Variational Treatment of Trial-by-Trial Drift-Diffusion Models of Behaviour

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

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

[b] *INCIA*
Université de Bordeaux, France

Abstract

Drift-Diffusion Models (DDM) are widely used to model behaviour. Accounting for trial-to-trial variations of the parameters in DDM leads to better fits and allows researchers to investigate variations of parameters over time. However, parameter estimation with conventional integration techniques is slow and cannot be applied to large problems. Here we propose to rely on Variational Bayesian inference to estimate the posterior probability of the parameters at the local (trial) and global (subject and population) levels. We present a data-driven model that uses Inference Networks to estimate an approximate distribution for each of the local parameters given an arbitrary set of regressors. We apply the method to simulated and real datasets and show that it converges within reasonable time frame, achieves good fitting performance and highlights complex auto-correlation structures between trials. Using simulations, we show that our approach outperforms the gold-standard Linear-Regression Model, estimated both using Variational Inference and Markov-Chain Monte-Carlo method. Therefore, the present Bayesian approach, relying on variational inference networks, is a promising method to solve trial-by-trial DDM.

Keywords: Drift-Diffusion Model, Variational Bayes, Inference Network, Recurrent Neural Network, Data-driven behavioral analysis

1. Introduction

Sequential sampling models (SSM) (1) are models of behaviour assuming that decisions are the result of a noisy accumulation of evidence. The most famous of these models called the Drift Diffusion Model (DDM), based on a Wiener process, 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 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

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

account for some specific patterns of reaction times (RT), which led to subsequent improvements (3; 2; 4; 5; 6) 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 (7), Extended DDM (8) or Full DDM (9).

While the Full DDM is more realistic, it is also much more complex and computationally expensive. So far, most methods used to fit the Full DDM to behavioural datasets have relied on Numerical Quadrature (NQ) (10; 11; 12; 13; 14). However, NQ is computationally expensive, which precludes its use for large problems. Typically, only trial-to-trial variability in three of the four parameters of the DDM are assessed (drift, start-point and non-decision time). Also, large datasets (e.g. > 100 trials per subject) take an unreasonable amount of time to be optimized.

Another issue with many implementations of the DDM lies in the core assumption of the method used to fit the parameters to the data. Usually, a Maximum Likelihood Estimate (MLE; i.e. finding the most likely value parameters would endorse in order to generate the data at hand) is used (10; 15; 16; 11; 17; 18; 12; 13), although a few Bayesian attempts have been made (19; 20; 14). 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 (21). 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.

In this paper, we will show that the Full DDM finds a natural formulation in Bayesian Statistics (Section 2.1 and Section 2.2), and that we can use this formulation to solve efficiently this problem using recent advances in Variational Inference (Section 2.3). This method will be first implemented in a simple case of (non-)linear Regression Model (Section 2.4). Next, we will show how the temporal, trial-to-trial dependency between the parameters of the DDM can be exploited to find a closer fit to the generative parameters (Section 2.5). Section 3 provides some experimental results using simulated data of low and high complexity, and a real-life example. In Section 4, we will discuss our model in detail, highlight situations where it might prove useful, see its limitations and introduce potential improvements .

2. Methods

2.1. DDM

The Drift-Diffusion model is a member of the SSM family. The basic idea 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 choice C_i and RT_i is formulated as the probability that the evidence reached the threshold b at time t. This accumulation of evidence is stochastic: given the same initial state, two agents will produce a different response time and, possibly, choose a different item because the evidence accumulation process will be different. This stochasticity is the main asset of this model, since it can explain speed-accuracy trade-offs: indeed, certain parameter settings will produce faster but more chaotic response distributions, whereas other will produce slower, more precise responses.

Formally, in the DDM, the evidence accumulation process is assumed to follow a Wiener process (Figure 1):

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

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 v, called the drift, meaning that in the case of a classical Two-alternative forced choice (TAFC) task, he will also accumulate evidence if favour of the alternative action C_2 with a rate $-v$. 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 delay t_0, this action is executed. The evidence accumulation can start wherever at $z_0 \in [0; a]$, although it is more convenient to model the relative start point $w \in [0; 1]$ where $w = z_0/a$. Finally, this 5-parameter model is non-metric: multiplying v, a 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.

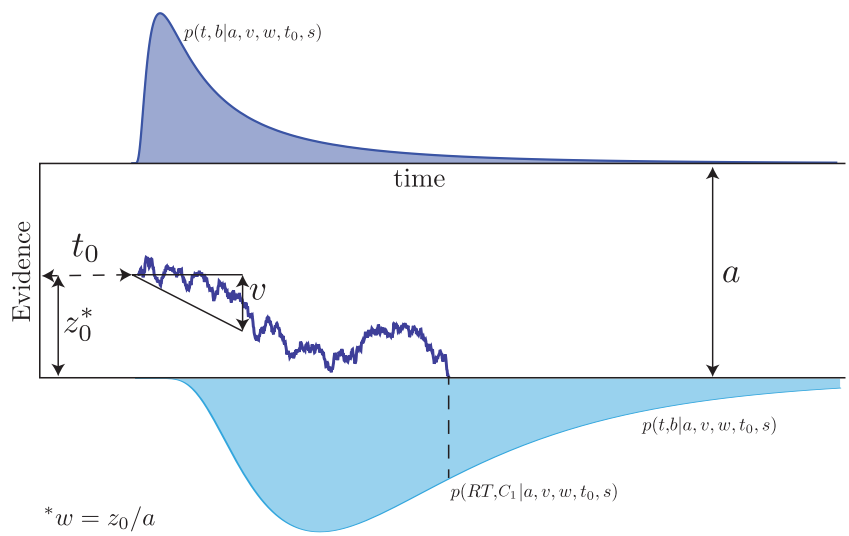

Figure 1: DDM structure. The x axis represents the time scale, the y axis the evidence level. The four parameters (threshold a, drift v, starting point w and nondecision time t_0) are represented. At the two boundaries, the first passage probability density function is represented.

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, if $\mathbf{z} = \{a, v, w, t_0\}$ is the vector of the DDM parameters, 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 (22):

$$Wiener(t, b; \mathbf{z}) = \begin{cases} \frac{\pi}{a^2} \exp(-vaw - \frac{v^2(t-t_0)}{2}) \sum_{k=1}^{\infty} k \exp(-\frac{k^2\pi^2(t-t_0)}{2a^2}) \sin(k\pi w) \text{ if } t > t_0 \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

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

$$\sum_{\beta \in \{b, \neg b\}} \int_{t=t_0}^{\infty} Wiener(t, \beta; \mathbf{z}) dt = 1$$
$$\text{and } Wiener(t, \beta; \mathbf{z}) \geq 0 \text{ for t} \in [t_0, \infty] \text{ and } \beta \in b, \neg b. \tag{2}$$

The function in Equation (1) involves an infinite sum, and must therefore be implemented through approximations. Navarro et al. (23) 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 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 parameters vary from trial-to-trial, the fit of the average of these parameters will suffer from some specific, consistent biases that can have an important impact on the interpretation (5). For instance, a high variability in the value of the starting point with mean .5 will generally produce a dataset that will be better fitted by a point-estimate $w^* < .5$. Accounting for the trial-to-trial variability in w with distribution parameters $\boldsymbol{\theta^w}$ such that $w \sim p(w; \boldsymbol{\theta^w})$ solves this problem.

2.2. The Full DDM as a Bayesian model

The MLE of the Full DDM as formulated by (10) consists in finding the value $\boldsymbol{\theta}^*$ that maximizes the likelihood function $p(\mathbf{y}; \boldsymbol{\theta})$ for $\mathbf{y} = \{RT_{t,s}, C_{t,s} \, \forall s, t\}$ (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 $\boldsymbol{\theta}$. This choice can be justified from the asymptotic normality of the posterior distribution theorem (21): 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 the true value $\boldsymbol{\theta}_0$ and covariance matrix equal to the inverse of the Observed Fisher Information matrix, i.e. $\boldsymbol{\theta}^* \to \boldsymbol{\theta}_0$ as $N \to \infty$ and $p(\boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{\theta}^*, I(\boldsymbol{\theta}^*)^{-1})$, where $I(\boldsymbol{\theta}^*) = -\nabla^2_{\boldsymbol{\theta}^*} \log p(D; \boldsymbol{\theta})$. Also, following the Bernstein-von Mises theorem (24; 25; 26), 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.

When maximizing a likelihood function, it is usually easier to work with its log transform, because it alleviates the need for computing the product of possibly small values, thereby avoiding possible computational underflow issues. Moreover, it is a valid approach since the log function is monotonically increasing, and therefore the log-likelihood function shares the same maxima as the likelihood

function. If we are to use the MLE method for the Full DDM for a set of S subjects with T trials per subject, we therefore need to solve the following optimization problem:

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(\mathbf{y};\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \frac{1}{S}\sum_{s=1}^{S}\frac{1}{T}\sum_{t=1}^{T}\ell(\mathbf{y}_{t,s};\boldsymbol{\theta}_s) \\ \text{where } \ell(\mathbf{y}_{t,s};\boldsymbol{\theta}_s) &= \log\int Wiener(\mathbf{y}_{t,s};\mathbf{z})p(\mathbf{z}_t;\boldsymbol{\theta}_s)d\,\mathbf{z}\end{aligned} \tag{3}$$

For simplicity, we will consider that the DDM parameters all lie in an unbounded space, so that their prior probability can be formulated as a Gaussian with parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ (see Appendix A for justification and implementation).

In Equation (3), we have used the average log-likelihood with the purpose of showing that, for a sufficiently large T, this average log-likelihood takes the form of a sample of the true expected log-likelihood function:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\ell(\mathbf{y}_t;\boldsymbol{\mu},\boldsymbol{\sigma}) &\approx \mathbb{E}[\ell(\mathbf{y}_t;\boldsymbol{\mu},\boldsymbol{\sigma})|\,\mathbf{y}] \\ &= \hat{\ell}(\boldsymbol{\mu},\boldsymbol{\sigma}).\end{aligned} \tag{4}$$

Since the integral in Equation (3) has no analytical form for a,w and t_0, and since we need to evaluate it for every trial (10), finding the MLE of $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ 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 (4) is equal to 0:

$$\begin{aligned}\{\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*\} &= \arg\max_{\boldsymbol{\mu},\boldsymbol{\sigma}} \frac{1}{T}\sum_{t=1}^{T}\ell(\mathbf{y}_t;\boldsymbol{\mu},\boldsymbol{\sigma}) \\ \Leftrightarrow \frac{1}{T}\sum_{t=1}^{T}\nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}}\ell(\mathbf{y}_t;\boldsymbol{\mu},\boldsymbol{\sigma})\Big|_{\boldsymbol{\mu}=\boldsymbol{\mu}^*,\boldsymbol{\sigma}=\boldsymbol{\sigma}^*} &= 0\end{aligned} \tag{5}$$

