CREST
CENTER FOR RESEARCH
IN ECONOMICS AND STATISTICS

**Série des Documents de Travail**

# n° 2017-33

# Bayesian Hierarchical Finite Mixture Models of Reading Times: A Case Study

**S.VASISHTH[1]**
**B.NICENBOIM[2]**
**N.CHOPIN[3]**
**R.RYDER[4]**

Les documents de travail ne reflètent pas la position du CREST et n'engagent que leurs auteurs.
Working papers do not reflect the position of CREST but only the views of the authors.

[1] University of Potsdam. E-mail : Shravan.vasishth@ostdam.de

[2] University of Potsdam. E-mail: Bruno.nicenboim@potsdam.de

[3] CREST; ENSAE. E-mail : nicolas.chopin@ensae.fr

[4] CNRS; Université Paris-Dauphine; PSL. E-mail: robin@ceremade.dauphine.fr

Submitted to Psychological Review Draft of July 12, 2017

# Bayesian Hierarchical Finite Mixture Models of Reading Times: A Case Study

Shravan Vasishth
University of Potsdam, Potsdam, Germany

Bruno Nicenboim
University of Potsdam, Potsdam, Germany

Nicolas Chopin
Ecole Nationale de la Statistique et de l'administration économique, Malakoff, France.

Robin Ryder
Centre de Recherche en Mathématiques de la Décision, CNRS, UMR 7534, Université Paris-Dauphine,
PSL Research University, Paris, France.

July 12, 2017

## Abstract

This theoretical note presents a case study demonstrating the importance of Bayesian hierarchical mixture models as a modelling tool for evaluating the predictions of competing theories of cognitive processes. This note also contributes to improving current practices in data analysis in the psychological sciences. As a case study, we revisit two published data sets from psycholinguistics. In sentence comprehension, it is widely assumed that the distance between linguistic co-dependents affects the latency of dependency resolution: the longer the distance, the longer the time taken to complete the dependency (e.g., Gibson 2000). An alternative theory, direct access (McElree, 2000), assumes that retrieval times are a mixture of two distributions (Nicenboim & Vasishth, 2017): one distribution represents successful retrievals and the other represents an initial failure to retrieve the correct dependent, followed by a reanalysis (McElree, 1993) that leads to successful retrieval. Here, dependency distance has the effect that in long-distance conditions the proportion of reanalyses is higher (due to similarity-based interference). We implement both theories as Bayesian hierarchical models and show that the direct-access model fits the Chinese relative clause reading time data better than the dependency-distance account. This work makes several novel contributions. First, we demonstrate how the researcher can reason about the underlying generative process of their data, thereby expressing the underlying cognitive process as a statistical model. Second, we show how models that have been developed in an exploratory manner to represent different underlying generative processes can be compared in terms of their predictive performance, using both K-fold cross validation on existing data, and using completely new data. Finally, we show how the models can be evaluated using simulated data; this is a method that is standardly used in Bayesian statistics, but remains unutilized in data analysis within the psychological sciences.

*Keywords:* Bayesian Hierarchical Finite Mixture Models; Psycholinguistics; Sentence Comprehension; Chinese Relative Clauses; Direct-Access Model; K-fold Cross-Validation

## Introduction

Bayesian cognitive modelling (e.g., Lee & Wagenmakers, 2014) has long played an important role in cognitive science. We present a case study from psycholinguistics showing how hierarchical mixture models can be profitably used to statistically model alternative underlying generative processes that produce the observed data. The case study relates to data from two self-paced reading studies investigating Chinese relative clauses. The hierarchical mixture modelling approach we present will be of general interest to researchers in psychology and related areas.

Please send correspondence to vasishth@uni-potsdam.de.

The processing difficulty associated with Chinese subject vs. object relative clauses has been an important topic of investigation (see Vasishth, Chen, Li, & Guo, 2013, for a review and meta-analysis). Chinese is interesting here because it speaks to an important process that is triggered when we read or hear a sentence. In order to understand the meaning of a sentence, we have to at least figure out who did what to whom. For example, consider a sentence such as (1):

(1) a. The man (on the bench) was sleeping.

To interpret the sentence, the noun *The man* must be recognized to be the subject of the verb phrase *was sleeping*; this dependency is represented here as a directed arrow. One well-known proposal (Just & Carpenter, 1992) is that dependency distance between linguistically related elements partly determines comprehension difficulty as measured by reading times or question-response accuracy. The Just and Carpenter proposal also appears in two current theories, the Dependency Locality Theory (DLT) (Gibson, 2000) and the activation-based account (Lewis & Vasishth, 2005). Both these theories assume that the longer the distance between two co-dependents such as a subject and a verb, the greater the retrieval difficulty at the moment of dependency completion. In the DLT, the reason for the greater difficulty is decay of the dependent in memory, whereas for the activation account, the explanation is that the activation of the dependent is attenuated due to decay as well as interference from other word(s) intervening between the dependents. As shown in (1), the distance between co-dependents increases if a phrase (here, a prepositional phrase) intervenes, leading to greater decay of the mental representation of the noun phrase *the man* by the time it is accessed at the verb, and possibly also greater interference from the intervening noun *bench*.

Consider now the self-paced reading study concerning Chinese subject and object relative clauses reported by Gibson and Wu (2013). As shown in (2), in Chinese, the relative clause precedes the head noun (unlike English, where the relative clause follows the head noun); as a consequence, when the head noun is read, a dependency must be completed between the noun and the corresponding gap in the relative clause.[1]

(2) a. Subject relative

[GAP$_i$ yaoqing fuhao de] guanyuan$_i$ xinhuaibugui

GAP invite tycoon DE official have bad intentions

'The official who invited the tycoon has bad intentions.'

b. Object relative

[fuhao yaoqing GAP$_i$ de] guanyuan$_i$ xinhuaibugui

tycoon invite GAP DE official have bad intentions

'The official who the tycoon invited has bad intentions.'

Interestingly, this dependency distance is larger in subject relatives compared to object relatives, which leads to the surprising prediction that the head noun in subject relatives

[1] The dependency could be equally well be between the relative clause verb and the head noun; nothing hinges on assuming a gap-head noun dependency.

should be harder to process than in object relatives. This prediction is surprising because, in almost all languages that have been investigated, object relatives are more difficult to process than subject relatives (Hsiao & Gibson, 2003).

Thus, the Dependency Locality Theory and the activation account both predict an object relative advantage for Chinese. For simplicity, we operationalize distance here as the number of words intervening between the gap inside the relative clause and the head noun. In the DLT, distance is operationalized as the number of (new) discourse referents intervening between two co-dependents; and in the activation model, the key constructs are not distance per se but decay and interference in working memory. In this paper, nothing hinges on these underlying theoretical details.

In the Gibson and Wu study, reading times were recorded using self-paced reading in the two conditions, with 37 subjects and 15 items, presented in a standard Latin square design. The experiment originally had 16 items, but one item was removed in the published analysis due to a mistake in the item.

