Contents

S1 The Diffusion Decision model

This section gives further details of the DDM density and its approximation. Section S1.1 considers a simple version of DDM; section S1.2 discusses the standard DDM; section S1.3 considers the required transformation for the random effects in the DDM; section S1.4 gives further details of the hierarchical DDM.

S1.1 A simple DDM

The simplest diffusion model assumes that the information accumulates continuously by a Wiener diffusion process (Smith, 2000; Smith and Ratcliff, 2015). Let $X(t)$ be the "state of evidence" at the time t. Figure S1 shows that the process begins accumulating evidence at time $t = 0$, with the initial evidence state $X(0) = z$ lying in the range $0 < X(0) < a$. Information continues accumulating at the drift rate v until one of the two boundaries is reached. The decision depends on the boundary reached; i.e., if the process reaches the upper boundary first, response A is made. The sign of the drift rate v (the drift direction) depends on the presented stimulus, i.e., when stimulus A is presented, the drift is positive and the accumulating evidence state tends to increase with time, making it more likely to terminate at the upper boundary and result in response A. Analogously, when stimulus B is presented, the drift rate is negative, and the process tends to end up at the lower boundary, resulting in response B. The drift rate's magnitude is determined by the stimulus discriminability. For example, in a lexical decision task, high frequency words correspond

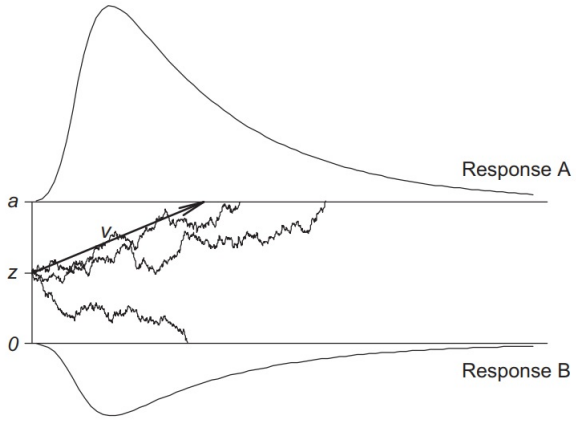

Figure S1: A simple diffusion decision model with constant drift v starts at z and terminates as soon as it reaches one of the thresholds at a or 0, respectively. The figure is used with permission from *Experimental Psychology* 2013; Vol. 60(6):385–402 ©2013 Hogrefe Publishing `www.hogrefe.com https://doi.org/10.1027/1618-3169/a000218.`

to high drift rates and low frequency words correspond to low drift rates. Hence, the drift rate v is used to quantify "ease of processing" (Voss et al., 2004). Large values of drift rate result in fast and accurate responses, whereas small drift rates produce slow and inaccurate decisions.

The stochastic differential equation

$$dX(t) = vdt + sdW(t), \tag{S1}$$

defines the accumulation process, where $W(t)$ is a Brownian motion (Wiener process), v is the drift rate and s is the standard deviation which determines the variability in the sample paths of the process.

The diffusion model parameters are only identified up to a ratio; i.e., all the model parameters can be multiplied by a constant without affecting the predictions. Hence, to make the parameters estimable, one parameter needs to be fixed. It is common in practice to fix s at $s = 1$ (see, e.g., Donkin et al., 2009, and references therein). The other parameters are then expressed in units of infinitesimal standard deviation per unit time[1].

S1.1.1 The Wiener first-passage time density

The observable model random variables are the choices c and the corresponding decision-time t_d (the time it takes for the accumulator to reach a boundary). The joint density of

[1]Ratcliff (1978) fixes $s = 0.1$. The only difference is that the other parameters will all be 10 times smaller than the size of the corresponding parameters if $s = 1$.

the choice and the decision time is called the Wiener first-passage time density, denoted WFPT$(c, t_d|v, a, z)$. Let $f(t_d|v, a, z)$ be the WFPT density [2] describing the chance that the diffusion process is absorbed at time t_d at the lower boundary.

Then,

$$\text{WFPT}(c, t_d|v, a, z) = \begin{cases} f(t_d|v, a, z) & \text{if } c = \text{“lower”}, \\ f(t_d|-v, a, a-z) & \text{if } c = \text{“upper”}. \end{cases}$$

Feller (1968) provides two equivalent representations for the function $f(t_d|v, a, z)$:

$$f^l(t_d|v, a, z) = \frac{\pi}{a^2} \exp\left(-vz - \frac{v^2 t_d}{2}\right) \sum_{k=1}^{\infty} k \sin\left(\frac{k\pi z}{a}\right) \exp\left(-\frac{k^2\pi^2 t_d}{2a^2}\right),$$

and

$$f^s(t_d|v, a, z) = \frac{a}{\sqrt{2\pi t_d^3}} \exp\left(-vz - \frac{v^2 t_d}{2}\right) \sum_{k=-\infty}^{\infty} \left(\frac{z}{a} + 2k\right) \exp\left(-\frac{(z+2ka)^2}{2t_d}\right).$$

The former is called the “large-time” representation and the latter is called the “small-time” representation. Their convergence rates depend on the value of the decision time t_d. The reason for referring to the two representations as large-time and small-time is that $f^l(t_d|v, a, z)$ converges at a faster rate for large values of t_d, whereas $f^s(t_d|v, a, z)$ has a better rate of convergence with small values of t_d (Navarro and Fuss, 2009).

S1.1.2 Approximating the Wiener first-passage time density:

In practice, it is crucial to approximate the WFPT density and evaluate the approximation error. For recent reviews on how to evaluate the density accurately and efficiently, see Foster and Singmann (2021) and references therein. Our work is based on the important paper by Navarro and Fuss (2009) who propose an automatic rule to choose between the two representations and to truncate the series such that the approximation error is less than a predetermined accuracy level ϵ. For mathematical convenience, we replace the starting point z $(0 < z < a)$ with the relative starting point $\omega = z/a$ $(0 < \omega < 1)$. Under the new

[2] To be precise, $f(t_d|v, a, z)$ is not a density in the strict sense as $\int_0^\infty f(t_d|v, a, z)dt_d$ is often not equal to 1. For simplicity, following Gondan et al. (2014); Navarro and Fuss (2009) and Foster and Singmann (2021), we will refer to $f(t_d|v, a, z)$ as a density.

parameterisation, the large-time representation density becomes

$$f^l(t_d|v,a,\omega) = \frac{\pi}{a^2}\exp\left(-va\omega - \frac{v^2 t_d}{2}\right)\sum_{k=1}^{\infty} k\sin\left(k\pi\omega\right)\exp\left(-\frac{k^2\pi^2 t_d}{2a^2}\right).$$

When $v = 0$ and $a = 1$,

$$f^l(t_d|0,1,\omega) = \pi\sum_{k=1}^{\infty} k\sin\left(k\pi\omega\right)\exp\left(-\frac{k^2\pi^2 t_d}{2}\right).$$

It is now straightforward to show that

$$f^l\left(\frac{t_d}{a^2}\mid 0,1,\omega\right) = \pi\sum_{k=1}^{\infty} k\sin\left(k\pi\omega\right)\exp\left(-\frac{k^2\pi^2 t_d}{2a^2}\right),$$

and

$$f^l(t_d|v,a,\omega) = \frac{1}{a^2}\exp\left(-va\omega - \frac{v^2 t_d}{2}\right) f^l\left(\frac{t_d}{a^2}\mid 0,1,\omega\right).$$

Analogously, we can factorise the small-time density similarly, that is,

$$f^s(t_d|v,a,\omega) = \frac{1}{a^2}\exp\left(-va\omega - \frac{v^2 t_d}{2}\right) f^s\left(\frac{t_d}{a^2}\mid 0,1,\omega\right),$$

where

$$f^s\left(t_d\mid 0,1,\omega\right) = \frac{1}{\sqrt{2\pi t_d^3}}\sum_{k=-\infty}^{\infty}(\omega+2k)\exp\left(-\frac{(\omega+2k)^2}{2t_d}\right).$$

In general, both the small-time and large-time densities, denoted by the function $f(t_d|v,a,z)$, can be factorized as

$$f(t_d|v,a,\omega) = \frac{1}{a^2}\exp\left(-va\omega - \frac{v^2 t_d}{2}\right) f\left(\frac{t_d}{a^2}\mid 0,1,\omega\right).$$

The problem of approximating $f(t_d|v,a,\omega)$, a density with three parameters, now becomes a problem of estimating a density with a single parameter $f(t_d|0,1,\omega)$.

Navarro and Fuss suggest the following rule to choose a suitable representation and to truncate the infinite series.

$$f(t_d|0,1,\omega) \approx \begin{cases} \dfrac{1}{\sqrt{2\pi t_d^3}}\displaystyle\sum_{k=-\lfloor(\kappa-1)/2\rfloor}^{\lceil(\kappa-1)/2\rceil}(\omega+2k)\exp\left(-\frac{(\omega+2k)^2}{2t_d}\right) & \text{if } \lambda(t_d) < 0 \\ \pi\displaystyle\sum_{k=1}^{\kappa} k\exp\left(-\frac{k^2\pi^2 t_d}{2}\right)\sin\left(k\pi\omega\right) & \text{if } \lambda(t_d) \geq 0, \end{cases}$$

where

$$\lambda(t_d) = 2 + \sqrt{-2t_d \log(2\epsilon\sqrt{2\pi t_d})} - \sqrt{\frac{-2\log(\pi t_d \epsilon)}{\pi^2 t_d}},$$

and ϵ is a predefined level of accuracy. The number of terms needed in the truncated series is denoted by κ and is determined by the following rule:

- large-time representation: $\kappa = \left\lceil \max\left\{ \sqrt{\frac{-2\log(\pi t_d \epsilon)}{\pi^2 t_d}}, \frac{1}{\pi\sqrt{t_d}} \right\} \right\rceil$,

- small-time representation: $\kappa = \left\lceil \max\left\{ 2 + \sqrt{-2t_d \log(2\epsilon\sqrt{2\pi t_d})}, 1 + \sqrt{t_d} \right\} \right\rceil$,

where $\lceil \cdot \rceil$ and $\lfloor . \rfloor$ are the ceiling and floor functions, respectively.

Remark: In the simple DDM, we do not observe the decision-time t_d, but the response time $RT = t_d + \tau$. Hence, the simple DDM density is the WFPT density with $t_d = RT - \tau$.

S1.2 The standard diffusion decision model

The drawback with the simple model is that it cannot capture many important RT distributions observed from empirical data. To address this, Ratcliff (1978) extends the model by assuming that the drift rate v, the starting point z and the non-decision time τ can change from trial to trial. More specifically, at each trial, the value of the drift rate v is assumed to be drawn from a Normal distribution with mean v and variance s_v^2. Independently, the starting point z and the non-decision time τ are drawn from uniform distributions centred at z and τ, respectively. The width of the uniform distributions are s_z (of the starting point) and s_τ (of the non-decision time). With these assumptions, the density of the standard diffusion model is defined as:

$$\text{DDM}(c, RT) = \frac{1}{s_z s_\tau} \int_{z-\frac{s_z}{2}}^{z+\frac{s_z}{2}} \int_{\tau-\frac{s_\tau}{2}}^{\tau+\frac{s_\tau}{2}} \int_{-\infty}^{\infty} \text{WFPT}(c, t_d | v, a, z) N(v | v, s_v^2) dv d\tau dz,$$

with $t_d = RT - \tau$. Similarly to the WFPT density, the standard diffusion density has two equivalent representations. Denote the large-time and the small-time representations of the standard diffusion density by $\text{DDM}^l(c, RT)$ and $\text{DDM}^s(c, RT)$, respectively. Then, the

densities are:

$$\begin{aligned}
\text{DDM}^{l}(\text{"lower"}, RT) &= \frac{1}{s_z s_\tau} \int_{z-\frac{s_z}{2}}^{z+\frac{s_z}{2}} \int_{\tau-\frac{s_\tau}{2}}^{\tau+\frac{s_\tau}{2}} \int_{-\infty}^{\infty} \frac{\pi}{a^2} \exp\left(-vz - \frac{v^2 t_d}{2}\right) \sum_{k=1}^{\infty} k \exp\left(-\frac{k^2\pi^2 t_d}{2a^2}\right) \\
&\quad \times \sin\left(\frac{k\pi z}{a}\right) N(v|v, s_v^2) dv d\tau dz \\
&= \frac{1}{s_z s_\tau} \int_{z-\frac{s_z}{2}}^{z+\frac{s_z}{2}} \int_{\tau-\frac{s_\tau}{2}}^{\tau+\frac{s_\tau}{2}} \frac{\pi}{a^2\sqrt{1+t_d s_v^2}} \exp\left(-\frac{v^2 t_d + 2vz - z^2 s_v^2}{2(t_d s_v^2 + 1)}\right) \\
&\quad \times \sum_{k=1}^{\infty} k \sin\left(\frac{\pi k z}{a}\right) \exp\left(-\frac{\pi^2 k^2 t_d}{2a^2}\right) d\tau dz.
\end{aligned}$$

$$\begin{aligned}
\text{DDM}^{s}(\text{"lower"}, RT) &= \frac{1}{s_z s_\tau} \int_{z-\frac{s_z}{2}}^{z+\frac{s_z}{2}} \int_{\tau-\frac{s_\tau}{2}}^{\tau+\frac{s_\tau}{2}} \int_{-\infty}^{\infty} \frac{a t_d^{-3/2}}{\sqrt{2\pi}} \exp\left(-vz - \frac{v^2 t_d}{2}\right) \\
&\quad \times \sum_{k=-\infty}^{\infty} \left(\frac{z}{a} + 2k\right) \exp\left(-\frac{(z+2ka)^2}{2t_d}\right) N(v|v, s_v^2) dv d\tau dz \\
&= \frac{1}{s_z s_\tau} \int_{z-\frac{s_z}{2}}^{z+\frac{s_z}{2}} \int_{\tau-\frac{s_\tau}{2}}^{\tau+\frac{s_\tau}{2}} \frac{a t_d^{-3/2}}{\sqrt{2\pi(1+t_d s_v^2)}} \exp\left(-\frac{v^2}{2s_v^2} + \frac{(v - z s_v^2)^2}{s_v^2(1+t_d s_v^2)}\right) \\
&\quad \times \sum_{k=-\infty}^{\infty} \left(\frac{z}{a} + 2k\right) \exp\left(-\frac{(z+2ka)^2}{2t_d}\right) d\tau dz.
\end{aligned}$$

The integrals with respect to z and τ are analytically intractable and must be approximated numerically using Gaussian quadrature.

S1.3 Transformations

The standard DDM has seven main parameters satisfying several constraints. The variability parameters s_v, s_z, and s_τ are positive. The boundary separation must be greater than the upper bound of the starting point, i.e., $a > z + 0.5s_z$. The lower bound of the starting point and the non-decision time must be greater than zero, i.e., $z - 0.5s_z > 0$, and $\tau - 0.5s_\tau > 0$. The full hierarchical structure considered here allows all the parameters to vary across subjects; hence, each subject has a vector of random effects,

$$\boldsymbol{\eta} = (v, s_v, a, z, s_z, \tau, s_\tau).$$

The correlation between the subject parameters is captured by using a multivariate Gaussian with a full covariance matrix. To do so, we transform the subject parameters as follows, and denote the transformed parameters by

$\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7)$, with

$$\begin{cases} \alpha_1 &= v, \alpha_2 = \log s_v, \\ \alpha_3 &= \log(a - z - 0.5s_z), \alpha_4 = \log(z - 0.5s_z), \alpha_5 = \log(s_z), \\ \alpha_6 &= \log(\tau - 0.5s_\tau), \alpha_7 = \log s_\tau. \end{cases} \tag{S2}$$

S1.3.1 Hierarchical DDM - Experiment 1

The transformations for the DDM in fitting the real data are discussed in Section 5.1. Each subject has the following parameters

$$(v^{(hf)}, v^{(lf)}, v^{(vlf)}, v^{(nw)}, s_v, a^{(speed)}, a^{(acc.)}, z^{(speed)}, z^{(acc.)}, s_z, \tau, s_\tau).$$

To capture the constraints on the DDM parameters, we apply the following transformations:

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \alpha_{10}, \alpha_{11}, \alpha_{12}),$$

with

$$\begin{cases} \alpha_1 &= v^{(hf)}, \alpha_2 = v^{(lf)}, \alpha_3 = v^{(vlf)}, \alpha_4 = v^{(nw)}, \alpha_5 = \log s_v, \\ \alpha_6 &= \log(a^{(speed)} - z^{(speed)} - 0.5s_z), \alpha_7 = \log(z^{(speed)} - 0.5s_z), \\ \alpha_8 &= \log(a^{(acc.)} - z^{(acc.)} - 0.5s_z), \alpha_9 = \log(z^{(acc.)} - 0.5s_z), \\ \alpha_{10} &= \log(s_z), \alpha_{11} = \log(\tau - 0.5s_\tau), \alpha_{12} = \log s_\tau. \end{cases}$$

S1.3.2 Simulation study for EAMs

To capture the natural constraints of the random effects $\boldsymbol{\eta}$ for simulation study discussed in section S7 of the online supplement, we apply the following transformations. Denote the transformed random effects by $\boldsymbol{\alpha}$, we have

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9),$$

with

$$\begin{cases} \alpha_1 &= v^{(h)}, \alpha_2 = v^{(m)}, \alpha_3 = v^{(e)}, \alpha_4 = \log s_v, \\ \alpha_5 &= \log(a - z - 0.5s_z), \alpha_6 = \log(z - 0.5s_z), \alpha_7 = \log(s_z), \\ \alpha_8 &= \log(\tau - 0.5s_\tau), \alpha_9 = \log s_\tau. \end{cases} \tag{S3}$$

S1.3.3 EAMs - real data

RegLBA We discuss the transformations applied for the RegLBA considered in Section 5.2. The constraints on the random effects are: $b > A$ and all parameters must be positive. To capture these conditions, we apply the following transformations:

$$\begin{cases} \alpha_1 &= \log(b - A), \\ \alpha_2 &= \log(A), \\ \alpha_3 &= v_s^s, \alpha_4 = v_m^s, \alpha_5 = v_s^m, \alpha_6 = v_m^m, \alpha_7 = v_c, \alpha_8 = v_e, \\ \alpha_9 &= \log \tau. \end{cases} \tag{S4}$$

Denote the transformed random effects by $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9)$.