We can unpack this expression for each trial and make use of the Fisher's identity (27) to show

that:

$$
\begin{aligned}
\nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} \log \int Wiener(\mathbf{y}_t \,|\, \mathbf{z}_t) p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}) dz &= \frac{\int \overbrace{\nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma})}^{p(\mathbf{z}_t \,|\, \boldsymbol{\mu},\boldsymbol{\sigma}) \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} \log p(\mathbf{z}_t \,|\, \boldsymbol{\mu},\boldsymbol{\sigma})} Wiener(\mathbf{y}_t \,|\, \mathbf{z}_t) d\,\mathbf{z}_t}{\int Wiener(\mathbf{y}_t \,|\, \mathbf{z}_t) p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}) d\,\mathbf{z}_t} \\
&= \int \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} \log p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}) \frac{Wiener(\mathbf{y}_t \,|\, \mathbf{z}_t) p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma})}{\int Wiener(\mathbf{y}_t \,|\, \mathbf{z}_t) p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}) d\,\mathbf{z}_t} d\,\mathbf{z}_t \\
&= \int \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} \log p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}) \frac{Wiener(\mathbf{y}_t \,|\, \mathbf{z}_t) p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma})}{p(\mathbf{y}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma})} d\,\mathbf{z}_t \\
&= \int \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} \log p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma}) d\,\mathbf{z}_t \\
&= \mathbb{E}_{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})} [\nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} \log p(\mathbf{z}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma})]
\end{aligned}
\tag{6}
$$

where we have used the Leibniz rule, the log-derivative trick and finally the Bayes theorem $p(\mathbf{z}_t \,|\, \mathbf{y}_t) = \frac{p(\mathbf{y}_t \,|\, \mathbf{z}_t) p(\mathbf{z}_t)}{\int p(\mathbf{y}_t, \mathbf{z}_t) d\,\mathbf{z}_t}$. Equation (6) shows that the DDM problem can be solved if we can compute (or approximate) the posterior probability $p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})$. Indeed, since the gradient formulated in Equation (6) must be zero at the mode, we can expand this formula and apply it to all trials to show the following:

$$
\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})} \left[\nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} \log p(\mathbf{z}_t; \boldsymbol{\mu}, \boldsymbol{\sigma})\right] && = 0 \\
&\Leftrightarrow \begin{cases} \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})} \left[-\frac{(\boldsymbol{\mu} - z)}{\boldsymbol{\sigma}^2} \right] \\ \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})} \left[\frac{(\boldsymbol{\mu} - z)^2}{\boldsymbol{\sigma}^3} - 1/\boldsymbol{\sigma} \right] \end{cases} && = 0.
\end{aligned}
\tag{7}
$$

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

$$
\begin{aligned}
\boldsymbol{\mu}^* &= \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})} [\mathbf{z}_t] \\
\boldsymbol{\sigma}^{2^*} &= \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})} [\mathbf{z}_t]^2 + \mathbb{V}\mathrm{ar}_{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})} [\mathbf{z}_t] - (\boldsymbol{\mu}^*)^2.
\end{aligned}
\tag{8}
$$

Equation (8) shows that $\boldsymbol{\mu}^*$ and $\boldsymbol{\sigma}^{2^*}$ will have a closed form formula as long as we can estimate the first and second moment of $p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})$ for each trial. This cross-dependency between the posterior probability over $\mathbf{z}$ and the value of $\boldsymbol{\mu}^*$ and $\boldsymbol{\sigma}^{2^*}$ suggests an expectation maximization (EM) algorithm (Algorithm 1): we can iteratively estimate the posterior $p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma})$ under the current belief of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, and then, we can estimate $\boldsymbol{\mu}^*$ and $\boldsymbol{\sigma}^*$ using Equation (8), until some convergence criterion is reached. This can be viewed as an Empirical Bayes problem (28) where a MLE of the prior distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ has to be retrieved from the data. Using a Hierarchical Bayesian terminology, $\mathbf{z}_t$ is the latent (or local) variable and $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ are the prior or global parameters

(Figure 2).

Algorithm 1: Variational Expectation Maximization for the DDM

Data: Choices and reaction times $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T\}$, initial guess of $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$
Result: MLE of $\{\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*\}$
repeat
  Estimate posterior probability $\left\{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma}) \,\middle|\, t \in 1:T\right\}$ using VB ; `// E step (Section 2.3)`
  Estimate $\{\boldsymbol{\mu}', \boldsymbol{\sigma}'\} \,\middle|\, \left\{p(\mathbf{z}_t \,|\, \mathbf{y}_t, \boldsymbol{\mu}, \boldsymbol{\sigma}) \,\middle|\, t \in 1:T\right\}$; `// M step (Equation (8))`
  Set $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}'$, $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}'$
until *convergence*;

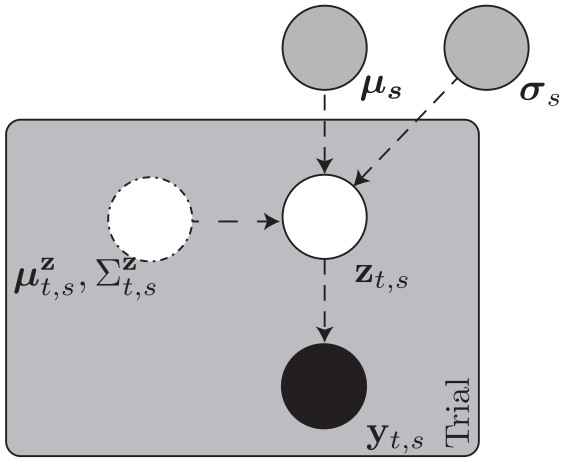

Figure 2: Directed Acyclic Graphical Model of the VEM method. The dashed circle stand for the approximate posterior of the local DDM parameters, grouped under the general indicator $\boldsymbol{\mu}^{\mathbf{z}}$ and $\boldsymbol{\sigma}^{\mathbf{z}}$. The plain circle is the observed variable $\mathbf{y}_t$. The empty circle represents a latent variable, the dashed circle the approximate posterior parameters and the plain, grey circle on top are MLE estimates of the mean and standard deviation at the subject level.

2.3. Variational Inference

The EM algorithm exposed above still relies on computing the posterior over $\mathbf{z}_t$, which is intractable due to its normalizing factor (i.e. model evidence) $p(\mathbf{y}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma})$. A first approach could be to use simulation techniques to approximate this posterior, but the size of the problem precludes this choice, since we would need to compute T independent integrals at each step of the EM algorithm. Instead, we can make use of Variational Bayes techniques (VB) (29; 30; 31; 32) 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 Expectation Maximisation.

VB works as follows: we first define a proxy to the posterior of an arbitrary shape $q(\mathbf{z}; \boldsymbol{\delta})$, where $\boldsymbol{\delta}$ is a general indicator of the variational parameters that we will optimize in order to match the real posterior as close as possible. In order to do so, we can choose to minimize the Kullback-Leibler

(KL) divergence between this distribution and the true posterior, since

$$\begin{aligned} q(\mathbf{z}) &= p(\mathbf{z} \,|\, \mathbf{y}) \\ \Leftrightarrow \mathrm{D}_{\mathrm{KL}}[q(\mathbf{z};\boldsymbol{\delta})||p(\mathbf{z} \,|\, \mathbf{y})] &= 0 \\ &\triangleq \int q(\mathbf{z};\boldsymbol{\delta}) \log q(\mathbf{z};\boldsymbol{\delta}) dz - \int q(\mathbf{z};\boldsymbol{\delta}) \log p(\mathbf{z} \,|\, \mathbf{y}) dz \end{aligned} \tag{9}$$

We can expand $\log p(\mathbf{z} \,|\, \mathbf{y}) = \log p(\mathbf{y}, \mathbf{z}) - \log p(\mathbf{y})$, and, considering that $\int q(\mathbf{z}) \log p(\mathbf{y}) dz = \log p(\mathbf{y})$ and that $-\int q(\mathbf{z};\boldsymbol{\delta}) \log q(\mathbf{z};\boldsymbol{\delta}) dz \triangleq H[q(\mathbf{z};\boldsymbol{\delta})]$ is the differential entropy of $q(\mathbf{z};\boldsymbol{\delta})$, we have:

$$\arg\min_{\boldsymbol{\delta}} \mathrm{D}_{\mathrm{KL}}[q(\mathbf{z};\boldsymbol{\delta})||p(\mathbf{z} \,|\, \mathbf{y})] = \arg\min_{\boldsymbol{\delta}} \underbrace{-H[q(\mathbf{z};\boldsymbol{\delta})] - \int q(\mathbf{z};\boldsymbol{\delta}) \log p(\mathbf{y}, \mathbf{z}) dz}_{-\mathcal{L}(q(\mathbf{z};\boldsymbol{\delta}))} + \log p(\mathbf{y}) \tag{10}$$

which implies that

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}[q(\mathbf{z};\boldsymbol{\delta})||p(\mathbf{z} \,|\, \mathbf{y})] &= 0 \\ \Leftrightarrow \mathcal{L}(q(\mathbf{z};\boldsymbol{\delta})) &= \log p(\mathbf{y}) \end{aligned} \tag{11}$$

and therefore $\mathcal{L}(q(\mathbf{z};\boldsymbol{\delta})) = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\delta})}[\log p(\mathbf{y}, \mathbf{z})] + H[q(\mathbf{z};\boldsymbol{\delta})]$ is a lower bound to the model evidence (or ELBO, for Evidence Lower Bound, also sometimes called negative Free Energy). Equation (11) tells us that the value of $\boldsymbol{\delta}$ that minimizes $\mathrm{D}_{\mathrm{KL}}[q(\mathbf{z};\boldsymbol{\delta})||p(\mathbf{z} \,|\, \mathbf{y})]$ is the one that maximizes the ELBO $\mathcal{L}(q(\mathbf{z};\boldsymbol{\delta}))$. Once this value of $\boldsymbol{\delta}$ has been optimized, we can consider that $q(\mathbf{z};\boldsymbol{\delta})$ is as close as it could from $p(\mathbf{z} \,|\, \mathbf{y})$ given the distribution shape we have chosen, the prior probability $p(\mathbf{z})$, the data $\mathbf{y}$ and the likelihood function $p(\mathbf{y} \,|\, \mathbf{z})$.

Optimizing the approximate posterior for each of the local variables is not trivial, as this posterior can have a complex shape, and the initial draw of $\{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, ..., \boldsymbol{\delta}_t, ... \boldsymbol{\delta}_T\}$, together with the choice of the form of the approximate posterior, can frame drastically the final result of the optimization process. To solve this problem, and get an approximate posterior that matches the true posterior as close as possible, we will try to minimize the impact of the initial drawing of the variational parameters by using a deterministic mapping between the data and the variational parameters.

2.3.1. Inference Network

Flexibility and sparsity are two of the main desired properties of the approximate posteriors we want to fit to our dataset. A solution to reduce the number of variational parameters consists in using a direct, deterministic mapping from the datapoints to a given approximate posterior $\boldsymbol{\delta}_t = f(\mathbf{y}_t; \boldsymbol{\xi})$ where $\boldsymbol{\xi}$ is the vector of parameters of the function f. This leads the approximate posterior to look like $q(\mathbf{z}_t \,|\, \mathbf{y}_t; \boldsymbol{\xi})$ (Figure 3). Our goal is now to optimize the mapping function parameters $\boldsymbol{\xi}$ instead of the local variational posteriors $\boldsymbol{\delta}_t$. This optimization can be achieved using some Stochastic Gradient Descent algorithm, as we will see shortly.

This use of an inference network (IN) is known as Neural Variational Inference and Learning (NVIL) (33) or amortized variational inference (34). It alleviates the need of optimizing each of the local approximate posteriors, which can be problematic for large datasets based on stochastic gradient descent optimization where each point has to be visited many times to obtain an accurate estimate

of the true posterior, especially when the prior is optimized (or inferred) at the same time. This technique has led to many applications (31; 35; 34; 36; 37; 38; 39), because it allows the model to find a likely posterior for datapoints that might not have been seen yet, and, for large problems, it is cheaper than estimating each local variational parameters separately. Comparison of Algorithm 2 and Algorithm 3 highlights the difference between Amortized inference and classical VB method in the VEM context.

