Recurrent Auto-Encoding Drift Diffusion Model

MOENS Vincent[a], ZENON Alexandre[a,b]

[a] *IoNS, Université Catholique de Louvain*
Brussels, Belgium

[b] *INCIA*
Université de Bordeaux, France

Abstract

The Full Drift Diffusion Model (DDM) is challenging to fit to behavioural data. Precision of the fits are usually poor for some if not all parameters, and computationally expensive to obtain. Moreover, inference at the trial level for each and every parameters, threshold included, has so far been considered as impossible. Most approaches rely on strong assumptions about the model structure, such as selection of the parameters that are subject to trial-to-trial variability or prior distribution of these parameters, that usually lack precise mathematical or empirical justifications. The fact that, in most versions of the DDM, the sequence of participants' choices are considered as independent and identically distributed (*i.i.d.*), has been mainly overlooked so far. Our contribution to the field is threefold: first, we introduce Variational Bayes as a method to fit the full DDM. Second, we relax the *i.i.d.* assumption, and propose a data-driven algorithm based on a Recurrent Auto-Encoder, that estimates the local posterior probability of the DDM parameters at each trial based on the sequence of parameters and data preceding the data. Finally, we show 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 and real-world examples, we show that this method outperforms by several order of magnitude the *i.i.d.*-based ones, either Markov Chain Monte Carlo or *i.i.d.*-VB.

Keywords: Drift-Diffusion Model, Importance-Weighted Auto-Encoder, Amortized Inference, Recurrent Neural Network, Particle Filtering, Data-driven behavioral analysis

Email address: `vincent.moens@uclouvain.be` (MOENS Vincent)

1. Introduction

Sequential sampling models (SSM) (1) are models of behaviour that assume that decisions are the result of a noisy accumulation of evidence. The most famous of these models, called the Drift Diffusion Model (DDM), was introduced in the neuroscience field by Roger Ratcliff (2) 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 (3; 2; 4; 5; 6; 7) 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 (8), Extended DDM (9) or Full DDM (10).

While the Full DDM is more realistic (see (11) 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 version 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) (12; 13; 7; 14; 15) or Markov Chain Monte Carlo (16). However, NQ is computationally expensive, which precludes its use for large problems. Markov Chain Monte Carlo (MCMC) (17) on the other hand, can be cheaper, but is still computationally expansive and requires an optimization time and memory storage proportional to the size of the problem: this precludes its use on very large datasets, which are, thanks to the spread of mobile devices such as smartphones, becoming more and more used to sharpen the predictions in cognitive neuroscience (18; 19; 20).

The second important drawback of this model regards its identifiability. Typically, only trial-to-trial variability in three (12; 7) or less (5; 21) 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 a variability in the boundary *position* is equivalent

to a variability in the start point position (7; 22). We claim here and show that a variability in the boundary *separation* is instead identifiable [1]. Indeed, precise mathematical justifications of the motivation for discarding the threshold variability are still lacking, and justifications for this choice are mainly experimental (7). Nevertheless, when when the threshold is considered as fixed, identifiability remains an issue (23). In the same vein, it is usually considered that one can expect the fit of the mean of the parameters to be accurate (7), but not the variance (i.e. trial-to-trial variability) (12). In short, large datasets (e.g. > 100 trials per subject) take an unreasonable amount of time to be optimized with these models, for a quality of fit that is mostly poor, and worsen dramatically for 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 Estimate (MLE; i.e. finding the most likely value parameters would endorse in order to generate the data at hand) was traditionally used (12; 24; 25; 13; 26; 27; 7; 14), until Bayesian attempts were made (28; 29; 15). Crucially, the validity of the MLE approach is based on some asymptotic properties of the summary statistics of a given dataset when the amount of data tends to infinity (30). 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 MLE. Again, MCMC only partly solves this problem, as its computational cost is still a function of the size of the problem, and, moreover, 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 barely considered as independent (see for instance (31; 32; 33; 34; 35; 36; 37)). Let us for instance consider the toy example where a participant has pressed 9 times the same button (e.g. right), and we are performing inference about the parameter at the 10th button press. From a Bayesian point

[1] to see this, one can perfectly imagine a variability of the start point equal to 0 but a high variability of the distance between the boundaries.

of view, it is very likely that, at this trial, the subject will choose the same action as before. If the participant indeed chooses the right response, we can expect that the RT at this trial will be, on average, lower than the one at the trial before. However, if the participant has chosen the left response at this trial, it is even more likely that this response will be slower than the previous trial. 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 cross-correlation between parameters value is likely to be complex, and hard to model precisely with informative models. The fitting approach we propose can be used in a first step to uncover the DDM parameters, before trying to interpret the pattern of the sequence observed.

An important remark we make is that, in this perspective, we may find a great interest in modelling the threshold variability: post-error slowing is indeed a well-known psychological phenomenon, that is usually interpreted as a change of speed-accuracy trade-off following an error (38; 39).

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 (14) at the within and between subject levels. Also, it can prevent overfitting by the use of dimensionality reduction, 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 will show that the Full DDM finds a natural formulation in Bayesian Statistics (Section 2.2 and Section 2.3), and that we can use this formulation to solve efficiently this problem using recent advances in Approximate inference, mostly with Variational Inference techniques (Section 2.4). We then present a computationally efficient Variational Bayes model (*i.i.d.*-VB) to compute the marginal probability density of the Full DDM. This algorithm has the net advantages 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 manner using a Recurrent Auto-Encoder (RAE-DDM) (Section 2.6). This approach is about to be assumption-free, meaning that it does not require the user to select a specific prior distribution to each of the parameters nor a precise form of trial-to-trial cross-correlation.

We show in the results (Section 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. An important and compelling result we will provide with the RAE-DDM consists in showing that the trial-to-trial variability of the threshold parameter is possible to estimate as for any other parameter of the DDM.

In Section 4, we will discuss our model in detail, highlight situations where it might prove useful, see its limitations and introduce potential improvements.

1.1. Related Work

The Wake-Sleep algorithm (40) developed almost twenty years ago by Peter Dayan constitutes a cornerstone in the history of Bayesian inference. This algorithm is a two-side 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 hold 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) (41) and its recurrent form (42; 43) have quickly become standard and popular tool to model data in many areas (44). 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. (45; 46)). In short, the first challenge with these techniques regards determination of the best approximate posterior shape for the latent variable $\mathbf{z}_{j,n}$ (47; 48; 49). An efficient solution to improve the performance of a VAE is to replace the Variational estimator by an Importance Weighted (IW) (50) 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 (51) has been shown to efficiently prevent overfitting with similar models (52), motivating our choice of relying on variational Dropout to perform inference about the value of the network weights (53).

Many routines have been proposed to optimize the DDM. Ratcliff and Tuerlinckx (12), and others (54; 55; 10) have used a NQ algorithm with a numerical estimate of the gradient to optimize the

likelihood function at the subject level. The computational cost of this method is extremely high, justifying the development of less flexible, but simpler methods (56; 57; 8; 58). Vandekerckhove et al. (29) used a Markov Chain Monte Carlo (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 (59), WinBUGS (29), or Stan (60), which has broaden the use of this technique for the optimization of the Full DDM in many recent publications (e.g. (61; 62; 63; 64; 65)).

The algorithm we propose differs in many ways from the work of Vandekerckhove and colleagues (29): 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 ELBO is estimated stochastically. The use of Amortized Variational Inference (66) makes it also possible to control the convergence of the algorithm using a test 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.

HDDM is a Python toolbox written by Wiecki and colleagues (15), 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 DDN parameters. Our aim will be to alleviate the computational burden of the integration process without compromising on accuracy and while keeping the entire original structure of the DDM.

None of the approaches presented so far took advantage of the 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: (32; 33; 34; 67) 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, (68; 21) 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 kind 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 neuro-physiological measure. Notably, this approach requires a large dataset (69) and inference is achieved with a MCMC method. Also, it is still based upon a quite 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 parameters, 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.

A final remark concerns identifiability of the model we propose, and of the Full DDM in general. The impact of the choice of prior probability shape has lead to vivid debates in the SSM community, some authors claiming that the choice of the prior probability completely determines the capability of the DDM to predict data: (70) have claimed that a prior probability of an arbitrary complexity can always be found in order to reflect perfectly the data that are modeled. This observation, also disputed by (71), goes way beyond the inclusion of the threshold variability in the Full DDM: following these authors' logic, a Jth component linear mixture model prior (where J is the number of trials of the dataset) over the sole parameter ξ can actually reflect perfectly the dataset. From a Bayesian perspective, such observation is quite fruitless, as it could in theory be applied to any hierarchical model where local variables are modeled: as a matter of fact, there should always exist a MLE estimate of any model whatsoever that would predict perfectly the data at hand with little or no generalization capability. Two specific – and widely used – solutions exist for this kind of issue: the first one, Bayesiam Model Comparison (72; 73) consists in measuring the model evidence (i.e. probability of the data given the model marginalized over the space of parameters) and compare it with simpler or more complex models. The second one would consist in assessing the model predictive capacity on a *test* dataset, separate from the *training* dataset that is used for fitting the model. The approach we propose can achieve these two goals efficiently, as the model

is able to generalize its predictions to unseen data and provides an lower-bound to the log-model evidence.

Furthermore, the method we propose alleviates the need to specify a precise prior probability distribution (and would, therefore, following a naive interpretation of the results of (70) be unidentifiable), but, by the use of dimensionality reduction, it ensures that the maximum of the variability is captured each component by the trial-wise latent variable $\mathbf{z}_{j,n}$: by doing so, we keep a highly generalizable model structure with little or no cost caused by the specific prior specifications.

Finally, the capacity of estimating the threshold variability is a considerable benefit of our approach, which has never been considered before. Other authors (e.g. (39; 62)) 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. Methods

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 a, the drift-rate ξ, the relative start point w and the non-decision time τ will be grouped under a single vector $\boldsymbol{\omega}$ (or, if applicable $\boldsymbol{\omega}_n$ if it is assumed that $\boldsymbol{\omega}_{n,j} = \boldsymbol{\omega}_{n,j'} \, \forall j, j' \in 1 : J$). The subject-specific prior distribution parameters of $\boldsymbol{\omega}_{j,n}$ will be generically grouped under the parameter vector $\boldsymbol{\theta}_n$. As we will see in the 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 set of parameters will have a specific form ($\boldsymbol{\theta}_n \triangleq \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2\}$ 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 more often use the generic indicators than their implementation-specific counterparts.

The behavioural observations are grouped under $\mathbf{y}_{j,n} \triangleq \{t_{j,n}, c_{j,n}\}$, where $t_{j,n}$ and $c_{j,n}$ 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 c, although we will freely use $p(t, c)$ and $p(t, b)$ interchangeably following the context. In general, capital letters will represent the set of all the corresponding parameters or random variables (i.e. $\mathbf{Y} \triangleq \{\mathbf{y}_{j,n} \text{ for } j \in 1 : J, n \in 1 : N\}$, $\boldsymbol{\Theta} \triangleq \{\boldsymbol{\theta}_n \text{ for } n \in 1 : N\}$ etc.). The kth 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 $\boldsymbol{\delta}_{j,n}(\mathbf{y}_{j,n}; \boldsymbol{\rho})$.

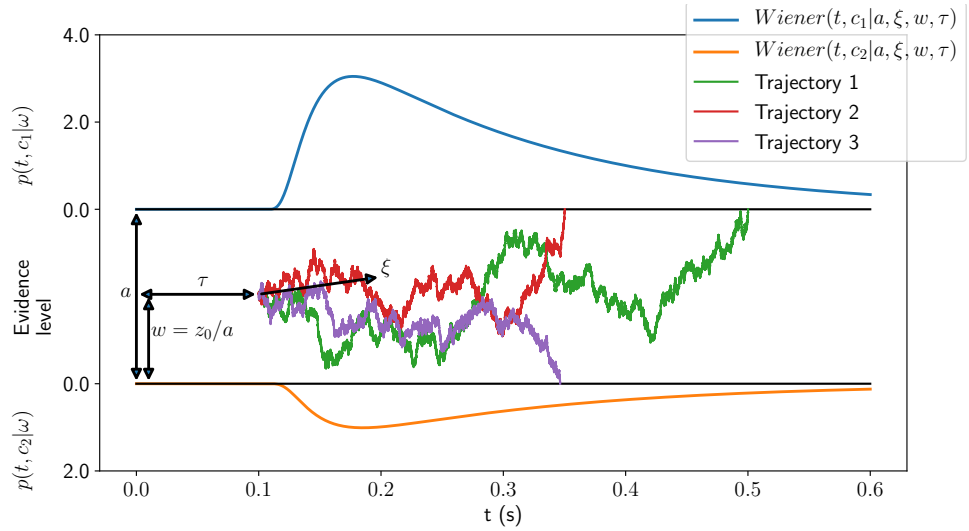
2.2. DDM

The Drift-Diffusion model is a member of the SSM 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 $\{c_{j,n}, t_{j,n}\}$ 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 (71), 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 1):

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

where dx is the evidence step, dt is the time step and sdW a Gaussian noise with a standard deviation s. In this specific model, the agent accumulates evidence in favour of an action c_1 with a rate ξ, called the drift rate, meaning that in the case of a classical Two-alternative forced choice task, he will also accumulate evidence in favour of the alternative action c_2 with a rate $-\xi$. Once the evidence crosses one of the two absorbing boundaries (the boundary b positioned at a distance a corresponding to the action c_1 or the boundary $\neg b$ at position 0 corresponding to the action c_2), the corresponding action is selected and, after some physiological delay τ, this action is executed. The evidence accumulation can start wherever at $z_0 \in [0; a]$, although it is more convenient to model



RT

Figure 1: DDM structure. The x axis represents the time scale, the y axis the evidence level. Four parameters (threshold a, 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.

the relative start point $w \in [0; 1]$ where $w = z_0/a$. Finally, this 5-parameter model is non-metric: multiplying ξ, a, z_0 and s 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 $s = 0.1$ is usually used as a metric, but, for the sake of mathematical explicitness and simplicity, we will use the metric $s = 1$ 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 (74):

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

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 (6; 21)) forms a valid probability distribution, i.e.:

$$\sum_{b \in \mathbf{b}} \int_{t=\tau}^{\infty} Wiener(t, b; \boldsymbol{\omega}) dt = 1$$
$$\text{and } Wiener(t, \mathbf{b}; \boldsymbol{\omega}) \geq 0 \text{ for t} \in [\tau, \infty] \text{ and } b \in \mathbf{b} \tag{2}$$

where $\mathbf{b} = \{b, \neg b\}$. The function in Equation (1) involves an infinite sum, and must therefore be implemented through approximations. Navarro et al. (75) 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 $\boldsymbol{\omega}$ vary 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 (5). Formally,

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

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 $w^* < .5$ (5). Accounting for the trial-to-trial variability in w can solve this problem. The problem then becomes to fit a prior $\boldsymbol{\theta}_n$ according to which the trial-wise latent parameters $\boldsymbol{\omega}_{j,n}$ are distributed. In frequentist paradigms, the parameters $\boldsymbol{\omega}_{j,n}$ are usually integrated out of the model, and $\boldsymbol{\theta}_n$ only is fitted. As we will see in 2.3, in a fully Bayesian framework, all the parameters ($\boldsymbol{\omega}_{j,n}$, $\boldsymbol{\theta}_n$ 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 justified 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 that the performance of the model we propose still outperforms other implementations, and that we are able to find meaningful results for the fit of the threshold.

