure a decision (Section 1.3). For instance, Frank and colleagues found that decision making in a RL task was well modelled by a Drift-diffusion model (DDM) with the difference of actions

modelling the accumulation of evidence in favour of each of the options [11]. In this framework, the decisions are based on a parametric model (the DDM) that adjusts its parameters to the current contingency: for instance, the threshold whose value is well reflected by the activity of the subthalamic nucleus. Also, the drift rate, assumed to reflect the difference in value of the various options, was shown to be modulated by caudate activity. As such, this is a form of actor-critic RL model: control and learning interplay to optimize the decision policy. The actor can "decide" to make more exploratory choices when the environment is less certain [247, 222, 248]. This will motivate our Actor-Critic model in Chapter 4.

1.2.4.2 Model-Based vs. Model-Free Reinforcement Learning

There exists a whole constellations of opposite concepts that relate to the distinction between habitual and goal-directed behaviours [249], such as stimulus-response vs. planned behaviours [232], type I vs type II [250, 251], procedural (or automatic) vs. declarative [252, 233, 253, 254, 255] etc.

In a seminal paper, Daw, Niv and Dayan [256] proposed that the balance between goal-directed and habitual behaviour was algorithmically equivalent to a bi-controller agent that had to arbitrate between a model-based (tree-search) and a model-free (cache) controller. Following Daw and colleagues, this arbitration is supposed to result from the comparison of the two model uncertainties[14]: this is roughly equivalent to say that each model is selected proportionally to its *model evidence*, which relates to the concept of surprise minimization seen above Section 1.1.3. Importantly, uncertainty has to be understood as uncertainty of moments: in the exponential family for instance, it is clear that a stronger knowledge of the environment narrows down the width of the posterior distribution of the model parameters, therefore reducing the variance of the moments estimates. Consequently, the variance of the reward (i.e. the risk [256] or the expected uncertainty [257]) should not impact the arbitration if this risk is well known, which is the case when the model evidence is high: in other words, a model is certain when it is reliable. Nevertheless, the concept of uncertainty is quite ill-defined [258], as it is sometimes understood as *sensory* understanding [150] or, in terms of learning, in as a (frequentist) average prediction error [259] or as a (Bayesian) model evidence [260].

The distinction between model-based and model-free RL in human behaviour motivates us to question the classical view that a habitual behaviour is simply defined as a behaviour insensitive to contingency changes. To solve this issue, we will in the present

[14] The two aspects of this theory, namely the model-free / model-based opposition and the uncertainty driven arbitration mechanism have lead to a tremendous amount of publications in the past decades, reaching more than 1400 citations in 2018.

work distinguish two non-exclusive features of habitual behaviour[15]: the first feature is what we will call *inflexibility*, which is negatively defined as the propensity of an individual to keep acting in a maladaptive way after a change of contingency even if this failure is made explicit. Inflexibility could also be positively defined as a trust in the environment stability.

The second is the *model-free* nature of some behaviours, which can be understood as the observation that some actions seem to reflect a lack of planning[16]. We name a behaviour *habitual* if it exhibits one of these two features.

For instance, in the framework we propose, a model-based (i.e. planned) behaviour can be *inflexible* (and therefore habitual or automatic [262]) if the agent is not able to account for an observed change in the environment transition table. Similarly, a model-free agent can be more flexible than another (in a sense that will be made explicit short after) if it adapts faster to a change of contingency, assuming that both agents start at an equal learning stage.

Even though a behaviour can be a mixture of model-free and model-based control, we acknowledge that these are two *qualitatively* different aspects of the behaviour. Unlike for the model-based / model-free distinction, there is no clear qualitative duality that can characterize an inflexible behaviour: on the contrary, there is a continuum between flexible and inflexible policies.

The advantage of this distinction is that it can flexibly account for the observation that behaviour can be more *model-free* in early stages of learning and more *model-based* later on [263, 262][17], without contracting the view that overtraining leads to a lack of flexibility [223, 225]. The learning and decision making scheme proposed in Chapters 3 and 4 is also compatible with the observation that habit learning and skills learning are distinct features [265]: in our scheme, an agent learns latent causes of the observations and related policies (either model-based or model-free) separately from the stability of the environment. If the environment is considered as highly stable, skills may still be improved.

[15] We loosely define habitual behaviour as any behaviour that is highly based on the history of reinforced state-action pairs and less on current observations of the environment.

[16] The use we have made of the word *flexibility* is merely a semantic statement: for instance Daw and colleagues [256] use the word inflexibility to characterize the lack of planning characteristic of model-free controller. Note that the authors themselves acknowledge the distinction between habits (viewed from a behavioural point of view) and model-free RL (the algorithmic pendant of this behaviour) [261].

[17] Note that from a Bayesian point of view, this observation is sound: at early stages, model-based has theoretically no knowledge of the transition table, and a model-free controller can have a lower uncertainty about its predictions. This balance will depend on the task: some may require more model-based in early stages, some others no model-based learning at all. See the work of Kool and colleagues [264] for an account of conditions where model-based should be preferred over model-free control.

1.2.4.3 Planning and model-based control

Planning has long been known to be an essential feature of human behaviour relying on evolutionary advanced brain regions such as the frontal lobe [228]. As model-free learning required neural substrates for learning and execution, so does the model-based controller.

In its frequentist formulation, online learning of the transition table in model-based RL contrasts with model-free RL by the expected occurrence of state-prediction error, which recalls the RPE of model-free RL. This has lead scientists to look for a signature of model-based learning, which was found in the lateral prefrontal cortex [266], suggesting that model-based learning is evolutionary more recent [267] than its model-free counterpart [268]. At decision time, model-based decisions can be shown to activate brain regions associated with the expected outcome [259] and to require the hippocampal activation [269, 270]. A final question to address is the mechanism by which the controller are weighted: the dorsolateral prefrontal cortex, seems to be involved in the balance between the two controllers [271] but also in value comparison between goal-directed actions [272].

1.2.4.4 Uncertainty and cognitive cost of actions

1.2.4.4.1 Uncertainty for Arbitration Intuitively, the uncertainty based arbitration mechanism outlined by [256] recalls Bayesian Model Comparison, selection [35, 273] and Averaging [32]. This is the idea proposed in [260], where the model selection framework is mapped onto the theory of active inference.

Box 1.2.9| Using lower bounds to compare models

Using the ELBO to compare models can be very tempting: as noted in Section 1.1.3, the Free Energy has many interesting formulations, one of which is the difference between the model accuracy and the model complexity, which may seem to be an optimal measure of the model optimality. There is, however, a catch: the ELBO does not indicate the true log-model evidence, but a proxy to this value. If the KL between the true posterior and the approximation is high for one model and not the other, we may severely overestimate or underestimate a model optimality. In fact, model comparison using VI may appear to be theoretically unsound. Moreover, the value of the ELBO is split between a likelihood term and a complexity term, and it may be unclear which one is the most optimized when reaching a solution, leading sometimes to non-complex but poorly generalizable solutions [123]. Finally, in the context of AVI and IWAE, [119] show that a tighter bound does not necessarily give a better estimate of the latent variable

distribution, giving the intuition that the ELBO is far from being the ultimate measure of model accuracy.

Fortunately, a few observations can soften this claim: for instance, it has been shown that using the ELBO empirically outperforms classical approximations such as the Akaike and Bayesian Information Criteria (AIC and BIC respectively) [137]. Importantly, the above discussion is biased by the fact that we have implicitly assumed that the model evidence is the utmost measure of a model predictive capacity. This is not the case for the following reasons: first, when using model evidence $p(\mathcal{D} \mid M)$, we are still using a form of (although refined) maximum likelihood, as we cannot marginalize over the space of models. Indeed, the true quantity we would like to assess is $p(M \mid \mathcal{D})$, but this is not only intractable, it is theoretically impossible to estimate (see Section 1.1). Next, the model evidence is still dependent on the prior choice, and changing the prior value can drastically change the model evidence, and so even for a weakly informative prior. It could be argued that the prior is part of the model, and that this is the whole point of Bayesian statistics, but again, model comparison will rely on a maximum likelihood estimate (of the prior setting) and will therefore not be reliable in a Bayesian sense, as the risk of overfitting will persist. In [274], it is suggested that an approximate IS-based Leave-One-Out (LOO) cross-validation procedure could be used to achieve a better comparison of models (see [275, 276, 277] for reviews and comparison with other methods). This can be implemented using Pareto Smoothed Importance Sampling [56]. When generalized to VI [278], this technique also gives a way to assess the quality of the lower-bound (i.e. the variational KL divergence), thereby solving the main problems outlined in this section. In summary, one should prefer model comparison procedures that assess how generalizable is the model to new data, rather than merely looking at the marginal likelihood or to a proxy to this value.

Because it is standard in neuroscience to use the ELBO as a surrogate to the model optimality, we will follow this assumption in this work, but one should always keep in mind that this approach is biased and hardy justifiable in practice from a theoretical point of view, as the ultimate measure of model accuracy should be based on its posterior predictive capacity. More research is unquestionably needed to assess if the brain does or does not use a more refined model comparison method than the classical one to select the best strategy.

The use of VI (and Bayesian inference in general) as a tool to measure model reliability ties together several theoretical, experimental and rational aspects of habit learning: first, as we have seen so far, models should be weighted according to their accuracy,

but not only: this balance should somehow include a form of computational complexity [255]. Simpler model should indeed be favoured if they compromise well complexity with efficiency. This relates to the central idea of the study of automatic behaviours that automatization helps animals to perform multi-tasking [279, 280]. Finally, the use of such models provides a principled framework for model averaging in which it is assumed that the brain uses a mixture of models to predict the occurrence of future events [281], a mechanism that is reflected by observations of brain activity [282]. Following this line of thought, we can see the model-based controller as a training mechanism for the model-free controller: when the latter has reached a sufficient predictive accuracy, the control is preferentially allocated to it: this could be viewed somehow as the online version of the Dyna algorithm [283] where the process of habitization [225] is understood as a "training by and switching from" process involving the model-free and model-based controllers.

1.2.4.4.2 Cognitive cost of actions The view that model-based RL trains the model-free controllers mainly explains how the brain ensures a sufficient accuracy of the learning process, but not really how it deals with the difference of cost between the two strategies. For instance, Pezzulo and colleagues [284] propose a mixed controller that can flexibly balance model-based and model-free decisions by weighting the information gain of a model-based simulation with its cost.

Several observations and models support the idea that multiple competing controllers interact during learning with the aim of lowering the cognitive cost of decisions. In terms of RL for instance, goal-directed behaviour seems to be impaired in large state space [285], when the outcome involves a probabilistic feedback [286, 287, 288] or when the rules are difficult to verbalize (explicit vs implicit instructions for instance) [289, 290]. This suggest that actions with a higher cognitive cost are preferentially assigned to controllers that exhibit a low complexity.

Otto and colleagues show that participants with a higher working memory capacity were less sensitive to stress [291] and working memory load [292] when making decisions in a simple MDP than those with a lower capacity. It seems therefore reasonable to postulate that model-based and model-free RL systems are weighted with respect to their cost and benefits, an assertion that is supported by behavioural evidence [293, 294]. Collins and Frank [295] have suggested that working memory alone might constitute a controller that would compete with RL control (essentially model-free in their framework) to make decisions, showing that the use of TD learning can lower the cognitive cost of learning

and decision making[18]. Keramati and colleagues [198] proposed a decision model that is aimed at arbitrating between model-based (tree search) decision making, which is slow but accurate under optimal circumstances, and model-free decision making. Their contribution is essentially that model-free can, in some instances, maximize the reward rate: This work suggests that animals tend to make decisions that optimize the amount of reward per time unit rather than the overall return: a suboptimal decision dictated by the model-free controller can be more efficient on the long-run, as it can provide small rewards more frequently and at a lower cost.

The concept of maximizing the reward rate also recalls the actor-critic view of decision making: a stochastic accumulator such as the DDM [307] is able to trade speed and accuracy of decisions, which ultimately can optimize the number of decisions made - and hence the rewards received – per unit of time [308, 309].

Box 1.2.10| Single controller hypothesis

Arbitration between the two strategies is a problem *per se* that has received an extensive amount of attention [198, 221, 310, 264, 285]. Not only the brain is supposed to arbitrate between actions, but in the initial paradigm proposed by Daw and colleagues, it also needs to arbitrate between models [256]. The dual model-free / model-based controller predicts in its initial flavour that both strategies are continuously used to make predictions about the return of each action – or at least that both are learned throughout the task – but this is obviously not optimal from a computational point of view. Indeed, if model-based RL is not necessary to make decisions, keeping the system "active" when the model-free has reached an acceptable performance will hamper the utility of using a model-free controller. Another difficulty when thinking about these structures as working in parallel is that the same neural structures (namely the midbrain dopamine neurons and the striatum) [202, 283, 311] and the same neurotransmitter (namely dopamine) [312, 282, 313] appear to be involved in learning both strategies. These observations motivate more hierarchical or nested models of RL in the brain.
Moreover, it has been repeatedly observed that during goal-directed choices the hippocampus and the ventral striatum perform mental simulations of the events to come to make a decision [314, 315, 316, 317, 318], which is known in the rat literature as vicarious trial and error [227]. The same process has been shown in

[18]Generally, the role of working and episodic memory in RL is still a matter of debate [296, 297, 298], especially given the role that the hippocampus plays in model-based learning [299, 269]. For instance, replay-driven learning of the Dyna algorithm [169] recalls the offline replay of hippocampal cells [300, 301, 302] which supports the idea that a similar phenomenon might occur in the brain [303, 304, 305, 306]. The wake-sleep algorithm (Section 1.1.3 and [152]) builds on the same idea: the brain learns a generative model from sensory experience, then optimizes its recognition model by simulating real life experience.

decisions involving model-based planning in humans [269, 259]. However, this activity vanishes with training [314], a phenomenon hardly explained by the vanilla version of the competitor model, which are supposed to work in parallel. Pezzulo and colleagues [284] propose a model of decision making where the agent decides whether or not it should use simulation to refine its estimates of the state-action return. The arbitration is based on the value of information, implemented as the ratio between the uncertainty of the Q value and the absolute difference in expected return for the two actions considered. A high value of information indicates that choosing the most valuable strategy is uncertain, and that simulation could be used to refine the estimates of the returns. This algorithm is roughly similar to a Dyna algorithm, except that simulation is executed in full (over the whole episode), in part or not at all for high, moderate and low value of information respectively.

Another unified algorithm has been proposed by Keramati and colleagues [319] where the model-based controller prunes the decision tree and then relies on the model-free controller in a "plan-until-habit" manner to optimize speed accuracy trade-off. By modulating the time pressure, these authors were able to assess the depth of the tree search by fitting a computational model to the data. The fit of this model is consistent with the hypothesis that pruning occurs earlier when the time pressure is high than when it is low.

A final set of example is provided by theories of hierarchical control [320, 321, 322]. Cushman and colleagues [323] propose that the goal-directed controller is subordinate to the habitual controller. In this framework, the model-free system learns the desirability of the outcome, and instructs the model-based system which then plan the decision tree to reach this outcome. In contrast to this idea, Dezfouli and al. [324, 325, 326] suggest that a goal-directed learner might benefit from concatenating actions into action sequences to lower the cost of decision making. This is in accordance with the observation that the same structures in the BG are involved in habitization and sequence learning [231]. In this context, the lack of sensitivity to a change of contingency would be explained by the fact that sequences of actions need to be broken down to adapt to the new observations. It is important to emphasize that this process of habitization predicts that the first action should be the least subject to a lack of flexibility, as it plays the role of an observation-contingent trigger for a whole action sequence that will be (at least partially) unconditioned on the following observations. On the contrary, the model-based / model-free dual control theory [256] predicts that actions situated earlier in the chain of events are the more habitual, because the model-based uncertainty correlates positively with the length of the chain of events that

can follow from a decision, penalizing it with respect to the model-free learner uncertainty, which is roughly insensitive to the distance from the absorbing state. This latter hypothesis links the problem of habitization to the sensory uncertainty that arises from the noise in perception. In general, it might be more accurate to consider that animals evolve in a POMDP (and not an MDP) environment: as such, they need memory and time to build a belief about the state where they stand [327, 328, 145]. If evolving in a POMDP, where model-based has to be preferred [329], the agent may use its memory and knowledge of estimated state-action transition table to refine, correct or substitute its sensory estimate of the state it has just reached [315]: this process biases perception itself [330]. In other words, in a POMDP neither a pure model-based nor a pure model-free reinforcement learning can be used [49], as states need to be inferred from an imperfect knowledge of the past states. As a consequence, if a prior aggregated evidence for a state-to-state transition biases decision making in a SSM sense [331] (i.e. bias the initial position of the accumulator), then a model-free controller can be used to accumulate evidence given the current state belief. It is easy to see that this kind of mechanisms could account for the formation of action sequences, especially under high time pressure where the state cannot be observed exactly (see Box 4.2.2). This is compatible with the observation that chunked action sequences take a long time to be loaded (model-based step) before being executed rapidly (model-free step) [332] and that time pressure facilitates the activation of the striatum [333].

1.3 Sequential Sampling Algorithms

Maintaining and updating a model of the world with a certain confidence is crucial for survival. In the first two sections, we have seen that learning and inference are two central features of the Bayesian brain, and that related machine learning tools can be used to model neural processes at the computational, algorithmic and implementation levels [334]. The common requirements of these paradigms is that they ultimately rely on the capacity to make a decision, either between actions and/or models. In this section, we will focus on biologically driven neurocomputational models of decision making.

To make decisions, the brain does not only react passively to inputs: it builds one or several models of the world [335, 336, 337], compare these models, infer possible causes and generates possible observations from these causes. A simple way to observe the requirement of a model is to contrast the poverty of sensory inputs on the retina with the richness of the representations that we consciously perceive. These models guide behaviour by comparing their predictions with the observations.

By essence, a goal-directed decision requires us to deliberate between options, which is formally equivalent to decide what possible cause lies behind the observation that is made and/or to model the possible consequences of the actions available: in other words, inference about the world precedes action. This raises the question of how we can compare models or predictions given some observations, which can be either external (exteroceptive) or internal (interoceptive, memory-driven or simulated). In both cases, we will assume that the animal who is making decision acquires these observations sequentially, as the full representation of the state considered is complex and takes time to be integrated in the neural circuit. Indeed, decisions are based on noisy observations or estimates of the environment state. Yet, animals must make decisions under time constraints.

Take for instance an eagle, observing an isolated prey. This eagle must decide what would be the best strategy to catch this prey: it needs a predictive model of the prey's behaviour, because the prey itself is moving, probably trying to avoid the predator. Various hypotheses about the prey's behaviour might conflict, and each of the new observations will favour some but not all of these possibilities. It needs also to be fast: if the deliberation lasts too long, the prey might escape.

How can an agent solve the problem of selecting a model $p_1(x)$ or a model $p_2(x)$ [19] given a stream of observations $\mathbf{x} = \{ x_i \}_{i=1}^{N}$? Let

$$Z_i = \sum_{i=1}^{N} \log \frac{p_1(x_i)}{p_2(x_i)}$$

be the logarithm of the likelihood ratio of the two models. If $\mathbf{x}$ is Gaussian distributed, so is $\mathbf{Z}$ [5]. Moreover, if each of the elements of $\mathbf{x}$ is itself a large finite set, then the central limit theorem holds and $\mathbf{Z}$ is generally normally distributed (or χ^2 distributed if the assumptions of Wilk's theorem hold [338]). Bitzer and colleagues [145] show how this relates to a Bayesian-optimal policy[20]. Each time a positive $\mathbf{z}_i$ is obtained, it brings evidence in favour of the first model of the alternative, and each time it is negative it brings evidence to the other model. One strategy could be to stop at j if the sum of $x_{\leq j}$ is greater or equal to some fixed threshold ζ: this is known as a sequential probability ratio test (SPRT). Following Wald's theorem [340], this process is optimal in the sense that no other strategy will be on average faster than this one for a predefined error rate. Such decision scheme is central to the theory of decision making in both neuroscience and statistics, and many models and experimental evidence echo to it [341, 342, 308, 343, 344, 345, 346, 347, 348].

Box 1.3.1| SPRT as a Bayesian inference test

Interestingly, for sensory evidence integration, the work achieved by the accumulator can be represented with a loop that takes the previous posterior of the cause m_1, $p(m_1 \mid x_{t-1})$ as a prior at time t to perform inference about m_1 at time t given the current observation x_t [145]. In [349], the cortico-basal ganglia-thalamus circuit is postulated as a representing the neural implementation of this process.

1.3.1 The Drift-Diffusion Model

If $\mathbf{z}$ is observed at infinitesimal time interval, the process outlined here converges to a continuous random walk with stochastic differential equation given by

$$dZ_t = \xi_t dt + \varsigma_t dW_t$$

where ξ_t is the (time-dependant) drift rate, ς_t is a noise coefficient and W_t is some Gaussian noise (the diffusion coefficient). As before, the process stops when it crosses

[19] Model selection is intended here in its broadest meaning: for instance, discriminating a leftward or rightward motion in a random dot moving task can be viewed as selection the model predicting the left or rightward motion.

[20] For Wald and Wolfowitz [339], a Bayes Optimal decision function is a policy that minimizes the risk of making an erroneous decision given some prior and some observations.

a boundary b_t distant from the other boundary $\neg b_t$ by a threshold ζ_t[21]. We denote the set of parameters of the model $\boldsymbol{\omega}$. Any model of decision making that uses a similar accumulation of evidence Z_t is an instance of sequential sampling models (SSM) [5]. These processes have the compelling advantage that, unlike other evidence based decision policies such as the softmax (Boltzmann) decision rule, they can model both choices and RT density given some latent neurophysiological cause: as a consequence, inference can be achieved and refined using data in two dimensions rather than one. Because the data is more complete, more degrees of freedom can be added to the model. Fortunately, for some instances of SSM, the first passage time density (defined as the probability density of a particle being absorbed at the upper boundary at a given time) $p(\mathrm{t}, b \mid \boldsymbol{\omega})$, and/or cumulative density $P(\mathrm{t}, b \mid \boldsymbol{\omega}) = \int_0^{\mathrm{t}} p(\tau, b \mid \boldsymbol{\omega}) d\tau$ have a closed-form formula [350, 1, 351, 352, 353, 354].

Box 1.3.2| Optimizing SSM without using a first passage time density formula

If this density does not have a closed-form formula, there is still hope to find a solution to the problem. For time-dependant drift values, Ditterich [355] uses a an approximation based on the assumption that the boundary that is reached and the time at which it is reached are independent events ($p(\mathrm{t}, b) \approx p(\mathrm{t})p(b)$), but this obviously biases the estimates. For multiple-alternative decisions, Lette and Ratcliff [356] propose an approach based on the Nelder-Mead (or SIMPLEX) method [357] where each iteration involves the simulation of thousands of samples from a certain parameter estimate in order to match the simulated and observed histogram. The cost of such method is considerable and application to complex regression models can be difficult. Wagenmakers, Grasman and colleagues [358, 359] propose to use the closed-form formula (if it exists) of the mean and variance of the RTs and choices to find the most likely parameters configuration that generated them. This approach (the *EZ method*) ignores parameter variability, it does not handle well datasets with multiple conditions, it ignores the shape of the RT distribution and requires the first moments to have a closed-form and to be invertible. A critic of the EZ method is presented in [360]. Voss and Voss [361] propose *fast-dm*, a tool to fit the DDM using a numerical solution of the partial differential equation of the DDM, which results in a fast and generalizable algorithm for the evaluation of the CDF of the first passage time. Numerical solution of the PDE were also used by [362] to fit a diffusion process with choices involving multiple alternatives. In general, we have seen in 1.1.2.2 that implicit models can be optimized as long as data can be generated given a set of parameters, which

[21] fixing $\neg b_t = 0$ as a reference point and assuming that boundaries are fixed ensures that $b_t = \zeta_t$.

is the case of sequential sampling models.

More discussion about SSM optimization can be found in Section 2.1.

In Two-alternative forced choice (TAFC) tasks, a Wiener process (i.e. a diffusion process with constant drift $\xi \in \mathbb{R}$ and constant Gaussian noise $\varsigma \in \mathbb{R}^+$) with constant within trial threshold $\zeta \in \mathbb{R}^+$, is known as the Drift-Diffusion Model (DDM) [1]. Two more parameters are usually considered: a relative start-point (or initial bias) $z_0 \in (0, 1)$, with $z_0 = 1$ representing a process starting at the upper absorbing boundary and $z_0 = 0$ at the other, and a non-decision time τ. These five parameters define a non-metric model: changing ξ, ς and ζ by a constant positive factor does not change the predictions of the model. Therefore, one of these parameters needs to be fixed to give a metric to the model. Usually, ς is fixed to either 1 or 0.1.

1.3.1.1 Trial-by-trial variability in parameter values

A crucial aspect of human decision making is that errors and accurate responses[22] have usually very different timings. In stressful situations, and especially under high time pressure, errors tend to be faster than accurate responses. The opposite pattern is found when task difficulty is high and RTs are unconstrained. Explaining these behavioural patterns is hard with the vanilla version of the DDM, where the parameters generating the data are assumed to be fixed throughout the experiment: indeed, when responses are unbiased (i.e. when $z_0 = 0.5$), the resulting data will have an equal expected RT for errors and accurate responses.

1.3.1.1.1 Theoretical justifications of trial-to-trial variability in the DDM parameters An *ad-hoc* solution to this problem is to allow parameters to vary from trial to trial [363]. We thereby name *Simple* DDM the static version of the DDM, and *Full* DDM the version including trial-to-trial parameter variability. Usually, trial-to-trial variability is assumed to concern the drift, the starting point, the non-decision time but not the threshold value (see Section 2.1 for a discussion). First, variability in the drift rate [1, 364] can explain slow errors, because they allow the drift to be directed toward the wrong absorbing boundary, mostly with a low absolute value. Second, variability in the start-point [363, 365] can explain fast errors, because accumulation will start either

[22] Classically, it is assumed that the DDM is fitted to dataset where there is an accurate response and an error such as motion discrimination tasks. In this context, the drift should usually be oriented toward the accurate boundary. The concept of *accurate* response vanishes when working with RL models, but it is still interesting to introduce the requirement of a model in the context of perceptual tasks to generalize it to value-based decision making.

closer to the error boundary or closer to the accurate boundary, and so with an equal probability if the distribution of z_0 is centered on 0.5. The probability of the stochastic evidence accumulation to result in an error is higher in the former case (vicinity to the error boundary) than in the latter (vicinity to the accurate boundary). In this case, errors will be, on average, faster than accurate responses. These variations are usually thought to occur unpredictably. It is sometimes implicitly understood that between-trials variability of the parameters is introduced for practical purposes (i.e. to make the estimate of the mean drift and starting point unbiased), but we believe that this view does not render the full advantage of this feature. Indeed, trial-to-trial variations in the parameter values reflect the indubitable fluctuation of mental states during an experiment: variability in drift may for instance be caused by a random fluctuation in attention [366], whereas the variability in the starting point may be caused by a variable prior belief or by an anticipated accumulation of evidence [367]. Note that, once more, this idea relates to the concept that the brain optimizes inference as if it were evolving in a POMDP [368, 369, 370] where it gradually accumulates evidence for a perceptual state [371]: the initial bias somehow reflects evidence accumulation originating from memory only [372], whereas subsequent accumulation should reflect a mixture of both memory-driven and perception-driven evidence for a particular state-action optimality [373]. It makes then sense to assume that variability in the starting point predominates in experiments where time pressure is high, as the agent can then find a consequent advantage in biasing its decisions based on past experience, as the time to make a decision based on perceptual information is scarce.

The same kind of process, which bases the current perception on a prior-knowledge biased accumulation of evidence, relates to how animals implement selective attention [374] by considering the previous experience (predictions) as a prior (or a bias) for attention [375] and, hence, perception [376]. It also relates to the idea that animal sparsify the model of the world they built [377, 109, 378], by joining predictions from memory (belief states) with observations of the world [369, 370]. From a SPRT point of view, both these processes would appear to bias the SSM parameters (starting point for pre-trial inference, drift for within trial inference) from trial to trial based on some variable prior expectation.

In summary, the Full DDM is compatible with the idea that the brain achieves a form of sparse and fast Bayesian inference about the state where it stands and the cause of the observations [379, 380], that is conditioned on past experience in a predictive coding sense [375].

1.3.1.1.2 Identifiability of the variations Even if the parameters are fitted using a set of independent variables (such as instructions displayed on the screen, difficulty etc.), it is assumed that random fluctuations in these parameters (and also in the non-decision time) will still occur. Given what we have stated above, this appears to be a necessary assumption: neural activation patterns are by themselves probabilistic [381], and it is impossible to exactly predict inference processes occurring from trial to trial. Therefore, trial-wise parameters have to be considered as random variable, which follow a distribution that is conditioned on past trials and states of mind. Most authors suggest to marginalize the trial-by-trial values of the parameters given the prior $p(\boldsymbol{\omega} \mid \boldsymbol{\theta})$, and to fit the prior parameters $\boldsymbol{\theta}$ instead. This approach implicitly considers the parameters as independent and identically distributed (*i.i.d.*) given $\boldsymbol{\theta}$. At the same time, it makes the threshold variability unidentifiable.

1.3.1.1.2.1 Fitting the full DDM with the *i.i.d.* assumption Following this assumption, the classical approach to Full DDM fitting procedure was to find the maximizer of the marginal likelihood or some other loss, such as a Chi-squared[23] criterion [8]. This is equivalent to an empirical Bayes problem [29, 30].

Even in the *i.i.d.* setting, solving the DDM is not trivial (see [382] for a survey of methods), especially if one wants to maximize the marginal likelihood. First, the only known tractable integral in the marginal is achieved when the drift ξ is assumed to be normally distributed, leaving the problem of integrating out ζ and τ unresolved.

1.3.1.1.2.2 Relaxing the usual assumptions of the DDM The *i.i.d.* assumption is clearly too strong, and based on little or no theoretical and experimental ground. Next, the threshold variability is usually assumed to be equal to 0 for practical reasons (the model is usually thought to be unidentifiable if both the threshold and starting point variability are fitted [9]), but threshold variability should be considered to model random fluctuations in speed-accuracy trade-off, which is known to vary during task execution [383, 384, 11], a phenomenon that should differ from random prior bias in decision making (i.e. variability in starting point). It might be argued that variability in these parameters could be modelled with regression models: if a regressor is a good predictor of the value of a parameter, it could be used as a baseline for the trial-wise value of this parameter. These regressors could be experimental conditions (such as motion coherence in motion discrimination tasks) [9], neurophysiological measurements or behavioural effects [384, 11].

[23] More details can be found in Section 2.1 and section 2.1.1.

However, because regression model will not capture all the variability in drift and starting point [9], we can reasonably assume that it will not be the case for the threshold either. Moreover, the claim that including threshold variability in the model makes it unidentifiable in turn questions the identifiability of the full DDM itself [385]. Indeed, the second moments of the drift rate, starting point and non-decision time is usually considered as practically unidentifiable [8, 9], showing that these variables are introduced mainly for correcting the estimates of the respective first moments, whose fit is still usually unreliable [10]. In Chapter 2, a solution to these issues (*i.i.d.* assumption and threshold stability) is discussed, which consists in considering the data as cross-correlated in a forward manner.

1.3.1.2 Biological foundations of the DDM

A Bayesian use of the DDM can give an estimate of the posterior value of the model parameters, but this does not guarantee that the model is sound from a biological point of view. Fortunately, a whole literature supports the plausibility of the DDM and similar SSM [307]. Many observations of patterns of activity recalling those of the DDM have been observed in oculomotor-related areas. Surprisingly, more than twenty years have been needed to observe experimentally the ramp-like pattern of neuronal activation that was predicted by the neurocomputational models. Hanes and Schall [386] showed that (1) the random distribution of the RTs (of saccades) of a rhesus monkey was associated with a stochastic variability in the firing rate of neurons (thought to reflect the accumulation of evidence) situated in the frontal eye field and that (2) a thresholding effect linked the measured firing rate with the occurrence of voluntary movement. Using a motion discrimination task, Roitman and Shadlen [387] were able to link motion strength with drift rate and observed again a ramp-like activation pattern and a thresholding effect that mirrored the stochastic accumulation of evidence and decision making rule expected from theory. This observation has lead to some vivid debate in community [388, 389, 390, 391], some authors suggesting that the ramping pattern observed may originate from an effect of averaging a step signal over many trials. Nevertheless, SSM provide a good insight on the latent cause of behavioural observations, such as speed-accuracy trade-off, bias in decisions etc. which still relate to neurophysiological and behavioural observations (e.g. [392, 384, 393, 394, 370]).

Although it is theoretically grounded [395] from both a neuroeconomical and physiological point of view and some data seem to suggest that the hippocampus participates in an evidence accumulation process similar to the one observed in perceptual decision

making [346, 396, 397], gathering evidence for ramp-like activity at the cellular level for memory retrieval has been unsuccessful so far [398][24].

Box 1.3.3| Banburismus: where the machines once more meet the brain

Gold and Shadlen [341] show that there is a tight isomorphism between the theory of decision making in the brain and an algorithm for code breaking (named Banburism) developed by Alan Turing during the second world war to decode the German Enigma codes using Bombe machines, which can indisputably be considered as one of the most important ancestors of the modern computer. The SPRT outlined above was used to compare two models: one that predicted that two messages were encrypted with the same code, and one that predicted that they were not. The former hypothesis obviously predicts sequences of characters that are less random than the latter. Subtracting the log likelihood of a sequence of observations and summing these values up to a certain threshold gives rise to a decision process that is optimal in a Wald's theorem sense. This means that by using this technique, cracking the code was made as fast as it could for a given error rate.

It is interesting to note that once more, as we have seen with VI and RL, theories of human decision making tend to be inspired by – and perhaps inspire – machine learning methods.

1.3.1.2.1 Biological foundation of probabilistic (Bayesian) inference with SSM SSM are not only motivated by empirically linking a model of decision making to experimental observation of neural activity in the time preceding an action: they also find a grounded theoretical justification in the neurophysiological characteristics of neural firing and in the hierarchical organization of cortical structure. Here, we will briefly give an intuition of how neural response can code evidence integration, and how these two concepts relate to Bayesian inference (and especially variational inference).

It is a well known fact that neural response responses following a particular outcome is variable [399]. Intuitively, we understand that variability in firing rate must encode uncertainty about the feature that is observed: if neurons were to fire identically for two presentations of the same object, perception of this object should be exactly identical and hence perfect. Conversely, if there is noise in the perception process, there must be a variability in the firing rate. In this sense, the mean and variance of the firing rate of a

[24]Note that in the model proposed by Shadlen and Shahomy [395], no ramping pattern is assumed *in the hippocampus* itself. Instead, the mediotemporal cortex sends pieces of evidence to an cortical accumulator that integrates this information.

population of neurons encode a probability distribution, which can be roughly modelled by a Poisson distribution [400]:

$$p(\mathbf{r} \mid s) = \prod_{i=1}^{N} \frac{\exp\{-f_i(s)\}\, f_i(s)^{r_i}}{r_i!}$$

where $\mathbf{r} \triangleq \{ r_i \}_{i=1}^{N}$ is the population response to stimulus s and $f_i(s)$ is the tuning curve of neuron i to stimulus s. However, neurons do not respond specifically to a stimulus which is tried to be inferred, but to a set of features (contrast, color etc.) that are summed at the neuron and population level. If the distribution of the firing rate of a population is Poisson-like, this can be shown to be Bayes-optimal [381] (to give an optimal estimate of the posterior distribution of the stimulus given the features that are presented). The observation that this kind of dynamic models well neurophysiological and behavioural observations suggest that the brain performs probabilistic inference [401].

This idea is further developed and mapped to the problem of decision making by Beck and colleagues [381]. It is indeed not sufficient to make inference about a single observation of features in time, but one needs to accumulate evidence in favour of a hypothesis while defining a precise stopping criterion (i.e. error bound or decision threshold). These authors suggest that, by using the Poisson-like model of probability encoding, the following brain network can reach decisions in an optimal way (in the sense of Wald's optimality): in a saccade task, the Middle Temporal lobe of the monkeys encode the motion independently and with an inherent noise at each time point. Then, this region sends projections to the lateral intraparietal lobe, which integrates these pieces of evidence and accumulate them up to a certain threshold, before transferring this inferred preference to the superior colliculus which generates the motor command. This model is known as the linear Probabilistic Population Coding (PPC).

In a follow-up paper, Beck et al. [146] suggest that, by using the Mean-Field assumption, we can generalize linear PPC to a wider range of tasks, where conjugacy is not ensured anymore: this model takes the form of an online VB algorithm [402]. Pitkow and Angelaki [403] extend this framework to a view of the brain as a message-passing inference machine: as we have seen, under the mean-field assumption in VI, or using a localized approximate posterior in EP, information passing from one node to the other is achieved by transmitting summary statistics of the state of the nodes in the Markov blanket of the target node. The authors propose that the brain uses a similar strategy with a model of the world that can be represented as a directed graph [149]. This kind of structure is necessary for inference, as a fully connected network would require a number of neurons that would scale exponentially with the amount of features represented [376]:

by excluding some causalities, the brain trades sparsity over completeness (and overfitting) of the representations. This kind of message passing implies that information is passed not only in a bottom-up manner (from primary somatosensory receptors to higher cortical regions) but also – at least to some extent – top-down, which is at the basis of the predictive coding theory of brain functioning [404].

Overall, the beauty of this formulation of neural dynamics is that it ties together an elegant Bayesian formulation of inference in the brain in a simple mathematical form with a plausible and biologically grounded neural interpretation of the parameters of the model, which are just a set of scalars such as neural firing rate, neural gain and tuning curve.

Box 1.3.4| Working with unreliable inference units

A paradox that is hard to solve for the modern machine learning community is how can a deep learning model, which typically has much more parameters than training example, generalize so well to unseen data? Similarly, in biological agents, the number of units (neurons) that participate in inference sometimes greatly exceeds the number of elements that are used for learning: however, severe overfitting does not occur.
In deep learning, it is customary to use techniques (such as dropout) that ensure that the network is *redundant*: by sequentially hiding random parts of the network during training, we ensure that the network does not learn spurious patterns in the data and that random subsamples of the network can already perform a decent classification on their own. In the brain, the code is both distributed and multiplexed [405]: many neurons perform the same task (which makes sense, since each of them is unreliable *per se*) and each neuron is involved in many different tasks. This odd observation motivated Shannon [406] to state that

> We have, in human and animal brains, examples of very large and relatively reliable systems constructed from individual components, the neurons, which would appear to be anything but reliable, not only in individual operation but in fine details of interconnection. Furthermore, it is well known that under conditions of lesion, accident, disease and so on, the brain continues to function remarkably well even when large fractions of it are damaged.

However, this redundancy is somehow kept to a reasonable level, and the mature brain indeed prunes inactive cells and synapses when needed [407, 408].
Still, there is a notable contrast between artificial neural networks, for which we usually try to reduce the size to a minimum [409, 410, 411] and the brain, which

ensures some degree of compromise between redundancy and sparsity. Besides its resilience to network alterations, the structure of brain networks allows it to use the same network to treat a wide variety of input.
To our knowledge, only a few works in machine learning have treated the problem of balancing sparsification with generalization of a network (e.g. [412]).

1.3.1.3 DDM use in RL

The DDM was first proposed as a model of memory retrieval [1, 413]: in an item recognition task, several probes in memory have a respective evidence level that grows stochastically with time, while inhibiting each other. When a certain evidence threshold is crossed, the corresponding probe is chosen. Later, SSM have been mostly used for perceptual tasks, as a neural signature of evidence accumulation can be found in the brain. Other examples of the use of SSM are lexical decisions [414] and cognitive control [415]. However, value-based choices seem to involve evidence accumulation too [416, 395] with a mechanism that involves the dorsolateral prefrontal cortex and the anterior cingulate cortex for value comparison [417, 418, 419] and the tandem of the BG (ventral striatum) and the hippocampus [396] for value computation. This is compatible with the observation that value-based choices involve memory retrieval of recent events [298, 420] rather than mere summary statistics of the observations. These observations motivate a principled computational framework of value-based decision making that would involve choice values (and the BG), memory (and the hippocampus) and prospective planning (in the frontal cortex). Using the models proposed in Chapters 3 and 4, a sketch of this decision framework is proposed in Section 5.3.

Several attempts have been made to link together RL and SSM [421]. Relating the two models is tempting, as the same neural structures (the BG) are involved in both decision making and RL [349]. Frank and colleagues [11] showed that sequential sampling of rewards given the learned contingency explained well the RT distribution when modelled with a DDM where the difference in reward conditioned the drift rate. Pedersen and colleagues [12] showed that using the DDM as a choice rule in model-free RL gave better parameter estimates than the more classical Boltzmann softmax rule.

In their model of RL decision making based on the evidence integration, Bogacz and Larsen [349] assume that the ventral striatum (the critic) learns the state-action values through dopaminergic RPE signals. These signals also guide the decision process achieved by the dorsal striatum (the actor, which belongs to the cortico-baso-putaminal loop). These authors assume a point estimate of the learned action value: in this case,

no integration of information is needed from the RL point of view, as the estimated mean return is plain from the beginning. In a Bayesian and an active inference perspective, however, the optimal action is not known in advance, and the critic has an imperfect knowledge of the value functions. It has been suggested that the hippocampus provides the values to the cortico-striatal value network [397] that, in turn, is responsible for instructing the actor. This is in line with the Bayesian model of decision making proposed in Chapter 4, where the rewards are simulated from their posterior predictive distribution and compared by the actor to reach a decision.

Model-based RL has not been left behind of all these attempts to unify theories of decision making. For instance, Solway and Botvinick [422, 423] have proposed a decision scheme where evidence integration in model-based RL is achieved by simulating episodes of upcoming events and summing the sampled rewards. An interesting feature of this model is that when an agent samples a path, it keeps it in memory and, when reaching the next stage, it revises its strategy in the new state. This is again a way to sparsify the model (as what is done with amortized inference [109]) and is compatible with the idea that habits can be partially explained by the reliance on action sequences: previous inference conditions (or *biases*) the current strategy. Again, the hippocampus seems to guide the simulations executed by the model-based controller [269].

Here too, threshold variability should be taken into account, as it is thought that this factor (which controls the error rate) is crucial to control the reward rate [341, 198, 308]. Moreover, threshold adaptation has been observed to be modulated by decision conflict with a mechanism imputed to a network involving the subthalamic nucleus and the pre-SMA [384, 11]. Finally, controlling the threshold should also dictate some exploration/exploitation balance, as lower threshold will make the behaviour more erratic.

We know from the previous sections that accounting for uncertainty in decision policy is a crucial aspect of behavioural optimality for both model and feature selection. Dunovan and colleagues [246] review evidence for the role of the BG in value-based decision. In the model they propose, the BG are the convergence point of the various value signals (driven by contextual and sensory information), and are the central processing unit of the *actor-critic* agent outlined in the previous section. The synaptic weight in the direct and indirect pathways are adapted to modulate the drift rate value: higher and more certain values of rewards lead to a faster accumulation of evidence. This learning of action values is supposed to be mediated by *phasic dopamine* levels [244]. The *tonic dopamine* level, on the other hand, is thought to modulate the exploration/exploitation trade-off [424, 248, 425] by some sort of thresholding phenomenon [246]. Note that this is an overly simplified view [426]: multiple mechanisms seem to be involved in speed-accuracy trade-off adjustement. Moreover, lower levels of tonic dopamine level promotes exploratory

behaviours with widespread distributions of RTs and erratic behaviours. A threshold decrease should on the contrary promote erratic behaviours with *narrower* distributions of RTs. In this context, it is therefore more likely that the effect of the level of the tonic dopamine level on decision making would be better modelled as an effect of the precision on the threshold: if a *decrease* of tonic dopamine level measures an decreased precision (increased variance) of the moment estimates, triggering an *increase* of the threshold with a constant drift would lead to the desired behavioural pattern. This is equivalent to say that the tonic dopamine level measures the (inverse of) posterior variability of the average outcome in a Thompson sampling sense, and that the threshold is adapted to this measure of variability. By doing so, the agent ensures that it keeps the error rate of the SPRT roughly identical, which would not be the case if the threshold was kept constant. This view is compatible with the empirical evidence that tonic dopamine encodes the outcome posterior precision, whereas the phasic dopammine encodes outcome value [427, 428]. Figure 1.9 shows how the level of the tonic dopamine level controls speed-accuracy trade-off and the exploration/exploitation balance in this framework. As a mixture of Thompson sampling (see Chapter 4, Wald's SPRT and decision threshold adaptation, this model optimizes exploration/exploitation online and is asymptotically optimal (i.e. leads to the optimal policy) in a stable environment.

1.3.2 Other SSM

We briefly mention other SSM used to model decision making. The algorithmic details and comparison with the DDM of some of the SSM are provided by [308, 5]. The Full DDM assumes that the drift rate varies from trial to trial, but the possibility that this value changes during a trial should also be considered. This is usually modelled with a Ornstein-Uhlenbeck model, where the drift rate changes as a function of the evidence level [429] so that evidence is either attracted or repulsed by a fixed point. Ditterich [355] has used a somehow similar model to account for the fact that animals tend to speed up their decision following some sort of urgency when evidence accumulation is too slow. Algorithmically, increasing the drift rate *and noise* is indeed similar to lowering the threshold.

This *collapsing threshold* model has received much attention [430, 368, 431] (discussed in [432, 433]). Roughly speaking, the idea is that under time constrains, an agent can lower its response threshold, making less precise decisions for lower drift rates but ensuring that a response is made when urgency is present [362, 344, 345]. Note that urgency can be also understood as a compromise between optimality and maximization of the reward rate given by a suboptimal policy [198, 160]. This hypothesis is explored by [370] in the context of motion discrimination tasks, where the evidence accumulation during

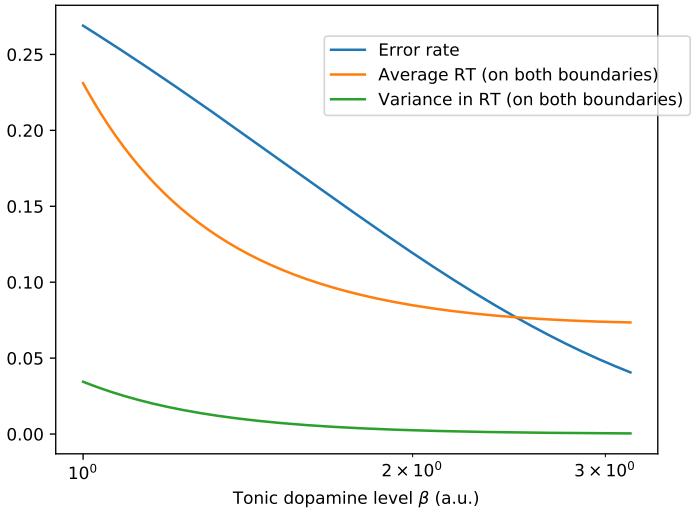


FIGURE 1.9: Exploration/exploitation control by tonic dopamine level β. The term "error rate" reflects the proportion of trials where the absorbing boundary is not the one toward which the drift is oriented. The figure was generated with the assumption that for a given noise ς_0 and a given threshold ζ_0, these values were adapted according to $\zeta = \zeta_0/\beta$ and $\varsigma = \varsigma_0/\beta$. In practice, measuring the noise in the accumulation process can dictate the level of the threshold that has to be adopted: if a decision is hard because it has a high variance (low tonic dopamine levels), the threshold is increased proportionally and the drift is kept unchanged, resulting in longer and more variable RTs, but a stable error rate (in a Thompson sampling sense). This corresponds to the behavioural and neurophysiological observation that an increase in tonic dopamine level leads to a less exploratory behaviour.

a trial is modelled as a POMDP belief update scheme. In their model, the threshold collapses at a rate dictated by the amount of information gathered in a trial (which, in their implementation, changes the variance of the estimate and consequently lowers the decision threshold) and by the long-term advantage of making a premature decision to maximize the overall return. Gluth and colleagues [434] showed that the phenomenon of collapsing threshold correlated with the activity in the pre-SMA and in the caudate nucleus.

1.4 Concluding remarks and open questions

This chapter provided some background theory on approximate inference, reinforcement learning and sequential sampling models of decision making. This thesis finds its place at the crossroad of these three fields.

The first theoretical question we will try to answer concerns the question of modelling Bayesian RL in unstable environments. We have seen that model-free RL can model the emergence of inflexible behaviours in MDPs as they need to re-explore the whole chain leading to a devalued action to learn its new long run utility. This does not tell us, however, how should an agent consider a surprising event in a Bayesian setting. Automaticity, which relates tightly with habitization, is the process by which action (or action sequence) costs are lowered with training. This process is thought to allow the agent to perform multi-tasking but reduces flexibility. In the following chapters, we will introduce a theoretical framework where the main assumption is that an agent using Bayesian inference to learn and forget a causal model of the world has advantage to make inference about the stability of this environment to be resistant to unexpected but noisy observations (outliers) while still keeping its ability to learn change of contingencies. The core idea of the algorithm we propose is that, under the assumption that the environment is meta-stable (i.e. that fewer change points in the past predict fewer change points in the future), there is an advantage to build a trust in the environment stability, as this can help the agent to soften the effect of outliers. The idea that the brain uses a strategy similar to the one of an actor-critic agent is still widely diffused, and, following this concept, we will propose an actor-critic decision making framework where the critic learns the posterior distribution of the reward distribution parameters, and the actor uses this distribution to sample a decision to maximize the long-term return.

Our aim will first be to develop a general variational framework to fit the DDM, which will serve as a decision rule in the following chapters. As a by-product, we will show that using recent advances in inference models such as auto-encoders can help to fit the DDM in a fully Bayesian way. This will be the topic of Chapter 2.

We will next tackle the problem of formalizing the theoretical framework sketched above. The model we propose is, in general, intractable, and we will use VI to implement a fast learning technique. In Chapter 3, the Hierarchical Adaptive Variational Filter will be presented. We will present this model in the context of RL, system identification and stochastic gradient descent.

Finally, in Chapter 4, we will join the HAFVF with the DDM in a united Actor-Critic model of learning and decision making.

Chapter 2

The Recurrent Auto-Encoding Drift-Diffusion Model

This work is currently under editor's review for the Journal of Mathematical Psychology. Alexandre Zénon is co-author.

The Drift Diffusion Model (DDM) is a popular model of behaviour that accounts for patterns of accuracy and reaction time data. In the Full DDM implementation, parameters are allowed to vary from trial-to-trial, making the model more powerful but also more challenging to fit to behavioural data. Current approaches yield typically poor fitting quality, are computationally expensive and usually require assuming constant threshold parameter across trials. Moreover, in most versions of the DDM, the sequence of participants' choices is considered independent and identically distributed (*i.i.d.*), a condition often violated in real data.

Our contribution to the field is threefold: first, we introduce Variational Bayes as a method to fit the full DDM. This will lead us to use the Full DDM as a principled model of decision making in Bayesian RL in Chapter 4. Second, we relax the *i.i.d.* assumption, and propose a data-driven algorithm based on a Recurrent Auto-Encoder (RAE-DDM), that estimates the local posterior probability of the DDM parameters at each trial based on the sequence of parameters and data preceding the current data point. Finally, we extend this algorithm to illustrate that the RAE-DDM provides an accurate modelling framework for regression analysis. An important result of the approach we propose is that inference at the trial level can be achieved efficiently for each and every parameter of the DDM, threshold included. This data-driven approach is highly generic and self-contained, in the sense that no external input (e.g. regressors or physiological measure) is necessary to fit the data. Using simulations, we show that this method outperforms

i.i.d.-based approaches (either Markov Chain Monte Carlo or *i.i.d.*-VB) without making any assumption about the nature of the between-trial correlation of the parameters.

2.1 Introduction

Sequential sampling models [435] are models of behaviour that assume that decisions are the result of a noisy accumulation of evidence. They have been presented theoretically in Section 1.3. Here, we focus on the very practical aspects of the use of DDM in research. The most famous of these models, called the Drift Diffusion Model (DDM), was introduced in the neuroscience field by Roger Ratcliff [1] as a tool for studying animal decision making and has led to an important amount of publications during the past decades.

The DDM is based on a Wiener process, where the evidence in favour of one of several (typically two) actions is represented by a particle moving towards a boundary with a certain drift rate and subject to a given noise. The simplest implementation of the DDM assumes the generative parameters of the model to be fixed across the different trials throughout the experiment. However, this simple version fails to account for some specific patterns of reaction times (RT), which led to subsequent improvements [363, 1, 7, 364, 436, 9] in which parameters are free to vary from trial to trial according to some assumed distribution, giving birth to what is known as the Ratcliff Diffusion Model [358], Extended DDM [308] or Full DDM [437].

While the Full DDM is more realistic (see [438] for a review and discussion), two main drawbacks of this method can be individuated: first, it is more computationally expensive to fit than its simpler counterpart by several orders of magnitudes, as a marginal probability density has to be estimated (or approximated) for each and every trial. So far, several methods used to fit the Full DDM to behavioural datasets have relied on Numerical Quadrature (NQ) [8, 439, 9, 440, 441] or Markov Chain Monte Carlo (MCMC) [442]. NQ is computationally expensive, which precludes its use for large problems. While MCMC [22] can be cheaper, it still requires an optimization time and memory storage that is proportional to the size of the problem: this precludes its use on large to very large datasets, which are encountered more and more frequently in cognitive neuroscience with the use of mobile devices and web tools such as Amazon Mechanical Turk [443, 444, 445, 446, 447, 264, 448, 449, 450, 451].

The second important drawback of the Full DDM regards its identifiability. Typically, only trial-to-trial variability in three [8, 9] or less [364, 15] of the four parameters of the DDM are assessed (drift, start-point and non-decision time, see below). Variability

of the fourth parameter, the threshold, is usually regarded as non-identifiable, because its trial-to-trial variability would be behaviourally equivalent to the variability in the starting point position [9, 452]. This can be understood intuitively by observing that an increased variability for both these parameters leads to the same behavioural pattern, namely more frequent and faster errors wrt accurate responses. In this perspective, [385] suggest that the choice of the prior distribution shape of the starting point (or threshold) determines whether the generative capacity of the model *for a given dataset*. However, the rationale for discarding threshold variability is mainly empirical and lack precise mathematical justification [9]: indeed, the lack of identifiability pointed by [385] generalizes to every DDM parameter. We suggest that, even though several models can predict accurately the data at hand, they will differ in their predictive capacity (i.e. their ability to generalize to unseen data). Validation (or cross-validation) and model evidence assessment will therefore be crucial to select model features that should be incorporated in the model. Here, we show that variability in threshold is instead identifiable together with the rest of DDM parameters, especially when the values of the threshold and starting point at a specific trial can be predicted from past behavioural observations (*time-dependency assumption*) and when these values covary with other parameters or physiological measures (*covariance assumption*). These two restrictions are likely to be observed in practice, as we will see later. In these two cases, one can take advantage of the fact that the values of other variables (latent or observed, past or concomitant) constrain the fit of the threshold variability. Although we will focus in the present paper on the first assumption, the approach we propose can model these two phenomena.

As mentioned above, even when the threshold is considered as fixed, identifiability of the Full DDM remains an issue [10]. It is usually considered that one can expect the fit of the mean of the DDM parameters to be accurate [9], but not their variance (i.e. trial-to-trial variability) [8], and that inter-trial variability identification requires more data than average parameter value identification [453]. In summary, large datasets (e.g. > 100 trials per subject) take unreasonable amount of time to be optimized with these models, for a very limited quality of fit, which worsens dramatically with smaller datasets.

On top of this, another issue with many implementations of the DDM lies in the core assumption of the method used to fit the parameters to the data. Maximum Likelihood (ML) estimate (i.e. finding the most likely value that the parameters would endorse in order to generate the data at hand) or similar frequentist methods such as Chi-Squared criterion were traditionally used [8, 454, 455, 439, 456, 457, 9, 440, 453] (see Box 2.1.1[1]),

[1]In the following chapters, boxes will indicate parts of the work that did not appear in the initial version of the paper (either publish or submitted). They might treat of issues of the algorithms that were not addressed properly, additional information discarded for the sake of space or suggestions for future work.

until Bayesian attempts were made [458, 459, 441]. Here, we focus on the ML technique.

Box 2.1.1| Frequentist fit of the DDM

In the literature, we distinguish three frequentist methods to fit the DDM: the first is based on ML estimates, which maximize the sum of the probability density of every point with respect to the parameters of interest. This approach is usually thought to be sensitive to outliers, so that these contaminant RTs need to be identified accurately and removed if one hopes to find a tangible fit of the parameters [8, 439][a]. The Chi-squared (CS) criterion [8] bins the data according to their quantiles, and find the parameter configuration that minimizes the difference between the expected and the real number of points in each bin. Both this approach and the Kolmogorov-Smirnov (KS) [460] work by considering the whole dataset as being *i.i.d.* (given the global parameters), and work well only in cases where many datapoints can be used to retrieve some summary statistics from the whole population.

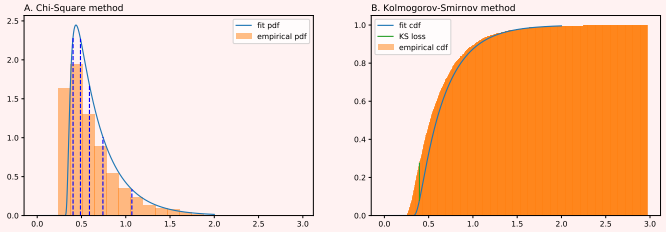


FIGURE 2.1: In the CS loss (**A.**), the aim is to minimize the discrepancy between the predicted and actual number of datapoints in each quantile (vertical dashed lines represent the quantile boundaries). The KS loss (**B.**) is computed as the maximum distance between the true and expected CDF for the distribution of interest. None of the methods rely on differentiable loss functions.

In a hierarchical and Bayesian perspective, we would like to be able to make inference about the posterior distribution of each local latent parameter, and not only about a fixed prior of these. As Figure 2.1 shows, both KS and CS require a large dataset to compute an empirical CDF of the data. This observation also gives us the intuition that relaxing the *i.i.d.* assumption might require us to derive a loss-function factorized over all datapoints, as we will need to minimize some form of loss with respect to each local parameter.

[a]Note that when a datapoint is considered as an outlier by Ratcliff and Tuerlinckx [8], it is simply removed from the dataset. In our approach, every datapoint is kept as part of a time series. Behaviourally, we argue that the former strategy makes more sense: it is likely that a very fast or very slow RT is caused by a process already in place in the preceding trials, and that it will instruct the subject to be more cautious in the following trials.

Crucially, the validity of the ML approach is based on some asymptotic properties of

the summary statistics of a given dataset when the amount of data tends to infinity [74]. However, as we have seen, the computational cost of the numerical methods commonly used to optimize the Full DDM prevents us from fitting large datasets. Hence, we face a dilemma: a large data set will prevent us from computing the ML using classical NQ algorithms, and a small dataset set will violate the assumptions of the ML estimate. Usage of MCMC only partly solves this problem, as its computational cost is still a function of the size of the dataset, and since it still relies on a strong *i.i.d.* assumption about the generative process.

The core idea of this paper is that trial-to-trial dependency of the parameter values and behavioural results can, and should be used to help estimate the posterior probability of the latent variables. This assertion is mainly theoretically justified by the observation that actions belonging to sequences can not be considered as independent (see for instance [380, 461, 462, 463, 464, 465, 466, 298, 325]). Let us illustrate this by considering the toy example where a participant has been asked to press 9 times the same button (e.g. right), and we are performing inference about the DDM parameters at the 10th button press. From a Bayesian point of view, it is very likely that, at this trial, the subject will choose the same action as before. If the participant is asked to choose, and indeed chooses, the right response, we can expect that the RT at this trial will be, on average, lower than the one at the trials before [461]: this is a typical sequential effect. However, if the participant is asked to choose the left response at this trial and responds accurately, it is evenly likely that this response will be slower than the previous ones, as the current action violates the previous pattern. Another example is post-error slowing, a well-known psychological phenomenon usually interpreted as a change of speed-accuracy trade-off following an error [467, 384]. This might be modeled as a resetting of the threshold to a high value following an error. But an error might also bias the decision in the next trial towards one of the responses, or impair the attention of the subject during the subsequent trials. Our aim will be to take advantage of these cross-correlations in a blind manner, i.e. without making any specific assumption about the precise structure of the trial-to-trial correlation. This will be achieved by capturing the most relevant trial-to-trial cross-dependencies with an assumption-free dimensionality reduction algorithm.

To do this, we first show that Approximate Bayesian inference techniques can efficiently uncover the parameter sequence: the algorithm we propose is a data-driven, nearly assumption-free model that relaxes many classical assumptions, such as the parametric form of the parameter distributions [440] at the within and between subject levels. Also, it can prevent overfitting by the use of dimensionality reduction and it is scalable to large datasets. Finally, training of this model can be trivially parallelized and accelerated using GPU.

More specifically, in this paper, we bring two nested solutions to the aforementioned problems: first, we show that the Full DDM finds a natural formulation in Bayesian Statistics (Section 2.2.2 and Section 2.2.3), and that we can use this formulation to solve efficiently this problem using recent advances in approximate inference. Here, we will focus on the use of Variational Bayes (VB) techniques (Section 2.2.4). We then present a computationally efficient VB model (*i.i.d.*-VB) to compute the marginal probability density of the Full DDM. This algorithm has the advantage of being scalable, its computational cost and memory usage being amortized wrt. the size of the problem.

Next, we show how, when the *i.i.d.* assumption is relaxed, this basic algorithm can be extended to account for trial-to-trial dependency in a data-driven mode using a Recurrent Auto-Encoder (RAE-DDM) (Section 2.2.6). This approach is free of any assumption, in that it does not require the user to select a specific prior distribution for the parameters nor a precise form of trial-to-trial cross-correlation.

We show in the results (Section 2.3) that our RAE approach outperforms classical approaches (*i.i.d.*-VB and Hamiltonian Monte Carlo, HMC) in terms of model evidence and fit of the data at the trial, subject and population level if one assumes that the parameters and datapoints are temporally ordered. An important and compelling result we provide with the RAE-DDM consists in showing that the trial-to-trial variability of the threshold parameter is possible to estimate together with the other parameters of the DDM.

In Section 2.4, we discuss our model in detail, highlight situations where it might prove useful, see its limitations and introduce potential improvements.

2.1.1 Related Work

Many routines have been proposed to optimize the DDM. Ratcliff and Tuerlinckx [8], and others [468, 437] have used a NQ algorithm with a numerical estimate of the gradient to optimize the likelihood function at the subject level. Voss and colleagues have used the partial differential equation to compute the CDF of the first passage density in the Full DDM setting, and have used a SIMPLEX method to fit the model to data [453]. The computational cost of these methods can be extremely high, as each point needs to be re-evaluated at each iteration, justifying the development of less flexible, but simpler methods [469, 358, 359]. Vandekerckhove et al. [459] used an MCMC technique to estimate each trial distribution in a fully Bayesian setting. This is the approach adopted in several Bayesian statistical softwares such as JAGS [470], WinBUGS [459], or Stan [471], which has broaden the use of this technique for the optimization of the Full DDM in many recent publications (e.g. [472, 11, 473, 474, 12]). HDDM is a Python

toolbox written by Wiecki and colleagues [441], aimed at fitting the DDM using MCMC. Peculiarly, this toolbox still relies on NQ to compute estimates at the lower level, rather than estimating each local posterior of the DDM parameters. One of the features of the algorithm we propose is that it alleviates the computational burden of the integration process while keeping the entire original structure of the DDM without compromising on accuracy.

The method detailed below differs in many ways from the work of Vandekerckhove and colleagues [459]: first, the RAE-DDM is scalable, such that we can make inference about the generative model of large (or very large) datasets with limited cost, as the gradient of the lower bound to the log-model evidence is estimated stochastically. Our use of Amortized Variational Inference [110] allows us to control the convergence of the algorithm using a validation dataset, which is challenging with MCMC techniques. Finally, the RAE-DDM, besides its ability to account for trial-by-trial variability in the threshold values, can also provide a more precise estimate of the posterior distribution of local parameters, as it conditions this posterior on more information than just the RT, the choice and the prior by adding the previous trials as regressors for the value of the prior and local variable. None of the approaches presented so far took advantage of this cross-dependency between the values of latent parameters, choices and reaction times from trial to trial. We are not, however, the first to propose a form of time-dependant estimate of the behavioural parameters: [461, 462, 463, 475] have for instance proposed simpler, informative models to uncover precise psychological heuristics that were always fit using Maximum-Likelihood methods. The assumption-free nature of our model, traded with its latent parameter sparsity, makes it a compelling tool to give a precise and generalizable estimate of the expected value of the DDM parameters at each trial as well as the uncertainty of this estimate.

Also, [14, 15] proposed a model (the Neural Diffusion Model) where the neural data and behavioural parameters are governed by a common distribution that ties together the two kinds of measures. Because the covariance of the two types of parameters is inferred, the fit can indirectly take into account the time evolution of the DDM parameters through the observed evolution of the neurophysiological measures. Notably, this approach requires a large dataset [16] and inference is achieved with MCMC method. Also, it is still based upon a highly informative model, where the user has to set (and possibly test) precisely the form of the model that couples the parameters. Finally, this model is complex and has a large number of latent variables, making overfitting an issue.

Our approach, on the contrary, takes full advantage of the dimensionality reduction property of RAE to set the number of parameters of the model to a minimum. It also contrasts with the neural diffusion model by the fact that it is self-contained, meaning

that no external input is needed, although it can easily be included in the model. It also relaxes greatly the need to set a precise form to the model that governs the data: as we will see, the use of non-linear mapping between the latent trial-specific prior and its DDM-mapped value flexibly models a large variety of prior shapes with a limited and controllable risk of overfitting.

Our approach to Bayesian inference relies also on a large body of previous work. The Helholtz machine and the Wake-Sleep algorithm [151, 152, 476] developed almost twenty years ago by Hinton, Dayan and colleagues constitute a cornerstone in the history of Bayesian inference. This model and algorithm are a mirrored recognition-generative model, which works in two phases: during the wake phase, the algorithm learns from the data the generative model by minimizing the error between the predictions and the observed data. During the sleep phase, the generative model is held fixed and is used to generate fake data that are used to train the recognition model. Built on the very same idea, Variational Auto-Encoders (VAE) [88] and its recurrent form [477, 478] have quickly become standard and popular tools to model data in many areas [**Goodfellow2016a**]. Advances in this area go fast, and giving an accurate account of all of them is beyond the scope of this paper. The present paper builds on recent developments of Recurrent Auto-Encoders that have been concomitantly achieved by several groups (e.g. [479, 480]). In short, the first challenge with these techniques regards determination of the best approximate posterior shape for the latent variable $\mathbf{z}$ [103, 105, 106]. An efficient solution to improve the performance of a VAE is to replace the Variational estimator by an Importance Weighted (IW) [118] estimator of the posterior probability. This simple trick can greatly improve the robustness and convergence of the model. We will show in what follows how this technique adapts to the problem of the DDM. Another issue consists in regularizing the parameters of the recognition and generative models: Dropout [481] has been shown to efficiently prevent overfitting with similar models [89], motivating our choice of relying on variational Dropout to perform inference about the value of the network weights [482].

A final remark concerns identifiability of the model we propose, and of the Full DDM in general. The impact of the shape of the prior used for inference has led to vivid debates in the DDM community. Some authors have claimed that a prior probability of an arbitrary complexity can always be found in order to reflect perfectly the data, implying that the choice of the prior completely determines the capability of the DDM to predict data [385]. For instance, following these authors' logic, also disputed by [483], a J-components linear mixture model prior (where J is the number of trials of the dataset) over the sole parameter ξ can reflect perfectly the dataset, even if the noise parameter is set to 0 and other parameters are kept fixed. However, this overfitting issue applies to any hierarchical model where local variables are modeled: as a matter of fact, there

should always exist an ML estimate of any model that would predict perfectly the data at hand with little or no generalization capability. Indeed, two specific – and widely used – solutions exist to deal with overfitting issues. The first approach consists in using Bayesian Model Comparison [132, 138], in which the model evidence (i.e. probability of the data given the model marginalized over the space of parameters) of models of different complexity is compared. The second approach consists in assessing the model predictive capacity on a *test* dataset, separate from the *training* dataset that is used for fitting the model. In the above example, validation of this *ad-hoc* model will obviously fail to show that the model is able to provide accurate insights about unseen data. Model comparison has been discussed in more detail in Box 1.2.9. Moreover, we propose another solution to this identifiability problem, which is that the values taken by the DDM parameters at each trial is only random to somne extend. We postulate that these values can, at least partially, be predicted based on the values that these parameters (and related data or regressors) took in the past trials. The approach we propose can achieve these three goals efficiently, as the model is scalabe (i.e. it is able to generalize its predictions to unseen data), provides a lower-bound to the log-model evidence, that can be used for Bayesian Model Comparison and can account for autocorrelation of the parameters and data in a blind manner.

Furthermore, the method we propose alleviates the need to specify a precise prior probability distribution, but, by the use of dimensionality reduction, it ensures that the trial-wise latent variable $\mathbf{z}$ captures a maximum of the variability, thus allowing us to keep a highly generalizable model structure with little or no cost caused by the specific prior specifications.

Finally, the capacity of our approach to estimate the threshold variability is a considerable benefit, which has never been considered before. Other authors (e.g. [384, 11]) have used a regression model to estimate the threshold value at the trial level using a measured neurophysiological signal or experimental condition. The RAE-DDM does not preclude the inclusion of these parameters in the model, but would most likely provide a better fit because it also could account for the trial-to-trial dependency in the parameter values that might not be captured by the external regressor.

2.2 Methods

2.2.1 Notation

It is convenient to first define the notation we will use. In the following, we will refer to the trial index as $j \in 1 : J$ and the subject index as $n \in 1 : N$. The trial-specific

configuration of the five DDM parameters, namely the threshold ζ, the drift-rate ξ, the relative starting point w and the non-decision time τ will be grouped under a single vector $\boldsymbol{\omega}_{j,n}$ (or, if applicable $\boldsymbol{\omega}_n$ if it is assumed that $\boldsymbol{\omega}_{j,n} = \boldsymbol{\omega}_{j',n} \forall j, j' \in 1 : J$). The subject-specific prior distribution parameters of $\boldsymbol{\omega}_{j,n}$ will be generically grouped under the parameter vector $\boldsymbol{\theta}_{\mathrm{n}}$. As we will see in Section 2.2.4, we will also define an approximate posterior distribution over $\boldsymbol{\omega}_{j,n}$ with parameters $\boldsymbol{\delta}_{j,n}$. In the *i.i.d.*-VB implementation we propose, these two sets of parameters will have a specific form ($\boldsymbol{\theta}_{\mathrm{n}} \triangleq \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2\}$ and $\boldsymbol{\delta}_{j,n} \triangleq \{\boldsymbol{\mu}_{j,n}^{\boldsymbol{\omega}}, \Sigma_{j,n}^{\boldsymbol{\omega}}\}$). For the purpose of sparsity of the notation and/or to keep the discourse as general as it may, we will use the generic indicators more often than their implementation-specific counterparts.

The behavioural observations are grouped under $\mathbf{y}_{j,n} \triangleq \{\mathrm{t}_{j,n}, a_{j,n}\}$, where $\mathrm{t}_{j,n}$ and $a_{j,n}$ are the corresponding Reaction time (RT) and choice, respectively. Note that, for the purpose of generality, we distinguish the model boundary b and the choice a, although we will freely use $p(\mathrm{t}, a)$ and $p(\mathrm{t}, b)$ interchangeably following the context. In general, capital letters will represent the set of all the corresponding parameters or random variables (i.e. $\mathbf{Y} \triangleq \{\mathbf{y}_{j,n} \text{ for } j \in 1 : J, n \in 1 : N\}$, $\boldsymbol{\Theta} \triangleq \{\boldsymbol{\theta}_{\mathrm{n}} \text{ for } n \in 1 : N\}$ etc.). The k^{th} draw of a sample will be indicated by a tilted symbol, such as $\widetilde{\boldsymbol{\omega}}_{j,n,k}$, and the true (and obviously unknown) value of a parameter will be indicated by $\widehat{\boldsymbol{\omega}}_{j,n}$.

Finally, if a parameter is deterministically determined by some other variables, such as $\boldsymbol{\delta}_{j,n} = f(\mathbf{y}_{j,n}; \boldsymbol{\rho})$, we will use interchangeably this notation with the sparser notation $\boldsymbol{\delta}_{j,n}(\mathbf{y}_{j,n}; \boldsymbol{\rho})$.

2.2.2 DDM

The Drift-Diffusion model is a member of the sequential sampling model family. The essence of these models is that the experimenter models the decision process as an accumulation of evidence in favour of one of the available responses: the probability of a pair of observations $\{a_{j,n}, \mathrm{t}_{j,n}\}$ is formulated as the probability that the evidence reached the threshold b at time t. Importantly, this accumulation of evidence is stochastic: given the same initial state, the same agent will produce a different response time and, possibly, choose a different item. This stochasticity is the main asset of this model [483], since it can explain speed-accuracy trade-offs: indeed, certain parameter settings will produce faster but more entropic response distributions, whereas other will produce slower, more directed responses.

Formally, in the DDM, the evidence accumulation process is assumed to follow a Wiener process (Figure 2.2):

$$dx = \xi\, dt + \varsigma\, dW$$

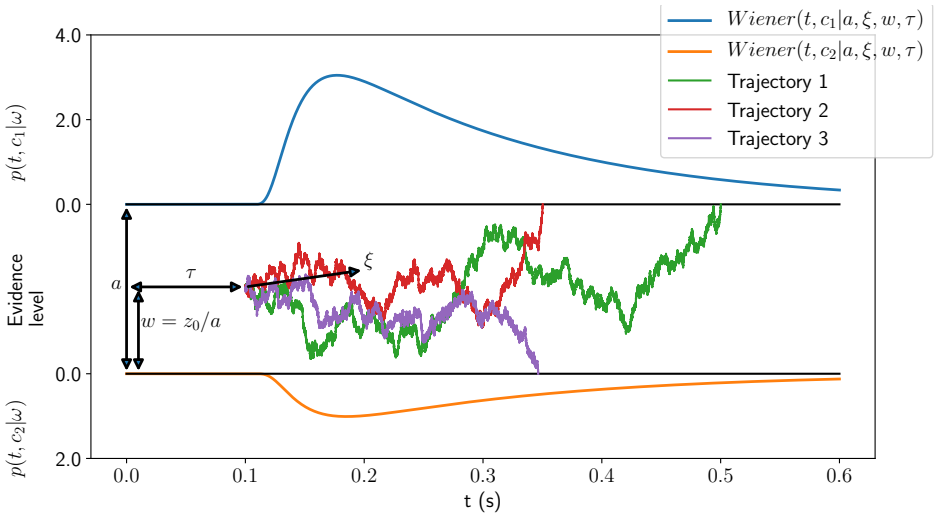


FIGURE 2.2: DDM structure. The x axis represents the time scale, the y axis the evidence level. Four parameters (threshold ζ, drift ξ, starting point w and non-decision time τ) are represented. At the two boundaries, the first passage probability density function (sometimes called defective density) is represented. Three accumulation trajectories are displayed.

where dx is the evidence step, dt is the time step and ςdW a Gaussian noise with a standard deviation ς. In this specific model, the agent accumulates evidence in favour of an action a_1 with a rate ξ, called the drift rate, meaning that in the case of a classical Two-alternative forced choice task, she will also accumulate evidence in favour of the alternative action a_2 with a rate $-\xi$. Once the evidence crosses one of the two absorbing boundaries (the boundary b positioned at a distance ζ corresponding to the action a_1 or the boundary $\neg b$ at position 0 corresponding to the action a_2), the corresponding action is selected and, after some physiological delay τ, this action is executed. The evidence accumulation can start wherever at $z_0 \in [0; \zeta]$, although it is more convenient to model the relative starting point $w \in [0; 1]$ where $w = z_0/\zeta$. Finally, as mentioned in Section 1.3.1, this 5-parameter model is non-metric: multiplying ξ, ζ, z_0 and ς by any positive scalar does not change the behavioural predictions. It is therefore necessary to fix one of them for the model to be used: The accumulation noise $\varsigma = 0.1$ is usually used as a metric, but, for the sake of mathematical explicitness and simplicity, we will use the metric $\varsigma = 1$ in the following.