DDM We use the following transformations to enforce the constraints on the DDM parameters in the DDM discussed in Section 5.2. Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8)$ denote the transformed parameters, with

$$\begin{cases} \alpha_1 &= v^0, \alpha_2 = v, \alpha_3 = \log s_v, \\ \alpha_4 &= \log(a - z - 0.5s_z), \alpha_5 = \log(z - 0.5s_z), \alpha_6 = \log(s_z), \\ \alpha_7 &= \log(\tau - 0.5s_\tau), \alpha_8 = \log s_\tau. \end{cases} \tag{S5}$$

S1.4 Hierarchical Diffusion Model

This section gives further details on the hierarchical DDM. Denote by $p(y_{ij}|\boldsymbol{\alpha}_j)$ the density of an EAM; the HEAM is specified as follows.

1. Conditional density:

$$p(\boldsymbol{y}|\boldsymbol{\alpha}_{1:J}) = \prod_{j=1}^{J} p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j),$$
$$p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j) = \prod_{i=1}^{n_j} p(y_{ij}|\boldsymbol{\alpha}_j), \text{ for } j = 1, \ldots, J, \; i = 1, \ldots, n_j.$$

2. A multivariate normal distribution for the random effects

$$\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} \overset{ind.}{\sim} N(\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}).$$

3. Priors for the model parameters

   (a) Group-level mean $\boldsymbol{\mu}_{\boldsymbol{\alpha}} \sim N(\boldsymbol{\mu}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}})$.

   (b) Group-level covariance $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$.

      i. Informative prior $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} \sim \text{IW}\left(k_\alpha, \boldsymbol{\Psi}\right)$, with the hyperparameters $k_\alpha = 20$ and $\boldsymbol{\Psi} = \boldsymbol{I}$.

      ii. Marginally noninformative prior by Huang and Wand (2013)

$$\begin{aligned}
&\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|a_1, \ldots, a_{D_\alpha} \sim \text{IW}\left(D_\alpha + 1, \boldsymbol{\Psi}\right), \boldsymbol{\Psi} = 4\text{diag}\left(\frac{1}{a_1}, \ldots, \frac{1}{a_{D_\alpha}}\right), \\
&a_1, \ldots, a_{D_\alpha} \sim \text{IG}\left(\frac{1}{2}, 1\right).
\end{aligned}$$

Remark 2

The model specified in equation (1) in section 2.3 assumes the same transformations are applied to the random effects and the model parameters. However, in practice, this is not always the case. For example, Section 5 considers a real example in which many different values of the drift rate can be modelled using just two parameters. This parsimonious parameterisation is crucial in obtaining reliable estimates as the sample size is small. Hence, we will introduce several new notations so that the model specification is more flexible, and hence allows various parameterisations. First, let $\boldsymbol{\omega}$ and $\boldsymbol{\psi}$ denote the set of model parameters with and without transformation, respectively. Similarly, we denote the natural and transformed random effects by $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$.

The linking equation is

$$\begin{aligned}
&p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j, \boldsymbol{\beta}, \boldsymbol{X}_j) = \prod_{i=1}^{n_j} p(y_{ij}|\boldsymbol{\omega}_{ij}), \\
&\boldsymbol{\omega}_{ij} = \boldsymbol{\xi}_j + \boldsymbol{\beta} X_{ij}, \text{ for } j = 1, \ldots, J, \, i = 1, \ldots, n_j,
\end{aligned} \tag{S6}$$

where $\boldsymbol{\xi}_j$ denotes the random effects that take part in the linking equation. The transformation on the random effects is needed, as we want to use an unconstrained Gaussian distribution to capture the correlation of the effects. Notice that it is $\boldsymbol{\xi}_j$ and not $\boldsymbol{\alpha}_j$, that appears in the linking equation. The transformed model parameters $\boldsymbol{\omega}_{ij}$ are needed as the

right side of the linking equation has an unrestricted range. Equation (S6) gives more modelling flexibility and allows parsimonious parameterisations such as the one in Section 5.

S2 Variational Bayes

This section gives further details of the variational Bayes method. Subsection S2.1 discusses variance reduction using the reparameterisation trick. Subsection S2.2 discusses the learning rates. Subsection S2.3 gives details for estimating EAMs. Subsection S2.4 discusses further details of the VB method for estimating a hierarchical DDM.

S2.1 The "reparameterisation trick"

The performance of stochastic gradient ascent depends greatly on the variance of the noisy gradient estimate $\widehat{\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda})}$. Its performance can be greatly improved by employing variance reduction methods. A popular variance reduction method is the so-called "reparameterisation trick" (Kingma and Welling, 2013; Rezende et al., 2014). If we can write $\boldsymbol{\theta} \sim q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ as $\boldsymbol{\theta} = u(\boldsymbol{\epsilon}; \boldsymbol{\lambda})$, with $\boldsymbol{\epsilon} \sim f_\epsilon$ not depending on $\boldsymbol{\lambda}$, then the lower bound and its gradient can be written as the expectations

$$\begin{aligned} \mathcal{L}(\boldsymbol{\lambda}) &= E_{f_\epsilon}\left[\log p(\boldsymbol{y}, u(\boldsymbol{\epsilon}; \boldsymbol{\lambda})) - \log q_{\boldsymbol{\lambda}}(u(\boldsymbol{\epsilon}; \boldsymbol{\lambda}))\right], \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda}) &= E_{f_\epsilon}\Big[\nabla_{\boldsymbol{\lambda}} u(\boldsymbol{\epsilon}; \boldsymbol{\lambda})\left[\nabla_\theta \log p(\boldsymbol{y}, \boldsymbol{\theta}) - \nabla_\theta \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})\right]\Big]. \end{aligned} \tag{S7}$$

By sampling $\boldsymbol{\epsilon} \sim f_\epsilon$, it is straightforward to obtain the unbiased estimates of the lower bound and its gradient

$$\begin{aligned} \widehat{\mathcal{L}(\boldsymbol{\lambda})} &:= \frac{1}{N}\sum_{i=1}^{N}\left[\log p(\boldsymbol{y}, u(\boldsymbol{\epsilon}^{(i)}; \boldsymbol{\lambda})) - \log q_{\boldsymbol{\lambda}}(u(\boldsymbol{\epsilon}^{(i)}; \boldsymbol{\lambda}))\right], \\ \widehat{\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda})} &:= \frac{1}{N}\sum_{i=1}^{N}\left[\nabla_{\boldsymbol{\lambda}} u(\boldsymbol{\epsilon}^{(i)}; \boldsymbol{\lambda})\left(\nabla_\theta \log p(\boldsymbol{y}, \boldsymbol{\theta}^{(i)}) - \nabla_\theta \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}^{(i)})\right)\right], \end{aligned} \tag{S8}$$

with $\boldsymbol{\epsilon}^{(i)} \sim f_\epsilon, i = 1, \ldots, N$. We use $N = 10$ in our applications.

S2.2 Learning rates and stopping rule

S2.2.1 ADADELTA

The elements of the vector $\boldsymbol{\lambda}$ may need very different step sizes (learning rates) during the search, to account for scale or the geometry of the space. We set the step sizes adaptively

using the ADADELTA method (Zeiler, 2012), with different step sizes for each element of $\boldsymbol{\lambda}$. At iteration $t+1$, the ith element λ_i of $\boldsymbol{\lambda}$ is updated as

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \Delta\lambda_i^{(t)}.$$

The step size $\Delta\lambda_i^{(t)} := \rho_i^{(t)} g_{\lambda_i}^{(t)}$, where $g_{\lambda_i}^{(t)}$ denotes the ith component of $\widehat{\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}^{(t)})}$ and

$$\rho_i^{(t)} := \frac{\sqrt{E(\Delta_{\lambda_i}^2)^{(t-1)} + \xi}}{\sqrt{E(g_{\lambda_i}^2)^{(t)} + \xi}},$$

where ξ is a small positive constant, with

$$E(\Delta_{\lambda_i}^2)^{(t)} = vE(\Delta_{\lambda_i}^2)^{(t-1)} + (1-v)(\Delta_{\lambda_i}^{(t)})^2,$$
$$E(g_{\lambda_i}^2)^{(t)} = vE(g_{\lambda_i}^2)^{(t-1)} + (1-v)(g_{\lambda_i}^{(t)})^2.$$

The ADADELTA default settings are $\xi = 10^{-6}, v = 0.95$ with initialisation $E(\Delta_{\lambda_i}^2)^{(0)} := E(g_{\lambda_i}^2)^{(0)} = 0$. However, in our experiments, we use $\xi = 10^{-7}$ as we obtain better results.

S2.2.2 ADAM

Another popular choice for selecting the learning rate is ADAM (Kingma and Ba, 2014). Recall that the vector variational parameters used in the variational densities considered in this paper is $\boldsymbol{\lambda} = (\boldsymbol{\mu}_{\boldsymbol{\lambda}}, B_{\boldsymbol{\lambda}}, d_{\boldsymbol{\lambda}})$. Notice also that, for notation simplicity, in this section we shall omit the subscript indices, i.e., we use $\boldsymbol{\mu}$ instead of $\boldsymbol{\mu}_{\boldsymbol{\lambda}}$. We first explain the updating rule using ADAM method for the first component $\boldsymbol{\mu}$. ADAM requires several tuning parameters such as a stepsize α_μ, and two exponential decay rates for the moment estimates denoted by β_1 and β_2.[3] The ADAM algorithm for other variational parameters B and $\boldsymbol{d}$ is performed in a similar manner. We fix $\beta_1 = 0.9, \beta_2 = 0.99$, and $\epsilon = 10^{-8}$. For the stepsizes, we set the stepsize of $\boldsymbol{\mu}$ to be equal to 0.01, and choose the same stepsize of 0.001 for B and $\boldsymbol{d}$.

[3] In the paper, we do use α and β to denote the random and fixed effects when introducing the models. However, as we are purely discussing the general optimization algorithm without being specific about any model, the duplication in notation here should not cause any confusion.

Algorithm S1 ADAM

1. Initialisation: select initial values: $\boldsymbol{\mu}^{(0)}$ and set $m_0 = v_0 = 0$.
2. While not convergence do:
   (1) $t \leftarrow t + 1$
   (2) Calculate $g_t \leftarrow \nabla_{\boldsymbol{\mu}} \widehat{\mathcal{L}(\boldsymbol{\lambda}^{(t)})}$
   (3) Update the biased first moment estimate

$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

   (4) Correct the biased first moment estimate

$$\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t}$$

   (5) Update the biased second moment estimate

$$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

   (6) Correct the biased second moment estimate

$$\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t}$$

   (7) Update parameters

$$\boldsymbol{\mu}^{(t)} \leftarrow \boldsymbol{\mu}^{(t-1)} - \alpha_\mu \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

S2.2.3 Stopping rule

A popular criterion for the search algorithm is to stop when the moving average lower bound estimate $\overline{\text{LB}}_t = \frac{1}{m} \sum_{i=t-m+1}^{t} \widehat{\mathcal{L}(\boldsymbol{\lambda}^{(i)})}$ does not improve after k consecutive iterations (Tran et al., 2017). We find that the choice of $m = 100$ and $k = 50$ works well in most cases.

S2.3 Implementing VB for approximating the EAMs

In Section 3.2.1 we introduce two VB classes. The first one, which we call VB 1, uses the hybrid Gaussian VB is of the form

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = N(\boldsymbol{\theta}_1 | \boldsymbol{\mu}_{\boldsymbol{\lambda}}, B_{\boldsymbol{\lambda}} B_{\boldsymbol{\lambda}}^\top + D_{\boldsymbol{\lambda}}^2) \text{IW}(\boldsymbol{\theta}_2 | \nu, \boldsymbol{\Psi}'),$$

where $\boldsymbol{\theta}_1 = (\boldsymbol{\alpha}_{1:J}, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \log \boldsymbol{a}), \nu = 2D_\alpha + J + 1$, and $\boldsymbol{\Psi}' = \sum_{j=1}^{J} (\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})^\top + 4 \operatorname{diag}\left(a_1^{-1}, \ldots, a_{D_\alpha}^{-1}\right)$. The second class, called VB 2, is a special case of the first one in the sense that that the random effects $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_J$ are further assumed to be conditionally independent, i.e.,

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = N(\boldsymbol{\theta}_2 | \boldsymbol{\mu}_{J+1}, B_{J+1} B_{J+1}^\top + D_{J+1}^2) \text{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} | \nu, \boldsymbol{\Psi}') \prod_{j=1}^{J} N(\boldsymbol{\alpha}_j | \boldsymbol{\mu}_j, B_j B_j^\top + D_j^2),$$

where $\boldsymbol{\theta}_1 = (\boldsymbol{\alpha}_{1:J}, \boldsymbol{\psi})$ with $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \log \boldsymbol{a})$. The best approximate distribution in terms of minimizing the KL divergence is found by maximizing the lower bound $\mathcal{L}(\boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$. Starting from some initial value $\boldsymbol{\lambda}^{(0)}$, the values of for $\boldsymbol{\lambda}$ are updated iteratively as

$$\boldsymbol{\lambda}^{(t+1)} = \boldsymbol{\lambda}^{(t)} + \boldsymbol{\rho}_t \odot \widehat{\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}^{(t)})}, \tag{S9}$$

where $\boldsymbol{\rho}_t$ is a vector of step sizes or learning rates, $\odot$ denotes the element-wise product of two vectors, and $\widehat{\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})}$ is an unbiased estimate of the gradient of $\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$. Dao et al.

(2022) show the gradient of the lower bound with respect to the variational parameters is[4]

$$\begin{aligned}
\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda}) &= E_{\boldsymbol{\epsilon}}\left[\nabla_{\boldsymbol{\lambda}} u(\boldsymbol{\epsilon};\boldsymbol{\lambda})\nabla_{\boldsymbol{\theta}_1}\log\left(\frac{p(\boldsymbol{\theta}_1,\boldsymbol{y})}{q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1)}\right)\right] \\
&= E_{(\boldsymbol{\epsilon},\boldsymbol{\theta}_2)}\left[\nabla_{\boldsymbol{\lambda}} u(\boldsymbol{\epsilon};\boldsymbol{\lambda})\nabla_{\boldsymbol{\theta}_1}\log\left(\frac{p(\boldsymbol{\theta}_1,\boldsymbol{y})p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y})}{q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y})}\right)\right] \\
&= E_{(\boldsymbol{\epsilon},\boldsymbol{\theta}_2)}\left[\nabla_{\boldsymbol{\lambda}} u(\boldsymbol{\epsilon};\boldsymbol{\lambda})\nabla_{\boldsymbol{\theta}_1}\log\left(\frac{p(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{y})}{q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}\right)\right] \\
&= E_{(\boldsymbol{\epsilon},\boldsymbol{\theta}_2)}\Bigg[\nabla_{\boldsymbol{\lambda}} u(\boldsymbol{\epsilon};\boldsymbol{\lambda})\Big(\nabla_{\boldsymbol{\theta}_1}\log p(y|\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) + \nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) \\
&\qquad - \nabla_{\boldsymbol{\theta}_1}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) - \nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y})\Big)\Bigg].
\end{aligned}$$

To implement VB, we need to derive $\nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{y}|\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)$, $\nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)$, $\nabla_{\boldsymbol{\theta}_1}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1)$ and $\nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y})$. We note that, because

$$E_{(\boldsymbol{\epsilon},\boldsymbol{\theta}_2)}\left[\nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y})\right] = E_{\boldsymbol{\epsilon}}\left[\int \nabla_{\boldsymbol{\theta}_1} p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y})d\boldsymbol{\theta}_2\right] = 0,$$

we can remove the term $\nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y})$ from the calculation of $\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda})$. However, this term also plays the role of a control variate and is useful in reducing the variance of the gradient estimate in finite sample sizes (recall we use $N = 10$). We therefore include this term in all the computations reported in the paper. The data-parameter joint density is

$$\begin{aligned}
p(\boldsymbol{y},\boldsymbol{\alpha}_{1:J},\boldsymbol{\beta},\boldsymbol{\mu}_{\boldsymbol{\alpha}},\boldsymbol{\Sigma}_{\boldsymbol{\alpha}},\log\boldsymbol{a}) =& N(\boldsymbol{\beta}|\boldsymbol{\mu}_{\boldsymbol{\beta}},\boldsymbol{\Sigma}_{\boldsymbol{\beta}})N(\boldsymbol{\mu}_{\boldsymbol{\alpha}}|\boldsymbol{\mu}_{\boldsymbol{\mu}},\boldsymbol{\Sigma}_{\boldsymbol{\mu}})\text{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu,\boldsymbol{\Psi})\times \\
&\prod_{d=1}^{D_\alpha}\text{IG}(a_d|1/2,1)\,|J_{a_d\to\log a_d}|\prod_{j=1}^{J}p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j)N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}},\boldsymbol{\Sigma}_{\boldsymbol{\alpha}})
\end{aligned}$$

where $J_{a_d\to\log a_d} = a_d$ is the Jacobian of the transformation.