2.3. The Full DDM as an Empirical Bayes problem

The MLE of the Full DDM as formulated by (12) consists in finding the value of the prior $\boldsymbol{\theta}_n^*$ that maximizes the log-likelihood function $\log p(\mathbf{y}_{j,n}; \boldsymbol{\theta}_n)$ 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 $\boldsymbol{\theta}_n$, namely $\widehat{\boldsymbol{\theta}}_n$. This choice can be justified from the asymptotic normality of the posterior distribution theorem (30; 77): as the dataset grows, and considering that the data are generated according 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. $\boldsymbol{\theta}_n^* \to \widehat{\boldsymbol{\theta}_n}$ as $NJ \to \infty$ and $p(\boldsymbol{\theta}_n) \approx \mathcal{N}(\boldsymbol{\theta}_n^*, I(\boldsymbol{\theta}_n^*)^{-1})$, where $I(\boldsymbol{\theta}_n^*) = -\nabla^2_{\boldsymbol{\theta}_n^*} \log p(\mathbf{y}_{j,n}; \boldsymbol{\theta}_n)$. Also, following the Bernstein-von Mises theorem (78; 79; 80), we can see that the choice of the prior becomes irrelevant to the posterior as the amount of data grows, which further justifies the MLE technique for large datasets.

We show in Appendix B that the MLE approach of the Full DDM reduces to what is known as an empirical Bayes estimator of $\widehat{\boldsymbol{\theta}}_n$: indeed, the value of $\boldsymbol{\theta}_n$ that maximizes the Wiener density can be directly reformulated as a function of the posterior probabilities of each trial parameter $p(\boldsymbol{\omega}_{j,n} \big| \mathbf{y}_{j,n}; \boldsymbol{\theta}_n)$: 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 2a shows the basic structure of the Full DDM in this perspective.

2.4. Variational Inference

In this section, we will present Variational Inference as a method to find the posterior of the trial-wise DDM parameters $\boldsymbol{\omega}_{j,n}$. We will first focus on the more traditional EM-like approach which is proposed in Appendix B, because it is widely used in neuroscience (81; 82; 83; 84; 85):

it consists in looping between the evaluation of each trial-specific posterior probability density function given the current estimate of $\boldsymbol{\theta}_n$, before then finding the value of $\boldsymbol{\theta}_n$ that maximizes the likelihood (or the lower-bound to the log-model evidence in a fully Bayesian setting) given the current posterior estimates. We will expose its weaknesses, and present another, neater method to maximize $\ell(\mathbf{y}_{j,n}; \boldsymbol{\theta}_n)$ that relies on Stochastic Gradient Variational Bayes (SGVB) and inference networks.

As stated above, the EM algorithm relies on computing the posterior over $\boldsymbol{\Omega}_n$, which reads

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

assuming that the posterior probability factorizes over each datapoint. This posterior probability intractable due to its normalizing factor (the model evidence) $p(\mathbf{Y}|\boldsymbol{\theta}_n)$. A first approach could be to use simulation techniques to approximate 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 Variational Bayes techniques (VB) (86; 87; 17; 88) to fit an approximate posterior to the true posterior. In the present case, in which we are interested in the MLE value of the prior, this approach is called Variational EM, or VEM.

In short, VB works as follows: 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 2b). In order to do so, we can choose to minimize the Kullback-Leibler (KL) divergence between this distribution and the true posterior, since this measure is minimized when the two distributions match exactly:

$$\begin{aligned} \text{if } \mathrm{D_{KL}}[q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) || p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n})] = 0 \\ \triangleq \int q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) \log q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) d\boldsymbol{\omega}_{j,n} - \\ \int q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) \log p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n}) d\boldsymbol{\omega}_{j,n} \\ \text{then } q(\boldsymbol{\omega}_{j,n}) = p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n}). \end{aligned} \tag{3}$$

The first term of the KL divergence is the negative differential entropy of $q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})$:

$$\mathrm{H}[q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})] \triangleq - \int q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) \log q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) d\boldsymbol{\omega}_{j,n}.$$

For the second term, we can expand $\log p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n}) = \log p(\mathbf{y}_{j,n}, \boldsymbol{\omega}_{j,n}) - \log p(\mathbf{y}_{j,n})$, and, considering that $\int q(\boldsymbol{\omega}_{j,n}) \log p(\mathbf{y}_{j,n}) d\boldsymbol{\omega}_{j,n} = \log p(\mathbf{y}_{j,n})$, the value of $\boldsymbol{\delta}_{j,n}$ that minimizes the KL divergence simplifies to:

$$\arg\min_{\boldsymbol{\delta}_{j,n}} \underbrace{-\mathrm{H}[q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})] - \int q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) \log p(\mathbf{y}_{j,n}, \boldsymbol{\omega}_{j,n}) d\boldsymbol{\omega}_{j,n}}_{-\mathcal{L}(q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}))} + \log p(\mathbf{y}_{j,n})$$

which implies that

$$\begin{aligned} &\mathrm{D}_{\mathrm{KL}}[q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) || p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n})] = 0 \\ \Leftrightarrow\ &\mathcal{L}(q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})) = \log p(\mathbf{y}_{j,n}) \end{aligned} \tag{4}$$

and therefore

$$\mathcal{L}(q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})) = \mathbb{E}_{q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})} [\log p(\mathbf{y}_{j,n}, \boldsymbol{\omega}_{j,n})] + \mathrm{H}[q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})] \tag{5}$$

is a lower bound to the log model evidence (or ELBO, for Evidence Lower Bound, also sometimes called negative Free Energy). In essence, Equation (4) tells us that the value of $\boldsymbol{\delta}_{j,n}$ that minimizes $\mathrm{D}_{\mathrm{KL}}[q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}) || p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n})]$ is the one that maximizes the ELBO $\mathcal{L}(q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n}))$. Once this value of $\boldsymbol{\delta}_{j,n}$ has been optimized, we can consider that $q(\boldsymbol{\omega}_{j,n}; \boldsymbol{\delta}_{j,n})$ is as close as it could from $p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n})$ given the distribution shape we have chosen, the prior probability $p(\boldsymbol{\omega}_{j,n})$, the data $\mathbf{y}_{j,n}$ and the likelihood function $p(\mathbf{y}_{j,n}|\boldsymbol{\omega}_{j,n})$. Most approaches (89; 90; 91; 92; 93; 94) rely on some form of (possibly natural) gradient ascent method to solve this problem: the gradient of the ELBO wrt. $\boldsymbol{\Delta} = \{\boldsymbol{\delta}_{j,n} \text{ for } j \in 1:J, n \in 1:N\}$ is computed, and the optimization proceeds by either optimizing all the variational parameters altogether, or by sequentially optimizing some of them while keeping the others fixed in an EM-like approach. VEM is a special case of the latter, where we are interested in the MLE of the prior.

This VEM method suggested above, where $\boldsymbol{\Delta}$ and $\boldsymbol{\theta}_n$ 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. 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 (17). Another valuable 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 Networks.

2.4.1. Stochastic Gradient Descent, Reparametrization trick and SGVB

Let us rewrite the optimization problem we need to solve to find the MLE of $\boldsymbol{\theta}_n$ using VB in a concise form:

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

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

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

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 estimator given by

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

for a number of K samples. The form we have given to the (expected) log-likelihood makes it clear that

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

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

The use of the auxiliary variable to compute the gradient estimator is sometimes called Infinitesimal Perturbation Analysis estimator (95; 96; 97) or Reparametrized Gradient (98). It has generally a

much lower variance than the gradient obtained with the Score function method (98; 93), and, for a large enough mini-batch size $US >> 0$, it can be used with a single sample $K = 1$.

This kind of noisy gradient can be used to find the minimum of a loss function using Stochastic Gradient Descent (SGD) techniques (99), 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. to both $\boldsymbol{\Theta}$ and $\boldsymbol{\Delta}$. This already amortizes the cost of the optimization, as 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 because they are caused by a small cluster of data that is partially ignored during this step (100).

Many SGD algorithms have been developed (e.g. (101; 102)). As our aim is not to make an extensive review of they respective advantages and disadvantages, we decides to use the Adam optimizer (41) and refer to the corresponding paper for a detailed algorithmic description and implementation.

2.4.2. Inference Network

Flexibility and sparsity are two of the main desired properties of the approximate posteriors q 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 value of the adjacent nodes of factor graph to a given approximate posterior $\boldsymbol{\delta}_{j,n} \triangleq f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_n; \boldsymbol{\rho})$ 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 $q_{\boldsymbol{\rho}}(\mathbf{z}_{j,n}|\mathbf{y}_{j,n}, \boldsymbol{\theta}_n)$ (Figure 2c). 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.

This use of an inference network (IN) is known as Neural Variational Inference and Learning (66) or amortized variational inference (AVI) (47). This technique thereby led to numerous applications (93; 103; 47; 53; 104; 48; 105), as it has several advantages: first, from a computational point of view, it alleviates the need of optimizing each of the local approximate posteriors, which can be problematic for large datasets. Instead, a generic function that maps the points to their approximate posterior, is fitted to the problem at hand. Second, this technique allows the model to find a likely posterior for datapoints that might not have been seen yet, or that have been seen many iterations ago.

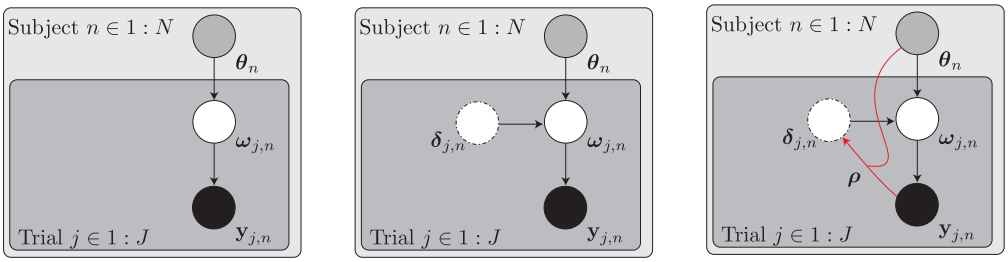
Finally, and following the same logic, algorithms using IN can spare large amount of memory usage by storing only the parameters of the mapping $\boldsymbol{\rho}$ and not each of the local variational parameters.

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

$$\widetilde{\nabla}^{\mathcal{L}} \triangleq \frac{1}{SUK} \sum_{n,j,k}^{S,U,K} \left[\begin{array}{c} \nabla_{\boldsymbol{\theta}_n} \left\{ g\left(f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_n; \boldsymbol{\rho}), \boldsymbol{\epsilon}_k \right) \right\} \nabla_{\widetilde{\boldsymbol{\omega}}_{j,n,k}} \left\{ \log p(\mathbf{y}_{j,n}, \widetilde{\boldsymbol{\omega}}_{j,n,k}; \boldsymbol{\theta}_n) \right\} + \\ \nabla_{\boldsymbol{\theta}_n} \left\{ \mathrm{H}[q(\mathbf{z}_{j,n}; f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_n; \boldsymbol{\rho}))] \right\} \\ \nabla_{\boldsymbol{\rho}} \left\{ g\left(f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_n; \boldsymbol{\rho}), \boldsymbol{\epsilon}_k \right) \right\} \nabla_{\widetilde{\boldsymbol{\omega}}_{j,n,k}} \left\{ \log p(\mathbf{y}_{j,n}, \widetilde{\boldsymbol{\omega}}_{j,n,k}; \boldsymbol{\theta}_n) \right\} + \\ \nabla_{\boldsymbol{\rho}} \left\{ \mathrm{H}[q(\mathbf{z}_{j,n}; f(\mathbf{y}_{j,n}, \boldsymbol{\theta}_n; \boldsymbol{\rho}))] \right\} \end{array} \right] \tag{6}$$

where $\widetilde{\boldsymbol{\omega}}_{j,n,k} = g\left(f(\mathbf{y}_{j,n}; \boldsymbol{\rho}), \boldsymbol{\epsilon}_k \right)$. The gradient in Equation (6) can be computed easily using an Automatic Differentiation routine if $g(\cdot)$ is differentiable, which excludes for instance random number generation algorithms based on acceptance-rejection algorithms (106; 107), although this limitation can be circumvented (see (108; 109)).

Approximate Posterior specifications. We now need to specify a convenient shape for the approximate posterior of the DDM parameters. In Appendix C, we opted for the use of Normalizing Flows (52) 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: Directed Acyclic Graphical Model (DAG) of the AVI approach to the Maximum-Likelihood (approximate empirical Bayes) Full DDM problem. Black filled circled 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 $\boldsymbol{\theta}_n$ that maximizes the marginal likelihood $p(\mathbf{y}_{j,n}|\boldsymbol{\theta}_n)$, 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 $q(\boldsymbol{\omega}_{j,n}|\boldsymbol{\delta}_{j,n})$. *C* Amortized Variational Inference (AVI) amortized the cost of inferring the approximate posterior for each latent variable by using an Inference network 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 $\boldsymbol{\theta}_n$ 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.