The predicted slowdown due to increased dependency distance can be evaluated by fitting the hierarchical linear model shown in (1). Assume that (i) $i$ indexes participants, $i = 1, \dots, I$ and $j$ indexes items, $j = 1, \dots, J$; (ii) $y_{ij}$ is the reading time in milliseconds for the $i$-th participant reading the $j$-th item; and (iii) the predictor $x$ is sum-coded: subject relatives are coded as $-1/2$, and object relatives as $+1/2$; this coding implies that an overall object relative advantage would show a negative coefficient. In other words, with this coding, an object relative advantage corresponds to a negative sign on the estimate. Then, the data $y_{ij}$ (reading times in milliseconds) are defined to be generated by the following model:

$$y_{ij} = \beta_1 + \beta_2 x_{ij} + u_i + w_j + \varepsilon_{ij} \tag{1}$$

In this model, there are three mutually independent sources of variance:

1. The variance associated with residual error: $\varepsilon_{ij} \sim Normal(0, \sigma_e^2)$
2. The variance associated with by-participant adjustments to the intercept: $u_i \sim Normal(0, \sigma_u^2)$
3. The variance associated with by-item adjustments to the intercept: $w_j \sim Normal(0, \sigma_w^2)$

The terms $u_i$ and $w_j$ are called varying intercepts for participants and items respectively; they represent by-participant and by-item adjustments to the fixed-effect intercept $\beta_0$. Their variances, $\sigma_u^2$ and $\sigma_w^2$, represent between-participant (respectively item) variance.[2]

The above model is effectively a statement about the generative process that produced the data. If increasing distance leads to longer reading times, we would expect to find evidence that the estimate of the slope $\beta_2$ is negative; specifically, reading times for object relatives are expected to be shorter than those for subject relatives. As shown in Table 1,

[2] This so-called crossed participants and items varying intercepts hierarchical linear model can be made more complex by adding varying slopes for the predictor $x$ by participant and by item, but for ease of exposition we do not consider these more complex models in the present paper. See Sorensen, Hohenstein, and Vasishth (2016) for a tutorial on how models with full variance-covariance matrices for random effects can be fit in Stan.

this prediction appears, at first sight, to be borne out. Object relatives are estimated to be read 120 ms faster than subject relatives. This effect is statistically significant because the absolute t-value associated with the effect of RC Type is greater than 2.[3]

| | Estimate | Std. Error | t value |
|---|---|---|---|
| Intercept | 548.43 | 51.56 | 10.64* |
| RC Type$_{OR:+0.5}$ | -120.39 | 48.01 | -2.51* |

Table 1

*A hierarchical linear model using raw reading times in milliseconds as dependent variable, corresponding to the reported results in Gibson and Wu 2013. Statistical significance is shown by an asterisk.*

To summarize, the conclusion from the above result would be that in Chinese, subject relatives are harder to process than object relatives because the gap inside the relative clause is more distant from the head noun in subject vs. object relatives. The larger distance makes it more difficult to complete the gap-head noun dependency in subject relatives. This explanation of processing difficulty is plausible given the considerable independent evidence from languages such as English, German, Hindi, Persian and Russian that dependency distance can affect reading time (see the review in Safavi, Husain, & Vasishth, 2016).

However, the distributions of the reading times in the subject vs. object relatives show an interesting and potentially important asymmetry that is not modelled by the standard hierarchical linear model. At the head noun, the reading times in subject relatives are more spread out than in object relatives. This is shown in Figure 1, where reading times are shown in the raw as well as log scale. The heterogeneity in variance seen in raw reading times is ignored by the hierarchical linear model shown above.

Because even a single extreme value can influence the mean, and because extreme values are widely assumed to be non-representative of the underlying generative process, a standard approach is to delete "outliers" based on some criterion. For example, one can delete all data lying beyond $\pm 2SD$ in each condition; the cut-off tends to vary between 2 and 3.5 SDs. In the published paper, Gibson and Wu (2013) did not delete any data, leading to the results shown in Table 1.

This deletion procedure has at least three problems: First, the cut-off criterion assumes that the underlying data have a symmetric, normal distribution. Data points that lie beyond a certain percentile on either side of the distribution would have a low probability of occurrence given a normal distribution with some mean and standard deviation. Such an assumption might be valid in some cases, but it is usually invalid for reading times, which cannot be lower than 0 ms and can only increase in one direction.[4] Second, this deletion approach assumes that the data points identified as extreme are irrelevant to the question being investigated or are recording errors. There are certainly situations where this assumption is justified. But, in situations where all data are in principle considered to be legitimate, it may be very informative to directly model the generative process that produces

[3] The object relative advantage shown in Table 1 was originally carried out by Gibson and Wu (2013) as a repeated measures ANOVA but the conclusion was the same as presented here.

[4] Technically, the lower bound must be greater that 0, and so ideally one should fit a shifted LogNormal distribution; see Rouder (2005) for discussion, and Nicenboim, Engelmann, Suckow, and Vasishth (2017) for an example of how such models can be implemented using Stan (Stan Development Team, 2013).

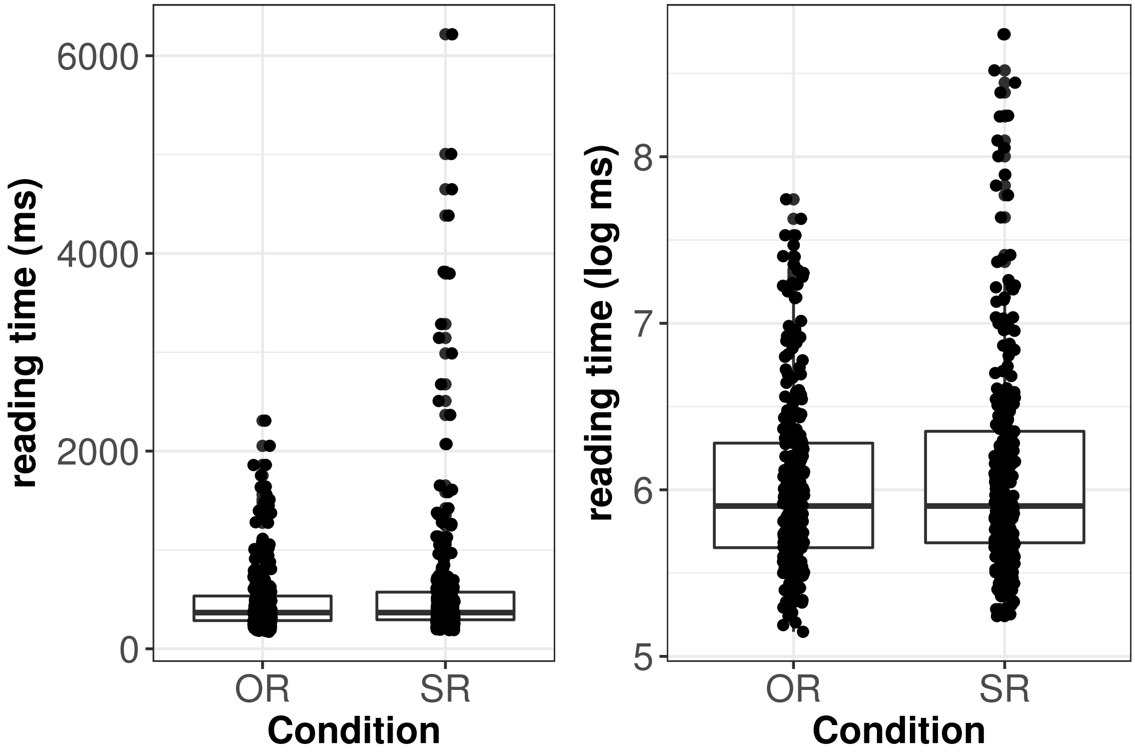

*Figure 1*. Boxplots showing the distribution of reading times (in raw and log ms scale) by condition of the Gibson and Wu (2013) data.