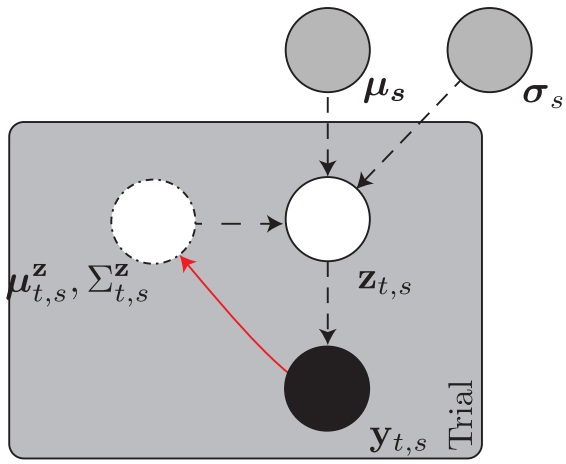

Figure 3: An inference network, here illustrated as a solid red arrow, is constituted by a deterministic mapping (e.g. multilayer perceptron) between the random variable of interest (e.g. RT and choice in our case) and the parameters of the approximate posterior (i.e. $\{\boldsymbol{\mu}^{\mathbf{z}}, \boldsymbol{\sigma}^{\mathbf{z}}\}$).

Algorithm 2: Classical E step for the VEM algorithm (Algorithm 1)

Data: Choices and reaction times $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T\}$, initial guess of $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$
Result: Variational Parameters $\boldsymbol{\delta} = \{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, ..., \boldsymbol{\delta}_T\}$
for $t \in 1 : T$ **do**
  Find $\arg\min_{\boldsymbol{\delta}_t} -\mathcal{L}(q(\mathbf{z}_t; \boldsymbol{\delta}_t))$; `// e.g. using SGD`

Algorithm 3: Stochastic Gradient Ascent Amortized E step for the VEM algorithm (Algorithm 1)

Data: Choices and reaction times $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T\}$, initial guess of $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$, initial guess of $\boldsymbol{\xi}$
Result: Inference network weights $\boldsymbol{\xi}$
repeat
  Select $t \in T$ randomly
  Compute an unbiased, noisy estimate of $\widehat{\nabla_{\boldsymbol{\xi}}\mathcal{L}} \approx -\nabla_{\boldsymbol{\xi}} f(\mathbf{y}_t; \boldsymbol{\xi}) \nabla_{\boldsymbol{\delta}_t} \mathcal{L}(q(\mathbf{z}_t \,|\, \boldsymbol{\delta}_t))$ where $\boldsymbol{\delta}_t = f(\mathbf{y}_t; \boldsymbol{\xi})$
  Update $\boldsymbol{\xi} = \boldsymbol{\xi} - \alpha \widehat{\nabla_{\boldsymbol{\xi}}\mathcal{L}}$; `// where` α `is some small step size`
until *convergence of* $\mathcal{L}(q(\mathbf{z}_t \,|\, \mathbf{y}_t; \boldsymbol{\xi})) = \sum_{t=1}^{T} \mathcal{L}(q(\mathbf{z}_t \,|\, \mathbf{y}_t; \boldsymbol{\xi}))$;

Contrary to Conjugate models optimized using Mean-Field VB, in which exact, closed-form updates of the variational parameters $\boldsymbol{\delta}_t$ can be implemented (40; 41), finding the mapping parameters ($\boldsymbol{\xi}$) in NVIL relies on non-linear optimization algorithms to maximize the ELBO. If we can derive

an unbiased, noisy, stochastic gradient of the ELBO, the optimization can be be achieved using Stochastic Gradient Variational Bayes (31) (SGVB). Such gradient is easily drawn in our case, as the gradient of ELBO can be approximated by its Monte-Carlo estimate: consider the probability distribution $q(\mathbf{z} \,|\, \mathbf{y}; \boldsymbol{\xi})$. If we choose this distribution so that, given a pseudo-random, auxiliary noise variable $\epsilon \sim p(\epsilon)$, the mapping from ϵ to $\tilde{\mathbf{z}}$, i.e. $\tilde{\mathbf{z}} = g(\epsilon, \mathbf{y}; \boldsymbol{\xi})$ is differentiable wrt $\boldsymbol{\xi}$, then the noisy gradient for a mini-batch of J trials and K approximate posterior draws looks like:

$$\nabla \mathcal{L}(q(\mathbf{z} \,|\, \mathbf{y}; \boldsymbol{\xi})) \approx \frac{1}{J} \sum_{j=1}^{J} \frac{1}{K} \sum_{k=1}^{K} \nabla_{\boldsymbol{\xi}} g(\epsilon_{k,j}, \mathbf{y}_j; \boldsymbol{\xi}) \nabla_{\tilde{\mathbf{z}}} [\log p(\mathbf{y}_j, \tilde{\mathbf{z}}) - \log q(\tilde{\mathbf{z}} | \, \mathbf{y}_j; \boldsymbol{\xi})] \quad \text{where } \epsilon_{k,j} \sim p(\epsilon). \tag{12}$$

Take for instance the approximate posterior $q(\mathbf{z} \,|\, \mathbf{y}; \boldsymbol{\xi}) = \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{y}; \boldsymbol{\xi}), \Sigma(\mathbf{y}; \boldsymbol{\xi})\right)$, where we use $\Sigma(\mathbf{y}; \boldsymbol{\xi}) = L(\mathbf{y}; \boldsymbol{\xi}) * L(\mathbf{y}; \boldsymbol{\xi})'$ with $L(\mathbf{y}; \boldsymbol{\xi})$ being the lower-Cholesky decomposition of $\Sigma(\mathbf{y}; \boldsymbol{\xi})$. Then, for a given $\epsilon \sim U(0, 1)$,

$$g(\epsilon, \mathbf{y}; \boldsymbol{\xi}) = \boldsymbol{\mu}(\mathbf{y}; \boldsymbol{\xi}) + L(\mathbf{y}; \boldsymbol{\xi}) \Phi^{-1}(\epsilon) \tag{13}$$

where Φ^{-1} is the inverse cdf function of the univariate Normal distribution. The gradient in Equation (12) can be computed easily using an Automatic Differentiation routine. This gradient estimator, sometimes called Infinitesimal Perturbation Analysis estimator (42; 43; 44) or Reparametrized Gradient (45), has generally a much lower variance than the gradient obtained with the Score function method (45), and, for a large enough mini-batch size $J >> 0$, can be used with a single sample $K = 1$.

2.4. Implementation for Hierarchical (Non-)Linear Regression Modelling

So far, we have looked restrictively at the case of the MLE for the DDM with only one subject, with no experimental conditions. However, in a broader setting, we would be more interested in retrieving the full posterior probability at each of the levels of the dataset (i.e. trials, subjects and population) for each condition. The first step is to specify the model at the subject level, before putting a prior on the prior of the local variables (i.e. a hyperprior). We will assume in the following that the local parameters $\mathbf{z}_{t,s}$ have a Gaussian prior with subject-specific, diagonal covariance matrix $\boldsymbol{\sigma}_s$ and trial- and subject-specific mean $\boldsymbol{\mu}_{t,s} = \zeta(\mathbf{x}_{t,s}; \boldsymbol{\omega}_s)$. ζ is a (non-)linear mapping of $\mathbf{x}_{t,s}$ (i.e. the vector of the trials-subject regressors) with latent subject-specific parameters (or weights) $\boldsymbol{\omega}_s$, that allows the prior to change according to condition and subject.

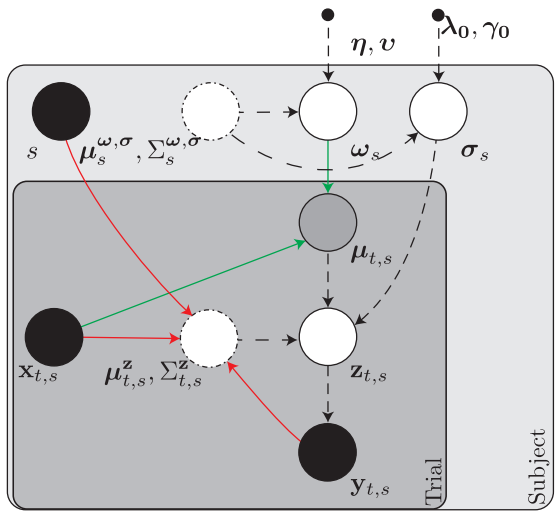

Figure 4: Hierarchical Regression Model implementation. At the bottom level, RT and choices ($\mathbf{y}_{t,s}$) are distributed according to a trial-specific Wiener FPT distribution with parameters $\mathbf{z}_{t,s}$. The approximate posterior parameters $\{\boldsymbol{\mu}_{t,s}^{\mathbf{z}}, L_{t,s}^{\mathbf{z}}\}$ of $\mathbf{z}_{t,s}$ come from an inference network, illustrated with red arrows, with inputs $\mathbf{y}_{t,s}$, $\mathbf{x}_{t,s}$ (regressors), s (subject index) and mapping parameters $\boldsymbol{\xi}^s, \boldsymbol{\xi}^{\mathbf{x}}, \boldsymbol{\xi}^{\mathbf{y}}$ and $\boldsymbol{\xi}^{\mathbf{z}}$). The prior over $\mathbf{z}_{t,s}$ is a normal distribution with means $\boldsymbol{\mu}_{t,s} = \zeta(\boldsymbol{\omega}_s, \mathbf{x}_{t,s})$, where the deterministic mapping ζ is represented with green arrows, and standard deviations $\boldsymbol{\sigma}_s$. Both these parameters have a common approximate posterior, and a fixed prior.

The recognition model (Figure 4) takes the form of:

$$
\begin{aligned}
p(\mathbf{y}, \mathbf{z}, \boldsymbol{\omega}, \boldsymbol{\sigma}) &= \prod_{s=1}^{S} \prod_{t=1}^{T} p(\mathbf{y}_{t,s} \,|\, \mathbf{z}_{t,s})\, p(\mathbf{z}_{t,s} \,|\zeta(\boldsymbol{\omega}_s, \mathbf{x}_{t,s}), \boldsymbol{\sigma}_s)\, p(\boldsymbol{\omega}_s)\, p(\boldsymbol{\sigma}_s^2) \\
\mathbf{y}_{t,s} &\sim Wiener(\mathbf{y}_{t,s} \,|\, \mathbf{z}_{t,s}) \\
\mathbf{z}_{t,s} &\sim \mathcal{N}(\zeta(\boldsymbol{\omega}_s, \mathbf{x}_{t,s}), \boldsymbol{\sigma}_s) \\
\boldsymbol{\omega}_s &\sim \mathcal{N}(\boldsymbol{\eta}_0, \boldsymbol{\upsilon}_0) \\
\boldsymbol{\sigma}_s^2 &\sim \mathcal{IG}(\boldsymbol{\lambda}_0, \gamma_0)
\end{aligned}
\tag{14}
$$

The hyperpriors $\{\boldsymbol{\eta}, \boldsymbol{\upsilon}, \boldsymbol{\lambda}_0, \gamma_0\}$ have to be chosen beforehand.