Fortunately, the First-Passage Time density of the Wiener process (*Wiener FPT density*) described before has a closed-form formula. For a Two-alternative forced choice task, the joint probability density of crossing a given boundary b (the lower boundary by convention) at a certain point in time t, is given by [350]:

$$Wiener(\mathrm{t}, b; \boldsymbol{\omega}) = \begin{cases} \frac{\pi}{\zeta^2} \exp\left\{-\xi\zeta w - \frac{\xi^2(\mathrm{t}-\tau)}{2}\right\} \times \\ \qquad \sum_{k=1}^{\infty} k \exp\left\{-\frac{k^2\pi^2(\mathrm{t}-\tau)}{2\zeta^2}\right\} \sin(k\pi w) & \text{if } \mathrm{t} > \tau \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

If the absorbing boundary is $\neg b$ (i.e. the upper boundary), the above equation can be adjusted by setting $w' = 1 - w$ and $\xi' = -\xi$. Together, the sum of the First Passage Time Densities at the two boundaries (sometimes called defective probability densities [436, 15]) forms a valid probability distribution, i.e.:

$$\begin{gathered} \sum_{b \in \mathbf{b}} \int_{\mathrm{t}=\tau}^{\infty} Wiener(\mathrm{t}, b; \boldsymbol{\omega}) d\mathrm{t} = 1 \\ \text{and } Wiener(\mathrm{t}, \mathbf{b}; \boldsymbol{\omega}) \geq 0 \text{ for } \mathrm{t} \in [\tau, \infty] \text{ and } b \in \mathbf{b} \end{gathered} \tag{2.2}$$

where $\mathbf{b} = \{b, \neg b\}$. The function in Equation (2.1) involves an infinite sum, and must therefore be implemented through approximations. Navarro et al. [351] provided a method to truncate this infinite sum to the appropriate number of terms κ as a function of truncation error. Here, we used their approximation with a numerical precision of 10^{-32} to estimate the parameters value.

The simplest and most intuitive way to fit the DDM to behavioural data is to look for the parameter setting of $\boldsymbol{\omega}_n$ that maximizes the likelihood of observing this dataset given the parameters and the model. If tempting, this approach overlooks a crucial detail of the DDM: if the true value of $\boldsymbol{\omega}$ varies from trial-to-trial, the fit of a single vector $\boldsymbol{\omega}_n^*$ is a biased estimator of $\mathbb{E}\left[\widehat{\boldsymbol{\omega}}_{j,n}\right]$, and this bias can have an important impact on the interpretation of the parameter values [364]. The theoretical advantages and computational challenges of the Full DDM have been introduced in Section 1.3.1.1. We focus here on the algorithmic challenge of this model. Formally, we have

$$\mathbb{E}\left[\boldsymbol{\omega}_{j,n} \text{ for } j \in 1:J\right] \neq \boldsymbol{\omega}_n^*$$

where $\boldsymbol{\omega}_n^*$ is the value $\boldsymbol{\omega}_n$ that maximizes the log-likelihood function.

For instance, a high variability in the value of the starting point w with mean .5 will generally produce a dataset that will be better fitted by a point-estimate of $w^* < .5$ [364]. Accounting for the trial-to-trial variability in w can solve this problem. The problem then is to fit a prior $\boldsymbol{\theta}_{\mathrm{n}}$ that determines the trial-wise distribution of latent parameters $\boldsymbol{\omega}_{j,n}$. In frequentist paradigms, the parameters $\boldsymbol{\omega}_{j,n}$ are usually integrated out of the model, and $\boldsymbol{\theta}_{\mathrm{n}}$ only is fitted. As we will see in Section 2.2.3, in a fully Bayesian

framework, all the parameters ($\boldsymbol{\omega}_{j,n}$, $\boldsymbol{\theta}_{\mathrm{n}}$ and its prior) can be regarded as latent random variables and are estimated accordingly.

Usually, only the subset $\{\xi, w, \tau\}$ is considered to vary from trial to trial, and the threshold is considered to be fixed for each participant. Although it may raise some concerns in readers familiar with the full DDM, we do no make this assumption here, as it is customary in Bayesian statistics and machine learning to consider that each datapoint is generated by its own set of parameters (see for instance [76]). We show in the results section that the performance of the model we propose still outperforms other implementations, and that the fit of the threshold provides meaningful results.

2.2.3 The Full DDM as an Empirical Bayes problem

The ML estimate of the Full DDM as formulated by [8][2] consists in finding the value of the prior ${\boldsymbol{\theta}_{\mathrm{n}}}^*$ that maximizes the log-likelihood function $\log p(\mathbf{y}_{j,n}; \boldsymbol{\theta}_{\mathrm{n}})$ while having marginalized out the trial-specific parameters $\boldsymbol{\omega}_{j,n}$ (and therefore the posterior probability under an unnormalized, unbounded uniform prior), and to consider this value as a consistent estimator of the true value of $\boldsymbol{\theta}_{\mathrm{n}}$, namely $\widehat{\boldsymbol{\theta}}_n$. This choice can be justified from the asymptotic normality of the posterior distribution theorem [74, 75]: as the dataset grows, and considering that the data are generated from the same process, the posterior distribution approaches a multivariate Gaussian distribution with a mean close to $\widehat{\boldsymbol{\theta}}_n$ and covariance matrix equal to the inverse of the Observed Fisher Information matrix, i.e. ${\boldsymbol{\theta}_{\mathrm{n}}}^* \to \widehat{\boldsymbol{\theta}_{\mathrm{n}}}$ as $NJ \to \infty$ and $p(\boldsymbol{\theta}_{\mathrm{n}}) \approx \mathcal{N}({\boldsymbol{\theta}_{\mathrm{n}}}^*, I({\boldsymbol{\theta}_{\mathrm{n}}}^*)^{-1})$, where $I({\boldsymbol{\theta}_{\mathrm{n}}}^*) = -\nabla^2_{{\boldsymbol{\theta}_{\mathrm{n}}}^*} \log p(\mathbf{y}_{j,n}; \boldsymbol{\theta}_{\mathrm{n}})$. Also, following the Bernstein-von Mises theorem [484, 485, 486], we can see that the choice of the prior becomes irrelevant to the posterior as the amount of data grows, which further justifies the ML technique for large datasets.

We show in Section 2.5.2 that the ML approach of the Full DDM reduces to what is known as an empirical Bayes [29, 30] estimator of $\widehat{\boldsymbol{\theta}}_n$: indeed, the value of $\boldsymbol{\theta}_{\mathrm{n}}$ that maximizes the Wiener density can be directly reformulated as a function of the posterior probabilities of each trial parameter $p(\boldsymbol{\omega}_{j,n} \mid \mathbf{y}_{j,n}; \boldsymbol{\theta}_{\mathrm{n}})$: this gives the intuition that the problem of fitting the Full DDM could be greatly simplified if one could derive an efficient technique to estimate the set of the trial-wise posterior probability density functions. Figure 2.3a shows the basic structure of the Full DDM in this perspective.

[2] As we have seen earlier, [8] does not advocate for the use of ML but rather for the use of a CS loss. This reference is used as a starting point for the discussion of the method we propose.

Box 2.2.1| Implicit-RQMC solution to the full DDM problem

In the following, we will be interested in deriving an approximate posterior for each local latent variable $\boldsymbol{\omega}_{j,n}$. One could use a more naive (and classical) formulation of the problem, such as $p(\mathbf{y} \mid \boldsymbol{\theta}_{\mathrm{n}})p(\boldsymbol{\theta}_{\mathrm{n}}) = \int p(\mathbf{y} \mid \boldsymbol{\omega}_{j,n})p(\boldsymbol{\omega}_{j,n} \mid \boldsymbol{\theta}_{\mathrm{n}})d\boldsymbol{\omega}_{j,n}p(\boldsymbol{\theta}_{\mathrm{n}})$. Inference with this model could use techniques developed for inference for implicit distribution (i.e. distributions which we can sample from but do not enjoy a tractable likelihood). In particular, the solution provided by [126] would be well suited for our problem. In a preliminary study, not disclosed here for the sake of space, we tried to solve the Full DDM problem with a similar approach that jointly used variance reduction techniques such as covariate, Rao-Blackwellization and randomized Quasi-Monte Carlo [487] to fit a VB lower bound to the log-model evidence. Unfortunately, although it did compete well with MCMC techniques for inference, we did not find that the quality improvement in this technique was worth being detailed thoroughly. Instead, we chose to provide more information about the *i.i.d.*-VB approach, which performed slightly worse than the implicit-RQMC counterpart, but was theoretically easier to introduce as a step towards the RAE-DDM. Moreover, the *i.i.d.*-VB is easy to plug in models (such as the AC-HAFVF presented in Chapter 4) that account for trial-to-trial variability in the DDM parameters in an informative way.

2.2.4 Variational Inference

In this section, we will present VB as a method to find the posterior of the trial-wise DDM parameters $\boldsymbol{\omega}_{j,n}$. We will first describe the most common EM approach (whose derivation for the DDM application is provided in Section 2.5.2), for the empirical Bayes problem. We will then see how VB can be used to efficiently implement EM.

2.2.4.1 Expectation Maximization

The EM algorithm consists in looping between the evaluation of each trial-specific posterior probability density function given the current estimate of $\boldsymbol{\theta}_{\mathrm{n}}$ (E step), before then finding the value of $\boldsymbol{\theta}_{\mathrm{n}}$ that maximizes the likelihood given the current posterior estimates (M step).

More specifically, the E step of an EM algorithm would rely on computing (or approximating) the posterior over $\boldsymbol{\Omega}_n$ (the set of all trial-by-trial DDM parameter values), which

reads

$$p(\boldsymbol{\Omega}_n \mid \mathbf{Y}_n) = \frac{\prod_{j=1}^{J} p(\mathbf{y}_{j,n} \mid \boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n} \mid \boldsymbol{\theta}_{\mathrm{n}})}{p(\mathbf{Y}_n | \boldsymbol{\theta}_{\mathrm{n}})},$$

assuming that the posterior probability factorizes over each datapoint. This posterior probability is intractable due to its normalizing factor (the model evidence) $p(\mathbf{Y}_n | \boldsymbol{\theta}_{\mathrm{n}})$. A first approach could be to use simulation techniques to simulate from this posterior density function, but the size of the problem precludes this choice, since we would need to compute NJ independent integrals at each step of the EM algorithm. Instead, we can make use of VB techniques [488, 489, 22, 490] to fit an approximate posterior to the true posterior: Variational EM (VEM) approximates the EM algorithm by using VMP (see Paragraphs 1.1.2.2.2 and 1.1.2.2.3) for the E step. We will use VEM as a starting point for the discussion of our algorithm. We will expose its weaknesses, and present another, neater method to maximizes the loss function $\ell(\mathbf{y}_{j,n}; \boldsymbol{\theta}_{\mathrm{n}})$ that relies on Stochastic Gradient Variational Bayes (SGVB) and INs.

2.2.4.1.1 Variational Expectation Maximization VB has been presented in Section 1.1.2.2. In short, we first define a proxy to the posterior density of an arbitrary shape $q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})$ where $\boldsymbol{\delta}_{j,n}$ is a general indicator of the variational parameters that we will optimize in order to match the real posterior as close as possible (see Figure 2.3b). In order to do so, we can choose to minimize the Kullback-Leibler (KL) divergence between this distribution and the true posterior. This will be done by maximizing the ELBO, which reads

$$\mathcal{L}(q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})) = \mathbb{E}_{q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})} \left[\log p(\mathbf{y}_{j,n}, \boldsymbol{\omega}_{j,n}) \right] + \mathbb{H} \left[q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) \right]. \tag{2.3}$$

Once the ELBO has been maximized wrt $\boldsymbol{\delta}_{j,n}$, we can consider that $q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})$ is as close as it could from $p(\boldsymbol{\omega}_{j,n} \mid \mathbf{y}_{j,n})$ given the shape of the distribution we have chosen, the prior probability $p(\boldsymbol{\omega}_{j,n})$, the data $\mathbf{y}_{j,n}$ and the likelihood function $p(\mathbf{y}_{j,n} \mid \boldsymbol{\omega}_{j,n})$.

This VEM method suggested above, where $\boldsymbol{\Delta}$ and $\boldsymbol{\theta}_{\mathrm{n}}$ are sequentially optimized, suffers from several issues: the first is that optimizing the approximate posterior for each of the local variables is not trivial, as the ELBO does not have, in general, a closed-form formula (see Section 2.5.2). The second problem is that this approach is highly sensitive to local optima: in other words, the initial draw of $\boldsymbol{\Delta}$ can frame drastically the final result of the optimization process [22]. Another important concern is that the cost of such optimization process is unreasonably high, as one needs to iterate across every datapoint in order to make a single update of the prior, which is not guaranteed to be

significant. All these obstacles find an elegant solution with the combined use of SGVB and inference networkd (IN).

2.2.4.2 Stochastic Gradient Descent, reparameterization trick and SGVB

Let us rewrite the optimization problem we need to solve to find the ML estimate of $\boldsymbol{\theta}_{\mathrm{n}}$ using VB in a concise form:

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \max_{\boldsymbol{\Delta}} \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{\omega}_{j,n};\boldsymbol{\delta}_{j,n})} \left[\log p(\mathbf{y}_{j,n};\boldsymbol{\omega}_{j,n}) + \log p(\boldsymbol{\omega}_{j,n};\boldsymbol{\theta}_{\mathrm{n}})\right] + \mathbb{H}\left[q(\boldsymbol{\omega}_{j,n};\boldsymbol{\delta}_{j,n}\right]].$$

The VEM approach detailed above essentially reduces to maximizing the ELBO wrt. $\boldsymbol{\Theta}$ and $\boldsymbol{\Delta}$. Intuitively, a more economical way to achieve this would be therefore to maximize both parameters at the same time. The gradient of this expression can be expressed exactly as

$$\nabla_{\boldsymbol{\Theta},\boldsymbol{\Delta}} \left\{\mathcal{L}\left(q(\boldsymbol{\Omega})\right)\right\} = \nabla_{\boldsymbol{\Theta},\boldsymbol{\Delta}} \left\{ \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{\omega}_{j,n};\boldsymbol{\delta}_{j,n})} \left[\log p(\mathbf{y}_{j,n},\boldsymbol{\omega}_{j,n};\boldsymbol{\theta}_{\mathrm{n}})\right] + \mathbb{H}\left[q(\boldsymbol{\omega}_{j,n};\boldsymbol{\delta}_{j,n})\right] \right\}.$$

With a little loss of generality, we consider that the entropy of the approximate posterior has a closed-form formula, and that samples can be generated from $q(\boldsymbol{\omega}_{j,n};\boldsymbol{\delta}_{j,n})$ using a differentiable function g st. $\widetilde{\boldsymbol{\omega}}_{j,n,k} = g(\boldsymbol{\delta}_{j,n},\boldsymbol{\epsilon})$ where $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ for some arbitrary $p(\boldsymbol{\epsilon})$. Under these conditions, the above expression has an unbiased, noisy Monte-Carlo reparameterized gradient estimator given by

$$\left\{ \begin{array}{ll} \nabla_{\boldsymbol{\Theta}} \left\{\mathcal{L}\left(q(\boldsymbol{\Omega})\right)\right\} & \approx \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \frac{1}{K} \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}_{\mathrm{n}}} \left\{\log p(\widetilde{\boldsymbol{\omega}}_{j,n,k};\boldsymbol{\theta}_{\mathrm{n}})\right\} \\ \nabla_{\boldsymbol{\Delta}} \left\{\mathcal{L}\left(q(\boldsymbol{\Omega})\right)\right\} & \approx \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \frac{1}{K} \sum_{k=1}^{K} \nabla_{\boldsymbol{\delta}_{j,n}} \left\{g(\boldsymbol{\delta}_{j,n},\boldsymbol{\epsilon}_k)\right\} \nabla_{\widetilde{\boldsymbol{\omega}}_{j,n,k}} \left\{\log p(\mathbf{y}_{j,n},\widetilde{\boldsymbol{\omega}}_{j,n,k};\boldsymbol{\theta}_{\mathrm{n}})\right\} + \\ & \qquad \nabla_{\boldsymbol{\delta}_{j,n}} \left\{\mathbb{H}\left[q(\boldsymbol{\omega}_{j,n};\boldsymbol{\delta}_{j,n})\right]\right\} \end{array} \right.$$

for a number of K samples (see Paragraph 1.1.2.2.7). As suggested in Paragraph 1.1.2.2.6, the form we have given to the (expected) log-likelihood makes it clear that

$$\widetilde{\nabla}^{\mathcal{L}} \triangleq \frac{1}{SUK} \sum_{n=1}^{S} \sum_{j=1}^{U} \sum_{k=1}^{K} \left[\begin{array}{l} \nabla_{\boldsymbol{\theta}_{\mathrm{n}}} \left\{\log p(\widetilde{\boldsymbol{\omega}}_{j,n,k};\boldsymbol{\theta}_{\mathrm{n}})\right\} \\ \nabla_{\boldsymbol{\delta}_{j,n}} \{g(\boldsymbol{\delta}_{j,n},\boldsymbol{\epsilon}_k)\} \nabla_{\widetilde{\boldsymbol{\omega}}_{j,n,k}} \left\{\log p(\mathbf{y}_{j,n},\widetilde{\boldsymbol{\omega}}_{j,n,k};\boldsymbol{\theta}_{\mathrm{n}})\right\} + \nabla_{\boldsymbol{\delta}_{j,n}} \left\{\mathbb{H}\left[q(\boldsymbol{\omega}_{j,n};\boldsymbol{\delta}_{j,n})\right]\right\} \end{array} \right]$$

is an unbiased, noisy estimator of $\nabla_{\boldsymbol{\Theta},\boldsymbol{\Delta}} \left\{\mathcal{L}(q(\boldsymbol{\Omega}))\right\}$, for any $S \leq N$ and $U \leq J$.

This kind of noisy gradient can be used to find the minimum of a loss function using Stochastic Gradient Descent (SGD) techniques [83], which have numerous advantages. First, as our notation suggests, we can subsample some trials in the whole set in order to maximize the ELBO wrt. both $\boldsymbol{\Theta}$ and $\boldsymbol{\Delta}$. This already amortizes the cost of the optimization, as we do not need to go through the whole dataset to make a step toward the maximum. Second, we do not need to compute the true posterior probability of each

of the sampled trials in order to estimate the gradient of the respective variational parameters $\boldsymbol{\delta}_{j,n}$: by sampling from this approximate posterior and computing the gradient of this sample, we can estimate the gradient of the ELBO wrt. $\boldsymbol{\delta}_{j,n}$.

Finally, as for the VEM approach, there exists a risk of ending in a local optimum. This risk is, however, mitigated by the use of SGD: as only part of the data are used at each time step to climb the likelihood curve, local peaks will usually be avoided (see Paragraph 1.1.2.2.6) because they are caused by a small cluster of data that is partially ignored during this step [84].

Many SGD algorithms have been developed (e.g. [491, 492]). As our aim is not to make an extensive review of their respective advantages and disadvantages, we decided to use the Adam optimizer [493] and refer to the corresponding paper for a detailed algorithmic description and implementation.

2.2.4.3 Inference Network

Flexibility and sparsity are two of the main desired properties of the approximate posteriors q that we want to fit to our dataset: a loss of flexibility will decrease the performance of the model, or equivalently, worsen the unknown KL divergence between the approximate and true posteriors. Sparsity is however the main asset of Variational Inference, and one would wish to find a cheap form of approximate posterior to perform inference in a reasonable amount of time. A solution to reduce the number of variational parameters consists in using a direct, deterministic mapping from the values of the adjacent nodes of factor graph to a given approximate posterior $\boldsymbol{\delta}_{j,n} \triangleq f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_{\mathrm{n}}; \boldsymbol{\rho})$ where $\boldsymbol{\rho}$ is the vector of parameters of the arbitrary function f that achieves this mapping. This leads the approximate posterior to have the form $q_{\boldsymbol{\rho}}(\mathbf{z} \,|\mathbf{y}_{j,n}, \boldsymbol{\theta}_{\mathrm{n}})$ (Figure 2.3c). Our goal has now become to optimize the mapping function parameters $\boldsymbol{\rho}$ instead of the local variational posteriors $\boldsymbol{\delta}_{j,n}$ which it determines.

Amortized Variational Inference has been presented in Paragraph 1.1.2.2.9. In short, it alleviates the need of optimizing each of the local approximate posteriors, which can be problematic for large datasets. It is scalable, and amortizes the storage of each approximate posterior to a single set of network weights.

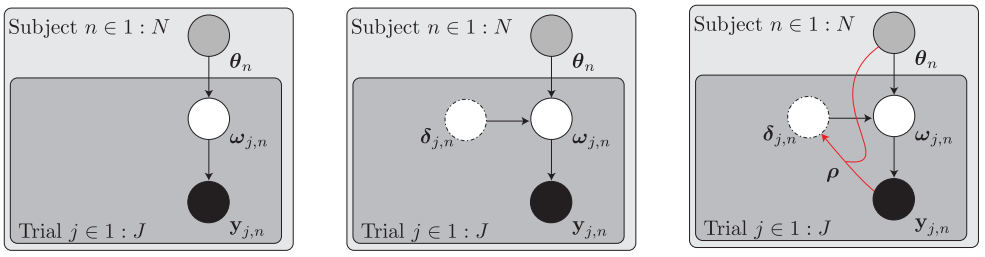
Optimizing the ELBO with an IN is straightforward: using the above definition of the approximate posterior, the noisy approximation to $\nabla_{\boldsymbol{\Theta},\boldsymbol{\rho}} \{\mathcal{L}(q(\boldsymbol{\Omega}))\}$ becomes

$$\widetilde{\nabla}^{\mathcal{L}} \triangleq \frac{1}{SUK} \sum_{n,j,k}^{S,U,K} \left[\begin{array}{c} \nabla_{\boldsymbol{\theta}_\mathrm{n}} \left\{ g\left(f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_\mathrm{n}; \boldsymbol{\rho}), \boldsymbol{\epsilon}_k \right) \right\} \nabla_{\widetilde{\boldsymbol{\omega}}_{j,n,k}} \left\{ \log p(\mathbf{y}_{j,n}, \widetilde{\boldsymbol{\omega}}_{j,n,k}; \boldsymbol{\theta}_\mathrm{n}) \right\} + \\ \nabla_{\boldsymbol{\theta}_\mathrm{n}} \left\{ \mathbb{H}\left[q(\mathbf{z}; f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_\mathrm{n}; \boldsymbol{\rho})) \right] \right\} \\ \nabla_{\boldsymbol{\rho}} \left\{ g\left(f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_\mathrm{n}; \boldsymbol{\rho}), \boldsymbol{\epsilon}_k \right) \right\} \nabla_{\widetilde{\boldsymbol{\omega}}_{j,n,k}} \left\{ \log p(\mathbf{y}_{j,n}, \widetilde{\boldsymbol{\omega}}_{j,n,k}; \boldsymbol{\theta}_\mathrm{n}) \right\} + \\ \nabla_{\boldsymbol{\rho}} \left\{ \mathbb{H}\left[q(\mathbf{z}; f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_\mathrm{n}; \boldsymbol{\rho})) \right] \right\} \end{array} \right] \tag{2.4}$$

where $\widetilde{\boldsymbol{\omega}}_{j,n,k} = g\left(f(\mathbf{y}_{j,n}; \boldsymbol{\rho}), \boldsymbol{\epsilon}_k \right)$. The gradient in Equation (2.4) can be computed easily using an Automatic Differentiation routine if $g(\cdot)$ is differentiable (see Box 1.1.4 for the treatment of $g(\cdot)$ that are not differentiable).

2.2.4.4 Approximate Posterior specifications

We now need to specify a convenient shape for the approximate posterior of the DDM parameters. In Section 2.5.3, we opted for the use of Normalizing Flows [89] as a highly flexible albeit simple distribution shape. We found that, for the DDM, this form of posterior distribution outperformed generally simpler approximate posterior (such as Gaussian distributions) in terms of KL divergence to the log-model evidence.



(A) Full DDM (B) VB-DDM (C) AVI-DDM

FIGURE 2.3: Directed Acyclic Graphical Model (DAG) of the AVI approach to the Maximum-Likelihood (approximate empirical Bayes) Full DDM problem. Black filled circles represent the data (RT and choices), white circles the latent variables, and the greyed circles the subject-specific priors. *a.* Full DDM as a empirical Bayes problem. The aim is to find the value of $\boldsymbol{\theta}_\mathrm{n}$ that maximizes the marginal likelihood $p(\mathbf{y}_{j,n} \mid \boldsymbol{\theta}_\mathrm{n})$, where $\boldsymbol{\omega}_{j,n}$ has been marginalized out. This problem reduces to an empirical Bayes problem. *b.* VB can provide a method to find a lower-bound to the log-marginal probability, by the use of approximate posteriors $q(\boldsymbol{\omega}_{j,n} \mid \boldsymbol{\delta}_{j,n})$. *c.* Amortized Variational Inference (AVI) amortizes the cost of inferring the approximate posterior for each latent variable by using an IN with parameters $\boldsymbol{\rho}$ that takes as input the datapoint and the prior in order to deterministically predict the approximate posterior. Note that, contrary to many similar approaches, here $\boldsymbol{\theta}_\mathrm{n}$ is taken as input to reflect the fact that, with a different prior, the posterior of $\boldsymbol{\omega}_{j,n}$ would have a different shape even if the data $\mathbf{y}_{j,n}$ were identical.

2.2.5 Preliminary summary

So far, we have presented a method to compute the maximum likelihood of $\boldsymbol{\Theta}$ using Variational Inference. Through the use of SGVB, we propose an optimization method that has a limited cost for big datasets: indeed, the number of samples per gradient estimation $S * U * K$ depends more on the complexity of the problem than on the size of the dataset. Also, it alleviates the need to compute the marginal likelihood to get a gradient estimate. Finally, it should be robust to the presence of local optima, thanks to the use of AVI, although this cannot be guaranteed. The use of IN scales down the computational burden of the problem, and ensures a certain similarity between the approximate posteriors of each datapoint. Moreover, it makes it possible to optimize the algorithm online, as the same mapping function as the one used for observed datapoints can be used for datapoints that have not been observed yet.

Comparison of Algorithm 6 and Algorithm 7 (see below) highlights the difference between the VEM method and AVI. Note that, in the former, the exact posterior probability from Equation (2.16) has been replaced by its approximation q whereas in the latter, the update have been formulated in terms of gradient *ascent* for simplicity of the notation.

Of course, the naive implementation presented in Algorithm 7 does not render the full richness of SGVB: the stopping criterion can, for instance, be refined st. the optimization process is stopped when the performance on a validation dataset starts to decrease (early stopping [22]). This is the method we will present in the Result section Section 2.3. Also, the learning rate η can be adapted to match an approximation of a preconditioning matrix, which is indeed what the Adam optimizer, mentioned above, does.

Algorithm 6: VEM algorithm.

Data: Choices and reaction times $\mathbf{Y}$
input: threshold u, maximum number of iteration M
Result: Variational Parameters $\mathbf{\Delta}$, Maximum-likelihood estimate of $\mathbf{\Theta}$

Initialize randomly $\mathbf{\Theta}, \mathbf{\Delta}$;
Set $i = 0$;
Set $ELBO_0 = -\infty$;
repeat
  $i \mathrel{+}= 1$;
  for $1 : N, j \in 1 : J$ **do** `// E step`
    Find $\arg\min_{\boldsymbol{\delta}} -\mathcal{L}(q(\mathbf{z}; \boldsymbol{\delta}))$ `// e.g. using SGD`
  end
  for $1 : N$ **do** `// M step`
    Set ${\boldsymbol{\mu}_n}^* = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_q[\mathbf{z}]$
    Set ${\boldsymbol{\sigma}_n^2}^* = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_q[\mathbf{z}]^2 + \mathbb{V}ar_q[\mathbf{z}] - ({\boldsymbol{\mu}_n}^*)^2$
  end
  Compute $ELBO_i = \mathcal{L}(q(\mathbf{Z}; \mathbf{\Delta}))$;
until $i > M$ *or* $ELBO_i - ELBO_{i-1} < u$;

Algorithm 7: SGVB Amortized Inference

Data: Choices and reaction times $\mathbf{Y}$
input: maximum number of iteration M, SGD learning rate η
Result: Variational Parameters $\mathbf{\Delta}$, Maximum-likelihood estimate of $\mathbf{\Theta}$

Initialize randomly $\mathbf{\Theta}, \mathbf{\Delta}$;
Set $i = 0$;
repeat
  $i \mathrel{+}= 1$;
  for $k \in 1 : S * U * K$ **do**
    Select $1 : N$ randomly;
    Select $j \in 1 : J$ randomly;
    Draw $\boldsymbol{\epsilon} \in p(\boldsymbol{\epsilon})$ randomly;
    Compute an unbiased, noisy estimate of $\widetilde{\nabla}_k^{\mathcal{L}} \approx -\nabla_{\boldsymbol{\rho},\boldsymbol{\theta}} \{\mathcal{L}(q(\mathbf{Z}; \boldsymbol{\rho}))\}$ `// see Equation (2.4)` ;
  end
  Update $\boldsymbol{\rho} = \boldsymbol{\rho} - \eta \frac{1}{SUK} \sum_{k=1}^{SUK} \widetilde{\nabla}_{\boldsymbol{\rho},k}^{\mathcal{L}}$;
  Update $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{1}{SUK} \sum_{k=1}^{SUK} \widetilde{\nabla}_{\boldsymbol{\theta},k}^{\mathcal{L}}$;
until $i > M$;

2.2.6 Recurrent Auto-encoder

We now develop the core idea of this chapter, which is that one can take advantage of the sequential nature of data acquisition to predict the trial-wise values of the DDM parameters.

The model we will develop is based on Variational Auto-Encoders. Auto-Encoders (AE) (Paragraph 1.1.2.2.9), and more specifically Variational AE (VAE), are a large family of models [88] that can be seen as two-side algorithms were one side (the *encoder* or *recognition model*) is the equivalent of what we have called an *IN* so far: it is a (non-)linear mapping with global parameters $\boldsymbol{\rho}$ that determines the shape of the approximate posterior of a local latent variable $\mathbf{z}$ of an arbitrary length d, based on the data $\mathbf{y}_{j,n}$, with

a prior $p(\mathbf{z})$. This variable $\mathbf{z}$ lies between the *encoder* and the *decoder*, which constitutes the *generative model* about which inference is made. Figure 2.4a sketches the structure of a Variational Auto-encoder.

This generative model maps the parameter $\mathbf{z}$ onto the parameters of the distribution of $\mathbf{y}_{j,n}$ ($\boldsymbol{\omega}_{j,n}$ in our case) thanks to a (non-)linear deterministic mapping $\boldsymbol{\omega}_{j,n} = f(\mathbf{z}; \boldsymbol{\phi})$.

This peculiar model class has the advantage that it allows the user to specify the degree of complexity of the inference process she is trying to achieve by adjusting the hyperparameter d. Indeed, VAE ultimately boil down to a (generative) dimensionality reduction algorithms, where the bottleneck is situated at the level of the latent variable $\mathbf{z}$.

Formally, a VAE for a trial indexed j, n is defined by the following formula of the ELBO:

$$\mathcal{L}\left(q(\mathbf{z} \mid \mathbf{y}_{j,n})\right) = \mathbb{E}_{q_{\boldsymbol{\rho}}(\mathbf{z}|\mathbf{y}_{j,n})}\left[\log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \mathbf{z}) - \log q_{\boldsymbol{\rho}}(\mathbf{z} \mid \mathbf{y}_{j,n})\right] \tag{2.5}$$

which recalls Equation (2.3) with the only difference that now an encoder/decoder scheme is used to link the data to the latent variable and the latent variable to the data.

We kept on purpose the subject indices in Equation (2.5), and the hierarchical treatment of this expression will be introduced shortly after. Let us emphasize in Equation (2.5) the dependency of the probability distributions on $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$: these are the only parameters to fit in order to maximize the ELBO. Ideally, one would like to make inference not only about $\mathbf{z}$ but also about $\boldsymbol{\phi}$, as these parameters also belong to the generative model. Often, however, authors treat these parameters in a Maximum-A-Posteriori (MAP) perspective [88], although inference about these parameters is far from being impossible [482, 105]. This problem will be treated in Section 2.2.6.2.

Chung and colleagues [478] have shown that one can include in this model a specific form of regressor that deterministically brings to the current trial the information about the state of the model at the trial before: this approach, known as Recurrent VAE (RVAE), has been the focus of several recent publications [477, 478] due to its efficiency to trade off accuracy with complexity of the inference process. Under this assumption, the ELBO now looks like:

$$\begin{aligned}\mathcal{L}\left(q(\mathbf{z} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1})\right) = \mathbb{E}_{q_{\boldsymbol{\rho}}(\mathbf{z}|\mathbf{y}_{j,n},\mathbf{h}_{j-1})}\left[\log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \mathbf{z} \mid \mathbf{h}_{j-1}) - \log q_{\boldsymbol{\rho}}(\mathbf{z} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1})\right] \\ \text{where } \mathbf{h}_{j-1} = f(\mathbf{y}_{j-1,n}, \mathbf{z}_{j-1,n}, \mathbf{h}_{j-2,n}; \boldsymbol{\phi}_{\mathbf{h}}).\end{aligned} \tag{2.6}$$

In Equation (2.6), we have made explicit the fact that the approximate posterior q and the likelihood $p(\mathbf{y}_{j,n}, \mathbf{z})$ depend on the previous value of $\mathbf{z}$, $\mathbf{y}$ and $\mathbf{h}$ through a hidden state

$\mathbf{h}_{j-1}$. As our notation suggests, $\mathbf{h}$ is also computed using a (non-)linear transformation of these variables, that takes as input a subset of the decoder parameters $\boldsymbol{\phi}$.

This model specification can be understood as follows: we are interested in retrieving a posterior probability distribution of a variable $\mathbf{z}$ that acts as regression weights of regressors $\mathbf{h}$, which brings to the current trial a selection of information about the trials before (either parameter values, external regressors such as experimental conditions, or behavioural measure, RT and choices included). By specifying the length of $\mathbf{z}$ to its minimum, we ensure that each element of $\mathbf{z}$ explains the maximum of the variance of the output parameters of the DDM, $\boldsymbol{\omega}_{j,n}$. Due to the form of the regression we use (i.e. multi-layered perceptron), we can get high flexibility in a space of reduced dimension: very rich patterns can be reflected by such a scheme, even in a low-dimensional space.

Note that we do *not* specify the distribution shape of $\boldsymbol{\omega}_{j,n}$, but the distribution of $\mathbf{z}$, which then maps non-linearly to $\boldsymbol{\omega}_{j,n}$. This contrasts with all DDM implementations we are aware of, where the choice of the DDM parameter prior is mostly arbitrary.

Regarding the choice of a prior for $\mathbf{z}$, one can assume that it is normally distributed in Equation (2.5) with a mean 0 and a covariance I_d, which ensures that the components of $\mathbf{z}$ are somehow independent and, therefore, that each of them captures the maximum variance in the data. However, this choice is not optimal for the model in Equation (2.6), as one would expect the prior of $\mathbf{z}$ to vary from trial to trial. We used therefore the recommendation of [478] and used a form of prior $\{\boldsymbol{\mu}_0^{\mathbf{z}}, (\boldsymbol{\sigma}_0^{\mathbf{z}})^2\} = f(\mathbf{h}_{j-1}; \boldsymbol{\phi}_p)$ where $\boldsymbol{\phi}_p$ is again a subset of the generative model regression weights and biases.

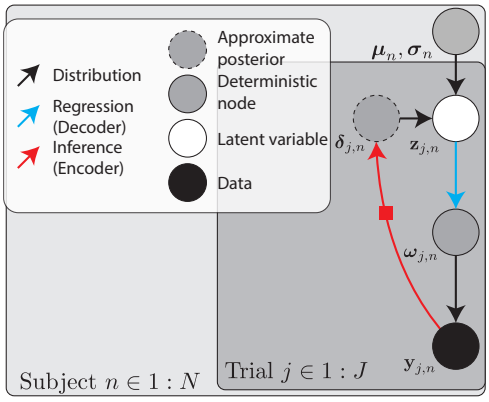
The final model specification we need to consider is the way we will capture inter-individual differences. This can be efficiently achieved using a subject-wise auxiliary latent variable $\boldsymbol{\gamma}_n$ of length d^* with a prior $\mathcal{N}(0, I_{d^*})$ and an approximate posterior $q(\boldsymbol{\gamma}_n \mid \boldsymbol{\upsilon}_n)$. This regressor was introduced in the generative model described above to obtain a set of DDM parameters that depended on the three factors $\mathbf{z}, \mathbf{h}_{j-1}$ and $\boldsymbol{\gamma}_n$:

$$\boldsymbol{\omega}_{j,n} = f\left(\mathbf{z}, \mathbf{h}_{j-1}, \boldsymbol{\gamma}_n; \boldsymbol{\phi}_{\boldsymbol{\omega}}\right). \tag{2.7}$$

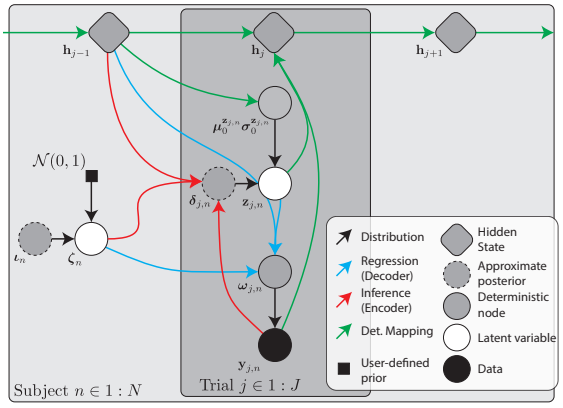
This simple manipulation ensures that inference at the trial level is a compromise between subject-wise behavioural specifics and consistent population-wise behavioural patterns. It goes beyond saying that if some regressor is added to the decoder, it should be added to the encoder too to improve inference about the local variables, which makes us define the following amortized family of approximate posterior over $\mathbf{z}$

$$\mathbf{z} \sim q_{\boldsymbol{\rho}}(\mathbf{z} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1}, \boldsymbol{\gamma}_n). \tag{2.8}$$

It flows directly from Equation (2.7) and Equation (2.8) that, if the dataset has been acquired using various experimental conditions, or if the subject is somehow informed about her performance in the task by the display of a sensory feedback, these regressors can easily be included in the set of the predictors of $\boldsymbol{\omega}$, making this model a compellingly generic tool for inference about the vector $\boldsymbol{\omega}$.



(A) Variational Auto-encoder with Maximal-Likelihood estimate of the top-level hyperprior. Note the difference with Figure 2.3c, where the latent variable directly conditioned the data $\mathbf{y}_{j,n}$.



(B) Recurrent Auto-Encoding Drift Diffusion Model structure. The whole encoder-decoder process is shown for a trial j and a subject n. The model is fully connected, in order to account for dependency between the adjacent levels at the recognition and generative level.

2.2.6.1 Training the RAE-DDM

All the tools we have provided so far can be used in order to train a VAE: the IN, with possibly the normalizing flow that is presented in Section 2.5.3 can be used to find the posterior probability of $\mathbf{z}$ with an amortized cost. SGVB (by the use of the reparameterization trick) could be used to optimize the ELBO, with an unbiased gradient of the form

$$\nabla_{\boldsymbol{\rho},\boldsymbol{\phi}} \left\{ \mathcal{L}\left(q(\mathbf{z} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1})\right) \right\} = \frac{1}{JSK} \sum_{j,n,k}^{J,S,K} \left\{ \begin{array}{c} \nabla_{\boldsymbol{\rho}} \left\{ \log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1}) - \log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1}) \right) \right\} \\ \nabla_{\boldsymbol{\phi}} \left\{ \log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1}) \right\} \end{array} \right. \tag{2.9}$$

where we have made implicit the fact that $\mathbf{z}$ was generated according to a IN approach with $\widetilde{\mathbf{z}}_{j,n,k} = g_{\boldsymbol{\rho}}(\boldsymbol{\delta}_{j,n}, \boldsymbol{\epsilon})$. Using this approach, one would generate K independent random sequences of $\widetilde{\mathbf{z}}_{j,n,k}$ for $j \in 1:J$, but this approach might suffer from several issues: specifically, the error of each sample will propagate through the whole sequence, and the effect of an early "bad" sample will impact all the successive trials, which will severely impair the convergence of the SGD algorithm.

Another issue, closely related to the previous one, regards the sampling of the non-decision time τ: it is hard to put constraints on the output of the generative regression

network, and some samples of τ might fall beyond the t, where no gradient can be computed.

To solve these problems, a typical solution can be to move the sampling average *inside the logarithm function* (see Box 1.1.7), which transforms the ELBO in an Importance Sampling (IS) estimate of the the log-model evidence:

$$\mathcal{L}\left(q(\mathbf{z} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1})\right) \approx \frac{1}{JS} \sum_{j,n}^{J,S} \log \frac{1}{K} \sum_{k} \frac{p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1})}{q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1}))}. \tag{2.10}$$

When used without recurrent predictions, this is known as an Importance Weighted AE (IWAE) [118], and when used in the context of sequential data, this algorithm can be viewed a special member of the Particle Filtering family of algorithms which has been named Filtering Variational Objective (FIVO) [479].

This improved estimate of the ELBO has many advantages, that are detailed in [479]: it allows us to weight each sequence of data to compute the approximate log-model evidence, and also to drop and resample sequences that fail to reflect the data in a meaningful way based on the other sequences. Moreover, this approach produces samples with a much lower variance than the SGVB approach, which can be simply understood by observing that the FIVO approach takes more samples than SGVB, which is usually taken with a local sample size of $K = 1$. Finally, this approach solves the problem of the bounded space of τ: we can assign a log-likelihood of $-\infty$ (i.e. a weight of 0) to the particles k whose τ_k fall beyond the t, and provided that at least one the particles does satisfy the conditions for the log-likelihood to be computed, the gradient of the estimator can be computed. However, it might still be the case that the whole set of particles at a given trial produces a sample $\widetilde{\tau}_{j,n} > \mathrm{t}$. We describe in Section 2.5.3.3 a simple solution to this problem based on Lagrange multipliers.

The complete model, which we name the Recurrent Auto-Encoding Drift Diffusion Model (RAE-DDM) is an application of the algorithm presented in [478, 479] to the problem of DDM fitting, and we refer to these articles for the implementation details. The model is sketched in Figure 2.4b and the training algorithm is displayed in Algorithm 8.

2.2.6.2 Variational Dropout

A final consideration regards the overfitting propensity of RNN and, consequently, the potential overfitting of the RAE-DDM. Dropout is a popular technique to prevent overfitting when training a neural network [481], and has been suggested as a necessary requirement for Boltzman-machine like algorithms [89] such as VAE and RAE. It simply consists in hiding part of the network at each iteration, in order to ensure recursivity in

Algorithm 8: RAE-DDM Training

Data: Choices and reaction times $\mathbf{Y}$

input: maximum number of iteration M, SGD learning rate η

Result: Encoder and decoder weights $\{\boldsymbol{\phi}, \boldsymbol{\rho}\}$, subject approximate posterior distributions $\boldsymbol{\upsilon}_n$

Initialize randomly $\{\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\upsilon}_n\}$;
repeat
  Set $\widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\upsilon}_n} \leftarrow 0$;
  repeat
    Select $n \in 1:N$ randomly;
    Initialize hidden state $\mathbf{h}_0 \leftarrow 0$, accumulated normalized weight $\boldsymbol{\beta}^n \leftarrow 1/K$;
    `/* Forward Pass */`
    Set $\mathcal{L} \leftarrow 0$;
    Draw $\widetilde{\boldsymbol{\gamma}}_n = g(\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n}; \boldsymbol{\upsilon}_n)$ with $\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n} \sim p(\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n})$;
    Increment $\mathcal{L}$ with subject-wise noisy KL approximation $\mathcal{L} \mathrel{+}= \log p(\widetilde{\boldsymbol{\gamma}}_n) - \log q(\widetilde{\boldsymbol{\gamma}}_n \mid \boldsymbol{\upsilon}_n)$;
    for $j \in 1:J$ **do**
      for $k \in 1:K$ **do**
        Draw $\widetilde{\boldsymbol{\gamma}}_n = g(\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n}; \boldsymbol{\upsilon}_n)$ with $\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n} \sim p(\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n})$;
        Map $\boldsymbol{\delta} = f(\boldsymbol{\rho}; \mathbf{y}, \mathbf{h}_{j-1,k}, \widetilde{\boldsymbol{\gamma}}_n)$;
        Draw $\widetilde{\mathbf{z}}_{j,n,k} = g(\boldsymbol{\epsilon}_{\widetilde{\mathbf{z}}_{j,n,k}}; \boldsymbol{\delta})$ with $\boldsymbol{\epsilon}_{\widetilde{\mathbf{z}}_{j,n,k}} \sim p(\boldsymbol{\epsilon}_{\widetilde{\mathbf{z}}_{j,n,k}})$;
        Map $\{\boldsymbol{\mu}_0^{\mathbf{z}}, (\boldsymbol{\sigma}_0^{\mathbf{z}})^2\} = f(\mathbf{h}_{j-1,k}; \boldsymbol{\phi}_p)$;
        Compute $H = \log p\left(\widetilde{\mathbf{z}}_{j,n,k} \mid \boldsymbol{\mu}_0^{\mathbf{z}}, (\boldsymbol{\sigma}_0^{\mathbf{z}})^2\right) - \log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k})$;
        Map $\widetilde{\boldsymbol{\omega}}_{j,n,k} = f(\widetilde{\mathbf{z}}_{j,n,k}, \mathbf{h}_{j-1,k}, \widetilde{\boldsymbol{\gamma}}_n; \boldsymbol{\phi})$;
        Compute unnormalized particle weight $\log \alpha_k^u = \log Wiener(\mathbf{y} \mid \widetilde{\boldsymbol{\omega}}_{j,n,k}) + H$;
      end
      Increment ELBO with weighted samples $\mathcal{L} \mathrel{+}= \log \sum_k \beta_k^n \alpha_k^u$;
      Update accumulated normalized weight $\beta_k^n = \beta_k^n \frac{\alpha_k^u}{\sum_{k'} \alpha_{k'}^u}$;
      If resampling criterion is met, resample $\{\beta_k^n, \widetilde{\mathbf{z}}_{j,n,k}\}$ for $k \in 1:K$;
      for $k \in 1:K$ **do**
        Map $\mathbf{h}_{j,k} = f(\mathbf{y}, \widetilde{\mathbf{z}}_{j,n,k}, \mathbf{h}_{j-1}; \boldsymbol{\phi}_{\mathbf{h}})$
      end
    end
    `/* Reverse Pass */`
    Backpropagate through lines 7 to 22 to get $\widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\upsilon}_n} = \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\upsilon}_n} - \nabla_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\upsilon}_n}\{\mathcal{L}\}$
  until *S sequences have been collected*;
  Update $\boldsymbol{\rho} = \boldsymbol{\rho} - \eta \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho}}$; $\boldsymbol{\phi} = \boldsymbol{\phi} - \eta \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\phi}}$; $\boldsymbol{\upsilon}_n = \boldsymbol{\upsilon}_n - \eta \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\upsilon}_n}$;
until *some convergence criterion is met*;

the network and maximum relevance of each of the inputs. Variational Dropout (VD) [482, 410] is an improved version of the Dropout method where a specific prior and corresponding variational distribution is assigned to each weight of the network. Using VD, we can ensure that the model takes maximum advantage of each of the inputs of the network, and prevents overfitting. Additionally, the use of VD makes the model fully Bayesian, as we make inference not only about $\mathbf{z}$ but also about the weights $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$. VD implementation of the RAE-DDM performed slightly better than the vanilla version on the long-run, and prevented the optimization to fall into spurious local posterior estimates of the DDM parameters. For each set of weights, including those of the GRU RNN algorithm, and according to [494], we sampled $\widetilde{\boldsymbol{\rho}}$ and $\widetilde{\boldsymbol{\phi}}$ according to their posterior distribution before achieving the whole sequence of forward/backward pass described in Algorithm 8, meaning that we used the same sample for each and every trial of the sequence instead of resampling at each trial.

2.3 Experiment

2.3.1 Data Generation Procedure

To provide a proof of concept of the usefulness of accounting directly for the trial-to-trial dependency of the DDM parameters, we generated a series of datasets with a random Gaussian-Inverse-Gamma prior over the DDM parameters (see Section 2.3.1), so that the average mean value of each parameter was variable across datasets. The values were chosen with the following heuristic: we ensured that the chosen parameters generated datasets with a good accuracy in the simulated task (see below) with reasonable RT. We always aim at a high trial-to-trial variability of ξ and w (lower-bounded at 0.3), and the variability of τ and ζ were chosen to be low (bounded at 0). These choices were motivated by the fact that we did not expect the non-decision time to vary more than by a few milliseconds. We also wanted to investigate whether the threshold variability was identifiable for low-to-moderate values of the variance, as this variability is usually assumed to be equal to zero in the DDM literature. The simulated task consisted of a highly generic TAFC task where the subjects needed to choose between a left and a right response to indicate, for instance, the motion direction of dots in a random kinetogram. There was no other manipulation imposed on these datasets. We assumed that the motion direction affected the drift and the drift only. This is a quite common assumption in the DDM literature [9], which is physiologically sensible, as the starting point should be set before the instruction is perceived by the agent [363, 380].

	ζ	ξ	w	τ
$\boldsymbol{\mu}_n$	$\mathcal{N}(1. + 0.2\epsilon_1, 0.1)$	$\mathcal{N}(1.5 + 0.33\epsilon_2, 0.2)$	$\mathcal{N}(0.05\epsilon_3, 0.1)$	$\mathcal{N}(0.3 + 0.1\epsilon_4, 0.01)$
$\boldsymbol{\sigma}_n^2$	$\mathcal{IG}^{-1}(5, u_1 * 0.33)$	$\mathcal{IG}^{-1}(5, 0.3 + u_2)$	$\mathcal{IG}^{-1}(5, 0.3 + u_3)$	$\mathcal{IG}^{-1}(5, u_4 * 0.01)$

TABLE 2.1: Generative distributions at the population level (i.e. dataset-specific). $\epsilon \sim \mathcal{N}(0, 1)$ and $u \sim U(0, 1)$ are random noise variables that controlled the distribution of the parameters at the population level. This manipulation ensured heterogeneity in the datasets.

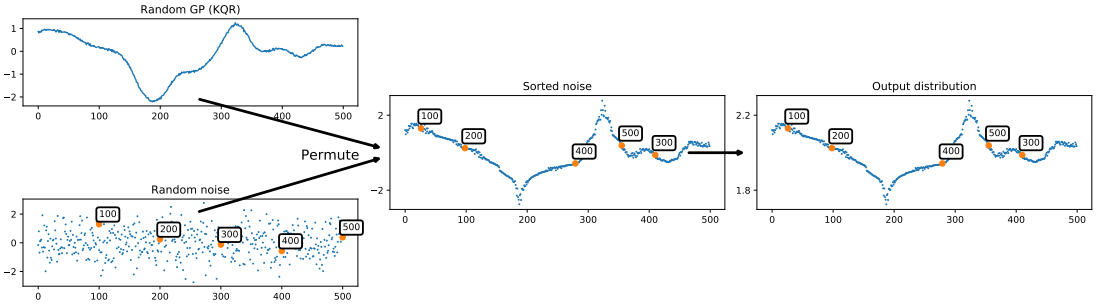


FIGURE 2.5: Data generation procedure for a prior distribution $\mathcal{N}(2, 0.1)$. All parameters were generated according to a diagonal univariate Gaussian distribution. For each parameter, the order of the appearance of each sample was permuted to minimize the distance wrt. a random GP trace (here with a Rational Quadratic kernel).

In order to avoid biasing the data generation process by using a generative model that would only be identifiable using a non-*i.i.d.* based approach, the only manipulation we allowed ourselves to do was to permute the parameters, initially generated according to *i.i.d.* normal distributions (Figure 2.5). This permutation induced specific patterns of cross-correlations between successive parameter values, while ensuring that a fit achieved using the *i.i.d.* assumption was not harmed by the assumption of the data generation procedure.

We proceeded in the following way: first, we generated a random matrix of unit-Gaussian distributed random variables $\boldsymbol{\epsilon} \in \mathbb{R}^{4 \times J} \sim \mathcal{N}(0, I)$. Then, we generated a trace of 4 signals following a Gaussian Process (GP) with a Rational quadratic or a Periodic Kernel [54], parameterized with a population-wise random parameter $a \sim U(0, 1)$ and with an input vector of equally distant variables u situated between -4 and 4. This simple manipulation ensured that the frequency of the parameter variability was mostly identical in the whole population but different between datasets, while keeping a rich family of traces form across subjects. Next, we permuted the elements of each column of $\boldsymbol{\epsilon}$ in order to minimize the Euclidean distance between each of these elements with the correspondent point generated by the GP. Finally, the data were mapped to their desired distribution location by simply applying the following equation:

$$\boldsymbol{\omega}_{j,n} = \boldsymbol{\mu}_n + \boldsymbol{\sigma}_n \odot \boldsymbol{\epsilon}_{j,n}$$

where $\boldsymbol{\sigma}_n^2$ is the Hadamard product operator.

RT and choices were simulated according to [8], where the Wiener process is discretized to an arbitrarily small dt (set to 10^{-6} in our case) and approximated as a binary gambler ruin problem [350]. An alternative method could consist in using the Euler-Maruyama algorithm [495, 496], but this algorithm did not prove to be of greater accuracy than the former, for a higher computational cost.

A total of 16 datasets of 64 subjects each and 500 trials per subject were generated with the same procedure, and different arbitrary random seeds (from 1 to 16) for reproductibility.

Each dataset was fitted to the RAE-DDM, to a hierarchical *i.i.d.*-VB version of the DDM (see below) and finally to an HMC version of this last model coded in Stan [471]. A single chain per dataset was generated, with 1000 iterations for warm-up and 1000 iterations for sampling. Because the threshold was generated with a random noise (as every other parameter), and because it is customary to fit the DDM while discarding the threshold variability, we also fitted the fixed-threshold of the *i.i.d.*-VB version of the DDM and similarly for the HMC version in order to assess how good the classical threshold steadiness assumption worked. For convenience, these fits will be named "classical" in what follows, by contrast with the "full" versions that included the threshold variability. This leads to a total of five fits per dataset.

2.3.2 Model Specification

For each of the $f(\cdot)$ transform in Section 2.2.6, we used a two-layer perceptron, except for the mapping of $\mathbf{h}$ for which a Gated Neural Network can be used in order to prevent the vanishing of the early gradient through the network [497]. We chose arbitrarily the Gated Recurrent Unit (GRU) [498] for the RNN model. An alternative strategy could consist in using a Long-Short Term Memory [499], and we leave the comparison between these various GNN for future studies. For the sake of sparsity and because it is not relevant to the present study, we do not detail these structures here, and refer to the corresponding papers for further details. We chose a hyperbolic tangent function activation function for all these networks, which proved to be of greater accuracy than the Restricted Linear Unit [500, 501] and softplus activators.

The RAE-DDM was configured with 50 nodes per layer in each perceptron, GRU included, for both the *i.i.d.*-VB and RAE-DDM models. For the latter, the length of the latent variable $\mathbf{z}$ was set to 4 for all experiments (so was the length of the subject-specific

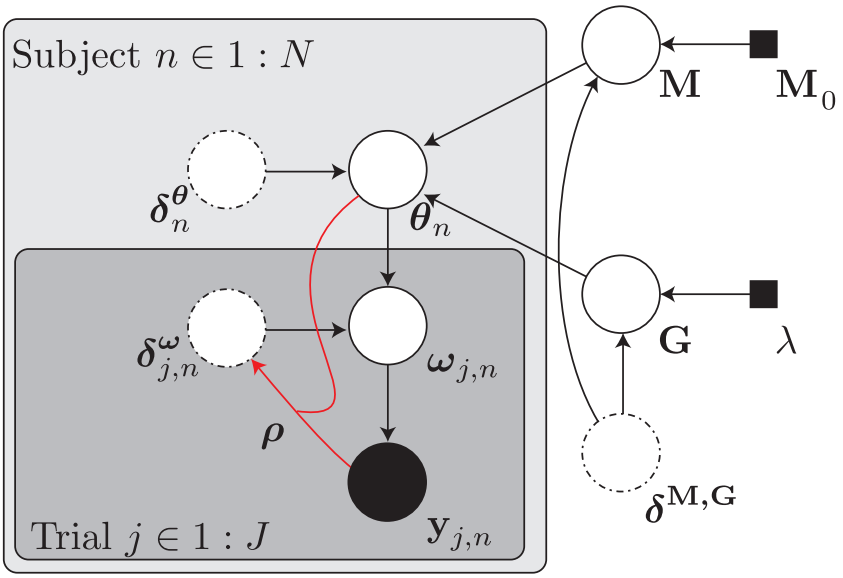


FIGURE 2.6: *i.i.d.*-VB and HMC full hierarchical structure. The DDM parameters were distributed according to a subject-specific prior $\boldsymbol{\theta}_{\mathrm{n}} = \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2\}$. In turn, these parameters were distributed according to a latent Normal $\mathbf{M}$ prior $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2)$ and a latent Inverse-Gamma $\mathbf{G}$ prior $\mathcal{IG}^{-1}(\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0)$ distributions, respectively. The hyperprior of the first distribution was chosen to be a Normal Inverse-Gamma distribution $\mathbf{M}_0$ with parameters $\mathcal{N}(0, 10\boldsymbol{\sigma}_0^2)\,\mathcal{IG}^{-1}(0.1, 0.1)$. The hyperprior of the second distribution was a set of Exponential distributions with shape parameters $\lambda = 0.01$. These values were chosen in order to be weakly informative. This model was identical for the VB and HMC tests. However, in the VB implementation, the approximate posteriors were modelled according to a 4-layers Normalizing Flow with parameters $\boldsymbol{\delta}^*$, where $*$ indicates the latent variable. The parameters $\boldsymbol{\delta}^{\boldsymbol{\omega}}_{j,n}$ were generated thanks to an IN with parameters $\boldsymbol{\rho}$.

latent variables $\boldsymbol{\zeta}_n$), as it showed to provide a sufficient predictive accuracy during piloting. A four component (or layer) normalizing flow was used for the *i.i.d.*-VB and RAE-DDM approximate posteriors.

Each of the VB models were coded in Julia [502], optimized with a noise-insensitive version of the Adam optimizer [493, 503] for 80000 iterations, in parallel with an implementation of the SGD algorithm similar to [504], over 8 CPUs node of various kinds (IvyBridge, Xeon, E5-2650v2, 2.60GHz; Skylake, Xeon, 4116, 2.10GHz; SandyBridge, Xeon, E5-4620, 2.20GHz) with no accelerating device such as GPUs. The code will be made available online.

The *i.i.d.* versions of the DDM (VB or HMC) were all tested with a model identical to the one used to generate the data, i.e. where the drift was influenced by the perceived motion.

2.3.3 Testing for model specifics: Convergence and overfitting assessment

We tested the convergence of the two *i.i.d.*-VB algorithms (with either fixed or variable thresholds) using a validation dataset during training. Convergence rate was fast, and the performance over the validation dataset worsened slightly during the optimization up to a certain plateau. Note that, in piloting experiments, we found that this effect was mitigated by the use of Variational Dropout. This effect was not observed for the *i.i.d.*-VB models.

We calculated *a posteriori* an ad-hoc measure of VB performance named Pareto Smoothed Importance Sampling (PSIS) [56]. PSIS provides a parameter $\hat{k}$ that gives an estimate of the goodness of fit of VB once the optimization procedure has been performed, and also provides a way to refine the accuracy of the posterior estimates (we refer to [278] for a full description of the method). In short, if $\hat{k} < 0.5$, one can infer that VB achieved a good performance for the task. Both *i.i.d.*-VB models showed a very good performance, with a maximum value of $\hat{k}$ of 0.51 for the classical versions of *i.i.d.*-VB and 0.41 for the full versions Figure 2.7. Figure 2.8 compares the convergence of the three VB methods.

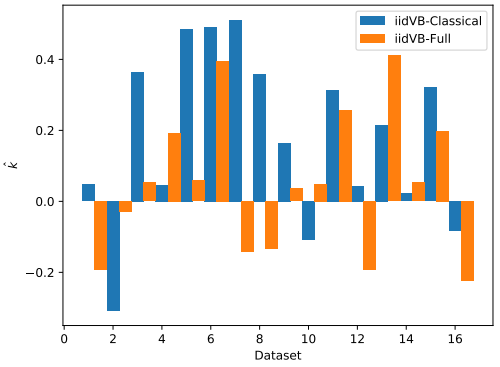


FIGURE 2.7: $\widehat{k}$ value for each dataset.

To compare the three VB models, we ran a vanilla Bayesian Model selection algorithm [132] to investigate the comparative probability that each model generated the data. This method assigns individuals (datasets in this case) to competing models, and can provide an exceedance probability ϕ which can roughly be interpreted as the probability that a model is better than all others. This analysis clearly favored the RAE-DDM with a total of 13.27 datasets (ϕ=0.999) assigned to this model, with respect to 1.39 and 1.34 datasets for the classical and full versions of the *i.i.d.*-VB model, respectively. Interestingly, these results did not favour any of the two *i.i.d.*-VB models.

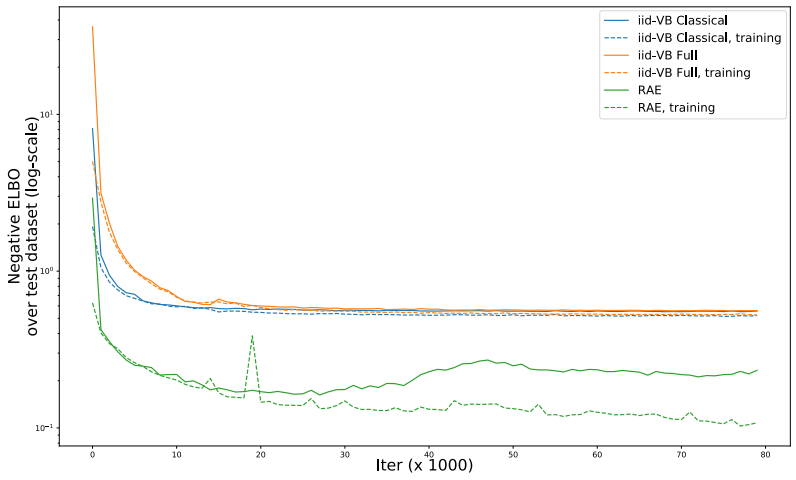


FIGURE 2.8: Convergence rate on a validation dataset (plain) and training dataset (dashed) for the *i.i.d.*-VB and RAE-DDM approaches.

2.3.4 Fit Comparison

We tested the fit of the five models on the training dataset, as HMC does not provide the opportunity to generalize the inference to unseen datapoints. We report here the results in terms of Pearson's correlation coefficients. Euclidean distance (or log-Euclidean distance for the variances) is reported in the Appendix Section 2.5.4. All results are reported in an unbounded space (ζ, w are mapped with the inverse mapping described in Section 2.5.1, and all statistics involving the variances were computed with the log-value of the variance). For the RAE-DDM, means and variances were computed according to Equation (2.16). Finally, all the results that regard the drift rate were corrected for motion direction.

2.3.4.1 Population-wise fit

At the population level, HMC performed well for all parameters average values wrt VB-based methods. From a correlation point of view, the RAE-DDM had an acceptable performance, although it tended to underestimate the variance of each parameter.

Quite interestingly, the HMC-Full performed well in discriminating datasets with high and low variance of the ζ and w, which would not have been possible if the variance of these two parameters were equivalent, as claimed in earlier studies [9]. Moreover, fit of the HMC-Full was always equal or better than the classical version with no threshold variability, which confirms that the Full-DDM with threshold variability is identifiable at the population level.

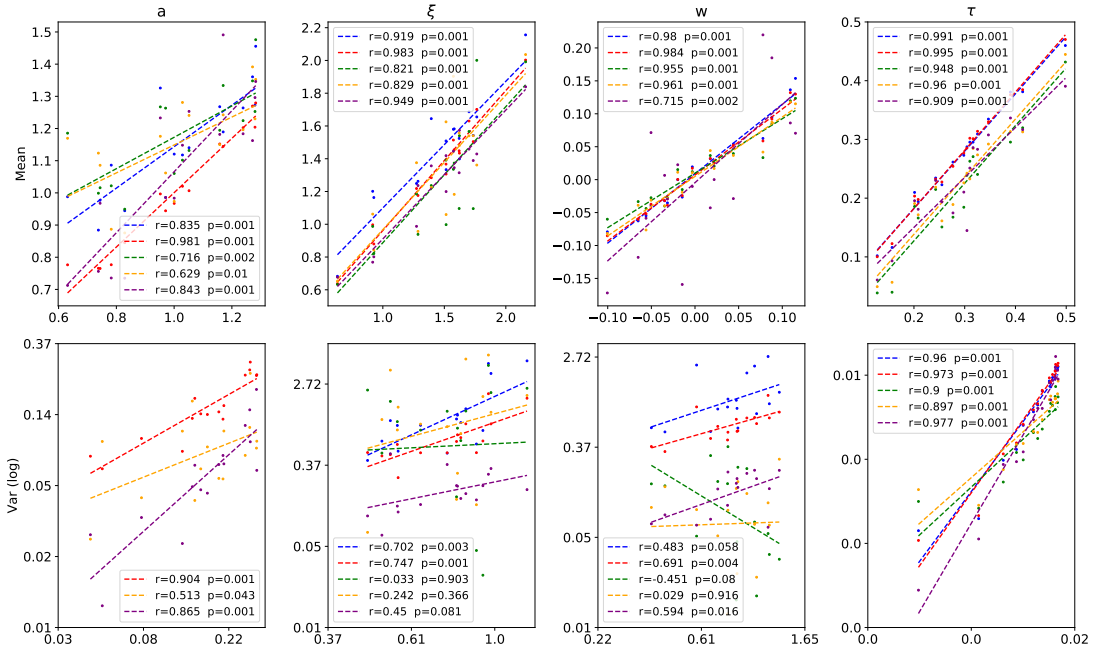


FIGURE 2.9: Population value correlations. **Blue**: HMC-Classical, **Red**: HMC-Full, **Green**: *i.i.d.*-VB -Classical, **Yellow**: *i.i.d.*-VB -Full, **Purple**: RAE-DDM. Fit of the Classical versions of HMC and *i.i.d.*-VB are lacking in the lower left panel, because these models did not account for threshold variability.

2.3.4.1.1 Subject-wise fit All methods provided subject-specific estimates of the mean parameters that correlated well with their true value, with the notable exception of τ, whose fit was acceptable only for the RAE-DDM. In all mean parameter estimations, the RAE-DDM outperformed or was undifferentiable (for w) from other methods.

Even more important is the difference between the evaluation of the subjects' specific trial-to-trial variability of the four DDM parameters by the five models. First, it can be clearly seen in Figure 2.10 that the RAE was able to assess variability with a reasonably high accuracy that was inaccessible to other methods. Second, the *i.i.d.* models (HMC or *i.i.d.*-VB) where the threshold variability were implemented did *not* perform significantly worse than others: on the contrary, the HMC-Full uncovered better the variability of the drift, starting point and non-decision time than his classical counterpart. These observations suggest again that relaxing the *i.i.d.* assumption and including the threshold variability in the DDM might be beneficial.

The *i.i.d.*-VB methods had reasonable performance for mean estimates but were poor at estimating trial-wise variability of the DDM parameters.

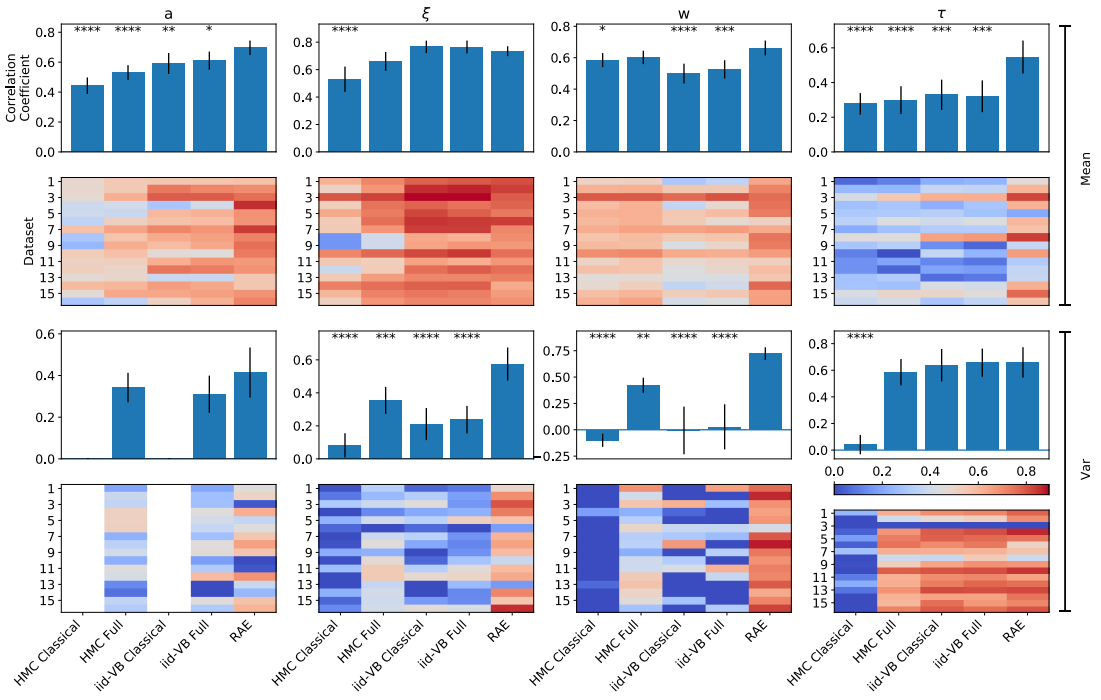


FIGURE 2.10: Subject value correlations. Heat plots show the value of the Pearson correlation coefficient for each dataset, for each model. The bar plots on top of these show the average of the corresponding heat plots. Error bars show the value of one standard deviation. The stars indicate the p-value range of a GLM model that accounted for the value of the true-to-estimate correlation coefficient value as a function of the fitting method (* for $p < 0.5$, ** for $p < 0.01$, *** for $p < 0.001$ and **** for $p < 0.0001$). It can be observed that when the fit of the RAE-DDM was poor, it was usually also the case for other methods.

2.3.4.2 Trial-wise fit

The RAE-DDM performed better at the trial level than each and every other model (Figure 2.11). Overall, *i.i.d.*-VB and HMC performed equally well for all parameters. The correlation at the trial level of the parameter ζ was slightly lower than others. Whether this difference was due to a lower relative variability of this parameter wrt. the others, or to the assumptions about the data-generation process remains to be explored.

Nevertheless, the RAE-DDM was able to assess the value of the ζ at each trial with a relatively good accuracy (Figure 2.12).

2.4 Discussion

In this paper, we present an advanced, scalable, data-driven algorithm to fit the Drift Diffusion Model (DDM) in a fully Bayesian setting that can recover the parameter value at the trial, subject and population levels. This algorithm, named the Recurrent

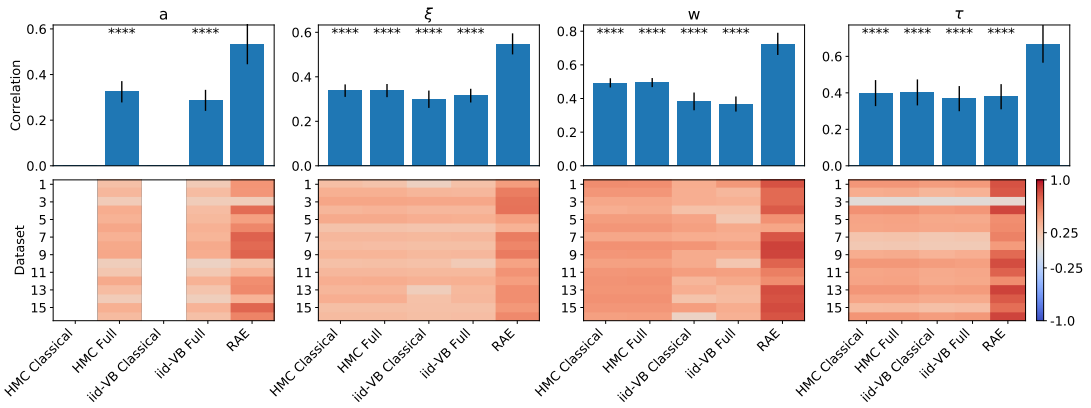


FIGURE 2.11: Trial-value correlations (averaged over subjects for each dataset).

Auto-encoding DDM (RAE-DDM) mainly relies on recent advances in Variational Bayes Inference (VB). The main assumption of this model is that the data are correlated in a forward manner, i.e. that the current value of the data and parameters are conditioned on the preceding value of the data and parameters. It relaxes the *i.i.d.* assumption about the parameter distribution, as well as the assumptions about the distribution of these parameters and, most importantly, about the stationarity of the threshold parameter. We show that the fit of this model outperforms other methods (*i.i.d.*-VB and Hamiltonian Monte Carlo (HMC)), and that inference can be efficiently achieved at every level of the hierarchy with correlation coefficients and distances between true and estimated values that are impossible to obtain with previous approaches.

The RAE-DDM enjoys countless advantages: it can be optimized in parallel or on GPU, it requires a low memory usage thanks to the use of AVI, and, for the same reason, it is generalizable to unseen data. Its structure allows it to be optimized in an online manner. Furthermore, by the use of dimensionality reduction and variational dropout, this model prevents overfitting.

It can account for the effect of regressors of any kind, without requiring to precisely set the relation that ties these regressors to the parameters. Yet, it provides useful information about the parameter values: since the quality of the fit can be expected to be high for a wide range of datasets, the precise estimation of the trial-wise parameters value allows the user to adequately interpret the parameters fit *a posteriori*, based on the trial-to-trial changes in input regressors. This contrasts with classical, non data-driven modeling approaches, where the model structure has to be set beforehand, fitted and then interpreted based on the quality of fit. Therefore, the RAE-DDM compromises informativeness of the model with assumptions about the model specifications.