these extreme values. Draper and Smith (1998, 76) make this point as well: "…outliers should be rejected out of hand only if they can be traced to causes such as errors in recording the observations or in setting up the apparatus. Otherwise careful investigation is in order." Third, this criterion of deleting extreme data introduces a degree of flexibility in data analysis. Simmons, Nelson, and Simonsohn (2011) report the results of their survey of the standard practice in psychology as follows: "…researchers excluded some responses for being too fast, but what constituted "too fast" varied enormously: the fastest 2.5%, or faster than 2 standard deviations from the mean, or faster than 100 or 150 or 200 or 300 ms. Similarly, what constituted "too slow" varied enormously: the slowest 2.5% or 10%, or 2 or 2.5 or 3 standard deviations slower than the mean, or 1.5 standard deviations slower from that condition's mean, or slower than 1,000 or 1,200 or 1,500 or 2,000 or 3,000 or 5,000 ms." For example, in the present case, if we decide to retain the extreme values seen in one condition, the mean for that condition would increase, leading to a statistically significant difference, as in Table 1 and in the original analysis by Gibson and Wu (2013). Removing only the three most extreme values (out of 547) that are greater than 4500 ms leads to an evaporation of the statistically significant effect; see Table 2. Thus, the decision to remove data can radically alter the conclusion; depending on whether the researcher hopes to argue for a null result or a significant difference, both outcomes can be accommodated given the data.

An alternative approach is to not delete extreme data but to downweight the extreme

| | Estimate | Std. Error | t value |
|---|---|---|---|
| Intercept | 523 | 45 | 12 |
| RC Type$_{OR:+0.5}$ | -70 | 39 | -1.8 |

Table 2
*A hierarchical linear model of the Gibson and Wu data using raw reading times in milliseconds as dependent variable, with 3 out of 547 extreme data point (reading times greater than 4.5 seconds) removed.*

values by applying a variance stabilizing transform (Box & Cox, 1964). Taking a log-transform of the reading time data, or a reciprocal transform (as recommended in some situations by Ratcliff, 1993), can reduce or eliminate the heterogeneity in variance; see Vasishth et al. (2013) for further discussion of this point for the specific case of Chinese relatives.

Log-transforming reading time data makes the assumption that the data were generated by LogNormal distributions with different means but identical standard deviations in the two conditions. Table 3 shows that if we assume such a generative model, and without deleting any data, there is no longer a statistically significant object relative advantage: the absolute t-value for the estimate of the effect of RC Type is now smaller than the critical value of 2. Thus, assuming that the present data are generated by LogNormal distributions with different means for subject and object relatives, and using statistical significance as a decision criterion, leads to the conclusion that there isn't any evidence against the null hypothesis of no effect of distance.

Note, however, that even in the log-transformed data, the heteroscedasticity in the two conditions is not being modelled (see Figure 1); the model simply assumes identical variances in both conditions.

| | Estimate | Std. Error | t value |
|---|---|---|---|
| Intercept | 6.06 | 0.07 | 92.64* |
| RC Type$_{OR:+0.5}$ | -0.07 | 0.04 | -1.61 |

Table 3
*A hierarchical linear model using log reading times in milliseconds as dependent variable in the Gibson and Wu (2013) data.*

The heteroscedasticity observed in the two conditions may have an alternative explanation in terms of the direct-access model of McElree (2000). Briefly, this theory asserts—in direct contrast to DLT and the activation model—that dependency completion takes constant time regardless of distance. However, when dependency distance increases, there is an increase in the proportion of trials where reanalysis occurs (McElree, 1993). Reanalysis here means that an attempt is made to retrieve the correct dependent but this attempt does not succeed; a process is then triggered that leads to a second attempt to complete the retrieval, and this results in a successful retrieval. This reanalysis process takes more time to complete than the case where retrieval succeeds immediately.

Under the direct-access model, in the Chinese relative clause example (2), retrieval at the head noun in subject relatives leads to more reanalyses due to the fact that a noun (the object of the relative clause) intervenes between the gap and the head noun; this intervening noun causes interference. By contrast, in object relatives, no noun intervenes between the gap and the head noun; rather, the subject of the relative clause precedes the gap.[5]

Nicenboim and Vasishth (2017) noticed that the direct-access model can be seen as assuming a mixture distribution where successful retrieval can be modelled as a LogNormal distribution: $y \sim LogNormal(\mu, \sigma_e^2)$. Reanalysis can be modelled as another LogNormal distribution with a different location parameter and, possibly, also a different scale parameter: $y \sim LogNormal(\mu + \delta, \sigma_{e'}^2)$, where $, \sigma_{e'}^2$ could either be identical to, or different from, $\sigma_e^2$, and $\delta > 0$. Following Nicenboim and Vasishth (2017), we can directly express the McElree model as a hierarchical mixture model and compare it to the hierarchical linear model that represents the proposal by Gibson and others. In this paper, we show that a hierarchical finite mixture model furnishes a better fit to the present data (in terms of predictive accuracy) than the simpler hierarchical linear model.

The finite mixture modelling approach we present here will be of broad interest because heteroscedastic distributions are ubiquitous in reading-time data (see, for example, the 10 data-sets discussed in Vasishth, Jäger, & Nicenboim, 2017), and such heteroscedasticity can be modelled directly. Our goal here is to demonstrate how the researcher can use a flexible set of statistical tools for reasoning about cognitive processes.

The structure of the paper is as follows. We begin by discussing the finite mixture model realization of the direct-access model as developed by Nicenboim and Vasishth (2017). Then, we implement a range of plausible models that could have generated the data. Next, we fit the different models to the Gibson and Wu data, and compare the relative predictive accuracy of the models using K-fold cross-validation. Posterior predictive checks are used to graphically evaluate whether the model selected using K-fold cross-validation realistically reflects the underlying generative process. We validate the conclusions we reached by analyzing new data that was an attempted replication of Gibson and Wu's study. Finally, the model chosen as the best one for both data-sets is validated using simulated data.

## Reading times as a mixture distribution

A finite mixture model assumes that the outcome (here, reading time in milliseconds, $y_i, i = 1, \ldots, N$) is drawn from one of several distributions; here, we consider the case of a two-mixture distribution as an implementation of the direct-access model. For example, consider the situation where 75% of the data come from a LogNormal distribution with mean 5.8 and standard deviation 0.2, and 25% of the remaining data come from a LogNormal(6.8,0.7). Figure 2 shows simulated data from such a mixture distribution. This mixture distribution has a single mode but has the characteristic skew that is also present in the Chinese data.

As mentioned above, we can implement the direct-access model as a hierarchical mixture model with retrieval time in milliseconds assumed to be generated from one of two distributions, where the proportion of trials in which a reanalysis occurs (the mixing proportion) is $p_{sr}$ in subject relatives, and $p_{or}$ in object relatives. The expectation here is the extreme values that are seen in subject relatives are due to $p_{sr}$ being larger than $p_{or}$. One way to write the mixture distribution is with respect to the probability density function

[5] The fact that intervention causes interference can be seen as an instance of the observation by Van Dyke and McElree (2006) that retroactive interference has stronger effects than proactive interference in sentence processing.