2.5. 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. Likewise, it makes it possible to optimize the algorithm in an online fashion, as the same mapping function can be used for datapoints that have and that do not have been observed yet.

Comparison of Algorithm 1 and Algorithm 2 highlights the difference between the more classical VEM method and AVI. Note that, in the former, the exact posterior probability from Equation (B.6) 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 2 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 test dataset starts to decrease (early stopping (17)). This is the method we will present in the Result section 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 1: 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 $n \in 1 : N, j \in 1 : J$ **do** `// E step`
    Find $\arg\min_{\boldsymbol{\delta}_{j,n}} -\mathcal{L}(q(\mathbf{z}_{j,n}; \boldsymbol{\delta}_{j,n}))$ `// e.g. using SGD`
  for $n \in 1 : N$ **do** `// M step`
    Set $\boldsymbol{\mu}_n{}^* = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_q\left[\mathbf{z}_{j,n}\right]$
    Set $\boldsymbol{\sigma}_n^{2\,*} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_q\left[\mathbf{z}_{j,n}\right]^2 + \mathbb{V}ar_q\left[\mathbf{z}_{j,n}\right] - (\boldsymbol{\mu}_n{}^*)^2$
  Compute $ELBO_i = \mathcal{L}(q(\mathbf{Z}; \mathbf{\Delta}))$;
until $i > M$ *or* $ELBO_i - ELBO_{i-1} < u$;

Algorithm 2: 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 $n \in 1 : N$ randomly;
    Select $j \in 1 : J$ randomly;
    Draw $\boldsymbol{\epsilon} \in p(\boldsymbol{\epsilon})$ randomly;
    Compute an unbiased, noisy estimate of $\widetilde{\nabla}_k^{\mathcal{L}} - \nabla_{\boldsymbol{\rho},\boldsymbol{\theta}_n}\left\{\mathcal{L}(q(\mathbf{Z}; \boldsymbol{\rho}))\right\}$ `// see Equation (6)` ;
  Update $\boldsymbol{\rho} = \boldsymbol{\rho} - \eta \frac{1}{SUK}\sum_{k=1}^{SUK} \widetilde{\nabla}_{\boldsymbol{\rho},k}^{\mathcal{L}}$;
  Update $\boldsymbol{\theta}_n = \boldsymbol{\theta}_n - \eta \frac{1}{SUK}\sum_{k=1}^{SUK} \widetilde{\nabla}^{\mathcal{L}\boldsymbol{\theta}_n,k}$;
until $i > M$;

2.6. Recurrent Auto-encoder

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

The model we will develop is based on Variational Auto-Encoders. Auto-Encoders (AE), and more specifically Variational AE (VAE), is a large family of models (93) 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 *inference network* so far: it is a (non-)linear mapping with global parameters $\boldsymbol{\rho}$ that predicts the shape of the approximate posterior of a local latent variable $\mathbf{z}_{j,n}$ of an arbitrary length d, based on the data $\mathbf{y}_{j,n}$, with a prior $p(\mathbf{z}_{j,n})$. This variable $\mathbf{z}_{j,n}$ lies between the *encoder* and the *decoder*, which constitutes the *generative model* about which inference is made. Figure 3a sketches the structure of a Variational Auto-encoder.

Again, this generative model maps the parameter $\mathbf{z}_{j,n}$ 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 $\boldsymbol{\omega}_{j,n} = f(\mathbf{z}_{j,n}; \boldsymbol{\phi})$.

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 dimensionality reduction algorithms, where the bottleneck is situated at the level of the latent variable $\mathbf{z}_{j,n}$.

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

$$\mathcal{L}\left(q(\mathbf{z}_{j,n}\big|\mathbf{y}_{j,n})\right) = \mathbb{E}_{q_{\boldsymbol{\rho}}(\mathbf{z}_{j,n}\big|\mathbf{y}_{j,n})}\left[\log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \mathbf{z}_{j,n}) - \log q_{\boldsymbol{\rho}}(\mathbf{z}_{j,n}\big|\mathbf{y}_{j,n})\right] \tag{7}$$

which recalls Equation (5) 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 (7), and the hierarchical treatment of this expression will be introduced shortly after. Let us emphasize in Equation (7) the dependency of the probability distributions on $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$: these are the only parameter to fit in order to maximize the ELBO. Ideally, one would like to make inference not only about $\mathbf{z}_{j,n}$ 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 (93), although inference about these parameters in far from being impossible (53; 48). This problem will be treated in 2.6.1.

Chung and colleagues (43) 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 (42; 43) due to its efficiency to trade accuracy and complexity of the inference process. Under this assumption, the ELBO now looks like:

$$\begin{gathered}\mathcal{L}\left(q(\mathbf{z}_{j,n}\big|\mathbf{y}_{j,n},\mathbf{h}_{j-1})\right) = \mathbb{E}_{q_{\boldsymbol{\rho}}(\mathbf{z}_{j,n}\big|\mathbf{y}_{j,n},\mathbf{h}_{j-1})}\left[\log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n},\mathbf{z}_{j,n}\big|\mathbf{h}_{j-1}) - \log q_{\boldsymbol{\rho}}(\mathbf{z}_{j,n}\big|\mathbf{y}_{j,n},\mathbf{h}_{j-1})\right] \\ \text{where } \mathbf{h}_{j-1} = f(\mathbf{y}_{j-1,n},\mathbf{z}_{j-1,n},\mathbf{h}_{j-2,n};\boldsymbol{\phi}_{\mathbf{h}}).\end{gathered} \tag{8}$$

In Equation (11), we have made explicit the fact that the approximate posterior q and the likelihood $p(\mathbf{y}_{j,n},\mathbf{z}_{j,n})$ 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 $\mathbf{z}_{j,n}$ 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 $\mathbf{z}_{j,n}$ to its minimum, we ensure that each element of $\mathbf{z}_{j,n}$ 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 the most 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 $\mathbf{z}_{j,n}$. 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, one can assume that $\mathbf{z}_{j,n}$ is normally distributed in Equation (7) with a mean 0 and a covariance I_d, which ensured that the components of $\mathbf{z}_{j,n}$ are somehow independent (**?**) and, therefore, that each of them captures the maximum variance in the data. This choice is not optimal for the model in Equation (11), as one would expect the prior of $\mathbf{z}_{j,n}$ to vary from trial to trial. We used therefore the recommendation of (43) and used a form of prior $\{\boldsymbol{\mu}_0^{\mathbf{z}_{j,n}}, (\boldsymbol{\sigma}_0^{\mathbf{z}_{j,n}})^2\} = f(\mathbf{h}_{j-1};\boldsymbol{\phi}_p)$ where, again $\boldsymbol{\phi}_p$ is 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 $\boldsymbol{\zeta}_n$ of length d^* with a prior $\mathcal{N}(0, I_{d^*})$ and an approximate posterior $q(\boldsymbol{\zeta}_n \big| \boldsymbol{\iota}_n)$. This regressor was introduced in the generative model described above to obtain a set of DDM parameters that depended on the three factors $\mathbf{z}_{j,n}, \mathbf{h}_{j-1}$ and $\boldsymbol{\zeta}_n$:

$$\boldsymbol{\omega}_{j,n} = f\left(\mathbf{z}_{j,n}, \mathbf{h}_{j-1}, \boldsymbol{\zeta}_n; \boldsymbol{\phi}_{\boldsymbol{\omega}}\right). \tag{9}$$

This simple manipulation ensured that inference at the trial level was 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 $\mathbf{z}_{j,n}$

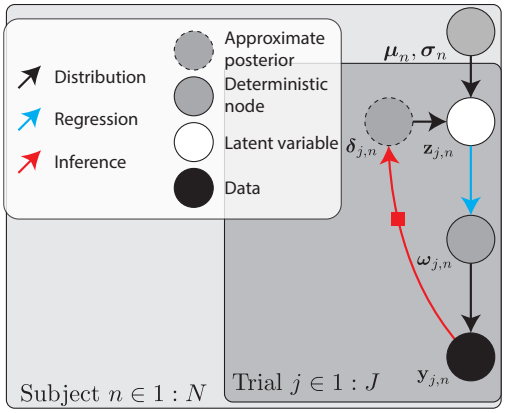
$$\mathbf{z}_{j,n} \sim q_{\boldsymbol{\rho}}(\mathbf{z}_{j,n} \big| \mathbf{y}_{j,n}, \mathbf{h}_{j-1}, \boldsymbol{\zeta}_n). \tag{10}$$

It flows directly from Equation (9) and Equation (10) that, if the dataset has been acquired using various experimental conditions, or if the subject is somehow informed about its performance in the task by the display of a sensory feedback, these regressors can be easily included the set of the predictors of $\boldsymbol{\omega}$, making this model a compellingly generic tool for inference about the vector $\boldsymbol{\omega}$.

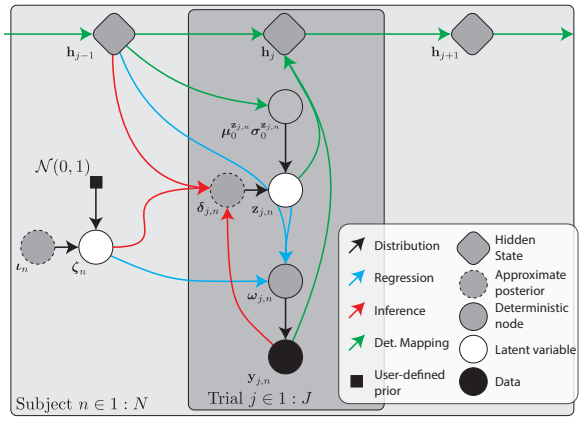
Training the RVAE DDM. All the tools we have provided so far can be used in order to train a VAE: the inference network, with possibly the normalizing flow that is presented in Appendix C can be used to find the posterior probability of $\mathbf{z}_{j,n}$ with an amortized cost. SGVB (by the use of the reparametrization trick) could be used to optimize the ELBO, with an unbiased gradient of the form

$$\nabla_{\boldsymbol{\rho},\boldsymbol{\phi}} \left\{ \mathcal{L}\left(q(\mathbf{z}_{j,n} \big| \mathbf{y}_{j,n}, \mathbf{h}_{j-1})\right) \right\} = \frac{1}{JSK} \sum_{j,n,k}^{J,S,K} \left\{ \begin{array}{c} \nabla_{\boldsymbol{\rho}} \left\{ \log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \big| \mathbf{h}_{j-1}) - \log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k} \big| \mathbf{y}_{j,n}, \mathbf{h}_{j-1}) \right) \right\} \\ \nabla_{\boldsymbol{\phi}} \left\{ \log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \big| \mathbf{h}_{j-1}) \right\} \end{array} \right. \tag{11}$$

where we have made implicit the fact that $\mathbf{z}_{j,n}$ 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



(a) Variational Auto-encoder with Maximul-Likelihood estimate of the top-level hyperprior. Note the difference with 2c, 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.

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*, which transforms the ELBO in an Importance Sampling (IS) estimate of the the log-model evidence:

$$\mathcal{L}\left(q(\mathbf{z}_{j,n}|\mathbf{y}_{j,n},\mathbf{h}_{j-1})\right) \approx \frac{1}{JS}\sum_{j,n}^{J,S}\log\frac{1}{K}\sum_{k}\frac{p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n},\widetilde{\mathbf{z}}_{j,n,k}|\mathbf{h}_{j-1})}{q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k}|\mathbf{y}_{j,n},\mathbf{h}_{j-1}))}. \tag{12}$$

When used without recurrent predictions, this is known as an Importance Weighted AE (IWAE) (50), 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) (45).

This improved estimate of the ELBO has many advantages, that are detailed in (45): 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 that the SGVB approach, which can be simply understood observing that the FIVO approach takes more sampled than SGVB. Finally, this approach mostly 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 that 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 $\widetilde{\tau}_{j,n} > t$. We describe in Appendix C.3 a simple solution to this problem based on Lagrange multipliers.

The complete model, which we named the Recurrent Auto-Encoding Drift Diffusion Model (RAE-DDM) is sketched in Figure 3b and the training algorithm is displayed in **??**. This algorithm is an ad-hoc application of the algorithm presented in (43; 45), and we refer to these articles for the implementation details.