Importantly, we do not claim that informative models should not be used. We suggest that, if an informative model needs to be tested, which will still be desirable in most

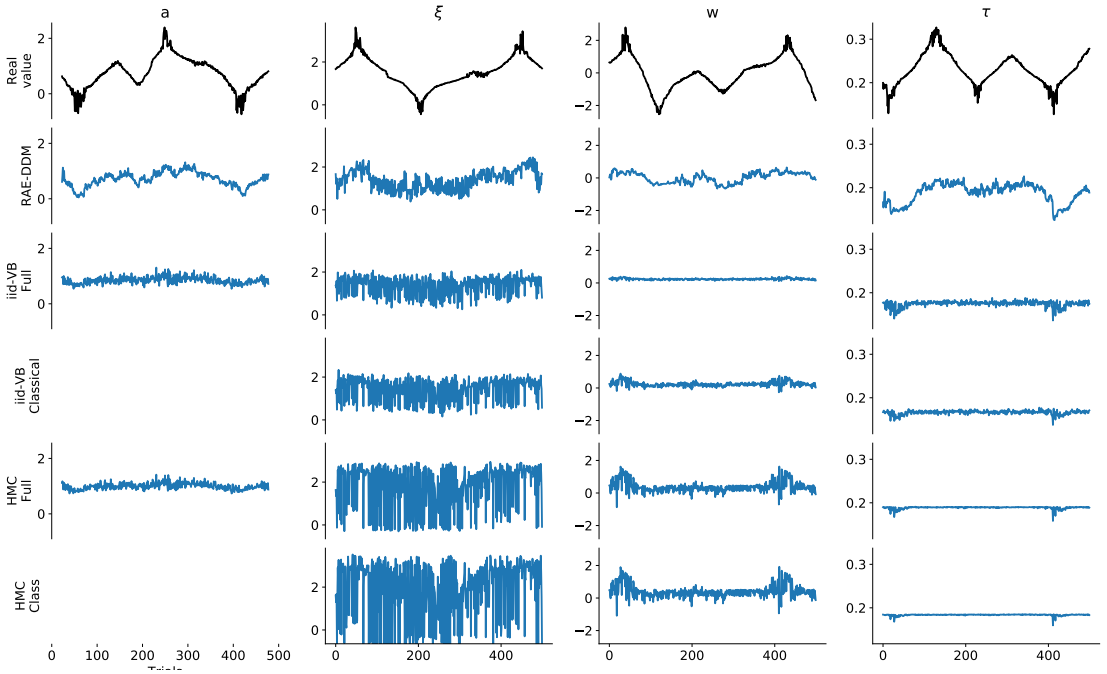


FIGURE 2.12: Randomly selected subject parameter values, and fit for each model. The RAE-DDM estimated the parameter trace with a high accuracy. We emphasize that the fit of the drift (corrected for motion direction in the figure) followed the motion direction pattern that was used to generate the data with a high accuracy, even though this was not directly instructed in the model. Also, other parameters did not follow this pattern, meaning that the algorithm was able to discriminate which parameter value did and did not depend upon the motion direction without explicit user input. This contrasts with the four other methods, where the effect of motion direction was explicitly implemented in the model. It is noteworthy that beside their lack of capacity to account for the pattern of variation of the parameters, the four other methods also failed to provide a good estimate of the parameter variance in this example.

instances, the universality of the RAE-DDM can provide a benchmark of the parameters fit and an upper bound to the model evidence that can be compared with the informative model fit, with closer fits between the two approaches providing empirical evidence in favour of the grounds of the informative model.

It is noteworthy that, contrary to previous approaches [364, 8, 456, 9, 458, 459, 472, 442], here we did not omit any parameter when modelling the trial-by-trial variability. An important contribution of the present paper is indeed to show that the threshold variability can, and hence should, be included in the model trial-to-trial autocorrelations are included in the model. The present method is the first, to our knowledge, in which this specific feature, previously regarded as unidentifiable, has been implemented. Despite the inclusion of this feature in our algorithm, we observed a better fit of the whole set of parameters, showing that its addition did not compromise, but actually improved the identifiability of our model.

Also, while testing the inclusion of the threshold in the models in several datasets fitted with the *i.i.d.*-VB and MCMC approach, we found that not only the RAE-DDM was able to uncover the value of the threshold variability, but that the versions of the HMC or *i.i.d.*-VB models that accounted for threshold variability did not perform worse than others in recovering the parameters of the model. This simple observation should motivate researchers to take with more caution the *a-priori* statement that the threshold variability should be discarded. However, we acknowledge the fact that the exact similarity between the true generative model and the fitted model specifications may have biased these results: as it has been shown by [385], the choice of a different prior distribution for ζ and w would probably have lead to different, less accurate results. Nevertheless, Bayesian Model Selection and (cross-)validation should be used to assess the quality of the fit.

The first innovation of our models is the use of VB to solve the Full DDM problem. Recent advances in VB have made it nearly as precise as MCMC [107, 103, 505, 486], easy to use [101, 102, 490] and adapted to potentially very large problems [76, 482, 105]. We used these advantages of VB to solve two challenging problems: the first one being to recover the DDM parameter variability from trial to trial [9] with a fast algorithm ; the second being to recover an estimate of the trial-to-trial cross-correlation of the DDM parameters and related behavioural data.

Another key feature of the RAE-DDM is its ability to make inference at the trial level, which should be a central required feature of future models cognitive neuroscience [506]. The applications of this feature are countless. For instance, one could assess how parameters change over time following pharmacological or neurophysiological intervention (e.g. dopamine modulation [507] or sub-thalamic nucleus stimulation [384]). It also makes it possible to determine the relationship between neurophysiological data (EEG, fMRI) and trial-by-trial values of DDM parameters, similarly to previous studies in which the post-error slowing phenomenon was related to Frontal Theta bands and the decision threshold was shown to correlate with this signal [508, 384, 509]. The present methods open the door to a more precise study of such phenomena, by allowing precise estimation of the generative parameters in a data-driven manner.

The highly generic structure of the RAE-DDM opens the door to many further developments. A potential improvement of the present approaches would be to broaden the models to other sequential sampling models. Many of them have a closed-form FPT density function formula [510], allowing our models to extend to these other frameworks. Applicable models include the Collapsing Barrier DDM [430, 416] or the Time-varying Ornstein-Uhlenbeck model [511, 512, 435, 328].

Box 2.4.1| From RAE-DDM to RAE-SSM

As noted in Paragraph 1.1.2.2.10, inference can be made about implicit models, where data generation methods exist but a probability density function is not available or hard to compute. Unfortunately, the random variable generator function of SSM is not differentiable, precluding the use of Automatic Variational ABC, which is based on the reparameterization trick. IVI [128] can be used instead. The same logic could be applied here: the value of the ratio

$$\frac{p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1})}{q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{y}_{j,n}, \mathbf{h}_{j-1}))}$$

in Equation (2.10) can be replaced by an estimator

$$\exp\left\{r_{\boldsymbol{\chi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k}; \mathbf{h}_{j-1})\right\} \triangleq \frac{p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1})}{q_{\boldsymbol{\rho}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1}))}$$

whose parameters $\boldsymbol{\chi}$ would be optimized according to Equation (1.29).
Because each and every SSM enjoys the property of being simulable up to a certain accuracy level, an RAE algorithm based on IVI could constitute a highly generic generative model for the SSM models. This is important, because it will ultimately allow us to model tasks more complex and more relevant than TAFC tasks [403].

Also, many appealing features could be implemented with the RAE-DDM to model neurophysiological measures together with behavioural data. Similarly to [14], we could set the physiological measure (e.g. ERP, pupil measure, fMRI measure) to be the output of the generative model.

Finally, the model we propose is scalable at the trial level, but the subject-specific latent variables $\boldsymbol{\zeta}_n$ approximate posterior still needs to be fitted independently. Scalable inference at the subject level (such as the one proposed by [513]) would probably greatly improve the performance and efficiency of the model.

The assumptions of the RAE-DDM should be discussed here. The first and most obvious assumption is that the parameters and the behavioural results are cross-correlated between trials in a forward manner. As discussed above, many behavioural observations are known to reflect complex psychological biases, which justifies the use of previous parameter values as regressors for the current one. A classical way to test which parameter of the DDM is impacted by the previous behaviour would be to explicitly model the supposed cross-dependency among parameters [384]. Thanks to its generic and flexible form, the RAE-DDM automatically selects the model configuration that explains the

most variance up to a predefined degree of complexity (represented by the length of the latent variable $\mathbf{z}$). The results we provide show that the presence of this cross-correlation does not harm the fit, but constrain it adequately, as only very few parameter sequences will explain a specific behavioural pattern across the whole experiment.

The second assumption of the model is that the parameters covary at the trial level in a complex way. In the RAE-DDM, this can be observed by the fact that the latent variable $\mathbf{z}$ is mapped onto $\boldsymbol{\omega}_{j,n}$ with a complex non-linear function. This, again, contrasts with most approaches where the variance of the DDM parameters is assumed to be independent. We think that this latter assumption is likely to be incorrect: for instance, after an error, a subject will likely modify both his bias and threshold value, and therefore these two parameters variability should not reasonably be considered as independent. Once more, the richness of the method we propose does improve rather than impair the fit quality by constraining parameter values to lie within a certain correlated subspace.

In summary, we present a VB-backed hierarchical, sequential and data-driven implementation of the DDM that can retrieve relevant information of the data-generating process at each level. We focused our attention on the possibility of modelling the trial-to-trial dependency of parameters through the development of Recurrent Auto-Encoders. We show that this model is highly flexible, scalable and much more precise than other classical approaches provided that there exists some form of auto-correlation in the values of the parameters and/or data.

2.5 Appendix

2.5.1 Normal prior for the DDM parameters

If we are to put a normal prior on the DDM parameters, it is important to transform the threshold, starting point and non-decision time such that they lie in an unbounded space. Consider the set of bounded parameters $\{\zeta, \xi, z_0, \tau\}$ where $\zeta \in \mathbb{R}^+ \triangleq f^{-1}(\zeta')$, $z_0 \in [0;1] \triangleq g^{-1}(z_0')$, $\tau \in [-\infty; RT] \triangleq h^{-1}(\tau')$ and $\{\zeta', \xi, z_0', \tau'\} \in \mathbb{R}$. We can construct a distribution

$$Wiener'(\mathbf{y}|\zeta', \xi, z_0', \tau') \triangleq Wiener(\mathbf{y}|\zeta, \xi, z_0, \tau)$$

that satisfies this condition for ζ, z_0 and τ by choosing appropriate transforms f, g and h. A natural choice for f and g are respectively the exponential or softplus function – the latter being more advantageous since its gradient is bounded to one, which avoids numerical overflow for samples with large values of the threshold – and some sigmoid function such that $\sigma(z_0') : \mathbb{R} \to [0;1]$.

2.5.2 Full DDM as an Empirical Bayes problem

If we are to use the MLE method for the Full DDM for a set of N subjects with J trials per subject, we therefore need to solve the following optimization problem:

$$\begin{aligned}\boldsymbol{\Theta}^* &= \arg\max_{\boldsymbol{\Theta}} p(\mathbf{Y};\boldsymbol{\Theta}) \\ &= \arg\max_{\boldsymbol{\Theta}} \frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\ell(\mathbf{y};\boldsymbol{\theta}) \\ \text{where } \ell(\mathbf{y};\boldsymbol{\theta}) &= \log\int Wiener(\mathbf{y};\boldsymbol{\omega}_{j,n})p(\boldsymbol{\omega}_{j,n};\boldsymbol{\theta})d\boldsymbol{\omega}_{j,n}\end{aligned} \tag{2.11}$$

For simplicity, we will consider that the DDM parameters all lie in an unbounded space, so that their prior probability can be formulated as a Gaussian with parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2 I\}$, with I being the identity matrix (see Section 2.5.1 for justification and implementation).

In Equation (2.11), we have used the average log-likelihood with the purpose of showing that, for a sufficiently large number of samples NJ, this average log-likelihood takes the form of a sample of the true expected log-likelihood function:

$$\begin{aligned}\frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\ell(\mathbf{y};\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) &\approx \mathbb{E}\left[\ell(\mathbf{y};\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)|\mathbf{y}\right] \\ &= \hat{\ell}(\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2).\end{aligned} \tag{2.12}$$

Since the integral in Equation (2.11) has no analytical form for ζ,z_0 and τ, and since we need to evaluate it for every trial [8], finding the MLE of $\{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2\}$ will require us to compute the gradient of each trial using numerical integration over three dimensions (or two in the classical version of the Full DDM), which we can expect to be computationally expensive.

We now show how Bayesian Statistics arise as a natural way to deal with this problem. The maxima of the likelihood function are located where the value of the gradient of the expression in Equation (2.12) is equal to 0:

$$\begin{aligned}\{\boldsymbol{\mu}_n{}^*, \boldsymbol{\sigma}_n^{2\,*}\} &= \arg\max_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2}\frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\ell(\mathbf{y};\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) \\ \Leftrightarrow \frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2}\ell(\mathbf{y};\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)\Big|_{\boldsymbol{\mu}_n=\boldsymbol{\mu}_n{}^*,\boldsymbol{\sigma}_n^2=\boldsymbol{\sigma}_n^{2\,*}} &= 0\end{aligned} \tag{2.13}$$

We can unpack this expression for each trial and make use of the Fisher's identity [514] to show that:

$$
\begin{aligned}
&\nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2} \log \int Wiener(\mathbf{y}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) d\boldsymbol{\omega}_{j,n} \\
&\qquad = \int \frac{\overbrace{\nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2} p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}^{p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)\nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2}\log p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}}{\int Wiener(\mathbf{y}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} Wiener(\mathbf{y}|\boldsymbol{\omega}_{j,n}) d\boldsymbol{\omega}_{j,n} d\boldsymbol{\omega}_{j,n} \\
&\qquad = \int \nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2} \log p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) \frac{Wiener(\mathbf{y}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}{\int Wiener(\mathbf{y}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) d\boldsymbol{\omega}_{j,n}} d\boldsymbol{\omega}_{j,n} \\
&\qquad = \int \nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2} \log p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) \frac{Wiener(\mathbf{y}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}{p(\mathbf{y}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} d\boldsymbol{\omega}_{j,n} \\
&\qquad = \int \nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2} \log p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2) d\boldsymbol{\omega}_{j,n} \\
&\qquad = \mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} \left[\nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2} \log p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)\right]
\end{aligned}
\tag{2.14}
$$

where we have used the Leibniz rule, the log-derivative trick and finally the Bayes theorem $p(\boldsymbol{\omega}_{j,n}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\omega}_{j,n})p(\boldsymbol{\omega}_{j,n})}{\int p(\mathbf{y},\boldsymbol{\omega}_{j,n})d\boldsymbol{\omega}_{j,n}}$. Equation (2.14) shows that the DDM problem can be solved if we can compute (or approximate) the posterior probability $p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)$. Indeed, since the gradient formulated in Equation (2.14) must be zero at the mode, we can expand this formula and apply it to all trials to show the following:

$$
\begin{aligned}
&\frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J} \mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} \left[\nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2} \log p(\boldsymbol{\omega}_{j,n};\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)\right] &&= 0 \\
&\Leftrightarrow \begin{cases} \frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J} \mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} \left[-\frac{(\boldsymbol{\mu}_n-\boldsymbol{\omega}_{j,n})}{\boldsymbol{\sigma}_n^2}\right] \\ \frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J} \mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} \left[\frac{(\boldsymbol{\mu}_n-\boldsymbol{\omega}_{j,n})^2}{\boldsymbol{\sigma}_n^3} - 1/\boldsymbol{\sigma}_n^2\right] \end{cases} &&= 0.
\end{aligned}
\tag{2.15}
$$

Solving for $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n^2$, we have:

$$
\begin{aligned}
{\boldsymbol{\mu}_n}^* &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} \left[\boldsymbol{\omega}_{j,n}\right] \\
{\boldsymbol{\sigma}_n^2}^* &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} \left[\boldsymbol{\omega}_{j,n}\right]^2 + \mathbb{V}ar_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)} \left[\boldsymbol{\omega}_{j,n}\right] - ({\boldsymbol{\mu}_n}^*)^2.
\end{aligned}
\tag{2.16}
$$

This can be viewed as an Empirical Bayes problem [515, 78] where a MLE of the prior distribution $\mathcal{N}(\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)$ has to be retrieved from the data. Using a Hierarchical Bayesian terminology, $\boldsymbol{\omega}_{j,n}$ is the latent (or local) variable and $\{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2\}$ are the prior or global parameters (Figure 2.3).

Equation (2.16) shows that ${\boldsymbol{\mu}_n}^*$ and ${\boldsymbol{\sigma}_n^2}^*$ will have a closed form formula as long as we can estimate the first and second moments of $p(\boldsymbol{\omega}_{j,n}|\mathbf{y},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)$ for each trial. If the posterior probability of $\boldsymbol{\omega}_{j,n}$ were easy to estimate, this cross-dependency between the posterior probability over $\boldsymbol{\omega}_{j,n}$ and the value of ${\boldsymbol{\mu}_n}^*$ and ${\boldsymbol{\sigma}_n^2}^*$ could suggests an

expectation maximization (EM) algorithm: we could iteratively estimate the posterior $p(\boldsymbol{\omega}_{j,n}|\mathbf{y}, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)$ under the current belief of $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n^2$, and then, we could estimate ${\boldsymbol{\mu}_n}^*$ and ${\boldsymbol{\mu}_n}^*$ using Equation (2.16), until some convergence criterion is reached. This posterior probability does not, however, have a closed-form: additional work is therefore necessary to infer the value of the posterior probability.

2.5.3 Approximate posterior Specification

2.5.3.1 Approximate posterior shape and reparametrization trick

Sampling the first three parameters of the DDM is trivial if these are distributed according to a (possibly diagonal) Gaussian distribution. Let $\{\boldsymbol{\mu}^{\boldsymbol{\omega}}_{j,n}, L^{\boldsymbol{\omega}}_{j,n}\}$ be the mean and lower-Cholesky decomposition of the amortized approximate posterior Gaussian distribution $q(\boldsymbol{\omega}_{j,n} \mid \mathbf{y}, \boldsymbol{\theta}; \boldsymbol{\chi})$, where $\{\boldsymbol{\mu}^{\boldsymbol{\omega}}_{j,n}, L^{\boldsymbol{\omega}}_{j,n}\} = f(\mathbf{y}, \boldsymbol{\theta}; \boldsymbol{\chi})$, then:

$$\widetilde{\boldsymbol{\omega}}_{j,n,k} = \boldsymbol{\mu}^{\boldsymbol{\omega}}_{j,n} + L^{\boldsymbol{\omega}}_{j,n}\, \boldsymbol{\epsilon} \qquad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(0,1).$$

One can easily see that this expression is differentiable and therefore suited for SGVB or IWAE.

Sampling the non-decision time is more tricky: whereas the domain of the prior is unbounded ($\tau_{j,n} \in [-\infty; \infty]$), the domain of the approximate posterior we would wish to derive is not ($\tau_{j,n} \in [-\infty; \mathrm{t}_t]$), as an unbounded probability distribution would generate samples that might give a gradient of 0 to the ELBO, which would in these case take a value of $-\infty$.

One solution to this problem would be to use a Truncated Gaussian ($\mathcal{N}_{-\infty,\mathrm{t}}(\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})$) distribution, where the upper bound would be set to the RT of interest, and the lower bound would be set to $-\infty$. Two main reasons might discourage us from adopting this strategy: the first would be that efficient random number generation algorithms for the truncated Gaussian are not differentiable, because they rely on acceptance-rejection algorithms [516]. However, as the cumulative density function (CDF) of the univariate truncated Gaussian can be computed efficiently, we can compute an estimator of the gradient $\nabla_{\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n}} g(\boldsymbol{\epsilon}; \mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})$ as [99]:

$$\nabla_{\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n}} g(\eta; \mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n}) = \frac{\nabla_{\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n}} P(\tau_{j,n} \mid \mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})}{p(\tau_{j,n} \mid \mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})} \tag{2.17}$$

where P is the CDF of $\mathcal{N}_{(-\infty,\mathrm{t}]}(\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})$ and p the PDF. We refer to the code available online for the derivation of this gradient. Note that, whereas for the multivariate

Gaussian, the gradient and random numbers were computed at the same time, here, one must first generate the random number and then, on this basis, compute the gradient estimator of g given the sample that has been drawn.

The second issue why not to use a univariate truncated Gaussian distribution for τ might be found in the strong mean-field assumption that it requires. Using a multivariate truncated Gaussian is, however, much more challenging that factorizing $q(\tau)$ from the other parameters of the DDM, as the CDF of this distribution is extremely hard to derive and expensive compute. Moreover, current implementations of this function are not differentiable wrt the distribution parameters (see for instance [517]). We will introduce how this issue can be solve by the use of Normalizing Flows short after.

2.5.3.2 Normalizing Flow

If the deterministic mapping from the datapoint to the approximate posterior shape described in Section 2.2.4.3 facilitates the optimization of the ELBO by reducing the size of the problem, it does not make the approximate posterior more flexible. Here, we used a Normalizing Flow (NF) to transform an initial approximate posterior (typically a Gaussian or uniform distribution) through a sequence of deterministic invertible mappings so that its shape gets closer to the true shape of the posterior [103]. The core idea of Normalizing Flows is quite simple: it consists in defining an arbitrarily complex approximate posterior $q^0(\boldsymbol{\omega}_{j,n} \mid \boldsymbol{\delta}, \boldsymbol{\chi})$, where both $\boldsymbol{\delta}$ and $\boldsymbol{\chi}$ are the trial and subject-wise outputs of the inference network.

Consider an initial approximate posterior distribution $q(\boldsymbol{\omega}_0)$. We would like to transform this probability distribution with a deterministic mapping $\boldsymbol{\omega}_K = f(\boldsymbol{\omega}_0, \boldsymbol{\chi})$ with K successive transformations, where $\boldsymbol{\chi} = \{\boldsymbol{\chi}_k \text{ for } k \in 1 : K\}$ are the parameters controlling how f changes the shape of q. In order to keep the volume of $q(\boldsymbol{\omega}_0)$ so that it still integrates to 1 over $\boldsymbol{\omega}_{j,n}$ after the sequence of transformations, the resulting probability density function must be multiplied by the absolute value of the determinant of the Jacobian of the inverse transform wrt $\boldsymbol{\omega}_K$, or alternatively divided by the absolute value of the determinant of the transform wrt $\boldsymbol{\omega}_0$:

$$q(\boldsymbol{\omega}_1) = q(\boldsymbol{\omega}_0) \left| \det \frac{\delta f^{-1}}{\delta \boldsymbol{\omega}_1} \right| = q(\boldsymbol{\omega}_0) \left| \det \frac{\delta f}{\delta \boldsymbol{\omega}_0} \right|^{-1} \tag{2.18}$$

and similarly for the following transformations. After K transformations, the ELBO to the model evidence looks like

$$\mathcal{L}(q(\boldsymbol{\omega}_K)) = \mathbb{E}_{q(\boldsymbol{\omega}_K)}\left[\log p(\mathbf{y}, \boldsymbol{\omega}_K)\right] - \mathbb{E}_{q(\boldsymbol{\omega}_K)}\left[\log q(\boldsymbol{\omega}_0) + \sum_{k=1}^{K} \log \left| \det \frac{\delta f_k}{\delta \boldsymbol{\omega}_{k-1}} \right| \right] \quad (2.19)$$
$$\text{where } \boldsymbol{\omega}_K = f_K \circ f_{K-1} ... f_1(\boldsymbol{\omega}_0)$$

For the sake of simplicity, we chose to use a series of planar transforms with parameters $\boldsymbol{\chi}_k = \{\mathbf{u}_k, \mathbf{w}_k, b_k\}$:

$$f_{\boldsymbol{\chi}_k}(\boldsymbol{\omega}_{k-1}) = \boldsymbol{\omega}_{k-1} + \mathbf{u}_k \tanh(\mathbf{w}_k^T \boldsymbol{\omega}_{k-1} + b_k).$$

This mapping is only valid if $f_k(\boldsymbol{\omega}_{k-1})$ is invertible. This can be easily achieved if $\mathbf{u}$ and $\mathbf{w}$ respect specific conditions, which can be found in [103].

2.5.3.2.1 DDM parameters correlation The issue of accounting for posterior correlation between τ and the other parameters of the DDM finds a convenient solution with the use of planar NF. To account for the covariance between the parameters, we sampled the parameters $\{\zeta, \xi, z_0\}$ according to a diagonal Gaussian distribution, and τ according to a truncated Gaussian distribution as stated above. We then transformed the shape of this factorized distribution by a series of planar transformations constituted of expansions and contractions around the initial draw of the parameters $\widetilde{\boldsymbol{\omega}}_{j,n,k}$. The important point here is that every parameter *except* τ can be transformed in such a way, but always *with* τ *as an input for the transformation*:

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \mathbf{u}_k \psi\left(\mathbf{w}_k{}^T \boldsymbol{\omega}_k + b_k\right)$$

where $\mathbf{u}_k$ and $\mathbf{w}_k$ are two four dimensional vectors, respectively, which are the output of the inference network, with the notable exception that the component of $\mathbf{u}_k$ corresponding to τ is equal to 0, which ensures that τ lies in the space bounded by the RT. Invertibility of this transformation can still be ensured by the algorithm provided by [103]. This form of approximate posterior proved to be efficient in retrieving covariance between τ and the other parameters.

2.5.3.3 FIVO and non-decision-time sampling

The use of the FIVO lead to a convenient sampling algorithm in which we were able to deal with $\widetilde{\tau}_{j,n}$ falling above the RT $\mathrm{t}_{j,n}$.

It might however still be the case that each particle of the FIVO was a degenerated sample. To deal with this problem, we artificially modified the gradients of the encoder and decoder parameters by adding a component to the samples of the ELBO such that, if all particles were degenerated at a trial j, then the SGD optimizer was instructed to correct the network weights to counter this effect. In practice, the ELBO was transformed to look like:

$$\mathcal{L}\left(q(\mathbf{z} \mid \mathbf{y}, \mathbf{h}_{j-1})\right) \approx \frac{1}{JS} \sum_{j,n}^{J,S} \log \frac{1}{K} \sum_{k} \exp \Big\{ \log p_{\boldsymbol{\phi}}(\mathbf{y}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1}) - \log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{y}, \mathbf{h}_{j-1})) - \frac{1}{2} \max(0, \widetilde{\tau}_k - \mathrm{t}_{j,n})^2 \Big\} \tag{2.20}$$

where $\widetilde{\tau}_k$ is the sampled non-decision time of the k particle. This is a simple case of the use of Lagrange multipliers for constraint optimization: in the case where $\widetilde{\tau}_k > \mathrm{t}$ for some but not all particles k, then the weight of these particles was set to 0 because $\log p_{\boldsymbol{\phi}}(\mathbf{y}, \widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{h}_{j-1}) \to -\infty$. Therefore, the Lagrange correction had no impact on the gradient estimate. The only case where this correction had an impact was therefore when all particles satisfied the above condition, in which case the weights were all set to $1/K$ and the gradient used for backpropagation was set to:

$$\nabla_{\boldsymbol{\rho},\boldsymbol{\phi}} \left\{ \mathcal{L}\left(q(\mathbf{z} \mid \mathbf{y}, \mathbf{h}_{j-1})\right) \right\} \approx \frac{1}{JS} \sum_{j,n}^{J,S} \log \frac{1}{K} \sum_{k} \nabla_{\boldsymbol{\rho},\boldsymbol{\phi}} \Big\{ \exp \Big\{ - \log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k} \mid \mathbf{y}, \mathbf{h}_{j-1})) - \frac{1}{2} \max(0, \widetilde{\tau}_k - \mathrm{t}_{j,n} \tag{2.21}$$

2.5.4 Euclidean Distances

We report here the Euclidean distances between the true values of the DDM parameters and their model estimates.

Overall, the RAE-DDM performed better or equally than the HMC approach at the trial and subject level.

At the population level (Figure 2.13), the mean and variance estimates of the RAE-DDM were slightly more biased than the HMC estimates. This confirms the results shown in Figure 2.9 that the RAE-DDM had a low discrimination power between datasets.

At the subject level, however, the RAE-DDM performed equally well or better than the HMC Figure 2.14. The same was true at the trial level Figure 2.15.

We also report here the corresponding correlation plot Figure 2.16 of the example subject trace in 2.12.

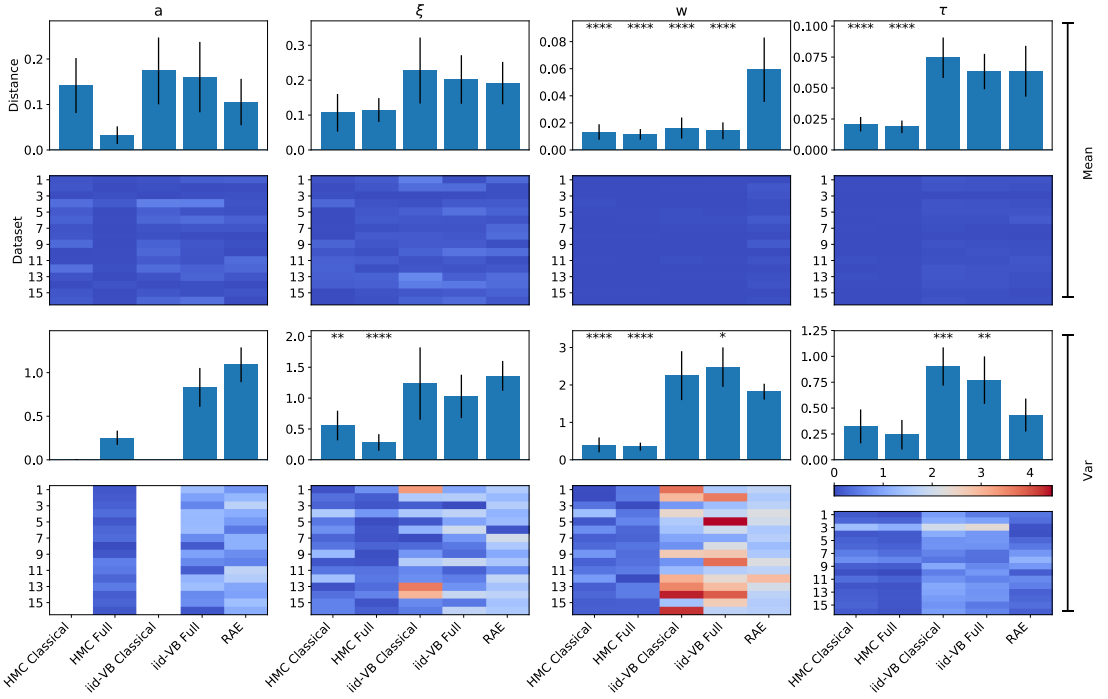


FIGURE 2.13: Population true to estimate (log-)Euclidean distances.

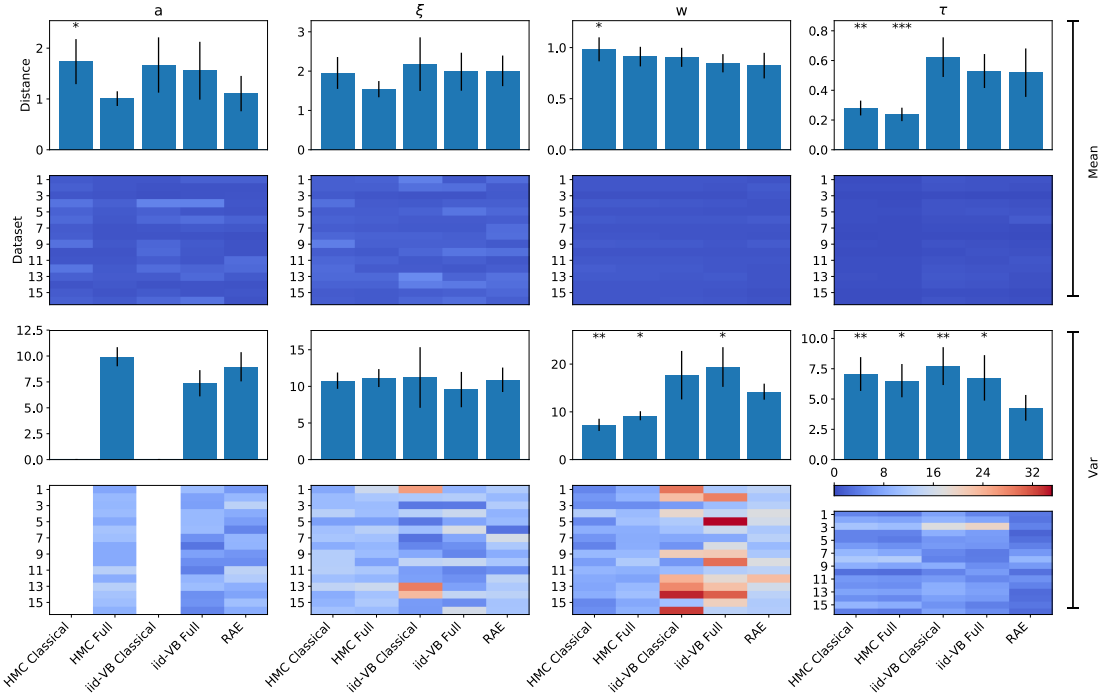


FIGURE 2.14: Subject-wise true to estimate (log-)Euclidean sum of distances.

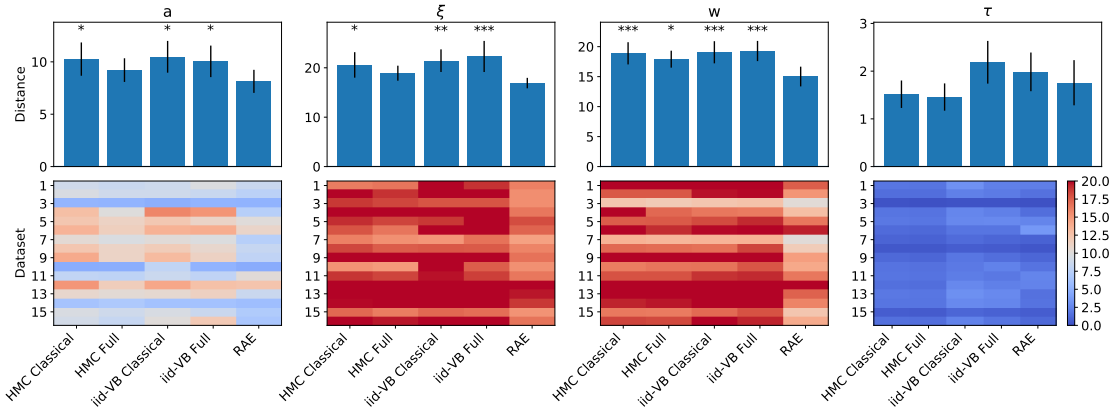


FIGURE 2.15: Trial-wise true to estimate (log-)Euclidean sum of distances. The RAE-DDM performed equally well or better than other approaches.

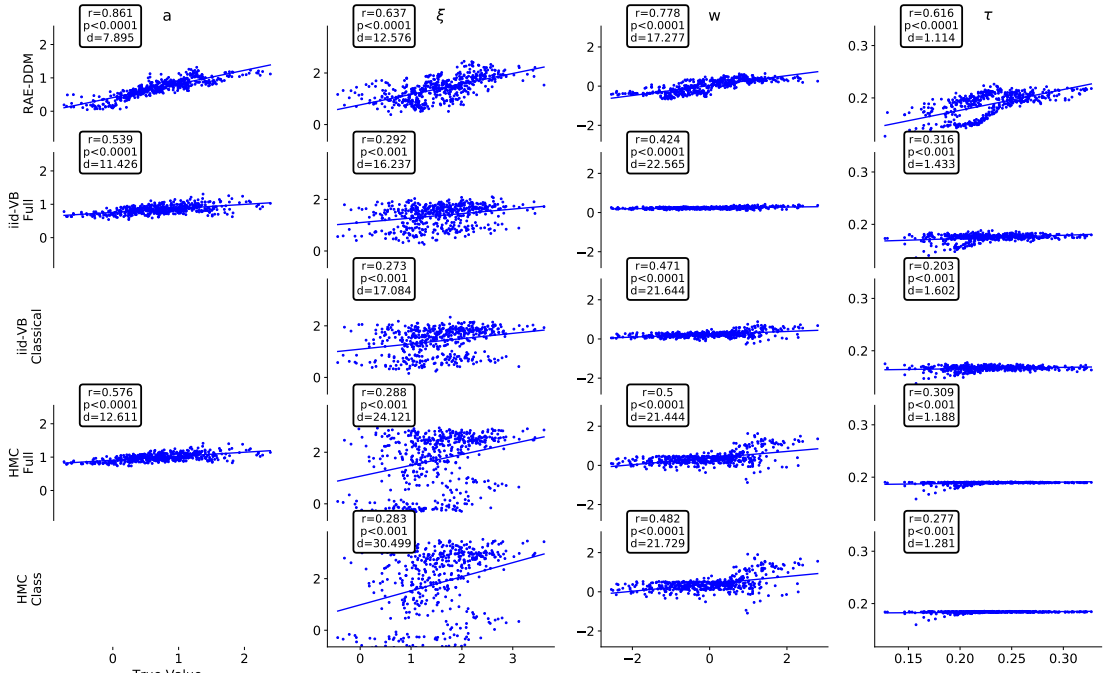


FIGURE 2.16: Correlation and distances of the same random subject as in Figure 2.12. RAE-DDM showed a high accuracy for both these measures, displayed in the upper left corner of each panel. p-values state for the significance of the Pearson's correlation coefficient displayed above.

Chapter 3

The HAFVF

This work has been published in the Proceedings of the 35th International Conference of Machine Learning (ICML).

A common problem in Machine Learning and statistics consists in detecting whether the current sample in a stream of data belongs to the same distribution as previous ones, is an isolated outlier or inaugurates a new distribution of data. We present a hierarchical Bayesian algorithm that aims at learning a time-specific approximate posterior distribution of the parameters describing the distribution of the data observed. We derive the update equations of the variational parameters of the approximate posterior at each time step for models from the exponential family, and show that these updates find interesting correspondents in Reinforcement Learning (RL). In this perspective, our model can be seen as a hierarchical RL algorithm that learns a posterior distribution according to a certain stability confidence that is, in turn, learned according to its own stability confidence. Finally, we show some applications of our generic model, first in a RL context, next with an adaptive Bayesian Autoregressive model, and finally in the context of Stochastic Gradient Descent optimization.

3.1 Introduction

Learning in a changing environment is a difficult albeit ubiquitous task. One key issue for learning in such context is to discriminate between isolated, unexpected events and a prolonged contingency change. This discrimination is challenging with conventional techniques because they rely on prior assumptions about environment stability. When assuming fluctuating context, past experience will be forgotten immediately when an unexpected event occurs, but if that event was just noise, this erroneous forgetting

might be very costly. In less variable contexts, model parameters will tend to change more gradually, thus sometimes missing fluctuations when they happen faster than expected. Most models cover one of the two possibilities, and either gradually adapt their predictions to the new contingency or do it abruptly, but not both.

One classical solution to the problem of change detection is to compare the likelihood of the current observation given the previous posterior distribution with a default probability distribution [518], representing an initial, naive state of the learner. Usually, the mixing coefficient (or forgetting factor) that is used to weight these two hypotheses is adapted to the current data in order to detect and account for the possible contingency change. This mixing coefficient can be implemented in a linear or exponential manner [519]. We will focus here on the exponential case.

In the past decade, several Bayesian solutions to this problem based on the aforementioned strategy have been proposed [520, 521, 522]. However, they usually suffer from several drawbacks: many of them put a restrictive prior on the mixing coefficient, e.g. [520, 523], and cannot account for the fact that an unexpected event is unlikely to be caused by a contingency change if the environment has been stable for a long time.

We propose the Hierarchical Adaptive Forgetting Variational Filter (HAFVF). The core idea of the model is that the the mixing coefficient can be learned as a latent variable with its own mixing coefficient. It is inspired by the observation that animals tend to decrease their flexibility (i.e. their capacity to adapt to a new contingency) when they are trained in a stable environment and that this flexibility is inversely correlated with the training length [225]. We suggest that this strategy may be beneficial in many environments, where the stability of the system identified by a learner is a variable that can be learned as an independent variable with a certain confidence: in certain environments, contingency changes are inherently more less than in others. Although this assumption may not hold in every case, we show that it helps the algorithm to stabilize and discriminate contingency changes from accidents.

Accordingly, we frame our algorithm in a RL framework [162]. We explore how the forward learning algorithm can be extended to the forward-backward case. We show three applications of our model: first in the case of a simple RL task, next to fit an autoregressive model and finally for gradient learning in a Stochastic Gradient Descent (SGD) algorithm.

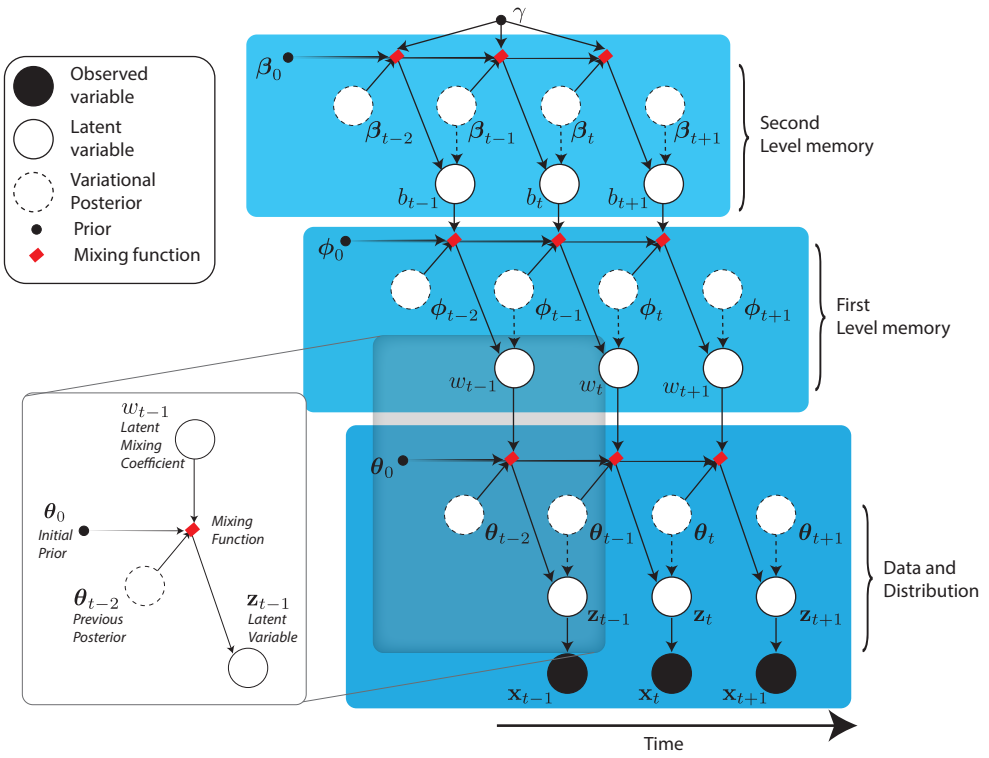


FIGURE 3.1: Directed Graph of the HAFVF. Our contribution with respect to previous works is to consider that the forgetting factor w has a mixture prior with forgetting factor b which acts on w as w acts on $\mathbf{z}$.

3.2 Hierarchical Model

Let $\mathbf{x} = \{x_t\}_{t=1}^{T}$ be a stream of data distributed according to a set of N distributions $\mathbf{p} = \{p_1(\mathbf{x}_{1:n_1} \mid \mathbf{z}_1), ..., p_N(\mathbf{x}_{n_{N-1}+1:n_N} \mid \mathbf{z}_N)\}$, where the change trials $\mathbf{n} := \{n_1, n_2, ...n_N \leq T\} \in \mathbb{Z}^+$ are unknown and unpredictable. We make the following assumptions:

Assumption 1. Let $\{t_1, t_2\} \in T$ and $n_x < t_1 < t_2$, then $p(t_2 \in \mathbf{n}) \leq p(t_1 \in \mathbf{n})$.

Corollary 3.1. *If $m(x_t, p_n)$ is a measure of the relative probability that x_t belongs to p_n wrt p_0, and if $x_2 = x_1$, then $m(x_2, p_n) \geq m(x_1, p_n)$.*

Assumption 1 and Theorem 3.1 state that the probability of seeing a contingency change decreases with time in a steady environment. This might seem counter-intuitive or even maladaptive in many situations, but it is a key assumption we use to discriminate artifacts from contingency change: after a long sequence, the amount of evidence needed to switch from the current belief to the naive belief is greater than after a short sequence. This assumption will lead us to build a model where, if the learner is very confident in his belief, it will take him more time to forget past observations, because he will need

more evidence for a contingency change. Therefore, in this context, the learner aims not only to learn the distribution of the data at hand, but also a measure of the confidence in the volatility of the environment.

Assumption 2. In the set of probability distributions $\mathbf{p}$, all elements have the same parametric form that belongs to the exponential family (EF) and have a conjugate prior that is also from the EF:

$$p_n \in \mathbf{p} \implies p_n(x_t|\,\mathbf{z}_n) = h(x_t)\exp\left\{\mathbf{z}_n^T\,\mathbf{T}(x_t) - A(\mathbf{z}_n)\right\}$$

and $p_n(\mathbf{z}_n) = \exp\left\{\mathbf{T}(\mathbf{z}_n)^T\,\boldsymbol{\theta} - B(\boldsymbol{\theta})\right\}$.

We now focus on the problem of approximating the current posterior distribution of $\mathbf{z}_{n_x}$ given the current and past observations. For clarity, we will make the n subscripts implicit in the following. Let us first focus on the problem of estimating the posterior distribution of $\mathbf{z}$ in the stationary case. After t steps, and given some prior distribution $p(\mathbf{z}\,|\,\boldsymbol{\theta}_0)$, the posterior distribution can be formulated recursively as:

$$p(\mathbf{z}\,|x_t, \mathbf{x}_{<t}) = \begin{cases} \frac{p(x_t|\,\mathbf{z})p(\mathbf{z}\,|\mathbf{x}_{<t})}{p(x_t|\mathbf{x}_{<t})} & \text{for } t > 1 \\ \frac{p(x_t|\,\mathbf{z})p(\mathbf{z}\,|\,\boldsymbol{\theta}_0)}{p(x_t)} & \text{otherwise.} \end{cases}$$

Given the restriction imposed by Assumption 2, this posterior probability distribution has a closed-form expression and can be estimated efficiently.

We enrich this basic model by first formulating the prior distribution of $\mathbf{z}$ at t as a mixture of the previous posterior distribution and an arbitrary prior:

$$p_t(\mathbf{z}\mid\mathbf{x}_{<t};\boldsymbol{\theta}_0, w) = \frac{p_{t-1}(\mathbf{z}\mid\mathbf{x}_{<t})^w p(\mathbf{z}\mid\boldsymbol{\theta}_0)^{1-w}}{Z(w,\mathbf{x}_{<t},\boldsymbol{\theta}_0)}. \tag{3.1}$$

Following Assumption 2, the conjugate distribution $p_{t-1}(\mathbf{z}\mid\mathbf{x}_{<t})$ is also from the EF and reads

$$p_{t-1}(\mathbf{z}\,|\mathbf{x}_{<t}) := \exp\left\{\mathbf{z}^T\,\boldsymbol{\theta}^\xi - \boldsymbol{\theta}^\eta\,A(\mathbf{z}) - B(\boldsymbol{\theta})\right\}$$

where we have expanded $\mathbf{T}(\mathbf{z})$, where $\boldsymbol{\theta} = \{\boldsymbol{\theta}^\xi, \boldsymbol{\theta}^\eta\}$. $\boldsymbol{\theta}^\eta$ is the part of $\boldsymbol{\theta}$ that indicates the effective (prior) number of observations. If p_0 has the same form as $p_{t-1}(\mathbf{z}\mid\mathbf{x}_{<t})$, then the log-partition function Z can be computed efficiently [524]:

$$\begin{aligned} Z(w,\boldsymbol{\theta}_{t-1},\boldsymbol{\theta}_0) = \exp\big\{ &- wB(\boldsymbol{\theta}_{t-1}) - (1-w)B(\boldsymbol{\theta}_0) + \\ &A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right)\big\}. \end{aligned}$$

Note that this result simplifies when combined with the numerator of Equation (3.1):

$$p(\mathbf{z} \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_0, w) = \exp\left\{\mathbf{T}(\mathbf{z})^T \boldsymbol{\vartheta} - B(\boldsymbol{\vartheta})\right\} \tag{3.2}$$

where $\boldsymbol{\vartheta} := w\,\boldsymbol{\theta}_{t-1} + (1-w)\,\boldsymbol{\theta}_0$. The latent variable $w \in [0;1]$ weights the initial prior with the posterior at the previous trial. We incorporate this variable in the set of the latent variables, and, we put a mixture prior on w with a weight b: following this approach, the previous posterior probability of w conditions the current one (similarly to $\mathbf{z}$), together with a prior that is blind to the stream of data up to now. Assuming that x, z and w each can be generated by changing distributions, the joint probability now reads:

$$\begin{aligned} p(x_t, \mathbf{z}, w, b|\mathbf{x}_{<t}; \boldsymbol{\theta}_0, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0, \gamma) := p(x_t \mid \mathbf{z})\times \\ \frac{p_{t-1}(\mathbf{z} \mid \mathbf{x}_{<t})^w p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w}}{Z(w, \mathbf{x}_{<t}, \boldsymbol{\theta}_0)}\times \\ \frac{p_{t-1}(w \mid \mathbf{x}_{<t})^b p(w \mid \boldsymbol{\phi}_0)^{1-b}}{Z(b, \mathbf{x}_{<t}, \boldsymbol{\phi}_0)}\times \\ \frac{p_{t-1}(b \mid \mathbf{x}_{<t})^\gamma p(b \mid \boldsymbol{\beta}_0)^{1-\gamma}}{Z(\gamma, \mathbf{x}_{<t}, \boldsymbol{\beta}_0)} \end{aligned} \tag{3.3}$$

where we have assumed that the posterior probability $p(\mathbf{z}, w, b|\mathbf{x})$ factorizes (Mean-Field assumption), and where $\{\boldsymbol{\theta}_0, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0\}$ are the parameters of the naive, initial prior distributions over $\{\mathbf{z}, w, b\}$ respectively. The model presented in Equation (3.3) is not conjugate anymore, and the posterior probability does not generally have an analytical solution. We therefore introduce a variational posterior to approximate the posterior probability $p(\mathbf{z}, w, b \mid \mathbf{x})$. In short, Variational Inference [525] works by replacing the posterior by a proxy of an arbitrary form and finding the configuration of this approximate posterior that minimizes the Kullback-Liebler divergence between this distribution and the true posterior. This is virtually identical to maximizing the Expected Lower-Bound to the log-model evidence (ELBO).

For simplicity, we use a factorized variational posterior $q_t(\mathbf{z}, w, b) = q_t(\mathbf{z} \mid \boldsymbol{\theta}_t) q_t(w \mid \boldsymbol{\phi}_t) q_t(b \mid \boldsymbol{\beta}_t)$ where each factor has the same form as the prior distribution of its latent variable. Assuming that $q_{t-1}(\cdot) \approx p_{t-1}(\cdot)$ Equation (3.3) conveniently simplifies to:

$$\begin{aligned} p(x_t, \mathbf{z}, w, b|\mathbf{x}_{<t}; \boldsymbol{\theta}_0, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0, \gamma) \approx p(x_t \mid \mathbf{z})\times \\ p(\mathbf{z} \mid w(\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0) + \boldsymbol{\theta}_0)\times \\ p(w \mid b(\boldsymbol{\phi}_{t-1} - \boldsymbol{\phi}_0) + \boldsymbol{\phi}_0)\times \\ p(b \mid \gamma(\boldsymbol{\beta}_{t-1} - \boldsymbol{\beta}_0) + \boldsymbol{\beta}_0). \end{aligned} \tag{3.4}$$

This model is shown in Figure 3.1. In what follows, we will restrict our analysis to the case where w and b are Beta distributed, meaning that the approximate posterior we

will optimize for these two variables will also be a Beta distribution.

3.2.1 Update equations

3.2.1.1 Notation

We first define the following notation: $\mathrm{d}_{\boldsymbol{\theta}} := \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0$ is the difference between the previous approximate posterior and the initial prior. We use $\boldsymbol{\vartheta} := w\,\boldsymbol{\theta}_{t-1} + (1-w)\,\boldsymbol{\theta}_0$ as the weighted prior parameters, and $\widehat{\boldsymbol{\vartheta}}$ as the expectation of $\boldsymbol{\vartheta}$ under $q(w)$. Similarly, $\boldsymbol{\varphi}$ and $\widehat{\boldsymbol{\varphi}}$ are the weighted prior over w and its expectation under q, respectively. Also, we will often abbreviate the summary statistics of $\mathbf{z}$ as $\mathbf{T}(\mathbf{z}) := \begin{bmatrix} \mathbf{z} \\ -A(\mathbf{z}) \end{bmatrix}$.

We now focus on the problem of finding the approximate posterior configuration that maximizes the ELBO. Various techniques have been developed to solve this problem: whereas Stochastic Gradient Variational Bayes [88] and Stochastic Variational Bayes [76] work well for large datasets, more traditional conjugate [69] or non-conjugate [70] Variational Message Passing (VMP) algorithms are better suited for our problem. This technique indeed allows us to derive closed-form update equations that can be sequentially applied to each of the nodes of the factorized posterior distribution until a certain convergence criterion is met. We interpret these results in a Hierarchical Reinforcement Learning framework, where each level adapts its learning rate (LR) as a function of expected log-likelihood of the current observation given the past.

Fortunately, under the form of the approximate posterior we chose and using Conjugate VMP, the variational parameters of the posterior over the latent parameters $\mathbf{z}$ have a simple form given the current value of $\boldsymbol{\phi}_t$ and $\boldsymbol{\beta}_t$. For a number of J observations observed at time t, we have:

$$\boldsymbol{\theta}_t^{\xi} = \widehat{\boldsymbol{\vartheta}}^{\xi} + \sum_{j=1}^{J} \mathbf{T}(x_{tj}) \tag{3.5}$$

$$\boldsymbol{\theta}_t^{\eta} = \widehat{\boldsymbol{\vartheta}}^{\eta} + J \tag{3.6}$$

Equation (3.5) finds an interesting correspondent in the RL literature [162]. Consider the limit case where $\boldsymbol{\theta}_0 = 0$ and $J = 1$ (which is still analytically tractable following Equation (3.2)). As the expectation of a distribution of the EF has the general form $\mathbb{E}_{p(x|\mathbf{z})}\left[\mathbf{T}(x)\right] = dA(\mathbf{z})/d\,\mathbf{z}$, one can derive a similar posterior expectation of $\mathbf{z}$ [526]:

$$\mathbb{E}_{q(\mathbf{z},w)}\left[\mathbf{z}\right] = \frac{\widehat{\boldsymbol{\vartheta}}^{\xi} + \mathbf{T}(x_t)}{\widehat{\boldsymbol{\vartheta}}^{\eta} + 1}$$

Now, replacing $\frac{1}{\boldsymbol{\vartheta}^{\eta}+1}$ by α, the above expression becomes [527]

$$\mathbb{E}_{q(\mathbf{z},w)}\left[\mathbf{z}\right] = Q + \alpha(\mathbf{T}(x_t) - Q) \tag{3.7}$$

where $Q := \frac{\boldsymbol{\theta}^{\xi}_{t-1}}{\boldsymbol{\theta}^{\eta}_{t-1}}$ is the average $\mathbf{z}$ at the time of the previous observation and α is the LR, whose value is inversely proportional to the effective memory $\boldsymbol{\theta}^{\xi}_{t-1}$ and to the current expected value of the forgetting factor $\mathbb{E}_{q(w)}\left[w\right]$[1]. Equation (3.7) is a classical incremental update rule in single-stage RL [212], and our algorithm can be viewed as a special case of such algorithms where the LR is adapted online to the data at hand.

The update equations of $\boldsymbol{\phi}_t$ is, however, not as simple to derive as $\boldsymbol{\theta}_t$, because $p(\mathbf{z} \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_0, w)$ is not conjugate to its Beta prior $p(w \mid \boldsymbol{\phi}_{t-1}, \boldsymbol{\phi}_0, b)$. To solve this problem, we used a Non-Conjugate VMP approach [70]. Briefly, NCVMP minimizes an approximate KL divergence in order to find the value of the approximate posterior parameters that maximizes the ELBO. In order to use NCVMP, the first step is to derive the expected log-joint probability of the model, which we will need to differentiate wrt $\boldsymbol{\phi}_t$ (or, in the case of the approximate posterior update rule for b, $\boldsymbol{\beta}_t$). It quickly appears that part of this expression does not always have an analytical form for common exponential distributions: indeed, the expected value of $\mathbb{E}_{q(w)}\left[B(\boldsymbol{\vartheta})\right]$ is, in general, intractable and needs to be approximated. Expanding the Taylor series of this expression around $\widehat{w} = \mathbb{E}_{q(w)}\left[w\right]$ up to the second order and taking the expectation, we have:

$$\mathbb{E}_{q(w)}\left[B(\boldsymbol{\vartheta})\right] \approx B(\widehat{\boldsymbol{\vartheta}}) + \frac{1}{2}\mathbb{E}_{q(w)}\left[(w-\widehat{w})^2\right] \nabla^2_{\widehat{w}} B(\widehat{\boldsymbol{\vartheta}}) \tag{3.8}$$

Notice that the second term of the sum in Equation (3.8) can be expressed as $\frac{1}{2}\mathbb{V}ar_{q(w)}\left[w\right] \mathrm{d}_{\boldsymbol{\theta}}{}^{T}\, C(\mathbf{T}(\mathbf{z})|\widehat{\boldsymbol{\vartheta}})$ where $C(\mathbf{T}(\mathbf{z})|\widehat{\boldsymbol{\vartheta}})$ is the prior covariance of $\mathbf{T}(\mathbf{z})$. Hence, this penalty term becomes important when the product of the following factors increase: the distance between the previous posterior and initial prior $\mathrm{d}_{\boldsymbol{\theta}}$, the posterior variance of w and the prior covariance of $\mathbf{T}(\mathbf{z})$. This has the effect of favoring values of $\boldsymbol{\phi}_t$ and w that have a low variance, especially when the two proposed distributions, q_{t-1} and p_0, are very distant from each other.

[1] One can easily see that $\mathbb{E}_{q(w)}\left[w\right]$ dictates the memory of the learner. If $J = 1$ and assuming that $\mathbb{E}_{q(w)}\left[w\right]$ is stationary, we have: $\lim_{t\to\infty} \boldsymbol{\theta}^{\eta}_t = \boldsymbol{\theta}^{\eta}_0 + \frac{1}{1-\mathbb{E}_{q(w)}[w]}$

We now derive the update equation for the approximate posterior of w. Let us first define

$$\begin{aligned}
\delta_{\mathrm{L}} &:= \frac{d}{d\widehat{w}} \mathbb{E}_{q(\mathbf{z})} \left[\log p(\mathbf{z} \mid \widehat{\boldsymbol{\vartheta}}) \right] \\
\delta_{\mathrm{C}} &:= -\frac{1}{2} \mathbb{V}ar_{q(w)} \left[w \right] \times \\
&\quad \frac{d}{d\widehat{w}} \, \mathrm{d}_{\boldsymbol{\theta}}{}^{T} \, C(\mathbf{T}(\mathbf{z}) \mid \boldsymbol{\vartheta}) \, \mathrm{d}_{\boldsymbol{\theta}} \\
\delta_{\mathrm{V}} &:= -\frac{1}{2} \, \mathrm{d}_{\boldsymbol{\theta}}{}^{T} \, C(\mathbf{T}(\mathbf{z}) \mid \widehat{\boldsymbol{\vartheta}}) \, \mathrm{d}_{\boldsymbol{\theta}} \times \\
&\quad C(\log w \mid \boldsymbol{\phi})^{-1} \nabla_{\boldsymbol{\phi}} \mathbb{V}ar_{q(w)} \left[w \right].
\end{aligned} \tag{3.9}$$

We obtain the following result:

Proposition 3.2. *Using Algorithm 1 of [70], the update equation for $\boldsymbol{\phi}_t$ has the form:*

$$\begin{aligned}
\phi_t^{\alpha} &= \widehat{\varphi}^{\alpha} + K(\phi_t^{\alpha}, \phi_t^{\beta}) \, \delta_{\mathrm{L}} + K(\phi_t^{\alpha}, \phi_t^{\beta}) \, \delta_{\mathrm{C}} + \delta_{\mathrm{V}}{}^{\alpha} \\
\phi_t^{\beta} &= \widehat{\varphi}^{\beta} \underbrace{- K(\phi_t^{\beta}, \phi_t^{\alpha}) \, \delta_{\mathrm{L}}}_{u_1} \underbrace{- K(\phi_t^{\beta}, \phi_t^{\alpha}) \, \delta_{\mathrm{C}}}_{u_2} + \underbrace{\delta_{\mathrm{V}}{}^{\beta}}_{u_3}
\end{aligned} \tag{3.10}$$

$$\begin{aligned}
\textit{where } K(x, y) &:= \frac{Mx + L(y)y}{(L(x)L(y) - M^2)(x + y)^2} > 0 \\
L(x) &:= \psi_1(x) + M \\
M &:= -\psi_1(\phi_t^{\alpha} + \phi_t^{\beta})
\end{aligned}$$

and $\psi_n(\cdot)$ is the n^{th} order polygamma function.

Proof. Follows directly from Algorithm 1 in [70][2]. □

The update equation in Equation (3.10) can be easily transposed for $\boldsymbol{\beta}_t$.

In Theorem 3.2, we show that the update of ϕ_t can be decomposed in four terms: the first is the (weighted) prior $\boldsymbol{\varphi}$, which acts as a reference for the update.

The second term, u_1, depends upon δ_{L}, the derivative wrt $\widehat{w}$ of the expectation of the log probability $p(\mathbf{z} \mid \widehat{\boldsymbol{\vartheta}})$ over $\mathbf{z}$, times a constant $K(\cdot, \cdot)$. δ_{L} has a simple form:

Lemma 3.3. *The derivative of the first order Taylor expansion of the expected log probability $\log p(\mathbf{z}) := \log p(\mathbf{z} \mid \widehat{\boldsymbol{\vartheta}})$ around $\widehat{w}$ has the form*

$$\delta_{\mathrm{L}} := \left(\left(\mathbb{E}_{q(\mathbf{z})} \left[\mathbf{T}(\mathbf{z}) \right] - \mathbb{E}_{p(\mathbf{z})} \left[\mathbf{T}(\mathbf{z}) \right] \right)^{T} \mathrm{d}_{\boldsymbol{\theta}} \right).$$

[2]The full development can be found in the supplementary materials.

The proof is given in the supplementary materials. The expression of δ_{L} is easily understood as a measure of similarity between the current update of the variational posterior $q_t(\mathbf{z})$ and the previous posterior dependent prior $p(\mathbf{z})$. Note that a rather straightforward result of Equation (3.11) is that $\lim_{\boldsymbol{\theta}_t^{\eta} \to \infty} \delta_{\mathrm{L}} = 0$: as the posterior becomes stronger, the relative change that one can expect tends to zero, and the impact of δ_{L} on the update of $\boldsymbol{\phi}$ can be expected to decrease. This is the behaviour we aimed at: a very strong posterior probability becomes more and more difficult to change as the training time increases.

Note also the opposite sign of the δ_{L} related increment in Equation (3.10) for ϕ_t^{α} and ϕ_t^{β}. This implies that if $\delta_{\mathrm{L}} > 0$, then $u_1^{\alpha} > 0$, and the update of ϕ_t^{α} will tend to increase. The opposite is true for ϕ_t^{β}, showing that the posterior of w effectively deals with the similarity between the current observation and the previous ones.

The third and fourth term of Equation (3.10), u_2 and u_3, are conditioned by the posterior variance of w and the prior variance of $\mathbf{T}(\mathbf{z})$. In brief, they push the update of $\boldsymbol{\phi}$ in a direction that lowers the variance of both $\boldsymbol{\theta}_t$ and $\boldsymbol{\phi}$. We will show in the next section a simple example of the relative contribution that each of these terms has in the update.

An important consideration to make is that the value of $\boldsymbol{\phi}$ must be > 0, which implies that $u_1 + u_2 + u_3 > -\boldsymbol{\varphi}$, a restriction that may be violated in practice, especially for low values of $\boldsymbol{\varphi}$. In such cases, we reset the value of $\boldsymbol{\phi}_t$ to some arbitrary value (typically $\boldsymbol{\phi} \gg 0$) where the above inequality holds, and resume the update loop until convergence or until a certain amount of iterations is reached. Note that NCVMP is not guaranteed to converge, but, as suggested by [70], exponential damping can improve the convergence of the algorithm.

3.2.2 Example: Binary distribution learning

In order to understand better the relative contribution of u_1, u_2 and u_3 to the variational update scheme, we generated a sequence of 200 binary data distributed according to a binomial distribution, whose probability was switching between 0.8 and 0.2 every 40 trials. This distribution can be modelled as a hierarchy of beta distributions, where the first level is a Bernoulli distribution with a conjugate, Beta approximate posterior, and the one or two levels above are both Beta distributions measuring the stability of the level below. We simulated the learning process in three cases:

- A two-layer HAFVF model, where only the posterior over $\mathbf{z}$ could be forgotten (incremental).

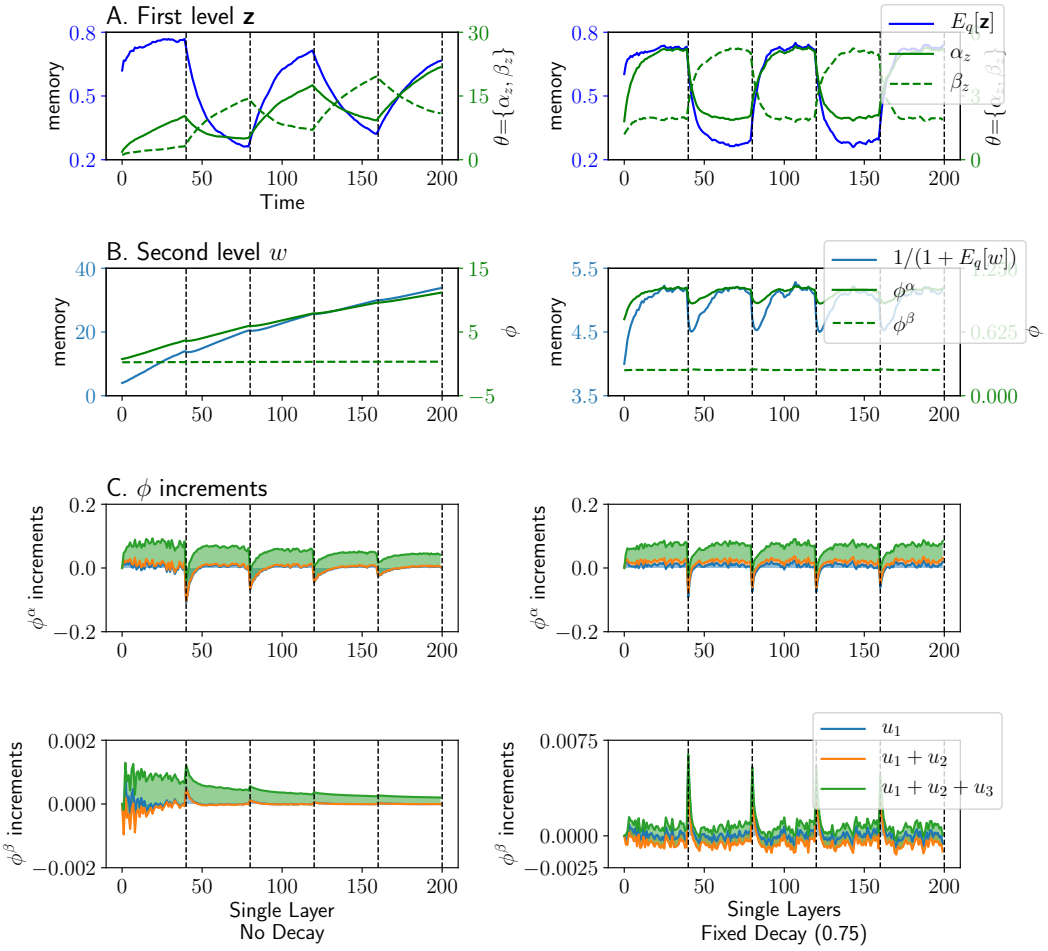


FIGURE 3.2: Binary learning with a single level of forgetting. Incremental (**Left**) and fixed-decay (**Right**) posterior learning of w. **A.** First level learning. The learner looses its capacity to forget as data are observed, because the expected effective memory (**B.**) tend to grow indefinitely when no decay was assumed. **C.** Trial-wise increment caused by u_1, u_2 and u_3. The effects of contingency changes decreased when no decay over w was considered.

- A two-layer HAFVF model, where the posterior of w was being forgotten at a fixed rate (i.e. b fixed to 0.75).
- A three-layer HAFVF model, where the posterior of β was being forgotten at a rate of $\gamma = 0.999$.

In each of these examples, we used the following implementation: the beta prior of the first level was set to $\boldsymbol{\theta}_0 = 1$. The value of $\boldsymbol{\phi}_0$ was set to $\{0.9, 0.1\}$, which showed to be a good compromise between informativeness and freedom to fit the data. If applicable, the top-level was set to $\boldsymbol{\beta} = \{0.75, 0.25\}$.

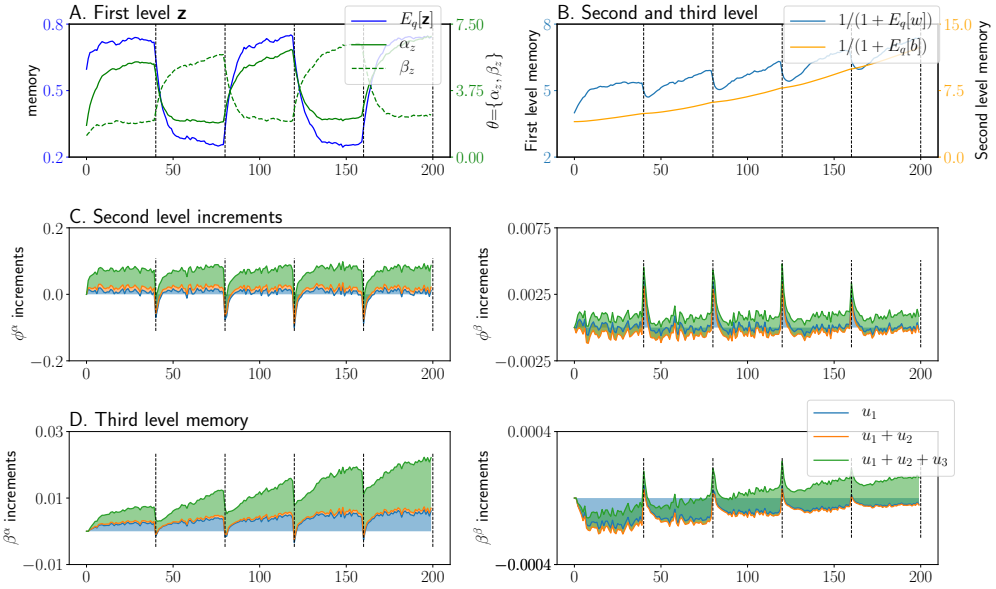


FIGURE 3.3: Binary learning with two levels of adaptive forgetting (w and b) and a third fixed level γ. **D.** is similar to **C.** for the third level updates $q(b)$.

In the first case, the fitting rapidly degenerated, as the memory grew at each trial. Figure 3.2, left column, gives a hint about the reason of this behaviour: each observation decreases the prior covariance $C(\mathbf{T}(\mathbf{z}) \mid \boldsymbol{\vartheta})$, which results in a positive increment for both ϕ_t^α and ϕ_t^β through u_3. This can be viewed as a form of confirmation bias: because the posterior over w and $\mathbf{z}$ are confident about the distribution of the data, they tend to reinforce each other and loose flexibility. Consequently, the impact of the contingency changes decreases as learning goes on. This might seem undesirable (and, in this pathological case, it is indeed the case), but in the case of datasets with outliers it can be very beneficial: a longer training in a stable environment will require a longer and/or stronger sequence of outliers to reset the parameters.

Adding a forgetting factor to the posterior of w can moderate the effect of overtraining. In the case of a fixed-forgetting for the posterior probability of w Figure 3.2, right column, the fitting is much more stable: the model is able to learn and forget the current distribution efficiently with a memory bounded at approximately 5 trials (i.e. $\mathbb{E}_{q(w)}[w] \approx 0.8$). This shows that adding a forgetting over the posterior of w effectively provides the flexibility we aim at: the contingency changes are efficiently detected, and the drop of δ_{L} (through u_1) triggers a resetting of the parameters in the following trials.

In the last case (Figure 3.3), the first level of the model acquires a higher memory than in the second example, due to the ability of the model to adapt the forgetting factor of

w, which relaxes its bound. It is, however, more flexible than the first example. Long-term and flexible learning of the posterior probability of w therefore requires at least two more levels of forgetting (b and γ).

Box 3.2.1| Impact of outliers on learning

To show the how the HAFVF dealt with the presence of noisy observations, we simulated the learning process of a full, 3 levels HAFVF with an uniformative optimistic prior on the environment stability learning a steady signal with one outlier situates at a certain distance of x standard deviation (SD) from the true mean at the observation y.

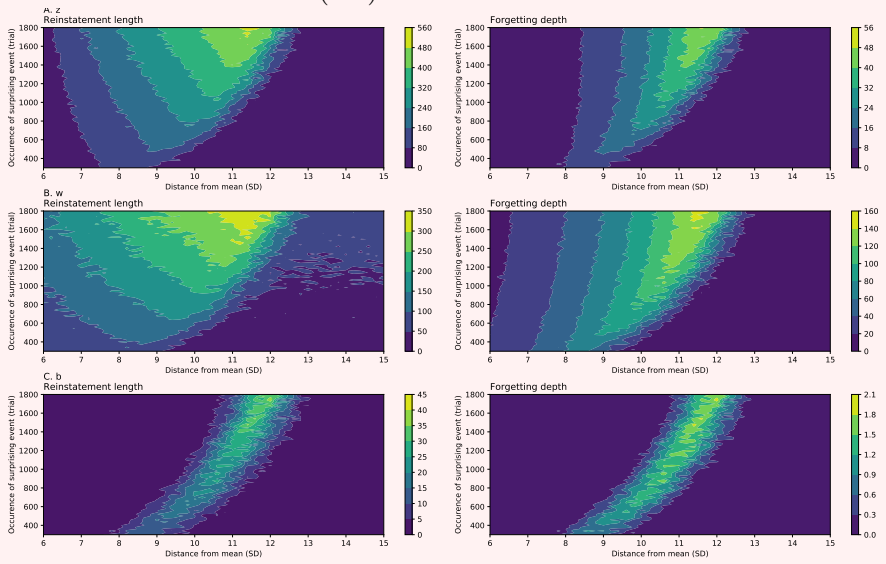


FIGURE 3.4

The distance x was increasing from 6 to 15 SD. The time of occurrence of the unlikely observation ranged between 400 and 1800. The reinstatement length shows the time it took for the learner to reach back the same memory as before, in number of trials. The forgetting gap shows the maximum number of trials forgotten because of the outlier. Results are shown in term of effective memory and efficient memory of both w and b.

The first conclusion we can draw from the simulation we conducted is that the later the unexpected observation occurred, the more it important it had to be to impact learning: this can be seen by the fact that the mode of the distribution of both the reinstatement length and the forgetting depth shifted progressively to higher distances as the training length grew. The second observation is that the hierarchical structure of the HAFVF effectively accounted for the fact that the effect we have just described vanished for higher values of outliers (i.e. forgetting propensity has an bell shape): this can be explained by the fact that, when an event was extremely unlikely, the model had more difficulty to consider that such

a contingency change had occurred.
Overall, these results show that the HAFVF was able to cushion the effect of outlier, especially when these events followed a long training length and/or when they were extremely unlikely given the previously learned posterior distribution.