Furthermore, the inference model takes the form of:

$$
\begin{aligned}
\mathbf{z}_{t,s} \,|\, \mathbf{y}_{t,s}, \mathbf{x}_{t,s}, s &\sim \mathcal{N}(\boldsymbol{\mu}_{t,s}^{\mathbf{z}}, \Sigma_{t,s}^{\mathbf{z}}) \\
&\quad \text{where } \{\boldsymbol{\mu}_{t,s}^{\mathbf{z}}, L_{t,s}^{\mathbf{z}}\} = f\left(f(s; \boldsymbol{\xi}^s), f(\mathbf{x}_{t,s}; \boldsymbol{\xi}^{\mathbf{x}}), f(\mathbf{y}_{t,s}; \boldsymbol{\xi}^{\mathbf{y}}); \boldsymbol{\xi}^{\mathbf{z}}\right) \\
&\quad \text{and } \Sigma_{t,s}^{\mathbf{z}} = L_{t,s}^{\mathbf{z}} {L_{t,s}^{\mathbf{z}}}^T \\
\boldsymbol{\omega}_s, \log \boldsymbol{\sigma}_s^2 &\sim \mathcal{N}(\boldsymbol{\mu}_s^{\boldsymbol{\omega},\boldsymbol{\sigma}}, \Sigma_s^{\boldsymbol{\omega},\boldsymbol{\sigma}})
\end{aligned}
\tag{15}
$$

where the upper index indicates to which latent variable the variational parameter relates, and where we have used $\boldsymbol{\xi} = \{\boldsymbol{\xi}^s, \boldsymbol{\xi}^{\mathbf{x}}, \boldsymbol{\xi}^{\mathbf{y}}, \boldsymbol{\xi}^{\mathbf{z}}\}$ as a general indicator for the weights of the inference

network. Appendix B gives an example of how to implement the Inference Network with the model described in (15). Note that the inference network is flexible enough to take into account the full hierarchy of the model: as $p(\mathbf{z}_{t,s} \,|\, \mathbf{y}_{t,s}) \propto p(\mathbf{y}_{t,s} \,|\, \mathbf{z}_{t,s})\, p(\mathbf{z}_{t,s} \,|\zeta(\mathbf{x}_{t,s}; \boldsymbol{\omega}_s), \boldsymbol{\sigma}_s)$, it is important for our approximate posterior $q(\mathbf{z}_{t,s})$ to incorporate information about the regressors $\mathbf{x}_{t,s}$ as well as the subject's identity (i.e. $q(\mathbf{z}_{t,s}) = q(\mathbf{z}_{t,s} \,|\, \mathbf{y}_{t,s}, \mathbf{x}_{t,s}, s; \boldsymbol{\xi})$) . This is the role of s in Equation (15): this indicator of the subject (and possibly block) identity (e.g. a binary vector encoding the subject number) allows the inference network to change the approximate posterior shape as a function of the subject (another option might be to incorporate the prior value as an input to the inference network).

To consider the possibility of a posterior covariance between $\log \boldsymbol{\sigma}_s^2$ and $\boldsymbol{\omega}_s$, we modelled both parameters with the same multivariate Gaussian. However, when passing to the exponent $\boldsymbol{\sigma}_s^2 = \exp(\log \boldsymbol{\sigma}_s^2)$, it is important to remember that the entropy term of the ELBO has to be corrected by the log-Jacobian of the transform, i.e. $-\log \boldsymbol{\sigma}_s^2$.

To allow the model to retrieve the posterior probability of the (non-)linear weights prior, we added one more level of hierarchy, as illustrated in Figure 5. We defined a Normal prior over $\boldsymbol{\omega}$ with a diagonal covariance matrix: $\boldsymbol{\omega}_s \sim \mathcal{N}(\boldsymbol{\eta}, \boldsymbol{\upsilon})$, whose parameters were distributed according to a fixed Normal-Inverse-Gamma hyperprior.

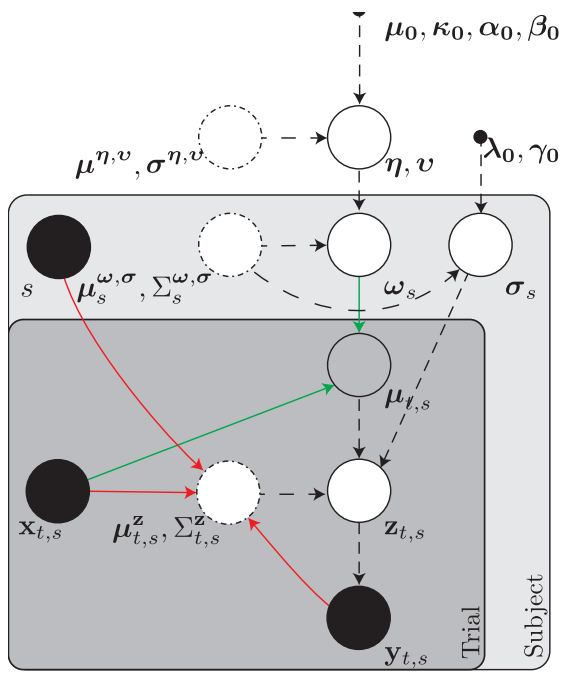

Figure 5: Deep Hierarchical Regression Model implementation. We now also consider $\boldsymbol{\eta}$ and $\boldsymbol{\upsilon}$ as being latent variables, and set a Normal-Inverse-Gamma ($\mathcal{NIG}(\boldsymbol{\mu}_0, \boldsymbol{\kappa}_0, \boldsymbol{\alpha}_0, \boldsymbol{\beta}_0)$) as a hyperprior. This allows for inference at the group level.

We implemented this model using a linear mapping from the regressors to $\boldsymbol{\mu}_{t,s}$, and named this model Hierarchical Linear Regression Model (HLRM).

2.5. RT-Choice Time Series and RNN implementation

The model described above can be improved by adding a trial-to-trial dependency in the various function implemented. Our objective is to fit a model that takes the form:

$$p(\mathbf{y}_{\leq T}, \mathbf{z}_{\leq T}, \boldsymbol{\psi}, \boldsymbol{\sigma}) = \prod_{s=1}^{S} \prod_{t=1}^{T} p(\mathbf{y}_{t,s} \,|\, \mathbf{z}_{\leq t,s}) p(\mathbf{z}_{t,s} \,|\zeta(\mathbf{z}_{<t,s}, \mathbf{x}_{\leq t,s}, \mathbf{y}_{<t,s}, s; \boldsymbol{\psi}), \boldsymbol{\sigma}_s)\, p(\boldsymbol{\sigma}_s)\, p(\boldsymbol{\psi}) \tag{16}$$

where $\boldsymbol{\psi}$ is a population-wise set of parameters linking the prior distribution to the past values of the local variables and behavioural data, as well as the past and current values of the regressors. In this model, the i.i.d of the local variables $\mathbf{z}_{t,s}$ is not assumed anymore but, instead, the prior distribution of these variables varies according to the current and past regressors, and as a function of the past values of the local variables. This rich, generic model can account for a large amount of trial-to-trial dependency in the value of $\mathbf{z}_{\leq T}$, depending on the trial-to-trial dependency that is used. It is implicitly hierarchical, as the subject index s is included as an input for the non-linear regressors.

We used a Gated Recurrent Unit (GRU) function (46) to implement the trial-to-trial dependency. Long Short-Term Memory networks could be used instead, presumably with larger computational cost. The recognition model now looks like:

$$\begin{aligned}
\mathbf{y}_{t,s} &\sim Wiener(\mathbf{y}_{t,s} \,|\, \mathbf{z}_{t,s}) \\
\mathbf{z}_{t,s} \,|\, \mathbf{z}_{<t,s}, \mathbf{x}_{\leq t,s}, s &\sim \mathcal{N}(\zeta(\mathbf{z}_{<t,s}, \mathbf{x}_{\leq t,s}, s; \boldsymbol{\psi}), \boldsymbol{\sigma}_s) \\
\boldsymbol{\psi} &\sim \mathcal{N}(\boldsymbol{\mu}_0^{\boldsymbol{\psi}}, \boldsymbol{\sigma}_0^{\boldsymbol{\psi}}) \\
\boldsymbol{\sigma}_s^2 &\sim \mathcal{IG}(\boldsymbol{\lambda}_0, \boldsymbol{\gamma}_0) \\
\text{with } \mathbf{h}_t &= f\left(\mathbf{P}^{\mathbf{v}}, f(\mathbf{z}_{t,s}; \boldsymbol{\xi}_{dec}^{\mathbf{z}}); \boldsymbol{\rho}\right) \\
\text{and } \mathbf{P}^{\mathbf{v}} &= f(f(s; \boldsymbol{\xi}^s), f(\mathbf{x}_{t,s}; \boldsymbol{\xi}^{\mathbf{x}}), f(\mathbf{y}_{t,s}; \boldsymbol{\xi}^{\mathbf{y}}); \boldsymbol{\xi}^{\mathbf{v}})
\end{aligned} \tag{17}$$

whereas the inference model has the form

$$\begin{aligned}
\mathbf{z}_{t,s} \,|\, \mathbf{y}_{\leq t,s}, \mathbf{x}_{\leq t,s} &\sim \mathcal{N}(\boldsymbol{\mu}_{t,s}^{\mathbf{z}}, \Sigma_{t,s}^{\mathbf{z}}) \\
&\text{where } \{\boldsymbol{\mu}_{t,s}^{\mathbf{z}}, L_{t,s}^{\mathbf{z}}\} = f\left((\mathbf{h}_{t-1}, \mathbf{P}^{\mathbf{v}}); \boldsymbol{\xi}_{enc}^{\mathbf{z}}\right) \\
\log \boldsymbol{\sigma}_s^2 &\sim \mathcal{N}(\boldsymbol{\mu}_s^{\boldsymbol{\sigma}}, \Sigma_s^{\boldsymbol{\sigma}}) \\
\boldsymbol{\psi} &\sim \mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{\psi}}, \boldsymbol{\sigma}^{\boldsymbol{\psi}}).
\end{aligned} \tag{18}$$

As Equation (18) shows, we did not use a multivariate Gaussian for the approximate posterior over $\boldsymbol{\psi}$, because its dimension precludes it (the size of the lower-Cholesky factorization of $\Sigma^{\boldsymbol{\psi}}$ is $\mathcal{O}(n^4)$, where n is the size of the hidden layer), and performance of the model did not seem to be much affected by this assumption. The GRU hidden state $\mathbf{h}_t$ keeps trace of the data in the previous trials. It receives input from the the behavioural data $\mathbf{y}_{t,s}$, the regressors $\mathbf{x}_{t,s}$, the subject index s and the value of $\mathbf{h}_{t-1}$ through the layer $\mathbf{P}^{\mathbf{v}}$. It sends outputs to the value of the approximate posterior over $\mathbf{z}_{t+1,s}$, and is used as a regressor for the value of $\boldsymbol{\mu}_{t+1,s}$. We name this model Variational

Recurrent Inference Network (VRIN) and it is represented in Figure 6. It shares some similarity with the VRNN presented in (47), although our model is fully Bayesian, as we fit an approximate posterior at the top level to all the parameters we wish to make inference about.

This implementation works better than just adding a RNN to the hierarchical Regression Model, because it allows for a refined modulation of the effect of the various regressors during the task. It comes, however, at the cost of the explicitness of the hierarchy (since posterior distribution of subject weights becomes inaccessible) and the interpretability of the regression parameters. Whereas in Section 2.4, inference about each of the variational parameters is possible, this is no longer the case with VRIN, because $\boldsymbol{\mu}_{t,s} = \zeta(\mathbf{z}_{<t,s}, \mathbf{x}_{\leq t,s}, s; \boldsymbol{\psi}), \boldsymbol{\sigma}_s)$ is non-linear, making the interpretation of the values of $\boldsymbol{\psi}$ harder. However, this is not an issue in practice, since we still have access to the posterior of the lower-level parameters, from which we can compute the summary statistics at the subject and population level.