S2.3.1 Deriving $\nabla_{\boldsymbol{\theta}_1}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)$

VB 1

Ong et al. (2018) show that $\nabla_{\boldsymbol{\theta}_1}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\theta}_1}\log N(\boldsymbol{\theta}_1|\boldsymbol{\mu}_{\boldsymbol{\lambda}}, B_{\boldsymbol{\lambda}}B_{\boldsymbol{\lambda}}^\top + D_{\boldsymbol{\lambda}}^2) = -(B_{\boldsymbol{\lambda}}B_{\boldsymbol{\lambda}}^\top + D_{\boldsymbol{\lambda}}^2)^{-1}(\boldsymbol{\theta}_1 - \boldsymbol{\mu}_{\boldsymbol{\lambda}})$. We then use the Woodbury formula (Petersen and Pedersen, 2012) to

[4]Here $\boldsymbol{\theta}_2$ represents $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$.

compute the inverse using

$$(B_{\boldsymbol{\lambda}}B_{\boldsymbol{\lambda}}^\top + D_{\boldsymbol{\lambda}}^2)^{-1} = D_{\boldsymbol{\lambda}}^{-2} - D_{\boldsymbol{\lambda}}^{-2}B_{\boldsymbol{\lambda}}(I + B_{\boldsymbol{\lambda}}^\top D_{\boldsymbol{\lambda}}^{-2}B_{\boldsymbol{\lambda}})^{-1}B_{\boldsymbol{\lambda}}^\top D_{\boldsymbol{\lambda}}^{-2}.$$

This is computationally efficient because it only requires finding the inverses of the diagonal matrix $D_{\boldsymbol{\lambda}}$ and of $(I + B_{\boldsymbol{\lambda}}^\top D_{\boldsymbol{\lambda}}^{-2}B_{\boldsymbol{\lambda}})$, which is a much smaller $r \times r$ matrix.

VB 2

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1) = N(\boldsymbol{\theta}_2|\boldsymbol{\mu}_{J+1}, B_{J+1}B_{J+1}^\top + D_{J+1}^2)\prod_{j=1}^{J} N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_j, B_jB_j^\top + D_j^2),$$

Similar to VB 1, we have

$$\nabla_{\boldsymbol{\alpha}_j}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\alpha}_j}\log N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_j, B_jB_j^\top + D_j^2) = -(B_jB_j^\top + D_j^2)^{-1}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_j), \text{ for } j = 1, \ldots, J;$$

and

$$\nabla_{\boldsymbol{\psi}}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\psi}}\log N(\boldsymbol{\psi}|\boldsymbol{\mu}_{J+1}, B_{J+1}B_{J+1}^\top + D_{J+1}^2) = -(B_{J+1}B_{J+1}^\top + D_{J+1}^2)^{-1}(\boldsymbol{\psi} - \boldsymbol{\mu}_{J+1}).$$

Then, $\nabla_{\boldsymbol{\theta}_1}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1) = (\nabla_{\boldsymbol{\alpha}_1}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1), \ldots, \nabla_{\boldsymbol{\alpha}_J}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1), \nabla_{\boldsymbol{\psi}}\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1))$

S2.3.2 Deriving $\nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$

The section derives the gradients required in the hybrid Gaussian VB for the new model. First, we derive $\nabla_{\boldsymbol{\theta}_1}\log p(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$.

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\alpha}_j}\log p(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \frac{\partial}{\partial\boldsymbol{\alpha}_j}\log p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j, \boldsymbol{\beta}, \boldsymbol{X}_j) \\
&= \sum_{i=1}^{n_j}\frac{\partial}{\partial\boldsymbol{\alpha}_j}\log p(y_{ij}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}, X_{ij}) \\
&= \sum_{i=1}^{n_j}\left(\frac{\partial\widetilde{\boldsymbol{\alpha}}_{ij}}{\partial\boldsymbol{\alpha}_j^\top}\right)^\top\frac{\partial}{\partial\widetilde{\boldsymbol{\alpha}}_{ij}}\log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) \\
&= \sum_{i=1}^{n_j}\left(\frac{\partial\widetilde{\boldsymbol{\alpha}}_{ij}}{\partial\boldsymbol{\alpha}_j^\top}\right)^\top\left(\frac{\partial\boldsymbol{\psi}_{ij}}{\partial\widetilde{\boldsymbol{\alpha}}_{ij}^\top}\right)^\top\frac{\partial}{\partial\boldsymbol{\psi}_{ij}}\log p(y_{ij}|\boldsymbol{\psi}_{ij}) \\
&= \sum_{i=1}^{n_j}\left(\frac{\partial\boldsymbol{\psi}_{ij}}{\partial\widetilde{\boldsymbol{\alpha}}_{ij}^\top}\right)^\top\frac{\partial}{\partial\boldsymbol{\psi}_{ij}}\log p(y_{ij}|\boldsymbol{\psi}_{ij});
\end{aligned}$$

$\boldsymbol{\psi}_{ij}$ denotes the model parameters in the natural scale. The derivatives $\frac{\partial}{\partial \boldsymbol{\psi}_{ij}} p(y_{ij}|\boldsymbol{\psi}_{ij})$ for LBA and DDM densities are given below. Note that $\widetilde{\boldsymbol{\alpha}}_{ij} = \boldsymbol{\alpha}_j + \boldsymbol{\beta} X_{ij}$, so $\frac{\partial \widetilde{\boldsymbol{\alpha}}_{ij}}{\partial \boldsymbol{\alpha}_j^\top} = I$.

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \sum_{j=1}^{J} \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j, \boldsymbol{\beta}, \boldsymbol{X}_j) \\ &= \sum_{j=1}^{J} \sum_{i=1}^{n_j} \frac{\partial}{\partial \boldsymbol{\beta}} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}).\end{aligned}$$

To derive the derivative with respect to the matrix $\boldsymbol{\beta}$, we derive the derivative with respect to its rows. Let $\boldsymbol{\beta}_k$ be the kth row in the coefficient matrix $\boldsymbol{\beta}, k = 1, \ldots, D_\alpha$. We have

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\beta}_k} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) &= \frac{\partial}{\partial \alpha_{ij;k}} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) \frac{\partial \alpha_{ij;k}}{\partial \boldsymbol{\beta}_k} \\ &= \frac{\partial}{\partial \alpha_{ij;k}} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) \frac{\partial \alpha_{ij;k}}{\partial \boldsymbol{\beta}_k},\end{aligned}$$

where $\widetilde{\alpha}_{ij;k}, \alpha_{j;k}$ are the kth element of vectors $\widetilde{\boldsymbol{\alpha}}_{ij}$ and $\boldsymbol{\alpha}_j$, respectively. Since $\alpha_{ij;k} = \alpha_{j;k} + \boldsymbol{\beta}_k X_{ij}^\top$, we have $\frac{\partial \alpha_{ij;k}}{\partial \boldsymbol{\beta}_k} = X_{ij}$ and therefore,

$$\frac{\partial}{\partial \boldsymbol{\beta}} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{\beta}_1^\top} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) \\ \vdots \\ \frac{\partial}{\partial \boldsymbol{\beta}_{D_\alpha}^\top} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \alpha_{ij;1}} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) X_{ij}^\top \\ \vdots \\ \frac{\partial}{\partial \alpha_{ij;D_\alpha}} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}) X_{ij}^\top \end{bmatrix},$$

and

$$\frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \frac{\partial}{\partial \boldsymbol{\beta}} \log p(y_{ij}|\widetilde{\boldsymbol{\alpha}}_{ij}).$$

S2.3.2.1 LBA density

Brown and Heathcote (2008) show the LBA density (the joint density over response time $RT = t$ and response choice $RE = c$) is

$$\text{LBA}(c, t|b, A, \boldsymbol{v}, s, \tau) = f_c(t) \times \prod_{k \neq c} (1 - F_k(t)),$$

where

$$F_c(t) = 1 + \frac{b - A - (t - \tau)v^c}{A} \Phi(\overline{\omega}_1 - \overline{\omega}_2) - \frac{b - (t - \tau)v^c}{A} \Phi(\overline{\omega}_1) + \frac{1}{\overline{\omega}_1} \phi(\overline{\omega}_1 - \overline{\omega}_2) - \frac{1}{\overline{\omega}_2} \phi(\overline{\omega}_1)$$

and

$$f_c(t) = \frac{1}{A}\left[-v^c\Phi(\overline{\omega}_1 - \overline{\omega}_2) + s\phi(\overline{\omega}_1 - \overline{\omega}_2) + v^c\Phi(\overline{\omega}_1) - s\phi(\overline{\omega}_1)\right]$$

with $\overline{\omega}_1 = \dfrac{b-(t-\tau)v^c}{(t-\tau)s}$, $\overline{\omega}_2 = \dfrac{A}{(t-\tau)s}$, where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the distribution function and density of the standard normal distribution, respectively. Hence,

$$\frac{\partial}{\partial \boldsymbol{\psi}_{ij}}\text{LBA}(y_{ji}|\boldsymbol{\psi}_{ij}) = \frac{\partial f_c(t)}{\partial \boldsymbol{\psi}_{ij}}(1 - F_{k\neq c}(t)) - \frac{\partial F_{k\neq c}(t)}{\partial \boldsymbol{\psi}_{ij}} f_c(t)$$

The partial derivatives of $f_c(t)$ with respect to $\boldsymbol{\psi}$[5] are

$$\frac{\partial}{\partial b} f_c(t) = \frac{1}{A}\Bigg[- v^c\phi(\overline{\omega}_1 - \overline{\omega}_2) + s\phi'(\overline{\omega}_1 - \overline{\omega}_2) + v^c\phi(\overline{\omega}_1) - s\phi'(\overline{\omega}_1)\Bigg]\frac{\partial \overline{\omega}_1}{\partial b};$$

$$\frac{\partial}{\partial A} f_c(t) = -\frac{1}{A}\Bigg[f_c(t) - v^c\phi(\overline{\omega}_1 - \overline{\omega}_2) + s\phi'(\overline{\omega}_1 - \overline{\omega}_2)\Bigg]\frac{\partial \overline{\omega}_1}{\partial A};$$

$$\frac{\partial}{\partial v^c} f_c(t) = \frac{1}{A}\Bigg[- \Phi(\overline{\omega}_1 - \overline{\omega}_2) + \Phi(\overline{\omega}_1)\Bigg] + \frac{1}{A}\Bigg[- v^c\phi(\overline{\omega}_1 - \overline{\omega}_2) + s\phi'(\overline{\omega}_1 - \overline{\omega}_2) + v^c\phi(\overline{\omega}_1) - s\phi'(\overline{\omega}_1)\Bigg]\frac{\partial \overline{\omega}_1}{\partial v^c};$$

$$\frac{\partial}{\partial \tau} f_c(t) = \frac{1}{A}\Bigg[-v^c\phi(\overline{\omega}_1 - \overline{\omega}_2) + s\phi'(\overline{\omega}_1 - \overline{\omega}_2)\Bigg]\left(\frac{\partial \overline{\omega}_1}{\partial \tau} - \frac{\partial \overline{\omega}_2}{\partial \tau}\right) + \frac{1}{A}\Bigg[v^c\phi(\overline{\omega}_1) - s\phi'(\overline{\omega}_1)\Bigg]\frac{\partial \overline{\omega}_1}{\partial \tau}.$$

The partial derivatives of $F_c(t)$ with respect to $\boldsymbol{\psi}$[6] are:

[5] We omit the subscript ij for simplicity.
[6] We omit the subscript ij for simplicity.

$$
\begin{aligned}
\frac{\partial}{\partial b}F_c(t) &= \left[\frac{1}{A}\Phi(\overline{\omega}_1-\overline{\omega}_2)+\frac{b-A-(t-\tau)v^c}{A}\phi(\overline{\omega}_1-\overline{\omega}_2)\frac{\partial\overline{\omega}_1}{\partial b}\right]\\
&+\frac{1}{\overline{\omega}_2}\left[\phi'(\overline{\omega}_1-\overline{\omega}_2)-\phi'(\overline{\omega}_1)\right]\frac{\partial\overline{\omega}_1}{\partial b}+\left[-\frac{1}{A}\Phi(\overline{\omega}_1)-\frac{b-(t-\tau)v^c}{A}\phi(\overline{\omega}_1)\frac{\partial\overline{\omega}_1}{\partial b}\right];\\
\frac{\partial}{\partial A}F_c(t) &= \frac{1}{A}\left[-\Phi(\overline{\omega}_1-\overline{\omega}_2)+\phi(\overline{\omega}_1-\overline{\omega}_2)(b-A-(t-\tau)v^c)\frac{\partial\overline{\omega}_2}{\partial A}\right]\\
&+\frac{1}{A^2}\left[(b-(t-\tau)v^c)\Phi(\overline{\omega}_1)-(b-A-(t-\tau)v^c)\Phi(\overline{\omega}_1-\overline{\omega}_2)\right]\\
&+\frac{\phi'(\overline{\omega}_1-\overline{\omega}_2)\frac{\partial\overline{\omega}_2}{\partial A}\overline{\omega}_2-\frac{\partial\overline{\omega}_2}{\partial A}\phi(\overline{\omega}_1-\overline{\omega}_2)}{\overline{\omega}_2^2}+\phi(\overline{\omega}_1)\frac{\partial\overline{\omega}_2}{\partial A};\\
\frac{\partial}{\partial v^c}F_c(t) &= -\frac{t-\tau}{A}\Phi(\overline{\omega}_1-\overline{\omega}_2)+\frac{b-A-(t-\tau)v^c}{A}\phi(\overline{\omega}_1-\overline{\omega}_2)\frac{\partial\overline{\omega}_1}{\partial v^c}\\
&+\frac{t-\tau}{A}\Phi(\overline{\omega}_1)-\phi(\overline{\omega}_1)\frac{b-(t-\tau)v^c}{A}\frac{\partial\overline{\omega}_1}{\partial v^c}\\
&+\frac{1}{\overline{\omega}_2}\left(\phi'(\overline{\omega}_1-\overline{\omega}_2)-\phi'(\overline{\omega}_1)\right)\frac{\partial\overline{\omega}_1}{\partial v^c};\\
\frac{\partial}{\partial\tau}F_c(t) &= \frac{v^c}{\tau}\Phi(\overline{\omega}_1-\overline{\omega}_2)+\frac{b-A-(t-\tau)v^c}{A}\phi(\overline{\omega}_1-\overline{\omega}_2)\left(\frac{\partial\overline{\omega}_1}{\partial\tau}-\frac{\partial\overline{\omega}_2}{\partial\tau}\right)\\
&-\frac{v^c}{\tau}\Phi(\overline{\omega}_1)-\frac{b-(t-\tau)v^c}{A}\phi(\overline{\omega}_1)\frac{\partial\overline{\omega}_1}{\partial\tau}\\
&+\left[\phi'(\overline{\omega}_1-\overline{\omega}_2)\overline{\omega}_2\left(\frac{\partial\overline{\omega}_1}{\partial\tau}-\frac{\partial\overline{\omega}_2}{\partial\tau}\right)-\phi(\overline{\omega}_1-\overline{\omega}_2)\frac{\partial\overline{\omega}_2}{\partial\tau}\right]/(\overline{\omega}_2^2)\\
&-\left[\phi'(\overline{\omega}_1)\overline{\omega}_2\frac{\partial\overline{\omega}_1}{\partial\tau}-\phi(\overline{\omega}_1)\frac{\partial\overline{\omega}_2}{\partial\tau}\right]/(\overline{\omega}_2^2).
\end{aligned}
$$

S2.3.2.2 The DDM density

Deriving gradients of the diffusion density (large-time representation) From Section S1.2, the density of the diffusion model (large-time representation) is

$$
\begin{aligned}
\text{DDM}^l(\text{“lower”},\text{rt}) = \frac{1}{s_z s_\tau}\int_{z-\frac{s_z}{2}}^{z+\frac{s_z}{2}}\int_{\tau-\frac{s_\tau}{2}}^{\tau+\frac{s_\tau}{2}}&\frac{\pi}{a^2\sqrt{1+t_d s_v^2}}\exp\left(-\frac{v^2t_d+2vz-z^2s_v^2}{2(t_ds_v^2+1)}\right)\\
&\times\sum_{k=1}^{\infty}k\sin\left(\frac{\pi kz}{a}\right)\exp\left(-\frac{\pi^2k^2t_d}{2a^2}\right)d\tau dz.
\end{aligned}
$$

Define the integrand as

$$
g^l(t_d|\boldsymbol{\theta}) := \frac{\pi}{a^2\sqrt{1+t_ds_v^2}}\exp\left(-\frac{v^2t_d+2vz-z^2s_v^2}{2(t_ds_v^2+1)}\right)\sum_{k=1}^{\infty}k\sin\left(\frac{\pi kz}{a}\right)\exp\left(-\frac{\pi^2k^2t_d}{2a^2}\right).
$$