3.2.3 Forward-Backward algorithm

Let us consider the conjugate posterior of the distribution $p(x|\,\mathbf{z})$ from the EF $p(\mathbf{z})$ when the whole dataset has been observed. For a given $t \in 1:T$, one can derive the posterior probability of $\mathbf{z}$ given x_t as:

$$p(\mathbf{z}\,|x_t, \mathbf{x}_{\neg t}; \boldsymbol{\theta}_0) = \frac{p(x_t \mid \mathbf{z})p(\mathbf{z} \mid \mathbf{x}_{<t}, \mathbf{x}_{>t}; \boldsymbol{\theta}_0)}{p(x_t \mid \mathbf{x}_{<t}, \mathbf{x}_{>t})} \tag{3.11}$$

Given Equation (3.2) and Equation (3.5), if $p(\mathbf{z} \mid \boldsymbol{\theta}_0)$ is the conjugate prior of $p(x_t \mid \mathbf{x})$ and is from the EF, we can substitute the prior $p(\mathbf{z} \mid \mathbf{x}_{<t}, \mathbf{x}_{>t}; \boldsymbol{\theta}_0)$ by $p(\mathbf{z} \mid \widetilde{\mathbf{x}}_{<t}, \widetilde{\mathbf{x}}_{>t}; \boldsymbol{\theta}_0)$, where $\widetilde{\mathbf{x}}_{<t}$ and $\widetilde{\mathbf{x}}_{>t}$ are the effective samples retrieved from the forward and the backward application of the HAFVF on the dataset, respectively. Formally, we have:

$$\begin{aligned} \boldsymbol{\theta}_t^{\xi} &= {}^{f}\boldsymbol{\theta}_t^{\xi} + {}^{b}\boldsymbol{\theta}_t^{\xi} - \sum_{j=1}^{J} \mathbf{T}(x_{tj}) - \boldsymbol{\theta}_0^{\xi} \\ \boldsymbol{\theta}_t^{\eta} &= {}^{f}\boldsymbol{\theta}_t^{\eta} + {}^{b}\boldsymbol{\theta}_t^{\eta} - J - \boldsymbol{\theta}_0^{\eta} \end{aligned} \tag{3.12}$$

where the f and b superscripts index the forward and backward pass, respectively. In offline learning, this technique can increase the effective memory of the approximate posterior distribution just before and after the change trials.

Box 3.2.2| Garbage collection and Automatic Relevance Determination

We briefly introduce two straightforward extensions of the HAFVF: the first is in outlier discrimination and multiple interfering distribution fitting. The second is in Automatic Relevance Determination for autoregressive models.

Garbage collection using linear mixture models The HAFVF presented here above can still fit poorly the data if the model is unfit for the data at hand. For instance, in finance, most of the data that are encountered are well modelled by long-tail distributions, whereas this model applies well to distributions belonging to the EF. We suggest that using a linear mixture model at the lower level can cushion this effect. In the following example, we used the mean-field

VI of a linear mixture model presented in [22] to account for two data streams:

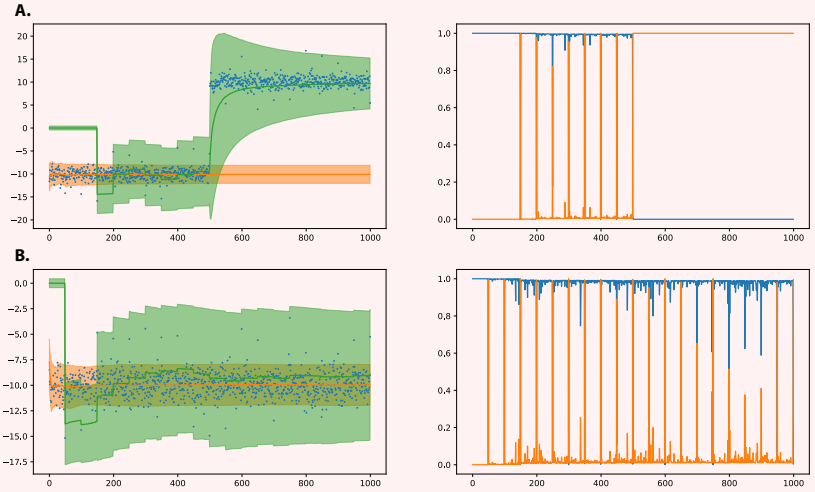


FIGURE 3.5: **A.** The Hierarchical Gaussian Mixture Adaptive Forgetting Variational Filter accounts for outliers (evenly placed once every 50 observations) by assigning them to a "garbage" collector, that becomes the target distribution when the contingency changes. In **B.** it is shown that this learning process is not compromised on the long run.

This approach is able to both detect contingency changes and adapt to them, and to discriminate outliers from data generated by the target process.

Automatic Relevance Determination for AR models When using AR models, it is most of the time not suitable to select arbitrarily the number of components of the model. Selecting the best number of components can be difficult. We develop an Automatic Relevance Determination (ARD) procedure to find the posterior distribution of each order in an online fashion. We define the model

$$p(x_t, \boldsymbol{\delta}, A, \sigma^2, \mathbf{z}, w, b \mid \mathbf{x}_{<t}) = \frac{\prod_{i=1}^{n} p(x_t \mid A_i \bar{\mathbf{x}}_i, \sigma^2)^{z_i}}{Z(x_t, \mathbf{z}, \bar{\mathbf{x}}, A, \sigma^2,)} \tag{3.13}$$

$$\frac{p(A, \sigma^2, \mathbf{z} \mid \mathbf{x}_{<t})^{w} p(A, \sigma^2, \mathbf{z})^{1-w}}{Z(\mathbf{x}_{<t})} \tag{3.14}$$

$$\frac{p(w \mid \mathbf{x}_{<t})^{b} p(w)^{1-b}}{Z(\mathbf{x}_{<t})} p(b \mid \mathbf{x}_{<t}) \tag{3.15}$$

for a strictly incremental posterior over b, where (A, σ^2) is Matrix-Normal-Inverse-Gamma distributed, $\bar{\mathbf{x}}$ is the vector of regressors and $\mathbf{z}$ is Dirichlet-distributed. The weights z_i can be roughly interpreted as the probability that the i^{th} component is necessary to predict the current observation.

Let $\boldsymbol{\eta}$ be the natural parameters of the normal distribution, i.e. $\boldsymbol{\eta} := \begin{bmatrix} \frac{\mu}{\sigma^2} \\ -\frac{1}{2\sigma^2} \end{bmatrix}$. Then the expected log of the first part of (3.13) looks like

$$\mathbb{E}_q \left[\log p(x_t \mid A, \sigma^2, \mathbf{z}) \right] = \mathbb{E}_q \left[\sum_{i=1}^{n} z_i \boldsymbol{\eta}^T \mathbf{T}(x_t) - A \left(\sum_{i=1}^{n} z_i \boldsymbol{\eta} \right) \right].$$

For the univariate normal, we can expand this expression:

$$\begin{aligned} \mathbb{E}_q \left[z_i \boldsymbol{\eta}^T \mathbf{T}(x_t) \right] &= \mathbb{E}_q \left[z_i \right] \mathbb{E}_q \left[\boldsymbol{\eta}^T \right] \mathbf{T}(x_t) \\ \mathbb{E}_q \left[A \left(\sum_{i=1}^{n} z_i \boldsymbol{\eta} \right) \right] &= -\mathbb{E}_q \left[\left(\sum_{i=1}^{n} z_i \eta_{1i} \right)^2 \right] \mathbb{E}_q \left[\frac{1}{4\eta_2} \right] - \frac{1}{2} \mathbb{E}_q \left[\log -2\eta_2 \right] \end{aligned}$$

The expected log-normalizer is quadratic wrt A and $\mathbf{z}$, and requires some tedious developments

$$\begin{aligned} \mathbb{E}_q \left[\left(\sum_{i=1}^{n} z_i \eta_{1i} \right)^2 \right] &= \mathbb{E}_q \left[\left(z_n \begin{pmatrix} A_1 \\ \vdots \\ A_n \end{pmatrix}^T \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{pmatrix} + \right. \right. \\ &\qquad \left. \left. z_{n-1} \begin{pmatrix} A_1 \\ \vdots \\ A_{n-1} \end{pmatrix}^T \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_{n-1} \end{pmatrix} + \ldots + z_1 A_1 \bar{x}_1 \right)^2 \right] \\ &\qquad \text{and given that by definition } \sum_{i=1}^{n} z_i = 1 \\ &= \mathbb{E}_q \left[\left(z_n A_n \bar{x}_n + \sum_{i=n-1}^{n} z_i A_{n-1} \bar{x}_{n-1} + \right. \right. \\ &\qquad \left. \left. \sum_{i=n-2}^{n} z_i A_{n-2} \bar{x}_{n-2} + \ldots + A_1 \bar{x}_1 \right)^2 \right] \\ &\qquad \text{let } c(\mathbf{z}) = \begin{pmatrix} 1 \\ \vdots \\ \sum_{i=n-1}^{n} z_i \\ z_n \end{pmatrix} \text{ be the cumulative sum of } \mathbf{z} \\ &= \mathbb{E}_q \left[\left(A^T \left(c(\mathbf{z}) \odot \bar{\mathbf{x}} \right) \right)^2 \right] \\ &= \sum c(\mathbb{E}_q \left[\mathbf{z} \right] \mathbb{E}_q \left[\mathbf{z} \right]^T + \mathbb{V}ar_q \left[\mathbf{z} \right]) \odot \left(\mu^A \mu^{A^T} + \Sigma_A \right) \odot \bar{\mathbf{x}} \, \bar{\mathbf{x}}^T \end{aligned}$$

where the $c(\cdot)$ operator is applied to each of the dimensions of the input square matrix. As the three factors of the quadratic product in the last expression are symmetric matrices, it has many other formulations, from which the most

economical is probably

$$\bar{\mathbf{x}}^T \left(c(\mathbb{E}_q[\mathbf{z}]\,\mathbb{E}_q[\mathbf{z}]^T + \mathbb{V}ar_q[\mathbf{z}]) \odot \left(\mu^A \mu^{A^T} + \Sigma_A \right) \right) \bar{\mathbf{x}}$$

The full expected log-joint probability at the lower level then reads

$$\mathbb{E}_q\left[\log p(x_t \mid \bar{x}; A, \sigma^2, \delta^z)\right] = -\frac{1}{2}\Bigg(\log 2\pi + \log \beta^\sigma - \psi(\alpha^\sigma) + \frac{\beta^\sigma}{\alpha^\sigma}\left(x_t^2 - 2\mu^{A^T}\left(c(\mathbb{E}_q[\mathbf{z}]) \odot \bar{x}\right) x_t + \mathbb{E}_q\left[\left(\sum_{i=1}^{n} z_i \eta_{1_i}\right)^2\right]\right)\Bigg).$$

The update equation of the Normal-Matrix-Inverse-Gamma approximate posterior distribution are

$$\boldsymbol{\theta}_t = \begin{bmatrix} \boldsymbol{\theta}_t^{\xi} + \sum_{i=1}^{n} z_i \mathbf{T}(x_t) \\ \boldsymbol{\theta}_t^{\chi} + 1 \end{bmatrix}$$

$$\Leftrightarrow \begin{cases} K_t &= \widehat{K(w)} + \mathbb{E}_q\left[c(\mathbf{z})c(\mathbf{z})^T\right] \odot \bar{x}\,\bar{x}^T \\ \alpha_t &= \widehat{\alpha(w)} \\ \mu_t &= K_t^{-1}\left(\widehat{w} K_{t-1}\mu_{t-1} + (1-\widehat{w}) K_0 \mu_0 + x_t c(\mathbb{E}_q[\mathbf{z}]) \odot \bar{x}\right) \\ \beta_t &= \widehat{\beta(w)} + \frac{1}{2}\Big((\mu_{t-1}-\mu_t)^T K_{t-1}(\mu_{t-1}-\mu_t) + \\ &\qquad (\mu_0-\mu_t)^T K_0(\mu_0-\mu_t) + \\ &\qquad (x_t - (c(\mathbf{z}) \odot \mu_t)^T \bar{x})^2 + \bar{x}^T\left(\mathbb{V}ar_q[c(\mathbf{z})] \odot (\mu_t \mu_t^T)\right)\bar{x} \Big) \end{cases}$$

where $\mathbb{E}_q\left[c(\mathbf{z})c(\mathbf{z})^T\right] = c(\mathbb{E}_q[\mathbf{z}]\,\mathbb{E}_q[\mathbf{z}]^T + \mathbb{V}ar_q[\mathbf{z}])$

The following figure shows an example of this mean-field ARD for a one-dimensional autoregressive model:

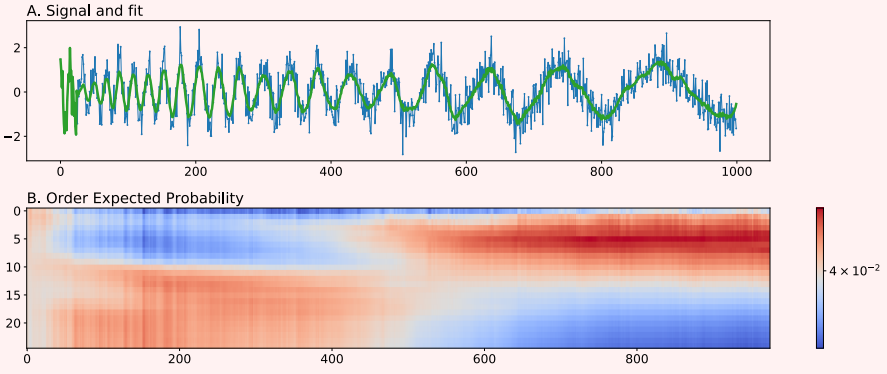


FIGURE 3.6: **A.** Combining the HAFVF-AR model with an Bayesian ARD algorithm shows that the model was able to account for the change of frequency of a synthetic dataset. **B.** The value of $\mathbf{z}$ (y axis), reflecting the importance each of the components shows that components of higher order were more relevant when the frequency was high than when it was low.

3.3 Related work

Change detection is a broad field of research, where no optimal and general solution exists [528]. Consequently, assumptions about the structure of the system can lead to very different algorithms and results.

The Kalman Filter (KF) [529] is a special case of Bayesian Filtering (BF) [530] that has had a large success in the Signal Processing literature due to its sparsity and efficiency. It is, however, highly restrictive and its assumptions need to be relaxed in many instances. Specifically, we are here interested in cases where the evolution of the environment is unpredictable, in contrast with Hidden Markov Models and KF which explicitly give a structure to these changes [22, 522]. One can discriminate two main approaches in order to deal with this problem: the first approach is to use a global approximation of BF such as Particle Filtering (PF) [531, 521, 532], which enjoys a bounded error but suffers from a lower accuracy than other local approximations. The second class of algorithms comprises the Stabilized Forgetting (SF) family of algorithms [528, 522, 533], from which our model is a special case. SF suffers from an unbounded error, but it usually has a greater accuracy for a given amount of resources [531]. Note that SF has been shown to be essential to reduce the divergence between the true posterior and its approximation in recursive Bayesian estimation [534]. As we apply the SF operator to estimate the posterior of $\mathbf{z}$ and the mixture weight w (through the b weighted mixture prior), we ensure that the divergence is reduced for both of these latent variables.

Box 3.3.1| HAFVF and KF

To highlight the specifics of the HAFVF, it might be worth comparing it to the the KF approach to dynamic system modelling.
The basic version of the KF assumes that the distribution of the observed data $\mathbf{x}$ evolves both stochastically and deterministically:

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{C}\,\boldsymbol{\theta}_t, \boldsymbol{\Sigma})$$
$$\boldsymbol{\theta}_t \sim \mathcal{N}(\mathbf{A}\,\boldsymbol{\theta}_{t-1}, \boldsymbol{\Gamma})$$

The "flow matrix" $\mathbf{G}$ models the deterministic evolution of the (unknown) state parameters, and the state covariance $\mathbf{\Gamma}$ models the stochastic evolution these parameters as a random walk. It is then assumed that the observations are drawn from a normal distribution with mean $\mathbf{C}\,\boldsymbol{\theta}_t$ with some transformation matrix $\mathbf{C}$ and known covariance matrix $\mathbf{\Sigma}$.

Under the assumption that $\{\mathbf{\Sigma}, \mathbf{\Gamma}\}$ are known, the posterior distribution of $\boldsymbol{\theta}$ has a closed form formula, and the problem of inferring $\boldsymbol{\theta}$ and predicting future values of $\mathbf{x}$ is tractable. In most cases, some of the variables (flow matrix, state noise, transformation matrix or observation noise) are unknown. In these cases, a maximum likelihood estimate of these matrices can be obtained online, using Expectation Maximization for instance [22].

Also, in its original flavour, the KF assumes that the state noise is constant: with an identity flow matrix, the state evolves with a Gaussian random walk.

The HAFVF relaxes many of these assumptions: crucially, it does not assume that the stochastic evolution of the environment state has a constant noise: it can dynamically adapt to newly observed volatility (either at the observation level – $\mathbf{\Sigma}$ – or at the state level – $\mathbf{\Gamma}$) and forget about past ones.

Also, it does not assume that some parameters are known and others are not. Instead, it makes the implicit assumption that given some initial prior distribution over the state variables (mean and variance of the observations for instance) and the environmental volatility one can derive an approximate inference algorithm that outputs the posterior distribution of the state variables under the past history.

Finally, the KF subordinates the problem of estimating the posterior covariance to the problem of estimating the observation and state volatility. The HAFVF estimates all of these quantities at the same time, but subordinates this process to a prior belief of the environmental evolution.

It might be argued that both models require some prior knowledge: the KF requires prior knowledge of the observation and state noise, the HAFVF requires a prior probability distribution of the state value and noise and environmental volatility. There is, however, a crucial difference between those two algorithms, which is that the HAFVF is able to forget (i.e. to reset its posterior belief to its prior distribution), whereas the KF is not.

Even though our model is described as a Stabilized Exponential Forgetting [519] algorithm and is well suited for signal processing, it can be generalized to models where there is no prediction of future states (e.g. smoothing of a signal, temporal difference learning etc.) Also, it overcomes other methods in several following ways:

First, it uses a Beta prior on the mixing coefficient. This is unusual (but not unique [535]), as most of previous approaches used a truncated exponential prior [536, 523] or a fixed, linear mixture prior that account for the stability of the process [521]. In Linear SF, a Bernoulli prior with a Beta hyperprior has been proposed for the mixture weight [533]. Our approach is designed to learn the posterior probability of the forgetting factor in a flexible manner. We show that this posterior probability depends upon its own (and possibly a mixture of) prior distribution and upon the prior covariance of the model parameters $C(\mathbf{T}(\mathbf{z}) \mid \widehat{\boldsymbol{\vartheta}})$. This makes the change detection more subtle than an all-or-none process, as one might observe with a Bernoulli distribution. It also enables us to accumulate evidence for a change of distribution across trials, which can help to discriminate outliers from real, prolonged contingency changes. This is, to our knowledge, an entirely novel feature in the adaptive forgetting literature.

The second important novelty of our model lies in its hierarchical learning of the environment stability. This is somehow similar to the Hierarchical Gaussian Filter (HGF) [537, 538]. The present model is, however, much more general, as the generic form we provide can be applied to several members of the EF. Also, although the KL divergence (error term) of our model is not bounded in the long run, it can be efficiently applied to a large subset of datasets and models, whereas the HGF often fails to fit processes that are highly stationary, with many datapoints and/or with abrupt contingency changes.

3.4 Experiments

The HAFVF was coded in the Julia language [502] using a Forward automatic differentiation algorithm [539] for the NCVMP for the RL and AR parts of this section, and using an analytical gradient for the SGD part.

3.4.1 Smoothing

We first look at the behaviour of the model in the simple case of estimating the current distribution of a random variable sampled from a moving distribution. We simulated two sequences of 2x200 datapoints where each pair of points was generated according to the same multivariate normal distribution with mean $\mu_1 = \{-2, +2\}$ and $\mu_2 = \{+2, -2\}$. We then added an independent random walk to these means.

We applied the Forward-Backward (FB) version of the HAFVF to these datasets. We used the same Normal Inverse Wishart prior for both of these results ($\mu_0 = 0$, $\kappa_0 = 0.1$, $\eta_0 = 3$, $\Lambda_0 = I$). The prior over w was manipulated to include a high confidence ($\phi_t^\alpha = 9$,

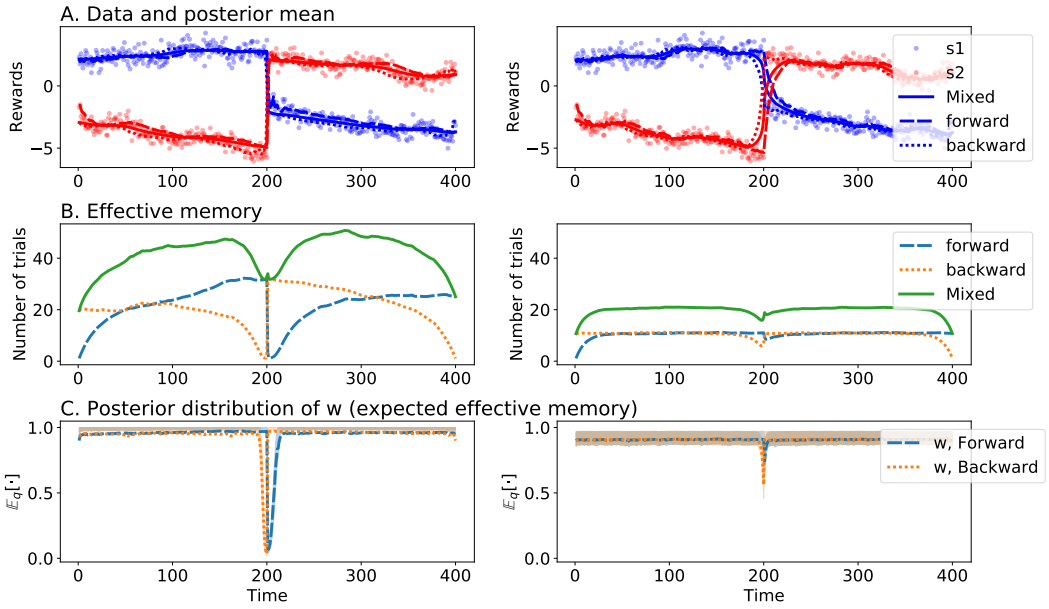


FIGURE 3.7: Experiment 1. Left column: weak prior over w. Right column: strong prior over w. Shaded areas represent the 95% posterior confidence interval. See text for more comments.

$\phi_t^\beta = 1$) or a low confidence ($\phi_t^\alpha = 0.9$, $\phi_t^\beta = 0.1$) about the average value of w. Note that both of these priors had the same expected value. To avoid overfitting of early trials (which may happen using weak priors) while keeping the distribution flexible, we used a flat prior over b: $\boldsymbol{\beta}_0 = 1$. The top level forgetting was ignored ($\gamma = 1$). Results are shown in Figure 3.7.

As the first setting had a weak prior over w, it had more freedom to adapt the posterior distribution to the current data. The effective memory trace (measured with the parameter κ_t) was greater when the environment was stable, and changed faster after the contingency change than when the prior was more confident, where the adaptation was slow and the effective memory did not increase much above the prior-defined threshold 10 (or 20 for the FB algorithm).

The behaviour of both models after the contingency change is informative about the effect that the prior had on the inference process: the weak-prior forgetting factor dropped immediately after an unexpected observation was made, which can be advantageous when sudden changes are expected, but maladaptive in the presence of outliers. The strong-prior model behaved in the opposite way, and handled the change more slowly than its weak-prior counterpart.

It is interesting to note that the posterior probability distribution of b (not shown in the figure) was also more flexible in the first model fit than in the second, because the

observations in the level below were also more variable, due to less confident prior over w: this had the effect of increasing the gain in precision over w, which increased the strength of the posterior over b (through u_2 and u_3 in Equation (3.10)).

3.4.2 Autoregressive model

We fitted the HAFVF to a simulated a non-stationary sinusoidal signal of 400 datapoints issued from two separate systems with a low and high frequency. These signals were randomly generated as the sum of five sinusoidal waves, with the scope of observing whether the algorithm was able to adapt to the abrupt contingency change.

Because we also aimed at a more informative view on the performance of the algorithm in the presence of artifacts, we altered this signal by adding two impulses of 2 a.u. at $t = 100$ and $t = 300$.

We studied a single implementation of the model, with a relatively strong prior over w ($\boldsymbol{\phi} = \{4.5, 0.5\}$) and a flat prior over b, ($\boldsymbol{\beta} = \{1.0, 1.0\}$). The Forward-backward version of the algorithm was applied. We arbitrarily chose a forward and a backward order of 10 samples. Figure 3.8 shows the results of this experiment.

3.4.3 Stochastic Gradient Descent

SGD is a popular technique to find the minimum of (often computationally expensive) loss-functions over large datasets [540] or involving intractable integrals [88] that can be sampled from. However, SGD can be unstable, especially with recurrent neural networks [477] where an isolated, highly noisy sample in the sequence can lead to a degenerate sample of the gradient over the whole sequence. This effect is further magnified when the sample size is low.

We implemented a slightly modified version of the HAFVF in a SGD framework, intended to be similar to the Adam optimizer[3] [493]. In short, we used two specific decays w_1 and w_2 for the posterior of the means and variances of the gradients, respectively, while ensuring that $w_1 < w_2$. We modelled these posteriors as a set of Normal-Inverse-Gamma distributions. Each set of weights and biases of the multilayer perceptron was provided with its own hierarchical decay, to take advantage of the fact that some groups of partial derivatives might be more noisy than others. We used this algorithm with a strong prior over w_1 and w_2 ($\boldsymbol{\phi}_1 = \{9, 1\}$ and $\boldsymbol{\phi}_2 = \{9.5, 0.5\}$), to limit the effect of degenerated gradients on the approximate posteriors.

[3] More details can be found in the supplementary materials

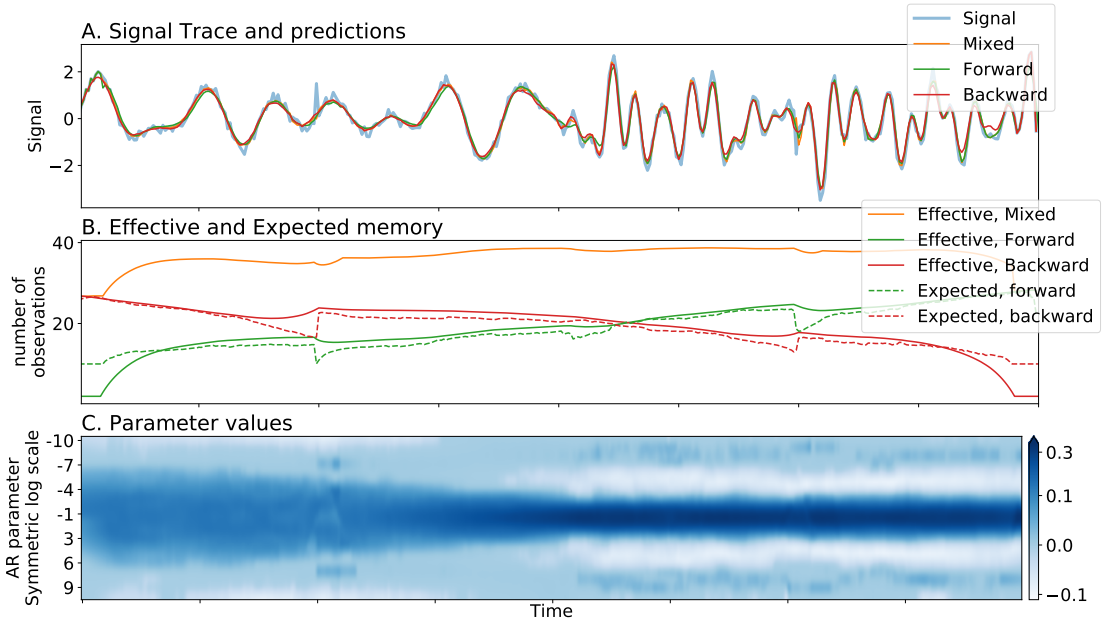


FIGURE 3.8: Experiment 2: Autoregressive model with a weak prior over w. **A.** Observations and simulated response of the models. The zoomed windows show the effect of the artifacts on the estimated mean value. **B.** Effective memory (the "effective number of observations" parameter of the posterior $\boldsymbol{\theta}_t$: namely κ_t) of the three parts of the algorithm (plain lines), and corresponding expected effective memory: $1/(1 + \mathbb{E}_{q(w)}[w]$ (dashed lines). Outliers had a limited impact on learning in both cases. **C.** Value of the AR mean weights through time. The model dealt adequately with the outliers (as the value of the parameters did not change substantially) and with the contingency change (as the values were adapted to the two different signals).

This algorithm was tested with a variational recurrent auto-encoding regression model inspired by [541], where the output probability density was a first passage density of a Wiener process [1]. The simulated dataset was composed of 64 subjects performing a 500 trials long two alternative forced choice task [542], where choices and reaction times were the model was aiming to predict. At each step of the SGD process, 5 subjects were sampled, for a total of 2500 trials.

Figure 3.9 compares the results of the AdaFVF SGD optimizer with the Adam optimizer, executed with the default parameters ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$)[4]. The AdaFVF showed to be less affected by degenerate samples than Adam, as can be seen from the ELBO trace and from the heat plots of the expected memories, for an estimated average negative ELBO of 1.08 for Adam and 0.85 for AdaFVF at the iteration 10000.

[4]The AdaFVF we propose adapts β_1 and β_2 (or at least their posterior distribution) to the data at hand. Optimization of α is proposed by [543] for instance. Future work might elucidate how this kind of online hyperparameter tuning could be framed in the current context.

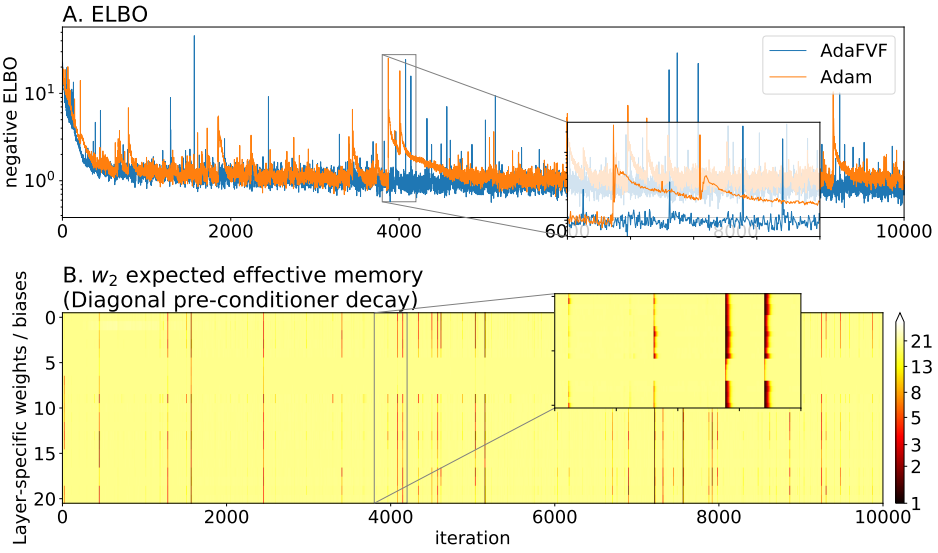


FIGURE 3.9: Experiment 3: SGD with the HAFVF. **A.** ELBO for the Adam optimizer and for the AdaFVF SGD. After an outlier was sampled, the AdaFVF forgot the gradient history, and reset its belief to the naive prior, thereby decreasing the relative contribution of this sample. **B.** Expected memory of the variance posteriors. The impact of outliers is highlighted by the zoomed windows.

3.5 Limitations, perspective and conclusion

Our algorithm has the following limitations: first, the exponential form we have given to the mixture distributions restricts the mixture prior to have a single mode. A linear form, similar to [533] could however also be implemented, at specific levels of the hierarchy of the whole model. It may also be difficult to choose adequate level-specific priors. A generic feature in adaptive forgetting is that the naive prior of the lower level $p_0(\mathbf{z})$ is usually crucial but hard to specify. For the two top levels, we propose as a rule of thumb to use a weak prior in situations where abrupt contingency changes are expected. Such configurations will usually provide a higher memory to the model but are, however, more affected by outliers than stronger priors. Performing updates using mini-batches of datapoints should mitigate this effect. If the environment is meta-stable (i.e. change points occur at a given frequency), derivation of an online type II maximum likelihood optimization algorithm of the prior parameters could improve the algorithm efficiency.

The HAFVF and variants could lead to many promising developments in RL related fields, where they might help to prevent unnecessary forgetting of past events during exploration, in signal processing and more distant fields such as deep learning, where they could be used to prevent the occurrence of catastrophic forgetting.

In conclusion, we present a new generic model aimed at coping with abrupt or slow signal changes and presence of artifacts. This model flexibly adapts its memory to the volatility of the environment, and reduces the risk of abruptly forgetting its learned belief when isolated, unexpected events occur. The HAFVF constitutes a promising tool for decay adaptation in RL, system identification and SGD.

3.6 Appendix

3.6.1 Proof of Proposition 1

Proof. To apply the NCVMP algorithm [70], we first need to compute the inverse posterior covariance of the sufficient statistics of the beta distribution:

$$C(\mathbf{T}(w) \mid \boldsymbol{\phi}) = \begin{bmatrix} \psi_1(\phi^\alpha) - \psi_1(\phi^\alpha + \phi^\beta) & -\psi_1(\phi^\alpha + \phi^\beta) \\ -\psi_1(\phi^\alpha + \phi^\beta) & \psi_1(\phi^\beta) - \psi_1(\phi^\alpha + \phi^\beta) \end{bmatrix}$$

Next, we need to take the derivative of the expected log-joint probability wrt. $\boldsymbol{\phi}$. Noting that $\mathbb{E}_{q(w)}[w] = \frac{\phi^\alpha}{\phi^\alpha + \phi^\beta}$ and that $\mathbb{E}_{q(w)}[w]$ intervenes twice in the expression of the ELBO (in $\mathbb{E}_q\left[\log p(\mathbf{z} \mid \widehat{\boldsymbol{\vartheta}})\right]$) and in $C(\mathbf{T}(\mathbf{z}) \mid \widehat{\boldsymbol{\vartheta}})$), we can use the chain rule and write:

$$C(\mathbf{T}(w) \mid \boldsymbol{\phi})^{-1} \nabla_{\boldsymbol{\phi}} \left\{ \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{z} \mid \widehat{\boldsymbol{\vartheta}})\right] + \mathbb{E}_{q(w)}\left[\log p(\boldsymbol{\phi} \mid \widehat{\boldsymbol{\varphi}})\right] - \frac{1}{2}\mathbb{V}ar_{q(w)}[w]\, C(\mathbf{T}(\mathbf{z}) \mid \widehat{\boldsymbol{\vartheta}}) \right\} =$$

$$C(\mathbf{T}(w) \mid \boldsymbol{\phi})^{-1} \Big(\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q(w)}[w]\, \delta_{\mathrm{L}} + \begin{bmatrix} \varphi^\alpha \left(\psi_1(\phi^\alpha) - \psi_1(\phi^\alpha + \phi^\beta)\right) - \varphi^\beta \psi_1(\phi^\alpha + \phi^\beta) \\ \varphi^\beta \left(\psi_1(\phi^\beta) - \psi_1(\phi^\alpha + \phi^\beta)\right) - \varphi^\alpha \psi_1(\phi^\alpha + \phi^\beta) \end{bmatrix} +$$

$$\frac{1}{2} \left(\nabla_{\boldsymbol{\phi}} \left\{ \mathbb{V}ar_{q(w)}[w] \right\} C(\mathbf{T}(\mathbf{z}) \mid \widehat{\boldsymbol{\vartheta}}) + \mathbb{V}ar_{q(w)}[w]\, \nabla_{\boldsymbol{\phi}_t} \left\{ \mathbb{E}_{q_t(w)}[w] \right\} \frac{d}{d\widehat{w}} C(\mathbf{T}(\mathbf{z}) \mid \widehat{\boldsymbol{\vartheta}}) \right) \Big)$$

Expanding this final expression gives back the expression in proposition 1. □

Corollary 3.4. *For a given value of δ_{L} and δ_{C}, $\phi_t^\alpha > \phi_t^\beta$ implies that ϕ_t^α is more affected (positively or negatively) than ψ_t^β by δ_{L} and δ_{C}, and conversely.*

Proof. This directly follows from the fact that

$$\phi_t^\alpha > \phi_t^\beta \Leftrightarrow C\phi_t^\alpha + B\phi_t^\beta > C\phi_t^\beta + A\phi_t^\alpha$$

. □

Note also that, for a given value of ϕ_1^α, $K(\phi^\beta, \phi^\alpha) \to 0$ if $\phi_t^\beta \to 0$, and conversely for $K(\phi^\alpha, \phi^\beta)$.

3.6.2 Proof of Lemma 1

Proof. Let $g_{\widehat{\boldsymbol{\vartheta}}} \triangleq \nabla_{\widehat{\boldsymbol{\vartheta}}} p(\mathbf{z} \mid \widehat{\boldsymbol{\vartheta}})$ be the score function of the prior distribution and $\widehat{w} \triangleq \mathbb{E}_{q_t(w)}[w]$. We want to solve the following equality wrt δ_{L}:

$$\nabla_{\boldsymbol{\phi}_t} \left\{ \mathbb{E}_{q_t(\mathbf{z})} \left[g_{\widehat{\boldsymbol{\vartheta}}} \right] \right\} \approx \nabla_{\boldsymbol{\phi}_t} \widehat{w}\, \delta_{\mathrm{L}}$$

Which has the following solution:

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}_t} \widehat{w}\, \delta_{\mathrm{L}} &= \nabla_{\boldsymbol{\phi}_t} \{\widehat{w}\} \frac{d}{d\widehat{w}} \mathbb{E}_{q_t(\mathbf{z})} \left[g_{\widehat{\boldsymbol{\vartheta}}} \right] \\
&= \nabla_{\boldsymbol{\phi}_t} \{\widehat{w}\} \times \\
&\mathbb{E}_{q_t(\mathbf{z})} \left[\mathbf{z}^T (\boldsymbol{\theta}^{\xi}_{t-1} - \boldsymbol{\theta}^{\xi}_0) - A(\mathbf{z}) \left(\boldsymbol{\theta}^{\eta}_{t-1} - \boldsymbol{\theta}^{\eta}_0 \right) - \frac{d\widehat{\boldsymbol{\vartheta}}}{d\widehat{w}} \nabla_{\widehat{\boldsymbol{\vartheta}}} B\left(\widehat{\boldsymbol{\vartheta}} \right) \right] \\
&= \nabla_{\boldsymbol{\phi}_t} \{\widehat{w}\} \times \\
&\Big(\mathbb{E}_{q_t(\mathbf{z})} [\mathbf{z}]^T \left(\boldsymbol{\theta}^{\xi}_{t-1} - \boldsymbol{\theta}^{\xi}_0 \right) - \mathbb{E}_{q_t(\mathbf{z})} [A(\mathbf{z})] \left(\boldsymbol{\theta}^{\eta}_{t-1} - \boldsymbol{\theta}^{\eta}_0 \right) + \\
&- \frac{d\widehat{\boldsymbol{\vartheta}}}{d\widehat{w}} \nabla_{\widehat{\boldsymbol{\vartheta}}} B\left(\widehat{\boldsymbol{\vartheta}} \right) \Big)
\end{aligned} \tag{3.16}$$

As $p(\mathbf{z})$ is assumed to be from the exponential family, the derivative of the log-partition function $B(\cdot)$ wrt the natural parameter $\boldsymbol{\theta}(w)$ is equal to the expected value of the sufficient statistics:

$$\frac{d\widehat{\boldsymbol{\vartheta}}}{d\widehat{w}} \nabla_{\widehat{\boldsymbol{\vartheta}}} B\left(\widehat{\boldsymbol{\vartheta}} \right) = \begin{bmatrix} \boldsymbol{\theta}^{\xi}_{t-1} - \boldsymbol{\theta}^{\xi}_0 \\ \boldsymbol{\theta}^{\eta}_{t-1} - \boldsymbol{\theta}^{\eta}_0 \end{bmatrix}^T \begin{bmatrix} \mathbb{E}_{p(\mathbf{z})} [\mathbf{z}] \\ \mathbb{E}_{p(\mathbf{z})} [-A(\mathbf{z})] \end{bmatrix}$$

which can be plugged into Equation (3.16) to retrieve the expression of Lemma 1. □

We see that the value of δ_{L} depends on two elements: the first being whether the sign of $\mathbb{E}_{q(\mathbf{z})}[\mathbf{z}] - \mathbb{E}_{p(\mathbf{z})}[\mathbf{z}]$ matches the sign of $\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0$, and if the difference of the expected log-partitions under the posterior q and prior p is negative (because $\boldsymbol{\theta}^{\eta}_{t-1}$ is always greater than $\boldsymbol{\theta}^{\eta}_0$). The first summand can therefore be understood as a measure of how much the new observations $T(x_t)$, which conditions q_t matches the observed difference between the previous posterior q_{t-1} and the initial prior p_0: if it does (i.e. both differences are negative / positive) then there is evidence that w should increase (because δ_{L} will grow, see below). The second summand somehow measures whether the new posterior q_t will on average decrease the entropy of the model $p(x \mid \mathbf{z})$, which is linearly determined by $A(\mathbf{z})$.

3.6.3 AdaFVF implementation

Following [493], we propose a scheme similar to the Adam optimizer where the mean gradient and preconditioner are optimized according to the HAFVF. We will consider the problem of inferring the vectors m and v^2, i.e. the vectors of first and second moments of the gradient. We will use a slightly different notation than [493]: the decays β_1 and β_2 will be replaced by w_1 and w_2 respectively, for the sake of coherence.

Let us consider that the partial derivative at the iteration t follows a normal distribution with mean and covariance m_t, s_t. The conjugate prior of this distribution is a Normal Inverse Gamma distribution with parameters $\boldsymbol{\theta} \triangleq \{\mu_0, \kappa_0, \alpha_0, \beta_0\}$. One could already apply the HAFVF to this model, with the restriction that $w_1 = w_2$. To keep the constraint $w_1 < w_2$, we assume a fully factorized posterior, and factorize the joint probability defined in the paper (equation 3) to:

$$
\begin{aligned}
&p(d_t, m_t, s_t^2, w_1, w_2, b|\mathbf{x}_{1:t-1}) \approx p(d_t \mid m_t, s_t^2)\times \\
&\frac{\left(q_{t-1}(m_t \mid m_{t-1}, s^2/\kappa_{t-1})^{w_1} p(s_t^2 \mid \alpha_{t-1}, \beta_{t-1})\right)^{w_2}}{Z(w, \boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_0)}\times \\
&\frac{\left(p(m_t \mid m_0, s^2/\kappa_0)^{\frac{1-w_1 w_2}{1-w_2}} p(s_t^2 \mid \alpha_0, \beta_0)\right)^{1-w_2}}{Z(w, \boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_0)}\times \\
&\frac{q(w_1 \mid \boldsymbol{\phi}_{1t-1})^b p(w_1 \mid \boldsymbol{\phi}_{10})^{b-1}}{Z(b, \boldsymbol{\phi}_{1t-1}, \boldsymbol{\phi}_{10})} \frac{q(w_2 \mid \boldsymbol{\phi}_{2t-1})^b p(w_2 \mid \boldsymbol{\phi}_{20})^{b-1}}{Z(b, \boldsymbol{\phi}_{2t-1}, \boldsymbol{\phi}_{20})}\times \\
&p(b \mid \boldsymbol{\beta}_{t-1})
\end{aligned}
\tag{3.17}
$$

where $\boldsymbol{\theta}$ is defined as the prior or approximate posterior parameters at the trial 0 or $t-1$, respectively. This new formulation ensured that the decay of m was equal to $w_1 * w_2$.

For a set of N partial derivatives, the natural implementation of the joint probability presented here before for multiple, univariate gaussians would be

$$
p(\mathbf{d}, \mathbf{m}, \mathbf{s}, w, b) = \prod_{i=1}^{N} p(d^i \mid m^i, s^i) p(m^i, s^i \mid \boldsymbol{\vartheta}^i) p(w_1, w_2 \mid \boldsymbol{\varphi}) p(b \mid \boldsymbol{\beta})
$$

but this model is hard to fit in practice, because the posterior over w is highly sensitive to the dimensionality of the data at the level below (see Proposition 1). In order to deal with this, we modified the above equation by taking the N^{th} radical of the joint probability $p(d^i, m^i, s^i)$. The normalized log-joint probability then reads:

$$
\log \widetilde{p}(\mathbf{d}, \mathbf{m}, \mathbf{s}, w, b) = \frac{\sum_{i=1}^{N} \log p(d^i \mid m^i, s^i) + \log p(m^i, s^i \mid \boldsymbol{\vartheta}^i)}{N} + \log p(w_1, w_2 \mid \boldsymbol{\varphi}) + \log p(b \mid \boldsymbol{\beta}).
$$

Algorithm 9: AdaFVF algorithm. $\mathcal{L}^j(q(m, s^2, w_t, b_t))$ stands for the ELBO value, $\widehat{\mathbf{d}}$ is the vector of expected first moment of the partial derivatives, and $\widehat{\mathbf{d}^2}$ the vector of expected second moments. η is the step size.

Input: noisy function $f(\mathbf{z})$ with parameters $\mathbf{z} = \{z^j \in \mathbb{R}^{D^j}\}$ for $j = 1 : J$, hyperparameters $\{\boldsymbol{\theta}_0, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0\}$, learning rate η;
Return: Optimum $\mathbf{z}^*$;
Initialize randomly $\mathbf{z}_1$;
for $t = 1$ ***to*** T **do**
  get $\mathbf{d}_t = \nabla_{\mathbf{z}_t} f(\mathbf{z}_t)$;
  for $j = 1$ ***to*** J **do**
    set $i \leftarrow 0$;
    set $\mathcal{L}^j(q(m_t, s_t, w_t, b_t)) \leftarrow -\infty$;
    while $i < 100$ *and* $\delta\mathcal{L}^j(q(m_t, s_t, w_t, b_t)) > 0.001$ **do**
      i += 1 update:
      $\boldsymbol{\theta}_t^j$ s.t. $q(m_t, s_t^2 | \boldsymbol{\theta}_t^j) = \frac{\exp \mathbb{E}_{q(w_{1t}, w_{2t}, b_t))}\left[\log \widetilde{p}(d, m, s^2, w_1, w_2, b\right]}{Z}$; `// see CVMP`
      $\{\boldsymbol{\phi}_t^j, \boldsymbol{\beta}_t^j\} = \arg\max_{\boldsymbol{\phi}_t^j, \boldsymbol{\beta}_t^j} \mathcal{L}^j(q(m, s^2, w_t, b_t))$; `// see NCVMP`
      compute $\mathcal{L}^j(q(m, s^2, w_t, b_t))$;
    end
    update ${z^j}_{t+1} = {z^j}_t - \eta * \widehat{\mathbf{d}}/\widehat{\mathbf{d}^2}$;
  end
end

An important consideration to make is that we fitted the HAFVF to each of the sets of weights and biases independently: this ensured that the decays were adapted to the scale of each of these gradients independently. For instance, the extreme layers of a neural network usually have a wider distribution of partial derivatives than the intermediate layers: the HAFVF can account for this and adapt accordingly. Also, unlike other approaches, our method does not deal with degenerate samples by ignoring the step, but by decreasing their relative importance: in this way, the algorithm can discriminate which layer should be ignored (or better, reset) and which should not.

The full algorithm is given in Line 9.

Considering the fact that the function f is supposedly computationally expensive, the computational cost of this approach comparatively not much higher than the one of other SGD algorithms: update of the variational posterior over m, s is virtually identical to the update achieved by Adam, and most of the cost comes from the (possibly grouped) computation of the expected log-joint probability and its gradient. Expensive function in the ELBO include mostly the logarithm of the rate parameter of the prior $\log \beta(w)$ and approximate posterior $\log \beta$ which appear in the Gamma log-likelihood and in the

expectation of the log-variance, respectively[5]. This evaluation is the step with the highest computational burden.

A final point to emphasize it that AdaFVF requires a careful choice of prior distribution over m, s^2, w_1, w_2 and b. As stated in the main text, we used high and confident prior over w_1 and w_2 to ensure that these parameters did not increase too much (so that they kept following the current distribution of gradients) and also to ensure that, after an degenerate gradient was observed, these parameters quickly came back to a value close the prior initially chosen. For m, s^2 the value of $\boldsymbol{\theta}$ was set to $\boldsymbol{\theta}_0 = \{\mu_0 = 0, \kappa_0 = 0, \alpha_0 = 2, \beta_0 = 10^{-5}\}$. The gradient over b was set to a highly informative value ($\boldsymbol{\beta} = 20$): this had the effect of allowing the posterior over w to change slowly, and helped stabilizing the algorithm. This configuration worked well across the few models we tested, regardless of the sample size or the complexity of the problem. Future theoretical and practical research should, however, be dedicated to explore in which way these choices impact the performance of the algorithm.

Although the proof of convergence of [88] does not hold anymore with our formulation, we observed that this algorithm was less sensitive to noise in the gradient than Adam, and performed at least equally well for problems ranging from simple auto-encoding variational inference to complex neural network training.

[5] the full expression of the latter is $\mathbb{E}_q\left[\log s^2\right] = \log\beta - \psi(\alpha)$, from which only the log function need to be computed for every sample, as the effective number of observations α is supposedly identical for all the elements of z^j

Chapter 4

The AC-HAFVF

This work is currently under review for a publication in Plos Computational Biology, with co-authorship with Alexandre Zénon.

Agents living in volatile environments must be able to detect changes in contingencies while refraining to adapt to unexpected events that are caused by noise. In Reinforcement Learning (RL) frameworks, this requires learning rates that adapt to past reliability of the model. The observation that behavioural flexibility in animals tends to decrease following prolonged training in stable environment provides experimental evidence for such adaptive learning rates. However, in classical RL models, learning rate is either fixed or scheduled and can thus not adapt dynamically to environmental changes. Here, we propose a new Bayesian learning model, using variational inference, that achieves adaptive change detection by the use of Stabilized Forgetting, updating its current belief based on a mixture of fixed, initial priors and previous posterior beliefs. The weight given to these two sources is optimized alongside the other parameters, allowing the model to adapt dynamically to changes in environmental volatility and to unexpected observations. This approach is used to implement the "critic" of an actor-critic RL model, while the actor samples the resulting value distributions to choose which action to undertake. We show that our model can emulate different adaptation strategies to contingency changes, depending on its prior assumptions of environmental stability, and that model parameters can be fit to real data with high accuracy. The model also exhibits trade-offs between flexibility and computational costs that mirror those observed in real data. Overall, the proposed method provides a general framework to study learning flexibility and decision making in RL contexts.

4.1 Introduction

Learning agents must be able to deal efficiently with surprising events when trying to represent the current state of the environment. Ideally, agents response to such events should depend on their belief about how likely the environment is to change. When expecting a steady environment, a surprising event should be considered as an accident and should not lead to updating previous beliefs. Conversely, if the agent assumes the environment is volatile, then a single unexpected event should trigger forgetting of past beliefs and relearning of the (presumably) new contingency. Importantly, assumptions about environmental volatility can also be learned from experience.

Here, we propose a general model that implements this adaptive behaviour using Bayesian inference. This model is divided in two parts: the critic which learns the environment and the actor that makes decision on the basis of the learned model of the environment.

The critic side of the model (called Hierarchical Adaptive Forgetting Variational Filter, HAFVF) discriminates contingency changes from accidents on the basis of past environmental volatility, and adapts its learning accordingly. This learner is a special case of Stabilized Forgetting (SF) [528]: practically, learning is modulated by a forgetting factor that controls the relative influence of past data with respect to a fixed prior distribution reflecting the naive knowledge of the agent. At each time step, the goal of the learner is to infer whether the environment has changed or not. In the former case, she erases her memory of past events and resets her prior belief to her initial prior knowledge. In the latter, she can learn a new posterior belief of the environment structure based on her previous belief. The value of the forgetting factor encodes these two opposite behaviours: small values tend to bring parameters back to their original prior, whereas large values tend to keep previous posteriors in memory. The first novel contribution of our work lies in the fact that the posterior distribution of the forgetting factor depends on the estimated stability of past observations. The second and most crucial contribution lies in the hierarchical structure of this forgetting scheme: indeed, the posterior distribution of the forgetting factor is itself subject to a certain forgetting, learned in a similar manner. This leads to a 3-level hierarchical organization in which the bottom level learns to predict the environment, the intermediate level represents its volatility and the top level learns how likely the environment is to change its volatility. We show that this model implements a generalization of classical Q-learning algorithms.

The actor side of the model is framed as a full Drift-Diffusion Model of decision making [364] (Normal-Inverse-Gamma Diffusion Process; NIGDM) that samples from the value distributions inferred from the critic in order to select actions in proportion to their probability of being the most valued. We show that this approach predicts plausible

results in terms of exploration-exploitation policy balance, reward rate, reaction times (RT) and cognitive cost of decision. Using simulated data, we also show that the model can uncover specific features of human behaviour in single and multi-stage environments. The whole model is outlined in Line 8: the agent first selects an action given an (approximate) Q-sampling policy, which is temporally implemented as a Full DDM [364] with variable drift rate and accumulation noise, then learns based on the return of the action executed (reward $r(s_j, a_j)$ and transition $s' = T(s_j, a_j)$). Then, the critic updates its approximate posterior belief about the state of the environment $q_j \approx p$.

Algorithm 10: AC-HAFVF sketch. a represents actions, r stands for rewards, s stands for state and $\mathbf{x}$ stands for observations. μ^r is the expected value of the reward. q represents the approximate posterior of the latent variable $\mathbf{z}$ and $\boldsymbol{\theta}_0$ stands for the prior parameter values of the distribution of $\mathbf{z}$.

1 **for** $j = 1$ ***to*** J **do**
2 | **Actor**: $NIGDM$ Section 4.2.5;
3 | select $a_j \sim p\left(a_j = \arg\max_{\mathbf{a}} \mu^r(s_j, \mathbf{a}) \mid \mathbf{x}_{<j}\right)$;
4 | Observe $\mathbf{x}_j = \{r(s_j, a_j), s_{j+1}\}$;
5 | ---
6 | **Critic**: $HAFVF$ Section 4.2.3.1;
7 | update $q_j(\mathbf{z}(s_j, a_j)) \approx p\left(\mathbf{z}(s_j, a_j) \mid \mathbf{x}_j; \mathbf{x}_{<j}, \boldsymbol{\theta}_0\right)$;
8 **end**

We apply the proposed approach to Model-Free RL contexts (i.e. to agents limiting their knowledge of the environment to a set of reward functions) in an extensive manner. We explore in detail the application of our algorithm to Temporal Discounting RL, in which we study the possibility of learning the discounting factor as a latent variable of the model. We also highlight various methods for accounting for unobserved events in a changing environment. Finally, we show that the way our algorithm optimizes the exploration-exploitation balance is similar to Q-Value Sampling when using large DDM thresholds.

Importantly, the proposed approach is very general, and even though we apply it here only to Model-Free Reinforcement Learning, it could be also extended to Model-Based RL [256], where the agent models a state-action to state transition table together with the reward functions. Additionally, other machine-learning algorithms could also benefit from this approach (Chapter 3).

The paper is structured as follows: first (Section 4.1.1) we review briefly the state of the art and place our work within the context of current literature. In Section 4.2, we present the mathematical details of the model. We derive the analytical expressions of the learning rule, and frame them in a biological context. We then show how this learning scheme directly translates into a decision rule that constitutes a special case

of the Sequential Sampling family of algorithms. In the result section Section 4.3, we show various predictions of our model in terms of learning and decision making. More importantly, we show that despite its complexity, the model can be fitted to behavioural data. We conclude by reviewing the contributions of the present work, highlighting its limitations and putting it in a broader perspective.

4.1.1 Related work

The adaptation of learning to contingency changes and noise has numerous connections to various scientific fields from cognitive psychology to machine learning. A classical finding in behavioural neuroscience is that instrumental behaviours tend to be less and less flexible as subjects repeatedly receive positive reinforcement after selecting a certain action in a certain context, both in animals [225, 544, 234] and humans [545, 324, 261, 546]. This suggests that biological agents indeed adapt their learning rate to inferred environmental stability: when the environment appears stable (e.g. after prolonged experience of a rewarded stimulus-response association), they show increased tendency to maintain their model of the environment unchanged despite reception of unexpected data.

Most studies on such automatization of behaviour have focused on action selection. However, weighting new evidence against previous belief is also a fundamental problem for perception and cognition [252, 547, 548]. Predictive coding [404, 549, 330, 550, 551, 552] provides a rich, global, framework that has the potential to tackle this problem, but an explicit formulation of cognitive flexibility is still lacking. For example, whereas [552] provides an elegant Kalman-like Bayesian filter that learns the current state of the environment based on its past observations and predicts the effect of its actions, it assumes a stable environment and cannot, therefore, adapt dynamically to contingency changes. The Hierarchical Gaussian Filter (HGF) proposed by Mathys and colleagues [537, 538] provides a mathematical framework that implements learning of a sensory input in a hierarchical manner, and that can account for the emergence of inflexibility in various situations. This model deals with the problem of flexibility (framed as expected "volatility") by building a hierarchy of random variables: each of these variables is distributed with a Gaussian distribution with a mean equal to this variable at the trial before and the variance equal to a non-linear transform of the variable at superior level. Each level encodes the distribution of the volatility of the level below. Although it has shown its efficiency in numerous applications [553, 554, 555, 556, 48, 557], a major limitation of this model, within the context of our present concern, is that it makes the assumption that the volatility of the environment depends deterministically on the variance of the

observations. Crucially, this model cannot forget past experience after changes of contingency, but can only adapt its learning to the current contingency. Consequently, the HGF model will not be able to account for environments whose volatility change over time, i.e. where changes in contingencies occur sometimes abruptly, sometimes progressively. Moreover, HGF will have difficulties accounting for changes in the variance of the observations. These caveats limit the applicability of the HGF to a narrow range of datasets.

As will be shown in detail below, in the model proposed in the present paper, volatility is not only a function of the variance of the observations: if a new observation falls close enough to previous estimates then the agent will refine its posterior estimate of the variance and will decrease its forgetting factor (i.e. will move its prior away from the fixed initial prior and closer to the learned posterior from the previous trial), but if the new observation is not likely given this posterior estimate, the forgetting factor will increase (i.e. will move closer to the fixed initial prior) and the model will tend to update to a novel state (because of the low precision of the initial prior). In Section 4.3.1, we show that our model outperforms the HGF in such situations. Another, more technical, limitation of the HGF comes from the non-conjugate form of the generative model, which makes the use of mean-field VB challenging, and requires reliance on numerous approximations. For instance, the quadratic approximations on which the HGF relies are only valid for large n [74], but the HGF is practically applicable only to small datasets because the parameters sub-space where updates are possible varies widely with the data, such that optimization fails in many instances and especially for large datasets. The model we provide, on the contrary, has no such constraint, and can be generalized to many combinations of exponential family models (e.g. Bernoulli or multivariate Gaussian distribution at the first level) without tedious derivations.

In Machine Learning and in Statistics, too, the question of whether new unexpected data should be classified as outlier or environmental change is important [25]. This problem of "denoising" or "filtering" the data is ubiquitous in science, and usually relies on arbitrary assumptions about environmental stability. In signal processing and system identification, adaptive forgetting is a broad field where optimality is highly context (and prior)-dependant [528]. Bayesian Filtering (BF) [530], and in particular the Kalman Filter [522] often lack the necessary flexibility to model real-life signals that are, by nature, changing. One can discriminate two approaches to deal with this problem: whereas Particle Filtering (PF) [531, 521, 532] is computationally expensive, the SF family of algorithms [528, 533], from which our model is a special case, usually has greater accuracy for a given amount of resources [531] (for more information, we refer to [522] where SF is reviewed). Most previous approaches in SF have used a truncated exponential prior [536, 523] or a fixed, linear mixture prior to account for the stability

of the process [521]. Our approach is innovative in this field in two ways: first, we use a Beta prior on the mixing coefficient (unusual but not unique [535]), and we adapt the posterior of this forgetting factor on the basis of past observations, the prior of this parameter and its own adaptive forgetting factor. Second, we introduce a hierarchy of forgetting that stabilizes the learning when the training length is long.

We therefore intend to focus our present research on the very question of flexibility. We will show how flexibility can be implemented in a Bayesian framework using an adaptive forgetting factor, and what prediction this framework makes when applied to learning and decision making in Model-Free paradigms.

4.2 Methods

4.2.1 Bayesian Q-Learning and the problem of flexibility

Classical RL [212], or Bayesian RL [162, 208] cannot discriminate learners that are more prone to believe in a contingency change from those who tend to disregard unexpected events and consider them as noise. To show this, we take the following example: let $p(\rho|r_{\leq j}) = \text{Beta}(\alpha, \beta)$ be the posterior probability at trial j of a binary reward $r_j \sim \text{Bern}(\rho)$ with prior probability $\rho \sim \text{Beta}(\alpha_0, \beta_0)$. It can be shown that, at the trial $j = v_j + u_j$, where v_j is the number of successes and u_j the number of failures, the posterior probability parameters read $\{\alpha_j = \alpha_0 + v_j, \beta_j = \beta_0 + u_j\}$. This can be easily mapped to a classical RL algorithm if one considers that, at each update of v and u, the posterior expectation of ρ is updated by

$$\begin{aligned}
\mathbb{E}\left[\rho|r_{\leq j}\right] &= \frac{v_{j-1} + r_j}{v_{j-1} + r_j + u_{j-1} + (1 - r_j)} \\
&= \frac{j-1}{j}\mathbb{E}\left[\rho_{j-1}\right] + \frac{r_j}{j} \\
&= \mathbb{E}\left[\rho_{j-1}\right] + \eta(r_j - \mathbb{E}\left[\rho_{j-1}\right]) \\
\text{where } \eta &\triangleq \frac{1}{j}
\end{aligned}$$

which has the form of a classical myopic Q-learning algorithm with a decreasing learning rate.

The drawback of this fixed-schedule learning rate is that, if the number of observed successes outnumbers greatly the number of failures ($v \gg u$) at the time of a contingency change in which failures become suddenly more frequent, the agent will need $v - u + 1$ failures to start considering that $p(r_j = 0|r_{\leq j}) > p(r_j = 1|r_{\leq j})$. This behaviour is obviously sub-optimal in a changing environment, and Dearden [162] suggests adding a

constant forgetting factor to the updates of the posterior, making therefore the agent progressively blind to past outcomes. Consider the case in which

$$\mathbb{E}\left[\rho|r_{\leq j}\right] = \frac{wv_{j-1} + r_j}{wv_{j-1} + r_j + wu_{j-1} + (1 - r_j)}$$

with $w \in (0;1)$ being the forgetting factor. We can easily see that, in the limit case of $\alpha_0 = 0$ and $\beta_0 = 0$, $\alpha_j + \beta_j \to \frac{1}{1-w}$ as $j \to \infty$. We can define $\frac{1}{1-w}$ as the *efficient memory* of the agent, which provides a bound on the *effective memory*, represented by the total amount of trials taken into account so far (e.g. $\alpha_j+\beta_j$ in the previous example). This produces an upper and lower bound to the variance of the posterior estimate of $p(\rho|r_{\leq j})$. This can be seen from the variance of the beta distribution $\mathbb{V}ar\left[\rho \mid r_{\leq j}\right] = \frac{(\alpha_j)(\beta_j)}{(\alpha_j+\beta_j)^2(\alpha_j+\beta_j+1)}$ which is maximized when $\alpha_j = \beta_j$, and minimized when either $\alpha_j = \alpha_0$ or $\beta_j = \beta_0$. In a steady environment, agents with larger memory are advantaged since they can better estimate the variance of the observations. But when the environment changes, large memory becomes disadvantageous because it requires longer time to adapt to the new contingency. Here, we propose a natural solution to this problem by having the agent erase its memory when a new observation (or a series of observations) is unlikely given the past experience.

4.2.2 General framework

Our framework is based on the following assumptions:

Assumption 3. The environment is fully Markovian: the probability of the current observation given all the past history is equal to the probability of this observation given the previous observation.

Assumption 4. At a given time point, all the observations (rewards, state transitions, etc.) are i.i.d. and follow a distribution $p(\mathbf{x} \mid \mathbf{z})$ that is issued from the exponential family and has a conjugate prior that also belongs to the exponential family $p(\mathbf{z} \mid \boldsymbol{\theta}_0)$.

For conciseness, the latent variables $\mathbf{z}$ (i.e. action value, transition probability etc.) and their prior $\boldsymbol{\theta}$ will represent the natural parameters of the corresponding distributions in what follows.

Assumption 5. The agent builds a hierarchical model of the environment, where each of the distributions at the lower level (reward and state transitions) are independent, i.e. the reward distribution for one state-action cannot be predicted from the distribution of the other state-actions.

Assumption 6. The agent can only observe the effects of the action she performs.

4.2.3 Model specifications

We are interested in deriving the posterior probability of some datapoint-specific measure $\mathbf{z}_j$, where j indicates the point in time, given the past and current observations $\mathbf{x}_{\leq j}$ and some prior belief $\boldsymbol{\theta}_0$. Bayes theorem states that this is equal to

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{z} \mid \boldsymbol{\theta}_0) \prod_{j=1}^{J} p(\mathbf{x}_j \mid \mathbf{z})}{\prod_{j=1}^{J} p(\mathbf{x}_j)}. \tag{4.1}$$

We now consider the case of a subject that observes the stream of data and updates her posterior belief on-line as data are gathered. According to Assumption 4, one can express the posterior of $\mathbf{z}$ given the current observation x_j

$$p(\mathbf{z} \mid x_j, \mathbf{x}_{<j}) = \frac{p(x_j \mid \mathbf{z}) p(\mathbf{z} \mid \mathbf{x}_{<j})}{p(x_j \mid \mathbf{x}_{<j})}. \tag{4.2}$$

It appears immediately that the prior $p(\mathbf{z} \mid \mathbf{x}_{<j})$ has the same form as the previous posterior, so that our posterior probability function can be easily estimated recursively using the last posterior estimate as a prior (yesterday's posterior is today's prior) until $p(\mathbf{z} \mid \boldsymbol{\theta}_0)$ is reached. Assumption 4 implies that the posterior $p(\mathbf{z} \mid x_j, \mathbf{x}_{<j})$ will be tractable: since $p(\mathbf{x} \mid \mathbf{z})$ is from the exponential family and has a conjugate prior $p(\mathbf{z} \mid \boldsymbol{\theta}_0)$, the posterior probability has the same form as the prior, and has a convenient expression:

$$\boldsymbol{\theta}_j = \begin{bmatrix} \boldsymbol{\theta}_j^{\xi} \\ \theta_j^{\eta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_0^{\xi} + \sum_{i=1}^{j} \mathbf{T}(\mathbf{x}_i) \\ \theta_0^{\eta} + j \end{bmatrix} \tag{4.3}$$

where $\mathbf{T}(\mathbf{x}_i)$ is the sufficient statistics of the i$^{\text{th}}$ sample, and where we have made explicit the fact that $\boldsymbol{\theta}_0 = \{\boldsymbol{\theta}_0^{\xi}, \theta_0^{\eta}\}$ can be partitioned in two parts, from which θ_0^{η} represents the prior number of observations. Consequently, θ_j^{η} is equivalent to the effective memory introduced above.

This simple form of recursive posterior estimate suffers from the drawbacks we want to avoid, i.e. it does not forget past experience. We therefore introduce a two-component mixture prior where the previous posterior is weighted against the original prior belief:

$$p(\mathbf{z} \mid \mathbf{x}_{<j}; \boldsymbol{\theta}_0) \triangleq \frac{p(\mathbf{z} \mid \mathbf{x}_{<j})^w p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w}}{Z(w, \mathbf{x}_{<j}, \boldsymbol{\theta}_0)} \quad \text{with } w \in [0; 1]. \tag{4.4}$$

The exponential weights on this prior mixture allow us to easily write its logarithmic form, but it still demands that we compute the normalizing constant $Z(w, \mathbf{x}_{<j}, \boldsymbol{\theta}_0) = \int p(\mathbf{z} \mid \mathbf{x}_{<j})^w p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w} d\,\mathbf{z}$. This constant has, however, a closed-form if both the prior

and the previous posterior are from the same distribution, issued from the exponential family.

In Equation (4.4), we assume that the forgetting factor is unknown and that the learner needs to infer it from the data at hand. We therefore put a beta prior on this parameter, and under the assumption that the posterior probability factorizes (Mean-Field assumption), the joint probability at time j reads:

$$p(x_j, \mathbf{z}, w|\mathbf{x}_{<j}; \boldsymbol{\theta}_0, \boldsymbol{\phi}_0) = p(x_j \mid \mathbf{z}) \frac{p(\mathbf{z} \mid \mathbf{x}_{<j})^w p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w}}{Z(w, \mathbf{x}_{<j}, \boldsymbol{\theta}_0)} p(w \mid \mathbf{x}_{<j}, \boldsymbol{\phi}_0) \tag{4.5}$$

where $\boldsymbol{\phi}_0$ is the vector of the parameters of the beta prior of w. The model in Equation (4.5) is not conjugate, and the posterior is therefore not guaranteed to be tractable anymore.

4.2.3.1 Hierarchical Filter

Let us now analyze the expected behaviour of an agent using a model similar to the one just described, in a steady environment: if all $\mathbf{x} \in \mathbf{x}_{\leq j}$ belong to the same, unknown distribution $p(\mathbf{x} \mid \mathbf{z})$, the value of $\mathbb{E}_{q_j(\mathbf{z})}[\mathbf{z}]$ will progressively converge to its true value as the prior (or the previous posterior) over w will eventually put a lot of weight on the past experience (i.e. it favours high values of w), since the distribution from which $\mathbf{x}$ is drawn is stationary. We have shown that such models rapidly tend to an overconfident posterior over w (Chapter 3). In practice, when the previous posterior of w is confident on the value that w should take (i.e. has low variance), it tends to favor updates that reduce variance further, corresponding to values of $p_j(w \mid \mathbf{x}_{\leq t})$ that match $p_{j-1}(w \mid \mathbf{x}_{<t})$, even if this means ignoring an observed mismatch between $p_{j-1}(\mathbf{z} \mid \mathbf{x}_{<t})$ and $p_j(\mathbf{z} \mid \mathbf{x}_{\leq t})$. In order to deal with this issue, we enrich our model by introducing a third level in the hierarchy.

We re-define the prior over w as a two-component mixture of priors:

$$p_j(w \mid \mathbf{x}_{<j}; b, \boldsymbol{\phi}_0) \triangleq \frac{p(w \mid \mathbf{x}_{<j})^b p(w \mid \boldsymbol{\phi}_0)^{1-b}}{Z(b, \mathbf{x}_{<j}, \alpha_0)}$$

and the full joint probability has the form

$$\begin{aligned} p(x_j, \mathbf{z}, w, b|\mathbf{x}_{<j}; \boldsymbol{\theta}_0, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0) = p(x_j \mid \mathbf{z}) &\frac{p(\mathbf{z} \mid \mathbf{x}_{<j})^w p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w}}{Z(w, \mathbf{x}_{<j}, \boldsymbol{\theta}_0)} \\ &\frac{p(w \mid \mathbf{x}_{<j})^b p(w \mid \boldsymbol{\phi}_0)^{1-b}}{Z(b, \mathbf{x}_{<j}, \boldsymbol{\phi}_0)} p(b \mid \mathbf{x}_{<j}, \boldsymbol{\beta}_0). \end{aligned} \tag{4.6}$$

4.2.3.2 Variational Inference

Two classes of methods can be used to estimate the posterior probability at the previous time step needed in Equation (4.6). Simulation-based algorithms [22] such as importance sampling, particle filtering or Markov Chain Monte Carlo, are asymptotically exact but computationally expensive, especially in the present case where the estimate has to be refined at each time step. The other class of methods, approximate inference [488, 525], consists in formulating, for a model with parameters $\mathbf{y}$ and data $\mathbf{x}$, an approximate posterior $q(\mathbf{y})$, that we will use as a proxy to the true posterior $p(\mathbf{y} \mid \mathbf{x})$. Roughly, approximate inference can be partitioned into Expectation Propagation and Variational Bayes (VB) methods. Let us consider in more detail VB, as it is the core of our learning model. In VB, optimizing the approximate posterior amounts to computing a lower-bound to the log model evidence (ELBO) $\mathcal{L}\left(q(\mathbf{y})\right) \leq \log p(\mathbf{x} \mid \mathbf{x}_{<j})$, whose distance from the true log model evidence can be reduced by gradient descent [490]. Hybrid methods, that combine sampling methods with approximate inference, also exist (e.g. Stochastic Gradient Variational Bayes [558] or Markov Chain Variational Inference [505]). With the use of refined approximate posterior distributions [88, 107, 103], they allow for highly accurate estimates of the true posterior with possibly complex, non-conjugate models.

We define a variational distribution over $\mathbf{y}$ with parameters $\boldsymbol{\upsilon}$: $q(\mathbf{y} \mid \boldsymbol{\upsilon})$, which we will use as a proxy to the real, but unknown, posterior distribution $p(\mathbf{y} \mid \mathbf{x})$. The two distribution match exactly when their Kullback-Leibler divergences are equal to zero, i.e.

$$\begin{aligned} & D_{KL}\left[q(\mathbf{y})\middle\|p(\mathbf{y} \mid \mathbf{x})\right] = 0 \\ \Leftrightarrow \quad & D_{KL}\left[p(\mathbf{y} \mid \mathbf{x})\middle\|q(\mathbf{y})\right] = 0 \\ \Leftrightarrow \quad & p(\mathbf{y} \mid \mathbf{x}) = q(\mathbf{y}) \end{aligned}$$

where we have omitted the approximate posterior parameters $\boldsymbol{\upsilon}$ for sparsity of the expressions. Given some arbitrary constraints on $q(\mathbf{y})$, we can choose (for mathematical convenience) to reduce $D_{KL}\left[q(\mathbf{y})\middle\|p(\mathbf{y} \mid \mathbf{x})\right]$ wrt $q(\mathbf{y})$. Formally, this can be written as

$$\begin{aligned} q^*(\mathbf{y}) &= \arg\min_{q(\mathbf{y})} D_{KL}\left[q(\mathbf{y})\middle\|p(\mathbf{y} \mid \mathbf{x})\right] \\ &= \arg\min_{q(\mathbf{y})} \int q(\mathbf{y}) \log \frac{q(\mathbf{y})}{p(\mathbf{y} \mid \mathbf{x})} d\mathbf{y} \\ &= \arg\min_{q(\mathbf{y})} \int q(\mathbf{y}) \log q(\mathbf{y}) d\mathbf{y} - \int q(\mathbf{y}) \log p(\mathbf{y} \mid \mathbf{x}) d\mathbf{y}. \end{aligned}$$

We can now substitute $\log p(\mathbf{y} \mid \mathbf{x})$ by its rhs in the log-Bayes formula

$$\begin{aligned} D_{KL}\left[q(\mathbf{y})\middle\|p(\mathbf{y} \mid x)\right] &= \int q(\mathbf{y}) \log q(\mathbf{y}) d\mathbf{y} - \int q(\mathbf{y}) \Big(\log p(\mathbf{x}, \mathbf{y}) - \log p(\mathbf{x}) \Big) d\mathbf{y} \\ \Leftrightarrow \log p(\mathbf{x}) &= \underbrace{\int q(\mathbf{y}) \log p(\mathbf{x}, \mathbf{y}) d\mathbf{y} - \int q(\mathbf{y}) \log q(\mathbf{y}) d\mathbf{y}}_{\mathcal{L}(q(\mathbf{y}))} + D_{KL}\left[q(\mathbf{y})\middle\|p(\mathbf{y} \mid \mathbf{x})\right]. \end{aligned} \tag{4.7}$$

Because $\log p(\mathbf{x})$ does not depend on the model parameters, it is fixed for a given dataset. Therefore, as we maximize $\mathcal{L}(q(\mathbf{y}))$ in Equation (4.7), we decrease the divergence $D_{KL}\left[q(\mathbf{y})\middle\|p(\mathbf{y} \mid \mathbf{x})\right]$ between the approximate and the true posterior. When a maximum is reached, we can consider that 1) we have obtained the most accurate approximate posterior given our initial assumptions about $q(\mathbf{y})$ and 2) $\mathcal{L}(q(\mathbf{y}))$ provides a lower bound to $\log p(\mathbf{x})$. It should be noted here that the more $q(\mathbf{y})$ is flexible, the closer we can hope to get from the true posterior, but this is generally at the expense of tractability and/or computational resources.