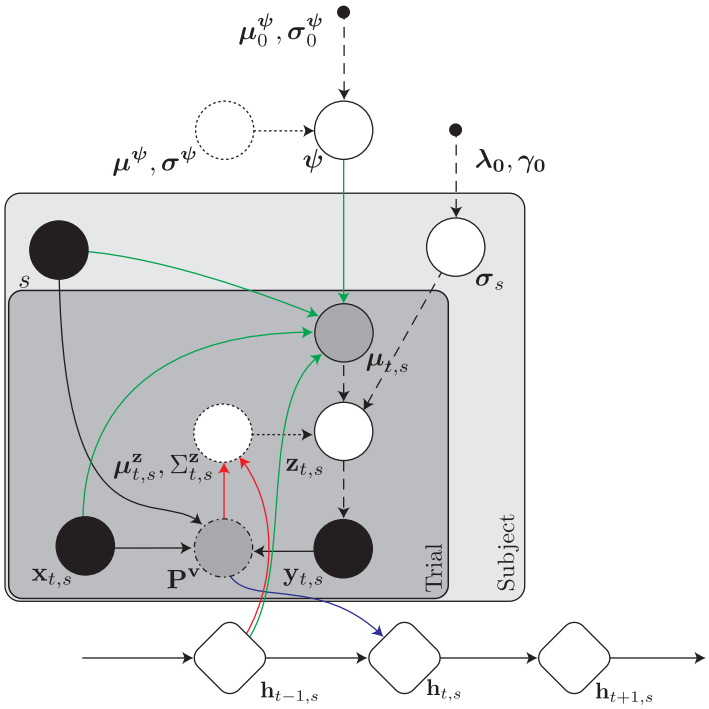

Figure 6: VRIN. The hidden state $\mathbf{h}_t$ brings the relevant information from trial to trial in order to make inference about $\mathbf{z}_{t+1,s}$. It receives input from the current local parameter value $\mathbf{z}_{t,s}$, the regressors (subject index s included) and the data through a layer $\mathbf{P}^{\mathbf{v}}$. $\mathbf{h}_t$ is also used as input for the non-linear mapping $\boldsymbol{\mu}_{t,s} = \zeta(\mathbf{z}_{<t,s}, \mathbf{x}_{\leq t,s}, s; \boldsymbol{\psi}), \boldsymbol{\sigma}_s)$, together with the regressors.

3. Results

3.1. Implementation

In all the simulations and tests shown hereafter, we assigned a linear form to $\zeta(\mathbf{x}_{t,s};\boldsymbol{\omega}_s)$ and a non-linear form (single or multi-layer perceptron) to $\zeta(\mathbf{z}_{<t,s},\mathbf{x}_{\leq t,s},s;\boldsymbol{\psi})$ and for each of the other mapping functions $\{f_{\boldsymbol{\rho}}, f_{\boldsymbol{\xi}^{\mathbf{x}}}, f_{\boldsymbol{\xi}^{\mathbf{y}}}, f_{\boldsymbol{\xi}^{\mathbf{z}}_{dec}}, f_{\boldsymbol{\xi}^{\mathbf{z}}_{enc}}, f_{\boldsymbol{\xi}^{s}}\}$. For the size of the problems considered here, layers of 50 cells were sufficient to reach a sufficient level of accuracy for the prediction of the posterior shape.

In order to have a posterior guided mostly by the data, we chose a weakly informative hyperprior at the top level, which was set as follows: $\boldsymbol{\lambda}_0$ and $\boldsymbol{\gamma}_0$ were both set to 0.001. The $\mathcal{NIG}$ of the HLRM was set with $\boldsymbol{\mu}_0 = 0$, $\boldsymbol{\kappa}_0 = 1$, $\boldsymbol{\alpha}_0 = 0.001$, $\boldsymbol{\beta}_0 = 0.001$. The unit-diagonal Gaussian prior was used for $\boldsymbol{\psi}$.

Some form of annealing can be used to ensure that the approximate posterior is as broad as possible. We opted for a scheduled tempering where the likelihood was decreased during optimization by a factor $\beta_i = \min(0.01 + i/100001)$, where i is the iteration of the optimization, and the entropy term was left constant (34).

The models were coded in Julia v0.6[1] (48), and gradients were computed using mixed-mode automatic differentiation with the ForwardDiff (49) and ReverseDiff packages.

All algorithms were given a maximum computational budget of 192 cpu-hours (24 hours on a 8 cpus machine) and 10^5 iterations per CPU. No hardware accelerator (GPU, OpenCL CPU) was used, although they would be well suited for the present problem. An Adam (50) optimizer was chosen because it performed better than AdaGrad and other methods.

It should be noted here that this setting was voluntary pessimistic, as the actual computational time and number of iterations necessary to reach the maximum of the ELBO were lower than maxima allowed.

3.2. Simulation analysis

We simulated a dataset of 64 hypothetical subjects performing a numerical Stroop task (51) of 500 trials. We considered two difficulty conditions: conflict (when the value of the numbers displayed anticorrelated with their size). Instructed response was either left or right. We therefore considered a 2x2 table of regressors. No deadline was imposed, as we did not restrain the RT to a certain range.

The trial-by-trial DDM parameters were generated as follows: overall, the threshold tended to decrease with a low, fixed rate. When an error was made, this distance increased slightly, implementing post-error slowing (52). The drift was updated from trial to trial as a function of 1) the instructed response (we considered that the drift always pointed towards the correct response), 2) the difficulty (its absolute value was higher for simple than difficult trials) and 3) the absolute value of the previous drift. Indeed, the absolute value of the drift followed a random walk with a

[1] A version of the code used for this example is available online here

positive drift that reflected a training effect. The start point reflected some sort of Gambler Fallacy (53; 54; 55; 56): if the instruction was left at the trial t, subject started with the prior belief that the next instruction (at $t+1$) was going to be right. This bias increased as the number of repetitions of the same instruction got higher: the rightward bias in the previous example was higher at $t+4$ than at $t+1$ if all the trials between t and $t+3$ had the left instruction. The non-decision time was generated with a Gaussian Process with a squared exponential kernel.

Simulation of the RT and choices was done according to (10).

We fitted the HLRM and the VRIN to this dataset. As a benchmark, we also fitted this dataset to a model identical to the HLRM using a No-U-Turn Hamiltonian Monte-Carlo method (57) with Stan (58). These results are reported in Appendix C. We expected that the VRIN would outperform the HLRM in term of parameter prediction and model evidence.

Both optimization procedures completed in the allotted time, and reached a stable ELBO before the end of the optimization.

A simple Bayesian Model Selection (BMS, (59)) performed on the basis of ELBO of the two models showed that the VRIN model was more likely for 62.77 of the 64 subjects, with exceedance probability φ close to 1.

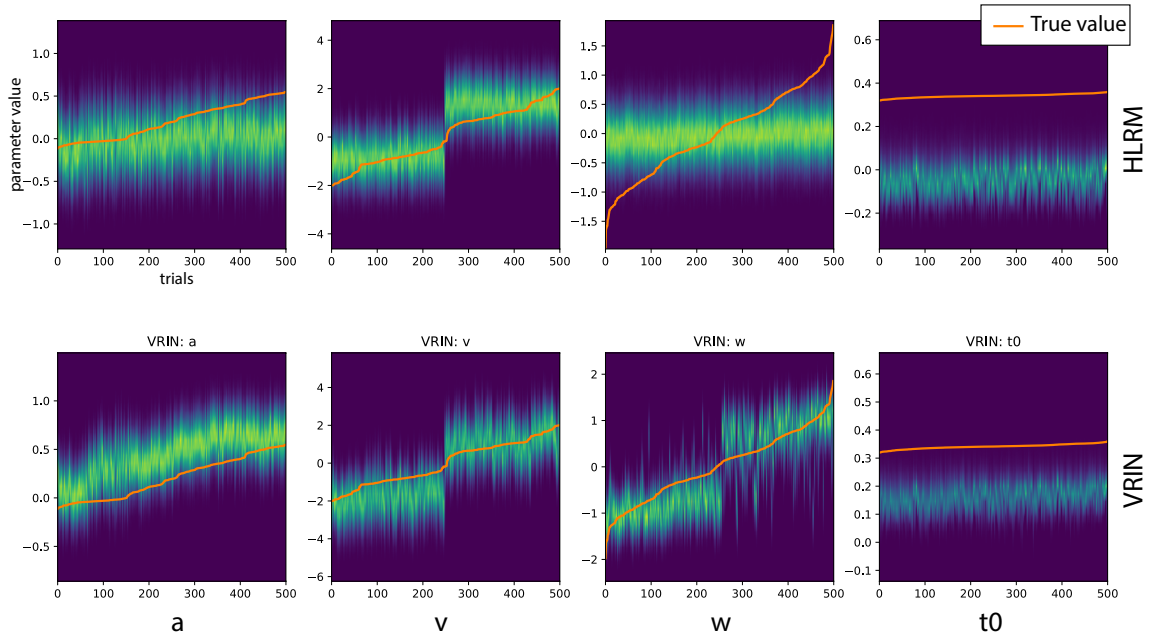

Figure 7: Trial-by-trial posterior probabilities of DDM parameters (heat map), compared to their real, generative value (orange line) for the HLRM (top) and VRIN models (bottom) at the first level ($\mathbf{z}_{t,s}$) for one subject. For clarity, the parameters have all been sorted so that the generative parameters values are strictly increasing. The VRIN model was able to recover the dynamic of a, w and t_0, whereas HLRM only estimated their mean value.

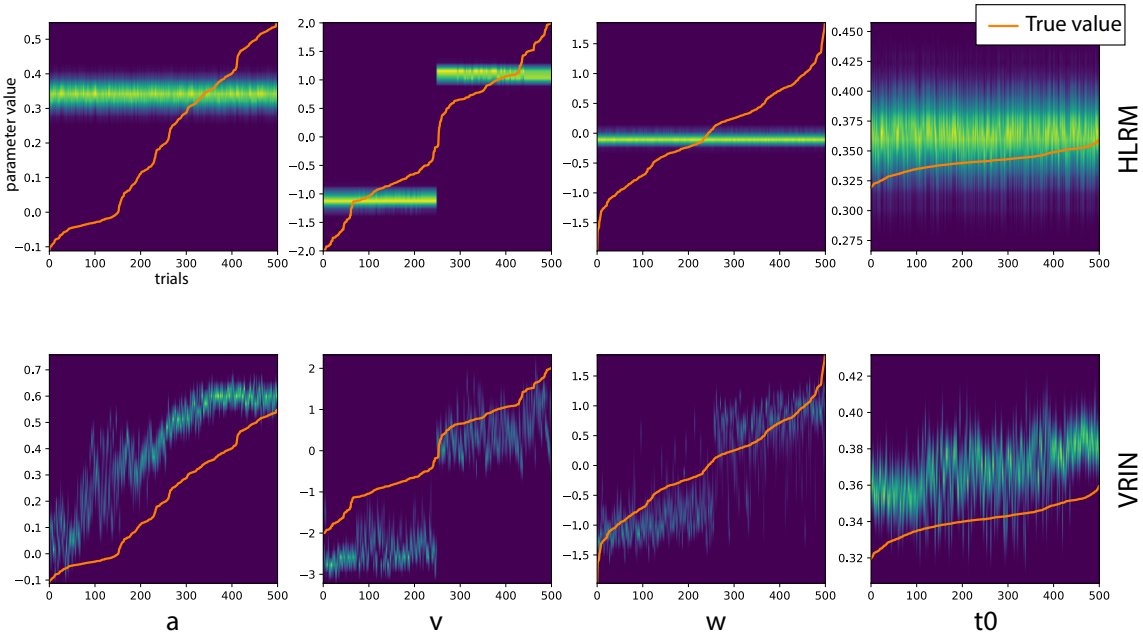

Figure 8: Posterior probability compared to real, generative value of the parameters (orange line) for the HLRM (top) and VRIN model (bottom) at the second level ($\boldsymbol{\mu}_{t,s}$) for the same subject as in Figure 7

Figure 7 and Figure 8 show the posterior probabilities of $\mathbf{z}_{t,s}$ (Figure 7) and $\mu_{t,s} = \zeta(...)$ (Figure 8) for a random subject for each of the two models. In general, the VRIN model was able to recover a more precise value for the parameters and, most importantly, to recover their individual trial-specific values. This was particularly striking for the starting point parameter $w_{t,s}$ and the threshold $a_{t,s}$, since these parameter values depended on instructed responses in previous trial and their position in the time course of the experiment and could, therefore, be recovered only from the time series.