We now derive the partial derivatives of the integrand with respect to v, s_v^2, a, z, and τ. The partial derivatives with respect to z, s_z, τ, and s_τ, as well as the transformed parameters can then be derived straightforwardly using the chain rule.

$$\frac{\partial}{\partial v} g^l(t_d|\boldsymbol{\theta}) = -g^l(t_d|\boldsymbol{\theta}) \left[\frac{t_d v + z}{t_d s_v^2 + 1} \right],$$
$$\frac{\partial}{\partial s_v^2} g^l(t_d|\boldsymbol{\theta}) = \frac{1}{2} g^l(t_d|\boldsymbol{\theta}) \left[\frac{1}{2} \frac{(z + v t_d)^2 - t_d(1 + t_d s_v^2)}{(1 + t_d s_v^2)^2} \right],$$

$$\frac{\partial}{\partial a} g^l(t_d|\boldsymbol{\theta}) = -g^l(t_d|\boldsymbol{\theta}) \frac{2}{a} + \frac{\pi}{a^2 \sqrt{t_d s_v^2 + 1}} \exp\left(-\frac{v^2 t_d + 2vz - z^2 s_v^2}{2(t_d s_v 2 + 1)} \right)$$
$$\times \sum_{k=1}^{\infty} k \sin\left(\frac{\pi k z}{a} \right) \exp\left(-\frac{\pi^2 k^2 t_d}{2a^2} \right) \left(-\frac{\pi k z}{a^2} \cot\left(\frac{\pi k z}{a} + \frac{\pi^2 k^2 t_d}{a^3} \right) \right),$$

$$\frac{\partial}{\partial z} g^l(t_d|\boldsymbol{\theta}) = g^l(t_d|\boldsymbol{\theta}) \left[\frac{z s_v^2 - v}{t_d s_v^2 + 1} \right] + \frac{\pi}{a^2 \sqrt{t_d s_v^2 + 1}} \exp\left(-\frac{v^2 t_d + 2vz - z^2 s_v^2}{2(t_d s_v^2 + 1)} \right)$$
$$\times \sum_{k=1}^{\infty} k \sin\left(\frac{\pi k z}{a} \right) \exp\left(-\frac{\pi^2 k^2 t_d}{2a^2} \right) \frac{\pi k}{a} \cot\left(\frac{\pi k z}{a} \right),$$

$$\frac{\partial}{\partial \tau} g^l(t_d|\boldsymbol{\theta}) = g^l(t_d|\boldsymbol{\theta}) \left[\frac{v^2(t_d s_v^2 + 1) - s_v^2 v^2 t_d}{2(t_d s_v^2 + 1)^2} + \frac{s_v^2}{2(t_d s_v^2 + 1)} - \frac{2vz - z^2 s_v^2}{2(t_d s_v^2 + 1)^2} s_v^2 \right]$$
$$+ \frac{\pi^3}{2a^4 \sqrt{t_d s_v^2 + 1}} \exp\left(-\frac{v^2 t_d + 2vz - z^2 s_v^2}{2(t_d s_v^2 + 1)} \right) \sum_{k=1}^{\infty} k^3 \sin\left(\frac{\pi k z}{a} \right) \exp\left(-\frac{\pi^2 k^2 t_d}{a^2} \right).$$

Deriving gradients of the diffusion density (small-time representation) From Section S1.2, the density of the diffusion model (small-time representation) is

$$\text{DDM}^s(\text{“lower”}, \text{rt}) = \frac{1}{s_z s_\tau} \int_{z - \frac{s_z}{2}}^{z + \frac{s_z}{2}} \int_{\tau - \frac{s_\tau}{2}}^{\tau + \frac{s_\tau}{2}} \frac{a t_d^{-3/2}}{\sqrt{2\pi(1 + t_d s_v^2)}} \exp\left(-\frac{v^2}{2 s_v^2} + \frac{(v - z s_v^2)^2}{s_v^2 (1 + t_d s_v^2)} \right)$$
$$\times \sum_{k=-\infty}^{\infty} \left(\frac{z}{a} + 2k \right) \exp\left(-\frac{(z + 2ka)^2}{2 t_d} \right) d\tau dz.$$

Define the integrand as

$$g^s(t_d|\boldsymbol{\theta}) = \frac{a t_d^{-3/2}}{\sqrt{2\pi(1 + t_d s_v^2)}} \exp\left(-\frac{v^2}{2 s_v^2} + \frac{(v - z s_v^2)^2}{s_v^2 (1 + t_d s_v^2)} \right) \times \sum_{k=-\infty}^{\infty} \left(\frac{z}{a} + 2k \right) \exp\left(-\frac{(z + 2ka)^2}{2 t_d} \right).$$

We now derive the partial derivatives of the integrand with respect to v, s_v^2, a, z, and τ. The partial derivatives with respect to z, s_z, τ, and s_τ, as well as the transformed parameters can then be derived straightforwardly using the chain rule.

$$\frac{\partial}{\partial s_v^2} g^s(t_d|\boldsymbol{\theta}) = g^s(t_d|\boldsymbol{\theta}) \left[-\frac{t_d}{2(1+t_d s_v^2)} + \frac{v^2}{2s_v^4} - \frac{z(v - zs_v^2)}{(1+t_d s_v^2)s_v^2} \right] - g^s(t_d|\boldsymbol{\theta}) \frac{(1+2t_d s_v^2)(v-zs_v^2)^2}{2(1+t_d s_v^2)^2 s_v^4},$$

$$\frac{\partial}{\partial s_v^2} g^s(t_d|\boldsymbol{\theta}) = g^s(t_d|\boldsymbol{\theta}) \left[-\frac{t_d}{2(1+t_d s_v^2)} + \frac{v^2}{2s_v^4} \right] - g^s(t_d|\boldsymbol{\theta}) \frac{2z(v-zs_v^2)(1+t_d s_v^2)s_v^2 + (1+2t_d s_v^2)(v-zs_v^2)^2}{2(1+t_d s_v^2)^2 s_v^4},$$

$$\frac{\partial}{\partial a} g^s(t_d|\boldsymbol{\theta}) = g^s(t_d|\boldsymbol{\theta}) \frac{1}{a} + \frac{a t_d^{-3/2}}{\sqrt{2\pi(1+t_d s_v^2)}} \exp\left(-\frac{v^2}{2s_v^2} + \frac{(v-zs_v^2)^2}{s_v^2(1+t_d s_v^2)} \right) \times \sum_{k=-\infty}^{\infty} \exp\left(-\frac{(z+2ka)^2}{2t_d} \right) \left(-\frac{zt_d + 2ka(z+2ka)^2}{a^2 t_d} \right),$$

$$\frac{\partial}{\partial z} g^s(t_d|\boldsymbol{\theta}) = g^s(t_d|\boldsymbol{\theta}) \left[\frac{zs_v^2 - v}{t_d s_v^2 + 1} \right] + \frac{a t_d^{-3/2}}{\sqrt{2\pi(1+t_d s_v^2)}} \exp\left(-\frac{v^2}{2s_v^2} + \frac{(v-zs_v^2)^2}{s_v^2(1+t_d s_v^2)} \right) \times \sum_{k=-\infty}^{\infty} \exp\left(-\frac{(z+2ka)^2}{2t_d} \right) \left(\frac{t_d - (z+2ka)^2}{a t_d} \right),$$

$$\frac{\partial}{\partial \tau} g^s(t_d|\boldsymbol{\theta}) = g^s(t_d|\boldsymbol{\theta}) \left[\frac{s_v^2(t_d s_v^2 + 1) + (v - zs_v^2)^2}{2(t_d s_v^2 + 1)^2} + \frac{3}{2t_d} \right] - \frac{a t_d^{-3/2}}{\sqrt{2\pi(1+t_d s_v^2)}} \exp\left(-\frac{v^2}{2s_v^2} + \frac{(v-zs_v^2)^2}{s_v^2(1+t_d s_v^2)} \right) \times \sum_{k=-\infty}^{\infty} \left(\frac{z}{a} + 2k \right) \exp\left(-\frac{(z+2ka)^2}{2t_d} \right) \frac{(z+2ka)^2}{2t_d^2}.$$

S2.3.3 Deriving $\nabla_{\boldsymbol{\theta}_1} \log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$

From above, that $\boldsymbol{\theta}_1 = (\boldsymbol{\alpha}_{1:J}, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \log \boldsymbol{a})$ and $\boldsymbol{\theta}_2 = \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ and

$$p(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = N(\boldsymbol{\beta}|\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \prod_{j=1}^{J} N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) N(\boldsymbol{\mu}_{\boldsymbol{\alpha}}|\boldsymbol{\mu}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}}) \mathrm{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}) \prod_{d=1}^{D_\alpha} \mathrm{IG}(a_d|\alpha_d, \beta_d) \left| J_{a_d \to \log a_d} \right|;$$

hence,

$$\begin{aligned}
\log p(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) &= \log N(\boldsymbol{\beta}|\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) + \sum_{j=1}^{J} \log N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) + \log N(\boldsymbol{\mu}_{\boldsymbol{\alpha}}|\boldsymbol{\mu}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}}) + \log \mathrm{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}) \\
&+ \sum_{d=1}^{D_\alpha} \Big(\log \mathrm{IG}(a_d|\alpha_d, \beta_d) + \log a_d \Big) \\
&= -\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}})^\top \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}) - \frac{1}{2}\sum_{j=1}^{J}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})^\top \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}}) \\
&- \frac{1}{2}(\boldsymbol{\mu}_{\boldsymbol{\alpha}} - \boldsymbol{\mu}_{\boldsymbol{\mu}})^\top \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{-1}(\boldsymbol{\mu}_{\boldsymbol{\alpha}} - \boldsymbol{\mu}_{\boldsymbol{\mu}}) + \frac{\nu}{2}\log\det(\boldsymbol{\Psi}) - \frac{1}{2}\operatorname{trace}(\boldsymbol{\Psi}\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}) \\
&+ \sum_{d=1}^{D_\alpha} \left\{ -(\alpha_d + 1)\log a_d - \frac{\beta_d}{a_d} + \log a_d \right\} + \text{constant.}
\end{aligned}$$

Hence,

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\alpha}_j} \log p(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = -\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}}), \\
&\frac{\partial}{\partial \boldsymbol{\beta}_{\text{vec}}} \log p(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = \frac{\partial}{\partial \boldsymbol{\beta}_{\text{vec}}} \log N(\boldsymbol{\beta}_{\text{vec}}|\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = -\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\left(\boldsymbol{\beta}_{\text{vec}} - \boldsymbol{\mu}_{\boldsymbol{\beta}}\right), \\
&\frac{\partial}{\partial \boldsymbol{\mu}_{\boldsymbol{\alpha}}} \log p(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = \sum_{j=1}^{J} \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}}) - \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \\
&\frac{\partial}{\partial \log a_d} \log p(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = -\frac{\nu}{2} - \alpha_d + \frac{\beta_d}{a_d} + \nu_\alpha \frac{(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}})_{dd}}{a_d}, \text{ for } d = 1, \ldots, D_\alpha. \\
&\frac{\partial}{\partial \boldsymbol{\beta}_{\text{vec}}} \log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y}) = \mathbf{0}.
\end{aligned}$$

S2.3.4 Deriving $\nabla_{\boldsymbol{\theta}_1} \log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y})$

Obviously $\nabla_{\boldsymbol{\theta}_1} \log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y}) \equiv \nabla_{\boldsymbol{\theta}_1} \log p(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\boldsymbol{\theta}_1, \boldsymbol{y}) \equiv \nabla_{\boldsymbol{\theta}_1} \log \mathrm{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}')$. We now have

$$\begin{aligned}
&\log \mathrm{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}') = \frac{\nu}{2}\log\det(\boldsymbol{\Psi}') - \frac{1}{2}\operatorname{trace}(\boldsymbol{\Psi}'\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}) + \text{constant.} \\
&\propto \frac{\nu}{2}\log\det\left(\sum_{j=1}^{J}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})^\top + 4\,\mathrm{diag}\left(a_1^{-1}, \ldots, a_{D_\alpha}^{-1}\right)\right) \\
&- \frac{1}{2}\operatorname{trace}\left(\left(\sum_{j=1}^{J}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})^\top + 4\,\mathrm{diag}\left(a_1^{-1}, \ldots, a_{D_\alpha}^{-1}\right)\right)\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}\right).
\end{aligned}$$

Hence,

$$\frac{\partial}{\partial \boldsymbol{\alpha}_j} \log \text{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}') = \left(\nu \left(\boldsymbol{\Psi}'\right)^{-1} - \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}\right)(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}}), \text{ for } j = 1, \ldots, J,$$
$$\frac{\partial}{\partial \boldsymbol{\beta}_{\text{vec}}} \log p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y}) = \mathbf{0},$$
$$\frac{\partial}{\partial \boldsymbol{\mu}_{\boldsymbol{\alpha}}} \log \text{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}') = \left(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}} - \nu \left(\boldsymbol{\Psi}'\right)^{-1}\right) \sum_{j=1}^{J} (\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}}),$$
$$\frac{\partial}{\partial \log a_d} \log \text{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}') = -\frac{v_\alpha}{a_d}\left(\nu \left(\left(\boldsymbol{\Psi}'\right)^{-1}\right)_{dd} + \left(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}\right)_{dd}\right), \text{ for } d = 1, \ldots, D_\alpha.$$

S2.4 VB for DDM

We now consider VB for the DDM. Dao et al. (2022) consider LBA models having the same hierarchical structure and show that the posterior distribution can be factored as

$$p(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}, \boldsymbol{a}|\boldsymbol{y}) = p(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{a}|\boldsymbol{y}) p(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\boldsymbol{\alpha}_{1:J}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{a}, \boldsymbol{y}),$$

where $p(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\boldsymbol{\alpha}_{1:J}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{a}, \boldsymbol{y})$ is the density of $\text{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}')$ with $\nu = 2D_\alpha + J + 1$ and $\boldsymbol{\Psi}' = \sum_{j=1}^{J}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_{\boldsymbol{\alpha}})^\top + 4\,\text{diag}\left(a_1^{-1}, \ldots, a_{D_\alpha}^{-1}\right)$. Denote $\boldsymbol{\theta}_1 = (\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \log \boldsymbol{a})$. To exploit the structure of the posterior distribution, Dao et al. (2022) propose the Hybrid Gaussian VB of the form

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = N(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1|\boldsymbol{\mu}_{\boldsymbol{\lambda}}, B_{\boldsymbol{\lambda}} B_{\boldsymbol{\lambda}}^\top + D_{\boldsymbol{\lambda}}^2)\text{IW}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}|\nu, \boldsymbol{\Psi}').$$

The LBA likelihood is equal to 0 if the observed response time RT is less than the non-decision time τ. Hence, the support[7] of the posterior density should not cover any regions where the likelihood is 0. However, the proposed Hybrid Gaussian VB does not capture this important feature. Instead, due to the tractability and simplicity of LBA density, Dao et al. handles the issue by numerically truncating the likelihood (set extreme values equal to some predetermined threshold). Compared to LBA models, DDM is more mathematically intractable, hence a better approach is needed. We improve the variational density to tackle the irregularity issue of the diffusion density as follows. Given the data for subject j, the lower limit of the non-decision time $L_{\tau_j} := \tau_j - \dfrac{s_{\tau_j}}{2}$ must be smaller than any observed response time, i.e.,

$$L_{\tau_j} := \tau_j - \frac{s_{\tau_j}}{2} < \min \text{RT}_j := \min_{i \in \{1, \ldots, n_j\}} \text{RT}_{ij}, \text{ for all } j = 1, \ldots, J. \tag{S10}$$

[7]The support of a density is understood as a region where the density is positive. The rigorous definition is omitted as it requires more advanced mathematical concepts which are beyond the scope of this paper.

Because these extra constraints in equation (S10) only arise after the data becomes available, they are not captured in the transformed random effects $\boldsymbol{\alpha}_{1:J}$ (see section S1.3 of the online supplement), meaning that the choice

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1) = N(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1 | \boldsymbol{\mu}_{\boldsymbol{\lambda}}, B_{\boldsymbol{\lambda}} B_{\boldsymbol{\lambda}}^\top + D_{\boldsymbol{\lambda}}^2)$$

is now unsuitable. To derive a better parametric form for $q_{\boldsymbol{\lambda}}(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1)$, we need to introduce some new transformations on the non-decision time parameters. Instead of using $\alpha_6 = \log(\tau_j - 0.5 s_\tau)$, we replace it with the new transformation

$$\tilde{\alpha}_6 = \log\left(\frac{\min \mathrm{RT}_j - (\tau_j - 0.5 s_{\tau_j})}{(\tau_j - 0.5 s_{\tau_j})}\right).$$

Notice that we only need to change the transformation on the non-decision time parameters (α_6), with all the other transformations unchanged. The new transformed random effects, denoted by $\tilde{\boldsymbol{\alpha}}_{1:J}$, are now unrestricted, so a multivariate Gaussian approximation can be used.

$$q_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\alpha}}_{1:J}, \boldsymbol{\theta}_1) = N(\tilde{\boldsymbol{\alpha}}_{1:J}, \boldsymbol{\theta}_1 | \boldsymbol{\mu}, BB^\top + D^2).$$

A better choice for $q_{\boldsymbol{\lambda}}(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1)$ is the one induced from $q_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\alpha}}_{1:J}, \boldsymbol{\theta}_1)$, i.e.