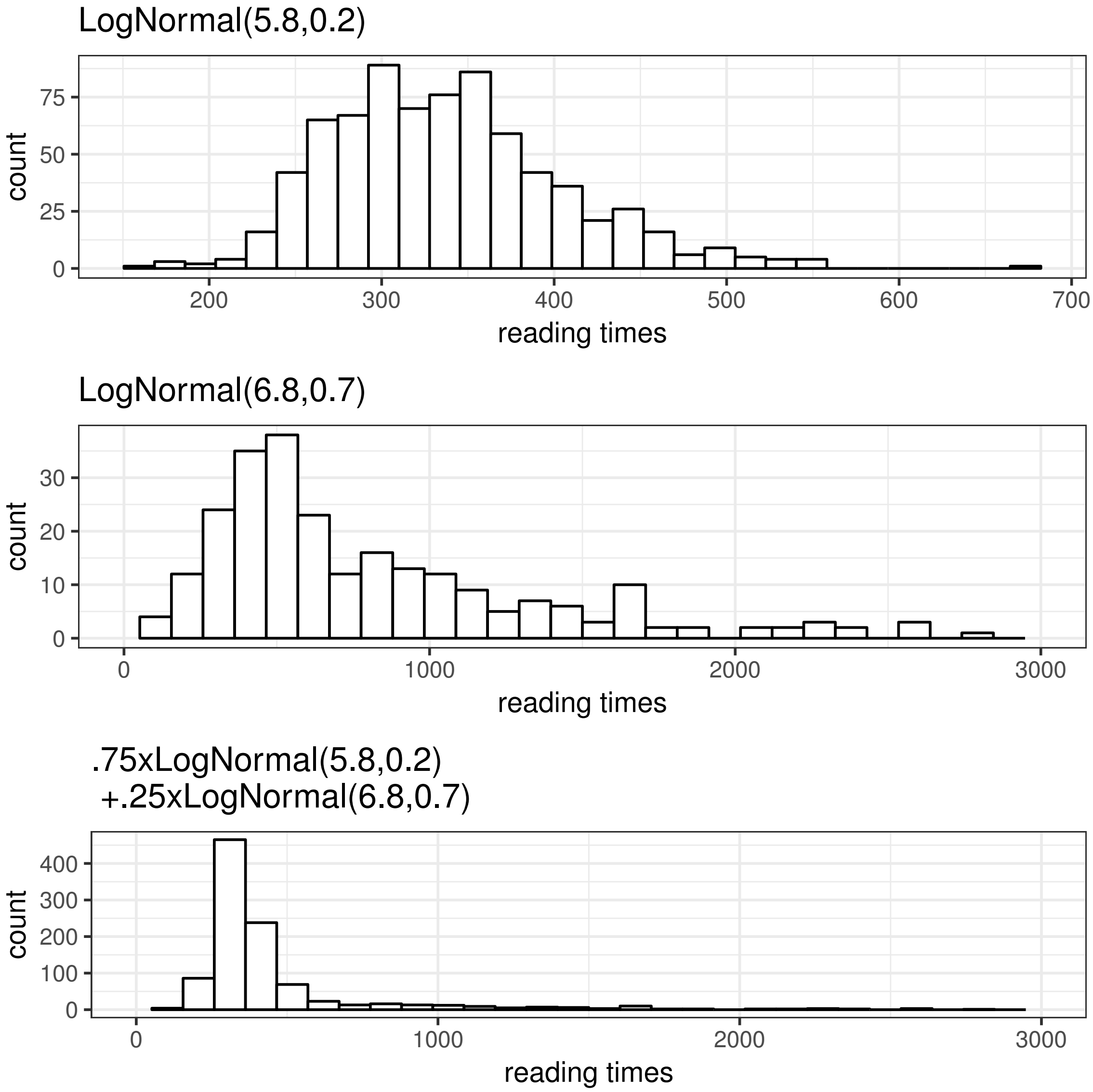

*Figure 2*. An illustration of the distribution of simulated data (1000 data points) consisting of a distribution of two LogNormals: 75% of the data come from LogNormal(5.8,0.2) and 25% come from LogNormal(6.8,0.7). The mixture data (shown in the right-most plot) have a characteristic skew in the right tail.

of the observed reading times $y$. Note that both the mixture distributions shown in (2,3) are identical; all that differs is that the proportion of reanalyses in subject relatives ($p_{sr}$) is expected to be higher than in object relatives ($p_{or}$).

Subject relatives:

$$y_{ij} \sim \begin{cases} LogNormal(\beta + u_i + w_j, \sigma_e^2), & \text{if retrieval succeeds} \\ LogNormal(\beta + \delta + u_i + w_j, \sigma_{e'}^2), & \text{if reanalysis occurs, prob. } p_{sr} \end{cases} \tag{2}$$

Object relatives:

$$y_{ij} \sim \begin{cases} LogNormal(\beta + u_i + w_j, \sigma_e^2), & \text{if retrieval succeeds} \\ LogNormal(\beta + \delta + u_i + w_j, \sigma_{e'}^2), & \text{if reanalysis occurs , prob. } p_{or} \end{cases} \tag{3}$$

Here, the terms $u_i$ and $w_j$ are the varying intercepts by participant and by item, as in equation (1).

In order to understand whether the Chinese relative clause data are better described as being generated by a mixture process, we implemented a series of increasingly complex models. All models were hierarchical, with varying intercepts for participants and for items. In order to compare the models, we used K-fold cross-validation, described next.

### Model comparison

Bayesian model comparison can be carried out using different methods (see Vehtari, Ojanen, et al., 2012, for an extended discussion). Here, we use K-fold cross-validation (Vehtari, Gelman, & Gabry, 2017) because it is a well-known method for model evaluation, and because it is computationally tractable.

This method splits the data into K subsets, taking care that the data remain balanced such that each participants contributes an approximately comparable amount of data from each condition that they saw. The number of subsets is typically 10. Then, the model is fit to the data after holding out 1/K-th of the data and the posterior distributions of the parameters recorded. These parameter estimates are then used to compute predictive accuracy on the held-out data.

The difference between the predicted and observed held-out value is used to quantify model quality. The sum of the expected log pointwise predictive density, $\widehat{elpd}$, can be used as a measure of predictive accuracy. For model comparison, the difference between the $\widehat{elpd}$'s of competing models can be computed, including the standard deviation of the sampling distribution of the difference in $\widehat{elpd}$. When we compare the model described in equation (1) with the mixture model described in equation (2, 3), if the latter has a higher $\widehat{elpd}$, then it has a better predictive performance compared to (1).

The quantity $\widehat{elpd}$ is a Bayesian alternative to the Akaike Information Criterion (Akaike, 1974). Note that it is not necessarily the case that the more complex the model, the better the predictive performance: as shown later, a complex model may well have as good or poorer performance than a simpler model (Gelman, Hwang, & Vehtari, 2014; Vehtari et al., 2012). An alternative to using $\widehat{elpd}$ is to examine $-2 \times \widehat{elpd}$, which is equivalent to deviance, and is called the LOO Information Criterion.