The ELBO in Equation (4.7) is the sum of the expected log joint probability and the entropy of the approximate posterior. In order for the former to be tractable, one must carefully choose the form of the approximate posterior. The Mean-field assumption we have made allows us to select, for each factor of the approximate posteriors, a distribution with the same form as their conjugate prior, which is the best possible configuration in this context [489].

Applying now this approach to Equation (4.6), our spherical approximate posterior looks like:

$$q(\mathbf{z}, w, b) \triangleq q(\mathbf{z} \mid \boldsymbol{\theta}) q(w \mid \boldsymbol{\phi}) q(b \mid \boldsymbol{\beta}).$$

In addition, in order to recursively estimate the current posterior probability of the model parameters given the past, we make the natural approximation that the true previous posterior can be substituted by its variational approximation:

$$p(\mathbf{z} \mid \mathbf{x}_{<j}) \approx q_{j-1}(\mathbf{z}) \tag{4.8}$$

and similarly for $p(w \mid \mathbf{x}_{<j})$ and $p(b \mid \mathbf{x}_{<j})$. The use of this distribution as a proxy to the posterior greatly simplifies the optimization of $q_j(\mathbf{z}, w, b)$.

The full, approximate joint probability distribution at time j therefore looks like

$$p(x_j, \mathbf{z}, w, b|\mathbf{x}_{<j}, \boldsymbol{\theta}_0, \boldsymbol{\phi}_0) \approx p(x_j \mid \mathbf{z}) \frac{q_{j-1}(\mathbf{z} \mid \boldsymbol{\theta}_{j-1})^w p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w}}{Z(w, \boldsymbol{\theta}_{j-1}, \boldsymbol{\theta}_0)} \frac{q_{j-1}(w \mid \boldsymbol{\phi}_{j-1})^b p(w \mid \boldsymbol{\phi}_0)^{1-b}}{Z(b, \boldsymbol{\phi}_{j-1}, \boldsymbol{\phi}_0)} q(b \mid \boldsymbol{\beta}_{j-1})$$

where $\boldsymbol{\theta}_{j-1}$, $\boldsymbol{\phi}_{j-1}$ and $\boldsymbol{\beta}_{j-1}$ are the variational parameters at the last trial for $\mathbf{z}, w$ and b respectively. A further advantage of the approximation made in Equation (4.8) is that the prior of $\mathbf{z}$ and w simplifies elegantly:

$$\begin{aligned} p(x_j, \mathbf{z}, w, b|\mathbf{x}_{<j}) \approx p(x_j \mid \mathbf{z}) p(\mathbf{z} \mid w(\boldsymbol{\theta}_{j-1} - \boldsymbol{\theta}_0) + \boldsymbol{\theta}_0) \times \\ p(w \mid b(\boldsymbol{\phi}_{j-1} - \boldsymbol{\phi}_0) + \boldsymbol{\phi}_0) p(b \mid \boldsymbol{\beta}_{j-1}) \end{aligned} \tag{4.9}$$

(see Section 4.5.1 for the full derivation).

A conjugate distribution for $p(w)$ is hard to find. Šmìdl and Quinn [559] propose a uniform prior and a truncated exponential approximate posterior over w. They interpolate the normalizing constant between two fixed value of w, which allows them to perform closed-form updates of this parameter. Here, we chose $p(w \mid \boldsymbol{\phi}_0)$ and $q(w \mid \boldsymbol{\phi}_j)$ to be both beta distributions, a choice that does not impair our ability to perform closed-form updates of the variational parameters as we will see in Section 4.2.4.1.

In this model (see Figure 4.1), named the Hierarchical Adaptive Forgetting Variational Filter, specific prior configurations will bend the learning process to categorize surprising events either as contingency changes, or as accidents. In contrast with other models [520], w and b are represented with a rich probability distribution where both the expected values and variances have an impact on the model's behaviour. For a given prior belief on $\mathbf{z}$, a confident prior over w, centered on high values of this parameter, will lead to a lack of flexibility that would not be observed with a less confident prior, even if they have the same expectation.

Box 4.2.1| The HAFVF as a model of automaticity: intuition from neurobiological models

The HAFVF recalls the theory of Hebbian learning, which assumes that synaptic connexions are strengthened when a pre-synaptic activation efficiently provokes a post-synaptic activation. We re-frame the HAFVF in this perspective, aligning it with the two-factor cortico-cortical model of categorization automaticity learning [560]. This model – which is a subpart of a larger model (SPEED) that encompasses both dopamine mediated striato-cortical learning and dopamine independent cortico-cortical learning – assumes that cortico-cortical synapses are

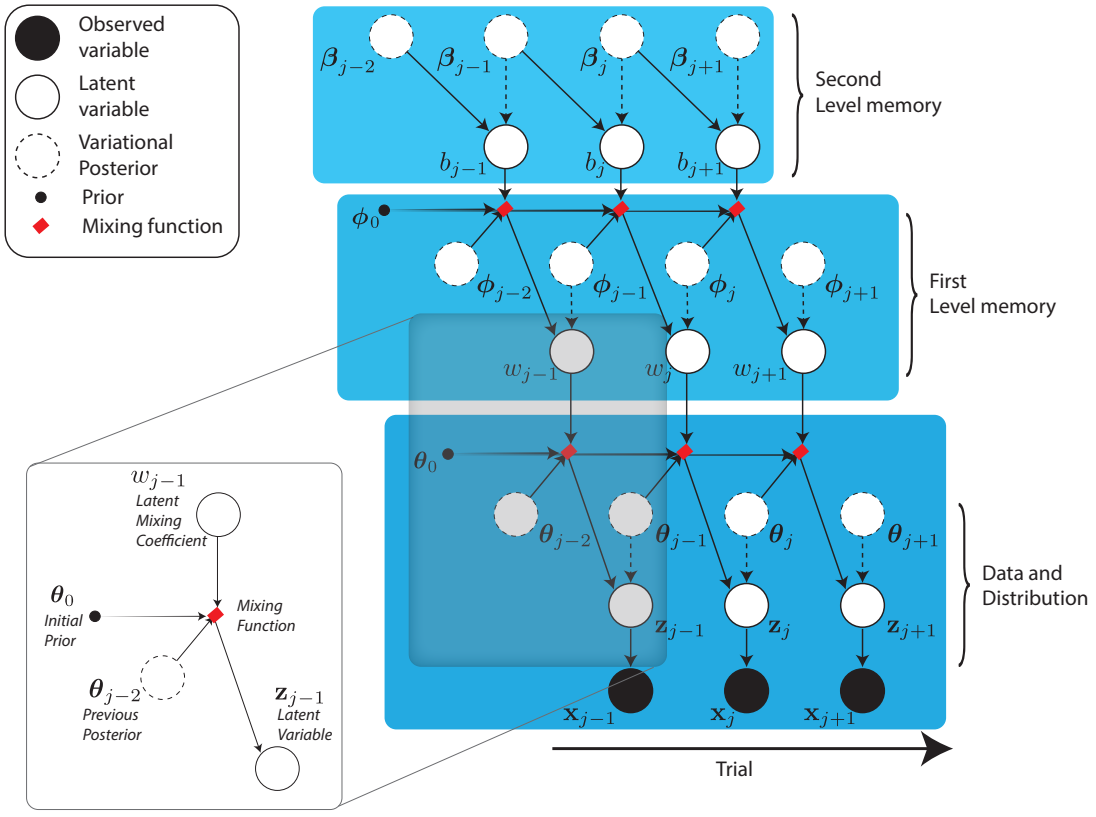


FIGURE 4.1: Directed Acyclic Graph of the HAFVF model. Plain circles represent observed variables, white circles represent latent variables and dots represents prior distribution parameters. Dashed circles and dashed arrows represent approximate posteriors and approximate posterior dependencies. A weighted prior latent node is highlighed.

strengthened whenever a supra-threshold activation occurs:

$$v_{j+1} = v_j + \alpha A(1 - v_j)\,[E - \tau_{\text{NMDA}}]^+ + \beta A v_j\,[E - \tau_{\text{NMDA}}]^-$$

where A is the total activation over the course of the trial, E is the total activation in premotor unit, τ_{NMDA} is the activation threshold and $[\cdot]^+$ and $[\cdot]^-$ are rectified linear activation units for positive and negative inputs, respectively (i.e. $[x]^+ = x \times \mathbb{1}(x > 0)$). The parameters α and β have a fixed value. This equation states that the strength of a synapse $v_j \in [0; 1]$ grows when an efficient activation occurs. This recalls the HAFVF, where the effective memory $\kappa_j \geq \kappa_0 > 0$ is the counterpart of the synapse strength.

For the exponential family, the HAFVF predicts that the memory is incremented (or decremented) when new observations are made by the following equation:

$$\kappa_{j+1} = \mathbb{E}_q\,[w]\,\kappa_j + (1 - \mathbb{E}_q\,[w])\kappa_0 + 1.$$

The condition for the effective memory κ_j to increase is that

$$\kappa_j - \kappa_0 < \frac{1}{1 - \mathbb{E}_q[w]}.$$

Recalling that $\frac{1}{1-\mathbb{E}_q[w]}$ is the efficient memory, we see that the value of w dictates whether a connection should be strengthened (Long term potentiation) or weakened (long-term depression). The former will occur if the current measure of stability tells the model (i.e. the synapse) that it is on the right track and should keep learning from past events. The latter occurs when the efficient memory is lower than the number of items kept in memory: this signal tells the model that it is overconfident about the quality of its predictions.
Coming back to the model of [560], we see that w plays the role of the activation pre-synaptic activation: when w is sufficiently high, the connection is strengthened, when it is too low it is weakened. There is, however, a crucial difference between the predictions of the HAFVF as a model of Hebbian learning and the other model outlined here, which is that the HAFVF can (and ultimately will in stable environment) stabilize its learning to some value of $\widehat{\kappa}_J = \frac{1}{1-\mathbb{E}_{q_J}[w]}$, whereas SPEED assumed that the strength of the synapse will either grow towards 1 or decrease towards 0. Therefore, the HAFVF allows for more flexibility by allowing local connections to reflect the true confidence they have in the model.

4.2.4 The critic: HAFVF as a Reinforcement Learning Algorithm

Application of this scheme of learning to the RL case is straightforward, if one considers $\mathbf{x}_j$ as being the observed rewards and $\mathbf{z}$ as the parameters of the distribution of these rewards. In the following, we will assume that the agent models a normally distributed state-action reward function $x_j = r(s,a)$, from which she tries to estimate the posterior distribution natural parameters $\mathbf{z} \triangleq \boldsymbol{\eta}(\mu(s,a), \sigma(s,a))$ where $\boldsymbol{\eta}(\cdot)$ is the natural parameter vector of the normal distribution. In this context, an intuitive choice for the prior (and the approximate posterior) of these parameters is a Normal Inverse-Gamma distribution ($\mathcal{NG}^{-1}$): for the prior, we have

$$\mu(s,a) \sim \mathcal{N}\left(\mu_0^\mu(s,a), \frac{\sigma^2(s,a)}{\kappa_0^\mu(s,a)}\right)$$
$$\sigma^2(s,a) \sim \mathcal{G}^{-1}\left(\alpha_0^\sigma(s,a), \beta_0^\sigma(s,a)\right)$$

and the approximate posterior can be defined similarly with a normal component $\mathcal{N}\left(\mu_j^\mu(s,a), \frac{\sigma^2(s,a)}{\kappa_j^\mu(s,a)}\right)$ and a gamma component $\mathcal{G}^{-1}(\alpha_j^\sigma(s,a), \beta_j^\sigma(s,a))$.

4.2.4.1 Update equation

Even though the model is not formally conjugate, the use of a mixture of priors with exponential weights makes the variational update equations easy to implement for the first level. Let us first assert a few basic principles from the Mean-Field Variational Inference framework: it can be shown that, under the assumption that the approximate posterior factorizes in $q(y_1 y_2) = q(y_1)q(y_2)$, then the optimal distribution $q^*(y_1)$ given our current estimate of $q(y_2)$ is given by

$$q^*(y_1) = \exp\left(\mathbb{E}_{q(y_2)}\left[\log p(\mathbf{x}, \mathbf{y})\right] - \log Z(\mathbf{x})\right) \tag{4.10}$$

where $\log Z$ is some log-normalizer that does not depend on $\mathbf{y}$. Equation (4.10) states that each set of variational parameters can be updated independently given the current value of the others: this usually lead to an approach similar to EM [22], where one iterates through the updates of variational posterior successively until convergence.

Fortunately, thanks to the conjugate form of the lower level of the HAFVF, Equation (4.10) can be unpacked to a form that recalls Equation (4.3) where the update of the variational parameters of $\mathbf{z}$ reads:

$$\boldsymbol{\theta}_j = \begin{bmatrix} \widehat{\boldsymbol{\vartheta}}^{\xi} + \mathbf{T}(\mathbf{x}_i) \\ \widehat{\vartheta}^{\eta} + 1 \end{bmatrix} \tag{4.11}$$

where $\widehat{\boldsymbol{\vartheta}} \triangleq \mathbb{E}_{q(w)}\left[w\right](\boldsymbol{\theta}_{j-1} - \boldsymbol{\theta}_0) + \boldsymbol{\theta}_0$ is the weighted prior of $\mathbf{z}$ (see Section 4.5.1). This update scheme can be mapped onto and be interpreted in terms of Q-learning [212] (see Section 4.5.2). Again, $\widehat{\vartheta}^{\eta} + 1$ is the updated effective memory of the subject, which is bounded on the long term by the efficient memory $1/(1 - \mathbb{E}_q\left[w\right])$.

More specifically, for the approximate posterior of a single observation stream $\mathbf{x} = \{x_j \forall j \in 1 : J\}$ with corresponding parameters $\{\mu^{\mu}_{j+1}, \kappa^{\mu}_{j+1}, \alpha^{\sigma}_{j+1}, \beta^{\sigma}_{j+1}\}$, we can apply this principle easily, as the resulting distribution has the form of a $\mathcal{NG}^{-1}$ with parameters:

$$
\begin{aligned}
\mu_j^\mu &= \frac{\widehat{\mathrm{w}}\ \kappa_{j-1}^\mu \mu_{j-1}^\mu + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^\mu \mu_0^\mu}{\kappa_j^\mu} + \frac{\mathbf{x}_j}{\kappa_j^\mu} \\
\kappa_j^\mu &= \widehat{\mathrm{w}}\ \kappa_{j-1}^\mu + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^\mu + 1 \\
\alpha_j^\sigma &= \widehat{\mathrm{w}}\ \alpha_{j-1}^\sigma + (1\text{-}\widehat{\mathrm{w}})\ \alpha_0^\sigma + \frac{1}{2} \\
\beta_j^\sigma &= \widehat{\mathrm{w}}\ \beta_{j-1} + (1\text{-}\widehat{\mathrm{w}})\ \beta_0^\sigma + \frac{1}{2}\left(\widehat{\mathrm{w}}\ \kappa_{j-1}(\mu_j^\mu - \mu_{j-1}^\mu)^2 + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0(\mu_j^\mu - \mu_0^\mu)^2 + (x_j - \mu_j^\mu)^2\right)
\end{aligned}
\tag{4.12}
$$

where we have used $\widehat{\mathrm{w}} = \mathbb{E}_{q(w)}[w]$.

Deriving updates for the approximate posterior over the mixture weights w and b is more challenging, as the optimal approximate posterior in Equation (4.10) does not have the same form as the beta prior due to the non-conjugacy of the model. Fortunately, non-conjugate variational message passing (NCVMP) [70] can be used in this context. In short, NCVMP minimizes an approximate KL divergence in order to find the value of the approximate posterior parameters that maximize the ELBO. Although NCVMP convergence is not guaranteed, this issue can be somehow alleviated by damping of the updates (i.e. updating the variational parameters to a value lying in between the previous value they endorsed and the value computed using NCVMP, see [70] for more details). The need for a closed-form formula of the expected log-joint probability constitutes another obstacle for the naive implementation of NCVMP to the present problem: indeed, computing the expected value of the log-partition functions $\log Z(w)$ and $\log Z(b)$ involves a weighted sum of the past variational parameters $\boldsymbol{\theta}_{j-1}$ and the prior $\boldsymbol{\theta}_0$, which are known, with a weight w, which is unknown. Expectation of this expression given $q(w)$ does not, in general, have an analytical expression. To solve this problem, we used the second order Taylor expansion around $\widehat{\mathrm{w}}$ (see Section 4.5.1).

4.2.4.2 Counterfactual learning

As an agent performs a series of choices in an environment, she must also keep track of the actions not chosen and update her belief accordingly: ideally, the variance of the approximate posterior of the reward function associated with a given action should increase when that action is not selected, to reflect the increased uncertainty about its outcome during the period when no outcome was observed. This requirement implies counterfactual learning capability [561, 562, 563].

Two options will be considered here: the first option consists in updating the approximate posterior parameters of the non-selected action at each time step with an update

scheme that pulls the approximate posterior progressively towards the prior $\boldsymbol{\theta}_0$, with a speed that depends on w, i.e. as a function of the belief the agent has about environment stability. The second approach will consist in updating the approximate posterior of the actions only when they are actually selected, but accounting for past trials during which that action was not selected. The mathematical details of these approaches are detailed in Section 4.5.3.

4.2.4.2.1 Continuous Updating When an action is not selected, the agent can infer what value it would have had given the observed stability of the environment. In practice, this is done by updating the variational parameters of the selected *and* non-selected action using the observed reward for the former, and the expected reward and variance of this reward for the latter.

This approach can be beneficial for the agent in order to optimize her exploration/exploitation balance. In Section 4.5.3.1, it is shown that if the agent has the prior belief that the reward variance is high, then the probability of exploring the non-chosen option will increase as the lag between the current trial and the last observation of the reward associated with this option increases.

4.2.4.2.2 Delayed Updating Even though the agent learns only actions that are selected, it can adapt its learning rate as a function of how distant in the past was the last time the action was selected. Formally, this approach considers that *if* the posterior probability had been updated at each time step and the forgetting factor w had been stable, the impact of the observations n trials back in time would currently have an influence that would decrease geometrically with a rate $\omega \triangleq w^n$. We can then substitute the prior over $\mathbf{z}$ by:

$$p(\mathbf{z}|\mathbf{x}_{<j}, \delta_j) \triangleq \frac{p_{j-1}(\mathbf{z} \mid \mathbf{x}_{<j})^{\omega} p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-\omega}}{Z(\omega, \mathbf{x}_{<j}, \boldsymbol{\theta}_0)} \tag{4.13}$$

which is identical to Equation (4.4) except that w has been substituted by ω.

We name this strategy Delayed Approximate Posterior Updating, and present the expected results with this scheme in Section 4.3.

4.2.4.3 Temporal Difference Learning

An important feature required for an efficient Model-Free RL algorithm is to be able to account for future rewards in order to make choices that might seem suboptimal to a

myopic agent, but that make sense on the long run. This is especially useful when large rewards (or the avoidance of large punishments) can be expected in a near future.

In order to do this, one can simply sum the expected value of the next state to the current reward in order to perform the update of the reward distribution parameters. However, because the evolution of the environment is somehow chaotic, it is usually considered wiser to decay slightly the future rewards by a discount rate γ. This mechanism is in accordance with many empirical observations of animal behaviours [564, 565, 566], neurophysiological processes [567, 568, 569] and theories [570, 571, 572].

As the optimal value of γ is unknown to the agent, we can assume that she will try to estimate its posterior distribution from the data as she does for the mean and variance of the reward function. Section 4.5.4 shows how this can be implemented in the current context. An example of TD learning in a changing environment is given in Section 4.3.4.

We now focus on the problem of decision making under the HAFVF.

4.2.5 The actor: Decision Making under the HAFVF

4.2.5.1 Bayesian Policy

In a stable environment where the distribution of the action values are known precisely, the optimal choice (i.e. the choice that will maximize reward on the long run) is the choice with the maximum expected value: indeed, it is easy to see that if $\mathbb{E}[r(s,a_1)] > \mathbb{E}[r(s,a_2)]$, then $\mathbb{E}[\sum_n r_n(s,a_1)] > \mathbb{E}[\sum_n r_n(s,a_2)]$ (here and for the next few paragraphs, we will restrict our analysis to the case of single stage tasks, and omit the s input in the reward function). However, in the context of a volatile environment, the agent has no certainty that the reward function has not changed since the last time she visited this state, and she has no precise estimate of the reward distribution. This should motivate her to devote part of her choices to exploration rather than exploitation. In a randomly changing environment, there is no general, optimal balance between the two, as there is no way to know how similar is the environment wrt the last trials.The best thing an agent can do is therefore to update her current policy wrt her current estimate of the uncertainty of the latent state of the environment.

Various policies have been proposed in order to use the Bayesian belief the agent has about its environment to make a decision that maximizes expected rewards in the long run. We review briefly Q-value sampling (QS) [186][1]. Our framework can also be

[1]Here, we use the terminology Q-value sampling in accordance with Dearden [162, 186], but one should recall that QS is virtually indistinguishable from Thompson sampling, see Paragraph 1.2.3.1.1.

connected to another algorithm used in the study of animal RL [198, 573], based on the Value of Perfect Information (VPI) [162, 208, 574], which we describe in Section 4.5.5.

QS [186] is an exploration policy based on the posterior predictive probability that an action value exceeds all the other actions available:

$$p(a) \triangleq p(a = \arg\max_{\mathbf{a}} \mu(\mathbf{a})) = \mathbb{E}_{p(\mu(\mathbf{a}))}\left[p\big(\mu(a) > \mu(a')\,\forall a' \neg a\big)\right]. \tag{4.14}$$

The expectation of Equation (4.14) provides a clear policy to the agent. The QS approach is compelling in our case: in general, the learning algorithm we propose will produce a trial-wise posterior probability that should, most of the time, be easy to sample from.

The policy dictated by QS is optimal provided that the subject has an equal knowledge of all the options she has. If the environment is only partly explored, the value of some actions may be overestimated (or underestimated), in which case QS will fail to detect that exploration might be beneficial. QS can lead to the same policy when two actions (a_1 and a_2) have similar uncertainty associated with their reward distribution ($\sigma_1 = \sigma_2$) but a different reward ($\mu_1 > \mu_2$), and when one action has a much larger expected reward ($\mu_1 \gg \mu_2$) but also larger uncertainty ($\sigma_1 \gg \sigma_2$) (see [162] for an example). This can be sub-optimal, as the action with the larger uncertainty could lead to a higher (or lower) reward than expected: in this specific case, choosing the action with the largest expected reward should be even more encouraged due to the lack of of knowledge about its true reward distribution, which might be much higher than expected. A strategy to solve this problem is to correct the policy by a factor, the Value of Perfect Information, that reflects the ratio between the potential information gain that will follow the selection of an action with the reward cost of this choice. This approach, and its relationship to our algorithm, is discussed in Section 4.5.5.

4.2.5.2 Q-Sampling as a Stochastic Process

Let us get back to the case of QS, and consider an agent solving this problem using a gambler ruin strategy [350]. We assume that, in the case of a two-alternative forced choice task, this agent has equal initial expectations that either a_1 or a_2 will lead to the highest reward, represented by a start point $z_0 = \zeta/2$, where ζ will be described shortly. The gambler ruin process works as follows: this agent samples a value $\widetilde{r}\,(a_1) \sim p\,(r(a_1) \mid \mathbf{x}_{<j})$ and a value $\widetilde{r}\,(a_2) \sim p\,(r(a_2) \mid \mathbf{x}_{<j})$ and assess which one is higher. If $\widetilde{r}\,(a_1)$ beats $\widetilde{r}\,(a_2)$, she computes the number of wins of a_1 until now as $z_1 = z_0 + 1$, and displaces her belief the other way $(z_0 - 1)$ if a_2 beats a_1. Then, she starts again and moves in the direction indicated by $\text{sign}(r(a_1) - r(a_2))$ until she reaches one of the

two arbitrary thresholds situated at 0 or ζ that symbolize the two actions available. It is easy to see that the number of wins and looses generated by this procedure gives a Monte Carlo sample of $p(r(a_1) > r(a_2) \mid \mathbf{x}_{<j})$. We show in Section 4.5.6.1 that this process tends to deteministically select the best option as the threshold grows.

So far, we have studied the gambler ruin problem as a discrete process. If the interval between the realization of two samples tends to 0, this accumulation of evidence can be approximated by a continuous stochastic process [350]. When the rewards are normally distributed, as in the present case, this results in a Wiener process, or Drift-Diffusion model (DDM, [1]), since the difference between two normally distributed random variables follows a normal distribution. This stochastic accumulation model has a displacement rate (drift) that is given by

$$d\frac{z}{dt} \sim \mathcal{N}(\mu(a_1) - \mu(a_2), \sigma^2(a_1) + \sigma^2(a_2)) \qquad \text{see [575].}$$

Crucially, it enjoys the same convergence property of selecting almost surely the best option for high thresholds (see Section 4.5.6.2).

4.2.5.2.1 Sequential Q-Sampling as a Full-DDM model This simple case of a fixed-parameters DDM, however, is not the one we have to deal with, as the agent does not know the true value of $\{\boldsymbol{\mu}(a), \boldsymbol{\sigma}^2(a) \, \forall a \in \mathbf{a}\}$, but she can only approximate it based on her posterior estimate. Assuming that the approximate posterior over the latent mean and variance of the reward distribution is a $\mathcal{NG}^{-1}$ distribution, and keeping the original statement $d\frac{X}{dt} = r(a_1) - r(a_2)$, we have

$$\begin{aligned} d\frac{X}{dt} &\sim \mathcal{N}(\widetilde{\mu}_1 - \widetilde{\mu}_2, \widetilde{\sigma}_1^2 + \widetilde{\sigma}_1^2) \\ \text{where} \quad \widetilde{\mu}_i, \widetilde{\sigma}_i^2 &\sim \mathcal{N}(\mu_i^\mu, \sigma_i^2/\kappa_i^\mu)\, \mathcal{G}^{-1}(\alpha_i^\sigma, \beta_i^\sigma) \quad \text{for } i = \{1, 2\} \\ \Leftrightarrow \quad \widetilde{\mu}_1 - \widetilde{\mu}_2 &\sim \mathcal{N}(\mu_1^\mu - \mu_2^\mu, \tfrac{\sigma_1^2}{\kappa_1^\mu} + \tfrac{\sigma_2^2}{\kappa_2^\mu}) \\ \widetilde{\sigma}_1^2 &\sim \mathcal{G}^{-1}(\alpha_1^\sigma, \beta_1^\sigma) \\ \widetilde{\sigma}_2^2 &\sim \mathcal{G}^{-1}(\alpha_2^\sigma, \beta_2^\sigma) \end{aligned} \tag{4.15}$$

and where, for the sake of sparsity of the notation, the indices are used to indicate the corresponding action-related variable.

To see how such evidence accumulation process evolves, one can discretize Equation (4.15): this would be equivalent to sample at each time t a displacement

$$\Delta x = \Delta t \; \widetilde{\xi} + \sqrt{\Delta t} \; \widetilde{\zeta} \; \epsilon \tag{4.16}$$

where Δx stands for $x_t - x_{t-1}$. The drift $\tilde{\xi}$ in Equation (4.16) is sampled as the difference between two sampled means $\tilde{\mu}_1 - \tilde{\mu}_2$ and the squared noise $\tilde{\varsigma}^2$ is sampled as the sum of the two sampled variances $\tilde{\sigma}_1^2 + \tilde{\sigma}_2^2$. At each time step (or at each trial, quite similarly as we will show), the tuple of parameters $\{\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}_1^2, \tilde{\sigma}_2^2\}$ is drawn from the current posterior distribution.

Hereafter, we will refer to this process as the Normal-Inverse-Gamma Diffusion Process, or *NIGDM*.

4.2.5.3 NIGDM as an exploration rule

Importantly, and similarly to QS, this process has the desired property of selecting actions with a probability proportional to their probability of being the best option. This favours exploratory behaviour since, assuming equivalent expected rewards, actions that are associated with large reward uncertainty will tend to be selected more often. We show that the NIGDM behaves like QS in Section 4.5.6.3. There, it is shown (Theorem 4.4) that, as the threshold grows, the NIGDM choice pattern resembles more and more the QS algorithm. For lower values of the threshold, this algorithm is less accurate than QS (see further discussion of the property of NIGDM for low thresholds in Section 4.5.5).

Box 4.2.2| Biasing decisions before accumulation: the role of starting point

A constellation of studies have suggested that decisions are not made based solely on the current observation, but also on some conditioning that ensues from past experience. It is indeed well known that the brain works not only by reacting to the current contingency, but also by predicting upcoming events [576, 577, 578]. For instance, model-based RL itself is based upon the assumption (and observation) that neural population simulate upcoming events to make a decision [259]. Also, it has been shown that assuming that action sequences can be selected as macro can explain the emergence of inflexible behaviours with training without recurring to a clear-cut dual component learning model (we refer the reader to the discussion of deciding in POMDPs in Box 1.2.10). Here, we suggest that the bias in decision (the starting point in DDM terms) reflects how some prior knowledge about state transitions can guide decisions when no data is available yet: by adapting its starting point, an accumulator can deal with the urgency associated with decision making and make use of all the time available between the end of an action and the presentation of the successive state.

Here, we assume that the agent has some knowledge of the transition probability of state to state given a certain action. We also assume that following some decision

rule, the agent is able to observe its own policy and learn it in a state-action pairing paradigm. Formally, in a mean-field framework and without forgetting, this posterior learning of the decision contingency takes the form of a multinomial distribution with a Dirichlet approximate posterior. Each time an agent takes a decision $a_k^j \in \mathbf{a}$ in state $s_i^j \in \mathbf{s}$, it updates the approximate posterior $q(\boldsymbol{\alpha} \mid \boldsymbol{\delta})$ over the decision probability $p(\mathbf{a} \mid \boldsymbol{\alpha})$ according to

$$\delta_k^j = \delta_k^{j-1} + 1.$$

As soon as an action has been selected in $j-1$, the agent can start simulating a range of possible next states given its transition belief, and from there simulate possible actions based on the past observed policy. Evidence can therefore be accumulated in favour of each action available before observations are provided to the agent: at $t = 0$, the accumulation process is already biased toward some preferred actions. Because this policy was outcome or reward-driven, this strategy will lead to select rewarded actions after some training.

The advantage of this decision scheme becomes clear in the context of Semi-MDP (Box 1.2.1) and when time pressure is high: it might be the case that not enough observations can be retrieved from the environment to clearly discriminate the current state. Therefore, one needs to take advantage of past observed transitions to make a decision when the current state is ill-defined. When sensory evidence is provided, the decision scheme resembles to a POMDP posterior evaluation of the current state (or action) conditioned on transitions and observations. If the transitions are highly predictable, sensory information can be discarded and decisions can be anticipated, giving rise to a chunking of action sequences.

The proposed decision framework is based upon the observation that human decisions are conditioned on past experience in a way that is compatible with a change of starting point [380]. It is also compatible with intracerebral recordings showing that the hippocampal pre-activation facilitates decision making in predictable environments [579] and that RT are usually shorter when pre-activation is high. Finally, it recalls the various neurocomputational theories that argue for a critical role of policy learning in decision making [580, 581, 582] where stimuli are paired directly with actions, with or without recurring to value learning [246, 583].

4.2.5.4 Cognitive cost optimization under the AC-HAFVF

Algorithm 11 summarizes the AC-HAFVF model. This model ties together a learning algorithm - that adapts how fast it forgets its past knowledge on the basis of its assessment of the stability of the environment - with a decision algorithm that makes full use of the posterior uncertainty of the reward distribution to balance exploration and exploitation.

Importantly, these algorithms make time and resource costs explicit: for instance, time constraints can make decisions less accurate, because they will require a lower decision threshold. Figure 4.2 illustrates this interpretation of the AC-HAFVF by showing how accuracy and speed of the model vary as a function of the variance of the reward estimates: the cognitive cost of making a decision using the AC-HAFVF is high when the agent is uncertain about the reward mean (low κ_j) but has a low expectation of the variance (low β_j). In these situations, the choices were also more random, allowing the agent to explore the environment. Decisions are easier to make when the difference in mean rewards is clearer or when the rewards were more noisy.

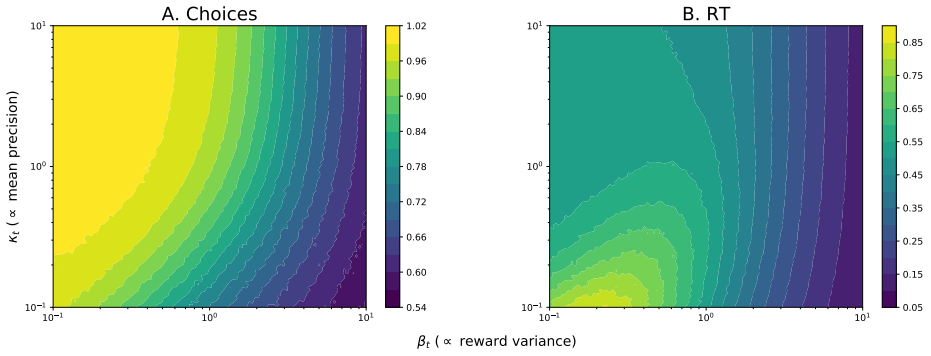


FIGURE 4.2: Policies for the AC-HAFVF as a function of reward variance β_j and number of effective observations κ_j, for a fixed value of posterior mean rewards ($\mu_1 = -\mu_2 = 1$), shape parameter $\alpha_1 = \alpha_2 = 3$ threshold $\zeta = 2$, start point $z_0 = \zeta/2$ and $\tau = 0$. **A.** Choices were more random for more noisy reward distributions (i.e. high values of β_j) and for mean estimates with a higher variance (i.e. with a lower number of observations κ_j). **B.** Decisions were faster when the difference of the means was clearer (high κ_j) and when the reward distributions was noisy (high β). Subjects were slower to decide what to do for noisy mean values but precise rewards, reflecting the high cognitive cost of the decision process in these situations.

Besides the decision stage, the computational cost of the inference step can also be determined. To this end, one must first consider that the HAFVF updates the variational posterior using a natural gradient-based [77] approach with a Fisher preconditioning matrix [584, 70, 76]: to learner computes a gradient of the ELBO wrt the variational parameters, then moves in the direction of this gradient with a step length proportional to the posterior variance of the loss function. Because of this, the divergence between the

true posterior probability $p(\mathbf{z} \mid \mathbf{x}_{\leq j})$ and the prior probability $p(\mathbf{z})$ (i.e. the mixture of the default distribution and previous posterior) conditions directly the expected number of updates required for the approximate posterior to converge to a minimum $D_{KL}\left[\left|\right|\right]$ (see for instance [585]). Also, in a more frequentist perspective and at the between-trial time scale, convergence rate of the posterior probability towards the true (if any) model configuration is faster when the KL divergence between this posterior and the prior is small [586]: in other words, in a stable environment, a higher confidence in past experience will require less observations for the same rate of convergence, because it will tighten the distance between the prior and posterior at each time step.

These three aspects of computational cost (for decision, for within-trial inference and for across-trial inference) can justify the choice (or emergence) of lower flexible behaviours as they ultimately maximize the reward rate [198, 587]: indeed, such strategies will lead in a stable environment to faster decisions, to faster inference at each time step and to a more stable and accurate posterior.

Algorithm 11: AC-HAFVF. For simplicity, the NIGDM process has been discretized.

input: prior belief $\{\boldsymbol{\theta}_0, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0\}$
for $j = 1$ ***to*** J **do**
  Actor: $NIGDM$;
  input: Start point z_0, threshold ζ, non-decision time τ
  $k \leftarrow 0$;
  sample $\widetilde{\mu}_i, \widetilde{\sigma}_i^2 \sim \mathcal{N}(\mu_i^\mu, \sigma_i^2/\kappa_i^\mu)\, \mathcal{G}^{-1}(\alpha_i^\sigma, \beta_i^\sigma)$ for $i = \{1, 2\}$;
  while $0 < z_k < \zeta$ **do**
    $k \mathrel{+}= 1$;
    sample $\delta_z \sim \mathcal{N}\left(\widetilde{\mu}_1 - \widetilde{\mu}_2, \widetilde{\sigma}_1^2 + \widetilde{\sigma}_2^2\right)$;
    move $z_k \mathrel{+}= \delta_z$;
  end
  select $a_j \leftarrow \begin{cases} 1 & \text{if } z_k \geq \zeta \\ 2 & \text{otherwise} \end{cases}$;
  Get reward $r_j = r(s_j, a_j)$, Observe transition $s_{j+1} = j(s_j, a_j)$;

  Critic: $HAFVF$;
  $ELBO \leftarrow -\infty$;
  while $|\delta_L| \leq 10^{-3}$ **do**
    update $\boldsymbol{\theta}_j(s_j, a_j) = \arg\max_{\boldsymbol{\theta}_j(s_j,a_j)} \mathcal{L}(q(\boldsymbol{\theta}_j(s_j, a_j), \boldsymbol{\phi}_j, \boldsymbol{\beta}_j))$ using CVMP;
    update $\{\boldsymbol{\phi}_j, \boldsymbol{\beta}_j\}$ using NCVMP;
    $\delta_L \leftarrow \mathcal{L}(q(\boldsymbol{\theta}_j(s_j, a_j), \boldsymbol{\phi}_j, \boldsymbol{\beta}_j)) - ELBO$;
    $ELBO \leftarrow \mathcal{L}(q(\boldsymbol{\theta}_j(s_j, a_j), \boldsymbol{\phi}_j, \boldsymbol{\beta}_j))$;
  end
end

Box 4.2.3| Adapting the threshold to posterior variance

In Section 1.3.1.3, we suggested that the observation that the tonic dopamine level was associated with less random responses and faster RTs (which is equivalent to a lower exploration/exploitation trade-off) was compatible with the view that the tonic dopamine level linked the measure of the posterior variance to a corresponding decision threshold. However, in the simulations we have ran in this chapter, the threshold was kept constant throughout the experiment, as exploration/exploitation adaptation to learning was not the primary aim of this study. Additional experiments should be conducted to investigate the possibility that the threshold response is indeed adapted proportionally to the posterior variance.

4.2.6 Fitting the AC-HAFVF

So far, we have provided all the necessary tools to simulate behavioural data using the AC-HAFVF. It is now necessary to show how to fit model parameters to an acquired dataset. We will first describe how this can be done in a Maximum Likelihood framework, before generalizing this method to Bayesian inference using variational methods.

The problem of fitting the AC-HAFVF to a dataset can be seen as a State-Space model fitting problem. We consider the following family of models:

$$\begin{cases} \boldsymbol{\Omega}_{\mathrm{j,n}} &= \underbrace{f(\boldsymbol{\Omega}_{\mathrm{j\text{-}1,n}}, \boldsymbol{\Omega}_{0,\mathrm{n}}, \mathbf{x}_{j-1,n})}_{HAFVF} \\ \mathbf{y}_{j,n} &\sim \underbrace{\mathbb{E}_{p(\mu_1,\mu_2,\sigma_1^2,\sigma_2^2\,|\,\boldsymbol{\Omega}_{\mathrm{j,n}})}\left[Wiener\left(\mu_1-\mu_2, \sqrt{\sigma_1^2+\sigma_2^2}, \zeta_n, z_{0n}, \tau_n\right)\right]}_{NIGDM} \end{cases}$$
$$\text{where } \boldsymbol{\Omega}_{\mathrm{j,n}} = \{\boldsymbol{\mu}^\mu, \boldsymbol{\kappa}^\mu, \boldsymbol{\alpha}^\sigma, \boldsymbol{\beta}^\sigma, \boldsymbol{\phi}, \boldsymbol{\beta}\}_{j,n}$$
$$\text{and } \mathbf{y}_{j,n} = \{\mathrm{t}, a\}_{j,n} \tag{4.17}$$

and $\mathrm{t}_{j,n}$ stands for the reaction time associated with the state-action pair (s,a) of the subject n at the trial j. Unlike many State-Space models, we have made the assumption in Equation (4.17) that the transition model $\boldsymbol{\Omega}_{\mathrm{j,n}} = f(\boldsymbol{\Omega}_{\mathrm{j\text{-}1,n}}, \boldsymbol{\Omega}_{0,\mathrm{n}}, \mathbf{x}_{j-1,n})$ is entirely deterministic given the subject prior $\boldsymbol{\Omega}_{0,\mathrm{n}}$ and by the observations $\mathbf{x}_{<j}$, which is in accordance with the model of decision making presented in Section 4.2.5[2].

[2]The Bayesian procedure we will adopt hereafter is formally identical to considering that the drift and noise are drawn according to the rules defined in Section 4.2.5, making this model equivalent to:

$$\mathbf{y}_{j,n} \sim Wiener(\xi, \tilde{\varsigma}, \zeta_n, z_{0n}, \tau_n)$$
$$\xi, \tilde{\varsigma} \sim f(\boldsymbol{\Omega}_{\mathrm{j\text{-}1,n}}, \boldsymbol{\Omega}_{0,\mathrm{n}}, \mathbf{x}_{j-1,n})$$

Quite importantly, we have made the assumption in Equation (4.17) that the threshold, the non-decision time and the start-point were fixed for each subject throughout the experiment. This is a strong assumption, that might be relaxed in practice. To simplify the analysis, and because it is not a mandatory feature of the model exposed above, we do not consider this possibility here and leave it for further developments.

We can now treat the problem of fitting the AC-HAFVF to behavioural data as two separate sub-problems: first, we will need to derive a differentiable function that, given an initial prior $\mathbf{\Omega}_{0,\mathrm{n}}$ and a set of observations $\mathbf{x}_n$ produces a sequence $\boldsymbol{\theta}_n = \{\mathbf{\Omega}_{\mathrm{j,n}} \forall j \in 1:J\}$, and second (Section 4.2.6.1) a function that computes the probability of the observed behaviour given the current variational parameters.

4.2.6.1 Maximum A Posteriori Estimate of the AC-HAFVF

The update equations described in Section 4.2.4.1 enable us to generate a differentiable sequence of approximate posterior parameters $\mathbf{\Omega}_{\mathrm{j,n}}$ given some prior $\mathbf{\Omega}_{0,\mathrm{n}}$ and a sequence of choices-rewards $\mathbf{x}$. We can therefore reduce Equation (4.17) to a loss function of the form

$$\log p(\mathbf{y}_{j,n} \mid \mathbf{x}_{\leq j,n}; \mathbf{\Omega}_{0,\mathrm{n}}, \zeta_n, z_{0n}, \tau_n)$$

whose gradient wrt $\mathbf{\Omega}_{0,\mathrm{n}}$ can be efficiently computed using the chain rule:

$$\begin{aligned}
&\nabla_{\mathbf{\Omega}_{0,\mathrm{n}}} \{\log p(\mathbf{y}_{j,n} \mid \mathbf{x}; \mathbf{\Omega}_{0,\mathrm{n}}, \zeta_n, z_{0n}, \tau_n)\} = \\
&\quad \nabla_{\mathbf{\Omega}_{0,\mathrm{n}}} \{f(\mathbf{\Omega}_{0,\mathrm{n}}, \mathbf{x}_{\leq j})\}^T \nabla_{\mathbf{\Omega}_{\mathrm{j,n}}} \{\log p(\mathbf{y}_{j,n} \mid \mathbf{\Omega}_{\mathrm{j,n}}, \zeta_n, z_{0n}, \tau_n)\} \\
&\quad \text{where } \mathbf{\Omega}_{\mathrm{j,n}} = f(\mathbf{\Omega}_{0,\mathrm{n}}, \mathbf{x}_{\leq j}).
\end{aligned}$$

Recall that $\nabla_{\mathbf{\Omega}_{0,\mathrm{n}}} \{f(\mathbf{\Omega}_{0,\mathrm{n}}, \mathbf{x}_{\leq j})\}^T$ is the jacobian (i.e. matrix of partial derivative) of $\mathbf{\Omega}_{\mathrm{j,n}}$ wrt each of the elements of $\boldsymbol{\theta}_0$ that are optimized, and $\nabla_{\mathbf{\Omega}_{\mathrm{j,n}}} \{\log p(\mathbf{y}_{j,n} \mid \mathbf{\Omega}_{\mathrm{j,n}}, \zeta_n, z_{0n}, \tau_n)\}$ is the gradient of the loss function (i.e. the NIGDM) wrt the output of $f(\cdot)$.

As the variational updates that lead to the evaluation of $\mathbf{\Omega}_{\mathrm{j,n}}$ are differentiable, the use of VB makes it possible to use automatic Differentiation to compute the Jacobian of $\mathbf{\Omega}_{\mathrm{j,n}}$ wrt $\mathbf{\Omega}_{0,\mathrm{n}}$.

The next step, is to derive the loss function $\log p(\mathbf{y}_{j,n} \mid \mathbf{\Omega}_{\mathrm{j,n}}, \zeta_n, z_{0n}, \tau_n)$. In this log-probability density function, the local, trial-wise parameters

$$\boldsymbol{\chi}_{j,n} \triangleq \left\{\xi_{j,n}, \lambda^{-1}\left(\sigma^2_{j,n}(a_1)\right), \lambda^{-1}\left(\sigma^2_{j,n}(a_2)\right)\right\}$$

have been marginalized out. This makes its evaluation hard to implement with conventional techniques. Variational methods can be used to retrieve an approximate Maximum A Posteriori (MAP) in these cases (Chapter 2). The method is detailed in Section 4.5.7. Briefly, VB is used to compute a lower bound ($\ell_{j,n} \leq \log p(\mathbf{y}_{j,n} \mid \mathbf{\Omega}_{\mathrm{j,n}}, \zeta_n, z_{0n}, \tau_n)$) to the marginal posterior probability described above for each trial. Instead of optimizing each variational parameters independently, we optimize the parameters $\boldsymbol{\rho}$ of an inference network [110] that maps the current HAFVF approximate posterior parameters $\mathbf{\Omega}_{\mathrm{j,n}}$ and the data $\mathbf{y}_{j,n}$ to each trial-specific approximate posterior. This amortizes greatly the cost of the optimization (hence the name Amortized Variational Inference, AMI), as the non-linear mapping (e.g. multilayered perceptron) $h(\mathbf{y}_{j,n}; \boldsymbol{\rho})$ can provide the approximate posterior parameters of any datapoint, even if it has not been observed yet. We chose $q_{\boldsymbol{\rho}}(\boldsymbol{\chi}_{j,n} \mid \mathbf{y}_{j,n})$ to be a multivariate Gaussian distribution, which leads to the following form of variational posterior:

$$\begin{aligned} q_{\boldsymbol{\rho}}(\boldsymbol{\chi}_{j,n} \mid \mathbf{y}_{j,n}) &\triangleq \mathcal{N}(\mu^{\boldsymbol{\chi}_{j,n}}, \Sigma^{\boldsymbol{\chi}_{j,n}}) \\ \text{where } \mu^{\boldsymbol{\chi}_{j,n}}, L^{\boldsymbol{\chi}_{j,n}} &= h(\mathbf{y}_{j,n}; \boldsymbol{\rho}); \boldsymbol{\rho}) \end{aligned} \tag{4.18}$$

and $L^{\boldsymbol{\chi}_{j,n}}$ is the lower Cholesky factor of $\Sigma^{\boldsymbol{\chi}_{j,n}}$. Another consideration is that, in order to use the multivariate normal approximate posterior, the unbounded variances sample $\lambda^{-1}\left(\sigma^2_{j,n}(a_i)\right)$, $i = 1, 2$ must be transformed to an unbounded space. We used the inverse softplus transform $\lambda \triangleq \log\left(\exp\left(\cdot\right) - 1\right)$, as this function has a bounded gradient, in contrasts with the exponential mapping, which prevents numerical overflow. We found that this simple trick could regularize greatly the optimization process. However, this transformation of the normally distributed $\lambda^{-1}(\cdot)$ variables requires us to correct the ELBO by the log-determinant of the Jacobian of the transform [103], which for the sofplus transform of x is simply $\log|\delta\frac{\lambda(x)}{\delta x}| = -\log\left(1 + \exp(-x)\right)$.

The same transformation can be used for the parameters of $\boldsymbol{\theta}_0$ that are required to be greater than 0 (i.e. all parameters except μ_0), which obviously do not require any log-Jacobian correction.

In Equation (4.18), we have made explicit that we used the three latent variables: the drift rate and the two action-specific noise parameters. This is due to the fact that, unfortunately, the distribution of the sum of two Inverse-Gamma distributed random variables does not have a closed form formula, making the use of single random variable $\varsigma^2_{j,n} = \sigma^2_{j,n}(a_1) + \sigma^2_{j,n}(a_2)$ challenging, whereas it can be done easily for $\xi_{j,n} = \mu_{j,n}(a_1) - \mu_{j,n}(a_2)$, which is normally distributed (see Equation (4.15)).

The final step to implement a MAP estimation algorithm is to set a prior for the parameters of the model. We used a simple L2 regularization scheme, which consists trivially in a normal prior $\mathcal{N}(0, 1)$ over all parameters, mapped onto an unbounded space if needed.

Algorithm 12 shows how the full optimization procedes.

Algorithm 12: MAP estimate of AC-HAFVF parameters.

input: Data $\mathbf{x} = \{r_{j,n}, \mathbf{y}_{j,n} \text{ for } j = 1 \textbf{ to } J,\ n = 1 \textbf{ to } N\}$

initialize $\boldsymbol{\Omega}_{\text{n}} = \{\boldsymbol{\theta}_0, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0\}_n$ **for** $n = 1$ **to** N and IN[§] weights $\boldsymbol{\rho}$;

repeat

  set $L \leftarrow 0$;

  for $n = 1$ ***to*** N **do**

    Set $\widetilde{\nabla}_{\boldsymbol{\rho},\boldsymbol{\Omega}_{0,\text{n}},\zeta_n,z_{0n},\tau_n} \leftarrow 0$;

    for $j = 1$ ***to*** J **do**

      Learning Step: $HAFVF$;

      Get $\boldsymbol{\Omega}_{\text{j,n}} = f(\boldsymbol{\Omega}_{\text{j-1,n}}, \boldsymbol{\Omega}_{0,\text{n}}, \mathbf{x}_{j-1,n})$

        and Jacobian wrt $\boldsymbol{\Omega}_{0,\text{n}}$: $\nabla_{\boldsymbol{\Omega}_{0,\text{n}}} \{f(\boldsymbol{\Omega}_{0,\text{n}}, \mathbf{x}_{\leq j})\}$ using FAD*;

      Decision Step: $HAFVF$;

      Get ELBO $\ell_{j,n}$ of $\log p(\mathbf{y}_{j,n} \mid \boldsymbol{\Omega}_{\text{j,n}}, \zeta_n, z_{0n}, \tau_n)$ using AVI†;

        and corresponding gradient $\nabla_{\boldsymbol{\rho},\boldsymbol{\Omega}_{\text{j,n}},\zeta_n,z_{0n},\tau_n} \{\ell_{j,n}\}$ using RAD‡;

      Increment $L \mathrel{+}= \ell_{j,n}$;

      Compute gradient wrt $\boldsymbol{\Omega}_{0,\text{n}}$ using the chain rule:

        $\widetilde{\nabla}_{\boldsymbol{\Omega}_{0,\text{n}}} \mathrel{+}= \nabla_{\boldsymbol{\Omega}_{0,\text{n}}} \{f(\boldsymbol{\Omega}_{0,\text{n}}, \mathbf{x}_{\leq j})\} \nabla_{\boldsymbol{\Omega}_{\text{j,n}}} \{\ell_{j,n}\}$;

      Increment gradient of DDM and IN parameters

        $\widetilde{\nabla}_{\boldsymbol{\rho},\zeta_n,z_{0n},\tau_n} \mathrel{+}= \nabla_{\boldsymbol{\rho},\zeta_n,z_{0n},\tau_n} \{\ell_{j,n}\}$;

    end

    L2-norm regularization:

      $L \mathrel{+}= -\frac{1}{2}\|\{\boldsymbol{\Omega}_{0,\text{n}}, \zeta_n, z_{0n}, \tau_n\}\|_2^2$;

      $\widetilde{\nabla}_{\boldsymbol{\Omega}_{0,\text{n}},\zeta_n,z_{0n},\tau_n} \mathrel{+}= -\{\boldsymbol{\Omega}_{0,\text{n}}, \zeta_n, z_{0n}, \tau_n\}$;

  end

  Perform gradient step $\boldsymbol{\rho}, \boldsymbol{\Omega}_{0,\text{n}}, \zeta_n, z_{0n}, \tau_n \mathrel{+}= \eta\, \widetilde{\nabla}_{\boldsymbol{\rho},\boldsymbol{\Omega}_{0,\text{n}},\zeta_n,z_{0n},\tau_n}$ for a small η;

until *Some convergence criterion is met*;

§ IN=Inference Network, * FAD=Forward Automatic Differenciation, † AVI=Amortized Variational Inference, ‡ RAD=Reverse Automatic Differentiation (i.e. backpropagation).

4.3 Results

We now present four simulated examples of the (AC-)HAFVF in various contexts. The first example compares the performance of the HAFVF to the HGF (REF) in a simple contingency change scenario. The second example provides various case scenarios in a changing environment, illustrating the trade-off between flexibility and the precision of the predictions (Section 4.3.2), including cases where agents fail to adapt to contingency changes following prolonged training in a stable environment, as commonly observed in behavioural experiments [225]. The third example shows that the HAFVF can be efficiently fitted to a RL dataset using the method described in Section 4.2.6. The

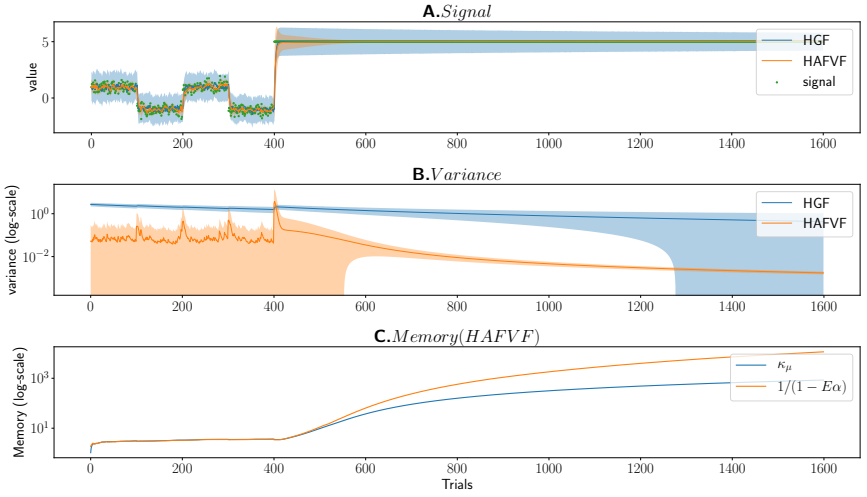


FIGURE 4.3: HAFVF and HGF performance on the same dataset. Shaded areas represent the ±3 standard error interval. **A.** Observations, mean and standard error of the mean estimated by both models. **B.** The variance estimates show that the HAFVF adapted better to the variance in the first part of the experiment, reflected better the surprise at the contingency change and adapted successfully its estimate when the environment was highly stable. The HGF, on the contrary, rapidly degenerated its estimate of the variance, and did not show a significant trace of surprise when the contingency was altered. **C.** The value of the effective memory of the HAVFV is represented by the approximate posterior parameter κ_μ, and the maximum memory (efficient memory, see Section 4.2.1) allowed by the model at each trial.

fourth and final example shows how this model behaves in multi-stage environments, and compares various implementations.

4.3.1 Adaptation to contingency changes and comparison with the HGF

In order to compare the performance of our model to the HGF, we generated a simple dataset consisting of a noisy square-wave signal of two periods of 200 trials, alternating between two normally distributed random variables ($\mathbb{N}(1, 0.33)$ and $\mathbb{N}(-1, 0.33)$). We fitted both a Gaussian HGF and the HAFVF to this simple dataset by finding the MAP parameter configuration for both models. The default configuration of the HGF was used, whereas in our case we put a normal prior of $\mathbb{N}(0, 1)$ on the parameters (with inverse softplus transform for parameters needing positive domains).

This fit constituted the first part of our experiment, which is displayed on the left part of Figure 4.3. Comparing the quadratic approximation of the Maximum Log-model evidence [73] of both models, we found a value of -186.42 for HAVFV and -204.73 for the HGF, making the HAVFV a better model of the data, with a Bayes Factor [273] greater than $8 * 10^7$.

The second part of the experiment consisted in adding to this 400-trial long signal a 1200-trial long signal of input situated at $y = 5$, and simulate learning with the MAP fitted parameter with the whole signal for both algorithms (Figure 4.3, right part). Our aim was to see how the two models would behave in such case. An optimal agent in such a situation should first account for the surprise associated with the sudden contingency change, and then progressively reduce its expected variance estimate to reflect the steadiness of the environment.

The HGF was unable to exhibit the expected behaviour: it hardly adapted its estimated variance to the contingency change and did not adjust it significantly afterwards. This contrasted with the HAFVF, in which we observed initially an increase in the variance estimate at the point of contingency change (reflecting a high surprise), followed by progressively decreasing variance estimate, reflecting the adaptation of the model to the newly stable environment.

Together, these results are informative of the comparative performance of the two algorithms. The Maximum Log-model Evidence was larger for the HAFVF than for the HGF by several orders of magnitude, showing that our approach modelled better the data at hand than the HGF. Moreover, the lack of generalization of the HGF to a simple, new signal shows that this model tended to overfit the data, as can be seen from the estimated variance at the time of the contingency change.

4.3.2 Learning and flexibility assessment

In the following datasets, we simulated the learning process of four hypothetical subjects differing in their prior distribution parameters $\boldsymbol{\phi}_0, \boldsymbol{\beta}_0$, whereas we kept $\boldsymbol{\theta}_0$ fixed for all of them (Table 4.1). The choice of the subject parameters was made to generate limit and opposite cases of each expected behaviour. With these simulations, we aimed at showing how the prior belief on the two levels of forgetting conditioned the adaptation of the subject in case of contingency change (CC, **Experiment 1**) or isolated expected event (**Experiment 2**).

In both experiments, these agents were confronted with a stream of univariate random variables from which they had to learn the trial-wise posterior distribution of the mean and standard deviation.

In **Experiment 1**, we simulated the learning of these agents in a steady environment followed by an abrupt CC, occurring either after a long (900 trials) or a short (100 trials) training. The signal $\mathbf{r} = \{ r_i \}_{i=1}^{n}$ was generated according to a Gaussian noise with mean $\mu = 3$ before the CC and $\mu = -3$ after the CC, and a constant standard deviation $\sigma = 1$.

	Subjects	μ_0	κ_0	α_0	β_0	ϕ_{1_0}	ϕ_{2_0}	β_{1_0}	β_{2_0}
(1)	LL	0	0.1	1	1	4.5	0.5	4.5	0.5
(2)	SL					0.5	4.5	4.5	0.5
(3)	LS					4.5	0.5	0.5	4.5
(4)	SS					0.5	4.5	0.5	4.5

TABLE 4.1: This table summarizes the parameters of the beta prior of the two forgetting factors w and b used in Section 4.3.2, as well as the initial prior over the mean and variance. A low value of initial number of observations κ_0 was used, in order to instruct learner to have a large prior variance over the value of the mean. Each subject will be referred by its expected memory at the lower and higher level (i.e. L = long, S = short memory). For instance, the subject number 3 (LS) is expected to have a long first-level memory, but a short second-level memory, which should make her more flexible than subject 2 (SL) after a long training, whom has a short first-level memory but a long second-level memory.

Figure 4.4 summarizes the results of this first simulation. During the training phase, the subjects with a long memory on the first level learned the observation value faster than others. Conversely, the SS subject took a long time to learn the current distribution. More interesting is the behaviour of the four subjects after the CC. In order to see which strategy reflected best the data at hand, we computed the average of the ELBOs for each model. The winning agent was the Long-Short memory, irrespective of training duration, because it was better able to adapt its memory to the contingency.

The two levels had a different impact on the flexibility of the subjects: the first level indicated how much a subject should trust his past experience when confronted with a new event, and the second level measured the stability of the first level. On the one hand, subjects with a low prior on first-level memory were too cautious about the stability of the environment (i.e. expected volatile environments) and failed to learn adequately the contingency at hand. On the other hand, after a long training, subjects with a high prior on second-level memory tended to over-trust environment stability, compared to subjects with a low prior on second level memory, impairing their adaptation after the CC.

The expected forgetting factors also shed light on the underlying learning process occurring in the four subjects: $\widehat{w}$ grew or was steady until the CC for the four subjects, even for the SL subject which showed a rapid growth of $\widehat{w}$ during the first trials, but failed to reduce it at the CC. In contrast, the LS subject did not exhibit this weakness, but rapidly reduced its expectation over the stability of the environment after the CC thanks to her pessimistic prior belief over b.

In **Experiment 2**, we simulated the effect of an isolated, unexpected event ($r_j = -3$) after long and short training with the same distribution as before. For both datasets, we focused our analysis on the value of the expected forgetting factors $\widehat{w}$ and $\widehat{b}$, as well as the effective memory of the agents, represented by the parameter κ^{μ}. As noted earlier,

	Experiment 1				Experiment 2			
Dataset	LL	SL	LS	SS	LL	SL	LS	SS
1	-1.719	-1.657	-1.62	-1.863	-1.459	-1.652	-1.501	-1.863
2	-1.526	-1.649	-1.519	-1.866	-1.453	-1.648	-1.495	-1.866

TABLE 4.2: Average ELBOs for Experiment 1 and 2. Higher ELBOs stand for more probable models.

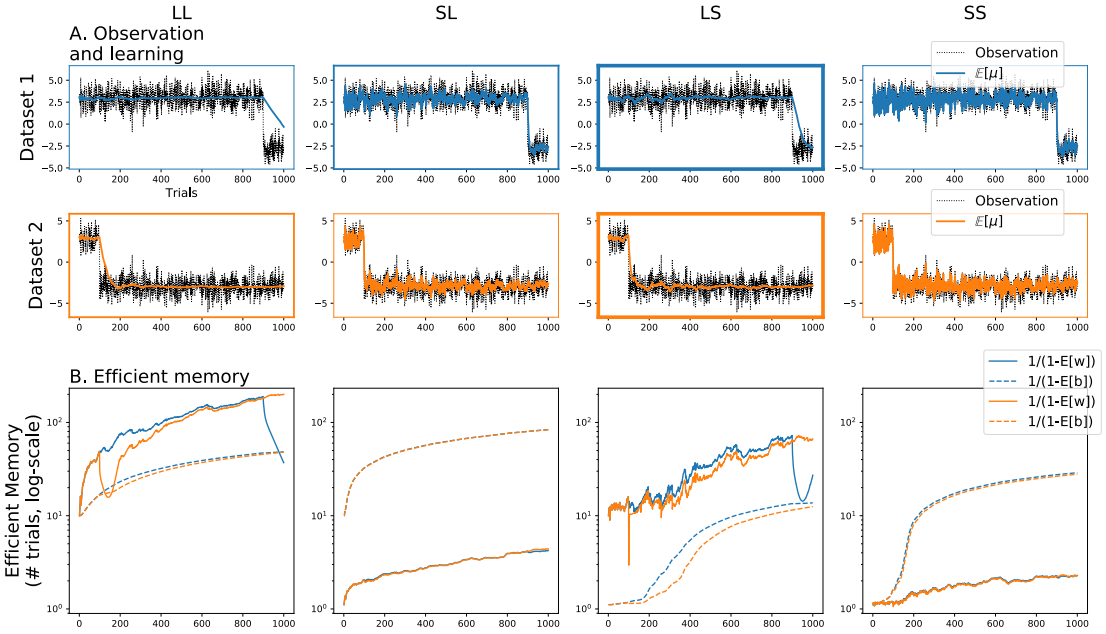


FIGURE 4.4: HAFVF predictions after a CC. Each column displays the results of a specific hyperparameter setting. The blue traces and subplots represent the learning in an experiment with a long training, the orange traces and subplots show learning during a short training experiment. **A.** The stream of observations in the two training cases are shown together with the average posterior expected value of the mean μ_j^{μ}. The box line width illustrates the ranking of the ELBO of each specific configuration for the dataset considered, with bolder borders corresponding to larger ELBOs. For both training conditions, the winning model (i.e. the model that best reflected the data) was the Long-Short memory model. This can be explained by the fact that the first two models trusted too much their initial knowledge after the CC, whereas the Short-Short learner was too cautious. **B.** Efficient memory (defined as $1/(1-\mathbb{E}_q[\cdot])$) for the first level ($\widehat{w}$, plain line) and second level ($\widehat{b}$, dashed line).

the value of $\widehat{w}$ sets an upper bound (the efficient memory) on κ^{μ}, which represented the actual number of trials kept in memory up to the current trial.

Figure 4.5 illustrates the results of this experiment. Here, the flexible agents (with a low memory on either the first or second memory level, or both) were disadvantaged wrt the low flexibility agent (mostly LL). Indeed, following the occurrence of a highly unexpected observation, one can observe that the LS learner memory dropped after either long or short training. The LL learner, instead, was able to cushion the effect of this outlier, especially after a long training, making it the best learner of the four to learn these datasets (Table 4.2).

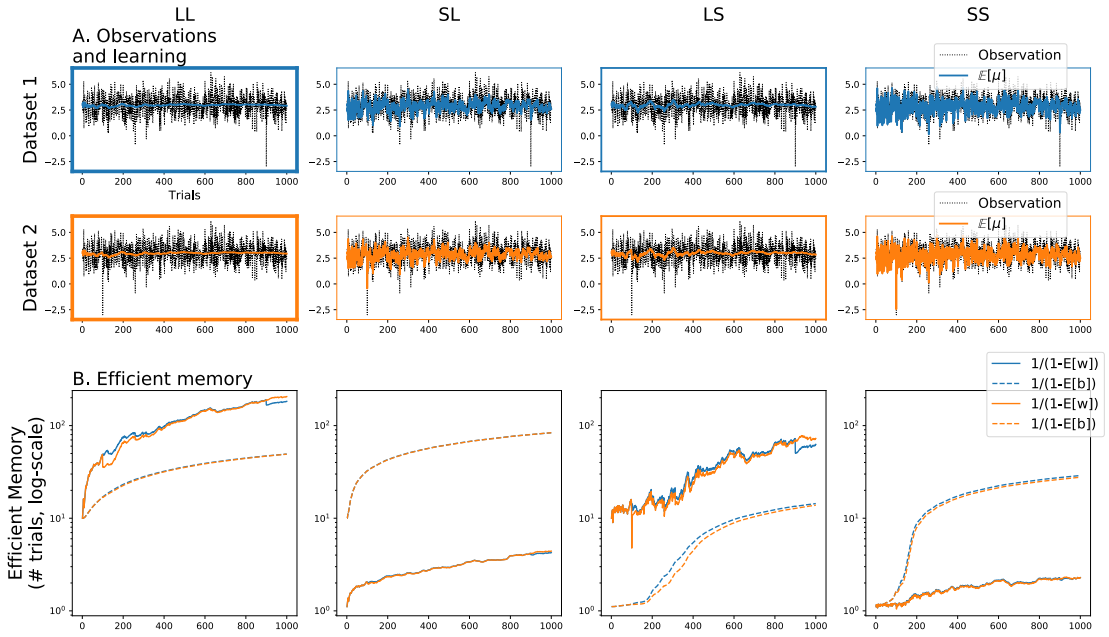


FIGURE 4.5: HAFVF predictions after an isolated unexpected event. The figure is similar to Figure 4.4. Here, the winning model was the one with a high memory on the first and second levels.

4.3.3 Fitting the HAFVF to a behavioural task

The model we propose has a large number of parameters, and overfitting could be an issue. To show that inference about the latent variables of the model depicted in Section 4.2.6 is possible, we simulated a dataset of 64 subjects performing a simple one-stage behavioural task, that consisted in trying to choose at each trial the action leading to the maximum reward. In any given trial $j \in 1:1000$, the two possible actions (e.g. left or right button press) were associated to different, normally distributed, reward probabilities with a varying mean and a fixed standard deviation $\sigma^2_{1,2} = 1$: for the first action (a_1) the reward had a mean of 0 for the first 500 trials, then switched abruptly to $+2$ for 100 trials and then to -2 for the rest of the experiment. The second action value was identically distributed but in the opposite order and with the opposite sign (Figure 4.6). This pattern was chosen in order to test the flexibility of each agent after an abrupt CC: after the first CC, an agent discarding completely exploration in favour of exploitation would miss the CC. The second and third CC tested how fast did the simulated agents adapt to the observed CC.

Individual prior parameters $\boldsymbol{\Omega_j} \triangleq \{\mathcal{NG}^{-1}$ prior $\boldsymbol{\theta}_{0,n}$, Beta priors $\boldsymbol{\phi}_{0,n}, \boldsymbol{\beta}_{0,n}\}$ and thresholds ζ_n were generated as follows [3]: we first looked for the L2-regularized MAP estimates

[3] With a negligible loss of generality, the prior mean μ_0 was considered to be equal to 0 for all subjects for both the data generation and the fitting procedures.

of these parameters that led to the maximum total reward:

$$\Omega^*, \zeta^* = \arg\max_{\Omega,\zeta} \prod_{j=1}^{J} p(a_j = \arg\max_{\mathbf{a}_j} r_j | r_{<j}, \mathbf{a}_{<j}, \mathbf{\Omega}, \zeta) p(\mathbf{\Omega}, \zeta)$$

using a Stochastic Gradient Variational Bayes (SGVB) [88] optimization scheme.

We then simulated individual priors centered around this value with a covariance matrix arbitrarily sampled as

$$\Sigma_{\Omega,\zeta} \sim \mathcal{W}_{10}^{-1}\left(\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0.1 \end{bmatrix}\right)$$

where $\mathcal{W}_n^{-1}(\cdot)$ is an inverse-Wishart distribution with n degrees of freedom. This choice of prior lead to a large variability in the values of the AC-HAFVF parameters, except for the NIGDM threshold whose variance was set to a sufficiently low value (hence the 0.1 value in the prior scale matrix) to keep the learning performance high.

This method ensured that each and every parameter set was centered around an unknown optimal policy. This approach was motivated by the desire of preventing any strong constrain on the data generation pattern while keeping a behaviour that was, on average, close to be optimal, as it might be expected with a healthy population. The other DDM parameters, ν_n and τ_n, were generated according to a Gaussian distribution centered on 0 and 0.3, respectively. Importantly, simulated subjects with a performance lower than 70% were rejected and re-sampled to avoid irrelevant parameter patterns.

Learning was simulated according to the Continuous Learning strategy (see Section 4.2.4.2), because it was supposed to link more comprehensively the tendency to explore the environment with the choice of prior parameters $\mathbf{\Omega}$. Choices and RT where then generated according to the decision process described in Section 4.2.5 using the algorithm described by [8]. Figure 4.6 shows two examples of the simulated behavioural data.

The behavioural results showed a clear tendency of subjects with large expected variance in action selection to act faster and less precisely than others. This follows directly from the structure of the NIGDM: larger variance of the drift-rate leads to faster but less precise policies. More interesting is the negative correlation between the expected stability and the reward-rate and average reaction time: this shows that the AC-HAFVF was able to encode a form of subject-wise computational complexity of the task. Indeed, large stability expectation leads subjects to trust more their past experience, thereby decreasing the expected reward variance after a long training, but it also leads to a lower

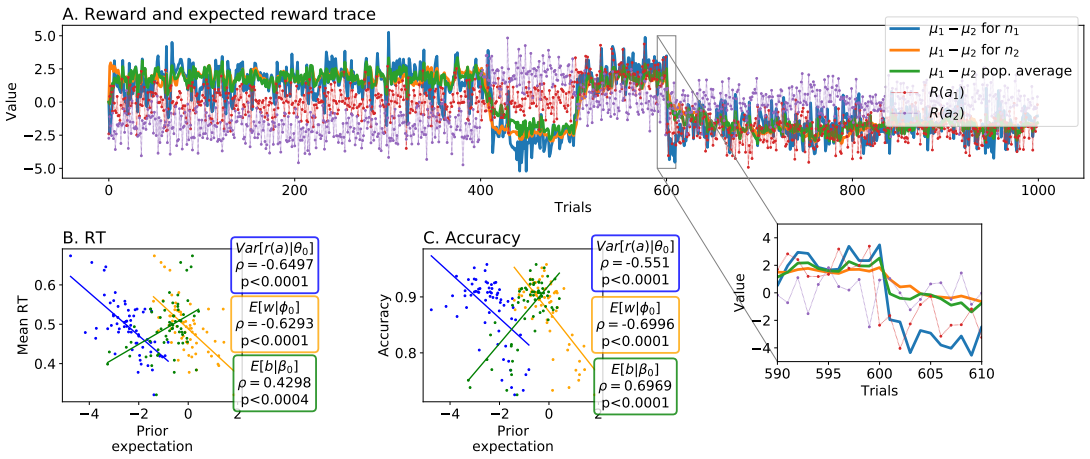


FIGURE 4.6: Simulated behavioral results. **A.** The values of the two available rewards are shown with the dotted lines. The average drift rate $\mu_1 - \mu_2$ is shown in plain lines for two selected simulated subjects n_1 and n_2, and population average. Subject n_1 was more flexible than subject n_2 on both the first and the second level, making her more prone to adapt after the CCs, situated at trials 400, 500 and 600. This result is highlighted in the underlying zoomed box. **B.** The subjects' expected variance (blue, log-valued) correlated negatively with the mean RT. The same correlation existed with the expected stability on the first level (orange, logit-valued), but not with the second level, which correlated positively with the average RT (green, logit-valued). Pearson correlation coefficient and respective p-values are shown in rounded boxes. **C.** Similarly, subjects with a higher expected variance and first-level stability had a lower average accuracy. Again, second-level memory expectation had the opposite effect.

capacity to adapt to CCs. For subjects with low expectation of stability, the second level memory was able to instruct the first-level to trust past experience when needed, as the positive correlation between accuracy and upper level memory shows.

4.3.3.1 Fitting results

The fit was achieved with the Adam [493] Stochastic Gradient Descent optimizer with parameters $s = 0.005, \beta_1 = 0.9, \beta_2 = 0.99$, where s decreased with a rate $\sqrt{\lceil i/1000 \rceil}$ where i is the iteration number of the SGD optimizer. We used the following annealing procedure to avoid local minima: at each iteration, the whole set of parameters was sampled from a diagonal Gaussian distribution with covariance matrix $1/i$. This simple manipulation greatly improved the convergence of the algorithm.

The MAP fit of the model is displayed in Figure 4.7. In general, the posterior estimates of the prior parameters were well correlated with their true value, except for the prior shape parameter α_0. This lack of correlation did not, however, harm much the fit of the variance (see below), showing that the model fit was able to accurately recover the expected prior variability of each subject, which depended on α_0. The NIGDM parameters were highly correlated with their original value.