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1) = N\left(T_{\boldsymbol{y}}(\boldsymbol{\alpha}_{1:J}), \boldsymbol{\theta}_1 | \boldsymbol{\mu}, BB^\top + D^2\right) \left| J_{\tilde{\boldsymbol{\theta}}_{\boldsymbol{y}} \to \boldsymbol{\theta}} \right|,$$

where

$$\left| J_{\tilde{\boldsymbol{\theta}}_{\boldsymbol{y}} \to \boldsymbol{\theta}} \right| = \prod_{j=1}^{J} \frac{\partial \tilde{\alpha}_{j;6}}{\partial \alpha_{j;6}} = \prod_{j=1}^{J} \frac{\min \mathrm{RT}_j}{e^{\alpha_{j;6}} - \min \mathrm{RT}_j},$$

is the Jacobian of the function transforming $\tilde{\boldsymbol{\theta}}_{\boldsymbol{y}} = (\tilde{\boldsymbol{\alpha}}_{1:J}, \boldsymbol{\theta}_1)$ to $\boldsymbol{\theta} = (\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_1)$. To see this, note that

$$\tilde{\alpha}_{j;6} = \log\left(\frac{\min \mathrm{RT}_j - (\tau_j - 0.5 s_{\tau_j})}{(\tau_j - 0.5 s_{\tau_j})}\right) = \log\left(\frac{\min \mathrm{RT}_j - e^{\alpha_{j;6}}}{e^{\alpha_{j;6}}}\right), \text{ for } j = 1, \ldots, J.$$

We obtain

$$\frac{\partial \tilde{\alpha}_{j;6}}{\partial \alpha_{j;6}} = \frac{\min \mathrm{RT}_j}{e^{\alpha_{j;6}} - \min \mathrm{RT}_j}, \text{ for } j = 1, \ldots, J.$$

S3 Particle Metropolis within Gibbs (PMwG)

This section gives further details of the PMwG algorithm. Let $\boldsymbol{\theta}$ be all the model parameters, except the random effects, i.e., $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}, \boldsymbol{a})$. For convenience, we denote by $\boldsymbol{\theta}_{-\boldsymbol{\beta}}$ all the model parameters except $\boldsymbol{\beta}$, i.e., $\boldsymbol{\theta}_{-\boldsymbol{\beta}} = (\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}, \boldsymbol{a})$. We similarly define $\boldsymbol{\theta}_{-\boldsymbol{\mu}_{\boldsymbol{\alpha}}}, \boldsymbol{\theta}_{-\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}}$, and $\boldsymbol{\theta}_{-\boldsymbol{a}}$. With the model structure and the choice of prior distribution introduced in Section 2.3, the full conditional densities of $\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$, and $\boldsymbol{a}$ (hyperparameters) are known, hence sampling from these distributions is straightforward. However, the full conditional densities $p(\boldsymbol{\alpha}_j|\boldsymbol{\theta}, \boldsymbol{y}_j), j = 1, \ldots, J$, and $p(\boldsymbol{\beta}|\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_{-\boldsymbol{\beta}}, \boldsymbol{y})$ are unavailable in closed form, so sampling directly from them is difficult.

Algorithm S2 is the PMwG sampler for estimating the EAMs with covariates. The algorithm starts by initialising the values for $\boldsymbol{\theta}, \boldsymbol{\beta}$, and random effects $\boldsymbol{\alpha}_{1:J}$. The sampler then runs for a predetermined number of iterations T_{iter}. In each iteration, we sequentially sample the group-level parameters $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ (Step 2a), $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ (Step 2b) and the hyperparameters $\boldsymbol{a}$ (Step 2c) using Gibbs step conditional on the selected particles $\boldsymbol{\alpha}_{1:J}^{\boldsymbol{k}}$ from the previous iteration. In Step 2d, $R - 1$ new particles are generated using the conditional MC method given in algorithm S3 in section S3. Note that we keep the particles $\boldsymbol{\alpha}_{1:J}^{\boldsymbol{k}}$ fixed and set the first set of particles $\boldsymbol{\alpha}_{1:J}^{1} = \boldsymbol{\alpha}_{1:J}^{\boldsymbol{k}}$. The conditional MC algorithm gives the particles $\boldsymbol{\alpha}_{1:J}^{1:R}$ and the normalised weights $W_j^{(r)}$, for $j = 1, \ldots, J$ and $r = 1, \ldots, R$. Step 2e samples the new index vector $\boldsymbol{k} = (k_1, \ldots, k_J)$ and the random effects are selected according to the index vector $\boldsymbol{k}$, i.e., we discard the rest of the particles $\boldsymbol{\alpha}_{1:J}^{(-\boldsymbol{k})}$ and only keep $\boldsymbol{\alpha}_{1:J}^{\boldsymbol{k}}$. Similar steps (2f and 2g) are applied to sample $\boldsymbol{\beta}$. The PMwG sampler in Algorithm S2 is for the hierarchical EAM with covariates. When there are no covariates, steps (2f) and (2g) can be omitted.

S3.1 Conditional Monte Carlo Algorithm

This section presents the conditional Monte Carlo (CMC) algorithm used in the PMwG algorithm. The CMC is required to sample the random effects $\boldsymbol{\alpha}_{1:J}$ (Algorithm S3) and the parameter $\boldsymbol{\beta}$ (Algorithm S4).

S3.2 Proposal distributions and tuning parameters

The efficiency of the PMwG sampler depends on a careful choice of the proposal densities $m_j(\boldsymbol{\alpha}_j|\boldsymbol{y}_j, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}), j = 1, \ldots, J$ for the random effects and $m(\boldsymbol{\beta}_{vec}|\boldsymbol{y}_j, \boldsymbol{\alpha}_j), j = 1, \ldots, J$ for the parameters $\boldsymbol{\beta}$. This section discusses how we select and tune these distributions.

Algorithm S2 Particle Metropolis within Gibbs (PMwG) for the hierarchical EAM with covariates

1. Initialisation: select initial values for $\boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}$ and set $\boldsymbol{\alpha}_j^1 = \boldsymbol{\alpha}_j^{k_j}, \boldsymbol{\beta}^1 = \boldsymbol{\beta}^{k_{\boldsymbol{\beta}}}$.
2. For $t = 1, \ldots, T_{iter}$:
   (a) Sample $\boldsymbol{\mu}_{\boldsymbol{\alpha}} | \boldsymbol{k}, \boldsymbol{\alpha}_{1:J}^{\boldsymbol{k}}, \boldsymbol{\theta}_{-\boldsymbol{\mu}_{\boldsymbol{\alpha}}}, \boldsymbol{y}$ from $N(\overline{\boldsymbol{\mu}}, \overline{\boldsymbol{\Sigma}})$, where

$$\begin{aligned} \overline{\boldsymbol{\mu}} &= \overline{\boldsymbol{\Sigma}} \left(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}} \sum_{j=1}^{J} \boldsymbol{\alpha}_j^{k_j} \right), \\ \overline{\boldsymbol{\Sigma}} &= \left(J \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}} + I \right)^{-1}. \end{aligned}$$

   (b) Sample $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} | \boldsymbol{k}, \boldsymbol{\alpha}_{1:J}^{\boldsymbol{k}}, \boldsymbol{\theta}_{-\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}}$ from IW(k_α, B_α), where

$$\begin{aligned} k_\alpha &= v_\alpha + D_\alpha - 1 + J, \\ B_\alpha &= 2 v_\alpha \operatorname{diag}\left(a_1^{-1}, \ldots, a_1^{-1}\right) + \sum_{j=1}^{J} (\boldsymbol{\alpha}_j^{k_j} - \boldsymbol{\mu}_{\boldsymbol{\alpha}})(\boldsymbol{\alpha}_j^{k_j} - \boldsymbol{\mu}_{\boldsymbol{\alpha}})^\top. \end{aligned}$$

   (c) Sample $a_d | \boldsymbol{k}, \boldsymbol{\alpha}_{1:J}^{\boldsymbol{k}}, \boldsymbol{\theta}_{-a_d}$ from IG$(\overline{\alpha}, \overline{\beta})$, where

$$\begin{aligned} \overline{\alpha} &= \frac{v_\alpha + D_\alpha}{2}, \\ \overline{\beta} &= v_\alpha \left(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-\mathbf{1}}\right)_{dd} + \frac{1}{\mathcal{A}_d^2}, \text{ for } d = 1, \ldots, D_\alpha. \end{aligned}$$

   (d) Sample $\boldsymbol{\alpha}_{1:J}^{-\boldsymbol{k}}$ using Algorithm S3 in section S3 of the online supplement
   (e) Sample the index vector $\boldsymbol{k} = (k_1, \ldots, k_J)$ with probability given by

$$P(k_1 = l_1, \ldots, k_J = l_J | \boldsymbol{\alpha}_{1:J}, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = \prod_{j=1}^{J} W_j^{l_j}.$$

   (f) Sample $\boldsymbol{\beta}^{-k_{\boldsymbol{\beta}}}$ using Algorithm S4 in section S3 of the online supplement
   (g) Sample the index $k_{\boldsymbol{\beta}}$ with probability given by

$$P(k_{\boldsymbol{\beta}} = l | \boldsymbol{\alpha}_{1:J}, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = W_\beta^l.$$

Algorithm S3 Conditional Monte Carlo for sampling $\boldsymbol{\alpha}_{1:J}$

Require: proposal densities $m_j(\boldsymbol{\alpha}_j|\boldsymbol{y}_j, \boldsymbol{X}_j, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}), j = 1, \ldots, J$.

1. Fix $\boldsymbol{\alpha}_{1:J}^{(1)} = \boldsymbol{\alpha}_{1:J}^{(\boldsymbol{k})}$.

2. For $j = 1, \ldots, J$

   - Sample $R-1$ particles $\boldsymbol{\alpha}_j^{(r)}, r = 2, \ldots, R$ from the proposal $m_j(\boldsymbol{\alpha}_j|\boldsymbol{y}_j, \boldsymbol{X}_j, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$.
   - Compute the weights

$$\omega_j^{(r)} = \frac{p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j^{(r)}, \boldsymbol{\beta}, \boldsymbol{X}_j)p(\boldsymbol{\alpha}_j^{(r)}|\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})}{m_j(\boldsymbol{\alpha}_j^{(r)}|\boldsymbol{y}_j, \boldsymbol{X}_j, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})}, \text{ for } r = 1, \ldots, R.$$

   - Normalize the weights

$$W_j^{(r)} = \frac{\omega_j^{(r)}}{\sum_{r=1}^{R} \omega_j^{(r)}}, \text{ for } r = 1, \ldots, R.$$

Algorithm S4 Conditional Monte Carlo for sampling $\boldsymbol{\beta}$

Require: proposal density $m(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{\alpha}_{1:J}, \boldsymbol{X})$.

1. Fix $\boldsymbol{\beta}^{(1)} = \boldsymbol{\beta}^{k_{\boldsymbol{\beta}}}$.

2. Sample $R-1$ particles $\boldsymbol{\beta}^{(r)}, r = 2, \ldots, R$ from the proposal $m(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{\alpha}_{1:J}, \boldsymbol{X})$.

3. Compute the weights

$$\omega_\beta^{(r)} = \frac{\prod_{j=1}^{J} p(\boldsymbol{y}_j|\boldsymbol{\alpha}_j, \boldsymbol{\beta}^{(r)}, \boldsymbol{X}_j)p(\boldsymbol{\beta}^{(r)}|\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})}{m(\boldsymbol{\beta}^{(r)}|\boldsymbol{y}, \boldsymbol{\alpha}_{1:J}, \boldsymbol{X})}, \text{ for } r = 1, \ldots, R.$$

4. Normalise the weights

$$W_\beta^{(r)} = \frac{\omega_\beta^{(r)}}{\sum_{r=1}^{R} \omega_\beta^{(r)}}, \text{ for } r = 1, \ldots, R.$$

As recommended by Gunawan et al., the proposal densities are adaptively improved over time. At the first stage, the proposal density for subject j at iteration t is chosen to be a mixture of the prior $N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}}{}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}{}^{(t)})$ and a normal distribution centred on the previous sample for the random effect, $N(\boldsymbol{\alpha}_j^{(t-1)}, \epsilon\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(t)})$, i.e.,

$$m_j(\boldsymbol{\alpha}_j|\boldsymbol{y}_j, \boldsymbol{X}_j, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = \omega N(\boldsymbol{\alpha}_j|\boldsymbol{\alpha}_j^{(t-1)}, \epsilon\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(t)}) + (1-\omega)N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(t)}).$$

The scale parameter ϵ ($0 < \epsilon < 1$) is used to control the "step-size" and is determined by trial-and-error. The scale parameter should be small when the number of random effects is large. We set $\epsilon = 0.5$. Wall et al. (2021) set $\epsilon = 0.1$ for estimating a hierarchical LBA with 110 subjects, each having 30 random effects. Similarly, the proposal density for sampling $\boldsymbol{\beta}$ is the mixture of a Gaussian distribution centred at the previous draw $\boldsymbol{\beta}^{(t-1)}$ with the block diagonal covariance matrix $\boldsymbol{\Sigma}_0$ having D_α blocks with the matrix $(\boldsymbol{X}^\top\boldsymbol{X})^{-1}$ on each diagonal block and the prior

$$m(\boldsymbol{\beta}_{\text{vec}}|\boldsymbol{y}, \boldsymbol{\alpha}_{1:J}, \boldsymbol{X}) = \omega N(\boldsymbol{\beta}_{\text{vec}}|\boldsymbol{\beta}_{\text{vec}}^{(t-1)}, \epsilon\boldsymbol{\Sigma}_0) + (1-\omega)N(\boldsymbol{\beta}_{\text{vec}}|\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}).$$

After the burn-in and initial adaptation stages, at iteration $t(t > T_{burn} + T_{adapt})$, more efficient proposal densities for subject j are obtained by first approximating the joint posterior distribution $p(\boldsymbol{\alpha}_j, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}, \boldsymbol{\beta}|\boldsymbol{y})$ by fitting a multivariate Gaussian distribution using the draws from the adaptation stage $\left\{\boldsymbol{\alpha}_j^{(i)}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}{}^{(i)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}{}^{(i)}, \boldsymbol{\beta}^{(i)}\right\}_{i=T_{\text{burn}}+1}^{t-1}$. From this multivariate Gaussian, we obtain the conditional distribution of $\boldsymbol{\alpha}_j$ given $\boldsymbol{\mu}_{\boldsymbol{\alpha}} = \boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} = \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\beta} = \boldsymbol{\beta}^{(t)}$ and $\boldsymbol{y}$, which we denote by $N(\widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{\Sigma}}_j)$. The process of updating better proposal distributions can be done at each iteration. However, for computational efficiency, we update new proposals after every 20 iterations and stop updating when the proposal distributions stablise (after 5,000 iterations).

The proposal density for subject j is taken as a' mixture of the conditional distribution $N(\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)$, the Gaussian distribution centred at the previous draws with the covariance matrix $\hat{\boldsymbol{\Sigma}}_j$ and the prior group-level distribution for the random effects, $N(\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(t)})$,

$$m_j(\boldsymbol{\alpha}_j|\boldsymbol{y}_j, \boldsymbol{X}_j, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = \omega_1 N(\boldsymbol{\alpha}_j|\hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j) + \omega_2 N(\boldsymbol{\alpha}_j|\boldsymbol{\alpha}_j^{(t)}, \hat{\Sigma}_j) + \omega_3 N(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(t)}). \tag{S11}$$

We set $\omega_1 = 0.65$, $\omega_2 = 0.3$ and $\omega_3 = 0.05$. Following Hesterberg (1995), including the prior density $p(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$ in Equation (S11) with small weight $\omega_3 = 0.05$ ensures that the importance weights are bounded. We put a larger weight $\omega_1 = 0.65$ on the most efficient

proposal $N(\boldsymbol{\alpha}_j|\hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)$, and a smaller weight $\omega_2 = 0.3$ for the random walk type proposal.

More efficient proposal distributions for $\boldsymbol{\beta}$ are obtained similarly. Since the full conditional density $p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{\alpha}_{1:J}, \boldsymbol{\theta}_{-\boldsymbol{\beta}}) = p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{\alpha}_{1:J})$, we use $\left\{\boldsymbol{\alpha}_{1:J}^{(i)}, \boldsymbol{\beta}^{(i)}\right\}_{i=T_{\text{burn}}+1}^{t-1}$ to fit a multivariate Gaussian. From the fitted Gaussian, obtain the conditional distribution of $\boldsymbol{\beta}$ given $\boldsymbol{\alpha}_{1:J} = \boldsymbol{\alpha}_{1:J}^{(t)}$ and $\boldsymbol{y}$, which we denote by $N(\widehat{\boldsymbol{\mu}}_\beta, \widehat{\boldsymbol{\Sigma}}_\beta)$. The better proposal density for $\boldsymbol{\beta}$ is now taken to be the mixture of the conditional distribution $N(\widehat{\boldsymbol{\mu}}_\beta, \widehat{\boldsymbol{\Sigma}}_\beta)$, the Gaussian distribution centred at the previous draws with the covariance matrix $\widehat{\boldsymbol{\Sigma}}_\beta$ and the prior,

$$m(\boldsymbol{\beta}_{\text{vec}}|\boldsymbol{y}, \boldsymbol{\alpha}_{1:J}, \boldsymbol{X}) = \omega_1 N(\boldsymbol{\beta}|\widehat{\boldsymbol{\mu}}_\beta, \widehat{\boldsymbol{\Sigma}}_\beta) + \omega_2 N(\boldsymbol{\beta}_{\text{vec}}|\boldsymbol{\beta}_{\text{vec}}^{(t-1)}, \widehat{\boldsymbol{\Sigma}}_\beta) + \omega_3 N(\boldsymbol{\beta}_{\text{vec}}|\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}).$$

Again, we set $\omega_1 = 0.65$, $\omega_2 = 0.3$ and $\omega_3 = 0.05$.