Once we repeatedly carry out this procedure K times, with different held-out data each time, we can compute $\widehat{elpd}$. Further details of the K-fold algorithm are provided in Appendix A, and the code is provided in Appendix B.

## Definitions of the hierarchical mixture models

In all the models described below, the dependent variable is reading time in milliseconds, and the reading times are assumed to be generated from a LogNormal.

In Bayesian modelling, the goal is to derive the posterior distribution of the parameters given the data using Bayes' rule, which states that the posterior distribution of the parameters is proportional to the product of the prior distribution of the parameters multiplied by the likelihood (see Kruschke, 2014, for an accessible introduction for psychologists). Parameters are thus not unknown point values but are random variables with prior distributions (usually mildly informative priors). We defined priors for the model parameters as follows. All standard deviations are constrained to be greater than 0 and have priors $\text{Cauchy}(0, 2.5)$; probabilities have priors $\text{Beta}(1, 1)$; and all other coefficients have priors $\text{Cauchy}(0, 2.5)$.

Common to all the models fitted here is the assumption that participants and items have varying intercepts: $u_i \sim Normal(0, \sigma_u^2)$, $w_j \sim Normal(0, \sigma_w^2)$.

In the mixture models, we will call the distribution that corresponds to the successful retrieval the *success distribution*, and the one corresponding to the reanalysis the *reanalysis distribution*. We fit four models, described below. These models have incrementally increasing complexity.

- M0: A standard hierarchical linear model (no mixture). This corresponds to a test of Gibson's DLT and Lewis and Vasishth's activation account.

- M1: This model assumes a mixture distribution in both subject and object relatives. The model also assumes that there is no difference in retrieval time in ORs vs SRs, but only in the probability of successful retrieval. The variances of the success and reanalysis distributions are assumed to be identical (homogeneous variances).

- M2: This model assumes a mixture in both relative clause types just like M1. It differs from M1 in that the variances of the success and reanalysis distributions are assumed to be different (heterogeneous variances).

- M3: This model assumes that retrieval time in SRs and ORs is different, and that the variances of the two distributions are different (heterogeneous variance). Thus, M3 is like M2, but with the additional assumption that distance may affect dependency completion time, as proposed by Gibson and others. This model is therefore a hybrid of the two proposals.

Thus, the models directly link to the theoretical proposals. M0 corresponds to the DLT and activation account; M1 is the implementation of the direct-access model proposed by Nicenboim and Vasishth (2017); M2 is an extension of the Nicenboim proposal; and M3 extends M2 by incorporating the assumption that, in addition to reanalyses, mean dependency completion times might be different in subject vs. object relatives.

## The data

The evaluation of these models was carried out using two separate data-sets. The first was the original study from Gibson and Wu (2013) that was discussed in the introduction. The second study was a replication attempt of the Gibson and Wu study that was published in Vasishth et al. (2013). This second study will serve the purpose of validating whether independent evidence can be found for the mixture model selected using the original Gibson and Wu data-set.

## Results

We first present results for the Gibson and Wu data, and then for the replication data reported in Vasishth et al. (2013). In the models reported below, the dependent variable is always reading time in milliseconds.

**The original Gibson and Wu study.** As shown in Tables 4 and 5, a comparison of the models M0-M3 using K-fold cross-validation shows that model M2 has a better predictive performance than M0 and M1, but M2 and M3 have similar $\widehat{elpd}$. Among the models considered, the best model—the one with the highest $\widehat{elpd}$—is therefore the heterogeneous variance mixture model M2. As mentioned above, this is an extension of the Nicenboim and Vasishth (2017) model, where the variances of the success and reanalysis distributions are different. Although model M3—which incorporates the assumptions of both the DLT/activation model and the direct-access model—is as good as M2, due to the fact that it has a greater complexity, we settle on M2. Note that our choice of M2 is only on grounds of parsimony; the model comparison does not imply that M3 is worse than M2.

| | models | $\widehat{elpd}$ | se |
|---|---|---|---|
| 1 | M0 | -3759.56 | 37.80 |
| 2 | M1 | -3641.56 | 37.39 |
| 3 | M2 | -3611.96 | 35.12 |
| 4 | M3 | -3614.01 | 34.74 |

Table 4

*Estimated elpd values and their standard errors for the four models investigated for the Gibson and Wu 2013 data.*

| | models | $\Delta\widehat{elpd}$ | se |
|---|---|---|---|
| 1 | M1 vs M0 | 118.00 | 13.82 |
| 2 | M2 vs M1 | 29.61 | 9.28 |
| 3 | M3 vs M2 | -2.05 | 2.52 |

Table 5

*Model comparison using K-fold cross-validation for the Gibson and Wu 2013 data. Shown are the differences in $\widehat{elpd}$, along with standard errors of the differences. In a comparison between model B vs. A, a positive $\widehat{elpd}$ favours model B.*

The estimates from the mixture model (equations 2,3, model M2) are shown in Table 6. Note that in Bayesian modelling, we are not interested in "statistical significance"; the primary goal is the estimation of parameters, understanding the generative process, and

| | mean | lower | upper |
|---|---|---|---|
| $\hat{\beta}_0$ | 5.85 | 5.76 | 5.95 |
| $\hat{\delta}$ | 0.93 | 0.73 | 1.14 |
| $\hat{p}_{sr}$ | 0.25 | 0.17 | 0.34 |
| $\hat{p}_{or}$ | 0.21 | 0.14 | 0.29 |
| $\hat{p}_{sr} - \hat{p}_{or}$ | 0.04 | -0.04 | 0.13 |
| $\hat{\sigma}_{e'}$ | 0.64 | 0.54 | 0.74 |
| $\hat{\sigma}_e$ | 0.22 | 0.20 | 0.25 |
| $\hat{\sigma}_u$ | 0.24 | 0.18 | 0.31 |
| $\hat{\sigma}_w$ | 0.09 | 0.05 | 0.16 |

Table 6

*Posterior parameter estimates from the hierarchical mixture model (equations 2,3) corresponding to the direct-access model. The data are from Gibson and Wu, 2013. Shown are the mean and 95% credible intervals for each parameter.*

comparing the predictive performance of competing models. All parameter estimates are written with a hat ($\hat{\cdot}$). Table 6 shows that the mean difference between the probability $p_{sr}$ and $p_{or}$ is 4% (95% credible interval $-4, 13\%$); the posterior probability of this difference being greater than zero is 82%.

***Model evaluation using posterior predictive checks.*** Posterior predictive checks (Gelman, Carlin, et al., 2014) are a useful and informative way to establish whether a model generates simulated data that is consistent with the observed data. One approach, advocated by Gelman and colleagues, is to compute a relevant statistic derived from the observed data, and then to graphically compare the posterior distribution of that statistic using data predicted by the model. This graphical check allows us to determine the extent to which the generative process implied by the model is realistic. Since models are approximations, the posterior predictive distributions will rarely match the observed data perfectly. Even if the generative process matches the observed data, this does not entail that the model is *the* correct one. However, posterior predictive checks are useful for determining whether essential or particularly interesting aspects of the data are captured. In our case, we want to characterize the generation of relatively rare but very slow reading times in subject relatives. For this reason, we chose as a statistic the 90th percentile of the reading time observed in subject and object relatives. This statistic was chosen because it is one way to characterize the distribution of slower values.