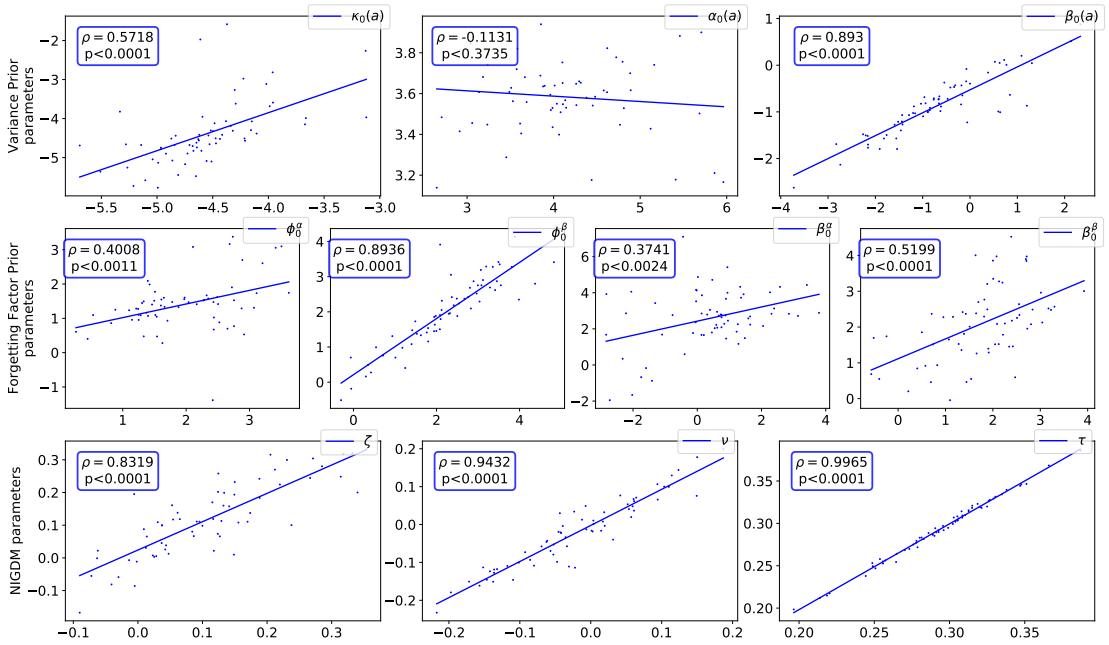


FIGURE 4.7: Correlation between the true (x axis) and the posterior estimate (y axis) of the parameters of the prior distributions across subjects. The first row displays the correlations between true value and estimated $\boldsymbol{\theta}_0$. The second row focuses on $\boldsymbol{\phi}_0$ and $\boldsymbol{\beta}_0$, whereas the third row shows the correlations for the NIGDM parameters (threshold, relative start-point and non-decision time). Correlation coefficients and associated p-value (with respect to the posterior expected value) are displayed in blue boxes. All parameters are displayed in the unbounded space they were generated from. Overall, all parameters correlated well with their true value, except for the $\alpha_0(a)$.

We also looked at how the true prior expected value of w and b and variance correlated with their estimate from the posterior distribution. All of these correlated well with their generative correspondent, with the notable exception of the expected value of the second-level memory $\mathbb{E}_q[b]$ (Figure 4.8). This poorer fit of $\mathbb{E}_q[b]$ was, however, compensated by the good correlations of the generative parameters $\boldsymbol{\beta}_0$.

4.3.4 TD learning with the HAFVF

To study the TD learning described in Section 4.2.4.3, we built two similar Markov Decision Processes (MDP) which are described in Figure 4.9a and Figure 4.9b. In brief, they consist of 5 different states where the agent has to choose the best action in order to reach a reward ($r = 5$) delivered once a specific state has been reached (hereafter, we will use "rewarded state" and "rewarded state-action" interchangeably). The task consisted for the agent to learn the current best policy during a 1000-trial experiment where the contingency was fixed to the first MDP during the first 500 trials, and then switched abruptly to the second MDP during the second half of the experiment.

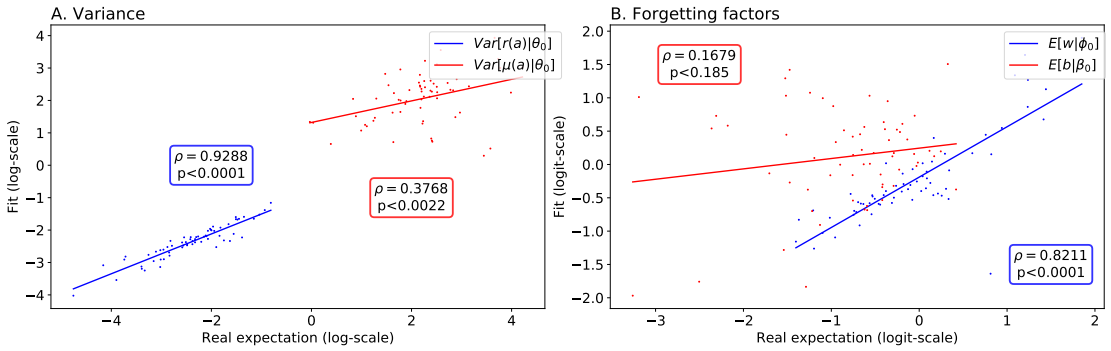


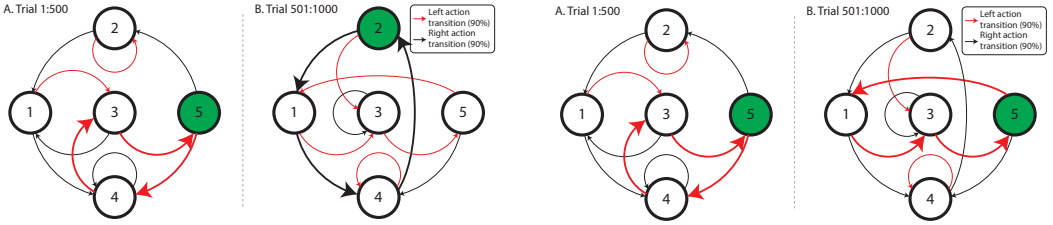
FIGURE 4.8: Correlation between true and expected values of the variances and forgetting factors. All the fitted values (y axis) are derived from the expected value of $\boldsymbol{\theta}_0$ (**A.**, variance) and $\{\boldsymbol{\phi}_0, \boldsymbol{\beta}_0\}$ (**B.**, forgetting factors) under the fitted approximate posterior distribution. Each dot represents a different subject. **A.** True (x-axis) to fit (y axis) correlation for the reward (blue) and mean reward (red) variance. Both expected values correlated well with their generative parameter, although the initial number of observations of the gamma prior α_0 did not correlate well with its generative parameter. **B.** True (x-axis) to fit (y axis) correlation for the first (blue) and second (red) level expected forgetting factor.

The same prior was used for all subjects: the mean and variance prior was set to $\mu_0 = 0$, $\kappa_0 = 0.5$, $\alpha_0 = 3$, $\beta_0 = 0.5$. The forgetting factors shared the same flat prior $\alpha_0^{w,b} = 1, \beta_0^{w,b} = 1$. The priors on the discounting factor were set to a high value $\alpha_0^{\gamma} = 9, \beta_0^{w,b} = 1$ in order to discourage myopic strategies. The policy prior (see Section 4.5.4) was set to a high value ($\pi_0 = 5.$) in order to limit the impact of initial choices on the computation of the state-action value.

Results are displayed in Figure 4.10. In both experiments, agents had a similar behaviour during the first phase of the experiment: they both learned well the first contingency by assigning an accurate value to each state-action pair in order to reach the rewarded state more often. As expected, after the contingency change, the agents in experiment (i) took a longer time to adapt than they took to learn the initial contingency, which can be seen from the steeper slope of the reward rate during the first half of the experiment wrt the second. This feature can only be observed if the agent weights its belief by some measure of certainty, which is not modelled in classical, non-Bayesian RL.

In experiment (ii), the CC changes less the environment structure than the CC in experiment (i). The agents were able to take this difference into account: the value of the effective memories dropped less, and so did the reward rate.

An important feature observed in these two experiments is that the expected value of γ adapted efficiently to the contingency: although we used a prior skewed towards high values of γ, its value tended to be initially low as the agents had no knowledge of the various action values. Also, this value increased afterwards, reflecting a gain in predictive accuracy. The drop of $\mathbb{E}_q[w]$ had the effect of pushing $\mathbb{E}_q[\gamma]$ towards its prior, which



(A) First experiment MDPs. (B) Second experiment MDPs.

FIGURE 4.9: Rewarded states are displayed in green. Each action had a probability of 90% to lead to the end state indicated by the red and black arrows (respectively left and right action). The remaining 10% transition probabilities were evenly distributed among the other states. For clarity, the thick arrows show the optimal path the agent should aim to take during the two contingencies. Note that the only difference between experiment (i) and (ii) is the location of the rewarded state after the CC (state 2 for (i) and 5 for (ii)).

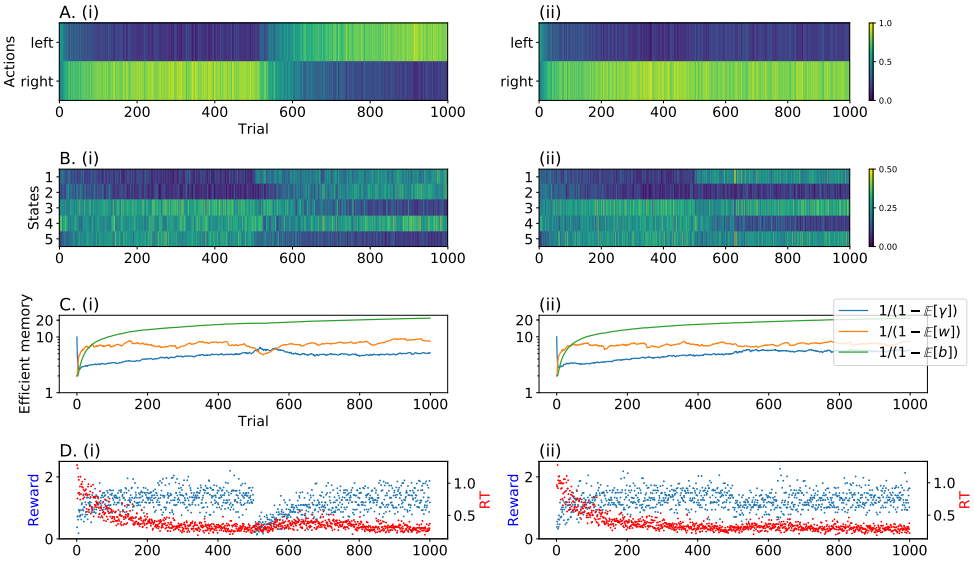


FIGURE 4.10: Behavioural results in the first (left, Figure 4.9a) and second experiments (right, Figure 4.9b). **A.** and **B.** Heat plot of the probability of visiting each state and selecting each action for the 64 agents simulated. (i) Agents progressively learned the first optimal actions (left action in state 3-4-5) during the first half of the experiment, then adapted their behaviour to the new contingency (right action in states 1-4-2). (ii) Similarly, in the second experiment, agents adapted their behaviour according to the new contingency (left action in 1-3-5). **C.** Efficient memory on the first and second level, and foreseeing capacity. Since the CC was less important in (ii) than in (i), because the left action in state 3 kept being rewarded, the expected value of w dropped less. The behaviour of the foreseeing capacity ($1/(1 - \mathbb{E}_q[\gamma])$ and, therefore, of the expected value of γ, is indicative of the effect that a CC had on this parameter: when the environment became less stable, $\mathbb{E}_q[\gamma]$ tended to *increase* which had the effect of increasing the impact of future states on the current value. **D.** (i) Reward rate dropped after the CC, whereas the RT increased. The fact that subjects made slower choices after the CC can be viewed as a mark of the increased task complexity caused by the re-learning phase. Along the same line, RT decreased again when the subjects were confident about the structure of the environment. (ii) The CC had also a lower impact on the reward rate and RT in experiment (ii).

in this case *increased* the posterior expectation of γ. At the same time, the uncertainty about γ increased, thereby enhancing the flexibility of this parameter.

4.4 Discussion

In this paper, we propose a new Bayesian Reinforcement Learning (RL) algorithm aimed at accounting for the adaptive flexibility of learning observed in animal and human subjects. This algorithm adapts continuously its learning rate to inferred environmental variability, and this adaptive learning rate is optimal under some assumptions about statistical properties of the environment. These assumptions take the form of prior distributions on the parameters of the latent and mixing weight variables. We illustrate different types of behaviour of the model when facing unexpected contingency changes by taking extreme case scenarios. These scenarios implemented four types of assumptions on the tendency of the environment to vary over time (first-level memory) and on the propensity of this environmental variability to change itself over time (second-level memory). This approach allowed us to reproduce the emergence of inflexible behaviour following prolonged experience of a stable environment, similar to empirical observations in animals. Indeed, it has long been known that extensive training leads to automatization of behaviour, called habits [588, 249, 545, 589], or procedural "system 1" actor ([590, 249]), which is characterized by a lack of flexibility (i.e. failure to adapt to contingency changes) and by reduction of computational costs, illustrated by the capacity to perform these behaviours concomitantly to other tasks [279, 255, 253]. These automatic types of behaviour are opposed to Goal-Directed behaviours ([255, 236, 253]) and share the common feature of being inflexible, either in terms of planning (for Model-Free RL for instance) or in terms of adaptation in general.

Regarding the actor part, we implemented a general, Bayesian decision-making algorithm that reflects in many ways the Full-DDM proposed by [364], as it samples the reward distribution associated to each action and selects at each time step the best option. These elementary decisions are integrated until a decision boundary is reached. Therefore, the actor maps the cognitive predictions of the critic onto specific behavioural outputs such as choice and reaction time (RT). Importantly, this kind of Bayesian evidence accumulation process for decision making is biologically plausible [386, 387], well suited for decision making in RL [591, 434, 592, 593] and makes predictions that are in accordance with physiological models of learning and decision making [5]. Other noteworthy attempts have been made to integrate sampling-based decision-making and RL [11, 12] using the DDM. However, the present work is the first, to our knowledge, to frame the DDM as an optimal, Bayesian decision strategy to maximize long-term

utility on the basis on value distributions inferred from a RL algorithm. We show that this RL-DDM association, and especially in the framework of the Full DDM [364], finds a grounded algorithmic justification in a Bayesian perspective, as the resulting policy mimics the one of an agent trying to infer the best decision given its posterior belief about the reward distribution.

Interestingly, under some slight modifications (i.e. assuming that the sampled rewards are simulated from inverse causal mappings of samples drawn from the subject memory [594, 109, 378]), it can recall similar to the model recently proposed by Bornstein et al. [298, 420]. Similarly, while our decision making scheme used a heuristic based on the asymptotic property of MCMC, the scheme of decision making proposed by Bornstein and colleagues might recall other approximation techniques such as Approximate Bayesian Computation (ABC [595]). Following this approach, data samples are generated according to some defined rule, and only the samples that match the actual previous observations are kept in memory to approximate the posterior distribution or, in our case, to evaluate the option with the greatest reward. This simple trick in the decision making process keeps the stochastic nature of the accumulation process, while directly linking the level of evidence to items retrieved from memory. We will discuss the advantages of this decision scheme in Section 5.3.

One major advantage of our model is that its parameters are easily interpretable as reflecting hidden behavioural features such as trial-wise effective memory, prior and posterior expected stability, etc. We detail how model parameters can be fitted to data in order to recover these behavioural features at the trial, subject and population levels. This approach could be used, for instance, to cluster subjects in high-stability seeking and low-stability seeking sub-populations, and correlate these behaviours to health conditions, neurophysiological measures or training condition (stress, treatment etc.) We show that, for a simulated dataset, the fitted posterior distribution of the parameters correlate well with their original value. Moreover, each of the layers of the model (learning, first level memory and second level memory) have interesting behavioural correlates in terms of accuracy and RT. These results show that the model is identifiable and that there is a low redundancy in the various layers of the model. Moreover, different priors over the expected distribution of rewards and environment stability led also to different outcomes in terms of reward rate and RT: subjects whom assumed large environmental stability were likely to act faster, but also to be less flexible and to gain less on average, than subjects whom assumed high likelihood of contingency changes. This finding also resonates with the habitual learning literature, in which decreased flexibility is also typically associated with short average RT, reflecting lower cognitive cost (see also [198]). The second-level memory had different effect, in the sense that large memory tended to

be associated with flexible or inflexible behavior, depending on whether past experience corresponded to volatile or stable environment, respectively.

Finally, we extended our work to MDP and multi-stages tasks. More than adding a mere reparametrization of TD learning, we propose a framework in which the time discount factor is considered by the agent as a latent variable: this opens the possibility of studying how animals and humans adapt their long-term / short-term return balance in different conditions of environmental variability. We show in the results that deep CC provoke a reset of the parameters, and lead subjects to erase their acquired knowledge of the posterior value of γ. We also show that this ability to adapt γ to the uncertainty of the environment can also be determinant at the beginning of the task, where nothing is known about the long term return of each action, and where individuals might benefit of a low expectation of γ that contrasts with the high expectation of subjects that have a deeper knowledge of the environment.

The present model is based on forgetting, which is an important feature in RL that should be differentiated from learning. In signal processing and related fields where online learning is required, learning can be seen as the capacity to build on previous knowledge (or assess a posterior probability distribution in a Bayesian context) to perform inference at the present step. Forgetting, on the contrary, is the capacity to erase the learning to come back at a naive, initial state of learning. The algorithm we propose learns a posterior belief of the data distribution and infers how likely this belief is to be valid in the next time step, on the basis of past environmental stability. This allows the algorithm to decide when and how much to forget its past belief in order to adapt to new contingencies. This feature sets our algorithm apart from previous proposals such as HGF, in which the naive prior looses its importance as learning goes on, and where the learner has no possibility of coming back to his initial knowledge. This lack of capacity to forget implies that the agent can be easily fooled by its past experience, whereas our model is more resistant in such cases, as its point of reference is fixed (which is the common feature of SF algorithms, see above). We have shown in the results that our model outperforms the HGF both in its fitting capability and its capacity to learn new observations with a given prior configuration. The AC-HAFVF should help to flexibly model learning in these contexts, and find correlates between physiological measures, such as dopaminergic signals [596, 597], and precise model predictions in term of memory and flexibility.

The SF scheme we have used, where the previous posterior is compared with a naive prior to optimize the forgetting factor, is widely diffused in the signal processing community [518, 528, 519, 536, 531, 535, 521, 522] and finds grounded mathematical justifications for error minimization in recursive Bayesian estimation [534]. However, it is the first

time, to our knowledge, that this family of algorithms is applied to the study of RL in humans. We show that the two algorithms (RL and SF) share deep common features: for instance, the HAFVF and other similar algorithms ([520]) can be used with a naive prior $\boldsymbol{\theta}_0$ set to 0, in which case the update equations reduce somehow to a classical Q-learning algorithm (Chapter 3 and Section 4.5.2). Another interesting bound between the two fields emerges when the measure of the environment volatility is built hierarchically: an interesting consequence of the forgetting algorithm we propose is that, when observations are not made, the agent erases progressively its memory of past events. This leads to counterfactual learning schedule that favors exploration over exploitation at a rate dictated by the learned stability of the environment (see Section 4.5.3 for a devlopement). Crucially, this updating scheme, and the consecutive exploration policy, flows directly from the hierarchical implementation of the SF scheme.

This work provides a tool to investigate learning rate adaptation in behaviour. Previous work has shown, for instance that the process of learning rate adaptation can be decomposed into various components that relate to different brain areas or networks [598]. Nassar and colleagues [599] have also looked at the impact of age on learning rate adaptation, and found that older subjects were more likely to have a narrow expectation of the variance of the data they were observing, impairing thereby their ability to detect true CC. The AC-HAFVF shares many similarities with the algorithm proposed by Nassar and McGuire: it is designed to detect how likely an observation is to be caused by an abrupt CC, and adapts its learning rate accordingly. Also, this detection of CC depends in both models not only on the first and second moment of the observations, but also on their prior average and variability. We think, however, that our model is more flexible and biologically plausible than the model of Nassar and McGuire for three reasons. First, it is fully Bayesian: when fitting the model, we do not fit an expected variance of the outcome observed, but a prior distribution on this variance. This important difference is likely to predict better the observed data, as the subjects performing an experiment have probably some prior uncertainty about the variability of the outcome they will witness. Second, we considered the steadiness of the environment as another Bayesian estimate, meaning that the subjects will have some confidence (and posterior distribution) in the fact that the environment has truly changed or not. Third, we believe that our model is more general (i.e. less ad hoc) than the model of Nassar, as the general form of Equation (4.9) encompasses many models that can be formulated using distributions issued from the exponential family. This includes behavioural models designed to evolve in multi-stage RL tasks, such as TD learning or Model-Based RL [600].

It is important to emphasize that the model we propose does not intend to be universal. In simple situations, fitting a classical Q-learning algorithm could lead to similar or

even better predictions than those provided by our model. We think, however, that our model complexity makes it useful in situations where abrupt changes occur during the experiment, or where long sequences (several hundreds of trials) of data are acquired. The necessity to account for this adaptability could be determined by comparing the accuracy of the HAFVF model (i.e. the model evidence) to the one obtained from a simpler Bayesian Q-Learning algorithm without forgetting.

The richness and generality of the AC-HAFVF opens countless possibilities of future developments. The habitual/goal-directed duality has been widely framed in terms of Model-Free/Model-Based control separation (e.g. [256, 283, 601, 589, 291, 602, 603, 261, 293]). Although we do not model here a Model-Based learning algorithm, we think our work will ultimately help to discriminate various forms of inflexibility, and complete the whole picture of our understanding of human RL: the current implementation of the AC-HAFVF can model a lack of flexibility due to an overconfidence in the volatility of the environment, whereas adding a Model-Based component to the model might help to discriminate a lack of flexibility due to the overuse of a Model-Free strategy[4] that characterize the Model-Free/Model-Based paradigm. In short, implementation of Model-Based RL in an AC-HAFVF context might enrich greatly our understanding of how the balance between Model-Based and Model-Free RL works. This is certainly a development we intend to implement in the near future.

In conclusion, we provide a new Model-Free RL algorithm aimed at modelling behavioural adaptation to continuous and abrupt changes in the environment in a fully Bayesian way. We show that this model is flexible enough to reflect very different behavioural predictions in case of isolated unexpected events and prolonged change of contingencies. We also provide a biologically plausible decision making model that can be integrated elegantly in our learning algorithm, and completes elegantly the toolbox to simulate and fit datasets.

[4] Overconfidence in a Model-Free controller means that the subject will need to go through the same sequences of state-action transitions over and over to downweight actions situated early in a sequence. This contrasts with Model-Based control, which can immediately adapt its policy when an action situated far from the current state is devaluated [266].

4.5 Appendix

4.5.1 Normalizing constant of the exponential mixture prior distribution

Let us consider a prior distribution from the exponential family of the form

$$f(\mathbf{z} \mid \boldsymbol{\theta}) = h(\mathbf{z}) \exp(\boldsymbol{\theta}\, \mathbf{T}(\mathbf{z}) - A(\boldsymbol{\theta})). \tag{4.19}$$

Assuming that the approximate posterior $q(\mathbf{z} \mid \boldsymbol{\theta}_{j-1})$ has the same form as the prior distribution (which arises naturally in the VB scheme we are using), we are interested in deriving

$$\begin{aligned}
Z(w, \boldsymbol{\theta}_{j-1}, \boldsymbol{\theta}_0) &= \int f(\mathbf{z} \mid \boldsymbol{\theta}_{j-1})^w f(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w} d\,\mathbf{z} \\
&= \int \exp\left(w\Big(\boldsymbol{\theta}_{j-1}\, \mathbf{T}(\mathbf{z}) + \log h(\mathbf{z}) - A(\boldsymbol{\theta}_{j-1}) \Big) + (1-w)\Big(\boldsymbol{\theta}_0\, \mathbf{T}(\mathbf{z}) + \log h(\mathbf{z}) - A(\boldsymbol{\theta}_0) \Big) \right) dz \\
&= \int h(\mathbf{z}) \exp\left(w\Big(\boldsymbol{\theta}_{j-1}\, \mathbf{T}(\mathbf{z}) - A(\boldsymbol{\theta}_{j-1}) \Big) + (1-w)\Big(\boldsymbol{\theta}_0\, \mathbf{T}(\mathbf{z}) - A(\boldsymbol{\theta}_0) \Big) \right) \\
&\times \exp\left(A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right) - A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right) \right) d\,\mathbf{z} \\
&= \underbrace{\int h(\mathbf{z}) \exp\left(\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right) \mathbf{T}(\mathbf{z}) - A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right) \right) d\,\mathbf{z}}_{=1} \\
&\times \exp(-wA(\boldsymbol{\theta}_{j-1}) - (1-w)A(\boldsymbol{\theta}_0) + A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right)) \\
&= \exp(-wA(\boldsymbol{\theta}_{j-1}) - (1-w)A(\boldsymbol{\theta}_0) + A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right)).
\end{aligned} \tag{4.20}$$

This result simplifies when combined with the numerator of Equation (4.4):

$$\begin{aligned}
p(\mathbf{z} \mid \boldsymbol{\theta}_{j-1}, \boldsymbol{\theta}_0, w) &= \frac{h(\mathbf{z}) \exp\left(\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right) \mathbf{T}(\mathbf{z}) - wA(\boldsymbol{\theta}_{j-1}) - (1-w)A(\boldsymbol{\theta}_0) \right)}{\exp(-wA(\boldsymbol{\theta}_{j-1}) - (1-w)A(\boldsymbol{\theta}_0) + A(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0))} \\
&= h(\mathbf{z}) \exp\left(\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right) \mathbf{T}(\mathbf{z}) - A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right) \right)
\end{aligned} \tag{4.21}$$

which has the form of a distribution from the exponential family where the natural parameters of the two parts of the mixture are weighted according to w and $1-w$.

If we are to find an analytical form to the lower bound $\mathcal{L}(q(\mathbf{z}, w)) = \mathbb{E}_{q(\mathbf{z},w)}\left[\log p(\mathbf{x}, \mathbf{z}, w) - \log q(\mathbf{z}, w)\right]$, we also need to compute the expected log-partition function given the approximate posterior $q(\mathbf{z}, w)$: $-\mathbb{E}_{q(\mathbf{z},w)}\left[A\left(w\,\boldsymbol{\theta}_{j-1} + (1-w)\,\boldsymbol{\theta}_0\right)\right]$. This expectation has usually not a

closed-form expression, but it can be approximated efficiently if we consider its second order Taylor expansion. Indeed,

$$\begin{aligned} A\left(w\,\boldsymbol{\theta}_{j-1}+(1-w)\,\boldsymbol{\theta}_0\right) &= A\left(w_0\,\boldsymbol{\theta}_{j-1}+(1-w_0)\,\boldsymbol{\theta}_0\right) \\ &+ \nabla_{w_0} A\left(w_0\,\boldsymbol{\theta}_{j-1}+(1-w_0)\,\boldsymbol{\theta}_0\right)(w-w_0) \\ &+ \frac{\nabla^2_{w_0} A\left(w_0\,\boldsymbol{\theta}_{j-1}+(1-w_0)\,\boldsymbol{\theta}_0\right)}{2}(w-w_0)^2 + \mathcal{O}\big(|w_0-w|^3\big). \end{aligned} \tag{4.22}$$

If we replace w_0 by the expected value of w under the approximate posterior $w_0 \triangleq \mathbb{E}_{q(w)}[w] = \widehat{w}$, and given that $\mathbb{E}_{q(w)}[(w-\widehat{w})] = 0$, the expected value of the log-partition function can be approximated with

$$\begin{aligned} \mathbb{E}_{q(w)}\left[A\left(w\,\boldsymbol{\theta}_{j-1}+(1-w)\,\boldsymbol{\theta}_0\right)\right] &\approx A\left(\widehat{w}\,\boldsymbol{\theta}_{j-1}+(1-\widehat{w})\,\boldsymbol{\theta}_0\right) \\ &+ \frac{1}{2}\nabla^2_{\widehat{w}} A\left(\widehat{w}\,\boldsymbol{\theta}_{j-1}+(1-\widehat{w})\,\boldsymbol{\theta}_0\right)\mathbb{V}ar_{q(w)}[w] \\ \text{with } \widehat{w} &= \mathbb{E}_{q(w)}[w]\,. \end{aligned} \tag{4.23}$$

4.5.2 HAFVF update equations correspondence in classical RL

The update equations of Equation (4.12) have a simple correspondence in classical frequentist Q-learning. Let us first consider the case of the mean update in Bayesian Q-Learning without adaptive forgetting where we have

$$\begin{aligned} \mu^\mu_j &= \frac{\kappa^\mu_{j-1}\mu^\mu_{j-1} + \mathbf{x}_j}{\kappa^\mu_{j-1}+1} \\ &= (1-\eta)\mu^\mu_{j-1} + \eta x_j \\ &\quad \text{where } \eta = \frac{1}{\kappa^\mu_{j-1}+1} \\ &= \mu^\mu_{j-1} + \eta\underbrace{(x_j - \mu^\mu_{j-1})}_{\text{RPE}}. \end{aligned} \tag{4.24}$$

We can see that the Reward Prediction Error (RPE) of the Rescola-Wagner (RW) update emerges naturally in one considers that the learning rate η is equal to the inverse of the sum of the number of trials observed plus the prior observation belief κ^μ_0 (meaning that the learning rate decreases each time an observation is made by a factor $\frac{\kappa^\mu_0+n}{\kappa^\mu_0+n+1}$).

We can now look at the update of the expected value of the mean reward. It can be reformulated as

$$\begin{aligned}\mu_j^\mu &= \frac{\widehat{\mathrm{w}}\ \kappa_{j-1}^\mu \mu_{j-1}^\mu + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^\mu \mu_0^\mu + x_j}{\kappa_j^\mu} \\ &= \frac{\widehat{\mathrm{w}}\ \kappa_{j-1}^\mu}{\kappa_j^\mu}\mu_{j-1}^\mu + \frac{(1-\widehat{\mathrm{w}})\kappa_0^\mu}{\kappa_j^\mu}\mu_0^\mu + \frac{x_j}{\kappa_j^\mu}.\end{aligned}$$

Without loss of generality, we can consider that the prior mean reward is equal to 0, which simplifies the above equation:

$$\mu_j^\mu = \frac{\widehat{\mathrm{w}}\ \kappa_{j-1}^\mu}{\kappa_j^\mu}\mu_{j-1}^\mu + \frac{x_j}{\kappa_j^\mu}.$$

One can easily see that, in this context, we can recover the classical RW update term:

$$\begin{aligned}\mu_j^\mu &= (1-\eta)\mu_{j-1}^\mu + \frac{\eta}{(1-\widehat{\mathrm{w}})\kappa_{j-1}^\mu + 1}x_j \\ &= \mu_{j-1}^\mu + \eta \underbrace{\left(\frac{x_j}{(1-\widehat{\mathrm{w}})\kappa_{j-1}^\mu + 1} - \mu_{j-1}^\mu\right)}_{\mathrm{RPE}} \\ \text{where } \eta &= \frac{(1-\widehat{\mathrm{w}})\kappa_{j-1}^\mu + 1}{\kappa_j^\mu}.\end{aligned}$$

The current observation x_j is decayed by a factor proportional to the product of the previous effective memory κ_{j-1} and the trust allowed to the prior $1-\widehat{\mathrm{w}}$. This means that the agent learns less trials that need to be discarded (as they are more likely to be generated by the prior distribution than the previous posterior), especially when the agent has a high memory. This account for the fact that an accident is less likely after a long than a short steady training.

4.5.3 Counterfactual learning update equations

4.5.3.1 Continuous Updating

We consider that the agent adopts some update policy based on a family of inferred update distributions $\widetilde{x}_{j+1} \sim f\left(\boldsymbol{\theta}_j(s,a), \boldsymbol{\theta}_0(s,a), \boldsymbol{\phi}_j, \boldsymbol{\phi}_0, \boldsymbol{\beta}_0\right)$. As our notation suggests, this update distribution is either a function of the previous approximate posterior $\boldsymbol{\theta}_j$, of the prior $\boldsymbol{\theta}_0$ or both.

If $\widetilde{x}_{j+1} \sim f(\boldsymbol{\theta}_j(s,a))$, we are in the presence of an optimistic limit case where the agent does not consider that the environment will change anymore when she has selected an

action: once some $r_\tau(s,a)$ with $\tau \in 1:j$ is observed, the resulting posterior predictive distribution is used to update the variational parameters in the next trials until a is selected again.

Conversely, the pessimistic model of evolution of $\widetilde{x}_{j+1}$ considers that the upcoming values of $r(s,a)$ are extremely variable, and that the agent should not trust anything more than its broad, initial marginal prior $p(\widetilde{x}_{j+1}|\,\boldsymbol{\theta}_0)$.

A third case of update distribution can be used: the agent can actually consider that the new value of $\widetilde{x}_{j+1}$ would likely be drawn from a mixture of the prior and posterior approximate marginal distributions, similar to the distribution implemented in Equation (4.5) (and possibly Equation (4.9)):

$$\begin{aligned} \widetilde{x}_{j+1} &\sim p(\widetilde{x}_{j+1}|\,\mathbf{z}) \\ \mathbf{z} &\sim p(\mathbf{z}\,|w(\boldsymbol{\theta}_j(s,a) - \boldsymbol{\theta}_0(s,a)) + \boldsymbol{\theta}_0). \end{aligned} \tag{4.25}$$

The next section details the update equation in these three cases.

4.5.3.2 Approximate posterior updating of non-selected actions

In order to develop the update equations of the variational parameters in the case of counterfactual learning, one must first derive an ELBO for this specific case. Recall that, if x_{j+1} were observed, the ELBO would have the form:

$$\mathcal{L}_{j+1}(q(\mu,\sigma,w,b)) = \mathbb{E}_{(\mu,\sigma,w,b)}\left[\log p(x_{j+1},\mu,\sigma,w,b|\,\boldsymbol{\theta}_{j+1},\boldsymbol{\theta}_0) - \log q(\mu,\sigma,w,b)\right].$$

In the case of an unobserved datapoint, we take the expected value of the ELBO under some model of the evolution of $x_{j+1} \sim f(\boldsymbol{\theta}_j,\boldsymbol{\theta}_0)$

$$\mathbb{E}_f\left[\mathcal{L}_{j+1}(q(\mu,\sigma,w,b))\right] = \mathbb{E}_f\left[\mathbb{E}_{(\mu,\sigma,w,b)}\left[\log p(x_{j+1},\mu,\sigma,w,b|\,\boldsymbol{\theta}_{j+1},\boldsymbol{\theta}_0) - \log q(\mu,\sigma,w,b)\right]\right].$$

One can easily see that the update of the variational parameters of the action not taken under the optimistic assumption that the environment will not change takes the form

of:

$$\mu_{j+1}^{\mu} = \frac{\widehat{\mathrm{w}}\ \kappa_j^{\mu}\mu_j^{\mu} + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^{\mu}\mu_0^{\mu}}{\kappa_{j+1}^{\mu}} + \frac{\widehat{x_{j+1}}}{\kappa_{j+1}^{\mu}}$$

$$\kappa_{j+1}^{\mu} = \widehat{\mathrm{w}}\ \kappa_j^{\mu} + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^{\mu} + 1$$

$$\alpha_{j+1}^{\sigma} = \widehat{\mathrm{w}}\ \alpha_j^{\sigma} + (1\text{-}\widehat{\mathrm{w}})\ \alpha_0^{\sigma} + \frac{1}{2}$$

$$\beta_{j+1}^{\sigma} = \widehat{\mathrm{w}}\ \beta_j^{\sigma} + (1\text{-}\widehat{\mathrm{w}})\ \beta_0^{\sigma} + \frac{1}{2}\left(\widehat{\mathrm{w}}\ \kappa_{j+1}^{\mu}(\mu_{j+1}^{\mu} - \mu_j^{\mu})^2 + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^{\mu}(\mu_{j+1}^{\mu} - \mu_0^{\mu})^2 + (\widehat{x_{j+1}} - \mu_{j+1}^{\mu})^2 + \mathbb{V}ar\left[x_{j+1}\right]\right)$$

$$\begin{aligned}
\text{where } \widehat{x_{j+1}} &= \mathbb{E}_q\left[\mathbb{E}_p\left[x_{j+1}\right]\right] \\
&= \mu_j^{\mu} \\
\text{and } \mathbb{V}ar\left[x_{j+1}\right] &= \mathbb{E}_q\left[\mathbb{E}_p\left[x_{j+1}^2\right]\right] - \mathbb{E}_q\left[\mathbb{E}_p\left[x_{j+1}\right]\right]^2 \\
&= \frac{\beta_j^{\sigma}}{\alpha_j^{\sigma} - 1} + \frac{\beta_j^{\sigma}}{\kappa_j^{\mu}(\alpha_j^{\sigma} - 1)}.
\end{aligned}$$

Note that the posterior predictive variance of x_{j+1} is equal to the sum of the expected variance and the variance of the mean. A similar update paradigm can be used for the opposite limit case where it is assumed that the distribution of x_{j+1} is totally undetermined (i.e. that it has come back to the marginal prior distribution $p(x_j|\,\boldsymbol{\theta}_0)$):

$$\begin{aligned}
\widehat{x_{j+1}} &= \mathbb{E}_q\left[\mathbb{E}_p\left[x_{j+1}\right]\right] \\
&= \mu_0^{\mu} \\
\mathbb{V}ar\left[x_{j+1}\right] &= \mathbb{E}_q\left[\mathbb{E}_p\left[x_{j+1}^2\right]\right] - \mathbb{E}_q\left[\mathbb{E}_p\left[x_{j+1}\right]\right]^2 \\
&= \frac{\beta_0^{\sigma}}{\alpha_0^{\sigma} - 1} + \frac{\beta_0^{\sigma}}{\kappa_0^{\mu}(\alpha_0^{\sigma} - 1)}.
\end{aligned}$$

Finally, the mixed approach consist in considering that the value x_{j+1} is a weighted average of the two given the learned mixing coefficient:

$$\begin{aligned}
\widehat{x_{j+1}} &= \mu_*^\mu \\
\mathbb{V}ar\left[x_{j+1}\right] &= \frac{\beta_*^\sigma}{\alpha_*^\sigma - 1} + \frac{\beta_*^\sigma}{\kappa_*^\mu(\alpha_*^\sigma - 1)} \\
&\text{where} \\
\mu_*^\mu &= \frac{\widehat{\mathrm{w}}\ \kappa_j^\mu \mu_j^\mu + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^\mu \mu_0^\mu}{\kappa_*^\mu} \\
\kappa_*^\mu &= \widehat{\mathrm{w}}\ \kappa_j^\mu + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^\mu \\
\alpha_*^\sigma &= \widehat{\mathrm{w}}\ \alpha_j^\sigma + (1\text{-}\widehat{\mathrm{w}})\ \alpha_0^\sigma \\
\beta_*^\sigma &= \widehat{\mathrm{w}}\ \beta_j^\sigma + (1\text{-}\widehat{\mathrm{w}})\ \beta_0^\sigma + \\
&\quad \frac{1}{2}\left(\widehat{\mathrm{w}}\ \kappa_j^\mu(\mu_*^\mu - \mu_j^\mu)^2 + (1\text{-}\widehat{\mathrm{w}})\ \kappa_0^\mu(\mu_*^\mu - \mu_0^\mu)^2\right).
\end{aligned}$$

Using this update scheme, the agent erases his memory of the posterior distribution at a rate dictated by the stability of the environment, and $\boldsymbol{\theta}_j \to \boldsymbol{\theta}_0$ as $j \to \infty$. Together with the decision algorithm presented in Section 4.2.5, this result shows that, as the posterior distributions broadens and approaches the initial prior, the likelihood of choosing this action will also increase because the expected random noise of the evidence accumulation process will also increase.

4.5.3.3 Delayed approximate posterior updating

Another approach for the agent to consider the evolution of the environment when selecting an action whose outcome has not been seen for a long time is to simulate the expected waning of the previous update across the elapsed interval of time.

Let us consider the case where the agent has chosen an action (e.g. left) at the trial j, and then the opposite action (right) for n trials. We assume that, during these n trials, she has been updating only the value of the right action, leaving the approximate posterior over the distribution parameters of the value of the left action untouched. When selecting left at the trial $j + n$, she considers that the weight of the posterior component of the prior distribution has decreased exponentially for n trials. The prior then looks like:

$$\frac{p(\mathbf{z} \mid \boldsymbol{\theta}_{j-n})^{w^n} p(\mathbf{z} \mid \boldsymbol{\theta}_0)^{1-w^n}}{Z}$$

In order to compute the NCVMP update of the approximate posterior parameters over w, we then need to compute the expected value of w^n. If $q_j(w)$ is set to be a beta

distribution with parameters $\boldsymbol{\phi}_j = \{\phi_{1j}, \phi_{2j}\}$, we have

$$\mathbb{E}_{q_j(w)}\left[w^n\right] = \frac{\Gamma(\phi_{2j})\Gamma(\phi_{1j}+n)}{B(\phi_{1j},\phi_{2j})\Gamma(\phi_{1j}+\omega_{2j}+n)} \tag{4.26}$$

where $B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$. The expected value of the squared decay w^{2n} (used to compute the variance in the expected value of the second order Taylor expansion of the log-partition function.

Unlike the previous method, the shape parameter of the gamma distribution over the variance does not need to be greater than 1 here, as we do not need to simulate the variance of the non-selected action value. However, this is true only for the single stage-case, as the multi-stage case also bases its value updates on expected squared values of rewards, thereby requiring $\alpha_0^\sigma > 1$ too.

4.5.4 Discount factor inference

4.5.4.1 Model specification

The main idea we will try to develop here is that the discount factor γ can be considered as a latent random variable, so that inference is made about its value in order to get the most precise estimate of the posterior predictive distribution of the long term return of an action $p(V(s,a) \mid x_{\leq j})$.

For practical reasons, it is easier to model separately the factorized distribution of the current, immediate state-action reward $p(r(s,a))$ and the discounted value of the future state $p(\gamma v(s' \mid s,a))$. In this notation, $\gamma v(s' \mid s,a)$ is a random variable representing the discounted long-term value of taking the action a in state s minus the value of the reward received while performing this action $r(s,a)$:

$$V(s,a) = r(s,a) + \gamma v(s' \mid s,a)$$

Following this logic, we define the following factorized distribution:

$$p\big(r(s,a), v(s' \mid s,a)\big) = p\big(r(s,a) \mid \mu^r(s,a), \sigma^r(s,a)\big)\, p\big(v(s' \mid s,a) \mid \mu^v(s,a), \sigma^v(s' \mid s,a), \gamma\big)$$

where $r(s,a)$ is normally distributed and $v(s' \mid s,a)$ is distributed according to the following modified Gaussian distribution:

$$p\left(v(s' \mid s,a) \mid \mu^v(s,a), \sigma^v(s,a), \gamma\right) = \gamma \frac{\exp\left\{\frac{(\mu^v(s,a) - \gamma v(s'|s,a))^2}{2\sigma^v(s,a)^2}\right\}}{\sqrt{2\pi}\,\sigma^v(s,a)}.$$

It becomes clear that this distribution integrates to 1 over $v(s' \mid s, a)$, and that it corresponds to a Normal distribution over $\gamma v(s' \mid s, a)$. Therefore, the probability distribution of $V(s, a) = r(s, a) + \gamma v(s' \mid s, a)$ can be written as:

$$p(V(s,a)) = \mathcal{N}(\mu^r(s,a) + \mu^v(s,a),\, \sigma^r(s,a)^2 + \sigma^v(s,a)^2).$$

This joint probability distribution encodes, for each state-action pair the probability distribution of the associated long-term, discounted return.

We naturally define the prior probability of the discounting factor $\gamma \in [0; 1]$ at $j = 1$ as a Beta distribution with parameters $\mathrm{B}(\alpha_0^\gamma, \beta_0^\gamma)$, and a $\mathcal{NG}^{-1}$ prior over the Gaussian parameters $\{\mu^r(s,a), \sigma^r(s,a), \mu^v(s,a), \sigma^v(s,a)\}$.

4.5.4.2 Learning from observed transitions and rewards

We now tackle the question of how to estimate the value of the state that has been reached $v(s' \mid s, a)$. Indeed, this value, contrary to the reward, is not directly observed but must be inferred from the state of arrival s'. We consider the following state-value inference model:

$$\begin{aligned} v(s' \mid s, a) &= \mathbb{E}_{\pi(a'|s')}\left[V(s', a')\right] \\ &= \mathbb{E}_{\pi(a'|s')}\left[r(s', a') + v(s' \mid s, a)\right]. \end{aligned} \tag{4.27}$$

Equation (4.27) stated that the agent assumes that the expected value of the next state is equal to the average of the various state-action values available, weighted by the probability of selection the corresponding action (the observed policy $\pi(a' \mid s')$). We assume that the agent does not have a direct access to the complete description of the policy (meaning that, whereas she can make a decision efficiently, she cannot retrieve the summary statistics of this policy). This assumption is in accordance with the decision process described in Section 4.2.5. This policy can, however, be learned similarly to $r(s, a)$ using a multinomial distribution with a Dirichlet (or Beta in the case of Two-Alternative Forced Choice tasks) prior and approximate posterior: at each trial, the agent observes the action she has performed, and update her belief of the current policy.

Let us now consider how the expected log-joint probability of $v(s' \mid s, a)$ would appear in the ELBO if this value was observed under the mean-field assumption:

$$\begin{aligned} \mathbb{E}_{q_{j+1}(\mu^v(s,a),\sigma^v(s,a)^2,\gamma)}\left[\log p(v(s' \mid s,a)|\mu^v(s,a),\sigma^{v2},\gamma)\right] &= -\frac{1}{2}\mathbb{E}_{q_{j+1}(\sigma^v(s,a)^2)}\left[\log \sigma^v(s,a)^2\right] \\ - \frac{1}{2}\mathbb{E}_{q_{j+1}(\mu^v(s,a),\sigma^v(s,a)^2)}\left[\frac{(\gamma v(s' \mid s,a) - \mu^v(s,a))^2}{\sigma^v(s,a)^2}\right] &+ \mathbb{E}_{q_{j+1}(\gamma)}\left[\log \gamma\right] + \text{cst}. \end{aligned} \tag{4.28}$$

Once the agent reaches the next state $\widehat{s'}$, inference of the value $\mathbb{E}\left[v(s' \mid s,a) \mid s' = \widehat{s'}\right]$ can be achieved by using the posterior predictive expectation of this formula:

$$\begin{aligned}
\mathbb{E}\left[v(s' \mid s,a) \mid s' = \widehat{s'}\right] &= \mathbb{E}_{p(v|x_{\leq j})}\left[v(\widehat{s'} \mid s,a)\right] \\
&\approx \mathbb{E}_{q_j(\mathbf{z})}\left[\mathbb{E}_{p(V,\pi|\mathbf{z})}\left[\mathbb{E}_{\pi}\left[V(\widehat{s'},a')\right]\right]\right] \\
&= \mathbb{E}_{q_j(\mathbf{z})}\left[\mathbb{E}_{p(r,v,\pi|\,\mathbf{z})}\left[\mathbb{E}_{\pi}\left[r(\widehat{s'},a') + \gamma v(s''|s',a')\right]\right]\right] \\
&= \mathbb{E}_{q_j(\mathbf{z})}\left[\mathbb{E}_{p(r,v,\pi|\,\mathbf{z})}\left[\sum_{a' \in A} \pi(a' \mid \widehat{s'})(r(\widehat{s'},a') + \gamma v(s''|\widehat{s'},a'))\right]\right] \\
&= \sum_{a' \in a_1,a_2} \frac{\delta_j^{\pi}(\widehat{s'},a')}{\sum_{a \in A}\delta_j^{\pi}(\widehat{s'},a)}(\mu_j^r(\widehat{s'},a') + \mu_j^v(\widehat{s'},a'))
\end{aligned} \tag{4.29}$$

where A is the set of actions available, and where we indexed with j each variational parameter to emphasize the fact that this expectation is made with respect to the previous posterior belief.

Equation (4.28) being linear and quadratic wrt $v(s' \mid s,a)$, we also need to solve the expected value $\mathbb{E}\left[v(s' \mid s,a)^2 \mid s' = \widehat{s'}\right]$:

$$\begin{aligned}
&\mathbb{E}_{p(v|x_{\leq j})}\left[v(s' \mid s,a)^2\right] \approx \\
&\mathbb{E}_{q_j(\mathbf{z})}\left[\mathbb{E}_{p(r,v,\pi|\,\mathbf{z})}\left[\left(\sum_{a' \in A} \pi(a' \mid s')(r(s',a') + \gamma v(s''|s',a'))\right)^2\right]\right].
\end{aligned} \tag{4.30}$$

The expression in Equation (4.30) can be computed iteratively (Algorithm 13):

using the following equivalences

$$\mathbb{E}_{q_{j-1}}\left[\mathbb{E}_{p(r|\mu^r,\sigma^r)}\left[r\right]\right] = \mu_{j-1}^r \tag{4.31}$$

$$\mathbb{E}_{q_{j-1}}\left[\mathbb{E}_{p(r|\mu^r,\sigma^r)}\left[r^2\right]\right] = {\mu_{j-1}^r}^2 + \frac{\beta_{j-1}^r}{(\alpha_{j-1}^r - 1)\kappa_{j-1}^r} + \frac{\beta_{j-1}^r}{(\alpha_{j-1}^r - 1)} \tag{4.32}$$

$$\mathbb{E}_{q_{j-1}}\left[\mathbb{E}_{p(v|\mu^v,\sigma^v,\gamma)}\left[\gamma v\right]\right] = \mu_{j-1}^v \tag{4.33}$$

$$\mathbb{E}_{q_{j-1}}\left[\mathbb{E}_{p(v|\mu^v,\sigma^v,\gamma)}\left[(\gamma v)^2\right]\right] = {\mu_{j-1}^v}^2 + {\sigma_{j-1}^v}^2 \tag{4.34}$$

$$\mathbb{E}_{q_{j-1}}\left[\pi(a|s)^2\right] = \left(\frac{\delta_{j-1}^{\pi(a|s)}}{\delta_0}\right)^2 + \frac{\delta_{j-1}^{\pi(a|s)}(\delta_0 - \delta_{j-1}^{\pi(a|s)})}{\delta_0^2(\delta_0 + 1)} \tag{4.35}$$

$$\mathbb{E}_{q_{j-1}}\left[\pi(a|s)\pi(a' \mid s)\right] = \frac{\delta_{j-1}^{\pi(a|s)}\delta_{j-1}^{\pi(a'|s)}}{\delta_0^2} - \frac{\delta_{j-1}^{\pi(a|s)}\delta_{j-1}^{\pi(a'|s)}}{\delta_0^2(\delta_0 + 1)} \tag{4.36}$$

$$\text{with } \delta_0 = \sum_{a' \in A} \delta^{\pi(a'|s)}$$

where we have dropped several state-action indices for clarity.

Algorithm 13: $\mathbb{E}_{p(v(s'|s,a)|x_{\leq j})}\left[v(s' \mid s,a)^2 | s' = \widehat{s'}\right]$ iterative computation

input: Factorized approximate posterior distribution q_{j-1}, state reached at trial j $\widehat{s'}$

Results: Posterior predictive estimate $\nu = \mathbb{E}_{p(v(s'|s,a)|x_{\leq j})}\left[v(s' \mid s,a)^2 | s' = \widehat{s'}\right]$

$\nu = 0$
for $a \in A$ **do**
  for $a' \in A$ **do**
    if $a == a'$ **then**
      $\nu \mathrel{+}= \mathbb{E}_{q_{j-1}}\left[\pi(a|s')^2\right] \times$ // Eq 4.35
        $\left(\mathbb{E}_{q_{j-1}}\left[\mathbb{E}_p\left[r(s',a)^2\right]\right] + \mathbb{E}_{q_{j-1}}\left[\mathbb{E}_p\left[(\gamma v(s''|s',a))^2\right]\right]\right.$
        // Eq 4.32,Eq 4.34
        $\left. +2\mathbb{E}_{q_{j-1}}\left[\mathbb{E}_p\left[r(s',a)\right]\right]\mathbb{E}_{q_{j-1}}\left[\mathbb{E}_p\left[\gamma v(s''|s',a)\right]\right]\right)$ // Eq 4.31,Eq 4.33
    else
      $\nu \mathrel{+}= \mathbb{E}_{q_{j-1}}\left[\pi(a|s')\pi(a' \mid s')\right] \times$ // Eq 4.36
        $\left(\mathbb{E}_p\left[r(s',a)\right] + \mathbb{E}_{q_{j-1}}\left[\mathbb{E}_p\left[\gamma v(s''|s',a)\right]\right]\right)\times$ // Eq 4.31,Eq 4.33
        $\left(\mathbb{E}_p\left[r(s',a')\right] + \mathbb{E}_{q_{j-1}}\left[\mathbb{E}_p\left[\gamma v(s''|s',a')\right]\right]\right)$ // Eq 4.31,Eq 4.33
  end
end

We can now take the expectation of Equation (4.28) under Equation (4.29) and Equation (4.30).

This formula has the advantage that the learning of the current distribution of $V(s,a)$ takes into account the uncertainty about the policy, about the reward distribution at the next trial and about the long-term discounted return of the actions in the next state.

4.5.5 VPI

VPI [162, 208, 198, 574, 573] solves this problem by giving a positive bonus to each action value corresponding to the potential payoff of selecting this action given the current uncertainty about its value distribution. Formally, this bonus is computed as the posterior expectation of the gain of performing an action given the belief we have of the value taken by the other actions:

$$VPI(a) = \int p(\mu(a) \mid x_{<j}) Gain_a(\mu(a))\, d\,\mu(a)$$

$$\text{where } Gain_a(\mu(a)) = \begin{cases} \mathbb{E}\left[\mu(a_2)\right] - \mu(a) & \text{if } a = a_1 \text{ and } \mu(a) < \mathbb{E}\left[\mu(a_2)\right] \\ -\mathbb{E}\left[\mu(a_1)\right] + \mu(a) & \text{if } a \neq a_1 \text{ and } \mu(a) > \mathbb{E}\left[\mu(a_1)\right] \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

and where we have used the convention that a_1 and a_2 are the actions with the highest and second highest reward respectively. Section 4.5.5 shows that the bonus of a given action is proportional to the expected gain of discovering that this action leads to a higher reward than all the others when it was thought to be sub-optimal, plus the expected gain of discovering that this action is sub-optimal when it was thought to be optimal.

Interestingly, low threshold of the NIGDM Section 4.2.5 favor choices that would have been also favored by the VPI approach: if an action has currently the highest estimate, it will be more encouraged if its variance is wide than if it is narrow, and conversely for punished action with a high variance.

Moreover, the evidence accumulation process can be enriched to incorporate the expected gain of performing an action: as the agent samples the means of the two action values given its current belief, she can add the difference in the gain bonuses computed as in Section 4.5.5. The resulting process can still be modelled as a Wiener process using the Stochastic Gradient Variational Bayes approach described in Section 4.2.6.

4.5.6 NIGDM properties

4.5.6.1 Discrete convergence property

Importantly, as for every Monte Carlo estimate, this estimate becomes more accurate (i.e. less entropic) as the expected number of iterations (absorption time) grows. In the previous example, this happens when the distance between the absorbing boundaries (threshold) is larger, in which case the agent is more likely to select the optimal option. To see this, consider the case where $p\left(r(a_1) \mid x_{<j}\right)$ and $p\left(r(a_2) \mid x_{<j}\right)$ are both Gaussian distributions with a known mean and variance, and where $\mu(a_1) > \mu(a_2)$. We then introduce the following result:

Lemma 4.1. *The probability of hitting the upper boundary ζ tends to 1 as the threshold grows, and the average displacement $\widehat{\delta z}$ concomitantly approaches the true probability that a_1 is more rewarded than a_2: $\widehat{\delta z} = \mathbb{E}_{p(r(a_1)|x_{<j}),p(r(a_2)|x_{<j})}\left[z - z_0\right] \rightarrow p(r(a_1) > r(a_2) \mid x_{<j})$.*

4.5.6.2 Continuous convergence property

The DDM has interesting properties that make it a good candidate to model decision making in Bayesian Reinforcement Learning. We first introduce the following result, which is a generalization of Theorem 4.1 to the continuous case:

Proposition 4.2. *If the DDM parameters (threshold ζ, relative start point $\nu \triangleq z_0/\zeta$, drift rate ξ and noise ς) are fixed within and between trials, the probability of hitting the boundary pointed by the drift approaches* 1 *as the distance between the boundaries grows.*

To show this, one can first start by showing that the Error Rate probability

$$P = \frac{\exp(-2\xi\zeta\nu/\varsigma^2) - \exp(-2\xi\zeta/\varsigma^2)}{1 - \exp(-2\xi\zeta/\varsigma^2)} \tag{4.38}$$

is monotonically increasing wrt ξ. This can be easily seen from the differential equation of this formula. Next, since $\xi\frac{dP}{d\xi} = \zeta\frac{dP}{d\zeta}$, and since $\frac{dP}{d\xi} > 0\,\forall\xi \in \nabla$, then $\text{sign}\left(\frac{dP}{d\zeta}\right) = \text{sign}(\xi)\,\forall\xi \in \nabla$, $\zeta \in \nabla^+$ and, therefore, P is monotonically increasing wrt ζ if $\xi > 0$ and decreasing if $\xi < 0$.

4.5.6.3 NIGDM similarity to QS

In order to show that the NIGDM mimics a QS algorithm for high thresholds, we need to consider example choices and reaction times generated by this process.

To this end, we first introduce an important result:

Proposition 4.3. *If a data collection is generated according to a classical Wiener Diffusion process where the drift and squared noise are sampled* ***at each trial*** *following some distribution $\xi, \varsigma^2 \sim p(\xi, \varsigma^2 \mid \boldsymbol{\theta}_j)$, then this dataset cannot be distinguished from a dataset where the drift and noise are sampled similarly* ***within*** *trials.*

This can be proven trivially by considering the moment generating function of these two distributions. For the trial-by-trial process, we have:

$$\mathbb{E}\left[\mathrm{t}, a \mid \xi_j, \varsigma_j, \zeta\right] = \sum_{i\in\{1,2\}} e^{la_i} \int e^{l\mathrm{t}} p(\mathrm{t}, a_i \mid \xi_j, \varsigma_j, \zeta, \tau, \nu) d\,\mathrm{t}\big|_{l=1} \tag{4.39}$$

where t and a are the reaction time and choice, respectively. Now, as ξ and ς are unknown to us, we marginalize over their prior distribution (i.e. the current approximate posterior of $\boldsymbol{\theta}_j$):

$$\mathbb{E}\left[\mathrm{t}, a \,\middle|\, \boldsymbol{\theta}_j, \zeta\right] = \sum_{i\in\{1,2\}} e^{lc_i} \int e^{l\mathrm{t}} \iint p\left(\mathrm{t}, a_i \mid \zeta, \xi_j, \varsigma_j\right) p\left(\xi_j, \varsigma_j^2 \mid \boldsymbol{\theta}_j\right) dv_j\, d\varsigma_j^2\, dt\big|_{l=1}.$$

One can see that this expression can be re-arranged into the moment generating function of the within-trial process. Because the two distributions have the same moments and

the moment generating function exist for the DDM [359, 469], the two distributions are equal [604].

Theorem 4.3 is important because it states that, if we are able to generate from the NIGDM or fit this probability distribution to a dataset under the assumption of a between-trial variability, then this fit will also be valid in the case of a within-trial variability.

We will show in Section 4.2.6 how this model can be optimized in a Maximum Likelihood and a Bayesian context.

We can now use Theorem 4.2 and Theorem 4.3 to show that this process does not automatically hit the current best estimate when the boundaries are distant enough from each other. During some trials, the difference between the sampled means $\widetilde{\mu}_1$ and $\widetilde{\mu}_2$ will have a sign opposite to the sign of the current estimate of $\mu_1^\mu - \mu_2^\mu$, and the agent will choose the action with the lowest value. More formally:

Proposition 4.4. *Under the NIGDM, the proportion of choices $p(a_1)$ tends to the posterior predictive probability $p(\mu_1 > \mu_2 \mid x_{<j})$ as the threshold tends to infinity.*

This can be shown by considering Theorem 4.3 and Theorem 4.2 in the fixed-parameters case, and then considering that the expected probability of hitting the positive (or similarly the negative) boundary has the following distribution when ζ tends to infinity:

$$\begin{aligned}\lim_{\zeta\to\infty} \mathbb{E}\left[p(a_1) \mid \boldsymbol{\theta}_j, \zeta\right] &= \iiiint \mathbb{1}\left[\text{sign}(\mu_1 - \mu_2) = +1\right] \times \\ &\qquad p\left(\mu_1, \mu_2, \sigma_1, \sigma_2 \mid \boldsymbol{\theta}_j\right)\, d\mu_1\, d\mu_2\, d\sigma_1^2\, d\sigma_2^2 \\ &= p(\mu_1 > \mu_2 \mid \boldsymbol{\theta}_j)\end{aligned} \tag{4.40}$$

$$\text{where } \boldsymbol{\theta}_j = \{\boldsymbol{\mu}^\mu, \boldsymbol{\kappa}^\mu, \boldsymbol{\alpha}^\sigma, \boldsymbol{\beta}^\sigma\} \tag{4.41}$$

where $\mathbb{1}[x]$ is an operator that takes the value of 1 if its condition x is met and zero otherwise.

4.5.7 Sampled Drift Optimisation

Under the assumption that decisions are made according to the scheme described in Section 4.2.5, the HAFVF can be optimized using a simple, informative model. Indeed, we can express the First Passage Time probability density function (and similarly the cumulative distribution function) as

$$p(\mathbf{y}_j|\, \boldsymbol{\theta}_j, \zeta, \nu, \tau) = \int p(\mathbf{y}_j|a_j, \mu_j, w_j, \tau_j, \sigma_j^2)\, p(\mu_j, \sigma_j^2|\, \boldsymbol{\theta}_j) d\mu_j, \sigma_j^2. \tag{4.42}$$

The central idea of the Sampled Drift Optimisation technique is to use the joint distribution (and not the marginal) in a VB framework. Considering that $\boldsymbol{\theta}_j = f(\boldsymbol{\theta}_0, x_{<j})$, we can write the posterior probability of the DDM parameters at the trial j as:

$$p(\boldsymbol{\theta}_0, \mu_j, \sigma_j^2 | \mathbf{y}_j, \zeta, \nu, \tau) = \frac{p(\mathbf{y}_j | \zeta, \nu, \tau, \mu_j, \sigma_j^2) p(\mu_j, \sigma_j^2 | f(\boldsymbol{\theta}_0, x_{<j}))}{p(\mathbf{y}_j, \zeta, \nu, \tau)}. \tag{4.43}$$

Let us now consider the posterior predictive probability density of the difference between the value of two action values at a specific trial. In the case of a single step task (i.e. without temporal discounting), the posterior predictive distribution of the reward difference is:

$$\begin{aligned} p(R_1 - R_2 | x_{1:j-1}) &\approx \int p(R_1 - R_2 | \, \mathbf{z}_j) q(\mathbf{z}_j \, | \, \boldsymbol{\theta}_{j-1}) d \, \mathbf{z}_j \\ &\quad \text{with } R_i = r(s, a_i), \qquad \mathbf{z}_j = \{\mu_1^r, \mu_2^r, \sigma_1^{r2}, \sigma_2^{r2}\} \\ &= \iint p(\phi + R_2 | \mu_1^r, \sigma_1^{r2}) p(R_2 | \mu_2^r, \sigma_2^{r2}) dR_2 \, q(\mathbf{z}_j \, | \, \boldsymbol{\theta}_{j-1}) d \, \mathbf{z}_j. \\ &\quad \text{where } \phi = R_1 - R_2 \\ &= \int p(\phi | \mu_1^r - \mu_2^r, \sigma_1^{r2} + \sigma_2^{r2}) q(\mathbf{z}_j \, | \, \boldsymbol{\theta}_{j-1}) d \, \mathbf{z}_j \end{aligned} \tag{4.44}$$

We see that we can express the posterior predictive probability of the difference of the Reward values as a normal distribution of mean $\mu^r - \mu^r$ and standard deviation $\sqrt{\sigma^{r2} + \sigma^{r2}}$, marginalized over these parameters given their posterior value at the present trial. We can plug the result of Equation (4.44) into Equation (4.42) and then into Equation (4.43) to get:

$$p(\boldsymbol{\theta}_0 \, | \mathbf{y}_j) = \frac{p(\mathbf{y}_j | \zeta, \nu, \tau, \mathbf{z}_j) p(\mathbf{z}_j \, | f(\boldsymbol{\theta}_0, x_{<j}) p(\zeta, \nu, \tau)}{p(\mathbf{y}_j)}. \tag{4.45}$$

This model is easy to optimize using SGVB provided that we can estimate the Jacobian $\nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{\theta}_0, x_{<j})$.

It can also be applied to the discounted value $V(s,a)$ in multistep tasks:

$$
\begin{aligned}
p\left(V(s,a)-V(s,a')|x_{1:j-1}\right) &\approx \int p\left(V(s,a)-V(s,a')|\,\mathbf{z}_j\right) q(\mathbf{z}_j\,|\,\boldsymbol{\theta}_{j-1}) d\,\mathbf{z}_j \\
&= \iint p\left(\phi+V(s,a')|\mu_1^R+\mu_1^v, {\sigma_1^R}^2+{\sigma_1^v}^2\right)\times \\
&\qquad p\left(V(s,a')|\mu_2^R+\mu_2^v, {\sigma_2^R}^2+{\sigma_2^v}^2\right) dr(s,a')\, q(\mathbf{z}_j\,|\,\boldsymbol{\theta}_{j-1}) d\,\mathbf{z}_j \\
&\text{where } \phi = V(s,a)-V(s,a') \\
&= \int p\left(\phi\,\Big|\sum_{u\in\{r,v\}}\mu_1^u-\mu_2^u, \sum_{u\in\{r,v\}}\sum_{a\in A}{\sigma_a^u}^2\right) q(\mathbf{z}_j\,|\,\boldsymbol{\theta}_{j-1}) d\,\mathbf{z}_j
\end{aligned}
\tag{4.46}
$$

Chapter 5

Future Directions and Discussion

It is a very exciting time to be working in cognitive neuroscience and in machine learning: each day, significant steps toward the understanding of the brain are made. The increasing number of collaborations between the various fields of machine learning indicate that the field is sufficiently mature to provide real answers to concrete questions. Statisticians, mathematicians, engineers and neuroscientists now realize how powerful are each of their studies for solving problems that might have appeared unrelated to their area of expertise. From the point of view of a neuroscientist, it is extremely encouraging to see that several influential milestones of the recent models and theories proposed in machine learning have been first proposed in neuroscience or psychological research journals [605, 606, 607, 608, 609, 151, 109].

In this chapter, we follow the same line of thought as before: we will first summarize and link the results of our three studies. Then, we will expand some of the ideas we have developed and propose some further studies (theoretical or experimental) that could unify the methods and theories we propose to other related concepts in neuroscience and machine learning presented in the introduction.

5.1 The quest of a unifying theory of learning and decision making

It should be clear for the reader by now that our aim is not to propose new testable and groundbreaking theories of brain functioning, but rather to unify at the computational and algorithm level several apparently disjoint and somehow contradictory findings about human behaviour. This is what we call *the quest of a unifying theory of learning and decision making*.

Biologically speaking, learning is useless by itself, if it is not aimed at acting on the knowledge acquired. But acting requires knowledge: this recalls RL, which we introduced as a family of algorithms aimed at make the interplay between acting and learning as efficient as possible.

If the actions we make are the prerequisite and the purpose of learning, then it makes little or no sense to assume that actions are independent: if we limit ourselves to this interplay between learning and decision making, the only way an action sequence can be considered as *i.i.d.* is if these actions are independent when conditioned on auto-correlated processes such as learning. However, behavioural optimality is limited [610, 611] and the rules underlying decision making in stable and/or changing environments can be difficult. In Chapter 2, we have proposed a general algorithm to unveil the local (trial-wise) parameters of decision making that compromises a data-driven (i.e. assumption-free) approach with an informative distribution of the behavioural data (the Drift-Diffusion Model, DDM).

We have also seen that VI can be used to fit the full DDM to behavioural data with a reliable and scalable algorithm.

We identify three further expected improvements of the RAE-DDM: first, this algorithm should include a whole set of recent advances in VAE, such as optimization of the sample size for the inner and outer expectation of the FIVO algorithm [119, 120], guarantee of an optimized mutual information between the latent and the observed variables [123], high exploration of the latent space [524], sparsified networks through the use of Variational dropout [410] etc. However, it is unclear how these various approaches can be linked together, especially in the case of a recurrent model such as ours. Second, this model should be extended to other behavioural models, including those that do not enjoy a tractable likelihood (see Box 2.4.1). This is certainly a work that should be conducted thoroughly in an independent study, as the technical challenges to solve are numerous: for instance, many SSM have a closed form first passage time density that involves an infinite sum [354], which obviously needs to be truncated [351]. It is unclear how this kind of approaches would compete with the IVI approach detailed in the introduction. Finally, uninformativeness of the RAE-DDM model constitutes both its strength and weakness: taking the best of both worlds may constitute an important breakthrough in behavioural modelling: a Bayesian model averaging approach could be able to inform the user on the amount of behavioural variability that is explained by her model. In this perspective, the holy grail of such paradigms would be to be able to assess how much a dataset is explained by an informative model somehow nested in an uninformative one.

Next, we then presented the HAFVF as a general memory-shaping algorithm (Chapter 3): we proposed a learning algorithm that actively learns forgets the stability of

the environment in order to cushion the effect of surprising events. We proposed several applications of this algorithm, for smoothing data, for system identification and for stochastic gradient descent.

A major challenge with this model is to find an optimal prior for the three layers of the model: the first layer prior should be broad enough (i.e. weak) to be "more likely" to explain surprising events that the current posterior, but also not too large: if the entropy of the reference prior is extremely high, it makes little or no sense to consider that a specific value the the latent variable is more likely given this prior whose probability density is very close to zero everywhere. The same applies to the two levels above: the prior over the stability w or meta-stability b requires some *a priori* knowledge of the environment structure and of what one should consider as a change point. The first issue would trivially consist in defining what we intend by "optimal": is an approach optimal if it can detect the better the change points? Should the emphasis be put on the capacity to maximize the model evidence? We see that the problem needs to be defined adequately, as different formulations may lead to different answers.

Intuitively, online optimization of the hyperparameters will be challenging because it will not be easy to derive an unbiased estimate of the gradient using the law of large numbers, as the data will not be *i.i.d.* by assumption. Offline learning, however, made on a representative training dataset, could probably be achieved efficiently in some instances.

A second challenge with the HAFVF would be to relax the mean-field assumption while keeping the convenient update scheme of (N)CvMP. Using a multivariate normal approximate posterior over $\{\mathbf{z}, \sigma^{-1}(w), \sigma^{-1}(b)\}$ (for some $\sigma^{-1} : (0,1) \mapsto \mathbb{R}$) could solve this problem. However, it requires us to set a normal prior[1] over $\mathbf{z}$, whatever it encodes (for instance discrete or strictly positive variables). A non-linear transformation might be used, but we immediately see that we would then need to compute the gradient of an intractable expectation in order to apply a NCVMP update scheme. Also, the advantage of the HAFVF is that once the algorithm has been derived for one member of the EF, it can easily be mapped to other members, which would not be the case with a Gaussian approximate posterior. Intuitively, it may seem to make sense to apply EP to the HAFVF to retrieve an upper bound on the estimate of the posterior variance and covariance of the parameters. It is however unclear how EP might be applied to this kind of model that is hard to factorize wrt $\mathbf{z}$, w and b.

In Chapter 4, we have used the HAFVF and coupled it with a theory of optimal decision making known as SPRTs. We have shown that this model was able to reflect various

[1] A normal reference prior is required if we want to be able to compute the normalizing constant of the mixture prior.

levels of sensitivity to overtraining, that it could be mapped to TD(0) learning and that it could be optimized using SGD. As it constitutes the core theoretical contribution of this thesis, we will discuss in what follows the implications of this learning and decision making framework.

5.2 The AC-HAFVF in an active inference perspective

In Section 1.1.3, we have presented active inference as a general theoretical framework for the organization of cerebral functioning. It might not be obvious to think of the AC-HAFVF as a particular instance of this theory of brain organization. Here, we use a slightly different formulation of active inference in order to frame it as a SPRT.

Let us assume that the agent has a prior probability that defines the level of homeostasis to reach: for instance, when an agent at rest experiences that the nutritive reserves are exhausted (*observations*), the prior probability of the *unknown state* (some level of satiety) is distant from the state that is inferred based on current and past observations (utility of the actions undertaken), and the KL divergence between posterior and prior becomes important. These observations lead to hunger, which will possibly drive the agent to reach a new state of equilibrium by taking some actions leading to satiety. Given the model that this agent has of the environment, it will compare various models (each conditioned on a specific *actions*) and select the one that is the most likely. We will see that actions leading to satiety will be more likely in a Thompson sampling (and similarly SPRT) sense.

We sketch the model on the following assumptions: the agent has some model of the distribution of observations given a state (inferred from past observations)

$$p(o \mid s, a) \triangleq p(o \mid \mu(s, a), \sigma(s, a))$$

a fixed prior of the state of equilibrium

$$p_0(s) \triangleq \mathcal{N}(0, 1)$$

and some policy

$$p(a \mid s).$$

Based on past observations, the agent has established the current state

$$q_j(s) \approx p(s \mid o_{\leq j}, a_{\leq j}).$$

Actions are selected if they maximize the posterior predictive probability of new observations. We make the assumption that the agent is able to achieve some form of cross-validation test of seing a new observation given the current belief and some action:

$$\log p(o_k \mid o_{\leq j}, a) = \log \int p(o_k \mid s, a) q_j(s) ds.$$

Deriving an ELBO for the above expression, we see again that the minimum description length principle is respected when the expected likelihood of the observation compensate for the complexity of the model:

$$\log p(o_k \mid o_{\leq j}, a) \geq \underbrace{\mathbb{E}_{q_k(s)}\left[\log p(o_k \mid s, a)\right]}_{\text{Likelihood}} - \underbrace{D_{KL}\left[q_k(s) \middle|\middle| q_j(s)\right]}_{\text{Complexity (least effort)}}. \tag{5.1}$$

Algorithm 14 details an SPRT test based on this measure. and returns a_1 if $z > \zeta$ and

Algorithm 14: Active Inference SPRT decision algorithm

```
Set z = 0;
while −ζ < z < ζ do
    Select an action a ∼ U({a_1, a_2});
    Draw a memory trace o_k ∈ o_{≤j} with k ∼ U(1 : j);
    Compute q_k(s) and respective ELBO L(q_k) ≤ log p(o_k | o_{≤j}, a);
    if a = a_1 then
        c=1;
    else
        c=-1;
    end
    z+ = c ∗ L(q_k);
end
Observe o_j ∼ p(o | s, a);
Update q_{j+1}(s) given o_j;
```

a_2 otherwise.

Note that the distribution of the observations is unstable: for instance, utility of doing nothing – i.e. not getting food – decreases over time, so that it finally becomes negative. Hence the approximate posterior changes with time, and as a consequence, the prior never really looses its impact on the complexity term in Equation (5.1) (see Chapter 3). Therefore, at some point, a threshold is crossed: the evidence (negative surprise) associated with the action "doing nothing" becomes lower than the evidence for gathering food. This mechanism prevents the organism to continuously look for a reward if doing nothing does no improve the model sufficiently. It also strengthen the will to reach a state of equilibrium when the posterior and prior have a high mismatch. Finally, because they suppose that forgetting of past states is necessary for updating the

inference, they explain why overtrained behaviours (where the posterior belief are too strong) would fail to adapt to new observations, such as satiety.

5.3 Amortized inference, memory coresets and Bayesian decison making

The AC-HAFVF in its original flavour makes decisions based on pure simulations from the posterior predictive distribution. In the previous section, we have suggested that the brain could use past events to perform a Bayesian probability ratio test of two alternative hypothesis (actions) that may bring him to a state of better homeostasis. Similarly, in Section 4.4, we have proposed that recent events might be recalled by the agent in order to derive some similar but noisy posterior samples that might be used to accumulate evidence in favour of the preferred option (in accordance with [298, 420]).