S4 Integrated Autocorrelation Time (IACT)

The *integrated autocorrelated time* (IACT) for a scalar parameter θ is defined as (Liu and Liu, 2001)

$$\text{IACT}_\theta := 1 + 2\sum_{j=1}^{\infty} \rho_{j,\theta},$$

where $\rho_{j,\theta}$ is the correlation of the iterates of θ in the MCMC after the chain has converged. The IACT_θ measures the inefficiency of the sampling scheme with respect to that parameter and can be interpreted as follows: if the IACT_θ = 100, then we need 100 times as many iterates as an independent scheme to obtain the same variance of the estimator. Hence, the larger the value of the IACT_θ, the poorer the performance.

We estimate IACT_θ based on M MCMC iterates $\theta^{[1]}, ..., \theta^{[M]}$ (after convergence) as

$$\widehat{\text{IACT}}_{\theta,M} = 1 + 2\sum_{j=1}^{L_M} \widehat{\rho}_{j,\theta},$$

where $\widehat{\rho}_{j,\theta}$ is the estimate of $\rho_{j,\theta}$, $L_M = \min(1000, L)$ and $L = \min_{j \le M} |\widehat{\rho}_{j,\theta}| < 2/\sqrt{M}$ because $1/\sqrt{M}$ is approximately the standard error of the autocorrelation estimates when the series is white noise.

S5 Initialisation Methods

The selection of starting values for the experiment in Section 3.3 for the DDM using the domain knowledge is now discussed.

1. First, we initialise $\boldsymbol{\mu}_2$ (the mean of $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$) in the natural scale: $v^{(h)} \sim U(-3,-2), v^{(m)} \sim U(1,2), v^{(e)} \sim U(5,6), s_v \sim U(0,2), a \sim U(0.5,2), z = a/2, s_z \sim U(0,0.5), s_\tau \sim U(0,0.1)$, and $\tau = \min \text{RT} + 0.5 s_\tau + 0.1u$, where $u \sim U(0,0.1)$ and $\min \text{RT}$ is the smallest observed response time in the sample. To obtain $\boldsymbol{\mu}_2$, apply the transformations defined in Equation (S3) on these values.

   We now explain how domain knowledge is used in specifying implausible regions for each parameter. Assuming the upper boundary of the diffusion process corresponds to correct responses, it is reasonable to expect, under 'hard' conditions, subjects tend to make incorrect decisions, hence, the drift rate $v^{(h)}$ is expected to be negative. For a similar reason, the drift rate under 'easy' conditions $v^{(e)}$ is expected to be positive and is greater than the drift rate under 'medium' conditions $v^{(m)}$. The plausible intervals for the variability parameters s_v, s_z, s_τ, and the boundary separation a are chosen based on the typical range (see "rtdists" R package). We set $z = a/2$ to exclude bias. Finally, the non-decision time τ is chosen such as the upper bound for non-decision time $L_\tau := \tau - \dfrac{s_\tau}{2}$ must be greater than any observed response time in the data, hence $\tau = \min \text{RT} + 0.5 s_\tau + 0.1u$, where the term $0.1u$ is added to represent some random noise.

2. Next, we initialise $\boldsymbol{\mu}_1$ (means of the random random effects) by sampling from a prior given the chosen value of the group-level mean $\boldsymbol{\mu}_2$ and the group-level covariance which is set to be a diagonal matrix with the diagonal elements equal 0.1, i.e., sample $\boldsymbol{\alpha}_j^0 \sim N(\boldsymbol{\mu}_{\boldsymbol{\alpha}}, 0.1I), j = 1, \ldots, J$ and set $\boldsymbol{\mu}_1 = (\boldsymbol{\alpha}_1^0, \ldots, \boldsymbol{\alpha}_{D_\alpha}^0)$.

3. Finally, we set $\boldsymbol{\mu}_3$ (the mean of the coefficients $\boldsymbol{\beta}$) and $\boldsymbol{\mu}_4$ (the mean of $\log \boldsymbol{a}$) to be zeros.

S6 Simulation study for DDM

The true values in the simulation study (both settings) are given by

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}} = (3.31, 2.12, 1.33, -2.17, -0.48, -0.29, -1.07, -0.28, -1.19, -1.28, -1.52),$$
$$\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}) = (1.44, 0.62, 0.41, 0.72, 0.38, 0.13, 0.09, 0.2, 0.11, 0.15, 0.08, 0.19).$$

We provide the true values for the group-level mean and variances. The true values for the covariances are available upon request. Note that the same true values are used in both

simulation settings ($J = 12$ and $J = 50$).

S7 Simulation study for DDM with regressors

S7.1 Data simulation

We study the performance of the proposed methods using simulation. The data is simulated from the DDM with neural covariates. There are 20 subjects, each subject performs 600 trials under three conditions ('easy ', 'medium ', and 'hard '; there are 200 trials for each condition). In each trial, 5 neural covariates are independently drawn from a Normal distribution with mean 0 and standard deviation 0.001. The fixed-effect coefficients are generated independently from a Normal distribution with mean 0 and standard deviation 2, i.e., $\beta_{ij} \sim N(0, 2^2)$, for $i = 1, \ldots, D_\alpha; j = 1, \ldots, d$.

$$\boldsymbol{\beta} = \begin{bmatrix} -0.24 & 3.46 & 2.96 & -0.71 & 0.03 \\ 1.1 & -2.16 & 3.03 & -0.19 & 0.1 \\ 0.7 & -0.55 & -1.88 & 2.2 & 0.03 \\ 0.72 & 0.36 & -0.37 & -3.93 & -0.02 \\ 1.8 & 3.02 & -2.2 & -2.9 & 0.13 \\ -3.85 & 3.21 & 2.42 & 2.04 & 0.12 \\ 0.52 & -3.68 & -3.25 & -2.84 & 0.02 \\ 1.83 & 3.25 & 0.21 & -1.21 & -0.02 \\ 0.03 & 0.26 & -2.91 & -3.17 & -0.12 \end{bmatrix}$$

Finally, the group-level parameters $\boldsymbol{\mu_\alpha}$ and $\boldsymbol{\Sigma_\alpha}$ are chosen as follows.

$$\boldsymbol{\mu_\alpha} = (0.65, 1.63, 3.53, 0.09, -0.41, -0.46, -0.59, -1.21, -2.14),$$
$$\text{diag}(\boldsymbol{\Sigma_\alpha}) = (0.24, 1.39, 3.83, 0.26, 0.13, 0.19, 0.18, 0.03, 0.45).$$

We provide the true values for the group-level mean and variances. The true values for the covariances are available upon request.

S7.2 Model specification

We consider fitting the DDM having the mixture form (2) with the weight $\overline{\omega} = 0.0001$. Note that the true data generating process is DDM with no contaminant reaction times

($\overline{\omega} = 0$). To capture the trial difficulty, different values for the drift rate means are considered: $v^{(h)}$ (drift rate mean under the hard condition), $v^{(m)}$ (drift rate mean under the moderate condition), and $v^{(e)}$ (drift rate mean under the easy condition). The remaining model parameters are held constant over all conditions. Each subject has 9 random effects

$$\boldsymbol{\eta} := (v^{(h)}, v^{(m)}, v^{(e)}, s_v, a, z, s_z, \tau, s_\tau).$$

To capture the natural constraints of the random effects $\boldsymbol{\eta}$, we apply proper transformations[8] and denote the transformed random effects by $\boldsymbol{\alpha}$. For demonstration purposes, we link all diffusion parameters to the neural covariates.

S7.3 Results

The performance of the PMwG sampler is assessed based on the IACT estimates (efficiency) and the true parameter recovery (accuracy). Tables S1 and S2 show the IACT estimates for the fixed-effect coefficients and the group-level parameters, respectively. Overall, the PMwG sampling scheme is efficient as most of the IACT estimates are small.

	IACT		IACT		IACT		IACT		IACT
β_{11}	1.01	β_{12}	1.00	β_{13}	1.03	β_{14}	1.00	β_{15}	1.04
β_{21}	1.00	β_{22}	1.03	β_{23}	1.02	β_{24}	1.02	β_{25}	1.03
β_{31}	1.03	β_{32}	1.00	β_{33}	1.02	β_{34}	1.02	β_{35}	1.02
β_{41}	1.02	β_{42}	1.00	β_{43}	1.02	β_{44}	1.02	β_{45}	1.01
β_{51}	1.02	β_{52}	1.05	β_{53}	1.01	β_{54}	1.04	β_{55}	1.02
β_{61}	1.01	β_{62}	1.00	β_{63}	1.04	β_{64}	1.00	β_{65}	1.01
β_{71}	1.00	β_{72}	1.02	β_{73}	1.01	β_{74}	1.02	β_{75}	1.04
β_{81}	1.11	β_{82}	1.12	β_{83}	1.03	β_{84}	1.04	β_{85}	1.06
β_{91}	1.03	β_{92}	1.03	β_{93}	1.00	β_{94}	1.01	β_{95}	1.01

Table S1: IACT estimates of the fixed effect coefficients $\boldsymbol{\beta}$ using the PMwG draws.

[8]The transformations are in Section S1.3.2.

Parameter	IACT	Parameter	IACT
$\boldsymbol{\mu}_{\boldsymbol{\alpha}1}$	3.33	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}1}$	5.28
$\boldsymbol{\mu}_{\boldsymbol{\alpha}2}$	1.78	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}2}$	2.27
$\boldsymbol{\mu}_{\boldsymbol{\alpha}3}$	2.19	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}3}$	3.11
$\boldsymbol{\mu}_{\boldsymbol{\alpha}4}$	8.03	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}4}$	7.99
$\boldsymbol{\mu}_{\boldsymbol{\alpha}5}$	14.31	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}5}$	14.99
$\boldsymbol{\mu}_{\boldsymbol{\alpha}6}$	7.25	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}6}$	6.88
$\boldsymbol{\mu}_{\boldsymbol{\alpha}7}$	11.29	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}7}$	14.42
$\boldsymbol{\mu}_{\boldsymbol{\alpha}8}$	2.20	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}8}$	2.14
$\boldsymbol{\mu}_{\boldsymbol{\alpha}9}$	2.80	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}9}$	3.43

Table S2: IACT estimates for the group-level mean $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ and variances (diagonal of $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$) using the PMwG draws.

Figures S2 and S3 display the marginal posterior densities estimated by the PMwG sampler of the fixed-effect coefficients $\boldsymbol{\beta}$ and the group-level parameters, respectively. The figures show that the true values (represented by vertical lines) are recovered accurately.

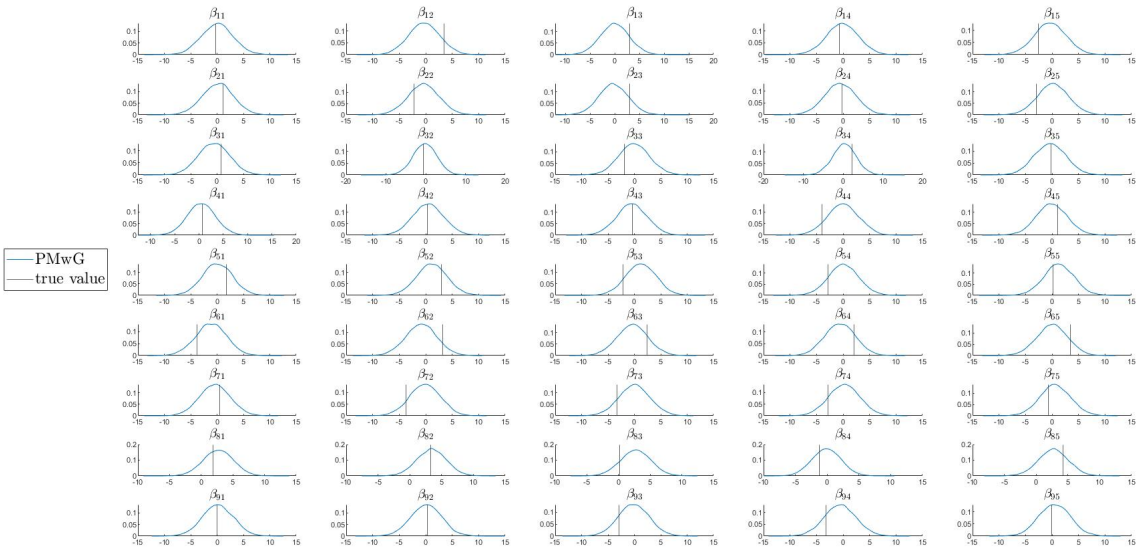

Figure S2: Posterior marginal densities of $\boldsymbol{\beta}$ estimated from PMwG (blue) and the true values indicated by the black vertical lines.

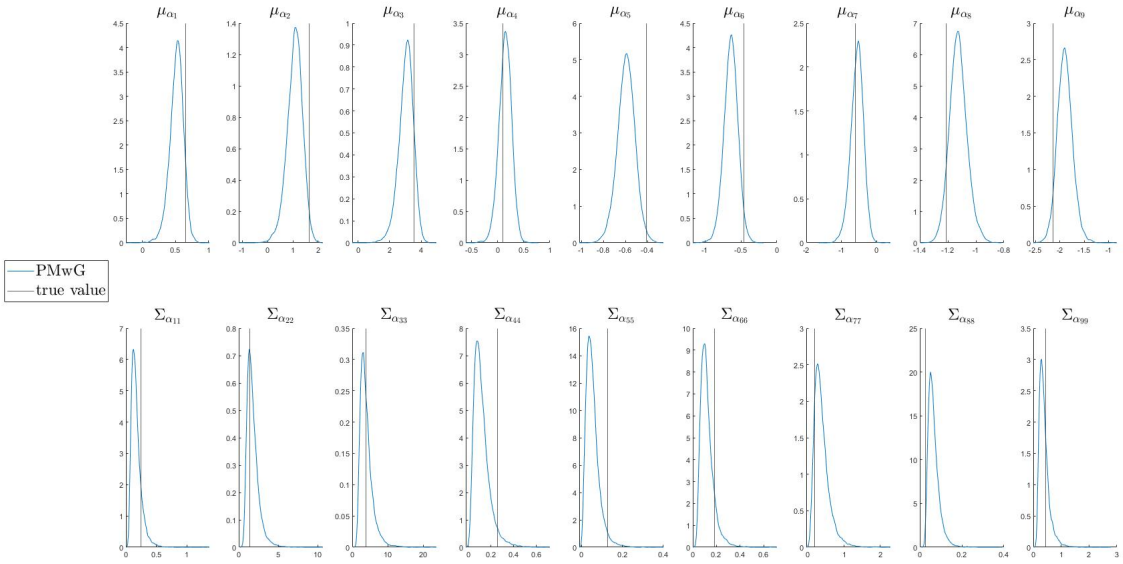

Figure S3: Posterior marginal densities estimated from PMwG of the group-level mean $\boldsymbol{\mu_\alpha}$ (top panels) and variance $\mathrm{diag}(\boldsymbol{\Sigma_\alpha})$ (bottom panels); the true values represented by the black vertical lines.

We now study the performance of the VB approximations. The overall accuracy is assessed by comparing the posterior means and the posterior standard deviations, estimated by PMwG and VB, of all the model parameters and the subject parameters as shown in Figure S4. The main benefit of using VB is its computational efficiency, which is shown in Table S3. VB is 14 times faster than PMwG, and requires much less computational resources.

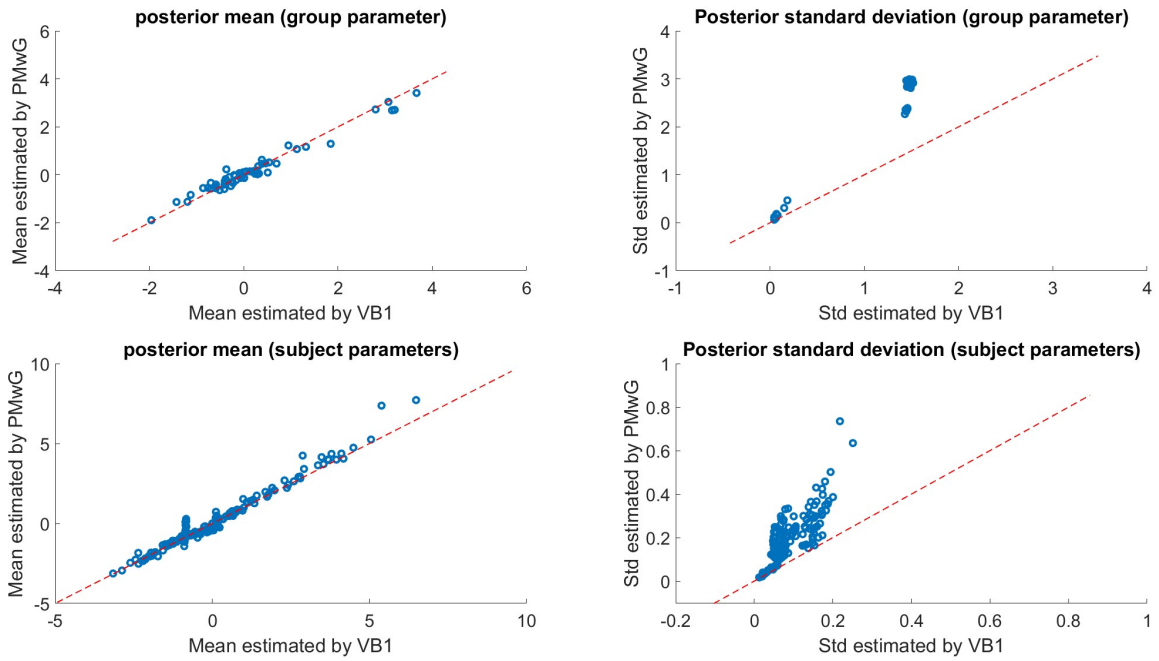

Figure S4: Comparing the means and standard deviations of the marginal posterior distributions estimated by VB (horizontal axis) against the exact values calculated using PMwG (vertical axis). The top panels show the means and standard deviations of the group-level parameters. The bottom panels show the means and standard deviations of the random effects.