The distributions of the 90th percentiles of the posterior predictive reading times for each relative clause type should, in the ideal case, include the observed percentile as a plausible value. We computed the distribution of the percentiles in 4000 instances of predicted data from the four models considered. Figure 3 shows that, compared to model M0, all the others models are better able to predict the observed 90th percentile of reading times in subject relatives: the observed percentile value falls within the distribution of posterior predicted percentiles. The observed percentile for object relatives is captured by all models.

**The replication of the Gibson and Wu study.** This data-set, originally reported by Vasishth et al. (2013), had 40 new participants but the same 15 items as in the Gibson and Wu data. Figure 4 shows the distribution of the data by condition; there seems to

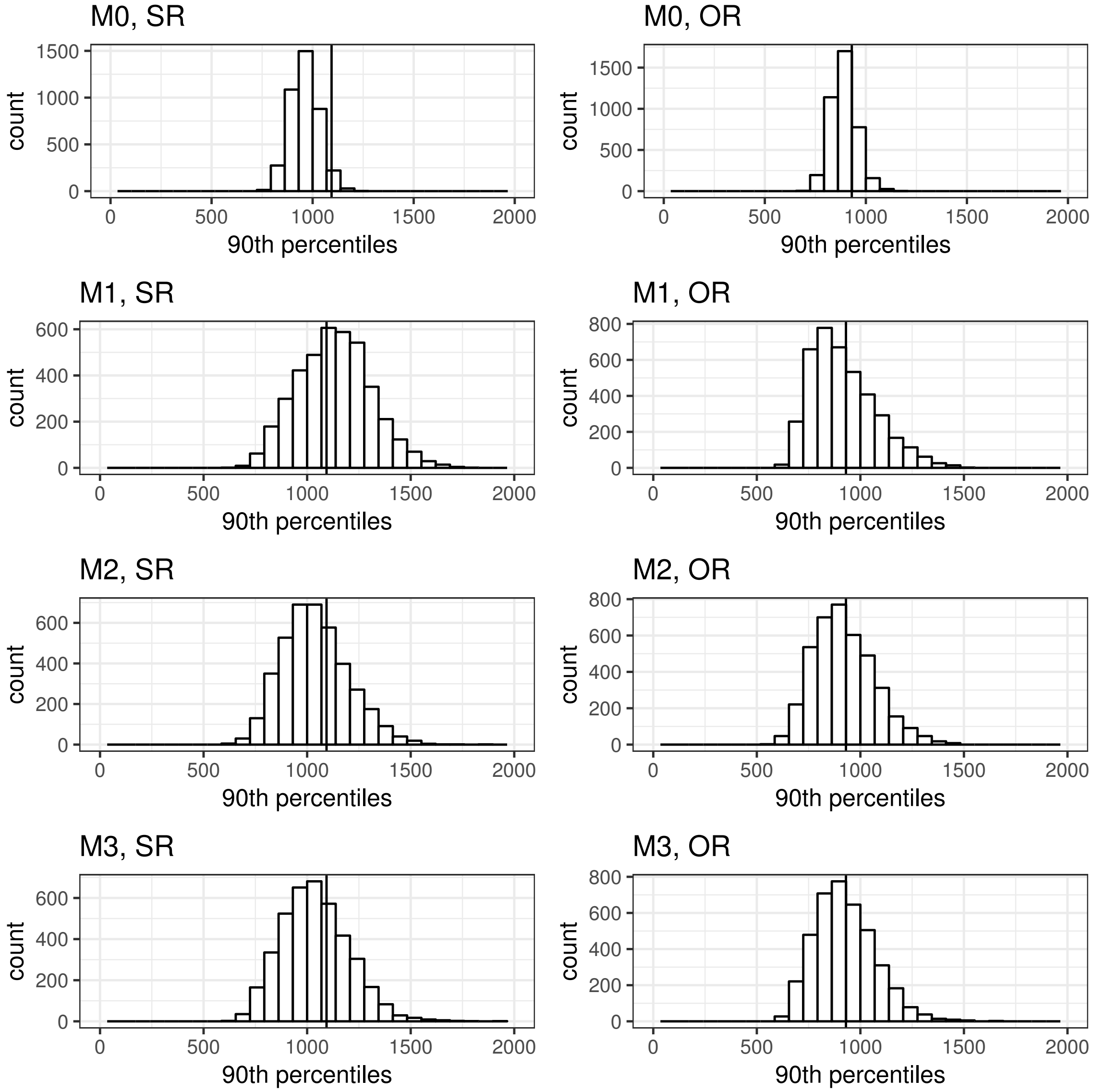

*Figure 3*. The distributions of 90th percentile reading times from the posterior predicted data for subject and object relatives, generated from models M0-M3 (4000 simulations). The vertical lines show the observed 90th percentile of the reading times in the Gibson and Wu 2013 data.

a similar skew as in the original study, although the spread is not as dramatic as in the original study.

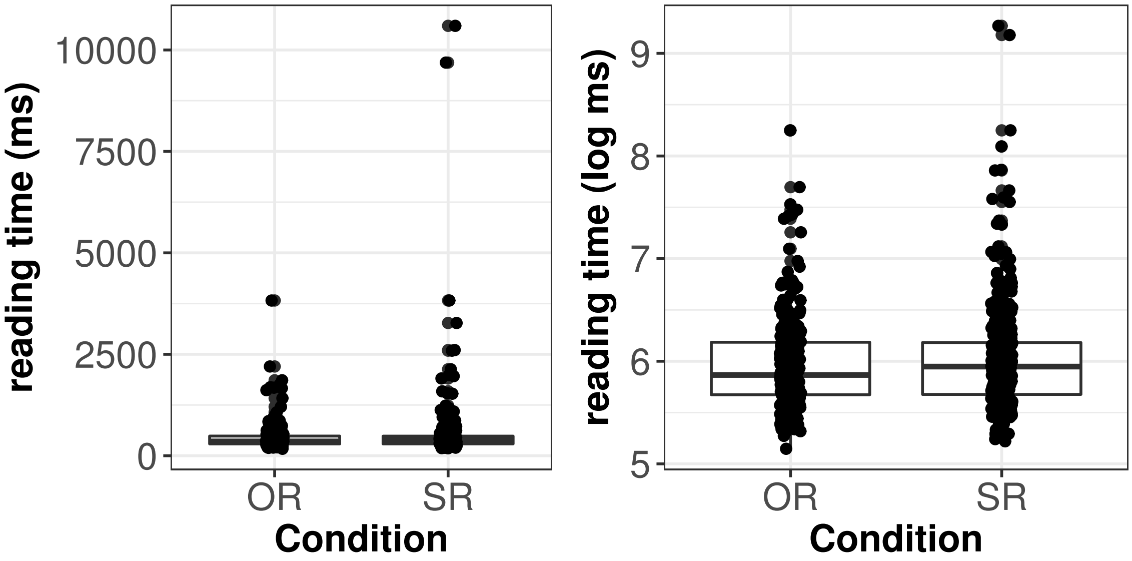

*Figure 4*. Boxplots showing the distribution of reading times by condition of the replication of the Gibson and Wu data.