Algorithm 2: 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 $\boldsymbol{\iota}_n$

1 **Initialize** randomly $\{\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\iota}_n\}$;

2 Set $i \leftarrow 0$;

3 **repeat**

4 $\quad i \mathrel{+}= 1$;

5 $\quad$ Set $\widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\iota}_n} \leftarrow 0$;

6 $\quad$ **repeat**

7 $\quad\quad$ Select $n \in 1:N$ randomly;

8 $\quad\quad$ Initialize hidden state $\mathbf{h}_0 \leftarrow 0$, accumulated normalized weight $\boldsymbol{\beta}^n \leftarrow 1/K$;

$\quad\quad$ `// Forward Pass`;

9 $\quad\quad$ Set $\mathcal{L} \leftarrow 0$;

10 $\quad\quad$ Draw $\widetilde{\boldsymbol{\zeta}}_n = g(\boldsymbol{\epsilon}_{\boldsymbol{\zeta}_n}; \boldsymbol{\iota}_n)$ with $\boldsymbol{\epsilon}_{\boldsymbol{\zeta}_n} \sim p(\boldsymbol{\epsilon}_{\boldsymbol{\zeta}_n})$;

11 $\quad\quad$ Increment $\mathcal{L}$ with subject-wise noisy $\mathrm{D_{KL}}$ approximation $\mathcal{L} \mathrel{+}= \log p(\widetilde{\boldsymbol{\zeta}}_n) - \log q(\widetilde{\boldsymbol{\zeta}}_n \big| \boldsymbol{\iota}_n)$;

12 $\quad\quad$ **for** $j \in 1:J$ **do**

13 $\quad\quad\quad$ **for** $k \in 1:K$ **do**

14 $\quad\quad\quad\quad$ Draw $\widetilde{\boldsymbol{\zeta}}_n = g(\boldsymbol{\epsilon}_{\boldsymbol{\zeta}_n}; \boldsymbol{\iota}_n)$ with $\boldsymbol{\epsilon}_{\boldsymbol{\zeta}_n} \sim p(\boldsymbol{\epsilon}_{\boldsymbol{\zeta}_n})$;

15 $\quad\quad\quad\quad$ Map $\boldsymbol{\delta}_{j,n} = f(\boldsymbol{\rho}; \mathbf{y}_{j,n}, \mathbf{h}_{j-1}, \widetilde{\boldsymbol{\zeta}}_n)$;

16 $\quad\quad\quad\quad$ Draw $\widetilde{\mathbf{z}}_{j,n,k} = g(\boldsymbol{\epsilon}_{\widetilde{\mathbf{z}}_{j,n,k}}; \boldsymbol{\delta}_{j,n})$ with $\boldsymbol{\epsilon}_{\widetilde{\mathbf{z}}_{j,n,k}} \sim p(\boldsymbol{\epsilon}_{\widetilde{\mathbf{z}}_{j,n,k}})$;

17 $\quad\quad\quad\quad$ Map $\{\boldsymbol{\mu}_0^{\mathbf{z}_{j,n}}, (\boldsymbol{\sigma}_0^{\mathbf{z}_{j,n}})^2\} = f(\mathbf{h}_{j-1}; \boldsymbol{\phi}_p)$;

18 $\quad\quad\quad\quad$ Compute $H = \log p\left(\widetilde{\mathbf{z}}_{j,n,k} \big| \boldsymbol{\mu}_0^{\mathbf{z}_{j,n}}, (\boldsymbol{\sigma}_0^{\mathbf{z}_{j,n}})^2\right) - \log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k})$;

19 $\quad\quad\quad\quad$ Map $\widetilde{\boldsymbol{\omega}}_{j,n,k} = f(\widetilde{\mathbf{z}}_{j,n,k}, \mathbf{h}_{j-1}, \widetilde{\boldsymbol{\zeta}}_n; \boldsymbol{\phi})$;

20 $\quad\quad\quad\quad$ Compute unnormalized particle weight $\log \alpha_k^u = \log Wiener(\mathbf{y}_{j,n} \big| \widetilde{\boldsymbol{\omega}}_{j,n,k}) + H$;

21 $\quad\quad\quad$ Increment approximate log-model evidence with

22 $\quad\quad\quad\quad$ previous normalized weights $\mathcal{L} \mathrel{+}= \log \sum_k \beta_k^n \alpha_k^u$;

23 $\quad\quad\quad$ Update accumulated normalized weight $\beta_k^n = \beta_k^n \frac{\alpha_k^u}{\sum_{k'} \alpha_k^{u'}}$;

24 $\quad\quad\quad$ If resampling criterion is met, resample $\{\beta_k^n, \widetilde{\mathbf{z}}_{j,n,k}\}$ for $k \in 1:K$;

25 $\quad\quad\quad$ **for** $k \in 1:K$ **do**

26 $\quad\quad\quad\quad$ Map $\mathbf{h}_j = f(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k}, \mathbf{h}_{j-1}; \boldsymbol{\phi}_{\mathbf{h}})$

$\quad\quad$ `// Reverse Pass`;

27 $\quad\quad$ Backpropagate through lines 9 to 24 to get $\widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\iota}_n} = \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\iota}_n} - \nabla_{\boldsymbol{\rho},\boldsymbol{\phi},\boldsymbol{\iota}_n}\{\mathcal{L}\}$

28 $\quad$ **until** S *sequences have been collected*;

29 $\quad$ Update $\boldsymbol{\rho} = \boldsymbol{\rho} - \eta \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\rho}}$;

30 $\quad$ Update $\boldsymbol{\phi} = \boldsymbol{\phi} - \eta \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\phi}}$;

31 $\quad$ Update $\boldsymbol{\iota}_n = \boldsymbol{\iota}_n - \eta \widetilde{\nabla}^{\mathcal{L}}_{\boldsymbol{\iota}_n}$;

32 **until** $i > M$;

2.6.1. 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 (51), and has been suggested as a necessary requirement for Boltzman-machine like algorithms (52) such as VAE and RAE. It simply consists in hiding part of the network at each iteration, in order to ensure recursivity in the network and maximum relevance of each of the inputs. Variational Dropout (VD) (53; 110) is an improved version of the Dropout method where each weight of the network is assigned to a specific prior and corresponding variational distribution. Using VD, we can ensure that the model takes the maximum advantage of each of the inputs of the network, and prevent overfitting. Additionally, the use of VD makes the model fully Bayesian, as we make inference not only about $\mathbf{z}_{j,n}$ but also about the weights $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$. We implemented VD for the RAE-DDM, as it shown to perform slightly better than the vanilla version on the long-run, and prevent 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 (111), 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 2, meaning that we used the same sample for each and every trial of the sequence instead of resampling at each trial.

3. Experiment

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 3.1). 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 the ξ and w (lower-bounded at 0.3), and the variability of τ and a were chosen to be low (bounded at 0). In each dataset, we did not consider that there could exist a consistent rightward or leftward bias caused by the parameter w at the population level: in other words, $\mathbb{E}\left[w_{j,n} \forall\, j, n\right]$ was always chosen to be 0. The same logic was used for τ, which mean was chosen to have an average value

	a	ξ	w	τ
$\boldsymbol{\mu}_n$	$\mathcal{N}(1.2 + 0.2\epsilon_1, 0.1)$	$\mathcal{N}(1.5 + 0.33\epsilon_2, 0.2)$	$\mathcal{N}(0, 0.1)$	$\mathcal{N}(0.3, 0.01)$
$\boldsymbol{\sigma}_n^2$	$\mathcal{IG}^{-1}(5, u_1 * 0.25)$	$\mathcal{IG}^{-1}(5, 0.3 + u_2 * 0.33)$	$\mathcal{IG}^{-1}(5, 0.3 + u_3 * 0.33)$	$\mathcal{IG}^{-1}(5, u_4 * 0.033)$

Table 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.

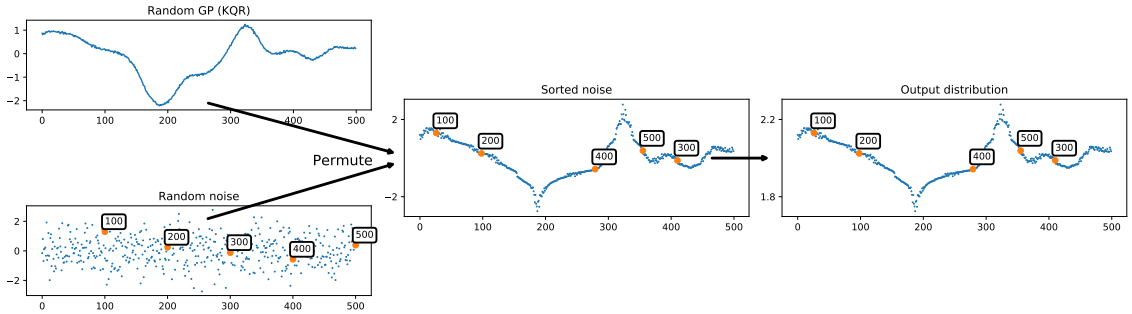


Figure 4: Data generation procedure for a prior distribution $\mathcal{N}(2, 0.1)$. All parameters parameters were generated according to a diagonal univariate Gaussian distribution. For each parameter, the order of the appearance of each sample was permuted to minimize the distance with a random GP trace (here with a Rational Quadratic kernel).

of 0.3 for each and every dataset across subjects and trials. For these two parameters, only the dataset-wise average variance was assumed to differ. The average mean value of ξ and a were, on the contrary, variable across datasets. The simulated task consisted in a highly generic TAFC task where the subjects needed to choose between a left and a right response following an instruction on the screen, consisting, for instance, in the motion direction of dots in a fixation patch whose direction indicated the instructed response. No difficulty or any more additional condition was 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 (7), which is physiologically sensible, as the start point should be set before the instruction is perceived by the agent (112).

In order to avoid biasing the data generation process toward a form of a cross-correlation between the DDM parameters (or the data), the only manipulation we allowed ourself to do was to permute the generated parameters in a specific order (Figure 4).

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 (113), parametrized with a population-wise random parameter $a \sim [0; 1]$ and with an input vector of equally distant variables u situed 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 Euclidian 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.

The above manipulation ensured that the data generated were sorted, but that a fit achieved using the *i.i.d.* assumption were not harmed by the assumption of the data generation procedure.

RT and choices were simulated according to (12), where the Wiener is discretized to an arbitrarily small δt (set to 10^{-6} in our case) and approximated as a binary gambler ruin problem (74). An alternative method could consist in using the Euler-Maruyama algorithm (114; 115), 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 arbitrarily chosen random seeds 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 exact similar HMC version of this last model coded in stan (60). 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 -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 lead to a total of five fits per dataset.

3.2. Model Specification

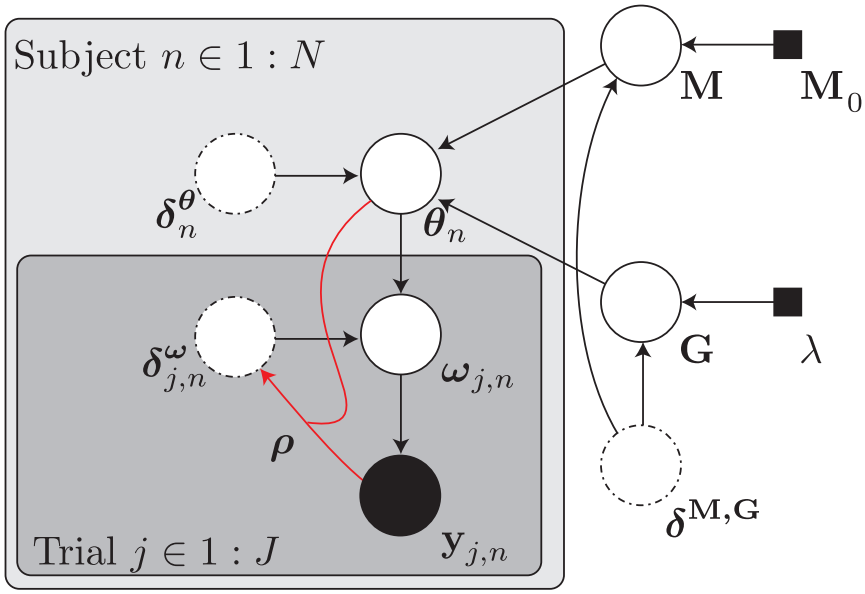


Figure 5: *i.i.d.*-VB and HMC full hierarchical structure. The DDM parameters were distributed according to a subject-specific prior $\boldsymbol{\theta}_n = \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2\}$. In turn, these parameters were distributed according to a latent Normal $\mathbf{M}$ prior $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2)$ 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 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 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 Normalizing Flow with parameters $\boldsymbol{\delta}^{\cdot}$, where $\cdot$ indicates the latent variable. The parameters $\boldsymbol{\delta}_{j,n}^{\boldsymbol{\omega}}$ were generated thanks to an inference network with parameters $\boldsymbol{\rho}$.

i.i.d.-VB . For each of the $f(\cdot)$ transform in **??**, we used a two-layer perceptron with a Restricted Linear Unit (ReLU) activation function (116; 117), 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 (118). We chose arbitrarily the Gated Recurrent Unit (GRU) (119) for the RNN model, with the same ReLU activation function. An alternative strategy could consist in using a Long-Short Term Memory (LSTM) (120), 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.

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 $\mathbf{z}_{j,n}$ was set to 4 for all experiments (so was the length of the subject-specific 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 (121), optimized with a noise-insensitive version of the Adam optimizer (41; 122) for 80000 iterations, in parallel with an implementation of the SGD algorithm similar to (123), 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 *i.i.d.* versions of the DDM (VB or HMC) were all tested with a model identical to the true one, i.e. where the drift was influenced by the perceived motion.

3.3. Testing for model specifics: Convergence and overfitting assessment

We tested the convergence of the two *i.i.d.*-VB algorithms using both a testing dataset during training. Convergence rate was fast, and the performance over the test dataset did not worsen during the optimization. We calculated *a posteriori* an ad-hoc measure of VB performance named Pareto Smoothed Importance Sampling (PSIS) (124). 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 (125) 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 0.07 for the classical versions of *i.i.d.*-VB and 0.01 for the full versions.

Interestingly, a Bayes Factor analysis calculated between the model evidence of the full and classical version of the model favoured almost exclusively, albeit slightly, the full version (mean BF: 1.3, sd=0.26, with 87.5% of BFs > 1). The measure of the BF did not correlate with the average threshold variability (pearson correlation coefficient r=-0.05).

Figure 6 compares the convergence of the three VB methods. The use of VD for the RAE-DDM efficiently prevented overfitting, which would have been caused a rebound of the negative ELBO on a test dataset.

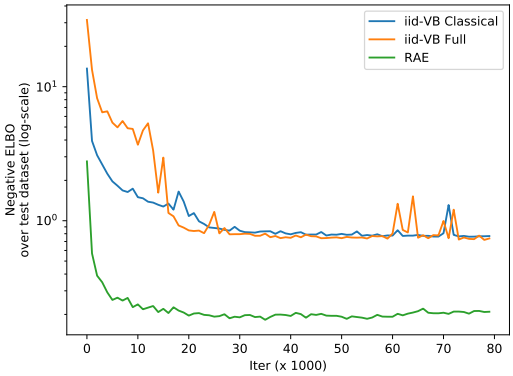


Figure 6: Convergence rate on a test dataset for the *i.i.d.*-VB and RAE-DDM approaches.

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 term of Pearson's correlation coefficients. Euclidian distance (or log-Euclidian distance for the variances) is reported in the Appendix Appendix D. All results are reported in an unbounded space (a, w are mapped with the inverse mapping described in Appendix A, and all statistics involving the variances were computed with the log-value of the variance). For the RAE-DDM, mean and variances were computed according to Equation (B.6). Finally, all the results that regard the drift rate were corrected for the motion direction for simplicity.