Method	**Running time (hour:minute)**	**Number of cpu-cores**
PMwG	41:47	48
VB	3:05	8

Table S3: A comparison between PMwG and VB in terms of running time and computational resources in the simulation study.

S8 Simulating predicted data from EAMs

This section discusses how to simulate predicted data from a RegEAM and estimate the predictive posterior distributions for the summary statistics considered in Section 5.2. Suppose we obtain draws $\{\boldsymbol{\alpha}_{1:J}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}{}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}{}^{(t)}\}_{t=1}^{T}$ from the posterior $p(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} | \boldsymbol{y}, \boldsymbol{X})$ by using PMwG. For each posterior draw, we simulate the predicted data according to the model and calculate the summary statistics. Algorithm S5 gives a step-by-step explanation.

Algorithm S5 Predicted data simulation for EAMs

1. Input: posterior draws $\{\boldsymbol{\alpha}_{1:J}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}{}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}{}^{(t)}\}_{t=1}^{T} \sim p(\boldsymbol{\alpha}_{1:J}, \boldsymbol{\beta}, \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} | \boldsymbol{y}, \boldsymbol{X})$.
2. For $t = 1, \ldots, T$:
   (a) Generate predicted data for subject $j (j = 1, \ldots, J)$ as

$$\tilde{y}_{ij}^{(t)} := (\widetilde{RT}_{ij}^{(t)}, \widetilde{RE}_{ij}^{(t)}) \sim p(y_{ij} | \widetilde{\boldsymbol{\alpha}}_{ij}^{(t)}), \widetilde{\boldsymbol{\alpha}}_{ij}^{(t)} = \boldsymbol{\alpha}_{j}^{(t)} + \boldsymbol{\beta}^{(t)} X_{ij}, \text{ for } i = 1, \ldots, n_j.$$

   (b) From the predicted data $\left\{\widetilde{RT}_{ij}^{(t)}, \widetilde{RE}_{ij}^{(t)}\right\}$, we calculate the summary statistics.
      i. The median response time statistics corresponding to stimulus type $S = s$ ($s \in$ {"same", "mirror"}) and rotation condition $E = e$ ($e \in \{0, 45, 90, 135, 180\}$), denoted by $\bar{M}_{s,e}^{(t)}$, is calculated as follows.

$$\bar{M}_{s,e}^{(t)} = \frac{1}{J} \sum_{j=1}^{J} M_{s,e}^{(t;j)},$$

      with

$$M_{s,e}^{(t;j)} = \text{Med}\left(\left\{\widetilde{RT}_{(s,e)}^{(t;j)}\right\}\right),$$

      where Med($\{U\}$) denotes the sample median of a set U, and $\widetilde{RT}_{(s,e)}^{(t;j)}$ represents all the predicted response time data of subject j from the trials in which the stimulus type $S = s$ and the rotation condition $E = e$.
      ii. The mean accuracy statistics corresponding to stimulus type $S = s$ ($s \in$ {"same", "mirror"}) and rotation condition $E = e$ ($e \in \{0, 45, 90, 135, 180\}$), denoted by $\bar{P}_{s,e}^{(t)}$, is calculated as follows.

$$\bar{P}_{s,e}^{(t)} = \frac{1}{J} \sum_{j=1}^{J} p_{s,e}^{(t;j)},$$

      where $p_{s,e}^{(t;j)}$ is the proportion of correct response of subject j calculated based on trials with stimulus type $S = s$ and rotation condition $E = e$.
3. Construct the box-plot of the summary statistics based on the predicted values $\left\{\bar{M}_{s,e}^{(t)}, \bar{P}_{s,e}^{(t)}\right\}_{t=1}^{T}$.

S9 Further results

This section represents further estimation results of the simulated and real data.

S9.1 Simulation Study

Parameter	IACT		Parameter	IACT	
	$J = 12$	$J = 50$		$J = 12$	$J = 50$
$\boldsymbol{\mu}_{\boldsymbol{\alpha}1}$	3.56	1.19	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}1}$	3.82	1.79
$\boldsymbol{\mu}_{\boldsymbol{\alpha}2}$	4.04	1.34	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}2}$	4.72	2.04
$\boldsymbol{\mu}_{\boldsymbol{\alpha}3}$	3.15	1.23	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}3}$	3.33	1.83
$\boldsymbol{\mu}_{\boldsymbol{\alpha}4}$	3.65	1.13	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}4}$	3.86	1.41
$\boldsymbol{\mu}_{\boldsymbol{\alpha}5}$	3.47	3.85	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}5}$	4.24	5.37
$\boldsymbol{\mu}_{\boldsymbol{\alpha}6}$	4.55	1.93	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}6}$	4.95	1.89
$\boldsymbol{\mu}_{\boldsymbol{\alpha}7}$	10.95	4.59	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}7}$	4.90	4.39
$\boldsymbol{\mu}_{\boldsymbol{\alpha}8}$	2.63	1.70	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}8}$	2.85	1.78
$\boldsymbol{\mu}_{\boldsymbol{\alpha}9}$	8.06	3.14	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}9}$	5.02	3.09
$\boldsymbol{\mu}_{\boldsymbol{\alpha}10}$	69.28	16.05	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}10}$	33.43	9.12
$\boldsymbol{\mu}_{\boldsymbol{\alpha}11}$	2.89	1.08	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}11}$	3.14	1.18
$\boldsymbol{\mu}_{\boldsymbol{\alpha}12}$	3.43	1.12	$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}12}$	3.30	1.23

Table S4: The efficiency (IACT) of the group-level means and variances estimated using the PMwG sampler. J is the number of participants.

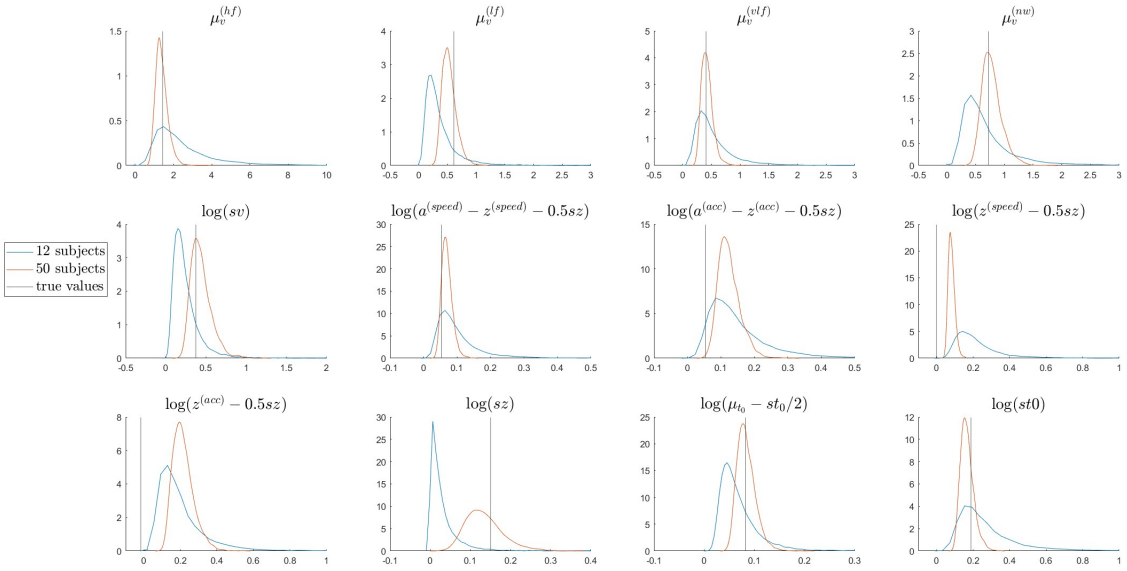

Figure S5: Kernel density estimates of the group-level variance ($\mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$) estimated using the PMwG method in the simulation setting 1 (12 subjects, blue color) and setting 2 (50 subjects, red color). The vertical lines show the true values.

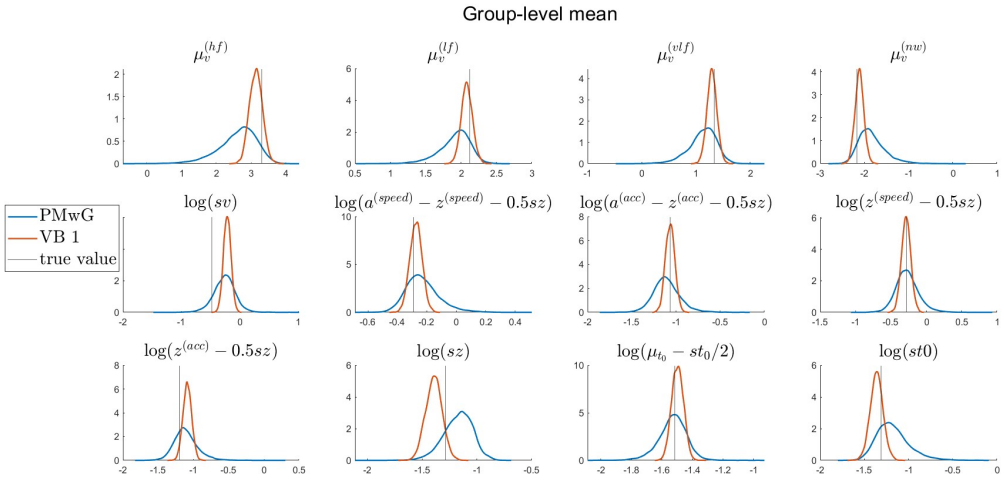

Figure S6: Kernel density estimates of marginal posterior densities of the group level means in the simulation study with $J = 12$ subjects estimated using PMwG and VB. The vertical lines show the true values.

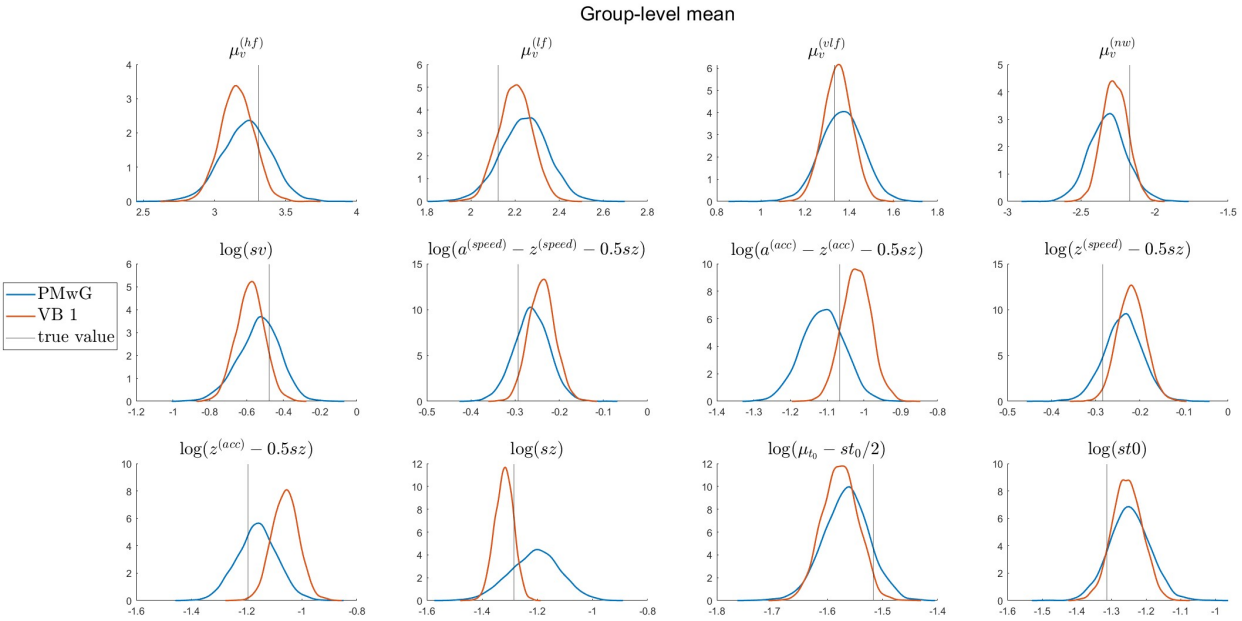

Figure S7: Kernel density estimates of marginal posterior densities of the group level means in the simulation study with $J = 50$ subjects estimated using PMwG and VB. The vertical lines show the true values.

S9.2 Data set 2

Figure S8 and S10 show the kernel density estimates of the group-level mean $\boldsymbol{\mu_\alpha}$ from the LBA and DDM model, respectively. VB captures relatively well the marginal posterior distributions of the group-level mean, except for some components in the LBA. For other group-level parameters as well as the random effects, we assess the accuracy of VB by comparing the mean and standard deviations as shown in figures S9 (LBA) and S11 (DDM). Overall, VB can approximate the posterior means relatively precisely. In terms of computational efficiency, table S5 displays the running time of the two methods. Again, VB is about 10 times faster than PMwG when estimating both models.

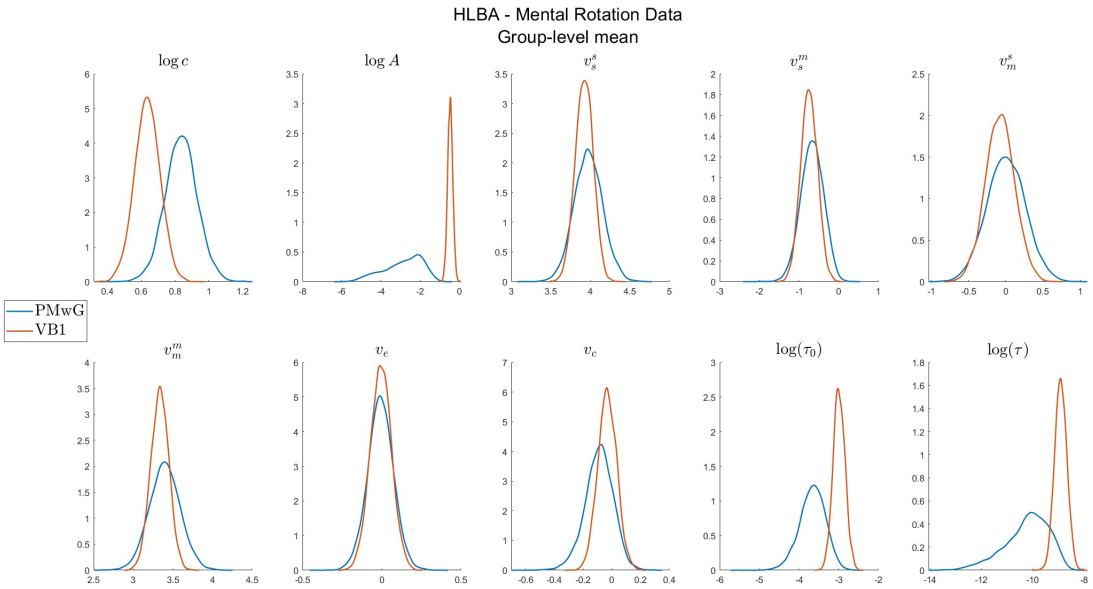

Figure S8: LBA - Kernel density estimates of marginal posterior densities of the group level means estimated using PMwG and VB.