To give an insight of the performance of the two models at the population level, we compared the actual standard deviations of generative values of the parameters across subjects with their estimates from the 2 models. Again, VRIN outperformed the HLRM (and Monte-Carlo method, see Appendix C) for almost every parameter (Figure 9) both in terms of correlation with generative values and in terms of bias (HLRM tended to underestimate parameter standard deviations). We also include the correlation between the true value of the threshold and non-decision time and their estimates in both models at the population level (drift and starting point were left out of the analysis, as the value of the first varied as a function of the conditions, and the second was centered on zero for all subjects). The correlations were similar for both models, although the threshold scaled better with its true value with VRIN.

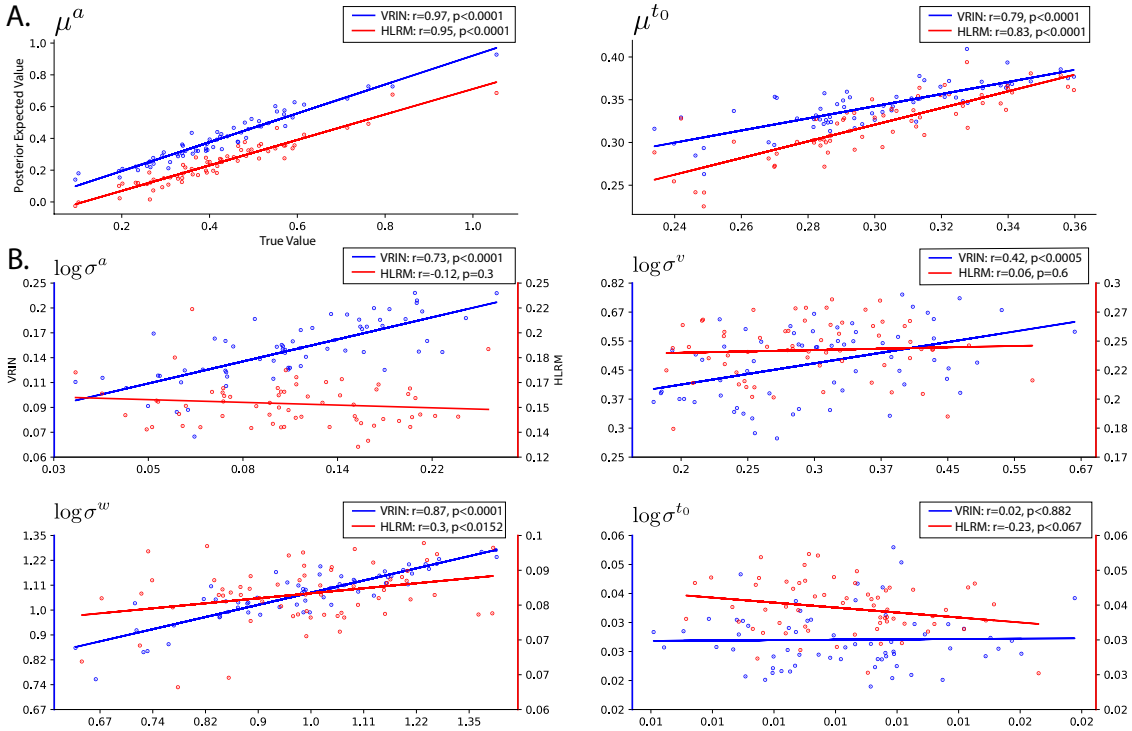

Figure 9: Correlations between real parameter value and model estimates. The true, empirical value is displayed on the x axis, and the estimates are displayed on the y axis. Pearson correlation coefficients are displayed in the legend box. Because the mean value was changing through the task for the starting point and the drift, the correlation at the population level is not studied here. Also, the values of the log-standard deviation displayed are the empirical value of the standard deviation of $\mathbf{z}_{t,s}$ around their regressor-specific average. A. Prior mean of the threshold and non-decision time. Correlations were roughly equal for the threshold, whereas VRIN estimated better the scale of the population average. For the non-decision time, the HLRM outperformed slightly the VRIN model (HLRM: $r = 0.83, p < 0.0001$, VRIN: $r = 0.79, p < 0.0001$). B. The left y axes (blue) represent the values fitted using the VRIN model, the right y axes the values obtained with the HLRM. Note that, for obvious reasons on interpretability, the per-category mean of the drift value has been subtracted before computing the standard deviation. VRIN provided a better estimate of the difference between subjects and provided a better estimate of the scale of the variance for each parameter (HLRM underestimated the variance of most parameters). It should be emphasized that, because of this difference in the estimation the scale of the variances, we used different scales for the two models to bring them on the same graph.

3.3. Example from real data: Random Dot Motion Task

To validate the use of the VRIN to real datasets, and show that it was able to extract relevant features of behaviour, we fitted the HLRM and the VRIN to a dataset issued from a Random Dot Motion Task where subjects had to discriminate leftward and rightward motion of the dots on the screen (60). The dots were all of the same colour but changed across trials (3 equiprobable colours). Two of these three colours predicted the motion of the dots with 80% accuracy (i.e. colour 1 =

leftward motion in 80% of the trials, colour 2 = rightward motion in 80% of the trials, colour 3 = 50% of rightward motion). Hence, a table of 3x2 regressors was used in both models. In this task, subjects received immediate feedback about their performance. Each block lasted for 60 trials, and each subject (N=20) performed a total of eight blocks.

In this task, subjects had to respond before a 1000 ms deadline, so that a few missing trials were present. Nevertheless, we included these trials in the analysis: indeed, if $p(\mathbf{y}_{t,s} \,|\, \mathbf{z}_{t,s})$ is the probability density that the subject chose the action $C_{t,s}$ at time $RT_{t,s}$ given $\mathbf{z}_{t,s}$, then $1 - \int_{-\infty}^{\tau} p(t, b|\, \mathbf{z}_{t,s}) + p(t, \neg b|\, \mathbf{z}_{t,s})dt$ is the probability that no response was given before the deadline τ. Therefore, in this context, even missing trials have a probability and can be included in the model.

However, inclusion of missing trials prevented us from using pre-recorded gradient tape[2], leading to an optimization around 10 times slower than for the simulated example. This prevented the optimization algorithms to complete the required number of iterations before the end of the allotted time. The ELBO estimate was, however, stable at this point.

Another difference between this dataset and the previous example is that, here, we had to use a uniform approximate posterior over t_0 to avoid the posterior expected value of $t_{0_{t,s}}$ tending to the value of $RT_{t,s}$ (see Appendix A).

When performing a BMS between the two, we could see that the VRIN outperformed the HLRM (number of subject where VRIN was preferred: 12.43, $\varphi = 0.85$).

The first interesting observation we can make about the fit of the VRIN is that, unlike the HLRM, it was able to account for the post-error slowing phenomenon. Analyzing the trial-to-trial changes in the posterior mean of the threshold ($\mu^a_{t,s} = \zeta(\mathbf{z}_{<t,s}, \mathbf{x}_{\leq t,s}, s, \boldsymbol{\sigma}_s; \boldsymbol{\psi})$), we observed that the value of $\mu^a_{t+1,s} - \mu^a_{t,s}$ was significantly lower after a correct response than after an error: after an accurate trial, threshold decreased slightly but significantly (average = -0.009, 95% CI =$[-0.012, -007]$), whereas after error, this difference was positive (average = 0.042 95% CI =$[0.036, 0.048]$). The difference between the two was also significant ($z = -18.3608$, $p < 1e - 74$) (Figure 10). When applied on the fitted HLRM parameters, this analysis showed a similar pattern, although of much lower magnitude (average effect of errors: 0.001, CI = $[0.0005, 0.002]$, effect for accurate trials not significant, difference magnitude $z = -3.99539$, $p < 1e-4$). The fact that this effect emerged (even slightly) with the HLRM fit should not surprise the reader: errors were more likely during some trial categories (color and motion direction) than others, and the result was that the regression weights of the HLRM encoded a larger mean threshold value for these trials. To show this, we executed the same test on our dataset, while shuffling the trials for each subject, and then subtracting the average per-category value. This obviously resulted in a zero-mean average difference in the mean threshold value for both error and rewarded trials for the HLRM and all the effects previously reported became non-significant. Interestingly, the effect for the VRIN model were still significant after these manipulations (Average reward difference = $-0,011$, 95% CI =$[-0.013, -0.008]$, average error difference = 0.049, 95% CI =$[0.043, 0.055]$, difference between the two: $z = -20.1101$, $p < 1e-89$). In summary, the VRIN model shows that $\mu^a_{t,s}$ tended to decrease after a correct response and to increase after an error.

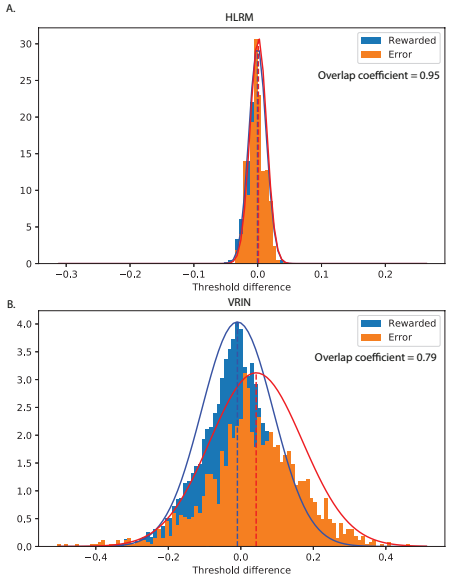

Figure 10: Difference in the posterior expected value of the mean threshold $\mu^a_{t,s}$ between consecutive trials. Results are shown for HLRM (top) and VRIN models (bottom). The histograms in A. and B. show the distributions of the difference in the expected value of the mean threshold between trials $t+1$ and t. After errors, threshold tended to increase, whereas it tended to decrease after accurate trials. The larger overlap observed with HLRM indicates a lower capacity of this model to discriminate between errors and correct trials.

A similar pattern was found for the start-point, where the value of $\mathbb{E}_q[\mu^w_{t,s}]$ tended to anticorrelate with the motion at the previous trial for the VRIN model only (Figure 11), reflecting some kind of Gambler Fallacy.

Finally, subjects executed more right than left clicks (an overall proportion of 0.4622 left clicks). This was reflected by a slight, average, positive bias (mean value of $\mathbb{E}_q[\mu^w_{t,s}] = 0.143$, 95% CI $= [-0.014, 0.301]$, since the convention used in our dataset encoded rightward choice at the lower boundary).

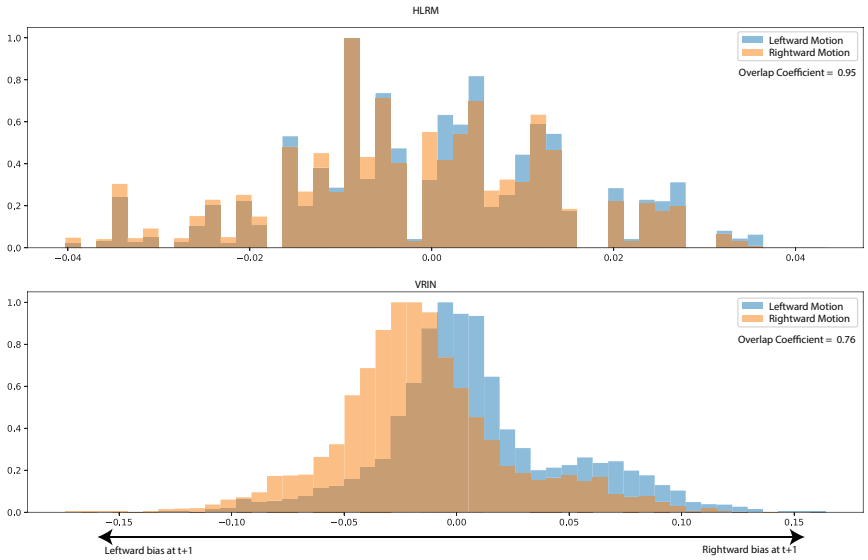

Figure 11: Illustration of the Gambler fallacy in the random dot moving task. The VRIN algorithm shows that participants were biased towards the side opposite the one instructed in the previous trial. Results are shown for the two models, with the corresponding overlap coefficient.