Population-wise fit. At the population level, each model performed well for the mean threshold and drift-rate. The start-point distribution was better estimated by the HMC approaches than VB, but, since this parameter was always assumed to be distributed around 0 at the population level, it is hard to interpret this result. The non-decision time was underestimated by each of the VB approaches, which might be due to the approximate posterior shape that was selected.

Quite interestingly, the HMC and *i.i.d.*-VB performed well in discriminating datasets with high and low variance of the a and w, which would not have been possible if the variance of these two parameters were equivalent. 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.

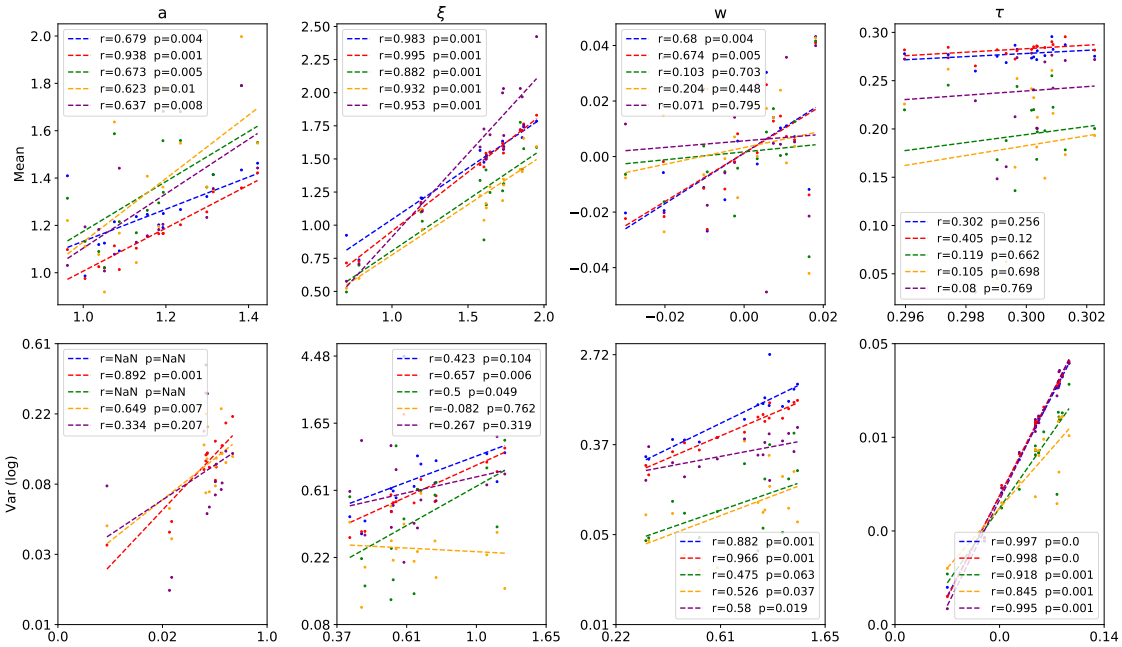


Figure 7: Population value correlations. **Blue**: HMC-Classical, **Red**: HMC-Full, **Green**: *i.i.d.*-VB -Classical, **Yellow**: *i.i.d.*-VB -Full, **Purple**: RAE-DDM. Fit of the Classical versions of HMC and *i.i.d.*-VB are lacking in the lower left panel, because these models did not account for threshold variability.

Subject-wise fit. Fit of the average parameter value at the subject level was accurately achieved for all mean parameters for all methods, with the notable exclusion of τ. In each of these assessments, the RAE-DDM outperformed 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 8 that the RAE was able to assess these variability with a reasonably high accuracy that was inaccessible to other methods. Second, the models (HMC of *i.i.d.*-VB) where the threshold

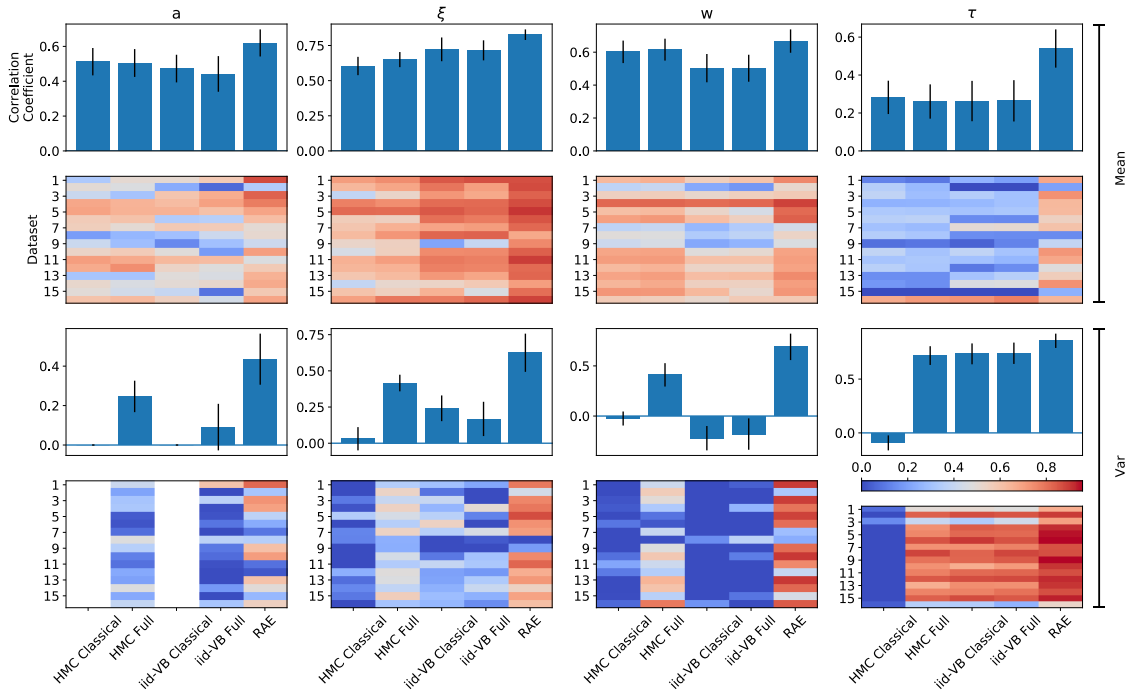


Figure 8: Subject value correlations. Heat plots show the value of the Pearson correlation coefficient for each dataset, for each model. The bar plots on top of these show the average of the corresponding head plot. Error bars show the value of one standard deviation.

variability were implemented did *not* perform worse than others: on the contrary, the HMC-Full uncovered better the variability of the drift, start 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.

Trial-wise fit. The RAE-RNN performed better at the trial level than each and every other model (Figure 9). Overall, VB and HMC performed equally well for all parameter. The correlation at the trial level of the parameter a was slightly lower than others. Whether this difference was due to a lower relative variability of this parameters with respect to 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 a at each trial with a relatively good accuracy (Figure 10).

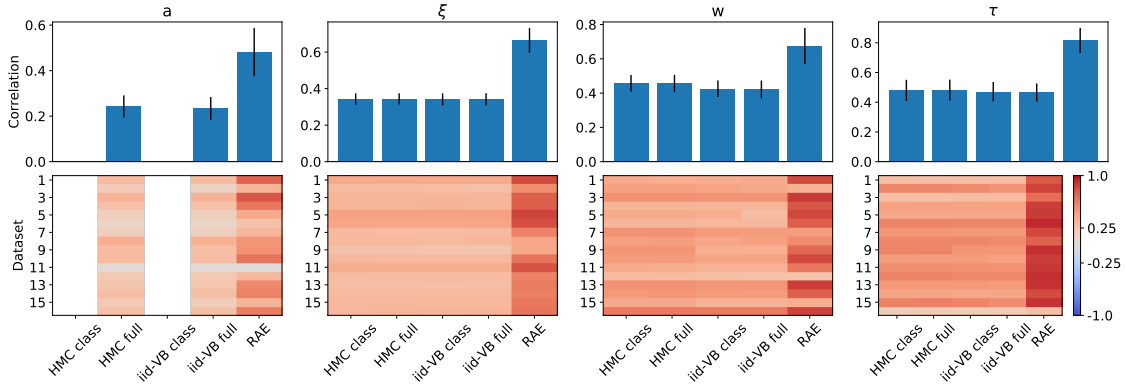


Figure 9: Trial-value correlations (averaged over subjects for each dataset).

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 Auto-encoding DDM (RAE-DDM) mainly relies on recent advances in Variational Inference. The main – and so to speak – only 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 Hamitlonian Monte Carlo (HMC) – by several orders of magnitude, and that inference can be efficiently achieved at every level of the hierarchy with correlation coefficients and distances between true and estimated values that were thought to be impossible to recover with previous approached.

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 Amortized Variational Inference, 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 dimentionality reduction, this model prevents overfitting.

It can account for the effect of regressors of any kind, without requiring the use to precisely set

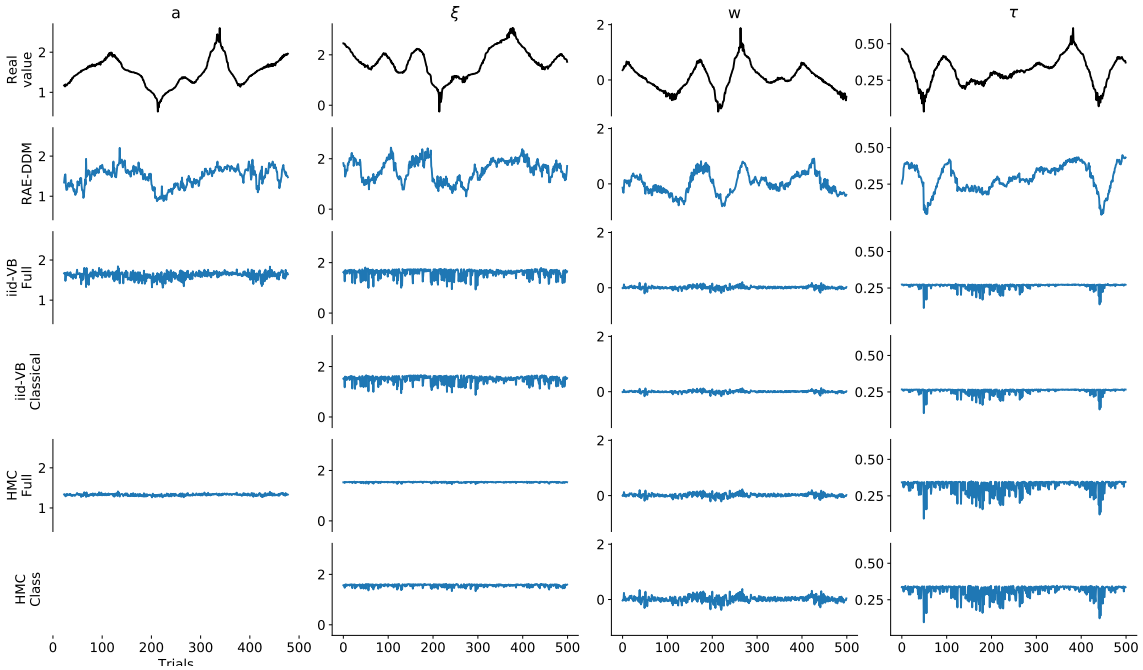


Figure 10: Randomly selected subject parameter values, and fit for each model. The RAE-DDM estimated the parameter trace with a high accuracy, that strikingly outperformed the other models. We emphasize the the fit of the drift (corrected for the motion in the figure) followed the motion direction pattern with a high accuracy even though this was not directly instructed in the model, whereas other parameters did not. This contrasts with the four other methods, where the effect of motion direction was explicitly implemented in the model.

the relation that ties these regressors to the parameters. Yet, it provides useful information about the parameters values: one can adequately classify the parameters fit *a posteriori* based on the input regressors, even though the approach is data-driven. Therefore, the RAE-DDM comprimises informativeness of the model with assumptions about the model specifications. Importantly, we do not claim that informative models should not be used: interpretation of the fit of the RAE-DDM might not be trivial. We suggest that, if an informative model needs to be tested, which will still be desirable in most instances, the universality of the RAE-DDM can provide a benchmark of the parameters fit 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 (5; 12; 26; 7; 28; 29; 61; 16), 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. The present method is the first, to our knowledge, in which this specific feature, which has been 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. 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 HMC of *i.i.d.*-VB models that accounted for this feature 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.

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 (103; 47; 126; 80), easy to use (127; 98; 88) and adapted to potentially very large problems (76; 53; 48). 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 (7) with a fast algorithm ; the second being to recover an estimate of the trial-to-trial dependency 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. 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 (128) or sub-thalamic nucleus stimulation (39)). 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 (129; 39; 130). 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 (SSM). Many of them have a closed-form FPT density function formula (131), allowing our models to extend to these other frameworks. Applicable models include the Collapsing Barrier DDM (132; 133) or the Time-varying Ornstein-Uhlenbeck model (134; 135; 1; 136).

Also, many appealing features could be implemented with the RAE-DDM to model neurophysiological measures together with behavioural data. Similarly to (68), 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 would probably greatly improve the performance and efficiency of the model.

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 exist some form of auto-correlation in the values of the parameters and/or data.

Acknowledgement

This work was performed at the Institute of Neuroscience (IoNS) of the Université catholique de Louvain (Brussels, Belgium); it was supported by grants from the ARC (Actions de Recherche Concerté es, Communauté Francaise de Belgique), from the Fondation Médicale Reine Elisabeth (FMRE), and from the Fonds de la Recherche Scientifique (F.R.S.-FNRS) to A.Z. A.Z. was a Senior Research Associate supported by INNOVIRIS and is currently 1st grade researcher for the french National Center for Scientific Research (CNRS).

This research used resources of the Center for High Performance Computing and Mass Storage (CISM, https://uclouvain.be/en/research/cism) located at the Université Catholique de Louvain, Belgium, which is supported by the Fonds de la Recherche Scientifique – Fonds National de la Recherche Scientifique (F.R.S.-FNRS). The CISM is member of the "Consortium des Équipements de Calcul Intensif (CÉCI)" (http://www.ceci-hpc.be).