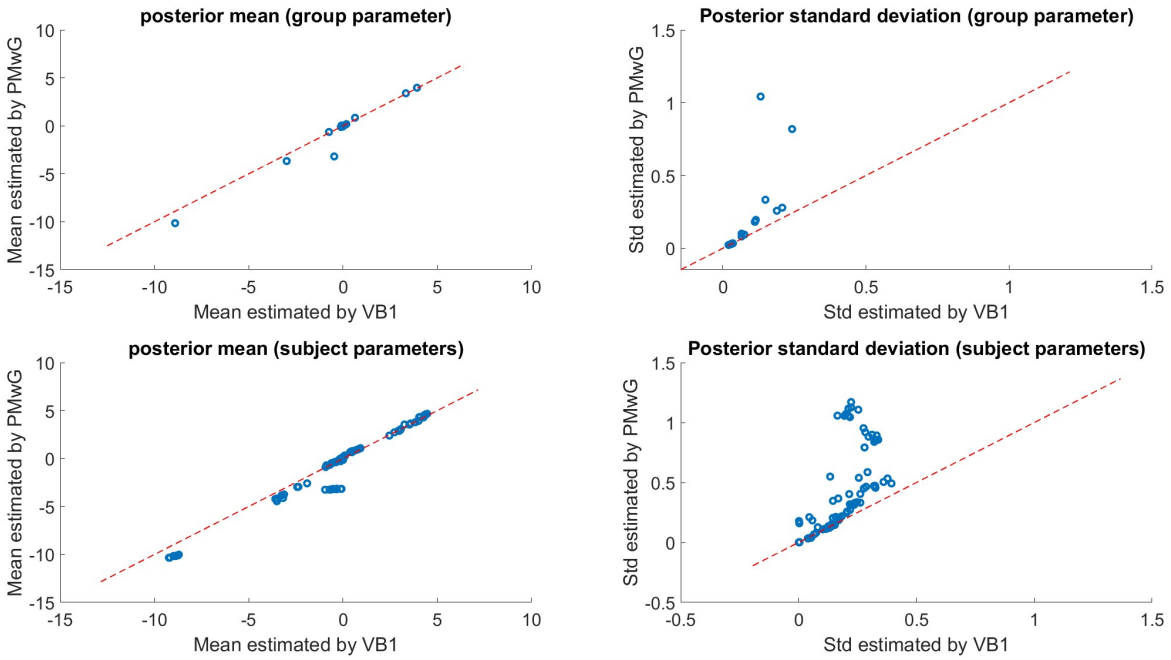

Figure S9: LBA - Comparing the means and standard deviations of the marginal posterior distributions estimated by VB (horizontal axis) against the exact values calculated using PMwG (vertical axis) for the simulation study with $J = 50$ subjects. The top panels give the means and standard deviations of the group-level parameters. The bottom panels show the means and standard deviations of the subject parameters.

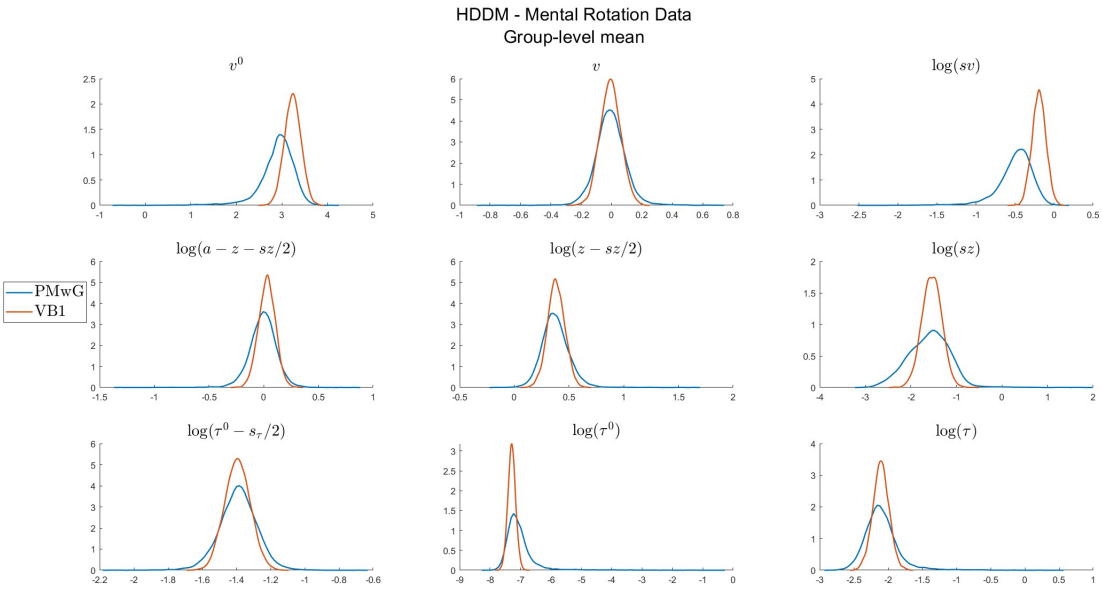

Figure S10: DDM - Kernel density estimates of marginal posterior densities of the group level means estimated using PMwG and VB.

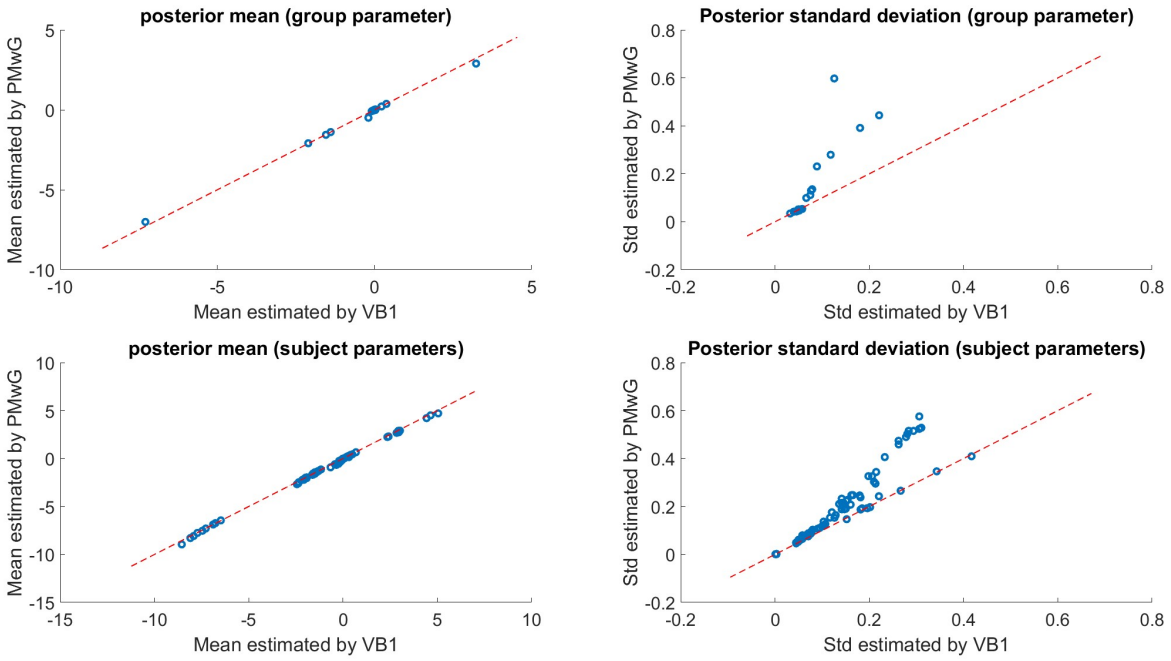

Figure S11: DDM - Comparing the means and standard deviations of the marginal posterior distributions estimated by VB (horizontal axis) against the exact values calculated using PMwG (vertical axis) for the simulation study with $J = 50$ subjects. The top panels give the means and standard deviations of the group-level parameters. The bottom panels show the means and standard deviations of the subject parameters.

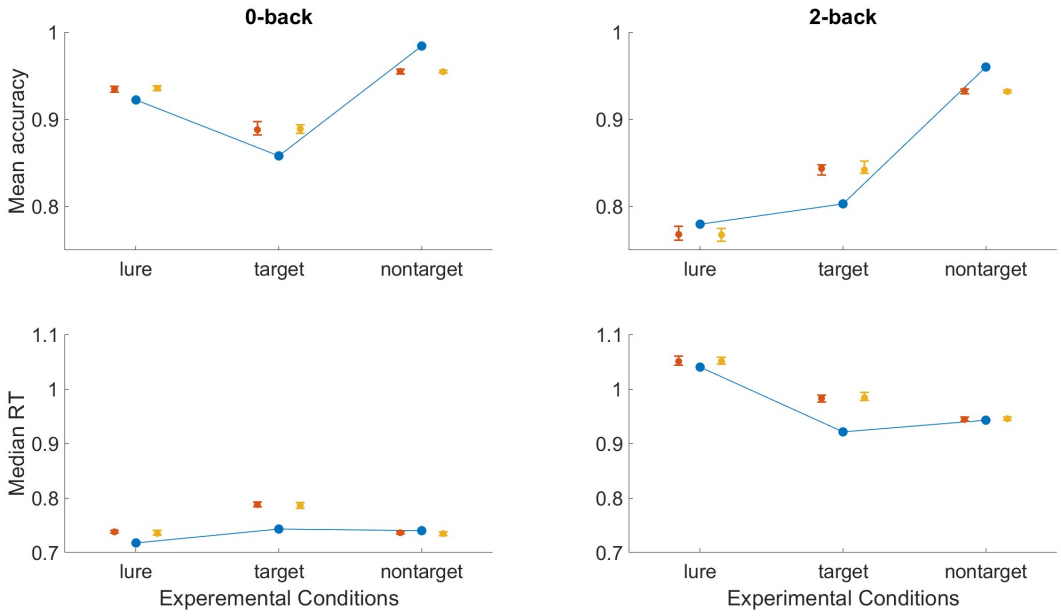

Figure S12: Comparing the estimates of posterior predictive summary statistics for the LBA model using two methods: PMwG (represented by red vertical bars) and VBL (represented by yellow vertical bars), against the observed data (represented by blue dots). Within each vertical bar, the upper and lower tails, along with the dot in between, represent the 2.5%, 97.5%, and 50% quantiles of the predictive distribution, respectively. The top panels show the posterior predictive mean accuracy, while the bottom panels show median RT. Left panels correspond to the 0-back conditions while the right panels correspond to the 2-back conditions.

	LBA	**DDM**
PMwG	1:15	30:56
VB	0:07	5:58

Table S5: Mental Rotation. A comparison between PMwG and VB in terms of running time (hour:minute).

S9.3 Data set 3

Figure S12 compares the posterior predictive distribution obtained from LBA (without covariates) with the observed data. Figure S13 compares the posterior predictive distribution obtained from DDM (without covariates) with the one obtained from LBA (without covariates). The (marginal) posterior distributions of the coefficients of the covariates for the LBA are shown in Figure S14.

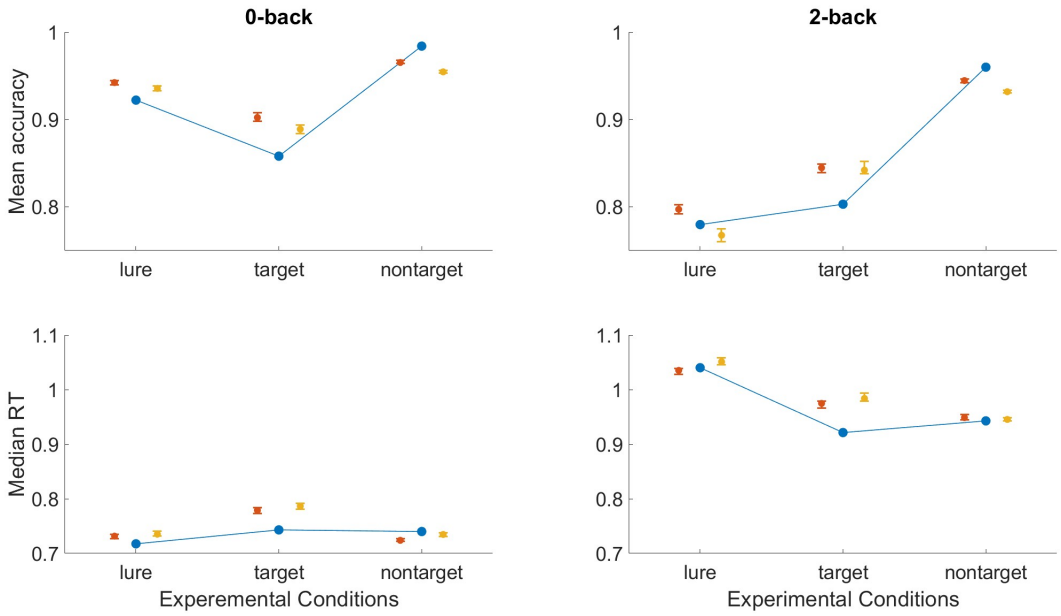

Figure S13: Comparing the estimates of posterior predictive summary statistics for the DDM (represented by red vertical bars) and LBA models (represented by yellow vertical bars) using VBL, against the observed data (represented by blue dots). Within each vertical bar, the upper and lower tails, along with the dot in between, represent the 2.5%, 97.5%, and 50% quantiles of the predictive distribution, respectively. The top panels show the posterior predictive mean accuracy, while the bottom panels show median RT. Left panels correspond to the 0-back conditions while the right panels correspond to the 2-back conditions.

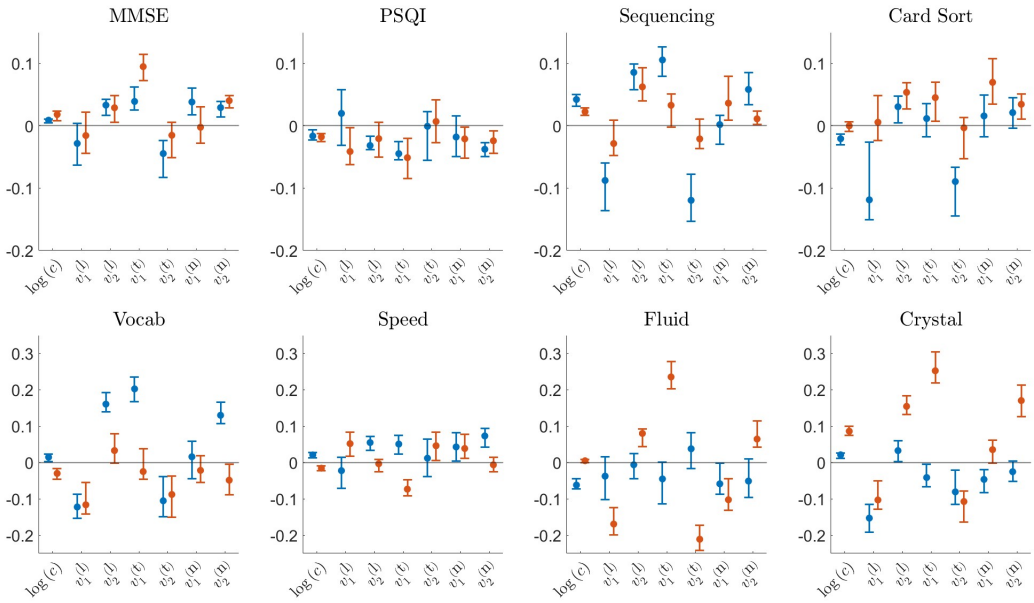

Figure S14: HCP 0-back and 2-back data. Posterior distribution of the coefficients $\boldsymbol{\beta}$ estimated using the VBL method for the LBA, with vertical bars representing the 0-back (blue) and 2-back (red) conditions. Within each vertical bar, the upper and lower tails, along with the dot in between, represent the 2.5%, 97.5%, and 50% quantiles of the posterior distribution, respectively. Each panel corresponds to a covariate. For example, the top left panel shows the posterior distribution of all the coefficients of MMSE. The second top left panel displays the posterior distribution of all coefficients corresponding to PSQI, and so on and for other panels.

References

Brown, S. D. and Heathcote, A. (2008). The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, 57(3):153–178.

Dao, V. H., Gunawan, D., Tran, M.-N., Kohn, R., Hawkins, G. E., and Brown, S. D. (2022). Efficient selection between hierarchical cognitive models: Cross-validation with variational Bayes. *Psychological Methods*.

Donkin, C., Brown, S. D., and Heathcote, A. (2009). The overconstraint of response time models: Rethinking the scaling problem. *Psychonomic Bulletin and Review*, 16(6):1129–1135.

Feller, W. (1968). *An introduction to probability theory and its applications*, volume 1. New York: Wiley, 3 edition.

Foster, K. and Singmann, H. (2021). Another approximation of the first-passage time densities for the ratcliff diffusion decision model. *arXiv preprint arXiv:2104.01902*.

Gondan, M., Blurton, S. P., and Kesselmeier, M. (2014). Even faster and even more accurate first-passage time densities and distributions for the wiener diffusion model. *Journal of Mathematical Psychology*, 60:20–22.

Gunawan, D., Hawkins, G. E., Tran, M.-N., Kohn, R., and Brown, S. (2020). New estimation approaches for the hierarchical linear ballistic accumulator model. *Journal of Mathematical Psychology*, 96.

Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194.

Huang, A. and Wand, M. P. (2013). Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, 8(2):439–452.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Liu, J. S. and Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*, volume 10. Springer.

Navarro, D. J. and Fuss, I. G. (2009). Fast and accurate calculations for first-passage times in wiener diffusion models. *Journal of mathematical psychology*, 53(4):222–230.

Ong, V. M.-H., Nott, D. J., and Smith, M. S. (2018). Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*,

27(3):465–478.

Petersen, K. B. and Pedersen, M. S. (2012). *The matrix cookbook*.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, 85:59–108.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Smith, P. L. (2000). Stochastic dynamic models of response time and accuracy: A foundational primer. *Journal of mathematical psychology*, 44(3):408–463.

Smith, P. L. and Ratcliff, R. (2015). An introduction to the diffusion model of decision making. In *An introduction to model-based cognitive neuroscience*, pages 49–70. Springer.

Tran, M.-N., Nott, D. J., and Kohn, R. (2017). Variational bayes with intractable likelihood. *Journal of Computational and Graphical Statistics*, 26(4):873–882.

Voss, A., Rothermund, K., and Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, 32:1206–1220.

Wall, L., Gunawan, D., Brown, S. D., Tran, M.-N., Kohn, R., and Hawkins, G. E. (2021). Identifying relationships between cognitive processes across tasks, contexts, and time. *Behavior Research Methods*, 53(1):78–95.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.