4. Discussion

In this paper, we show that the Drift Diffusion Model (DDM) is well suited to be used in combination with Variational Inference. We present two ways of using VB in this context: the first approach is a classical hierarchical regression model that goes deep in the data structure, with latent variables coming from every level of the hierarchy (trial, subject and population). The second approach takes advantage of the time series of the dataset to build a precise posterior probability distribution of the parameters at each trial. Our approaches are data-driven and can, therefore, be applied to a very broad range of datasets with little or no tuning. The only features the user needs to define are the regressors.

Many routines have been proposed to optimize the DDM. Ratcliff and Tuerlinckx (10), and others (61; 62; 9) have used a numerical quadrature (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 (63; 64; 7; 65). Vandekerckhove et al. (20) used a Markov Chain Monte Carlo (MCMC) technique to estimate each trial distribution in a fully Bayesian setting. Our approach differs in several ways from the work of Vandekerckhove et al.: first, the method we used to make inference about the generative model allows us to extend it to large (or very large) datasets with limited cost, as the gradient of the ELBO is estimated stochastically. The use of Variational Inference makes it also possible to control the convergence of the algorithm, which is challenging with MCMC techniques. Finally, the VRIN model, 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.

Another attempt to model the DDM in a Bayesian framework has been published by Wiecki et al. (14). This group wrote a Python toolbox (HDDM) aimed at fitting the DDM using MCMC. However, this toolbox still relies on NQ to compute estimates at the lower level, rather than estimating each local posterior $\mathbf{z}_{t,s}$. Our aim was to alleviate the computational burden of the integration process without compromising on accuracy and while keeping the entire original structure of the DDM.

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 (35; 34; 66; 26), easy to use (67; 45; 32) and adapted to potentially very large problems (41; 36; 38). 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 (12) without numerical quadrature ; the second being to recover an estimate of the trial-to-trial dependency of the DDM parameters in a behavioural task.

We present two efficient models to solve these problems: HLRM implements a fully hierarchical (non-)linear regression model with the DDM as the core probability distribution in a VB framework. VRIN takes into account the time series of the RT and choices to assess the value of the parameters at each trial as well as their distributions. VRIN shares the same core idea as the Variational Recurrent Neural Network (47), with the notable difference that it is fully Bayesian.

Importantly, VRIN does not assume that local variables $\mathbf{z}_{t,s}$ are fully determined by the time series, but rather that the time series act as non-linear regressors over the mean values of these local variables. This allows the model to account for volatility in the parameter values that would not be time-dependant but truly stochastic. Therefore, even though our model is more complex that the classical Full DDM, we predict that, in general, a more precise estimate of the posterior probability of the parameters will be obtained at the trial, subject and population level, because the full time series will be considered to estimate these values.

It is noteworthy that, contrary to previous approaches (12), here we did not omit any parameter when modelling the trial-by-trial variability. Usually, threshold a (or distance between boundaries) is considered fixed during the experiment. The present method is the first, to our knowledge, in which the variability for a has been implemented. We showed that accounting for variability in a is important and can be distinguished from variability in the start point, even when no time series are considered, unlike what has been previously stated (12). Furthermore, this feature allowed us to see the dynamic of the decision making strategy during the task (e.g. post-error slowing).

Another key feature of our models is their ability to make inference at the trial level, especially with the VRIN model. 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 (68) or sub-thalamic nucleus stimulation (69)). One could also 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 (70; 69; 71). The present methods open the door to a more precise study of such phenomena, by allowing precise estimation of the generative parameters.

The risk of overfitting with our methods is low: the advantage of Bayesian Statistics is that, if a

regressor is irrelevant and is not predictive of the value of a parameter, its posterior probability distribution will be centered on zero and, possibly, broad. This feature can be observed from the fit of the model. However, the artificial neural networks (ANN) used at the inference level (for the HLRM and the VRIN) and at the generative level (for the VRIN) may require some regularization in order to be as informative as possible. The Dropout regularization method (72) is a likely candidate for such purpose. It has the advantage that it is well suited for Variational Inference (36), and it can be used to reduce the size of the ANN (73).

Another limitation of the present approach is the use of the Gaussian distribution as an approximate posterior, which can be suboptimal. First the shape of the approximate posterior might be improved by using some more flexible family of distributions such as Normalizing Flows (34) or Inverse Autoregressive Flow (38). These distributions add to the outputs of the inference network some parameters that transform the initial distribution (for instance a Gaussian or uniform distribution) to more complex distributions, with possibly many local maxima and parameter dependencies. However, Normalizing and Inverse Autoregressive Flows have the drawback that outputs from the inference network grow substantially each time a new transformation is added, making the optimization slower and more difficult. Therefore, adjusting the trade-off between accuracy of posterior estimates and computational cost is an important challenge for future developments of our models.

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 (74), allowing our models to extend to these other frameworks. Applicable models include the Urgency Gating Model (75; 76; 77), the Collapsing Barrier DDM (78; 79) or the Time-varying Ornstein-Uhlenbeck model (80; 81; 1; 82).

In the presence of missing trials or blocks of variable length, the gradient function cannot be pre-recorded[2]. This makes the evaluation of the gradient much slower than without missing trials. This constitutes another crucial issue to improve in the future.

Finally, informative priors should probably be included for the variance parameters $\boldsymbol{\sigma}_s$, or another level of inference should be implemented to adjust this prior to the dataset. We did not consider this possibility here, however, for the sake of simplicity.

In summary, we present two VB-backed hierarchical implementations of the DDM that can retrieve relevant information of the data-generating process at each level. We focused our analysis on the possibility of modelling the trial-to-trial dependency of parameters. We show our models are flexible enough to account for a wide range of behavioural phenomena such as the gambler fallacy or post-error slowing. We believe these models provide an interesting alternative to current approaches to applying Full DDM models to behavioural data.

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

[2]see ReverseDiff documentation for a description

bounded parameters $\{a, v, w, t_0\}$ where $a \in \mathbb{R}^+ \triangleq f^{-1}(a')$, $w \in [0;1] \triangleq g^{-1}(w')$, $t_0 \in [-\infty; RT] \triangleq h^{-1}(t_0')$ and $\{a', v, w', t_0'\} \in \mathbb{R}$. We can construct a distribution

$$Wiener'(\mathbf{y}\,|a', v, w', t_0') \triangleq Wiener(\mathbf{y}\,|a, v, w, t_0)$$

that satisfies this condition for a and w and t_0 by choosing appropriate transforms f, g and h. A natural choice for f and g are respectively the exponential function and some sigmoid function such that $\sigma(w') : \mathbb{R} \to [0;1]$. Sampling a and w using SGVB is straightforward: we just have to sample these parameters from their unbounded approximate posterior $a', w' \sim q(a', w')$, and then apply the transformation.

Sampling the non-decision time is more tricky: whereas the domain of the prior is unbounded ($t_{0_t} \in [-\infty; \infty]$), the domain of the approximate posterior is not ($t_{0_t} \in [-\infty; RT_t]$). We therefore need to find a mapping $h(t_{0_t}) : \mathbb{R} \to [-\infty; RT_t]$ for the approximate posterior, otherwise a large amount of the samples risk to give a zero-valued likelihood when the first-passage density is computed, making the SGVB unstable (note that this is not the case for a' and w' that can be transformed beforehand and used in an unbounded space all over, as the posterior and prior can be unbounded). We decided to use

$$q(f(t_{0_t})|\,\mathbf{y}_t; \boldsymbol{\xi})\left|\frac{\boldsymbol{\delta}\, f}{\boldsymbol{\delta}\, t_{0_t}}\right|^{-1} = q(RT_t - \log\left(1 + \exp(-t_{0_t})\right)|\,\mathbf{y}_t; \boldsymbol{\xi})\frac{1}{1 + e^{t_{0_t}}}.$$

This valid density function has the desired property of having $f(t_{0_t}) < RT_t \quad \forall\, t_{0_t} \in \mathbb{R}$, making sampling of t_{0_t} easy. Moreover, whereas we can consider that if $t_{0_t} << RT_t$ then $q(f(t_{0_t})|\,\mathbf{y}_t; \boldsymbol{\xi})\left|\frac{\boldsymbol{\delta}\, f}{\boldsymbol{\delta}\, t_{0_t}}\right|^{-1} \approx q(f(t_{0_t}); \boldsymbol{\xi})$, when $t_{0_t} \to \infty$ the entropy component of the ELBO decreases nearly linearly. This property discourages higher values of t_{0_t} and ensures that the Inference Network does not converge on spurious patterns where $f(t_{0_{t,s}}) \approx RT_{t,s} \quad \forall t, s$, making the other parameters uninformative. Typically, the SGVB optimization starts with higher values of $t_{0_{t,s}}$ and progressively finds more informative and more complex patterns. To accelerate this process, the scheduled annealing mentioned in Section 3 can be used: by reducing the expected log-joint probability value, it consequently boosts the entropy term of the ELBO in the first iterations. Hence, we make sure that the optimization algorithm explores sufficiently the t_0 space before settling somewhere.

Sometimes, optimization may fail with this form of approximate posterior: we have encountered a few cases where the algorithm only found the spurious pattern we wanted to avoid. If this happens, an option is to use a simpler approximate posterior for $t_{0_{t,s}} \sim U(a_{t_0}, b_{t_0})$ subject to $b_{t_{0_{t,s}}} < RT_t$ and $a_{t_{0_{t,s}}} < b_{t_{0_{t,s}}}$. In practice, we just need to map the output of the inference network by using a transform ensuring that this inequality holds, such as $b_{t_{0_{t,s}}} = RT_t - \log(1 + \exp(b_{t_{0_{t,s}}}))$ and a similar transform for $a_{t_{0_{t,s}}}$.

Appendix B. Inference Network for the HLRM

Consider the restricted form of the HLRM is (14). As we have chosen a multivariate Normal approximate posterior for the local parameters $\mathbf{z}_t$, we have:

$$\mathbf{z}_{t,s} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathbf{z}}_{t,s}, \Sigma^{\mathbf{z}}_{t,s})$$

Our scope is to use amortized inference to find the value of $\{\boldsymbol{\mu}^{\mathbf{z}}_{t,s}, L^{\mathbf{z}}_{t,s}\}$. We can see from (14) that the posterior value of $\mathbf{z}_{t,s}$ depends upon each level of the hierarchy, i.e. the data point distributed according to $\mathbf{z}_{t,s}$: $\mathbf{y}_{t,s} \sim \mathbf{z}_{t,s}$, the regressors $\mathbf{x}_{t,s}$ and the subject identity s that define together the mean of the prior over $\mathbf{z}_{t,s} \sim \mathcal{N}(\zeta(\mathbf{x}_{t,s}; \omega_s), \boldsymbol{\sigma}_s)$.