Appendix A. 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 $\{a, \xi, w, \tau\}$ where $a \in \mathbb{R}^+ \triangleq f^{-1}(a')$, $w \in [0; 1] \triangleq g^{-1}(w')$, $\tau \in [-\infty; RT] \triangleq h^{-1}(\tau')$ and $\{a', \xi, w', \tau'\} \in \mathbb{R}$. We can construct a distribution

$$Wiener'(\mathbf{y}_{j,n}|a', \xi, w', \tau') \triangleq Wiener(\mathbf{y}_{j,n}|a, \xi, w, \tau)$$

that satisfies this condition for a, w 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 $\sigma(w') : \mathbb{R} \to [0; 1]$.

Appendix B. 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:

$$\begin{aligned} \boldsymbol{\Theta}^* &= \arg\max_{\boldsymbol{\Theta}} p(\mathbf{Y}; \boldsymbol{\Theta}) \\ &= \arg\max_{\boldsymbol{\Theta}} \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \ell(\mathbf{y}_{j,n}; \boldsymbol{\theta}_n) \\ \text{where } \ell(\mathbf{y}_{j,n}; \boldsymbol{\theta}_n) &= \log \int Wiener(\mathbf{y}_{j,n}; \boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}; \boldsymbol{\theta}_n) d\boldsymbol{\omega}_{j,n} \end{aligned} \tag{B.1}$$

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 $\boldsymbol{\theta}_n = \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2 I\}$, with I being the identity matrix (see Appendix A for justification and implementation).

In Equation (B.1), 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:

$$\begin{aligned} \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \ell(\mathbf{y}_{j,n}; \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) &\approx \mathbb{E}\left[\ell(\mathbf{y}_{j,n}; \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) | \mathbf{y}_{j,n}\right] \\ &= \hat{\ell}(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2). \end{aligned} \tag{B.2}$$

Since the integral in Equation (B.1) has no analytical form for a,w and τ, and since we need to evaluate it for every trial (12), 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 (B.2) is equal to 0:

$$\begin{aligned} \{\boldsymbol{\mu}_n{}^*, \boldsymbol{\sigma}_n^{2\,*}\} &= \underset{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2}{\arg\max} \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \ell(\mathbf{y}_{j,n}; \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) \\ \Leftrightarrow \frac{1}{NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} \nabla_{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2} \ell(\mathbf{y}_{j,n}; \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) &\Big|_{\boldsymbol{\mu}_n = \boldsymbol{\mu}_n{}^*, \boldsymbol{\sigma}_n^2 = \boldsymbol{\sigma}_n^{2\,*}} = 0 \end{aligned} \tag{B.3}$$

We can unpack this expression for each trial and make use of the Fisher's identity (137) to show that:

$$\begin{aligned} &\nabla_{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2} \log \int Wiener(\mathbf{y}_{j,n}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) d\boldsymbol{\omega}_{j,n} \\ &= \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)} \quad Wiener(\mathbf{y}_{j,n}|\boldsymbol{\omega}_{j,n}) d\boldsymbol{\omega}_{j,n}}{\int Wiener(\mathbf{y}_{j,n}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)} d\boldsymbol{\omega}_{j,n} \\ &= \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}_{j,n}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)}{\int Wiener(\mathbf{y}_{j,n}|\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} \\ &= \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}_{j,n}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n}|\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)}{p(\mathbf{y}_{j,n}|\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)} d\boldsymbol{\omega}_{j,n} \\ &= \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}_{j,n}, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) d\boldsymbol{\omega}_{j,n} \\ &= \mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n}, \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{B.4}$$

where we have used the Leibniz rule, the log-derivative trick and finally the Bayes theorem $p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n}) = \frac{p(\mathbf{y}_{j,n}|\boldsymbol{\omega}_{j,n}) p(\boldsymbol{\omega}_{j,n})}{\int p(\mathbf{y}_{j,n}, \boldsymbol{\omega}_{j,n}) d\boldsymbol{\omega}_{j,n}}$. Equation (B.4) shows that the DDM problem can be solved if we can compute (or approximate) the posterior probability $p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n}, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)$. Indeed, since the gradient formulated in Equation (B.4) must be zero at the mode, we can expand this formula and apply it to all trials

to show the following:

$$\begin{aligned}
&\frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}\left[\nabla_{\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2}\log p(\boldsymbol{\omega}_{j,n};\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)\right] &&= 0\\
\Leftrightarrow &\begin{cases}\frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}\left[-\frac{(\boldsymbol{\mu}_n-\boldsymbol{\omega}_{j,n})}{\boldsymbol{\sigma}_n^2}\right]\\ \frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}\left[\frac{(\boldsymbol{\mu}_n-\boldsymbol{\omega}_{j,n})^2}{\boldsymbol{\sigma}_n^3}-1/\boldsymbol{\sigma}_n^2\right]\end{cases} &&= 0.
\end{aligned} \tag{B.5}$$

Solving for $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n^2$, we have:

$$\begin{aligned}
{\boldsymbol{\mu}_n}^* &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}\left[\boldsymbol{\omega}_{j,n}\right]\\
{\boldsymbol{\sigma}_n^2}^* &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}\left[\boldsymbol{\omega}_{j,n}\right]^2+\mathbb{V}ar_{p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)}\left[\boldsymbol{\omega}_{j,n}\right]-({\boldsymbol{\mu}_n}^*)^2.
\end{aligned} \tag{B.6}$$

This can be viewed as an Empirical Bayes problem (138; 139) 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).

Equation (B.6) 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 $p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)$ 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 $p(\boldsymbol{\omega}_{j,n}|\mathbf{y}_{j,n},\boldsymbol{\mu}_n,\boldsymbol{\sigma}_n^2)$ 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 (B.6), 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.

Appendix C. Approximate posterior Specification

Appendix C.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 $\{\boldsymbol{\mu}_{j,n}^{\boldsymbol{\omega}},L_{j,n}^{\boldsymbol{\omega}}\}$ be the mean and lower-Cholesky decomposition of the amortized approximate posterior Gaussian distribution $q(\boldsymbol{\omega}_{j,n}\big|\mathbf{y}_{j,n},\boldsymbol{\theta}_n;\boldsymbol{\chi})$, where $\{\boldsymbol{\mu}_{j,n}^{\boldsymbol{\omega}},L_{j,n}^{\boldsymbol{\omega}}\}=f(\mathbf{y}_{j,n},\boldsymbol{\theta}_n;\boldsymbol{\chi})$, then:

$$\widetilde{\boldsymbol{\omega}}_{j,n,k}=\boldsymbol{\mu}_{j,n}^{\boldsymbol{\omega}}+L_{j,n}^{\boldsymbol{\omega}}\boldsymbol{\epsilon}\qquad\text{with }\boldsymbol{\epsilon}\sim\mathcal{N}(0,1).$$

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 ($\tau_{j,n} \in [-\infty; t_t]$), 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 ($\mathcal{N}_{-\infty,}(\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})$) 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 (106). 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 (108):

$$\nabla_{\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n}} g(\eta; \mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n}) = \frac{\nabla_{\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n}} P(\tau_{j,n} \big| \mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})}{p(\tau_{j,n} \big| \mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})} \tag{C.1}$$

where P is the CDF of $\mathcal{N}_{(-\infty,t]}(\mu^{\tau}_{j,n}, \sigma^{\tau}_{j,n})$ 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 (140)). We will introduce how this issue can be solve by the use of Normalizing Flows short after.

Appendix C.2. Normalizing Flow

If the deterministic mapping from the datapoint to the approximate posterior shape described in Section 2.4.2 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 (47). The core idea of Normalizing Flows is quite simple: it consists in defining an arbitrarily complex approximate posterior $q^0(\boldsymbol{\omega}_{j,n}|\boldsymbol{\delta}_{j,n}, \boldsymbol{\chi})$, where both $\boldsymbol{\delta}_{j,n}$ 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 $\boldsymbol{\omega}_0$:

$$q(\boldsymbol{\omega}_1) = q(\boldsymbol{\omega}_0) \left|\det \frac{\delta f^{-1}}{\delta \boldsymbol{\omega}_1}\right| = q(\boldsymbol{\omega}_0) \left|\det \frac{\delta f}{\delta \boldsymbol{\omega}_0}\right|^{-1} \tag{C.2}$$

and similarly for the following transformations. After K transformations, the ELBO to the model evidence looks like

$$\mathcal{L}(q(\boldsymbol{\omega}_K)) = \mathbb{E}_{q(\boldsymbol{\omega}_K)}\left[\log p(\mathbf{y}_{j,n}, \boldsymbol{\omega}_K)\right] - \mathbb{E}_{q(\boldsymbol{\omega}_K)}\left[\log q(\boldsymbol{\omega}_0) + \sum_{k=1}^{K} \log \left|\det \frac{\delta f_k}{\delta \boldsymbol{\omega}_{k-1}}\right|\right] \tag{C.3}$$
$$\text{where } \boldsymbol{\omega}_K = f_K \circ f_{K-1} ... f_1(\boldsymbol{\omega}_0)$$

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 (47).

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 $\{a, \xi, w\}$ 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 (47). This form of approximate posterior proved to be efficient in retrieving covariance between τ and the other parameters.

Appendix C.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 $t_{j,n}$.

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:

$$\begin{aligned}\mathcal{L}\left(q(\mathbf{z}_{j,n}|\mathbf{y}_{j,n}, \mathbf{h}_{j-1})\right) \approx \frac{1}{JS} \sum_{j,n}^{J,S} \log \frac{1}{K} \sum_{k} \exp \Big\{ & \log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k}|\mathbf{h}_{j-1}) \\ & - \log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k}|\mathbf{y}_{j,n}, \mathbf{h}_{j-1})) - \frac{1}{2} \max(0, \widetilde{\tau}_k - t_{j,n})^2 \Big\}\end{aligned} \tag{C.4}$$

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 $\widetilde{\tau}_k > t$ for some but not all

particles k, then the weight of these particles was set to 0 because $\log p_{\boldsymbol{\phi}}(\mathbf{y}_{j,n}, \widetilde{\mathbf{z}}_{j,n,k} \big| \mathbf{h}_{j-1}) \to -\infty$. 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:

$$\nabla_{\boldsymbol{\rho},\boldsymbol{\phi}} \left\{ \mathcal{L} \left(q(\mathbf{z}_{j,n} \big| \mathbf{y}_{j,n}, \mathbf{h}_{j-1}) \right) \right\} \approx \frac{1}{JS} \sum_{j,n}^{J,S} \log \frac{1}{K} \sum_{k} \nabla_{\boldsymbol{\rho},\boldsymbol{\phi}} \left\{ \exp \left\{ -\log q_{\boldsymbol{\rho}}(\widetilde{\mathbf{z}}_{j,n,k} \big| \mathbf{y}_{j,n}, \mathbf{h}_{j-1})) - \frac{1}{2} \max(0, \widetilde{\tau}_k - t_{j,n})^2 \right\} \right\}. \tag{C.5}$$

Appendix D. Euclidian Distances

We report here the Euclidian 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 each level.

At the population level (Figure D.11), the mean and variance estimates of the RAE-DDM were slightly more biased than the HMC estimates. This confirms the results shown in Figure 7 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 D.12. The same was true at the trial level Figure D.14.

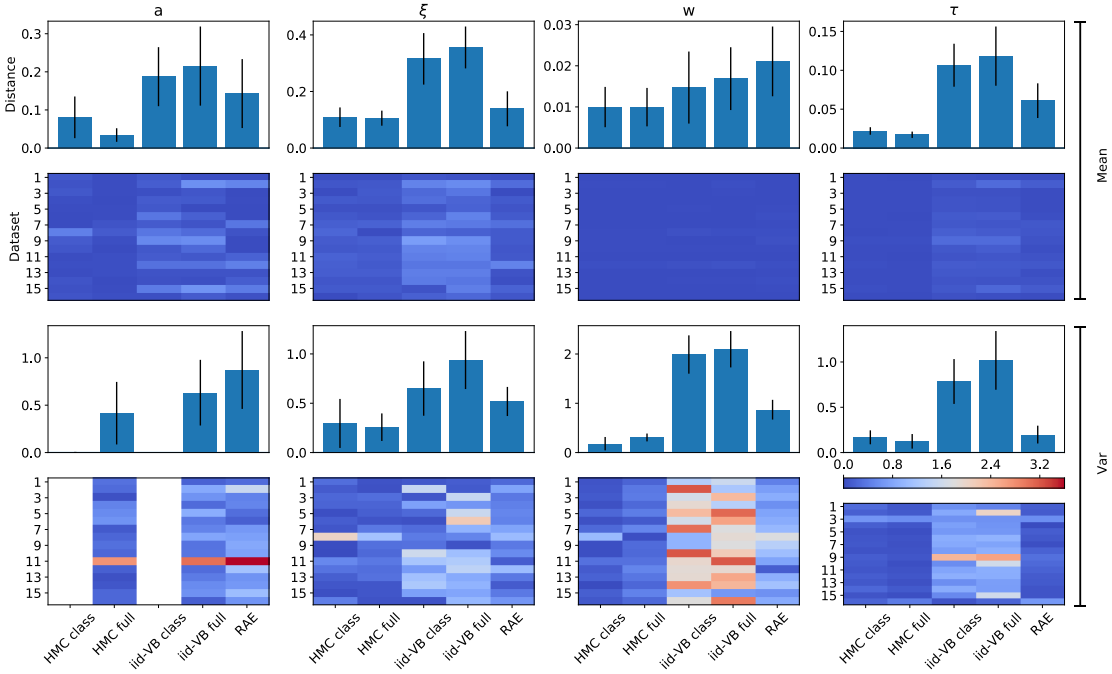


Figure D.11: Population true to estimate (log-)Euclidian distances.