As shown in Tables 7 and 8, the mixture model M2 shows the best performance and, as in the original data-set, model M3 does no better than M2. Table 9 shows the estimates of the posterior distributions from the mixture model M2. In the mixture model, the mean difference between the probability $p_{sr}$ and $p_{or}$ is 7% (95% credible interval $-1, 15\%$); the posterior probability of this difference being greater than zero is 96%.

| | models | $\widehat{elpd}$ | se |
|---|---|---|---|
| 1 | M0 | -3957.13 | 52.00 |
| 2 | M1 | -3856.58 | 45.87 |
| 3 | M2 | -3797.56 | 37.65 |
| 4 | M3 | -3799.88 | 37.10 |

Table 7
*Estimated elpd values for the four models investigated for the replication of the Gibson and Wu study.*

***Model evaluation using posterior predictive checks.*** We carried out posterior predictive checks here as well. As with the original data, we computed the distribution of the maximum values observed in 4000 instances of predicted reading times for the two relative clause types. As shown in Figure 5, model M2 characterizes the observed percentiles in the two relative clause types better than the other models considered.

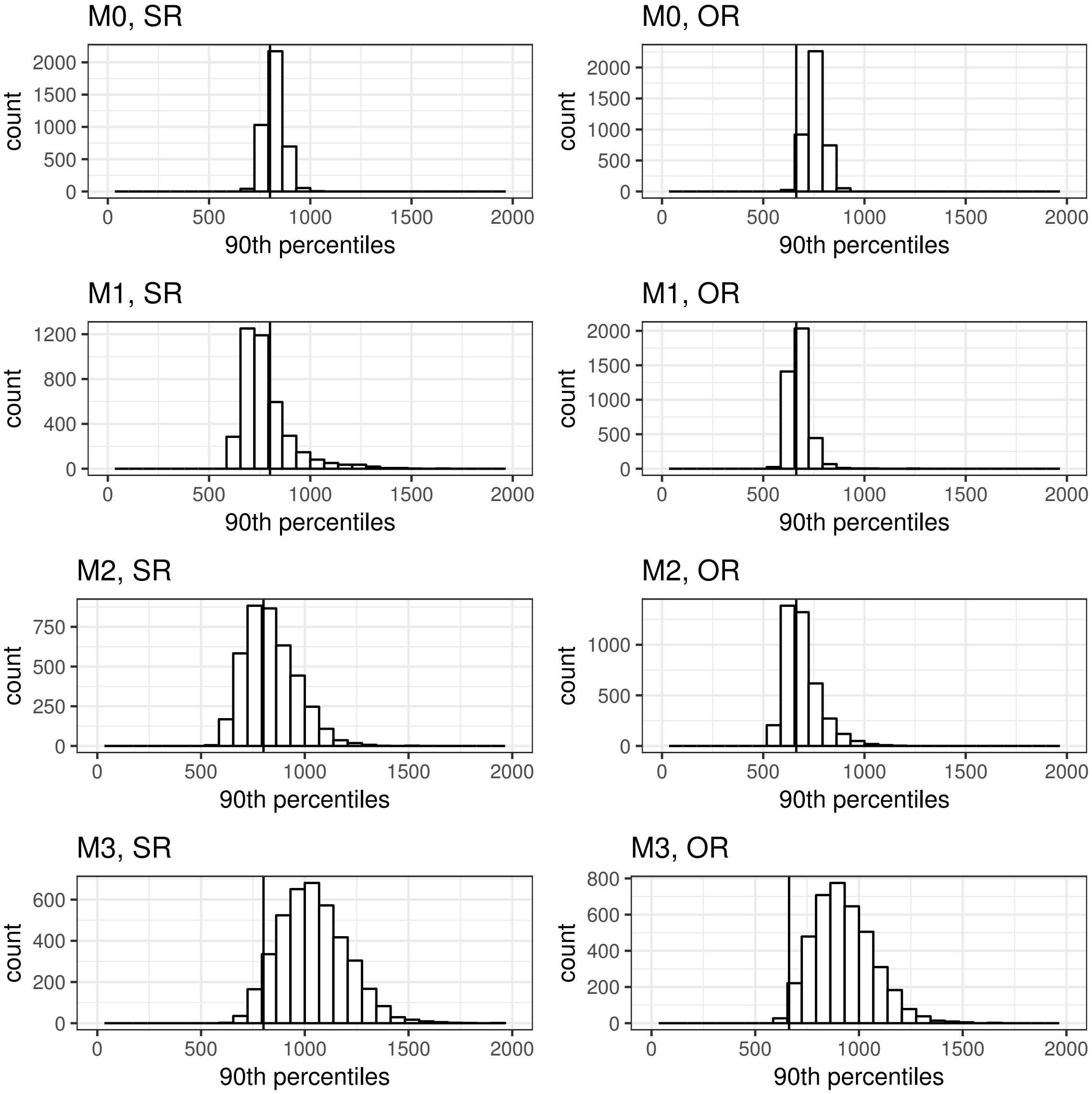

*Figure 5*. The distributions of 90th percentile of reading times from the posterior predicted data for subject and object relatives, generated from models M0-M3 (4000 simulations). The vertical lines show the observed 90th percentile reading times in the replication of the Gibson and Wu 2013 data.

| | models | $\Delta\widehat{elpd}$ | se |
|---|---|---|---|
| 1 | M1 vs M0 | 100.55 | 17.03 |
| 2 | M2 vs M1 | 59.02 | 18.10 |
| 3 | M3 vs M2 | -2.32 | 3.48 |

Table 8

*Model comparison using K-fold cross-validation for the replication of the Gibson and Wu study.*

| | mean | lower | upper |
|---|---|---|---|
| $\hat{\beta}_0$ | 5.86 | 5.78 | 5.95 |
| $\hat{\delta}$ | 0.75 | 0.56 | 0.97 |
| $\hat{p}_{sr}$ | 0.23 | 0.15 | 0.33 |
| $\hat{p}_{or}$ | 0.16 | 0.09 | 0.25 |
| $\hat{p}_{sr} - \hat{p}_{or}$ | 0.07 | -0.01 | 0.15 |
| $\hat{\sigma}_{e'}$ | 0.69 | 0.59 | 0.81 |
| $\hat{\sigma}_e$ | 0.21 | 0.18 | 0.23 |
| $\hat{\sigma}_u$ | 0.22 | 0.17 | 0.29 |
| $\hat{\sigma}_w$ | 0.07 | 0.04 | 0.12 |

Table 9

*Posterior parameter estimates from the hierarchical mixture model M2 corresponding to the direct-access model. The data are from the replication of Gibson and Wu, 2013 reported in Vasishth et al., 2013. Shown are the mean and 95% credible intervals for each parameter.*

## Discussion

The model comparison and parameter estimates presented above suggest that, at least as far as these Chinese relative clause data are concerned, the direct-access model may explain the data better than the DLT or the activation account. The Gibson and Wu (2013) data and the replication data suggest that a higher proportion of reanalyses occurred in subject relatives compared to object relatives. In other words, increased dependency distance may have the effect that it increases the proportion of reanalyses. It is important to note that this conclusion should be seen as tentative until confirmed or falsified by new data, preferably from a large sample.