We used a multi-layered perceptron to map these variational regressors to $\{\boldsymbol{\mu}^{\mathbf{z}}_{t,s}, L^{\mathbf{z}}_{t,s}\}$. First, the variable s has to be transformed into a binary vector of length $\log_2(S)$. Then, each of these variables activates its corresponding layer using the formula:

$$\mathbf{P}^{\mathbf{u}} = h(\mathbf{W}^{\mathbf{u}}\mathbf{u} + \mathbf{b}^{\mathbf{u}})$$

where $\mathbf{u}$ is the regressor and $h()$ is a smooth element-wise non-linear function with parameters $\boldsymbol{\xi}^{\mathbf{u}} = \{\mathbf{W}^{\mathbf{u}}, \mathbf{b}^{\mathbf{u}}\}$. We used the fast logistic function defined as:

$$\sigma_{fast} : x \in \mathbb{R} \mapsto \frac{x}{1 + |x|} \in (-1 : 1)$$

The vectors $\{\mathbf{P}^s, \mathbf{P}^{\mathbf{x}_{t,s}}, \mathbf{P}^{\mathbf{y}_{t,s}}\}$ can then be used similarly to find $\{\boldsymbol{\mu}^{\mathbf{z}}_{t,s}, L^{\mathbf{z}}_{t,s}\}$:

$$\{\boldsymbol{\mu}^{\mathbf{z}}_{t,s}, L^{\mathbf{z}}_{t,s}\} = \sigma_{fast}(\mathbf{W}^{\mathbf{z}}(\mathbf{P}^s, \mathbf{P}^{\mathbf{x}}_{t,s}, \mathbf{P}^{\mathbf{y}}_{t,s}) + \mathbf{b}^{\mathbf{z}})$$

where we followed the nomenclature of (15). In practice, however, we did not use two separate mappings for $\mathbf{P}^s$ and $\mathbf{P}^{\mathbf{x}_{t,s}}$ but a single mapping $P^{s,\mathbf{x}_{t,s}} = f((s, \mathbf{x}_{t,s}); \boldsymbol{\xi}^{s,\mathbf{x}_{t,s}})$ in order to reduce the size of the problem to its minimum.

Appendix C. No-U-Turn HMC fit to the simulated dataset

For completeness, we also fitted the simulated using HMC, a popular and efficient MCMC algorithm (57) available in the Stan library. We used the default configuration, and executed only one run of the sampling algorithm as the time required by this model was considerable (around 68 hours).

In terms of fit to the data at the subject level, we obtained a performance somehow superior to the HLRM Figure C.12: specifically, the algorithm provided better estimates of the starting point than the VB optimized HLRM, but still failed in fitting the threshold.

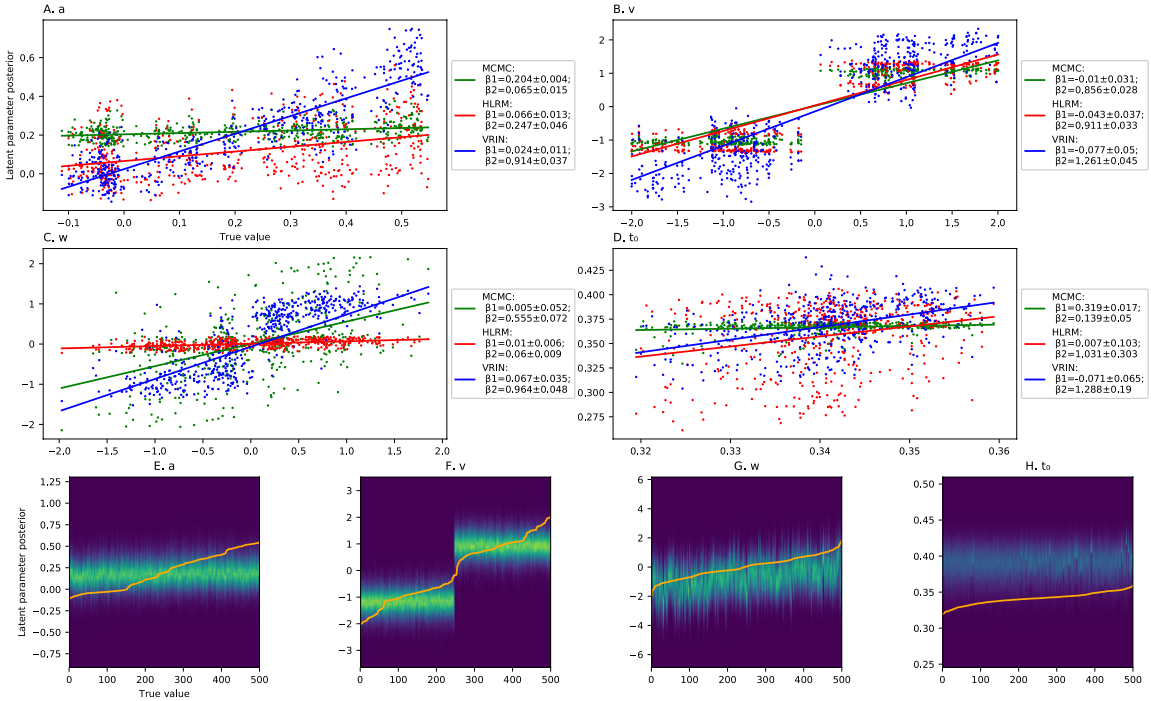

Figure C.12: MCMC performance at the level of the local variables $\mathbf{z}_{t,s}$ for the same subject as Figure 7. A-D display the linear regression model fitted to the posterior expectations of the three techniques, with the true values as regressors. The parameter $\beta 1$ in the legend stands for the intercept, and $\beta 2$ for the slope. The 95% confidence interval of these parameters is also shown. E-H illustrates MCMC performance with conventions similar to Figure 7.

Note that we show here the results of the fit of the local variables: the fit of prior distribution $\mathcal{N}(\boldsymbol{\mu}_{t,s}, \boldsymbol{\sigma}_{t,s})$ was undeniably poorer than the one obtained with the VRIN model.

At the population level, MCMC performed equally well with respect to both VB algorithms for the estimation of the threshold (Figure C.13), and better for the estimation of the non-decision time. However, its performance in estimating parameter standard deviations was much lower than for VRIN.

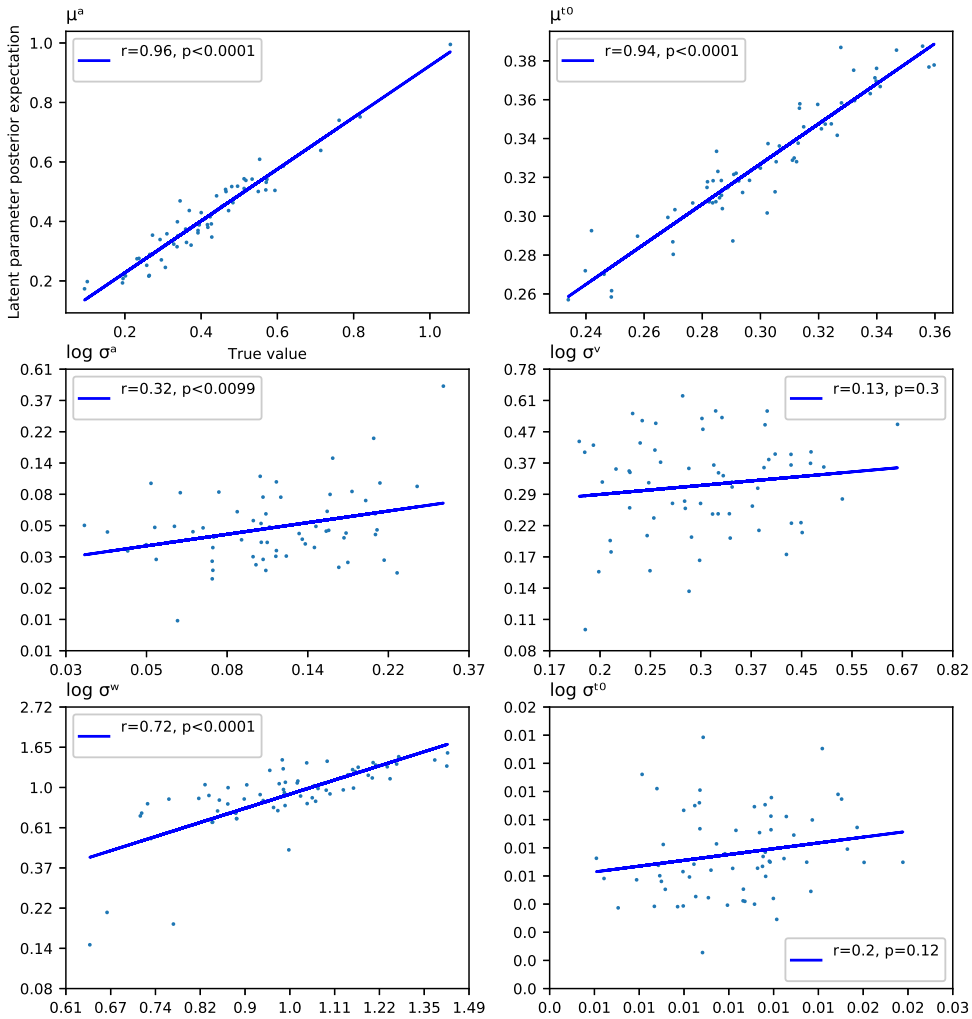

Figure C.13: MCMC performance at the population level. This plot is structured similarly to 9.

[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] Roger Ratcliff and Jeffrey 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] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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.

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

[18] Joachim Vandekerckhove and Francis Tuerlinckx. Diffusion model analysis with MATLAB: A DMAT primer. *Behavior Research Methods*, 40(1):61–72, feb 2008.

[19] 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.

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

[21] 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)*. 2013.

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

[23] 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.

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

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

[26] 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.

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

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

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

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

[31] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *Iclr*, (Ml):1–14, dec 2013.

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

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

[34] 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.

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

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

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

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

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

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

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

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

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

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

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

[46] 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.

[47] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical Multiscale Recurrent Neural Networks. (c):1–13, 2016.

[48] 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.

[49] Jarrett Revels, Miles Lubin, and Theodore Papamarkou. Forward-Mode Automatic Differentiation in Julia. (April):4, 2016.

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

[51] Derek Besner and Max Coltheart. Ideographic and alphabetic processing in skilled reading of English. *Neuropsychologia*, 17(5):467–472, jan 1979.

[52] 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.

[53] Greg Barron and Stephen Leider. The role of experience in the Gambler's Fallacy. *Journal of Behavioral Decision Making*, 23(1):117–129, jan 2010.

[54] Gui Xue, Zhonglin Lu, Irwin P. Levin, and Antoine Bechara. An fMRI study of risk-taking following wins and losses: Implications for the gambler's fallacy. *Human Brain Mapping*, 32(2):271–281, 2011.

[55] 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.

[56] 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.

[57] Michael Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. 2017.

[58] 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.

[59] 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.

[60] K H Britten, M N Shadlen, W T Newsome, and J a Movshon. The analysis of visual motion: a comparison of neuronal and psychophysical performance. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 12(12):4745–65, dec 1992.

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

[62] 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.

[63] 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.

[64] 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.

[65] 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.

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

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

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

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

[74] 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.

[75] Paul Cisek, Geneviève Aude Puskas, and Stephany El-Murr. Decisions in changing conditions: the urgency-gating model. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 29(37):11560–71, sep 2009.

[76] David Thura, Julie Beauregard-Racine, Charles-William Fradet, and Paul Cisek. Decision making by urgency gating: theory and experimental support. *Journal of neurophysiology*, 108(11):2912–30, 2012.

[77] Jasper Winkel, Max C Keuken, Leendert van Maanen, Eric-Jan Wagenmakers, and Birte U Forstmann. Early evidence affects later decisions: why evidence accumulation is required to explain response time data. *Psychonomic bulletin & review*, 21(3):777–84, 2014.

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

[79] 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.

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

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

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