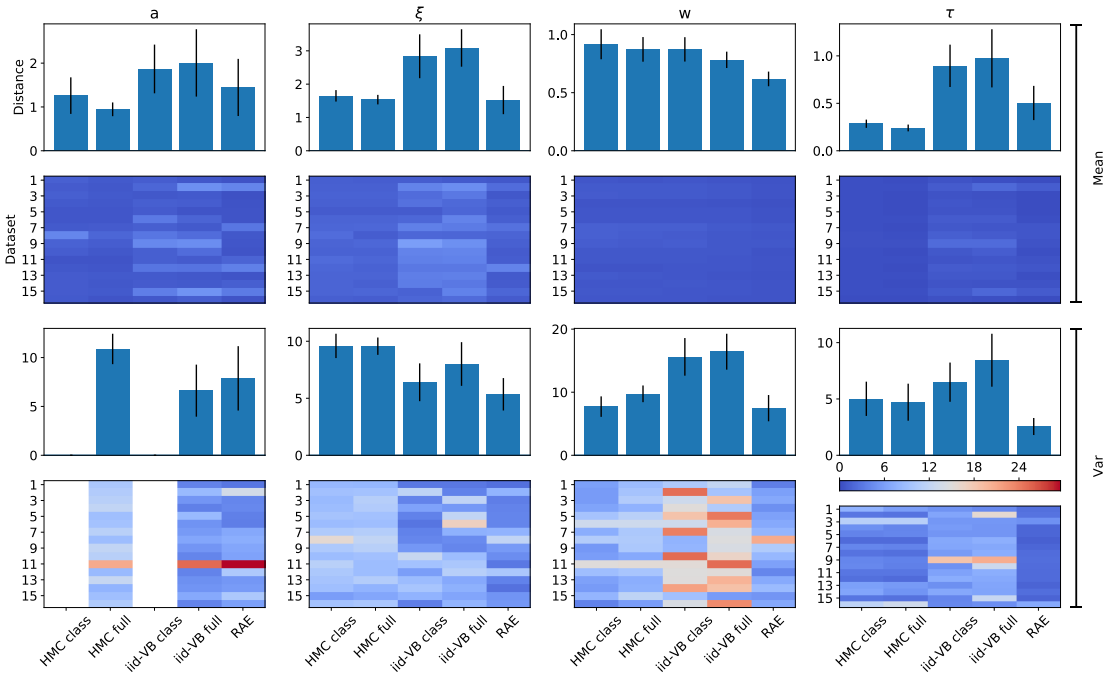


Figure D.12: Subject-wise true to estimate (log-)Euclidian sum of distances.

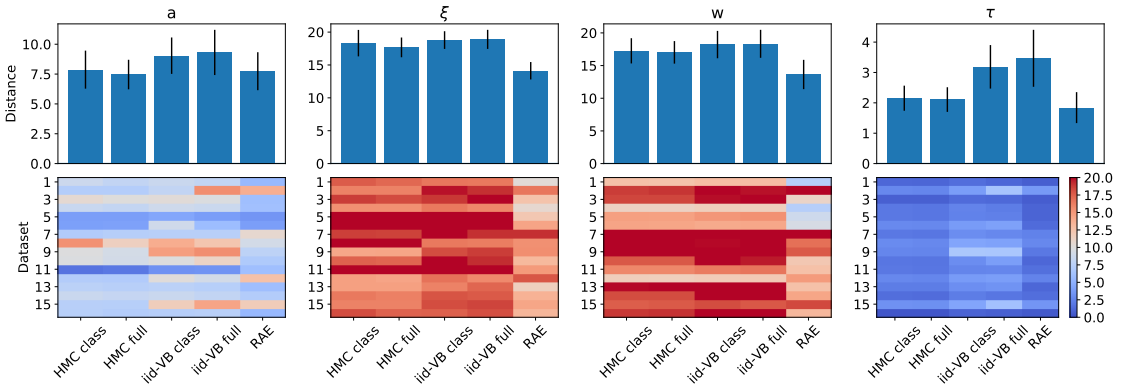


Figure D.13: Trial-wise true to estimate (log-)Euclidian sum of distances. The RAE-DDM performed equally well or better than other approaches.

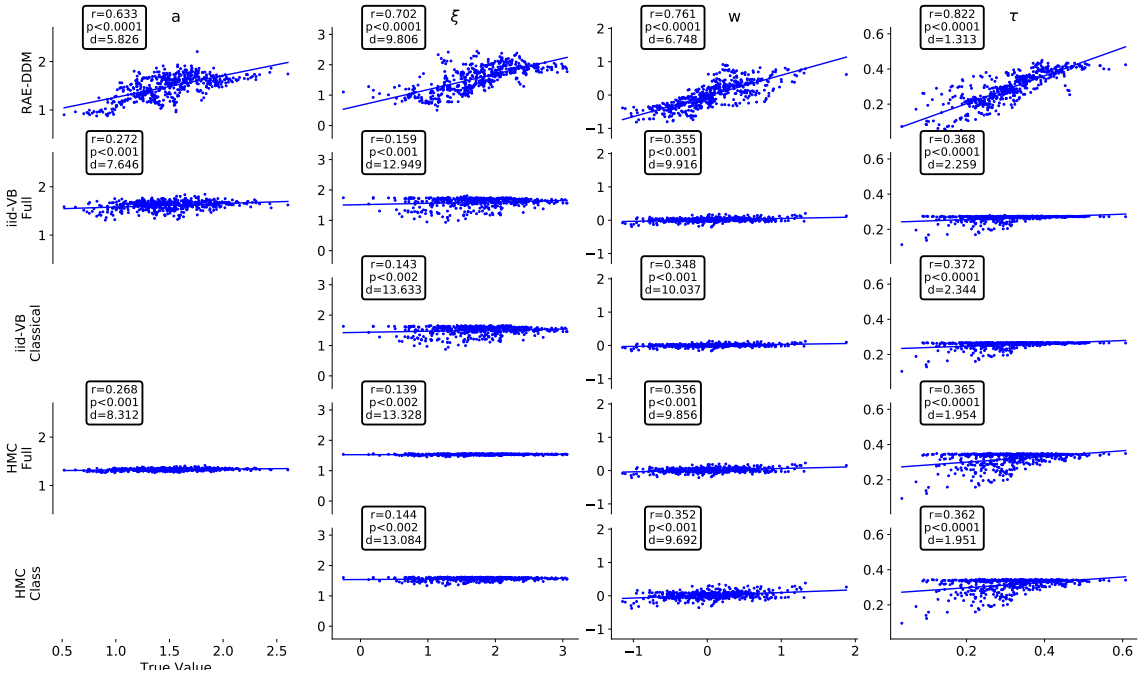


Figure D.14: Correlation and distances of the same random subject as in Figure 10. RAE-DDM showed a high accuracy for both these measures, displayed in the upper left corner of each panel. p-values state for the significance of the Pearson's correlation coefficient displayed above.

[1] Roger Ratcliff and Philip L Smith. A comparison of sequential sampling models for two-choice reaction time. *Psychological review*, 111(2):333–67, apr 2004.

[2] Roger Ratcliff. A theory of memory retrieval. *Psychological Review*, 85(2):59–108, 1978.

[3] Donald Laming. *Information theory of choice-reaction times.*, volume 14. 1968.

[4] Roger Ratcliff. Continuous versus discrete information processing modeling accumulation of partial information. *Psychological review*, 95(2):238–255, 1988.

[5] R. Ratcliff and J. N. Rouder. Modeling Response Times for Two-Choice Decisions. *Psychological Science*, 9(5):347–356, 1998.

[6] P L Smith and T Van Zandt. Time-dependent Poisson counter models of response latency in simple judgment. *The British journal of mathematical and statistical psychology*, 53 (Pt 2):293–315, nov 2000.

[7] Roger Ratcliff and Gail McKoon. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, 20:873–922, 2008.

[8] Eric-Jan Wagenmakers, Han L J van der Maas, and Raoul P P P Grasman. An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, 14(1):3–22, 2007.

[9] Rafal Bogacz, Eric Brown, Jeff Moehlis, Philip Holmes, and Jonathan D. Cohen. The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, 113(4):700–765, oct 2006.

[10] Thomas V Wiecki and Michael J Frank. A computational model of inhibitory control in frontal cortex and basal ganglia. *Psychological review*, 120(2):329–55, apr 2013.

[11] Roger Ratcliff, Philip L. Smith, Scott D. Brown, and Gail McKoon. Diffusion Decision Model: Current Issues and History. *Trends in Cognitive Sciences*, 20(4):260–281, 2016.

[12] Roger Ratcliff and Francis Tuerlinckx. Estimating parameters of the diffusion model: Approaching to dealing with contaminant reaction and parameter variability. *Psychonomic Bulletin & Review*, 9(3):438–481, 2002.

[13] Francis Tuerlinckx. The efficient computation of the cumulative distribution and probability density functions in the diffusion model. *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc*, 36(4):702–16, 2004.

[14] Roger Ratcliff. Parameter variability and distributional assumptions in the diffusion model. *Psychological Review*, 120(1):281–292, 2013.

[15] Thomas V Wiecki, Imri Sofer, and Michael J Frank. HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in neuroinformatics*, 7(August):14, 2013.

[16] Michael D. Nunez, Joachim Vandekerckhove, and Ramesh Srinivasan. How attention influences perceptual decision making: Single-trial EEG correlates of drift-diffusion model parameters. *Journal of Mathematical Psychology*, 76:117–130, 2017.

[17] Christopher Bishop. *Pattern Recognition And Machine Learning*. Springer-Verlag New York, 2006.

[18] Harriet R Brown, Peter Zeidman, Peter Smittenaar, Rick a Adams, Fiona McNab, Robb B Rutledge, and Raymond J Dolan. Crowdsourcing for cognitive science - the utility of smartphones. *PloS one*, 9(7):e100662, jan 2014.

[19] Fiona McNab, Peter Zeidman, Robb B. Rutledge, Peter Smittenaar, Harriet R. Brown, Rick A. Adams, and Raymond J. Dolan. Age-related changes in working memory and the ability to ignore distraction. *Proceedings of the National Academy of Sciences*, 112(20):6515–6518, 2015.

[20] Robb B. Rutledge, Peter Smittenaar, Peter Zeidman, Harriet R. Brown, Rick A. Adams, Ulman Lindenberger, Peter Dayan, and Raymond J. Dolan. Risk Taking for Potential Reward Decreases across the Lifespan. *Current Biology*, 26(12):1634–1639, 2016.

[21] Brandon M. Turner, Leendert van Maanen, and Birte U. Forstmann. Informing Cognitive Abstractions through Neurophysiology: The Neural Drift Diffusion Model. *Psychological Review*, 122(2):312–336, 2015.

[22] Roger Ratcliff. Measuring psychometric functions with the diffusion model. *Journal of Experimental Psychology: Human Perception and Performance*, 40(2):870–888, 2014.

[23] Veronika Lerche and Andreas Voss. Model complexity in diffusion modeling: Benefits of making the model more parsimonious. *Frontiers in Psychology*, 7(SEP):1–14, 2016.

[24] A Heathcote, S Brown, and D J K Mewhort. Quantile maximum likelihood estimation of response time distributions. *Psychonomic Bulletin & Review*, 9(2):394–401, 2002.

[25] Scott Brown and Andrew Heathcote. QMLE: Fast, robust, and efficient estimation of distribution functions based on quantiles. *Behavior Research Methods, Instruments, & Computers*, 35(4):485–492, nov 2003.

[26] Joachim Vandekerckhove and Francis Tuerlinckx. Fitting the Ratcliff diffusion model to experimental data. *Psychonomic bulletin & review*, 14(6):1011–1026, 2007.

[27] Joachim Vandekerckhove, Francis Tuerlinckx, and Michael D. Lee. A Bayesian approach to diffusion process models of decision-making. *Social Sciences*, (949):1429–1434, 2008.

[28] Joachim Vandekerckhove, Francis Tuerlinckx, and M. Lee. A Bayesian approach to diffusion process models of decision-making. *Proceedings of the 30th annual conference of the cognitive science society*, pages 1429–1434, 2008.

[29] Joachim Vandekerckhove, Francis Tuerlinckx, and Michael D Lee. Hierarchical diffusion models for two-choice response times. *Psychological methods*, 16(1):44–62, 2011.

[30] Andrew Gelman, John B Carlin, Hal S Stern, David B. Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis, Third Edition (Chapman & Hall/CRC Texts in Statistical Science)*. CRC, 2013.

[31] Nathaniel D Daw. Trial-by-trial data analysis using computational models. *Attention and Performance*, pages 1–26, 2009.

[32] Angela J Yu and Jonathan D Cohen. Sequential effects: Superstition or rational behavior? *Advances in Neural Information Processing Systems*, 21:1873–80, 2009.

[33] Juan Gao, KongFatt Wong-Lin, Philip Holmes, Patrick Simen, and Jonathan D. Cohen. Sequential Effects in Two-Choice Reaction Time Tasks: Decomposition and Synthesis of Mechanisms. *Neural Computation*, 21(9):2407–2436, sep 2009.

[34] M. H. Wilder, M. Jones, and M. C. Mozer. Sequential effects reflect parallel learning of multiple environmental regularities. *Advances in Neural Information Processing Systems*, 22:2053–61, 2009.

[35] Joshua Benjamin Miller and Adam Sanjurjo. Surprised by the Gambler's and Hot Hand Fallacies? A Truth in the Law of Small Numbers. *SSRN Electronic Journal*, pages 1–44, 2015.

[36] Gui Xue, Chi-Hung Juan, Chi-Fu Chang, Zhong-Lin Lu, and Qi Dong. Lateral prefrontal cortex contributes to maladaptive decisions. *Proceedings of the National Academy of Sciences of the United States of America*, 109(12):4401–6, mar 2012.

[37] 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. *Human Brain Mapping*, 37(3):1218–1234, 2016.

[38] Wim Notebaert, Femke Houtman, Filip Van Opstal, Wim Gevers, Wim Fias, and Tom Verguts. Post-error slowing: An orienting account. *Cognition*, 111(2):275–279, may 2009.

[39] James F Cavanagh, Thomas V Wiecki, Michael X Cohen, Christina M Figueroa, Johan Samanta, Scott J Sherman, and Michael J Frank. Subthalamic nucleus stimulation reverses mediofrontal influence over decision threshold. *Nature Neuroscience*, 14(11):1462–1467, 2011.

[40] Peter Dayan. Helmholtz machines and wake-sleep learning. *Handbook of Brain Theory and Neural Network. MIT* . . . , 44(0), 2000.

[41] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations 2015*, pages 1–15, 2015.