Here, we give an informal account of how this might be achieved in the brain at the computational level. There are several reasons why this might be a good strategy: using memory traces of past events allows for inference about unseen events at a lower cost (a recognition network, similar to the one used in Chapter 2, can be used to map new events to latent cause readily). Similarly, it makes the inference process incremental: if an observation is slightly different but related to another (when the same object is seen from two different point of views for instance), a previous inference attempt can be used as a starting point for the new one [109]. Finally, it allows the learning to be shaped to current believes: if a distant memory trace is stored in memory, it can be re-interpreted by a new recognition network when needed. Moreover, the weight of this distant memory can be re-evaluated, which predicts drastic differences with behaviours that would be just guided by mere summary statistics of past experience. In other words, one may classify a distant episode differently as time goes on, and re-interpret this memory in light of recent events. In the same vein, classifying once and for all the summary statistics of event is not optimal, as it does not allow the recognition model to adapt.

Scalability of memory-driven amortized inference More in detail, using memory traces to guide inference (instead of mere summary statistics of the posterior distribution only) can make the inference *scalable*: as we have seen in Paragraph 1.1.2.2.9, the posterior distribution is essentially determined by observations and observations only. Once a mapping from observation to approximate posterior configuration has been set, it can be reused at will given any set of observations. Here, we give a conceptual account

of a model of decision making that makes use of the concept of amortized inference [594, 109, 378] in the context of changing environment.

Specifically, we assume that the brain keeps trace of a recognition mapping $f_{\boldsymbol{\rho}}(\cdot)$ s.t. $\mathbf{z} \sim q_{\boldsymbol{\rho}}(\mathbf{z} \mid \mathbf{x})$ for a set of data $\mathbf{x} = \{ x_i \}_{i=1}^{N}$ that allows it to adapt to new data distributions with a few samples when needed (e.g. after entering in a new environment). Following this inference scheme, it becomes clear that the context (i.e. the global variable $\mathbf{z}$ posterior probability) can be determined by looking at a set of relevant data. Now one needs to ask what can be understood as "relevant".

Relevance of past observations Note that relying on all datapoints available to make inference may not be necessary, and that inference can be carried out at a lower cost if data points are assigned with a certain weight $u(x) \geq 0$ where ideally $N \gg ||\mathbb{1}[w > 0]||$ s.t. the difference of the model likelihoods $|\mathcal{L}_u - \mathcal{L}|$ is minimized. A dataset where data are weighted in such a way is named a *Bayesian coreset*. Several methods exist for building Bayesian coresets [612, 613, 614, 615]. It is likely that the brain uses a similar mechanism to store only memories that may prove to be relevant for future inference [616]: all observations made in the past may not be relevant *per se* to make a decision. For instance, it is useless to remember every instance of the episode "parking the car": the last one is probably the only relevant one to find it back (which is known as *recency effect*). Similarly, it is likely that an individual will keep a stronger memory of winning the lottery than of all the non-winning episodes that preceded: remembering vaguely a few of these (or just the first one, which is known as *primacy effect*) with a weight proportional to the utility of these events (given the model of the environment) is sufficient.

Now let us come back to the problem of making inference about the optimal policy in a changing environment. First, note that the stability measure w in Chapters 3 and 4 should be proportional to the weight given to memories stored (the memory trace u)[2]. Therefore, when forgetting the past contingency, an amortized version of the HAFVF might also reset the memory trace of distant event to 0. On average, recent events will have a higher importance (recency effect), because they are more relevant to make inference about the current contingency.

In this context, decision making with a SPRT (or in general with any SSM) will be carried on by repeatedly accumulating evidence up to some threshold with the following program: (1) retrieving memories based on their weight for the whole set of available

[2]Of course, other forms of utility of past memory might be used to build the coreset, which are less relevant to the present discussion. As mentioned above, the whole advantage of memory-guided amortized inference with coreset would be that the utility of past events can be re-assessed, if these events have not been assigned a utility of 0.

actions, (2) inferring an approximate posteriors from these, (3) drawing samples from these approximate posteriors and (4) comparing the expected return of each of these samples. The first posterior estimate does not need to be optimal (otherwise the mapping from memory samples to latent space would take a considerable amount of time): similar to the scheme proposed in [345], inference can be further amortized if successive memory samples refine dynamically the form of q.

This sketch of a decision making algorithm recalls many recent findings and theories about value-based decision making: it has been proposed for instance that accumulation of evidence for value-based choices is guided by memory [395] and that these choices also require the intervention of the mediotemporal lobe [397] for both memory retrieval and, importantly, simulation and planning [617, 270][3].

We stress that the idea presented in this section and outlined in Figure 5.1 is more a working plan than a mature and testable hypothesis about inference mechanisms in the brain. Several mathematical steps will need to be made to reach a feasible inference algorithm that would perform the kind of computations proposed here above. A crucial algorithmic challenge for a precise mathematical derivations of the ideas presented in this section is to find a scalable amortized inference paradigm that might work with subsets of data in an efficient way: for instance, Li and Mandt [513] propose an algorithm where each set of variables (for instance *each and every* trials belonging to a single participant in Chapter 2) are mapped to the relative global latent space. This is suboptimal for several reasons: (1) it requires us to look at all the data to find the approximate posterior. For computer vision for instance, in a set of images with the same label, some might be darker than others, or nearly exactly identical to many others, so that their information content about the global identifier is reduced. (2) In their implementation of this amortized inference scheme, the neural network mapping images to their latent space require the *exact* same amount of data for each inference process that is conducted, which obviously prevents online and/or amortized inference.

It might be useful for both the neuroscience and machine learning communities to understand how we can derive algorithms for approximate inference with Bayesian coresets that would be amortized, scalable and applicable online.

[3]Interestingly, the idea of using the DDM as a model of memory-based choices was the core idea of the seminal work of Ratcliff [1]. There, it is assumed that memory probes are retrieved and compared, and that each of these evidence pieces is integrated up to a threshold. Here, we extend this process to the problem of *inference*.

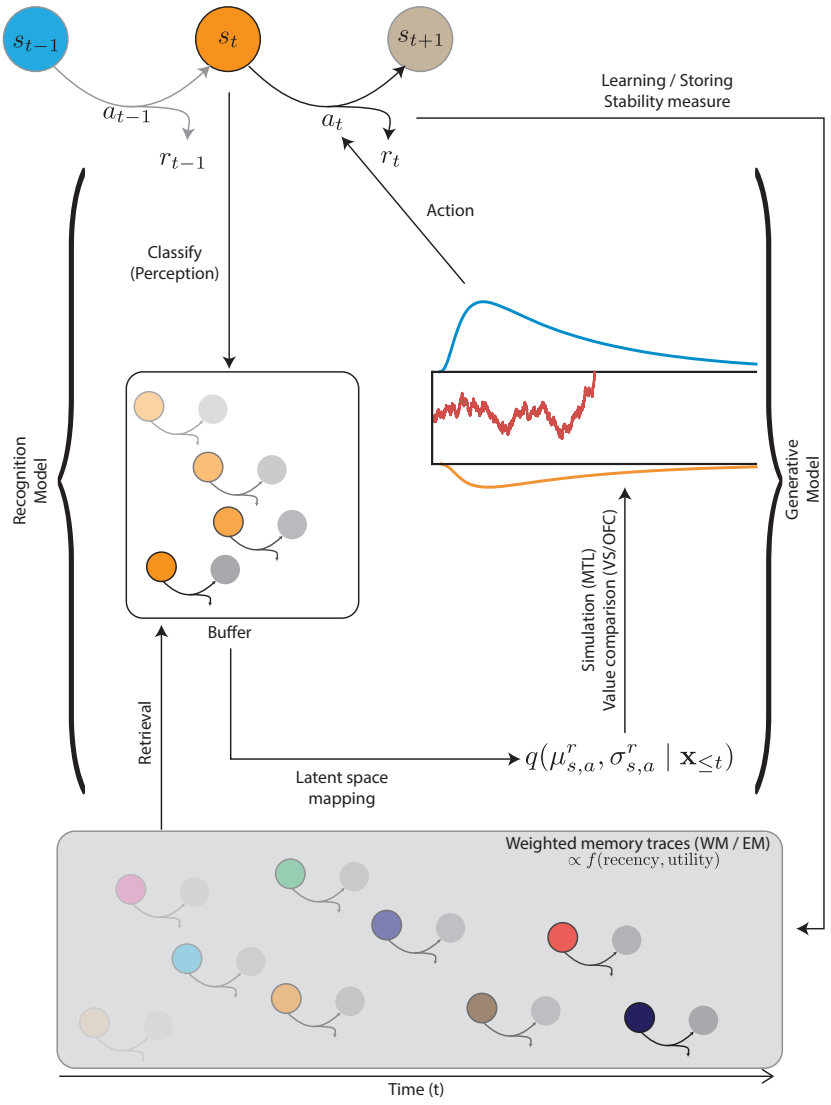


FIGURE 5.1: Use of memory coresets for amortized inference. After completing an action, observations are gathered and used to classify the current state, which in turn retrieves memory traces of similar events with a given weight (alpha level). These events are then used to infer an approximate posterior distribution which is used to simulate new events for the various actions available: these are then compared according to the decision scheme described in Chapter 4 to reach a final decision. Once an action has been taken and a reward retrieved, the generative and recognition models are adapted (learning), stability is measured based on the discrepancy between the observation and the current belief (Chapter 3) and a memory trace is stored. VS stands for ventral striatum, OFC is orbitofrontal cortex, MLT is mediotemporal lobe, WM stands for working memory and EM episodic memory.
The SPRT is shown as an end-point of the inference process, but this should be thought in a more dynamic way, where classification, retrieval, inference, simulation and value comparison interplay during the course of the evidence accumulation. For simplicity, we do not show cross-trial amortization of the inference process, but the simulation of future events achieved at trial t can of course bias (and optimize in a POMDP sense, see Section 1.3.1.1) inference at the successive trial.

5.4 The path to Bayesian model-based decision making

In Chapter 4, we have focused our attention of the problem of bandit tasks and TD learning, leaving the model-based problem aside.

We individuate quite a few challenges that need to be solved before being able to fully implement a model-based version of the HAFVF.

First, thinking of model-based prospective exploration as a process executed at decision time only makes little sense from an amortized inference perspective [109], as the policy resulting from a simulation at trial j could be re-used for future decision in upcoming states. Think for instance of a chess player re-running the same mental exploration of the environment each time she starts to play, and not re-using the previous simulation after observing a transition to a predicted state. This would obviously be a waste of resources: inference during previous games and previous moves should be used and refined each time a new action is undertaken. A good model-based model of decision making should be able to make use of past inference in a way similar to the one proposed in [423]. In Bayesian Model-based RL, this problem is the one that makes the model so difficult to implement: as the approximate posterior model is refined each time a new observation is made, re-elaborating the whole decision tree and exploration/exploitation balance at each time step is a difficult problem.

Another issue will be to derive some precise weighting mechanism between the various controllers (model-based, model-free and successor representation). We have seen in Section 1.2.4.2 that early implementations were simply assuming that the two models were competing based on their uncertainty, and that more synergistic approaches have been proposed (Box 1.2.10).

The use of a precise Bayesian model allows for more refined Bayesian model averaging methods [260]. However, as the two controllers work on essentially different observations and random variables (one on reward and transitions, the other on state-action values), it is unclear how model comparison can be implemented in this setting. Moreover, Bayesian model averaging requires the two models to be fully assessed: this would imply that execution of value-based decisions would require to merge the predictions of the two systems still working in parallel, but that does not put at rest any of these controllers. Therefore, cost minimization will only be achieved with more refined and symbiotic procedures (e.g. [618, 319, 172, 174] and [619, 620] for a review). Also, model comparison based on model evidence only is flawed by the fact that it does not really assessed how each model may predict new data. To do so, items stored in memory should be compared ot assess the posterior predictive capacity of the various models. A

similar algorithm might be used in the brain, but, to our knowledge, this hypothesis has not been explored so far.

A final question to answer will be to know whether the various models measure the model stability jointly or separately (i.e. can the model-based adapt independently of the model-free).

In general, Bayesian model-based RL is a difficult – but nonetheless exciting – problem that requires a deep understanding of POMDPs, but offers interesting possibilities of conducting optimized exploration/exploitation policies, balancing competing model, estimating model accuracy etc.

5.5 Concluding remarks

We conclude this thesis by some general comments about what we think that should define a good algorithm from a machine learning and from a neurocomputational point of view, both at the level of the modelling and at the level of the implementation. This will hopefully give to the reader the intuition of what we have tried to accomplish in this work, and of what grounds we hope that this work will provide for future research.

In general, the ultimate criterion for any model should be its capacity to predict and explain new data. This requires the model to know which subsets of data belong to the same distribution. Following this idea, in changing environment, causal models have to trade between flexibility and robustness of the inference process they rely on, and this is the problem we have addresses in Chapters 3 and 4. This quest for a universal theory of mind should also inspire research in cognitive sciences. From the point of view of the neuroscientist, we believe that it makes less sense to design a model that can explain one dataset generated to test it (through an *ad-hoc* experiment) as it is usually done, but one should aim at explaining a set of (ideally ecological) benchmark datasets first, as it is done in machine learning. We believe that the approach of designing an experiment to test a hypothesis is sound for scientific fields where there is a grounded knowledge of the mechanisms in play and where knowledge is highly incremental, such as physics and chemistry. In neuroscience and psychology, it is more important for a new causal model to be able to account better for old data than explain new ones: we can hope that the emergence of open-science will bring more collaborations and sharing of experimental datasets.

Next, a model should not repeat the same computation more than necessary. We believe that amortized inference can solve both of these problems. Many recent works in VI have shown that amortized inference cannot be implemented naively, and that the cost

of grouping items under the same recognition model cannot be overlooked. We believe that it is very likely that the brain uses similar mechanisms to make decisions: the same state (position in space etc.) is never presented in the same way, but inference must be mostly conducted similarly. Answers to the question of how this mechanism is implemented in the brain remain elusive. Similarly, the optimal way to implement this in machine learning still suffers from the accuracy cost mentioned before.

Finally, an algorithm should be sufficiently simple and general to be applicable to many problems: at the end of the day, an model that needs to be tuned specifically for a single problem and dataset contradicts the first requirement stated in this section. This was the aim pursued in Chapter 2. But there is more to it. We have introduced in Box 1.3.4 the idea that the brain and its components are a sets multiplexing transduction and inference machines: several signals of various nature can pass through the same structures and lead to different results. This is a great challenge for the machine learning community: most models can only handle one type of data (vision, speech, financial data) but even the largest neural networks so far cannot handle a variety of tasks similar to what the brain can do. It might be argued that the computer is the centralizer of all the skills provided by the coded models, but, again, the brain achieves all this work in a synergistic manner.

Moreover, the human brain can flexibly join information gathered from many different senses. To drive a car, we rely on visual and auditory perception, but also on sensory feedback about the actions undertaken. If audition is missing, the system can go on working, and other sensory organs can take the relay to solve the problem at hand: for instance, attention can be redirected. On the contrary, most machine learning algorithms can handle the data they have been programmed to handle, but not less and not more. We believe that understanding better how inference is carried on in the brain will help to build better models, models that we will be able to cross information, use the all the needed information when available, and optimize the way resources are allocated to make better decisions at a lower cost. Similarly, we hope that advances in machine learning will bring new ideas and methods to the neuroscience community in order to develop robust models of brain functioning, and to bring concrete solutions to the many challenges that neurology, psychology and psychiatry face.

Bibliography

[1] Roger Ratcliff. "A theory of memory retrieval." In: *Psychological Review* 85.2 (1978), pp. 59–108. DOI: 10.1037/0033-295X.85.2.59.

[2] Gt Fechner. "Elemente der Psychophysik". In: *Elemente dur psychophysik* (1860). DOI: 10.1111/j.2044-8317.1960.tb00033.x.

[3] Charles P Shimp. "Probabilistically reinforced choice behavior in pigeons1". In: *Journal of the Experimental Analysis of Behavior* 9.4 (July 1966), pp. 443–455. DOI: 10.1901/jeab.1966.9-443.

[4] Michael I Posner. "Orienting of attention". In: *Quarterly Journal of Experimental Psychology* 32.1 (Feb. 1980), pp. 3–25. DOI: 10.1080/00335558008248231.

[5] Philip L Smith and Roger Ratcliff. "Diffusion and Random Walk Processes". In: *International Encyclopedia of the Social & Behavioral Sciences*. Ed. by Elsevier Ltd. Vol. 6. Elsevier, 2015, pp. 395–401. DOI: 10.1016/B978-0-08-097086-8.43037-0.

[6] B.U. Forstmann, R Ratcliff, and E-J Wagenmakers. "Sequential Sampling Models in Cognitive Neuroscience: Advantages, Applications, and Extensions". In: *Annual Review of Psychology* 67.1 (Jan. 2016), pp. 641–666. DOI: 10.1146/annurev-psych-122414-033645.

[7] Roger Ratcliff. "Continuous versus discrete information processing: Modeling accumulation of partial information." In: *Psychological Review* 95.2 (1988), pp. 238–255. DOI: 10.1037/0033-295X.95.2.238.

[8] Roger Ratcliff and Francis Tuerlinckx. "Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability". In: *Psychonomic Bulletin & Review* 9.3 (Sept. 2002), pp. 438–481. DOI: 10.3758/BF03196302.

[9] Roger Ratcliff and Gail McKoon. "The Diffusion Decision Model: Theory and Data for Two-Choice Decision Tasks". In: *Neural Computation* 20.4 (Apr. 2008), pp. 873–922. DOI: 10.1162/neco.2008.12-06-420.

[10] Veronika Lerche and Andreas Voss. "Model Complexity in Diffusion Modeling: Benefits of Making the Model More Parsimonious". In: *Frontiers in Psychology* 7.SEP (Sept. 2016), pp. 1–14. DOI: 10.3389/fpsyg.2016.01324.

[11] M J Frank et al. "fMRI and EEG Predictors of Dynamic Decision Parameters during Human Reinforcement Learning". In: *Journal of Neuroscience* 35.2 (Jan. 2015), pp. 485–494. DOI: 10.1523/JNEUROSCI.2036-14.2015.

[12] Mads Lund Pedersen, Michael J Frank, and Guido Biele. "The drift diffusion model as the choice rule in reinforcement learning". In: *Psychonomic Bulletin & Review* 24.4 (Aug. 2017), pp. 1234–1251. DOI: 10.3758/s13423-016-1199-y.

[13] Brandon M Turner et al. "A method for efficiently sampling from distributions with correlated dimensions." In: *Psychological Methods* 18.3 (2013), pp. 368–384. DOI: 10.1037/a0032222.

[14] Brandon M Turner et al. "A Bayesian framework for simultaneously modeling neural and behavioral data". In: *NeuroImage* 72 (May 2013), pp. 193–206. DOI: 10.1016/j.neuroimage.2013.01.048.

[15] Brandon M. Turner, Leendert Van Maanen, and Birte U. Forstmann. "Informing cognitive abstractions through neuroimaging: The neural drift diffusion model". In: *Psychological Review* (2015). DOI: 10.1037/a0038894.

[16] Brandon M Turner et al. "Approaches to analysis in model-based cognitive neuroscience". In: *Journal of Mathematical Psychology* 76 (Feb. 2017), pp. 65–79. DOI: 10.1016/j.jmp.2016.01.001.

[17] James F Cavanagh et al. "Conflict acts as an implicit cost in reinforcement learning". In: *Nature Communications* 5.1 (Dec. 2014), p. 5394. DOI: 10.1038/ncomms6394.

[18] Clark L. Hull. "Principles of Behavior: An Introduction to Behavior Theory." In: *The Journal of Abnormal and Social Psychology*. 1943. DOI: 10.1037/h0051597.

[19] E. J G Pitman. "Sufficient Statistics and Intrinsic Accuracy". In: *Mathematical Proceedings of the Cambridge Philosophical Society* (1936). DOI: 10.1017/S0305004100019307.

[20] B. O. Koopman. "On distributions admitting a sufficient statistic". In: *Transactions of the American Mathematical Society* 39.3 (Mar. 1936), pp. 399–399. DOI: 10.1090/S0002-9947-1936-1501854-3.

[21] Matthias Seeger, Sebastian Gerwinn, and Matthias Bethge. "Bayesian Inference for Sparse Generalized Linear Models". In: *Most* (2007).

[22] Christopher M Bishop. *Pattern Recognition and Machine Learning*. 2006.

[23] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. New York, NY: Springer New York, 2004. DOI: 10.1007/978-1-4757-4145-2.

[24] Mark A. Beaumont, Wenyang Zhang, and David J. Balding. "Approximate Bayesian computation in population genetics". In: *Genetics* (2002). DOI: GeneticsDecember1, 2002vol.162no.42025-2035.

[25] Arnaud Doucet and Adam M Johansen. "A Tutorial on Particle filtering and smoothing: Fiteen years later". In: *The Oxford handbook of nonlinear filtering* December 2008 (2011), pp. 656–705. DOI: 10.1.1.157.772.

[26] Mario Peruggia. "On the Variability of Case-Deletion Importance Sampling Weights in the Bayesian Linear Model". In: *Journal of the American Statistical Association* (1997). DOI: 10.1080/01621459.1997.10473617.

[27] Andrew Gelman and Donald B Rubin. "Inference from Iterative Simulation Using Multiple Sequences". In: *Statistical Science* (1992). DOI: 10.2307/2246093.

[28] Andrew Gelman and Xiao-Li Meng. "Simulating normalizing constants: from importance sampling to bridge sampling to path sampling". In: *Statistical Science* (1998). DOI: 10.1214/ss/1028905934.

[29] James O Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. New York, NY: Springer New York, 1985. DOI: 10.1007/978-1-4757-4286-2.

[30] José M. Bernardo and Adrian F. M. Smith. "Bayesian Theory". In: *Wiley Series in Probability and Mathematical Statistics* (1994). DOI: 10.1088/0957-0233/12/2/702.

[31] Robert E. Kass and Adrian E. Raftery. "Bayes Factors". In: *Journal of the American Statistical Association* 90.430 (June 1995), p. 773. DOI: 10.2307/2291091.

[32] Jennifer A Hoeting et al. "Bayesian Model Averaging: A Tutorial". In: *Statistical Science* (1999). DOI: 10.2307/2676803.

[33] Donald B. Rubin. "Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician". In: *The Annals of Statistics* (1984). DOI: 10.1214/aos/1176346785.

[34] A Gelman, X.-L Meng, and H Stern. "Posterior predictive assessment of model fitness via realized discrepancies". In: *Statistica Sinica* 6.4 (1996), pp. 733–807. DOI: 10.1.1.142.9951.

[35] Adrian E Raftery. "Bayesian Model Selection in Social Research". In: *Sociological Methodology* 25.1995 (1995), p. 111. DOI: 10.2307/271063.

[36] Håvard Rue, Sara Martino, and Nicolas Chopin. "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71.2 (Apr. 2009), pp. 319–392. DOI: 10.1111/j.1467-9868.2008.00700.x.

[37] A O'Hagan. "Monte Carlo is Fundamentally Unsound". In: *The Statistician* 36.2/3 (1987), p. 247. DOI: 10.2307/2348519.

[38] Jo Berger and Rl Wolpert. *The likelihood principle: A review, generalizations, and statistical implications*. 1988.

[39] Radford M Neal. "Annealed Importance Sampling". In: *Statistics and computing* 11.Neal 2001 (Mar. 1998), p. 2005. DOI: 10.1023/A:1008923215028.

[40] Nicholas Metropolis et al. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092. DOI: 10.1063/1.1699114.

[41] Stuart Geman and Donald Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6 (Nov. 1984), pp. 721–741. DOI: 10.1109/TPAMI.1984.4767596.

[42] Carl Edward Rasmussen and Zoubin Ghahramani. "Bayesian Monte Carlo". In: *Advances in Neural Information Processing Systems 15* 1 (2003), pp. 489–496.

[43] A O'Hagan. "Bayes–Hermite quadrature". In: *Journal of Statistical Planning and Inference* 29.3 (Nov. 1991), pp. 245–260. DOI: 10.1016/0378-3758(91)90002-V.

[44] Marco Wiering and M van Otterlo. *Reinforcement Learning*. Ed. by Marco Wiering and Martijn van Otterlo. Vol. 12. Adaptation, Learning, and Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-27645-3.

[45] Jerome Spanier and Earl H Maize. "Quasi-Random Methods for Estimating Integrals Using Relatively Small Samples". In: *SIAM Review* 36.1 (Mar. 1994), pp. 18–44. DOI: 10.1137/1036002.

[46] William J Morokoff and Russel E Caflisch. "Quasi-Monte Carlo Integration". In: *Journal of Computational Physics* 122.2 (Dec. 1995), pp. 218–230. DOI: 10.1006/jcph.1995.1209.

[47] Russel E Caflisch. "Monte Carlo and quasi-Monte Carlo methods". In: *Acta Numerica* 7 (Jan. 1998), p. 1. DOI: 10.1017/S0962492900002804.

[48] Philipp Schwartenbeck et al. "Evidence for surprise minimization over value maximization in choice behavior". In: *Scientific Reports* 5.1 (Dec. 2015), p. 16575. DOI: 10.1038/srep16575.

[49] Karl Friston et al. “Active inference and learning”. In: *Neuroscience & Biobehavioral Reviews* 68 (Sept. 2016), pp. 862–879. DOI: 10.1016/j.neubiorev.2016.06.022.

[50] Karl Friston. “Life as we know it”. In: *Journal of The Royal Society Interface* 10.86 (July 2013), pp. 20130475–20130475. DOI: 10.1098/rsif.2013.0475.

[51] Guillaume Dehaene and Simon Barthelmé. “Expectation propagation in the large data limit”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80.1 (Jan. 2018), pp. 199–217. DOI: 10.1111/rssb.12241.

[52] Thomas P Minka. “Expectation Propagation for Approximate Bayesian Inference”. In: *Uncertainty in Artificial Intelligence (UAI)* (2001).

[53] Aki Vehtari et al. “Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data”. In: *arXiv:stat.CO* (Dec. 2014), p. 19. URL: http://arxiv.org/abs/1412.4869.

[54] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[55] Alexander Smola, Vishy Vishwanathan, and Eleazar Eskin. “Laplace Propagation”. In: *Advances in Neural Information Processing Systems 16* (2004).

[56] Aki Vehtari, Andrew Gelman, and Jonah Gabry. “Pareto Smoothed Importance Sampling”. In: (July 2015). URL: http://arxiv.org/abs/1507.02646.

[57] G Parisi. *Statistical Field Theory*. Frontiers in Physics. Addison-Wesley, 1988. URL: https://books.google.be/books?id=0F8sAAAAYAAJ.

[58] J. Rissanen. “Modeling by shortest data description”. In: *Automatica* (1978). DOI: 10.1016/0005-1098(78)90005-5.

[59] Shun-ichi Amari. *Differential-Geometrical Methods in Statistics*. Vol. 28. Lecture Notes in Statistics. New York, NY: Springer New York, 1985, pp. 19–94. DOI: 10.1007/978-1-4612-5056-2.

[60] Tom Minka. *Power EP*. Tech. rep. Jan. 2004, p. 6. URL: https://www.microsoft.com/en-us/research/publication/power-ep/.

[61] Thomas P Minka. “Divergence measures and message passing”. In: *Microsoft Research Technical Report* MSR-TR-2005-173 (2005), p. 17.

[62] Jose Hernandez-Lobato et al. “Black-Box Alpha Divergence Minimization”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 1511–1520. URL: http://proceedings.mlr.press/v48/hernandez-lobatob16.html.

[63] Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. "Stochastic Expectation Propagation". In: *Advances in Neural Information Processing Systems 28*. Ed. by C Cortes et al. Curran Associates, Inc., 2015, pp. 2323–2331. URL: http://papers.nips.cc/paper/5760-stochastic-expectation-propagation.pdf.

[64] Michael I. Jordan et al. "Introduction to variational methods for graphical models". In: *Machine Learning* (1999). DOI: 10.1023/A:1007665907178.

[65] Tommi S Jaakkola. "Tutorial on Variational Approximation Methods". In: *In Advanced Mean Field Methods: Theory and Practice* (2000), pp. 129–159. DOI: 10.1.1.31.8989.

[66] H Eugene Stanley. *Introduction to Phase Transitions and Critical Phenomena*. 1971. DOI: 10.1119/1.1986710.

[67] R P Feynman. *Statistical mechanics: a set of lectures*. Frontiers in physics. W. A. Benjamin, 1972. URL: https://books.google.be/books?id=5HG5AAAAIAAJ.

[68] Yaodong Yang et al. "Mean Field Multi-Agent Reinforcement Learning". In: (2018). URL: http://arxiv.org/abs/1802.05438.

[69] John Winn and Christopher M Bishop. "Variational Message Passing". In: *J. Mach. Learn. Res.* 6 (Dec. 2005), pp. 661–694. URL: http://dl.acm.org/citation.cfm?id=1046920.1088695.

[70] David Knowles and Thomas P Minka. "Non-conjugate variational message passing for multinomial and binary regression". In: *Nips* (2011), pp. 1–9. URL: http://eprints.pascal-network.org/archive/00008459/.

[71] Chong Wang and David M Blei. "Variational Inference in Nonconjugate Models". In: *Journal of Machine Learning Research* 14 (Sept. 2012), pp. 1005–1031. URL: http://arxiv.org/abs/1209.4360.

[72] Jean Daunizeau. "The variational Laplace approach to approximate Bayesian inference". In: *J. Daunizeau* Icm (Mar. 2017). URL: http://arxiv.org/abs/1703.02089.

[73] Karl Friston et al. "Variational free energy and the Laplace approximation". In: *NeuroImage* 34.1 (Jan. 2007), pp. 220–234. DOI: 10.1016/j.neuroimage.2006.08.035.

[74] A Gelman et al. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013.

[75] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge: Cambridge University Press, 1998, p. 443. DOI: 10.1017/CBO9780511802256.

[76] Matt Hoffman et al. “Stochastic Variational Inference”. In: (June 2012). URL: http://arxiv.org/abs/1206.7051.

[77] Shun-ichi Amari. “Natural Gradient Works Efficiently in Learning”. In: *Neural Computation* 10.2 (Feb. 1998), pp. 251–276. DOI: 10.1162/089976698300017746.

[78] Manfred Opper and Cédric Archambeau. “The Variational Gaussian Approximation Revisited”. In: *Neural Computation* 21.3 (Mar. 2009), pp. 786–792. DOI: 10.1162/neco.2008.08-07-592.

[79] Cédric Archambeau and M Opper. “Variational Inference for Diffusion Processes”. In: *Advances in Neural ...* (2008), pp. 1–8. URL: http://papers.nips.cc/paper/3282-variational-inference-for-diffusion-processes.

[80] James Hensman, Magnus Rattray, and Neil D Lawrence. “Fast Variational Inference in the Conjugate Exponential Family”. In: (2012), pp. 1–20. URL: http://arxiv.org/abs/1206.5162.

[81] Antti Honkela et al. “Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes”. In: *Journal of Machine Learning Research* 11.5 (2010), pp. 3235–3268. URL: http://eprints.pascal-network.org/archive/00007751/.

[82] M. P. Wand. “Fast Approximate Inference for Arbitrarily Large Semiparametric Regression Models via Message Passing”. In: *Journal of the American Statistical Association* 112.517 (Jan. 2017), pp. 137–168. DOI: 10.1080/01621459.2016.1197833.

[83] Herbert Robbins and Sutton Monro. “A Stochastic Approximation Method”. In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. DOI: 10.1214/aoms/1177729586.

[84] L Bottou. “Stochastic Gradient Learning in Neural Networks”. In: *Proceedings of Neuro-Nımes* 91.8 (1991).

[85] Tom Schaul, Sixin Zhang, and Yann LeCun. “No More Pesky Learning Rates”. In: (June 2012). URL: http://arxiv.org/abs/1206.1106.

[86] Soham De et al. “Big Batch SGD: Automated Inference using Adaptive Batch Sizes”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* 54 (Oct. 2016), pp. 1504–1513. URL: http://proceedings.mlr.press/v54/de17a.html%0Ahttp://arxiv.org/abs/1610.05792%20http://arxiv.org/abs/1610.05792.

[87] Yoon Kim et al. “Semi-Amortized Variational Autoencoders”. In: *Proceedings of the 35th international conference on Machine learning - ICML ’18*. Feb. 2018. URL: http://arxiv.org/abs/1802.02550.

[88] Diederik P Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: (Dec. 2013). URL: http://arxiv.org/abs/1312.6114.

[89] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, July 2014, pp. 1278–1286. URL: http://proceedings.mlr.press/v32/rezende14.html.

[90] John Schulman et al. “Gradient Estimation Using Stochastic Computation Graphs”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’15. Cambridge, MA, USA: MIT Press, 2015, pp. 3528–3536. URL: http://dl.acm.org/citation.cfm?id=2969442.2969633.

[91] Martin Jankowiak and Fritz Obermeyer. “Pathwise Derivatives Beyond the Reparameterization Trick”. In: *Proceedings of the 35th international conference on Machine learning - ICML ’18*. 2018. URL: http://arxiv.org/abs/1806.01851.

[92] Michael C. Fu, Hu Jian-Qiang, and Rakesh Nagi. “Bias properties of infinitesimal perturbation analysis for systems with parallel servers”. In: *Computers & Operations Research* 19.5 (July 1992), pp. 409–423. DOI: 10.1016/0305-0548(92)90070-L.

[93] Pierre L’Ecuyer. *A Unified View of Infinitesimal Perturbation Analysis and Likelihood Ratios*. Tech. rep. US Dept of the Army, Feb. 1989. DOI: 10.21236/ADA210682.

[94] P Glasserman. “Performance continuity and differentiability in Monte Carlo optimization”. In: *1988 Winter Simulation Conference Proceedings*. IEEE, pp. 518–524. DOI: 10.1109/WSC.1988.716212.

[95] Andrew C Miller et al. “Reducing Reparameterization Gradient Variance”. In: (May 2017). URL: http://arxiv.org/abs/1705.07880.

[96] Seiya Tokui and Issei Sato. “Reparameterization trick for discrete variables”. In: (Nov. 2016). URL: http://arxiv.org/abs/1611.01239.

[97] Francisco R Ruiz, Michalis Titsias, and David Blei. “The Generalized Reparameterization Gradient”. In: *30th Conference on Neural Information Processing Systems (NIPS 2016)*. Ed. by D D Lee et al. Curran Associates, Inc., 2016, pp. 460–468. URL: http://papers.nips.cc/paper/6328-the-generalized-reparameterization-gradient.pdf.

[98] Christian A Naesseth et al. “Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms”. In: 54 (Oct. 2016). DOI: 10.16373/j.cnki.ahr.150049.

[99] David A Knowles. "Stochastic gradient variational Bayes for gamma approximating distributions". In: (2015), pp. 1–14. URL: http://arxiv.org/abs/1509.01631.

[100] Michalis Titsias and Miguel Lázaro-Gredilla. "Doubly Stochastic Variational Bayes for non-Conjugate Inference". In: *Proceedings of The 31st International Conference on Machine Learning* 32 (2014), pp. 1971–1979. URL: http://jmlr.org/proceedings/papers/v32/titsias14.

[101] Rajesh Ranganath, Sean Gerrish, and David M Blei. "Black Box Variational Inference". In: *17th International Conerence on Artificial Intelligence and Statistics (AISTATS)* 33 (Dec. 2014). URL: http://arxiv.org/abs/1401.0118.

[102] Francisco J R Ruiz, Michalis K Titsias, and David M Blei. "Overdispersed Black-Box Variational Inference". In: *UAI'16 Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. 2016. URL: http://arxiv.org/abs/1603.01140.

[103] Danilo Jimenez Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In: *Proceedings of the 32nd International Conference on Machine Learning* 37 (May 2015), pp. 1530–1538. URL: http://arxiv.org/abs/1505.05770.

[104] Alp Kucukelbir et al. "Automatic Differentiation Variational Inference". In: *The Journal of Machine Learning Research* 18.1 (Mar. 2017), pp. 430–474. DOI: 10.3847/0004-637X/819/1/50.

[105] Diederik P Kingma et al. "Improving Variational Inference with Inverse Autoregressive Flow". In: Nips (2016). URL: http://arxiv.org/abs/1606.04934.

[106] Jakub M Tomczak and Max Welling. "Improving Variational Auto-Encoders using Householder Flow". In: 2 (Nov. 2016). URL: http://arxiv.org/abs/1611.09630.

[107] Rajesh Ranganath, Dustin Tran, and David M Blei. "Hierarchical Variational Models". In: *arXiv* (2014), pp. 1–9.

[108] Dustin Tran, Rajesh Ranganath, and David M Blei. "The Variational Gaussian Process". In: *Iclr* (Nov. 2015), pp. 1–14. URL: http://arxiv.org/abs/1511.06499.

[109] Samuel J Gershman and Noah D Goodman. "Amortized Inference in Probabilistic Reasoning". In: *Proceedings of the 36th Annual Conference of the Cognitive Science Society (CogSci 2014)* 1 (2014), pp. 517–522.

[110] Andriy Mnih and Karol Gregor. "Neural Variational Inference and Learning in Belief Networks". In: *ArXiv stat.ML* 32.October (2014), pp. 1–20.

[111] Chris Cremer, Xuechen Li, and David Duvenaud. "Inference Suboptimality in Variational Autoencoders". In: *Proceedings of the 35th international conference on Machine learning - ICML '18*. 2018. URL: http://arxiv.org/abs/1801.03558.

[112] Wei Ning Hsu, Yu Zhang, and James Glass. "Learning latent representations for speech generation and transformation". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2017. DOI: 10.21437/Interspeech.2017-349.

[113] Adam Roberts, Jesse Engel, and Douglas Eck. "Hierarchical Variational Autoencoders for Music". In: *Workshop on Machine Learning for Creativity and Design, NIPS*. 2017. URL: https://nips2017creativity.github.io/doc/Hierarchical_Variational_Autoencoders_for_Music.pdf.

[114] Alexey Tikhonov and Ivan P Yamshchikov. "Music generation with variational recurrent autoencoder supported by history". In: *Computer Music Multidisciplinary Research 2017*. 2017. URL: http://arxiv.org/abs/1705.05458.

[115] Shengjia Zhao, Jiaming Song, and Stefano Ermon. "Learning Hierarchical Features from Generative Models". In: *Proceedings of the 34th International Conference on Machine Learning 2017*. 2017. URL: http://arxiv.org/abs/1702.08396.

[116] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. "Unsupervised Image-to-Image Translation Networks". In: *31st Conference on Neural Information Processing Systems 2017*. 2017. URL: http://arxiv.org/abs/1703.00848.

[117] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. "Grammar Variational Autoencoder". In: *Proceedings of the 34th International Conference on Machine Learning 2017*. 2017. URL: http://arxiv.org/abs/1703.01925.

[118] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. "Importance Weighted Autoencoders". In: (Sept. 2015), pp. 1–14. URL: http://arxiv.org/abs/1509.00519.

[119] Tom Rainforth et al. "Tighter Variational Bounds are Not Necessarily Better". In: *Proceedings of the 35th international conference on Machine learning - ICML '18* 2018 (Feb. 2018), pp. 1–4. URL: http://arxiv.org/abs/1802.04537.

[120] Tom Rainforth et al. "On Nesting Monte Carlo Estimators". In: (2017). URL: http://arxiv.org/abs/1709.06181.

[121] Xi Chen et al. "Variational Lossy Autoencoder". In: *ICLR 2017*. 2016, pp. 1–17. URL: http://arxiv.org/abs/1611.02731.

[122] Samuel R. Bowman et al. “Generating Sentences from a Continuous Space”. In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*. 2015, pp. 10–21. DOI: 10.18653/v1/K16-1002.

[123] Alexander A Alemi et al. “Fixing a Broken ELBO”. In: *Proceedings of the 35th international conference on Machine learning - ICML ’18*. Nov. 2018. URL: http://arxiv.org/abs/1711.00464.

[124] Irina Higgins et al. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *Iclr* July (2017), pp. 1–13. URL: https://openreview.net/forum?id=Sy2fzU9gl.

[125] Shakir Mohamed and Balaji Lakshminarayanan. “Learning in Implicit Generative Models”. In: (2016). URL: http://arxiv.org/abs/1610.03483.

[126] Minh-Ngoc Tran, David J Nott, and Robert Kohn. “Variational Bayes with Intractable Likelihood”. In: 117546 (2015), pp. 1–40. URL: http://arxiv.org/abs/1503.08621.

[127] Alexander Moreno et al. “Automatic Variational ABC”. In: (2016), pp. 1–11. URL: http://arxiv.org/abs/1606.08549.

[128] Dustin Tran, Rajesh Ranganath, and David M Blei. “Hierarchical Implicit Models and Likelihood-Free Variational Inference”. In: *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Nips. 2017. URL: http://arxiv.org/abs/1702.08896.

[129] Tilmann Gneiting and Adrian E Raftery. “Strictly Proper Scoring Rules, Prediction, and Estimation”. In: *Journal of the American Statistical Association* 102.477 (Mar. 2007), pp. 359–378. DOI: 10.1198/016214506000001437.

[130] W D Penny et al. “Comparing dynamic causal models”. In: *NeuroImage* 22.3 (2004), pp. 1157–1172. DOI: 10.1016/j.neuroimage.2004.03.026.

[131] William D Penny, Nelson Trujillo-Bareto, and Guillaume Flandin. “Bayesian analysis of single-subject fMRI data: SPM implementation”. In: *Wellcome Department of Imaging Neuroscience* (2005), pp. 1–11.

[132] Klaas Enno Stephan et al. “Bayesian model selection for group studies”. In: *NeuroImage* 46.4 (July 2009), pp. 1004–1017. DOI: 10.1016/j.neuroimage.2009.03.025.

[133] Will D Penny et al. “Comparing families of dynamic causal models”. In: *PLoS Computational Biology* 6.3 (2010). DOI: 10.1371/journal.pcbi.1000709.

[134] J Daunizeau, O David, and K E Stephan. “Dynamic causal modelling: A critical review of the biophysical and statistical foundations”. In: *NeuroImage* 58.2 (2011), pp. 312–322. DOI: 10.1016/j.neuroimage.2009.11.062.

[135] William Penny. "Bayesian Models of Brain and Behaviour". In: *ISRN Biomathematics* 2012 (2012), pp. 1–19. DOI: 10.5402/2012/785791.

[136] W.D. Penny, J. Mattout, and N. Trujillo-Barreto. "Bayesian model selection and averaging". In: *Statistical Parametric Mapping*. Elsevier, 2007, pp. 454–467. DOI: 10.1016/B978-012372560-8/50035-8.

[137] W D Penny. "Comparing Dynamic Causal Models using AIC, BIC and Free Energy". In: *NeuroImage* 59.1 (Jan. 2012), pp. 319–330. DOI: 10.1016/j.neuroimage.2011.07.039.

[138] L Rigoux et al. "Bayesian model selection for group studies - Revisited". In: *NeuroImage* 84 (2014), pp. 971–985. DOI: 10.1016/j.neuroimage.2013.08.065.

[139] Jean Daunizeau, Vincent Adam, and Lionel Rigoux. "VBA: A Probabilistic Treatment of Nonlinear Models for Neurobiological and Behavioural Data". In: *PLoS Computational Biology* 10.1 (2014). DOI: 10.1371/journal.pcbi.1003441.

[140] Philipp Schwartenbeck et al. "Evidence for surprise minimization over value maximization in choice behavior". In: *Scientific Reports* 5 (2015), p. 16575. DOI: 10.1038/srep16575.

[141] K J Friston. "Variational filtering". In: *NeuroImage* 41.3 (2008), pp. 747–766. DOI: 10.1016/j.neuroimage.2008.03.017.

[142] R L Gregory. "Perceptions as Hypotheses". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 290.1038 (July 1980), pp. 181–197. DOI: 10.1098/rstb.1980.0090.

[143] Sophie Deneve. "Bayesian Spiking Neurons I: Inference". In: *Neural Computation* 20.1 (2008), pp. 91–117. DOI: 10.1162/neco.2008.20.1.91.

[144] M J Barber, J W Clark, and C H Anderson. "Neural Representation of Probabilistic Information". In: *Neural Computation* 15.8 (2003), pp. 1843–1864. DOI: 10.1162/08997660360675062.

[145] Sebastian Bitzer et al. "Perceptual decision making: drift-diffusion model is equivalent to a Bayesian model." In: *Frontiers in human neuroscience* 8.February (Jan. 2014), p. 102. DOI: 10.3389/fnhum.2014.00102.

[146] Jeff Beck et al. "Complex Inference in Neural Circuits with Probabilistic Population Codes and Topic Models". In: *Advances in Neural Information Processing Systems 25* (2013).

[147] Tai Sing Lee and David Mumford. "Hierarchical Bayesian inference in the visual cortex". In: *Journal of the Optical Society of America* 20.7 (2003), pp. 1434–1448. DOI: 10.1364/JOSAA.20.001434.

[148] Ronald J. Janssen et al. "Quantifying uncertainty in brain network measures using Bayesian connectomics". In: *Frontiers in Computational Neuroscience* 8 (Oct. 2014). DOI: 10.3389/fncom.2014.00126.

[149] Karl J. Friston, Thomas Parr, and Bert de Vries. "The graphical brain: Belief propagation and active inference". In: *Network Neuroscience* 1.4 (Dec. 2017), pp. 381–414. DOI: 10.1162/NETN{_}a{_}00018.

[150] David C Knill and Alexandre Pouget. "The Bayesian brain: The role of uncertainty in neural coding and computation". In: *Trends in Neurosciences* 27.12 (2004), pp. 712–719. DOI: 10.1016/j.tins.2004.10.007.

[151] Peter Dayan et al. "The Helmholtz Machine". In: *Neural Computation* 7.5 (Sept. 1995), pp. 889–904. DOI: 10.1162/neco.1995.7.5.889.

[152] G. Hinton et al. "The "wake-sleep" algorithm for unsupervised neural networks". In: *Science* 268.5214 (May 1995), pp. 1158–1161. DOI: 10.1126/science.7761831.

[153] Ronald Aylmer Fisher. *The genetical theory of natural selection.* Oxford: Clarendon Press, 1930. DOI: 10.5962/bhl.title.27468.

[154] Guy Sella and Aaron E Hirsch. "The application of statistical physics to evolutionary biology". In: *Pnas* 102.36 (2005), pp. 12690–12693. DOI: 10.1073/pnas.0501865102.

[155] Karl J Friston, Jean Daunizeau, and Stefan J Kiebel. "Reinforcement learning or active inference?" In: *PloS one* 4.7 (Jan. 2009), e6421. DOI: 10.1371/journal.pone.0006421.

[156] L Wittgenstein. "Tractatus Logico-Philosophicus". In: *London: Routledge, 1981* (1922). Ed. by D.F.Pears. URL: http://scholar.google.de/scholar.bib?q=info:1G2GoIkyCZIJ:scholar.google.com/&output=citation&hl=de&ct=citation&cd=0.

[157] S Russell and P Norvig. *Artificial Intelligence: A Modern Approach.* Always learning. Pearson Higher Education & Professional Group, 2016. URL: https://books.google.be/books?id=XS9CjwEACAAJ.

[158] Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. "Learning Without State-Estimation in Partially Observable Markovian Decision Processes". In: *Machine Learning Proceedings 1994*. 1994. DOI: 10.1016/B978-1-55860-335-6.50042-8.

[159] Michael L Littman. "Markov games as a framework for multi-agent reinforcement learning". In: *Machine Learning Proceedings 1994* (1994), pp. 157–163. DOI: 10.1016/B978-1-55860-335-6.50027-1.

[160] Arash Khodadadi, Pegah Fakhari, and Jerome R Busemeyer. "Learning to maximize reward rate: A model based on semi-Markov decision processes". In: *Frontiers in Neuroscience* 8.8 MAY (Jan. 2014), pp. 1–15. DOI: 10.3389/fnins.2014.00101.

[161] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. Second. Sept. 2017. URL: http://ieeexplore.ieee.org/document/712192/.

[162] Richard Dearden, N Friedman, and Stuart Russell. "Bayesian Q-Learning". In: *American Association of Artificial Intelligence (AAAI)-98*. 1998, pp. 761–768.

[163] Daniel Russo et al. "A Tutorial on Thompson Sampling". In: 1 (2017), pp. 1–43. URL: http://arxiv.org/abs/1707.02038.

[164] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. "Thompson Sampling: An Asymptotically Optimal Finite Time Analysis". In: *International Conference on Algorithmic Learning Theory* 1 (2012), pp. 199–213. URL: http://arxiv.org/abs/1205.4217.

[165] R A Howard. *Dynamic Programming and Markov Processes*. NEW YORK: JOHN-WILEY, 1960. URL: https://books.google.be/books?id=fXJEAAAAIAAJ.

[166] R S Sutton and a G Barto. "Temporal credit assignment in reinforcement learning". In: *Computer Science* (1984). DOI: 10.3102/00346543067001043.

[167] G A Rummery and M Niranjan. *On-Line Q-Learning Using Connectionist Systems*. Tech. rep. University of Cambridge, Department of Engineering Cambridge, England, 1994.

[168] Christopher John Cornish Hellaby Watkins. "Learning From Delayed Rewards". PhD thesis. King's College, Cambridge, 1989.

[169] Richard S Sutton. "Dyna, an integrated architecture for learning, planning, and reacting". In: *ACM SIGART Bulletin* 2.4 (1991), pp. 160–163. DOI: 10.1145/122344.122377.

[170] Andrew W Moore and Christopher G Atkeson. "Prioritized Sweeping: Reinforcement Learning with Less Data and Less Real Time". In: *Machine Learning* (1993), pp. 103–130. DOI: 10.1.1.28.7241.

[171] Peter Dayan. "Improving Generalization for Temporal Difference Learning: The Successor Representation". In: *Neural Computation* 5.4 (1993), pp. 613–624. DOI: 10.1162/neco.1993.5.4.613.

[172] Evan M Russek et al. "Predictive representations can link model-based reinforcement learning to model-free mechanisms". In: *PLOS Computational Biology* 13.9 (Sept. 2017). Ed. by Jean Daunizeau, e1005768. DOI: 10.1371/journal.pcbi.1005768.

[173] Kimberly Stachenfeld. "Learning Neural Representations That Support Efficient Reinforcement Learning". PhD thesis. 2018. URL: https://media.proquest.com/media/pq/classic/doc/4326402457/fmt/ai/rep/NPDF?_s=G9TCI3LD85Z6X39gD76ob7maI3D.

[174] I Momennejad et al. "The successor representation in human reinforcement learning". In: *Nature Human Behaviour* 1.9 (2017), pp. 680–692. DOI: 10.1038/s41562-017-0180-8.

[175] Richard S Sutton et al. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *In Advances in Neural Information Processing Systems 12* (1999). DOI: 10.1.1.37.9714.

[176] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8.3-4 (May 1992), pp. 229–256. DOI: 10.1007/BF00992696.

[177] Peter Dayan. "Reinforcement comparison". In: *Proceedings of the 1990 Connectionist Models Summer School* (1990), pp. 45–51.

[178] Lex Weaver and Nigel Tao. "The Optimal Reward Baseline for Gradient-Based Reinforcement Learning". In: *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence* (Jan. 2001), pp. 538–545. URL: http://arxiv.org/abs/1301.2315.

[179] Evan Greensmith, Pl Bartlett, and J Baxter. "Variance reduction techniques for gradient estimates in reinforcement learning". In: *The Journal of Machine Learning ...* 5 (2004), pp. 1471–1530. URL: http://dl.acm.org/citation.cfm?id=1044710.

[180] Sham Machandranath Kakade. "A Natural Policy Gradient". In: *Advances in neural information processing systems* (2001). DOI: 10.1.1.19.8165.

[181] Nicolas Heess et al. "Learning Continuous Control Policies by Stochastic Value Gradients". In: (Oct. 2015). URL: http://arxiv.org/abs/1510.09142.

[182] M Ghavamzadeh et al. "Bayesian Policy Gradient Algorithms". In: *Advances in Neural Information Processing Systems 19* (2007), pp. 457–464.

[183] Andrew G Barto, Richard S Sutton, and Charles W Anderson. "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems". In: *IEEE Transactions on Systems, Man and Cybernetics* (1983). DOI: 10.1109/TSMC.1983.6313077.

[184] Vijay R Konda and John N Tsitsiklis. "Actor-Critic Algorithms". In: *Control Optim* (2003). DOI: 10.1137/S0363012901385691.

[185] Jan Peters, Sethu Vijayakumar, and Stefan Schaal. "Natural Actor-Critic". In: *Neurocomputing* 71.7-9 (2008), pp. 1180–1190. DOI: 10.1016/j.neucom.2007.11.026.

[186] Jeremy Wyatt. "Exploration and Inference in Learning From Reinforcement". PhD thesis. 1998. URL: https://www.era.lib.ed.ac.uk/handle/1842/532.

[187] William R Thompson. "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples". In: *Biometrika* (1933). DOI: 10.2307/2332286.

[188] Eric M Schwartz, Eric T Bradlow, and Peter S Fader. "Customer Acquisition via Display Advertising Using Multi-Armed Bandit Experiments". In: *Marketing Science* (2017). DOI: 10.1287/mksc.2016.1023.

[189] Ian Osband et al. "Deep Exploration via Bootstrapped DQN". In: (Feb. 2016). URL: http://arxiv.org/abs/1602.04621.

[190] Olivier Chapelle and Lihong Li. "An Empirical Evaluation of Thompson Sampling". In: *Advances in neural information processing systems* (2011), pp. 2249–2257. URL: http://explo.cs.ucl.ac.uk/wp-content/uploads/2011/05/An-Empirical-Evaluation-of-Thompson-Sampling-Chapelle-Li-2011.pdf.

[191] Thore Graepel. "Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine". In: *Proceedings of the 27th international conference on machine learning*. 2010.

[192] Deepak Agarwal. "Computational advertising". In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*. New York, New York, USA: ACM Press, 2013, pp. 1585–1586. DOI: 10.1145/2505515.2514690.

[193] Steven L Scott. "Multi-armed bandit experiments in the online service economy". In: *Applied Stochastic Models in Business and Industry* (2015). DOI: 10.1002/asmb.2104.

[194] Tao Wang et al. "Bayesian sparse sampling for on-line reward optimization". In: *Proceedings of the 22nd international conference on Machine learning - ICML '05*. New York, New York, USA: ACM Press, 2005, pp. 956–963. DOI: 10.1145/1102351.1102472.

[195] Arthur Guez, David Silver, and Peter Dayan. "Better Optimism By Bayes: Adaptive Planning with Rich Models". In: *CoRR* abs/1402.1 (2014), p. 11. URL: http://arxiv.org/abs/1402.1958.

[196] Akshay Krishnamurthy, Zhiwei Steven Wu, and Vasilis Syrgkanis. "Semiparametric Contextual Bandits". In: (2018). URL: http://arxiv.org/abs/1803.04204.

[197] Daniel Russo and Benjamin Van Roy. "Learning to Optimize via Information-Directed Sampling". In: (2014), pp. 1–42. DOI: 10.1287/xxxx.0000.0000.

[198] Mehdi Keramati, Amir Dezfouli, and Payam Piray. "Speed/accuracy trade-off between the habitual and the goal-directed processes." In: *PLoS computational biology* 7.5 (May 2011), e1002055. DOI: 10.1371/journal.pcbi.1002055.

[199] Kenneth J Arrow. "Utilities, Attitudes, Choices: A Review Note". In: *Econometrica* 26.1 (1958), p. 1. DOI: 10.2307/1907381.

[200] Nir Vulkan. "An economist's perspective on probability matching". In: *Journal of Economic Surveys* (2000). DOI: 10.1111/1467-6419.00106.

[201] David R Shanks, Richard J Tunney, and John D McCarthy. "A Re-Examination of Probability Matching and Rational Choice". In: *Journal of Behavioral Decision Making* 15.3 (2002), pp. 233–250. DOI: 10.1002/bdm.413.

[202] Genela Morris et al. "Midbrain dopamine neurons encode decisions for future action". In: *Nature Neuroscience* 9.8 (2006), pp. 1057–1063. DOI: 10.1038/nn1743.

[203] Yaakov Engel, Shie Mannor, and Ron Meir. "Bayes meets Bellman: The Gaussian Process Approach to Temporal Difference Learning". In: *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)* (2003). DOI: 10.1017/CBO9781107415324.004.

[204] Y Engel, S Mannor, and R Meir. "Sparse Online Greedy Support Vector Regression". In: *Machine Learning: ECML 2002* 2430.Chapter 8 (2002), pp. 84–96. URL: http://www.springerlink.com/index/10.1007/3-540-36755-1_8%5Cnpapers2://publication/doi/10.1007/3-540-36755-1_8.

[205] Yaakov Engel. "Algorithms and Representations for Reinforcement Learning". In: *Doktorarbeit The Hebrew University of Jerusalem* April (2005). URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.124.6809.

[206] Mohammad Ghavamzadeh and Yaakov Engel. "Bayesian actor-critic algorithms". In: *Proceedings of the 24th international conference on Machine learning - ICML '07*. 2007. DOI: 10.1145/1273496.1273534.

[207] Mohammad Ghavamzadeh, Yaakov Engel, and Michal Valko. "Bayesian Policy Gradient and Actor-Critic Algorithms". In: *Jmlr* 17 (2016), pp. 1–53.

[208] R Dearden et al. "Model based Bayesian exploration". In: *Proceedings of the fifteenth Conference on Uncertainty in Artificial Intelligence* Howard 1966 (1999), pp. 150–159. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.4973&rep=rep1&type=pdf.

[209] Michael Duff. "Design for an Optimal Probe". In: *International Conference on Machine Learning* 1 (2003), pp. 131–138. URL: internal-pdf://duff_icml2003-1457432065/Duff_ICML2003.pdf%20LB%20-.

[210] Mohammad Ghavamzadeh et al. "Bayesian Reinforcement Learning: A Survey". In: *Foundations and Trends in Machine Learning* (2016). DOI: 10.1561/2200000049.

[211] Thomas Furmston and D Barber. "Variational methods for reinforcement learning". In: *International Conference on Machine Learning* (2010), pp. 241–248. URL: http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2010_FurmstonB10.pdf.

[212] Richard S Sutton and Andrew G Barto. "Introduction to Reinforcement Learning". In: *Learning* 4 (1998), pp. 1–5. DOI: 10.1.1.32.7692.

[213] R E Bellman. "Dynamic Programming". In: *Annals of Operations Research* (1957). DOI: 10.1007/BF02188548.

[214] Richard Bellman. *A Markovian decision process*. 1957. DOI: 10.1007/BF02935461.

[215] Peter Dayan et al. "The misbehavior of value and the discipline of the will." In: *Neural networks : the official journal of the International Neural Network Society* 19.8 (Oct. 2006), pp. 1153–1160. DOI: 10.1016/j.neunet.2006.03.002.

[216] Edward L Thorndike and Jelliffe. *Animal Intelligence. Experimental Studies*. 1912. DOI: 10.1097/00005053-191205000-00016.

[217] Christopher D Adams. "Variations in the Sensitivity of Instrumental Responding to Reinforcer Devaluation". In: *The Quarterly Journal of Experimental Psychology Section B* 34.2b (May 1982), pp. 77–98. DOI: 10.1080/14640748208400878.

[218] B Jack Copeland. "The Essential Turing Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma". In: *Cryptologia* (2004). DOI: http://dx.doi.org/10.1017/S1079898600003012.

[219] K Doya. "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?" In: *Neural networks : the official journal of the International Neural Network Society* 12.7-8 (Oct. 1999), pp. 961–974. URL: http://www.ncbi.nlm.nih.gov/pubmed/12662639.

[220] Brenden M Lake et al. "Building Machines That Learn and Think Like People". In: 2012 (2016), pp. 1–58. DOI: 10.1017/S0140525X16001837.

[221] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. *Computational rationality: A converging paradigm for intelligence in brains, minds, and machines*. 2015. DOI: 10.1126/science.aac6076.

[222] Samuel J Gershman. "Deconstructing the human algorithms for exploration". In: *Cognition* 173.August 2017 (2018), pp. 34–42. DOI: 10.1016/j.cognition.2017.12.014.

[223] Anthony Dickinson and D J Nicholas. "Irrelevant incentive learning during instrumental conditioning: The role of the drive-reinforcer and response-reinforcer relationships". In: *The Quarterly Journal of Experimental Psychology Section B* (1983). DOI: 10.1080/14640748308400909.

[224] Anthony Dickinson, D J Nicholas, and Christopher D Adams. "The effect of the instrumental training contingency on susceptibility to reinforcer devaluation". In: *The Quarterly Journal of Experimental Psychology Section B* (1983). DOI: 10.1080/14640748308400912.

[225] A Dickinson. "Actions and Habits: The Development of Behavioural Autonomy". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 308.1135 (Feb. 1985), pp. 67–78. DOI: 10.1098/rstb.1985.0010.

[226] Christopher D Adams and Anthony Dickinson. "Instrumental responding following reinforcer devaluation". In: *The Quarterly Journal of Experimental Psychology Section B : Comparative and Physiological Psychology* 33.October 2013 (1981), pp. 109–121. DOI: 10.1080/14640748108400816.

[227] E C Tolman. "Cognitive maps in rats and men." In: *Psychological review* 55.4 (July 1948), pp. 189–208. URL: http://www.ncbi.nlm.nih.gov/pubmed/18870876.

[228] Adrian M Owen. "Cognitive planning in humans: Neuropsychological, neuroanatomical and neuropharmacological perspectives". In: *Progress in Neurobiology* 53.4 (Nov. 1997), pp. 431–450. DOI: 10.1016/S0301-0082(97)00042-7.

[229] Bernard W Balleine and John P O'Doherty. "Human and rodent homologies in action control: corticostriatal determinants of goal-directed and habitual action." In: *Neuropsychopharmacology : official publication of the American College of Neuropsychopharmacology* 35.1 (Jan. 2010), pp. 48–69. DOI: 10.1038/npp.2009.131.

[230] Henry H Yin, Barbara J Knowlton, and Bernard W Balleine. "Lesions of dorsolateral striatum preserve outcome expectancy but disrupt habit formation in instrumental learning." In: *The European journal of neuroscience* 19.1 (Jan. 2004), pp. 181–189. URL: http://www.ncbi.nlm.nih.gov/pubmed/14750976.

[231] Ann M Graybiel. "Building action repertoires: memory and learning functions of the basal ganglia". In: *Current Opinion in Neurobiology* 5.6 (Dec. 1995), pp. 733–741. DOI: 10.1016/0959-4388(95)80100-6.

[232] Mark G Packard and Barbara J Knowlton. "Learning and Memory Functions of the Basal Ganglia". In: *Annual Review of Neuroscience* 25.1 (Mar. 2002), pp. 563–593. DOI: 10.1146/annurev.neuro.25.112701.142937.

[233] F Gregory Ashby, Benjamin O Turner, and Jon C Horvitz. "Cortical and basal ganglia contributions to habit learning and automaticity". In: *Trends in Cognitive Sciences* 14.5 (May 2010), pp. 208–215. DOI: 10.1016/j.tics.2010.02.001.

[234] Henry H Yin and Barbara J B J Barbara J Knowlton. "The role of the basal ganglia in habit formation." In: *Nature reviews. Neuroscience* 7.6 (June 2006), pp. 464–476. DOI: 10.1038/nrn1919.

[235] A. Floyer-Lea. "Distinguishable Brain Activation Networks for Short- and Long-Term Motor Skill Learning". In: *Journal of Neurophysiology* (2005). DOI: 10.1152/jn.00717.2004.

[236] Jennifer G Waldschmidt and F Gregory Ashby. "Cortical and striatal contributions to automaticity in information-integration categorization". In: *NeuroImage* 56.3 (2011), pp. 1791–1802. DOI: 10.1016/j.neuroimage.2011.02.011.

[237] Y. Matsuzaka, N. Picard, and P. L. Strick. "Skill Representation in the Primary Motor Cortex After Long-Term Practice". In: *Journal of Neurophysiology* (2006). DOI: 10.1152/jn.00784.2006.

[238] S. Helie, J. L. Roeder, and F. G. Ashby. "Evidence for Cortical Automaticity in Rule-Based Categorization". In: *Journal of Neuroscience* (2010). DOI: 10.1523/JNEUROSCI.2393-10.2010.

[239] Etienne Coutureau and Simon Killcross. "Inactivation of the infralimbic prefrontal cortex reinstates goal-directed responding in overtrained rats". In: *Behavioural Brain Research* (2003). DOI: 10.1016/j.bbr.2003.09.025.

[240] Sébastien Hélie, Shawn W. Ell, and F. Gregory Ashby. *Learning robust cortico-cortical associations with the basal ganglia: An integrative review*. 2015. DOI: 10.1016/j.cortex.2014.10.011.

[241] John P O'Doherty et al. "Temporal difference models and reward-related learning in the human brain." In: *Neuron* 38.2 (Apr. 2003), pp. 329–337. URL: http://www.ncbi.nlm.nih.gov/pubmed/12718865.

[242] Ben Seymour et al. "Temporal difference models describe higher-order learning in humans". In: *Nature* 429.6992 (June 2004), pp. 664–667. DOI: 10.1038/nature02581.

[243] J Mirenowicz and W Schultz. "Importance of unpredictability for reward responses in primate dopamine neurons". In: *Journal of Neurophysiology* 72.2 (Aug. 1994), pp. 1024–1027. DOI: 10.1152/jn.1994.72.2.1024.

[244] W Schultz, P Dayan, and P R Montague. "A Neural Substrate of Prediction and Reward". In: *Science* 275.5306 (Mar. 1997), pp. 1593–1599. DOI: 10.1126/science.275.5306.1593.

[245] J.C Horvitz. "Mesolimbocortical and nigrostriatal dopamine responses to salient non-reward events". In: *Neuroscience* 96.4 (Mar. 2000), pp. 651–656. DOI: 10.1016/S0306-4522(00)00019-1.

[246] Kyle Dunovan and Timothy Verstynen. "Believer-Skeptic Meets Actor-Critic: Rethinking the Role of Basal Ganglia Pathways during Decision-Making and Reinforcement Learning". In: *Frontiers in Neuroscience* 10.March (Mar. 2016), pp. 1–15. DOI: 10.3389/fnins.2016.00106.

[247] James F Cavanagh et al. "Frontal theta reflects uncertainty and unexpectedness during exploration and exploitation". In: *Cerebral Cortex* 22.11 (2012), pp. 2575–2586. DOI: 10.1093/cercor/bhr332.

[248] Mark D. Humphries, Mehdi Khamassi, and Kevin Gurney. "Dopaminergic control of the exploration-exploitation trade-off via the basal ganglia". In: *Frontiers in Neuroscience* (2012). DOI: 10.3389/fnins.2012.00009.

[249] Peter Dayan. "Goal-directed control and its antipodes". In: *Neural Networks* 22.3 (Apr. 2009), pp. 213–219. DOI: 10.1016/j.neunet.2009.03.004.

[250] Daniel Kahneman. *Maps of bounded rationality: Psychology for behavioral economics*. 2003. DOI: 10.1257/000282803322655392.

[251] Daniel Kahneman. *Thinking fast, thinking slow*. 2011.

[252] Sébastien Hélie, Jennifer G Waldschmidt, and F Gregory Ashby. "Automaticity in rule-based and information-integration categorization." In: *Attention, perception & psychophysics* 72.4 (2010), pp. 1013–1031. DOI: 10.3758/APP.72.4.1013.

[253] F Gregory Ashby and Matthew J Crossley. "Automaticity and multiple memory systems". In: *Wiley Interdisciplinary Reviews: Cognitive Science* 3.3 (May 2012), pp. 363–376. DOI: 10.1002/wcs.1172.

[254] Russell A Poldrack et al. "The neural correlates of motor skill automaticity." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 25.22 (2005), pp. 5356–5364. DOI: 10.1523/JNEUROSCI.3880-04.2005.

[255] Agnes Moors, Jan De Houwer, and Jan De Houwer. "Automaticity: A Theoretical and Conceptual Analysis." In: *Psychological Bulletin* 132.2 (Mar. 2006), pp. 297–326. DOI: 10.1037/0033-2909.132.2.297.

[256] Nathaniel D Daw, Yael Niv, and Peter Dayan. "Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control". In: *Nature Neuroscience* 8.12 (Dec. 2005), pp. 1704–1711. DOI: 10.1038/nn1560.

[257] Angela J Yu and Peter Dayan. "Uncertainty, neuromodulation, and attention". In: *Neuron* 46.4 (2005), pp. 681–692. DOI: 10.1016/j.neuron.2005.04.026.

[258] M L Platt and S a Huettel. "Risky business: the neuroeconomics of decision making under uncertainty". In: *Nature Neuroscience* 11.4 (2008), pp. 398–403. DOI: 10.1038/nn2062.

[259] Bradley B Doll et al. "Model-based choices involve prospective neural activity". In: *Nature Neuroscience* 18.February (2015), pp. 1–9. DOI: 10.1038/nn.3981.

[260] Thomas H B FitzGerald, Raymond J Dolan, and Karl J Friston. "Model averaging, optimal inference, and habit formation". In: *Frontiers in Human Neuroscience* 8.June (June 2014), pp. 1–11. DOI: 10.3389/fnhum.2014.00457.

[261] Claire M Gillan et al. "Model-based learning protects against forming habits". In: *Cognitive, Affective, & Behavioral Neuroscience* 15.3 (Sept. 2015), pp. 523–536. DOI: 10.3758/s13415-015-0347-6.

[262] Marcos Economides et al. "Model-Based Reasoning in Humans Becomes Automatic with Training". In: *PLoS Computational Biology* 11.9 (2015), pp. 1–19. DOI: 10.1371/journal.pcbi.1004463.

[263] Julie J. Lee and Mehdi Keramati. "Flexibility to contingency changes distinguishes habitual and goal-directed strategies in humans". In: *PLOS Computational Biology* 13.9 (Sept. 2017). Ed. by Samuel J. Gershman, e1005753. DOI: 10.1371/journal.pcbi.1005753.

[264] Wouter Kool, Fiery A Cushman, and Samuel J Gershman. "When Does Model-Based Control Pay Off?" In: *PLoS Computational Biology* 12.8 (2016), pp. 1–34. DOI: 10.1371/journal.pcbi.1005090.

[265] Robert M Hardwick et al. "Skill Acquisition and Habit Formation as Distinct Effects of Practice". In: *bioRxiv* (2017), pp. 1–35. DOI: 10.1101/201095.

[266] Jan Gläscher et al. "Article States versus Rewards : Dissociable Neural Prediction Error Signals Underlying Model-Based and Model-Free Reinforcement Learning". In: *Neuron* 66.4 (May 2010), pp. 585–595. DOI: 10.1016/j.neuron.2010.04.016.

[267] E C Bush and J M Allman. "Frontal Cortex Evolution in Primates". In: *Encyclopedia of Neuroscience*. Elsevier, 2009, pp. 363–366. DOI: 10.1016/B978-008045046-9.00960-8.

[268] Sten Grillner and Brita Robertson. "The Basal Ganglia Over 500 Million Years". In: *Current Biology* 26.20 (Oct. 2016), R1088–R1100. DOI: 10.1016/j.cub.2016.06.041.

[269] Aaron M Bornstein and Nathaniel D Daw. "Cortical and Hippocampal Correlates of Deliberation during Model-Based Decisions for Rewards in Humans". In: *PLoS Computational Biology* 9.12 (Dec. 2013). Ed. by Tim Behrens, e1003387. DOI: 10.1371/journal.pcbi.1003387.

[270] Kevin J. Miller, Matthew M. Botvinick, and Carlos D. Brody. "Dorsal hippocampus contributes to model-based planning". In: *Nature Neuroscience* (2017). DOI: 10.1038/nn.4613.

[271] Peter Smittenaar et al. "Disruption of Dorsolateral Prefrontal Cortex Decreases Model-Based in Favor of Model-free Control in Humans". In: *Neuron* 80.4 (Nov. 2013), pp. 914–919. DOI: 10.1016/j.neuron.2013.08.009.

[272] Richard W Morris et al. "Action-value comparisons in the dorsolateral prefrontal cortex control choice between goal-directed actions." In: *Nature communications* 5 (Jan. 2014), p. 4390. DOI: 10.1038/ncomms5390.

[273] Nathaniel D Daw. "Trial-by-trial data analysis using computational models". In: *Attention & Performance XXIII.* (Mar. 2011), pp. 1–26. DOI: 10.1093/acprof:oso/9780199600434.003.0001.

[274] A E Gelfand, D K Dey, and H Chang. "Model determination using predictive distributions, with implementation via sampling-based methods (with discussion)". In: *Bayesian Statistics 4* (1992).

[275] Aki Vehtari and Jouko Lampinen. "Bayesian model assessment and comparison using cross-validation predictive densities". In: *Neural Computation* (2002). DOI: 10.1162/08997660260293292.

[276] Aki Vehtari and Janne Ojanen. "A survey of Bayesian predictive methods for model assessment, selection and comparison". In: *Statistics Surveys* 6.1 (2012), pp. 142–228. DOI: 10.1214/12-SS102.

[277] Aki Vehtari, Andrew Gelman, and Jonah Gabry. "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC". In: *Statistics and Computing* September (2016), pp. 1–20. DOI: 10.1007/s11222-016-9696-4.

[278] Yuling Yao et al. "Yes, but Did It Work?: Evaluating Variational Inference". In: *Proceedings of the 35th international conference on Machine learning - ICML '18*. Feb. 2018. URL: http://arxiv.org/abs/1802.02538.

[279] Walter Schneider and Richard M Shiffrin. "Controlled and Automatic Human Information Processing: I. Detection, Search, and Attention". In: *Psychological Review* 84.1 (1977), pp. 1–66. DOI: 10.1037/h0021465.

[280] Richard M Shiffrin and Walter Schneider. "Controlled and automatic human information processing: II. Perceptual learning, automatic attending and a general theory." In: *Psychological Review* 84.2 (1977), pp. 127–190. DOI: 10.1037/0033-295X.84.2.127.

[281] Samuel J Gershman and Yael Niv. "Exploring a latent cause theory of classical conditioning". In: *Learning & Behavior* 40.3 (Sept. 2012), pp. 255–268. DOI: 10.3758/s13420-012-0080-8.

[282] Klaus Wunderlich, Peter Dayan, and Raymond J Dolan. "Mapping value based planning and extensively trained choice in the human brain." In: *Nature neuroscience* 15.5 (May 2012), pp. 786–791. DOI: 10.1038/nn.3068.

[283] Nathaniel D Daw et al. "Model-based influences on humans' choices and striatal prediction errors". In: *Neuron* 69.6 (Mar. 2011), pp. 1204–1215. DOI: 10.1016/j.neuron.2011.02.027.

[284] Giovanni Pezzulo, Francesco Rigoli, and Fabian Chersi. "The mixed instrumental controller: using value of information to combine habitual choice and mental simulation." In: *Frontiers in psychology* 4 (Jan. 2013), p. 92. DOI: 10.3389/fpsyg.2013.00092.

[285] Dylan a Simon and Nathaniel D Daw. "Environmental statistics and the trade-off between model-based and TD learning in humans". In: *Advances in Neural Information Processing Systems (NIPS)* (2011), pp. 1–9.

[286] Neal Schmitt, Bryan W Coyle, and Larry King. "Feedback and task predictability as determinants of performance in multiple cue probability learning tasks". In: *Organizational Behavior and Human Performance* 16.2 (Aug. 1976), pp. 388–402. DOI: 10.1016/0030-5073(76)90023-4.

[287] Berndt Brehmer and Jan Kuylenstierna. "Task information and performance in probabilistic inference tasks". In: *Organizational Behavior and Human Performance* 22.3 (Dec. 1978), pp. 445–464. DOI: 10.1016/0030-5073(78)90027-2.

[288] B J Knowlton, L R Squire, and Mark A Gluck. "Probabilistic classification learning in amnesia". In: *Learn Mem* 1.2 (1994), pp. 106–120. DOI: 10.1101/lm.1.2.106.

[289] W Todd Maddox et al. "Category Number Impacts Rule-Based but not Information-Integration Category Learning: Further Evidence for Dissociable Category-Learning Systems". In: *Journal of Experimental Psychology: Learning Memory and Cognition* 30.1 (2004), pp. 227–245. DOI: 10.1037/0278-7393.30.1.227.

[290] W Todd Maddox and F Gregory Ashby. "Dissociating explicit and procedural-learning based systems of perceptual category learning". In: *Behavioural Processes* 66.3 (2004), pp. 309–332. DOI: 10.1016/j.beproc.2004.03.011.

[291] A Ross Otto et al. "Working-memory capacity protects model-based learning from stress". In: *Proceedings of the National Academy of Sciences* 110.52 (Dec. 2013), pp. 20941–20946. DOI: 10.1073/pnas.1312011110.

[292] Darrell A. Worthy, A. Ross Otto, and W. Todd Maddox. "Working-memory load and temporal myopia in dynamic decision making". In: *Journal of Experimental Psychology: Learning Memory and Cognition* (2012). DOI: 10.1037/a0028146.

[293] Wouter Kool, Samuel J Gershman, and Fiery A Cushman. "Cost-Benefit Arbitration Between Multiple Reinforcement-Learning Systems". In: *Psychological Science* (2017), p. 095679761770828. DOI: 10.1177/0956797617708288.

[294] A. Ross Otto et al. "Cognitive Control Predicts Use of Model-based Reinforcement Learning". In: *Journal of Cognitive Neuroscience* (2014). DOI: 10.1162/jocn{_}a{_}00709.

[295] Anne G E Collins and Michael J Frank. "How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis". In: *European Journal of Neuroscience* 35.7 (Apr. 2012), pp. 1024–1035. DOI: 10.1111/j.1460-9568.2011.07980.x.

[296] Bradley B Doll, Daphna Shohamy, and Nathaniel D Daw. "Multiple memory systems as substrates for multiple decision systems." In: *Neurobiology of learning and memory* May (May 2014). DOI: 10.1016/j.nlm.2014.04.014.

[297] Samuel J Gershman and Nathaniel D Daw. "Reinforcement Learning and Episodic Memory in Humans and Animals: An Integrative Framework". In: *Annual Review of Psychology* 68.1 (2017), pp. 101–128. DOI: 10.1146/annurev-psych-122414-033625.

[298] Aaron M Bornstein et al. "Reminders of past choices bias decisions for reward in humans". In: *Nature Communications* 8.May 2015 (June 2017), p. 15958. DOI: 10.1038/ncomms15958.

[299] Aaron M Bornstein and Nathaniel D Daw. "Dissociating hippocampal and striatal contributions to sequential prediction learning". In: *European Journal of Neuroscience* 35.7 (Apr. 2012), pp. 1011–1023. DOI: 10.1111/j.1460-9568.2011.07920.x.

[300] Mattias P Karlsson and Loren M Frank. "Awake replay of remote experiences in the hippocampus". In: *Nature Neuroscience* 12.7 (2009), pp. 913–918. DOI: 10.1038/nn.2344.

[301] H Freyja Ólafsdóttir et al. "Hippocampal place cells construct reward related sequences through unexplored space". In: *eLife* 4.JUNE2015 (2015), pp. 1–17. DOI: 10.7554/eLife.06063.

[302] X Wu and D J Foster. "Hippocampal Replay Captures the Unique Topological Structure of a Novel Environment". In: *Journal of Neuroscience* 34.19 (2014), pp. 6459–6469. DOI: 10.1523/JNEUROSCI.3414-13.2014.

[303] Daphna Shohamy and Nathaniel D Daw. "Integrating memories to guide decisions". In: *Current Opinion in Behavioral Sciences* 5 (2015), pp. 85–90. DOI: 10.1016/j.cobeha.2015.08.010.

[304] Adam Johnson and A David Redish. "Hippocampal replay contributes to within session learning in a temporal difference reinforcement learning model". In: *Neural Networks* 18.9 (Nov. 2005), pp. 1163–1171. DOI: 10.1016/j.neunet.2005.08.009.

[305] Samuel J Gershman, Arthur B Markman, and A Ross Otto. "Retrospective revaluation in sequential decision making: A tale of two systems." In: *Journal of Experimental Psychology: General* 143.1 (2014), pp. 182–194. DOI: 10.1037/a0030844.

[306] J M Ellenbogen et al. "Human relational memory requires time and sleep". In: *Proceedings of the National Academy of Sciences* 104.18 (May 2007), pp. 7723–7728. DOI: 10.1073/pnas.0700094104.

[307] Joshua I Gold and Michael N Shadlen. "The Neural Basis of Decision Making". In: *Annual Review of Neuroscience* (2007). DOI: 10.1146/annurev.neuro.29.051605.113038.

[308] Rafal Bogacz et al. "The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks." In: *Psychological Review* 113.4 (Oct. 2006), pp. 700–765. DOI: 10.1037/0033-295X.113.4.700.

[309] Rafal Bogacz et al. "Do humans produce the speed-accuracy trade-off that maximizes reward rate?" In: *Quarterly Journal of Experimental Psychology* (2010). DOI: 10.1080/17470210903091643.

[310] Y-Lan Boureau, Peter Sokol-Hessner, and Nathaniel D Daw. "Deciding How To Decide: Self-Control and Meta-Decision Making". In: *Trends in Cognitive Sciences* xx.11 (2015), pp. 1–11. DOI: 10.1016/j.tics.2015.08.013.

[311] Brian F Sadacca, Joshua L Jones, and Geoffrey Schoenbaum. "Midbrain dopamine neurons compute inferred and cached value prediction errors in a common framework". In: *eLife* 5.MARCH2016 (2016), pp. 1–13. DOI: 10.7554/eLife.13665.

[312] Michael J Frank, Lauren C Seeberger, and Randall C O'reilly. "By carrot or by stick: cognitive reinforcement learning in parkinsonism." In: *Science (New York, N.Y.)* 306.5703 (Dec. 2004), pp. 1940–1943. DOI: 10.1126/science.1102941.

[313] Madeleine E Sharp et al. “Dopamine selectively remediates ’model-based’ reward learning: A computational approach”. In: *Brain* 139.2 (2016), pp. 355–364. DOI: 10.1093/brain/awv347.

[314] Matthijs A Van Der Meer, A David Redish, and Henry H Yin. “Covert expectation-of-reward in rat ventral striatum at decision points”. In: *Frontiers in integrative ...* 3.February (2009), pp. 1–15. DOI: 10.3389/neuro.07.

[315] Giovanni Pezzulo et al. “Internally generated sequences in learning and executing goal-directed behavior”. In: *Trends in Cognitive Sciences* 18.12 (Aug. 2014), pp. 647–657. DOI: 10.1016/j.tics.2014.06.011.

[316] Francesco P Battaglia et al. “The hippocampus: hub of brain network communication for memory”. In: *Trends in Cognitive Sciences* (June 2011). DOI: 10.1016/j.tics.2011.05.008.

[317] W E DeCoteau et al. “Learning-related coordination of striatal and hippocampal theta rhythms during acquisition of a procedural maze task”. In: *Proceedings of the National Academy of Sciences* 104.13 (Mar. 2007), pp. 5644–5649. DOI: 10.1073/pnas.0700818104.

[318] A B L Tort et al. “Dynamic cross-frequency couplings of local field potential oscillations in rat striatum and hippocampus during performance of a T-maze task”. In: *Proceedings of the National Academy of Sciences* 105.51 (Dec. 2008), pp. 20517–20522. DOI: 10.1073/pnas.0810524105.

[319] Mehdi Keramati et al. “Adaptive integration of habits into depth-limited planning defines a habitual-goal–directed spectrum”. In: *Proceedings of the National Academy of Sciences* 113.45 (2016), pp. 12868–12873. DOI: 10.1073/pnas.1609094113.

[320] Carlos Diuk et al. “Hierarchical learning induces two simultaneous, but separable, prediction errors in human basal ganglia.” In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 33.13 (Mar. 2013), pp. 5797–5805. DOI: 10.1523/JNEUROSCI.5445-12.2013.

[321] Matthew M Botvinick, Yael Niv, and Andrew C Barto. “Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective.” In: *Cognition* 113.3 (Dec. 2009), pp. 262–280. DOI: 10.1016/j.cognition.2008.08.011.

[322] Matthew Michael Botvinick. “Hierarchical reinforcement learning and decision making”. In: *Current Opinion in Neurobiology* 22.6 (Dec. 2012), pp. 956–962. DOI: 10.1016/j.conb.2012.05.008.

[323] Fiery Cushman and Adam Morris. "Habitual control of goal selection in humans". In: *Proceedings of the National Academy of Sciences* 112.45 (2015), p. 201506367. DOI: 10.1073/pnas.1506367112.

[324] Amir Dezfouli and Bernard W Balleine. "Habits, action sequences and reinforcement learning". In: *European Journal of Neuroscience* 35.7 (Apr. 2012), pp. 1036–1051. DOI: 10.1111/j.1460-9568.2012.08050.x.

[325] Amir Dezfouli and Bernard W Balleine. "Actions, Action Sequences and Habits: Evidence That Goal-Directed and Habitual Action Control Are Hierarchically Organized". In: *PLoS Computational Biology* 9.12 (Dec. 2013). Ed. by Tim Behrens, e1003364. DOI: 10.1371/journal.pcbi.1003364.

[326] Amir Dezfouli, Nura W Lingawi, and Bernard W Balleine. "Habits as action sequences: hierarchical action control and changes in outcome value". In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 369.September (2014), pp. 20130482–. DOI: 10.1098/rstb.2013.0482.

[327] Wako Yoshida and Shin Ishii. "Resolution of Uncertainty in Prefrontal Cortex". In: *Neuron* 50.5 (2006), pp. 781–789. DOI: 10.1016/j.neuron.2006.05.006.

[328] Adele Diederich. "Dynamic Stochastic Models for Decision Making under Time Constraints". In: *Journal of Mathematical Psychology* 41.3 (Sept. 1997), pp. 260–274. DOI: 10.1006/jmps.1997.1167.

[329] Matthijs T J Spaan. "Partially Observable Markov Decision Processes". In: *Reinforcement Learning: State-of-the-Art*. Ed. by Marco Wiering and Martijn van Otterlo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 387–414. DOI: 10.1007/978-3-642-27645-3{_}12.

[330] Karl J Friston et al. "Computational psychiatry: the brain as a phantastic organ". In: *The Lancet Psychiatry* 1.2 (2014), pp. 148–158. DOI: 10.1016/S2215-0366(14)70275-5.

[331] Dietrich Albert, K Michael Aschenbrenner, and Franz Schmalhofer. "Cognitive Choice Processes and the Attitude-Behavior Relation". In: *Attitudes and Behavioral Decisions*. Ed. by Arnold Upmeyer. New York, NY: Springer New York, 1989, pp. 61–99. DOI: 10.1007/978-1-4612-3504-0{_}3.

[332] Willem B Verwey. "Buffer Loading and Chunking in Sequential Keypressing". In: *Journal of Experimental Psychology: Human Perception and Performance* (1996). DOI: 10.1037/0096-1523.22.3.544.

[333] B U Forstmann et al. "Striatum and pre-SMA facilitate decision-making under time pressure". In: *Proceedings of the National Academy of Sciences* 105.45 (2008), pp. 17538–17542. DOI: 10.1073/pnas.0805903105.

[334] David Marr. *Vision : a computational investigation into the human representation and processing of visual information*. Cambridge, Mass: MIT Press, 1982.

[335] P Churchland and T Sejnowski. *The Computational Brain*. 1992.

[336] Oron Shagrir. "Structural representations and the brain". In: *British Journal for the Philosophy of Science* (2012). DOI: 10.1093/bjps/axr038.

[337] Oron Shagrir. "The Brain as an Input–Output Model of the World". In: *Minds and Machines* 28.1 (Mar. 2018), pp. 53–75. DOI: 10.1007/s11023-017-9443-4.

[338] Louis Lyons and Louis. "A Paradox about Likelihood Ratios?" In: *eprint arXiv:1711.00775* 1 (2017), pp. 1–10. URL: http://arxiv.org/abs/1711.00775.

[339] A Wald and J Wolfowitz. "Bayes Solutions of Sequential Decision Problems". In: *The Annals of Mathematical Statistics* (1950). DOI: 10.1214/aoms/1177729887.

[340] a. Wald and J Wolfowitz. "Optimum Character of the Sequential Probability Ratio Test". In: *The Annals of Mathematical Statistics* 19.3 (Sept. 1948), pp. 326–339. DOI: 10.1214/aoms/1177730197.

[341] Joshua I Gold and Michael N Shadlen. "Banburismus and the Brain". In: *Neuron* 36.2 (Oct. 2002), pp. 299–308. DOI: 10.1016/S0896-6273(02)00971-6.

[342] Lalit Jain and Kevin Jamieson. "Firing Bandits : Optimizing Crowdfunding". In: *Proceedings of the 35th international conference on Machine learning - ICML '18*. 2018.

[343] Tyler McMillen and Philip Holmes. "The dynamics of choice among multiple alternatives". In: *Journal of Mathematical Psychology* 50.1 (2006), pp. 30–57. DOI: 10.1016/j.jmp.2005.10.003.

[344] Paul Cisek, Geneviève Aude Puskas, and Stephany El-Murr. "Decisions in changing conditions: the urgency-gating model." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 29.37 (Sept. 2009), pp. 11560–11571. DOI: 10.1523/JNEUROSCI.1844-09.2009.

[345] David Thura et al. "Decision making by urgency gating: theory and experimental support." In: *Journal of neurophysiology* 108.11 (2012), pp. 2912–2930. DOI: 10.1152/jn.01071.2011.

[346] Shinichiro Kira, Tianming Yang, and Michael N. Shadlen. "A Neural Implementation of Wald's Sequential Probability Ratio Test". In: *Neuron* (2015). DOI: 10.1016/j.neuron.2015.01.007.

[347] Peter Dayan. "Rationalizable irrationalities of choice". In: *Topics in Cognitive Science* 6.2 (Apr. 2014), pp. 204–228. DOI: 10.1111/tops.12082.

[348] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. "On the Complexity of A/B Testing". In: 35 (May 2014), pp. 1–23. URL: http://arxiv.org/abs/1405.3224.

[349] Rafal Bogacz and Tobias Larsen. "Integration of reinforcement learning and optimal decision-making theories of the basal ganglia". In: *Neural Computation* (2011). DOI: 10.1162/NECO{_}a{_}00103.

[350] William Feller. "An Introduction to Probability Theory and Its Applications". In: *Wiley* 2 (1968), p. 509. DOI: 10.2307/1266435.

[351] Daniel J Navarro and Ian G Fuss. "Fast and accurate calculations for first-passage times in Wiener diffusion models". In: *Journal of Mathematical Psychology* 53.4 (2009), pp. 222–230. DOI: 10.1016/j.jmp.2009.02.003.

[352] Steven P Blurton, Miriam Kesselmeier, and Matthias Gondan. "Fast and accurate calculations for cumulative first-passage time distributions in Wiener diffusion models". In: *Journal of Mathematical Psychology* 56.6 (Dec. 2012), pp. 470–475. DOI: 10.1016/j.jmp.2012.09.002.

[353] Matthias Gondan, Steven P Blurton, and Miriam Kesselmeier. "Even faster and even more accurate first-passage time densities and distributions for the wiener diffusion model". In: *Journal of Mathematical Psychology* 60 (2014), pp. 20–22. DOI: 10.1016/j.jmp.2014.05.002.

[354] Vaibhav Srivastava et al. "First Passage Time Properties for Time-varying Diffusion Models: A Martingale Approach". In: (2015). URL: http://arxiv.org/abs/1508.03373.

[355] Jochen Ditterich. "Evidence for time-variant decision making". In: *European Journal of Neuroscience* 24.12 (2006), pp. 3628–3641. DOI: 10.1111/j.1460-9568.2006.05221.x.

[356] Fábio P Lette and Roger Ratcliff. "Modeling reaction time and accuracy of multiple-alternative decisions". In: *Attention, Perception, and Psychophysics* 72.1 (Jan. 2010), pp. 246–273. DOI: 10.3758/APP.72.1.246.

[357] J A Nelder and R Mead. "A simplex method for function minimization". In: *Computer Journal* (1965). DOI: 10.1093/comjnl/7.4.308.

[358] Eric-Jan Wagenmakers, Han L J van der Maas, and Raoul P P P Grasman. "An EZ-diffusion model for response time and accuracy". In: *Psychonomic Bulletin & Review* 14.1 (2007), pp. 3–22. DOI: 10.3758/BF03194023.

[359] Raoul P P P Grasman, Eric-Jan Wagenmakers, and Han L J Van Der Maas. "On the mean and variance of response times under the diffusion model with an application to parameter estimation". In: *Journal of Mathematical Psychology* 53.2 (Apr. 2009), pp. 55–68. DOI: 10.1016/j.jmp.2009.01.006.

[360] Roger Ratcliff. "The EZ diffusion method: Too EZ?" In: *Psychonomic Bulletin & Review* 15.6 (Dec. 2008), pp. 1218–1228. DOI: 10.3758/PBR.15.6.1218.

[361] Andreas Voss and Jochen Voss. "A fast numerical algorithm for the estimation of diffusion model parameters". In: *Journal of Mathematical Psychology* 52.1 (Feb. 2008), pp. 1–9. DOI: 10.1016/j.jmp.2007.09.005.

[362] Anne K Churchland, Roozbeh Kiani, and Michael N Shadlen. "Decision-making with multiple alternatives". In: *Nature Neuroscience* (2008). DOI: 10.1038/nn.2123.

[363] D R J Laming. *Information theory of choice-reaction times.* Vol. 14. 1968, p. 172. DOI: 10.1002/bs.3830140408.

[364] R Ratcliff and J N Rouder. "Modeling Response Times for Two-Choice Decisions". In: *Psychological Science* 9.5 (1998), pp. 347–356. DOI: 10.1111/1467-9280.00067.

[365] Roger Ratcliff, Gail McKoon, and Trisha van Zandt. "Connectionist and Diffusion Models of Reaction Time". In: *Psychological Review* 106.2 (1999), pp. 261–300. DOI: 10.1037/0033-295X.106.2.261.

[366] Benjamin Y Hayden and Rubén Moreno-Bote. "A neuronal theory of sequential economic choice". In: *Brain and Neuroscience Advances* 2 (2018), p. 239821281876667. DOI: 10.1177/2398212818766675.

[367] Donald Laming. "Choice reaction performance following an error". In: *Acta Psychologica* (1979). DOI: 10.1016/0001-6918(79)90026-X.

[368] Rajesh P N Rao. "Decision making under uncertainty: a neural model based on partially observable markov decision processes." In: *Frontiers in computational neuroscience* 4.November (Jan. 2010), p. 146. DOI: 10.3389/fncom.2010.00146.

[369] Yanping Huang et al. "How Prior Probability Influences Decision Making: A Unifying Probabilistic Model". In: *Advances in Neural Information Processing Systems 25*. 2012.

[370] Yanping Huang and Rajesh P N Rao. "Reward Optimization in the Primate Brain: A Probabilistic Model of Decision Making under Uncertainty". In: *PLoS ONE* 8.1 (2013), pp. 1–11. DOI: 10.1371/journal.pone.0053344.

[371] Paul W Glimcher and Ernst Fehr. *Neuroeconomics: Decision Making and the Brain: Second Edition*. 2013. DOI: 10.1016/C2011-0-05512-6.

[372] Timothy Doyeon Kim, Mohammad Kabir, and Joshua I Gold. "Coupled Decision Processes Update and Maintain Saccadic Priors in a Dynamic Environment". In: *The Journal of Neuroscience* 37.13 (2017), pp. 3632–3645. DOI: 10.1523/JNEUROSCI.3078-16.2017.

[373] Kyle E Dunovan, Joshua J Tremel, and Mark E Wheeler. "Prior probability and feature predictability interactively bias perceptual decisions." In: *Neuropsychologia* 61 (June 2014), pp. 210–221. DOI: 10.1016/j.neuropsychologia.2014.06.024.

[374] Rajesh P N Rao. "Bayesian inference and attentional modulation in the visual cortex". In: *NeuroReport* (2005). DOI: 10.1097/01.wnr.0000183900.92901.fc.

[375] M.W. Spratling. "Predictive coding as a model of biased competition in visual attention". In: *Vision Research* 48.12 (June 2008), pp. 1391–1408. DOI: 10.1016/j.visres.2008.03.009.

[376] Jozsef Fiser et al. "Statistically optimal perception and learning: from behavior to neural representations". In: *Trends in Cognitive Sciences* (2010). DOI: 10.1016/j.tics.2010.01.003.

[377] Sj Gershman, Jd Cohen, and Yael Niv. "Learning to selectively attend". In: *32nd Annual Proceedings of the Cognitive Science Society* (2010).

[378] Ishita Dasgupta et al. "Remembrance of inferences past: Amortization in human hypothesis generation". In: *Cognition* 178 (Sept. 2018), pp. 67–81. DOI: 10.1016/j.cognition.2018.04.017.

[379] Aaron M Bornstein et al. "Perceptual decisions result from the continuous accumulation of memory and sensory evidence". In: *bioRxiv* (2017). DOI: 10.1101/186817.

[380] Martijn J Mulder et al. "Bias in the brain: a diffusion model analysis of prior probability and potential payoff." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 32.7 (Feb. 2012), pp. 2335–2343. DOI: 10.1523/JNEUROSCI.4156-11.2012.

[381] Jeffrey M. Beck et al. "Probabilistic Population Codes for Bayesian Decision Making". In: *Neuron* 60.6 (Dec. 2008), pp. 1142–1152. DOI: 10.1016/j.neuron.2008.09.021.

[382] Udo Boehm and Et Al. "Estimating Between-Trial Variability Parameters of the Diffusion Decision Model: Expert Advice and Recommendations". In: *PsyArXiv* (2018). DOI: https://dx.doi.org/10.17605/OSF.IO/KM28U.

[383] Jiaxiang Zhang and James B Rowe. "Dissociable mechanisms of speed-accuracy tradeoff during visual perceptual learning are revealed by a hierarchical drift-diffusion model." In: *Frontiers in neuroscience* 8.April (Jan. 2014), p. 69. DOI: 10.3389/fnins.2014.00069.

[384] James F Cavanagh et al. "Subthalamic nucleus stimulation reverses mediofrontal influence over decision threshold". In: *Nature Neuroscience* 14.11 (Nov. 2011), pp. 1462–1467. DOI: 10.1038/nn.2925.

[385] Matt Jones and Ehtibar N Dzhafarov. "Unfalsifiability and mutual translatability of major modeling schemes for choice reaction time." In: *Psychological review* 121.1 (2014), pp. 1–32. DOI: 10.1037/a0034190.

[386] Doug P Hanes and Jeffrey D Schall. "Neural control of voluntary movement initiation". In: *Science* 274.5286 (Oct. 1996), pp. 427–430. DOI: 10.1126/science.274.5286.427.

[387] Jamie D Roitman and Michael N Shadlen. "Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 22.21 (2002), pp. 9475–9489. DOI: 10.1016/S0377-2217(02)00363-6.

[388] K. W. Latimer et al. "Single-trial spike trains in parietal cortex reveal discrete steps during decision-making". In: *Science* 349.6244 (July 2015), pp. 184–187. DOI: 10.1126/science.aaa4056.

[389] M. N. Shadlen et al. "Comment on "Single-trial spike trains in parietal cortex reveal discrete steps during decision-making"". In: *Science* 351.6280 (Mar. 2016), pp. 1406–1406. DOI: 10.1126/science.aad3242.

[390] Kenneth W. Latimer et al. "Response to Comment on "single-trial spike trains in parietal cortex reveal discrete steps during decision-making"". In: *Science* (2016). DOI: 10.1126/science.aad3596.

[391] Jacob L. Yates et al. "Functional dissection of signal and noise in MT and LIP during decision-making". In: *Nature Neuroscience* (2017). DOI: 10.1038/nn.4611.

[392] Takafumi Kajihara et al. "Neural dynamics in motor preparation: From phase-mediated global computation to amplitude-mediated local computation". In: *NeuroImage* 118 (2015), pp. 445–455. DOI: 10.1016/j.neuroimage.2015.05.032.

[393] Roozbeh Kiani, Leah Corthell, and Michael N Shadlen. "Choice certainty is informed by both evidence and decision time". In: *Neuron* 84.6 (2014), pp. 1329–1342. DOI: 10.1016/j.neuron.2014.12.015.

[394] V de Lafuente, M Jazayeri, and M N Shadlen. "Representation of Accumulating Evidence for a Decision in Two Parietal Areas". In: *Journal of Neuroscience* 35.10 (2015), pp. 4306–4318. DOI: 10.1523/JNEUROSCI.2451-14.2015.

[395] Michael N N Shadlen and Daphna Shohamy. "Decision Making and Sequential Sampling from Memory". In: *Neuron* 90.5 (June 2016), pp. 927–939. DOI: 10.1016/j.neuron.2016.04.036.

[396] Akram Bakkour, Michael N Shadlen, and Daphna Shohamy. "Where Does Time Go during Value-Based Decisions ? The Case for the Hippocampus". In: *Cognitive Computational Neuroscience* (2016).

[397] A Zeynep Enkavi et al. "Evidence for hippocampal dependence of value-based decisions". In: *Scientific Reports* 7.1 (Dec. 2017), p. 17738. DOI: 10.1038/s41598-017-18015-4.

[398] Marieke K van Vugt, Marijke A Beulen, and Niels A Taatgen. "Is There Neural Evidence for an Evidence Accumulation Process in Memory Decisions?" In: *Frontiers in Human Neuroscience* 10.March (2016), pp. 1–13. DOI: 10.3389/fnhum.2016.00093.

[399] D. J. Tolhurst, J. A. Movshon, and A. F. Dean. "The statistical reliability of signals in single neurons in cat and monkey visual cortex". In: *Vision Research* (1983). DOI: 10.1016/0042-6989(83)90200-6.

[400] Wei Ji Ma et al. "Bayesian inference with probabilistic population codes". In: *Nature Neuroscience* (2006). DOI: 10.1038/nn1790.

[401] Alexandre Pouget et al. *Probabilistic brains: Knowns and unknowns*. 2013. DOI: 10.1038/nn.3495.

[402] Matthew D Hoffman, David M Blei, and Francis Bach. "Online Learning for Latent Dirichlet Allocation". In: *Advances in Neural Information Processing Systems 23 (NIPS 2010)*. 2010, pp. 1–9.

[403] Xaq Pitkow and Dora E. Angelaki. "Inference in the Brain: Statistics Flowing in Redundant Population Codes". In: *Neuron* 94.5 (June 2017), pp. 943–953. DOI: 10.1016/j.neuron.2017.05.028.

[404] Karl Friston and Stefan Kiebel. "Predictive coding under the free-energy principle." In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 364 (2009), pp. 1211–1221. DOI: 10.1098/rstb.2008.0300.

[405] J Von Neumann. "Probabilistic logics and the synthesis of reliable organisms from unreliable components". In: *Automata Studies* (1956). DOI: 10.1128/AEM.00314-09.

[406] Claude E. Shannon. "Von Neumann's contributions to automata theory". In: *Bulletin of the American Mathematical Society* 64.3 (May 1958), pp. 123–130. DOI: 10.1090/S0002-9904-1958-10214-1.

[407] L. C. Katz and C. J. Shatz. *Synaptic activity and the construction of cortical circuits*. 1996. DOI: 10.1126/science.274.5290.1133.

[408] Susana Cohen-Cory. *The developing synapse: Construction and modulation of synaptic structures and circuits*. 2002. DOI: 10.1126/science.1075510.

[409] Song Han et al. "Learning both Weights and Connections for Efficient Neural Networks". In: *Advances in Neural Information Processing Systems 2015*. 2015, pp. 1–9. DOI: 10.1016/S0140-6736(95)92525-2.

[410] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. "Variational Dropout Sparsifies Deep Neural Networks". In: (2017). URL: http://arxiv.org/abs/1701.05369.

[411] Lu Hou and James T. Kwok. "Power Law in Sparsified Deep Neural Networks". In: (2018). URL: http://arxiv.org/abs/1805.01891.

[412] Juho Lee et al. "Adaptive Network Sparsification via Dependent Variational Beta-Bernoulli Dropout". In: (2018), pp. 1–10. URL: http://arxiv.org/abs/1805.10896.

[413] Roger Ratcliff. "A theory of order relations in perceptual matching". In: *Psychological Review* (1981). DOI: 10.1037/0033-295X.88.6.552.

[414] Eric-Jan Wagenmakers et al. "A model for evidence accumulation in the lexical decision task". In: *Cognitive Psychology* (2004). DOI: 10.1016/j.cogpsych.2003.08.001.

[415] Yuan Sophie Liu, Philip Holmes, and Jonathan D Cohen. "A Neural Network Model of the Eriksen Task: Reduction, Analysis, and Data Fitting". In: *Neural Computation* 20.2 (Feb. 2008), pp. 345–373. DOI: 10.1162/neco.2007.08-06-313.

[416] Milica Milosavljevic, Jonathan Malmaud, and Alexander Huth. "The Drift Diffusion Model can account for the accuracy and reaction time of value-based choices under high and low time pressure". In: *October* 5.6 (2010), pp. 437–449. DOI: 10.2139/ssrn.1901533.

[417] Hea-Jung Kim. "Moments of truncated Student- distribution". In: *Journal of the Korean Statistical Society* 37.1 (Mar. 2008), pp. 81–87. DOI: 10.1016/j.jkss.2007.06.001.

[418] K Wunderlich, A Rangel, and J P O'Doherty. "Neural computations underlying action-based decision making in the human brain". In: *Proceedings of the National Academy of Sciences* (2009). DOI: 10.1073/pnas.0901077106.

[419] Antonio Rangel and Todd Hare. "Neural computations associated with goal-directed choice." In: *Current opinion in neurobiology* 20.2 (Apr. 2010), pp. 262–270. DOI: 10.1016/j.conb.2010.03.001.

[420] Aaron M Bornstein and Kenneth A Norman. "Reinstated episodic context guides sampling-based decisions for reward". In: *Nature Neuroscience* 20.7 (2017), pp. 997–1003. DOI: 10.1038/nn.4573.

[421] Marius Usher, Anat Elhalal, and James L McClelland. "The neurodynamics of choice, value-based decisions, and preference reversal". In: *The Probabilistic Mind: Prospects for Bayesian cognitive science* (2012), pp. 277–300. DOI: 10.1093/acprof:oso/9780199216093.003.0013.

[422] Alec Solway and Matthew M Botvinick. "Goal-directed decision making as probabilistic inference: A computational framework and potential neural correlates". In: *Psychological Review* 119.1 (2012), pp. 120–154. DOI: 10.1037/a0026435.

[423] Alec Solway and Matthew M Botvinick. "Evidence integration in model-based tree search". In: *Proceedings of the National Academy of Sciences* 112.37 (2015), pp. 11708–11713. DOI: 10.1073/pnas.1505483112.

[424] Jeff A. Beeler et al. "Tonic Dopamine Modulates Exploitation of Reward Learning". In: *Frontiers in Behavioral Neuroscience* (2010). DOI: 10.3389/fnbeh.2010.00170.

[425] Andrew S Kayser et al. "Dopamine, Locus of Control and the Exploration-Exploitation Tradeoff". In: *Neuropsychopharmacology* 40.2 (Jan. 2015), pp. 454–462. DOI: 10.1038/npp.2014.193.

[426] Richard P. Heitz and Jeffrey D. Schall. "Neural Mechanisms of Speed-Accuracy Tradeoff". In: *Neuron* 76.3 (Nov. 2012), pp. 616–628. DOI: 10.1016/j.neuron.2012.08.030.

[427] Karl Friston et al. "The anatomy of choice: active inference and agency". In: *Frontiers in Human Neuroscience* 7.September (2013), p. 598. DOI: 10.3389/fnhum.2013.00598.

[428] Karl Friston et al. "The anatomy of choice: dopamine and decision-making". In: *Phil. Trans. R. Soc. B* 369.1655 (2014), p. 20130481. DOI: 10.1098/rstb.2013.0481.

[429] Jerome R Busemeyer and James T Townsend. "Decision field theory: A dynamic-cognitive approach to decision making in an uncertain environment". In: *Psychological Review* 100.3 (1993), pp. 432–459. DOI: 10.1037/0033-295X.100.3.432.

[430] Peter I Frazier and Angela J Yu. "Sequential hypothesis testing under stochastic deadlines". In: *Advances in Neural Information Processing Systems* (2008), pp. 1–8. DOI: 10.1.1.304.7658.

[431] Roger Ratcliff and Michael J Frank. "Reinforcement-Based Decision Making in Corticostriatal Circuits: Mutual Constraints by Neurocomputational and Diffusion Models". In: *Neural Computation* 24 (2012), pp. 1186–1229. DOI: 10.1162/NECO{_}a{_}00270.

[432] G E Hawkins et al. "Revisiting the Evidence for Collapsing Boundaries and Urgency Signals in Perceptual Decision-Making". In: *Journal of Neuroscience* 35.6 (2015), pp. 2476–2484. DOI: 10.1523/JNEUROSCI.2410-14.2015.

[433] Chelsea Voskuilen, Roger Ratcliff, and Philip L. Smith. "Comparing fixed and collapsing boundary versions of the diffusion model". In: *Journal of Mathematical Psychology* (2016). DOI: 10.1016/j.jmp.2016.04.008.

[434] S Gluth, J Rieskamp, and C Buchel. "Deciding When to Decide: Time-Variant Sequential Sampling Models Explain the Emergence of Value-Based Decisions in the Human Brain". In: *Journal of Neuroscience* 32.31 (Aug. 2012), pp. 10686–10698. DOI: 10.1523/JNEUROSCI.0727-12.2012.

[435] Roger Ratcliff and Philip L Smith. "A comparison of sequential sampling models for two-choice reaction time." In: *Psychological review* 111.2 (Apr. 2004), pp. 333–367. DOI: 10.1037/0033-295X.111.2.333.

[436] P L Smith and T Van Zandt. "Time-dependent Poisson counter models of response latency in simple judgment." In: *The British journal of mathematical and statistical psychology* 53 (Pt 2) (Nov. 2000), pp. 293–315. URL: http://www.ncbi.nlm.nih.gov/pubmed/11109709.

[437] Thomas V Wiecki and Michael J Frank. "A computational model of inhibitory control in frontal cortex and basal ganglia." In: *Psychological review* 120.2 (Apr. 2013), pp. 329–355. DOI: 10.1037/a0031542.

[438] Roger Ratcliff et al. "Diffusion Decision Model: Current Issues and History". In: *Trends in Cognitive Sciences* 20.4 (2016), pp. 260–281. DOI: 10.1016/j.tics.2016.01.007.

[439] Francis Tuerlinckx. "The efficient computation of the cumulative distribution and probability density functions in the diffusion model." In: *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc* 36.4 (2004), pp. 702–716. DOI: Doi10.3758/Bf03206552.

[440] Roger Ratcliff. "Parameter variability and distributional assumptions in the diffusion model." In: *Psychological Review* 120.1 (2013), pp. 281–292. DOI: 10.1037/a0030775.

[441] Thomas V Wiecki, Imri Sofer, and Michael J Frank. "HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python." In: *Frontiers in neuroinformatics* 7.August (2013), p. 14. DOI: 10.3389/fninf.2013.00014.

[442] Michael D Nunez, Joachim Vandekerckhove, and Ramesh Srinivasan. "How attention influences perceptual decision making: Single-trial EEG correlates of drift-diffusion model parameters". In: *Journal of Mathematical Psychology* 76 (2017), pp. 117–130. DOI: 10.1016/j.jmp.2016.03.003.

[443] Harriet R Brown et al. "Crowdsourcing for cognitive science - the utility of smartphones." In: *PloS one* 9.7 (Jan. 2014), e100662. DOI: 10.1371/journal.pone.0100662.

[444] Fiona McNab et al. "Age-related changes in working memory and the ability to ignore distraction". In: *Proceedings of the National Academy of Sciences* 112.20 (2015), pp. 6515–6518. DOI: 10.1073/pnas.1504162112.

[445] Robb B B Rutledge et al. "Risk Taking for Potential Reward Decreases across the Lifespan". In: *Current Biology* 26.12 (2016), pp. 1634–1639. DOI: 10.1016/j.cub.2016.05.017.

[446] Diego Fernandez Slezak, Mariano Sigman, and Guillermo A Cecchi. "An entropic barriers diffusion theory of decision-making in multiple alternative tasks". In: *PLOS Computational Biology* 14.3 (Mar. 2018). Ed. by Samuel J Gershman, e1005961. DOI: 10.1371/journal.pcbi.1005961.

[447] Mariano Sigman et al. "Response time distributions in rapid chess: A large-scale decision making experiment". In: *Frontiers in Neuroscience* 4.OCT (2010), pp. 1–12. DOI: 10.3389/fnins.2010.00060.

[448] Rachit Dubey et al. "Investigating Human Priors for Playing Video Games". In: (2018), pp. 1–11. DOI: arXiv:1802.10217v1.

[449] Samuel J Gershman et al. "Plans, habits, and theory of mind". In: *PLoS ONE* 11.9 (2016), pp. 1–24. DOI: 10.1371/journal.pone.0162246.

[450] Geoffrey Miller. "The Smartphone Psychology Manifesto". In: *Perspectives on Psychological Science* (2012). DOI: 10.1177/1745691612441215.

[451] Tal Yarkoni. "Psychoinformatics". In: *Current Directions in Psychological Science* 21.6 (Dec. 2012), pp. 391–397. DOI: 10.1177/0963721412457362.

[452] Roger Ratcliff. "Measuring psychometric functions with the diffusion model." In: *Journal of Experimental Psychology: Human Perception and Performance* 40.2 (2014), pp. 870–888. DOI: 10.1037/a0034954.

[453] Andreas Voss, Jochen Voss, and Veronika Lerche. "Assessing cognitive processes with diffusion model analyses: A tutorial based on fast-dm-30". In: *Frontiers in Psychology* 6.MAR (Mar. 2015), pp. 1–14. DOI: 10.3389/fpsyg.2015.00336.

[454] Andrew Heathcote, Scott Brown, and D J K Mewhort. "Quantile maximum likelihood estimation of response time distributions". In: *Psychonomic Bulletin & Review* 9.2 (2002), pp. 394–401. DOI: 10.3758/BF03196299.

[455] Scott Brown and Andrew Heathcote. "QMLE: Fast, robust, and efficient estimation of distribution functions based on quantiles". In: *Behavior Research Methods, Instruments, & Computers* 35.4 (Nov. 2003), pp. 485–492. DOI: 10.3758/BF03195527.

[456] Joachim Vandekerckhove and Francis Tuerlinckx. "Fitting the Ratcliff diffusion model to experimental data." In: *Psychonomic bulletin & review* 14.6 (2007), pp. 1011–1026. DOI: 10.3758/BF03193087.

[457] Joachim Vandekerckhove and Francis Tuerlinckx. "Diffusion model analysis with MATLAB: A DMAT primer". In: *Behavior Research Methods* 40.1 (Feb. 2008), pp. 61–72. DOI: 10.3758/BRM.40.1.61.

[458] Joachim Vandekerckhove, Francis Tuerlinckx, and M Lee. "A Bayesian approach to diffusion process models of decision-making". In: *Proceedings of the 30th annual conference of the cognitive science society* (2008), pp. 1429–1434. URL: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+Bayesian+Approach+to+Diffusion+Process+Models+of+Decision-Making#0.

[459] Joachim Vandekerckhove, Francis Tuerlinckx, and Michael D Lee. "Hierarchical diffusion models for two-choice response times." In: *Psychological methods* 16.1 (2011), pp. 44–62. DOI: 10.1037/a0021765.

[460] Andreas Voss and Jochen Voss. "Fast-dm: A free program for efficient diffusion model analysis". In: *Behavior Research Methods* (2007). DOI: 10.3758/BF03192967.

[461] Angela J Yu and Jonathan D Cohen. "Sequential effects: Superstition or rational behavior?" In: *Advances in Neural Information Processing Systems* 21 (2009), pp. 1873–1880. URL: http://www.pni.princeton.edu/ncc/publications/2009/YuNIPS09.pdf.

[462] Juan Gao et al. "Sequential Effects in Two-Choice Reaction Time Tasks: Decomposition and Synthesis of Mechanisms". In: *Neural Computation* 21.9 (Sept. 2009), pp. 2407–2436. DOI: 10.1162/neco.2009.09-08-866.

[463] M H Wilder, M Jones, and M C Mozer. "Sequential effects reflect parallel learning of multiple environmental regularities." In: *Advances in Neural Information Processing Systems* 22 (2009), pp. 2053–2061.

[464] Joshua Benjamin Miller and Adam Sanjurjo. "Surprised by the Gambler's and Hot Hand Fallacies? A Truth in the Law of Small Numbers". In: *SSRN Electronic Journal* (2015), pp. 1–44. DOI: 10.2139/ssrn.2627354.

[465] Gui Xue et al. "Lateral prefrontal cortex contributes to maladaptive decisions." In: *Proceedings of the National Academy of Sciences of the United States of America* 109.12 (Mar. 2012), pp. 4401–4406. DOI: 10.1073/pnas.1111927109.

[466] Robin Shao, Delin Sun, and Tatia M C Lee. "The interaction of perceived control and Gambler's fallacy in risky decision making: An fMRI study". In: *Human Brain Mapping* 37.3 (2016), pp. 1218–1234. DOI: 10.1002/hbm.23098.

[467] Wim Notebaert et al. "Post-error slowing: An orienting account". In: *Cognition* 111.2 (May 2009), pp. 275–279. DOI: 10.1016/j.cognition.2009.02.002.

[468] Trisha Van Zandt. "How to fit a response time distribution". In: *Psychonomic Bulletin & Review* 7.3 (Sept. 2000), pp. 424–465. DOI: 10.3758/BF03214357.

[469] Eric-Jan Wagenmakers, Raoul P P P Grasman, and Peter C M Molenaar. "On the relation between the mean and the variance of a diffusion model response time distribution". In: *Journal of Mathematical Psychology* 49 (2005), pp. 195–204. DOI: 10.1016/j.jmp.2005.02.003.

[470] Dominik Wabersich and Joachim Vandekerckhove. "Extending JAGS: A tutorial on adding custom distributions to JAGS (with a diffusion model example)". In: *Behavior Research Methods* 46.1 (2014), pp. 15–28. DOI: 10.3758/s13428-013-0369-3.

[471] Andrew Gelman, Daniel Lee, and Jiqiang Guo. "Stan: A Probabilistic Programming Language for Bayesian Inference and Optimization". In: *Journal of Educational and Behavioral Statistics* 40.5 (2015), pp. 530–543. DOI: 10.3102/1076998615606113.

[472] Peter R Murphy, Joachim Vandekerckhove, and Sander Nieuwenhuis. "Pupil-Linked Arousal Determines Variability in Perceptual Decision Making". In: *PLoS Computational Biology* 10.9 (2014). DOI: 10.1371/journal.pcbi.1003854.

[473] Rafael Polanía et al. "The precision of value-based choices depends causally on fronto-parietal phase coupling". In: *Nature Communications* 6 (2015). DOI: 10.1038/ncomms9090.

[474] Samuel R Mathias. "Unified analysis of accuracy and reaction times via models of decision making". In: vol. 050001. 2016. 2016, p. 50001. DOI: 10.1121/2.0000219.

[475] Matt Jones et al. “Supplemental Material for Sequential Effects in Response Time Reveal Learning Mechanisms and Event Representations”. In: *Psychological review* 120.3 (2013), pp. 628–666. DOI: 10.1037/a0033180.

[476] Peter Dayan. “Helmholtz machines and wake-sleep learning”. In: *Handbook of Brain Theory and Neural Network. MIT ...* 44.0 (2000). URL: http://www.gatsby.ucl.ac.uk/%7E~%7EDayan/papers/d2000a.pdf.

[477] Otto Fabius and Joost R van Amersfoort. “Variational Recurrent Auto-Encoders”. In: 2013 (Dec. 2014), pp. 1–5. URL: http://arxiv.org/abs/1412.6581.

[478] Junyoung Chung et al. “A Recurrent Latent Variable Model for Sequential Data”. In: (June 2015), pp. 1–9. URL: http://arxiv.org/abs/1506.02216.

[479] Chris J Maddison et al. “Filtering Variational Objectives”. In: Nips (2017), pp. 1–11. URL: http://arxiv.org/abs/1705.09279.

[480] Tuan Anh Le et al. “Auto-Encoding Sequential Monte Carlo”. In: (2017). URL: http://arxiv.org/abs/1705.10306.

[481] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. DOI: 10.1214/12-AOS1000.

[482] Diederik P Kingma, Tim Salimans, and Max Welling. “Variational Dropout and the Local Reparameterization Trick”. In: Mcmc (2015), pp. 1–14. URL: http://arxiv.org/abs/1506.02557.

[483] Philip L Smith, Roger Ratcliff, and Gail McKoon. “The diffusion model is not a deterministic growth model: Comment on Jones and Dzhafarov (2014).” In: *Psychological Review* 121.4 (2014), pp. 679–688. DOI: 10.1037/a0037667.

[484] Lucien Le Cam. *Asymptotic Methods in Statistical Decision Theory*. Springer Series in Statistics. New York, NY: Springer New York, 1986. DOI: 10.1007/978-1-4612-4946-7.

[485] Sungjin Ahn, Anoop Korattikara, and Max Welling. “Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring”. In: *International Conference on Machine Learning* cs.LG (2012). URL: http://arxiv.org/abs/1206.6380v1%5Cnpapers2://publication/uuid/FE42FF46-CA89-45F1-9647-50AC1372E1BD.

[486] Stephan Mandt, Matthew D Hoffman, and David M Blei. “Stochastic Gradient Descent as Approximate Bayesian Inference”. In: *Mathematical Biosciences and Engineering* 13.3 (Apr. 2017), pp. 613–629. URL: http://www.aimsciences.org/journals/displayArticlesnew.jsp?paperID=12221%20http://arxiv.org/abs/1704.04289.

[487] Alexander Buchholz et al. “Quasi-Monte Carlo Variational Inference”. In: (2018).

[488] Tommi S Jaakkola and Michael I Jordan. "A variational approach to Bayesian logistic regression models and their extensions". In: *Aistats* AUGUST 2001 (1996). DOI: 10.1.1.49.5049.

[489] David M Blei. "Variational Inference". In: *Cs.Princeton.Edu* (2002), pp. 1–12. DOI: 10.16373/j.cnki.ahr.150049.

[490] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational Inference: A Review for Statisticians". In: *arXiv* (2016), pp. 1–33. URL: http://arxiv.org/abs/1601.00670.

[491] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *Journal of Machine Learning Research* 12 (2011), pp. 2121–2159. DOI: 10.1109/CDC.2012.6426698.

[492] Matthew D Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: (Dec. 2012). URL: http://arxiv.org/abs/1212.5701.

[493] Diederik P Kingma and Jimmy Lei Ba. "Adam: a Method for Stochastic Optimization". In: *International Conference on Learning Representations 2015* (2015), pp. 1–15.

[494] Yarin Gal and Zoubin Ghahramani. "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks". In: International Series on Computational Intelligence 19994575 (Dec. 2015). Ed. by L Medsker and L Jain. DOI: 10.1201/9781420049176.

[495] P E Kloeden and E Platen. *Numerical solution of stochastic differential equations*. Vol. 23. 3. 1992, pp. xxxvi+632. DOI: 10.2977/prims/1166642153.

[496] Stijn Verdonck, Kristof Meers, and Francis Tuerlinckx. "Efficient simulation of diffusion-based choice RT models on CPU and GPU". In: *Behavior Research Methods* (2016). DOI: 10.3758/s13428-015-0569-0.

[497] Yuhuang Hu et al. "Overcoming the vanishing gradient problem in plain recurrent networks". In: Section 2 (2018), pp. 1–20. URL: http://arxiv.org/abs/1801.06105.

[498] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: (2014). DOI: 10.3115/v1/D14-1179.

[499] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[500] Richard H R Hahnloser et al. “Erratum: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit”. In: *Nature* 405.6789 (June 2000), pp. 947–951. DOI: 10.1038/35016072.

[501] Vinod Nair and Geoffrey E Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *Proceedings of the 27th International Conference on Machine Learning* 3 (2010), pp. 807–814. DOI: 10.1.1.165.6419.

[502] Jeff Bezanson et al. “Julia: A Fresh Approach to Numerical Computing”. In: *SIAM Review* 59.1 (Jan. 2017), pp. 65–98. DOI: 10.1137/141000671.

[503] Vincent Moens. “The Hierarchical Adaptive Forgetting Variational Filter”. In: *ICML proceedings* (May 2018). URL: http://arxiv.org/abs/1805.05703.

[504] John Langford, Alexander J Smola, and Martin Zinkevich. “Slow Learners are Fast”. In: *Advances in Neural Information Processing Systems - NIPS'09* 1.2099 (2009), pp. 2331–2339. DOI: 10.1016/0140-6736(91)90756-F.

[505] Tim Salimans, Diederik P Kingma, and Max Welling. “Markov Chain Monte Carlo and Variational Inference : Bridging the Gap”. In: *International Conference on Machine Learning* (2015).

[506] Mark Stokes and Eelke Spaak. *The Importance of Single-Trial Analyses in Cognitive Neuroscience*. 2016. DOI: 10.1016/j.tics.2016.05.008.

[507] Yu-Ting Huang et al. “Different effects of dopaminergic medication on perceptual decision-making in Parkinson's disease as a function of task difficulty and speed–accuracy instructions”. In: *Neuropsychologia* 75 (Aug. 2015), pp. 577–587. DOI: 10.1016/j.neuropsychologia.2015.07.012.

[508] James F Cavanagh et al. “Frontal theta links prediction errors to behavioral adaptation in reinforcement learning”. In: *NeuroImage* 49.4 (2010), pp. 3198–3209. DOI: 10.1016/j.neuroimage.2009.11.080.

[509] James F Cavanagh and Michael J Frank. “Frontal theta as a mechanism for cognitive control”. In: *Trends in Cognitive Sciences* 18.8 (Aug. 2014), pp. 414–421. DOI: 10.1016/j.tics.2014.04.012.

[510] Vaibhav Srivastava et al. “A martingale analysis of first passage times of time-dependent Wiener diffusion models”. In: *Journal of Mathematical Psychology* 77 (2017), pp. 94–110. DOI: 10.1016/j.jmp.2016.10.001.

[511] David Cox and Hilton David Miller. *The Theory of Stochastic Processes*. 1965, p. 408. URL: http://books.google.com/books?hl=en&lr=&id=76QOAAAAQAAJ&pgis=1.

[512] Mario Lefebvre. "First hitting place distributions for the Ornstein-Uhlenbeck process". In: *Statistics & Probability Letters* 34.3 (June 1997), pp. 309–312. DOI: 10.1016/S0167-7152(96)00195-2.

[513] Yingzhen Li and Stephan Mandt. "Disentangled Sequential Autoencoder". In: *Proceedings of the 35th international conference on Machine learning - ICML '18*. Mar. 2018. URL: http://arxiv.org/abs/1803.02991.

[514] Olivier Cappé, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models*. 2006, p. 653. DOI: 10.1198/tech.2006.s440.

[515] Herbert Robbins. "The Empirical Bayes Approach to Statistical Decision Problems". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 1–20. DOI: 10.1214/aoms/1177703729.

[516] George Marsaglia. "Generating a Variable from the Tail of the Normal Distribution". In: *Technometrics* 6.1 (Feb. 1964), p. 101. DOI: 10.2307/1266749.

[517] Z I Botev. "The normal law under linear restrictions: simulation and estimation via minimax tilting". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.1 (Jan. 2016), pp. 125–148. DOI: 10.1111/rssb.12162.

[518] R Kulhavy and M Karny. "Tracking of slowly varying parameters by directional forgetting". In: *Preprints 9ih IFAC Congress* 10 (1984), pp. 178–183. DOI: 10.1016/S1474-6670(17)61051-6.

[519] R Kulhavý and F J Kraus. "On Duality of Exponential and Linear Forgetting". In: *IFAC Proceedings Volumes* 29.1 (June 1996), pp. 5340–5345. DOI: 10.1016/S1474-6670(17)58530-4.

[520] V Smidl. "The Variational Bayes Approach in Signal Processing". PhD thesis. 2004. URL: http://staff.utia.cz/smidl/Public/Thesis-final.pdf.

[521] Vaclav Smidl and Fredrik Gustafsson. "Bayesian estimation of forgetting factor in adaptive filtering and change detection". In: *2012 IEEE Statistical Signal Processing Workshop (SSP)*. 1. IEEE, Aug. 2012, pp. 197–200. DOI: 10.1109/SSP.2012.6319658.

[522] S Azizi and A Quinn. "A data-driven forgetting factor for stabilized forgetting in approximate Bayesian filtering". In: *2015 26th Irish Signals and Systems Conference (ISSC)*. Vol. 11855. IEEE, June 2015, pp. 1–6. DOI: 10.1109/ISSC.2015.7163747.

[523] Andres Masegosa et al. "Bayesian Models of Data Streams with Hierarchical Power Priors". In: *International Conference on Machine Learning (ICM)* 70 (2017), pp. 2334–2343. URL: http://proceedings.mlr.press/v70/masegosa17a.html.

[524] Stephan Mandt et al. "Variational Tempering". In: 41 (2014). URL: http://arxiv.org/abs/1411.1810.

[525] T S Jaakkola and M I Jordan. "Bayesian parameter estimation via variational methods". In: *Statistics And Computing* 10.1 (2000), pp. 25–37. DOI: 10.1023/A:1008932416310.

[526] Persi Diaconis and Donald Ylvisaker. "Conjugate Priors for Exponential Families". In: *The Annals of Statistics* 7.2 (Mar. 1979), pp. 269–281. DOI: 10.1214/aos/1176344611.

[527] Christoph Mathys. "How could we get nosology from computation?" In: *Computational Psychiatry: New Perspectives on Mental Illness* 20 (2016), pp. 121–138. URL: https://mitpress.mit.edu/books/computational-psychiatry.

[528] R KULHAVÝ and M B ZARROP. "On a general concept of forgetting". In: *International Journal of Control* 58.4 (Oct. 1993), pp. 905–924. DOI: 10.1080/00207179308923034.

[529] Paul Zarchan and Howard Musoff, eds. *Fundamentals of Kalman Filtering: A Practical Approach, Fourth Edition*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., Jan. 2015. DOI: 10.2514/4.102776.

[530] a Doucet, N De Freitas, and N Gordon. "Sequential Monte Carlo Methods in Practice". In: *Springer New York* (2001), pp. 178–195. DOI: 10.1198/tech.2003.s23.

[531] V Smidl and Anthony Quinn. "Variational Bayesian Filtering". In: *IEEE Transactions on Signal Processing* 56.10 (Oct. 2008), pp. 5020–5030. DOI: 10.1109/TSP.2008.928969.

[532] Emre Özkan et al. "Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters". In: *Automatica* 49.6 (June 2013), pp. 1566–1575. DOI: 10.1016/j.automatica.2013.02.046.

[533] Thijs Van De Laar et al. "Variational Stabilized Linear Forgetting in State-Space Models". In: Section II (2017), pp. 848–852.

[534] Miroslav Kárný. "Approximate Bayesian recursive estimation". In: *Information Sciences* 285.1 (2014), pp. 100–111. DOI: 10.1016/j.ins.2014.01.048.

[535] Kamil Dedecius and Radek Hofman. "Autoregressive model with partial forgetting within Rao-Blackwellized particle filter". In: *Communications in Statistics: Simulation and Computation* 41.5 (2012), pp. 582–589. DOI: 10.1080/03610918.2011.598992.

[536] V Smidl and A Quinn. "Mixture-based extension of the AR model and its recursive Bayesian identification". In: *IEEE Transactions on Signal Processing* 53.9 (Sept. 2005), pp. 3530–3542. DOI: 10.1109/TSP.2005.853103.

[537] Christoph Mathys. "A Bayesian foundation for individual learning under uncertainty". In: *Frontiers in Human Neuroscience* 5.May (2011), pp. 1–20. DOI: 10.3389/fnhum.2011.00039.

[538] Christoph D Mathys et al. "Uncertainty in perception and the Hierarchical Gaussian Filter". In: *Frontiers in Human Neuroscience* 8.November (Nov. 2014), pp. 1–24. DOI: 10.3389/fnhum.2014.00825.

[539] Jarrett Revels, Miles Lubin, and Theodore Papamarkou. "Forward-Mode Automatic Differentiation in Julia". In: April (2016), p. 4. URL: http://arxiv.org/abs/1607.07892.

[540] Dustin Tran, Panos Toulis, and Edoardo M Airoldi. "Stochastic gradient descent methods for estimation with large data sets". In: VV.October (Sept. 2015). URL: http://arxiv.org/abs/1509.06459.

[541] Vincent Moens and Alexandre Zenon. "Recurrent Auto-Encoding Drift Diffusion Model". In: *bioRxiv* (2018). DOI: 10.1101/220517.

[542] K H Britten et al. "The analysis of visual motion: a comparison of neuronal and psychophysical performance." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 12.12 (Dec. 1992), pp. 4745–4765. DOI: 10.1.1.123.9899.

[543] Atilim Gunes Baydin et al. "Online Learning Rate Adaptation with Hypergradient Descent". In: *International Conference on Learning Representations 2018*. 2015. 2017, pp. 1–11. URL: http://arxiv.org/abs/1703.04782.

[544] A Dickinson et al. "Motivational control after extended instrumental training". In: *Animal Learning & Behavior* 23.2 (June 1995), pp. 197–206. DOI: 10.3758/BF03199935.

[545] Carol a Seger and Brian J Spiering. "A critical review of habit learning and the Basal Ganglia." In: *Frontiers in systems neuroscience* 5.August (Jan. 2011), p. 66. DOI: 10.3389/fnsys.2011.00066.

[546] Laurel S Morris et al. "Fronto-striatal organization: Defining functional and microstructural substrates of behavioural flexibility". In: *Cortex* 74 (2016), pp. 118–133. DOI: 10.1016/j.cortex.2015.11.004.

[547] Colin M MacLeod. "Half a century of research on the Stroop effect: An integrative review." In: *Psychological Bulletin* 109.2 (1991), pp. 163–203. DOI: 10.1037/0033-2909.109.2.163.

[548] Andy Clark. "Whatever next? Predictive brains, situated agents, and the future of cognitive science." In: *The Behavioral and brain sciences* 36.3 (June 2013), pp. 181–204. DOI: 10.1017/S0140525X12000477.

[549] Guido Hesselmann et al. "Predictive coding or evidence accumulation? False inference and neuronal fluctuations". In: *PLoS ONE* 5.3 (2010), pp. 1–5. DOI: 10.1371/journal.pone.0009926.

[550] Lisa Mayrhauser et al. "Neural repetition suppression: evidence for perceptual expectation in object-selective regions". In: *Frontiers in Human Neuroscience* 8.April (2014), pp. 1–8. DOI: 10.3389/fnhum.2014.00225.

[551] Roberto Limongi, Angélica M Silva, and Begoña Góngora-Costa. "Temporal prediction errors modulate task-switching performance". In: *Frontiers in Psychology* 6.August (2015), pp. 1–10. DOI: 10.3389/fpsyg.2015.01185.

[552] Jan Kneissler et al. "Simultaneous learning and filtering without delusions: a Bayes-optimal combination of Predictive Inference and Adaptive Filtering". In: *Frontiers in Computational Neuroscience* 9.April (2015), pp. 1–12. DOI: 10.3389/fncom.2015.00047.

[553] Sandra Iglesias et al. "Hierarchical Prediction Errors in Midbrain and Basal Forebrain during Sensory Learning". In: *Neuron* 80.2 (2013), pp. 519–530. DOI: 10.1016/j.neuron.2013.09.009.

[554] Simone Vossel et al. "Cholinergic stimulation enhances Bayesian belief updating in the deployment of spatial attention." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 34.47 (2014), pp. 15735–15742. DOI: 10.1523/JNEUROSCI.0091-14.2014.

[555] Tobias U Hauser et al. "Role of the Medial Prefrontal Cortex in Impaired Decision Making in Juvenile Attention-Deficit/Hyperactivity Disorder". In: *JAMA Psychiatry* 71.10 (Oct. 2014), p. 1165. DOI: 10.1001/jamapsychiatry.2014.1093.

[556] Andreea O Diaconescu et al. "Inferring on the Intentions of Others by Hierarchical Bayesian Learning". In: *PLoS Computational Biology* 10.9 (2014). DOI: 10.1371/journal.pcbi.1003810.

[557] Inti A Brazil et al. "Representational uncertainty in the brain during threat conditioning and the link with psychopathic traits". In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging* 14 (2017), pp. 1–7. DOI: 10.1016/j.bpsc.2017.04.005.

[558] John Paisley, David Blei, and Michael Jordan. "Variational Bayesian Inference with Stochastic Search". In: *Icml* 2000 (2012), pp. 1367–1374. URL: http://icml.cc/2012/papers/687.pdf.

[559] Anthony Quinn V. Smidl. "Bayesian estimation of non-stationary AR model parameters via an unknown forgetting factor". In: *3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th Digital Signal Processing Workshop, 2004.* 6. IEEE, 2004, pp. 221–225. DOI: 10.1109/DSPWS.2004.1437946.

[560] F. Gregory Ashby, John M. Ennis, and Brian J. Spiering. "A neurobiological theory of automaticity in perceptual categorization." In: *Psychological Review* 114.3 (2007), pp. 632–656. DOI: 10.1037/0033-295x.114.3.632.

[561] Léon Bottou et al. "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising". In: *Journal of Machine Learning Research* 14 (2013), pp. 3207–3260. DOI: 10.1145/2740908.2742564.

[562] Jakob Foerster et al. "Counterfactual Multi-Agent Policy Gradients". In: *Arxiv* (2017), pp. 1–12. URL: http://arxiv.org/abs/1705.08926.

[563] Carolin Lawrence, Artem Sokolov, and Stefan Riezler. "Counterfactual Learning from Bandit Feedback under Deterministic Logging: A Case Study in Statistical Machine Translation". In: (2017). URL: http://arxiv.org/abs/1707.09118.

[564] Walter Mischel, Ebbe B Ebbesen, and Antonette Raskoff Zeiss. "Cognitive and attentional mechanisms in delay of gratification." In: *Journal of Personality and Social Psychology* 21.2 (1972), pp. 204–218. DOI: 10.1037/h0032198.

[565] Jeffrey N Weatherly. "On several factors that control rates of discounting". In: *Behavioural Processes* 104 (2014), pp. 84–90. DOI: 10.1016/j.beproc.2014.01.020.

[566] Giles W Story et al. "Does temporal discounting explain unhealthy behavior? A systematic review and reinforcement learning perspective." In: *Frontiers in behavioral neuroscience* 8.March (Jan. 2014), p. 76. DOI: 10.3389/fnbeh.2014.00076.

[567] Samuel M McClure et al. "Separate neural systems value immediate and delayed monetary rewards." In: *Science (New York, N.Y.)* 306.5695 (Oct. 2004), pp. 503–507. DOI: 10.1126/science.1100907.

[568] Wolfram Schultz. "Updating dopamine reward signals." In: *Current opinion in neurobiology* 23.2 (Apr. 2013), pp. 229–238. DOI: 10.1016/j.conb.2012.11.012.

[569] Maria A Bermudez and Wolfram Schultz. "Timing in reward and decision processes." In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 369.1637 (2014), p. 20120468. DOI: 10.1098/rstb.2012.0468.

[570] Taiki Takahashi. "Loss of self-control in intertemporal choice may be attributable to logarithmic time-perception". In: *Medical Hypotheses* 65.4 (Jan. 2005), pp. 691–693. DOI: 10.1016/j.mehy.2005.04.040.

[571] Benjamin T Vincent. "Hierarchical Bayesian estimation and hypothesis testing for delay discounting tasks". In: *Behavior Research Methods* (2015). DOI: 10.3758/s13428-015-0672-2.

[572] Zeb Kurth-Nelson, Warren Bickel, and A David Redish. "A theoretical account of cognitive effects in delay discounting". In: *European Journal of Neuroscience* 35.7 (2012), pp. 1052–1064. DOI: 10.1111/j.1460-9568.2012.08058.x.

[573] G Viejo et al. "Modelling choice and reaction time during instrumental learning through the coordination of adaptive working memory and reinforcement learning". In: *Fourth Symposium on Biology of Decision - Making (SBDM 2014)* 9.August (2014). DOI: 10.3389/fnbeh.2015.00225.

[574] Rowan Mcallister and Karolina Dziugaite. "Bayesian Reinforcement Learning". In: 35.March (2013), pp. 1–21.

[575] Philip L Smith. "Stochastic Dynamic Models of Response Time and Accuracy: A Foundational Primer". In: *Journal of Mathematical Psychology* 44.3 (2000), pp. 408–463. DOI: 10.1006/jmps.1999.1260.

[576] Gregor Rainer, S Chenchal Rao, and Earl K Miller. "Prospective Coding for Objects in Primate Prefrontal Cortex". In: *The Journal of Neuroscience* 19.13 (July 1999), pp. 5493–5505. DOI: 10.1523/JNEUROSCI.19-13-05493.1999.

[577] J. Reutimann et al. "Climbing Neuronal Activity as an Event-Based Cortical Representation of Time". In: *Journal of Neuroscience* 24.13 (Mar. 2004), pp. 3295–3303. DOI: 10.1523/JNEUROSCI.4098-03.2004.

[578] Johanni Brea et al. "Prospective Coding by Spiking Neurons". In: *PLoS Computational Biology* (2016). DOI: 10.1371/journal.pcbi.1005003.

[579] Anna Jafarpour et al. "Human hippocampal pre-activation predicts behavior". In: *Scientific Reports* 7.1 (Dec. 2017), p. 5959. DOI: 10.1038/s41598-017-06477-5.

[580] Jian Li and Nathaniel D Daw. "Signals in human striatum are appropriate for policy update rather than value prediction." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 31.14 (Apr. 2011), pp. 5504–5511. DOI: 10.1523/JNEUROSCI.6316-10.2011.

[581] T. H. B. FitzGerald, P. Schwartenbeck, and R. J. Dolan. "Reward-Related Activity in Ventral Striatum Is Action Contingent and Modulated by Behavioral Relevance". In: *Journal of Neuroscience* 34.4 (Jan. 2014), pp. 1271–1279. DOI: 10.1523/JNEUROSCI.4389-13.2014.

[582] Kevin N. Gurney, Mark D. Humphries, and Peter Redgrave. "A New Framework for Cortico-Striatal Plasticity: Behavioural Theory Meets In Vitro Data at the Reinforcement-Action Interface". In: *PLoS Biology* 13.1 (Jan. 2015). Ed. by Peter Dayan, e1002034. DOI: 10.1371/journal.pbio.1002034.

[583] Miriam C. Klein-Flügge et al. "Dissociable Reward and Timing Signals in Human Midbrain and Ventral Striatum". In: *Neuron* (2011). DOI: 10.1016/j.neuron.2011.08.024.

[584] Masa-Aki Sato. "Online Model Selection Based on the Variational Bayes". In: *Neural Comput.* 13.7 (2001), pp. 1649–1681. DOI: 10.1162/089976601750265045.

[585] James Martens. "New insights and perspectives on the natural gradient method". In: (Dec. 2014). URL: http://arxiv.org/abs/1412.1193.

[586] Subhashis Ghosal, Jayanta K Ghosh, and Aad W van der Vaart. "Convergence rates of posterior distributions". In: *The Annals of Statistics* 28.2 (Apr. 2000), pp. 500–531. DOI: 10.1214/aos/1016218228.

[587] Alexandre Zenon, Oleg Solopchuk, and Giovanni Pezzulo. "An information-theoretic perspective on the costs of cognition". In: *bioRxiv* (2018), p. 208280. DOI: 10.1101/208280.

[588] Wendy Wood and Dennis Ruenger. "Psychology of Habit". In: *Annual Review of Psychology* September (2015), pp. 1–26. DOI: 10.1146/annurev-psych-122414-033417.

[589] Ray J Dolan and Peter Dayan. "Goals and habits in the brain." In: *Neuron* 80.2 (Oct. 2013), pp. 312–325. DOI: 10.1016/j.neuron.2013.09.007.

[590] Daniel Kahneman and Shane Frederick. "Representativeness Revisited: Attribute Substitution in Intuitive Judgment". In: *Heuristics and Biases*. Ed. by T Gilovich, D Griffin, and Daniel Kahneman. Cambridge University Press, 2001, pp. 49–81. DOI: 10.1017/CBO9780511808098.004.

[591] Alireza Soltani and Xiao Jing Wang. "Synaptic computation underlying probabilistic inference". In: *Nature Neuroscience* 13.1 (2010), pp. 112–119. DOI: 10.1038/nn.2450.

[592] Jaldert O Rombouts, Sander M Bohte, and Pieter R Roelfsema. "Neurally Plausible Reinforcement Learning of Working Memory Tasks". In: *Nips* (2012), pp. 1–9. DOI: 10.1371/journal.pcbi.1004489.

[593] Nils Kurzawa, Christopher Summerfield, and Rafal Bogacz. "Neural Circuits Trained with Standard Reinforcement Learning Can Accumulate Probabilistic Information during Decision Making". In: *Neural Computation* 29.2 (Feb. 2017), pp. 368–393. DOI: 10.1162/NECO{_}a{_}00917.

[594] Andreas Stuhlmueller, Jessica Taylor, and Noah D Goodman. "Learning Stochastic Inverses". In: *Advances in Neural Information Processing Systems 27 (NIPS 2013)* (2013), pp. 1–9.

[595] Jarno Lintusaari et al. "Fundamentals and recent developments in approximate Bayesian computation". In: *Systematic Biology* 66.1 (2017), e66–e82. DOI: 10.1093/sysbio/syw077.

[596] Kenji Morita and Ayaka Kato. "Striatal dopamine ramping may indicate flexible reinforcement learning with forgetting in the cortico-basal ganglia circuits". In: *Frontiers in Neural Circuits* 8.April (2014), pp. 1–15. DOI: 10.3389/fncir.2014.00036.

[597] Ayaka Kato and Kenji Morita. "Forgetting in Reinforcement Learning Links Sustained Dopamine Signals to Motivation". In: *PLoS Computational Biology* 12.10 (2016), pp. 1–41. DOI: 10.1371/journal.pcbi.1005145.

[598] Joseph T McGuire et al. "Functionally Dissociable Influences on Learning Rate in a Dynamic Environment". In: *Neuron* 84.4 (Nov. 2014), pp. 870–881. DOI: 10.1016/j.neuron.2014.10.013.

[599] Matthew R Nassar et al. "Age differences in learning emerge from an insufficient representation of uncertainty in older adults". In: *Nature Communications* 7.May 2015 (2016), pp. 1–13. DOI: 10.1038/ncomms11609.

[600] Bradley B Doll, Dylan A Simon, and Nathaniel D Daw. "The ubiquity of model-based reinforcement learning." In: *Current opinion in neurobiology* 22.6 (Dec. 2012), pp. 1075–1081. DOI: 10.1016/j.conb.2012.08.003.

[601] Klaus Wunderlich, Peter Smittenaar, and Raymond J Dolan. "Dopamine enhances model-based over model-free choice behavior." In: *Neuron* 75.3 (Aug. 2012), pp. 418–424. DOI: 10.1016/j.neuron.2012.03.042.

[602] A Ross Otto et al. "The curse of planning: dissecting multiple reinforcement-learning systems by taxing the central executive." In: *Psychological science* 24.5 (May 2013), pp. 751–761. DOI: 10.1177/0956797612463080.

[603] Daniel J Schad et al. "Processing speed enhances model-based over model-free reinforcement learning in the presence of high working memory functioning". In: *Frontiers in Psychology* 5.December (Dec. 2014), pp. 1–10. DOI: 10.3389/fpsyg.2014.01450.

[604] Geoffrey Grimmett and Dominic Welsh. *Probability, An Introduction*. second. Oxford University Press, 1986. URL: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0198503687.

[605] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* (1986). DOI: 10.1038/323533a0.

[606] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Psychological Review* (1958). DOI: 10.1037/h0042519.

[607] Kunihiko Fukushima and Sei Miyake. "NEOCOGNITION: SELF-ORGANIZING NETWORK CAPABLE OF POSITION-INVARIANT RECOGNITION OF PATTERNS." In: *NATO Conference Series, (Series) 4: Marine Sciences* (1980).

[608] Richard S. Sutton and Andrew G. Barto. "Toward a modern theory of adaptive networks: Expectation and prediction". In: *Psychological Review* (1981). DOI: 10.1037/0033-295X.88.2.135.

[609] D Ackley, G Hinton, and T Sejnowski. "A learning algorithm for boltzmann machines". In: *Cognitive Science* (1985). DOI: 10.1016/S0364-0213(85)80012-4.

[610] D. Kahneman, P. Slovic, and A. Tversky. "Judgment under uncertainty: heuristics and biases". In: *Science* (1974). DOI: 10.1126/science.185.4157.1124.

[611] Matt Jones and Bradley C. Love. "Bayesian fundamentalism or enlightenment? on the explanatory status and theoretical contributions of bayesian models of cognition". In: *Behavioral and Brain Sciences* (2011). DOI: 10.1017/S0140525X10003134.

[612] Sebastian Claici and Justin Solomon. "Wasserstein Coresets for Lipschitz Costs". In: (). URL: https://arxiv.org/pdf/1805.07412.pdf.

[613] Jonathan H. Huggins, Trevor Campbell, and Tamara Broderick. "Coresets for Scalable Bayesian Logistic Regression". In: *30th Conference on Neural Information Processing Systems (NIPS 2016)*. 2016, pp. 1–22. URL: http://arxiv.org/abs/1605.06423.

[614] Trevor Campbell and Tamara Broderick. "Automated Scalable Bayesian Inference via Hilbert Coresets". In: (2017), pp. 1–36. URL: http://arxiv.org/abs/1710.05053.

[615] Trevor Campbell and Tamara Broderick. "Bayesian Coreset Construction via Greedy Iterative Geodesic Ascent". In: (2018). URL: http://arxiv.org/abs/1802.01737.

[616] Falk Lieder, Thomas L. Griffiths, and Ming Hsu. "Overrepresentation of extreme events in decision making reflects rational use of cognitive resources." In: *Psychological Review* 125.1 (Jan. 2018), pp. 1–32. DOI: 10.1037/rev0000074.

[617] Jai Y. Yu and Loren M. Frank. *Hippocampal-cortical interaction in decision making*. 2015. DOI: 10.1016/j.nlm.2014.02.002.

[618] Quentin J M Huys et al. "Bonsai Trees in Your Head: How the Pavlovian System Sculpts Goal-Directed Choices by Pruning Decision Trees". In: *PLoS Computational Biology* 8.3 (Mar. 2012). Ed. by Laurence T. Maloney, e1002410. DOI: 10.1371/journal.pcbi.1002410.

[619] Kevin Miller et al. "Re-aligning models of habitual and goal-directed decision-making". In: *Goal-directed decision making: Computations and neural circuits. New York: Elsevier*. Academic Press, 2017.

[620] Wouter Kool, Fiery A Cushman, and Samuel J Gershman. "Competition and cooperation between multiple reinforcement learning systems". In: *Goal-directed decision making: Computations and neural circuits. New York: Elsevier*. Academic Press, 2017.