[42] Otto Fabius and Joost R. van Amersfoort. Variational Recurrent Auto-Encoders. (2013):1–5, dec 2014.

[43] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A Recurrent Latent Variable Model for Sequential Data. pages 1–9, jun 2015.

[44] Aaron Goodfellow, Ian, Bengio, Yoshua, Courville. Deep Learning. *MIT Press*, 2016.

[45] Chris J. Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering Variational Objectives. (Nips):1–11, 2017.

[46] Tuan Anh Le, Maximilian Igl, Tom Jin, Tom Rainforth, and Frank Wood. Auto-Encoding Sequential Monte Carlo. 2017.

[47] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. *Proceedings of the 32nd International Conference on Machine Learning*, 37:1530–1538, may 2015.

[48] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. (Nips), 2016.

[49] Jakub M. Tomczak and Max Welling. Improving Variational Auto-Encoders using Householder Flow. 2, 2016.

[50] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders. pages 1–14, 2015.

[51] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[52] D J Rezende, S Mohamed, and D Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of The 31st ...*, 32:1278–1286, 2014.

[53] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. (Mcmc):1–14, 2015.

[54] Trisha Van Zandt. How to fit a response time distribution. *Psychonomic Bulletin & Review*, 7(3):424–465, sep 2000.

[55] Andreas Voss, Jochen Voss, and Veronika Lerche. Assessing cognitive processes with diffusion model analyses: A tutorial based on fast-dm-30. *Frontiers in Psychology*, 6(MAR):1–14, mar 2015.

[56] Andreas Voss and Jochen Voss. A fast numerical algorithm for the estimation of diffusion model parameters. *Journal of Mathematical Psychology*, 52(1):1–9, feb 2008.

[57] 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. *Journal of Mathematical Psychology*, 49:195–204, 2005.

[58] 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. *Journal of Mathematical Psychology*, 53(2):55–68, apr 2009.

[59] Dominik Wabersich and Joachim Vandekerckhove. Extending JAGS: A tutorial on adding custom distributions to JAGS (with a diffusion model example). *Behavior Research Methods*, 46(1):15–28, 2014.

[60] Andrew Gelman, Daniel Lee, and Jiqiang Guo. Stan: A Probabilistic Programming Language for Bayesian Inference and Optimization. *Journal of Educational and Behavioral Statistics*, 40(5):530–543, 2015.

[61] Peter R. Murphy, Joachim Vandekerckhove, and Sander Nieuwenhuis. Pupil-Linked Arousal Determines Variability in Perceptual Decision Making. *PLoS Computational Biology*, 10(9), 2014.

[62] M. J. Frank, C. Gagne, E. Nyhus, S. Masters, T. V. Wiecki, J. F. Cavanagh, and D. Badre. fMRI and EEG Predictors of Dynamic Decision Parameters during Human Reinforcement Learning. *Journal of Neuroscience*, 35(2):485–494, jan 2015.

[63] Rafael Polanía, Marius Moisa, Alexander Opitz, Marcus Grueschow, and Christian C. Ruff. The precision of value-based choices depends causally on fronto-parietal phase coupling. *Nature Communications*, 6, 2015.

[64] Samuel R. Mathias. Unified analysis of accuracy and reaction times via models of decision making. volume 050001, page 050001, 2016.

[65] Mads Lund Pedersen, Michael J Frank, and Guido Biele. The drift diffusion model as the choice rule in reinforcement learning. *Psychonomic Bulletin & Review*, 24(4):1234–1251, aug 2017.

[66] Andriy Mnih and Karol Gregor. Neural Variational Inference and Learning in Belief Networks. *ArXiv stat.ML*, 32(October):1–20, 2014.

[67] Matt Jones, Tim Curran, Michael C Mozer, and Matthew H Wilder. Supplemental Material for Sequential Effects in Response Time Reveal Learning Mechanisms and Event Representations. *Psychological review*, 120(3):628–666, 2013.

[68] Brandon M. Turner, Birte U. Forstmann, Eric Jan Wagenmakers, Scott D. Brown, Per B. Sederberg, and Mark Steyvers. A Bayesian framework for simultaneously modeling neural and behavioral data. *NeuroImage*, 72:193–206, 2013.

[69] Brandon M. Turner, Birte U. Forstmann, Bradley C. Love, Thomas J. Palmeri, and Leendert Van Maanen. Approaches to analysis in model-based cognitive neuroscience. *Journal of Mathematical Psychology*, 76:65–79, 2017.

[70] Matt Jones and Ehtibar N Dzhafarov. Unfalsifiability and mutual translatability of major modeling schemes for choice reaction time. *Psychological review*, 121(1):1–32, 2014.

[71] Philip L. Smith, Roger Ratcliff, and Gail McKoon. The diffusion model is not a deterministic growth model: Comment on Jones and Dzhafarov (2014). *Psychological Review*, 121(4):679–688, 2014.

[72] Klaas Enno Stephan, Will D. Penny, Jean Daunizeau, Rosalyn J. Moran, and Karl J. Friston. Bayesian model selection for group studies. *NeuroImage*, 46(4):1004–1017, 2009.

[73] L. Rigoux, K. E. Stephan, K. J. Friston, and J. Daunizeau. Bayesian model selection for group studies - Revisited. *NeuroImage*, 84:971–985, 2014.

[74] William Feller. An Introduction to Probability Theory and Its Applications. *Wiley*, 2:509, 1968.

[75] Daniel J. Navarro and Ian G. Fuss. Fast and accurate calculations for first-passage times in Wiener diffusion models. *Journal of Mathematical Psychology*, 53(4):222–230, 2009.

[76] Matt Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. (2), 2012.

[77] a W van der Vaart. *Asymptotic Statistics*. 1998.

[78] Lucien Le Cam. *Asymptotic Methods in Statistical Decision Theory*. Springer Series in Statistics. Springer New York, New York, NY, 1986.

[79] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring. *International Conference on Machine Learning*, cs.LG, 2012.

[80] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic Gradient Descent as Approximate Bayesian Inference. *Mathematical Biosciences and Engineering*, 13(3):613–629, apr 2017.

[81] Karl Friston, Jérémie Mattout, Nelson Trujillo-Barreto, John Ashburner, and Will Penny. Variational free energy and the Laplace approximation. *NeuroImage*, 34(1):220–234, 2007.

[82] Christoph Mathys. A Bayesian foundation for individual learning under uncertainty. *Frontiers in Human Neuroscience*, 5(May):1–20, 2011.

[83] Kay H. Brodersen, Christoph Mathys, Justin R. Chumbley, Jean Daunizeau, Cheng Soon Ong, Joachim M. Buhmann, and Klaas E. Stephan. Bayesian mixed-effects inference on classification performance in hierarchical data sets. *Journal of Machine Learning Research*, 13:3133–3176, 2012.

[84] Christoph D Mathys, Ekaterina I Lomakina, Jean Daunizeau, Sandra Iglesias, Kay H Brodersen, Karl J Friston, and Klaas E Stephan. Uncertainty in perception and the Hierarchical Gaussian Filter. *Frontiers in Human Neuroscience*, 8(November):825, nov 2014.

[85] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, John O'Doherty, and Giovanni Pezzulo. Active inference and learning. *Neuroscience and Biobehavioral Reviews*, 68:862–879, 2016.

[86] Tommi S Jaakkola and Michael I Jordan. A variational approach to Bayesian logistic regression models and their extensions. *Aistats*, (AUGUST 2001), 1996.

[87] David M Blei. Variational Inference. *Cs.Princeton.Edu*, pages 1–12, 2002.

[88] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *arXiv*, pages 1–33, 2016.

[89] Masa-Aki Sato. Online Model Selection Based on the Variational Bayes. *Neural Comput.*, 13(7):1649–1681, 2001.

[90] John Winn, Cm Bishop, and T Jaakkola. Variational Message Passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

[91] David Knowles and Thomas P. Minka. Non-conjugate variational message passing for multinomial and binary regression. *Nips*, pages 1–9, 2011.

[92] James Hensman, Magnus Rattray, and Neil D. Lawrence. Fast Variational Inference in the Conjugate Exponential Family. pages 1–20, 2012.

[93] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. dec 2013.

[94] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *International Conference on Learning Representations*, pages 1–11, 2016.

[95] Michael C. Fu. Chapter 19 Gradient Estimation. In *Handbooks in Operations Research and Management Science*, volume 13, pages 575–616. 2006.

[96] Pierre L'Ecuyer. A Unified View of Infinitesimal Perturbation Analysis and Likelihood Ratios. Technical report, US Dept of the Army, feb 1989.

[97] P. Glasserman. Performance continuity and differentiability in Monte Carlo optimization. In *1988 Winter Simulation Conference Proceedings*, pages 518–524. IEEE.

[98] Francisco J. R. Ruiz, Michalis K. Titsias, and David M. Blei. Overdispersed Black-Box Variational Inference. 2016.

[99] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[100] L Bottou. Stochastic Gradient Learning in Neural Networks. *Proceedings of Neuro-Nımes*, 91(8), 1991.

[101] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[102] Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. dec 2012.

[103] Rajesh Ranganath, Dustin Tran, and David M Blei. Hierarchical Variational Models. *arXiv*, pages 1–9, 2014.

[104] Dustin Tran, Rajesh Ranganath, and David M. Blei. Variational Gaussian Process. *Iclr*, pages 1–14, 2015.

[105] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep Probabilistic Programming. pages 1–18, 2017.

[106] George Marsaglia. Generating a Variable from the Tail of the Normal Distribution. *Technometrics*, 6(1):101, feb 1964.

[107] George Marsaglia and Wai Wan Tsang. A simple method for generating gamma variables. *ACM Transactions on Mathematical Software*, 26(3):363–372, 2000.

[108] David A. Knowles. Stochastic gradient variational Bayes for gamma approximating distributions. pages 1–14, 2015.

[109] Christian A. Naesseth, Francisco J. R. Ruiz, Scott W. Linderman, and David M. Blei. Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms. 54, 2016.

[110] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational Dropout Sparsifies Deep Neural Networks. 2017.

[111] Yarin Gal and Zoubin Ghahramani. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. 19994575, dec 2015.

[112] Martijn J Mulder, Eric-Jan Wagenmakers, Roger Ratcliff, Wouter Boekel, and Birte U Forstmann. Bias in the brain: a diffusion model analysis of prior probability and potential payoff. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 32(7):2335–43, feb 2012.

[113] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.*, volume 14. 2004.

[114] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23. 1992.

[115] Stijn Verdonck, Kristof Meers, and Francis Tuerlinckx. Efficient simulation of diffusion-based choice RT models on CPU and GPU. *Behavior Research Methods*, mar 2015.

[116] Richard H. R. Hahnloser, Rahul Sarpeshkar, Misha A. Mahowald, Rodney J. Douglas, and H. Sebastian Seung. Erratum: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, jun 2000.

[117] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, (3):807–814, 2010.

[118] Yuhuang Hu, Adrian Huber, Jithendar Anumula, and Shih-Chii Liu. Overcoming the vanishing gradient problem in plain recurrent networks. (Section 2):1–20, 2018.

[119] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 2014.

[120] Sepp Hochreiter and J Urgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[121] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, jan 2017.

[122] Vincent Moens. Hierarchical Adaptive Forgetting Variational Filtering. *In prep.*, 2018.

[123] John Langford, Alexander J. Smola, and Martin Zinkevich. Slow Learners are Fast. *Advances in Neural Information Processing Systems - NIPS'09*, 1(2099):2331—-2339, 2009.

[124] Aki Vehtari, Andrew Gelman, and Jonah Gabry. Pareto Smoothed Importance Sampling. (October), 2015.

[125] Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but Did It Work?: Evaluating Variational Inference. pages 1–22, 2018.

[126] Tim Salimans, Diederik P Kingma, and Max Welling. Markov Chain Monte Carlo and Variational Inference : Bridging the Gap. *International Conference on Machine Learning*, 2015.

[127] Rajesh Ranganath, Sean Gerrish, and David M Blei. Black Box Variational Inference. *Aistats*, 33, 2013.

[128] Yu-Ting Huang, Dejan Georgiev, Tom Foltynie, Patricia Limousin, Maarten Speekenbrink, and Marjan Jahanshahi. Different effects of dopaminergic medication on perceptual decision-making in Parkinson's disease as a function of task difficulty and speed–accuracy instructions. *Neuropsychologia*, 75:577–587, aug 2015.

[129] James F. Cavanagh, Michael J. Frank, Theresa J. Klein, and John J.B. B Allen. Frontal theta links prediction errors to behavioral adaptation in reinforcement learning. *NeuroImage*, 49(4):3198–3209, 2010.

[130] James F. Cavanagh and Michael J. Frank. Frontal theta as a mechanism for cognitive control. *Trends in Cognitive Sciences*, 18(8):414–421, aug 2014.

[131] Vaibhav Srivastava, Samuel F. Feng, Jonathan D. Cohen, Naomi Ehrich Leonard, and Amitai Shenhav. A martingale analysis of first passage times of time-dependent Wiener diffusion models. *Journal of Mathematical Psychology*, 77:94–110, 2017.

[132] Peter I. Frazier and Angela J Yu. Sequential hypothesis testing under stochastic deadlines. *Advances in Neural Information Processing Systems*, pages 1–8, 2008.

[133] 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. *Judgment and Decision Making*, 5(6):437–449, 2010.

[134] David Cox and Hilton David Miller. *The Theory of Stochastic Processes*. 1965.

[135] Mario Lefebvre. First hitting place distributions for the Ornstein-Uhlenbeck process. *Statistics & Probability Letters*, 34(3):309–312, jun 1997.

[136] Adele Diederich. Dynamic Stochastic Models for Decision Making under Time Constraints. *Journal of Mathematical Psychology*, 41(3):260–274, sep 1997.

[137] Olivier Cappé, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models*. 2006.

[138] Herbert Robbins. The Empirical Bayes Approach to Statistical Decision Problems. *The Annals of Mathematical Statistics*, 35(1):1–20, 1964.

[139] Manfred Opper and Cédric Archambeau. The Variational Gaussian Approximation Revisited. *Neural Computation*, 21(3):786–792, 2009.

[140] Z. I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148, jan 2016.