There is one potential objection to the conclusion above. It would be important to obtain independent evidence as to which dependency was eventually created in each trial. This could be achieved by asking participants multiple-choice questions to find out which dependency they built in each trial. Although such data is not available for the present study, in other work (on number interference) Nicenboim et al. (2017) did collect this information. There, too, the direct-access model was able to characterize the data better than the baseline model (Nicenboim & Vasishth, 2017). In future work on Chinese relatives, it would be helpful to carry out a similar study to determine which dependency was completed in each trial. In the present work, the modelling at least shows how the extreme values in subject relatives can be better accounted for by assuming a two-mixture process.

A further question raised by the modelling must be addressed. It is not self-evident that the mixture model M2 that we settled on above is valid in the following sense: Does it recover the true underlying parameters when these are known? We investigate this next.

## Validating the mixture model using simulated data

We first simulated data from a mixture distribution with known parameter values, and then sampled from the models representing the direct-access model (M2), the model that we settled on. The data were generated from a mixture process defined as follows. Recall that the mixture model is:

Subject relatives:

$$y_{ij} \sim \begin{cases} LogNormal(\beta + u_i + w_j, \sigma_e^2), & \text{if retrieval succeeds} \\ LogNormal(\beta + \delta + u_i + w_j, \sigma_{e'}^2), & \text{if reanalysis occurs, prob. } p_{sr} \end{cases} \quad (4)$$

Object relatives:

$$y_{ij} \sim \begin{cases} LogNormal(\beta + u_i + w_j, \sigma_e^2), & \text{if retrieval succeeds} \\ LogNormal(\beta + \delta + u_i + w_j, \sigma_{e'}^2), & \text{if reanalysis occurs, prob. } p_{or} \end{cases} \quad (5)$$

The true parameter values were: $p_{sr} = 0.25$, $p_{or} = 0.21$, $\beta = 5.85$, $\delta = 0.93$, $\sigma_{e'} = 0.64$, $\sigma_e = 0.22$, $\sigma_u = 0.24$, $\sigma_w = 0.09$.

Using these parameter settings, we generated data first from 37 simulated participants, and then from 148 participants. In order to mimic the design in the original Gibson and Wu (2013) experiment, we always had 15 items. The large sample simulation helps us determine whether the posterior distributions contain the true value of the parameter in the case where these is enough data to obtain precise posterior distributions.

Having generated simulated data, we fit the mixture model (M2) to these data and examined the posterior distributions of the parameters in each model.

### Results based on simulated data

When the data were generated from a mixture process (simulating 37 or 148 participants and 15 items), the mixture model M2 can recover all the underlying parameters; this is shown in Figure 6 for 37 participants and in Figure 7 for 168 participants. These figures show that the true values of the parameters lie within the range of plausible values implied by the posterior distributions. In the large-sample simulation, the error term $\sigma_e$ is being overestimated. Nevertheless, the true value of $\sigma_e$ that was used for generating the data is not impossible given the posterior distribution.

## Conclusion

We presented a case study demonstrating how informative is to be able to specify Bayesian hierarchical models for theorizing about the generative process underlying behavioural (here, reading time) data. We compared the predictive performance of standard hierarchical linear models and two-mixture models for Chinese relative clause data. Model comparison suggests that the increased processing difficulty observed in Chinese subject relatives may not be due to dependency distance leading to longer reading times, as suggested

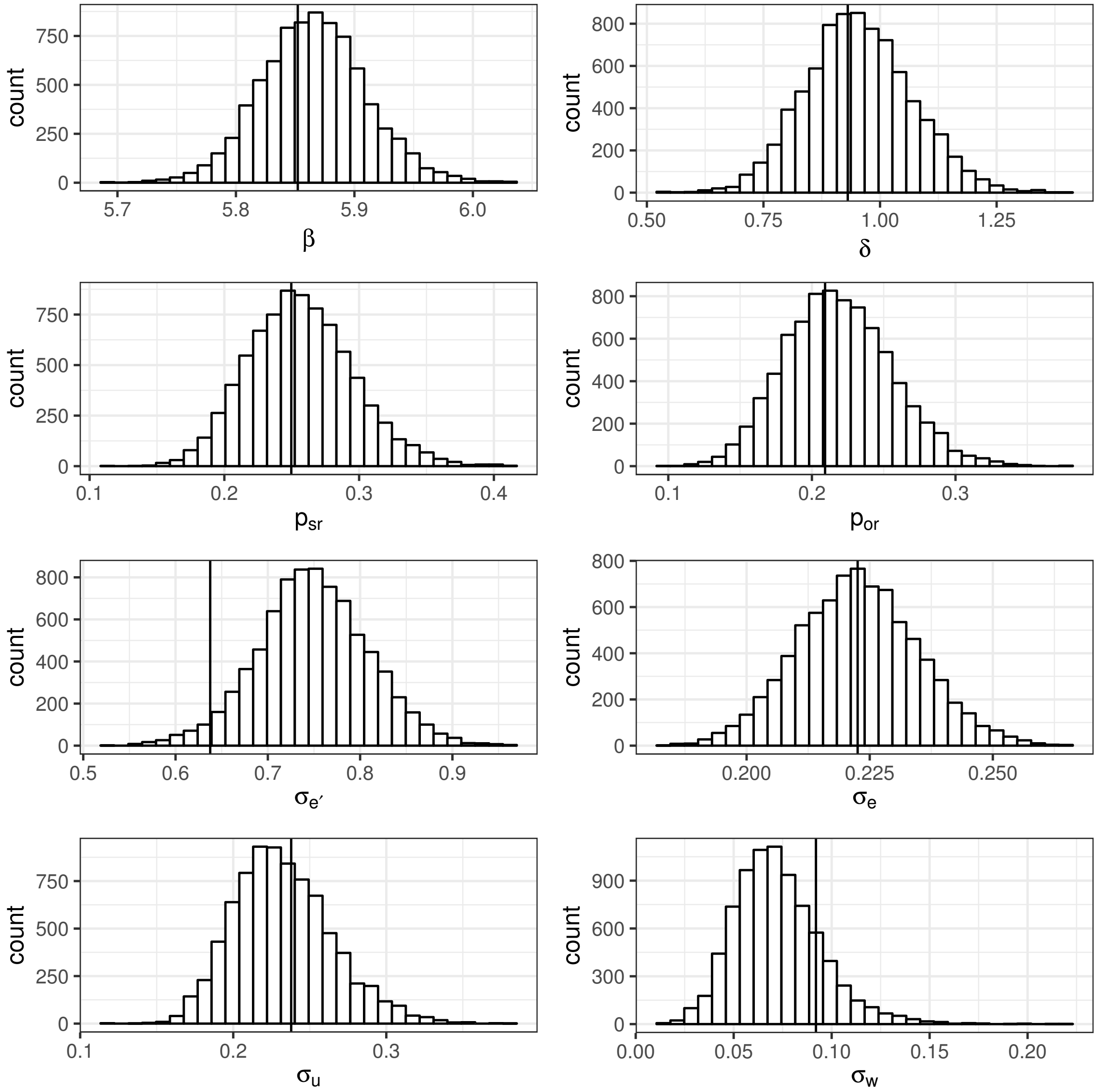

*Figure 6*. The vertical lines show the true point values that were used to generate simulated mixture distribution data (37 participants, and 15 items), and the posterior distributions of the parameters estimated from the hierarchical mixture model (M2).

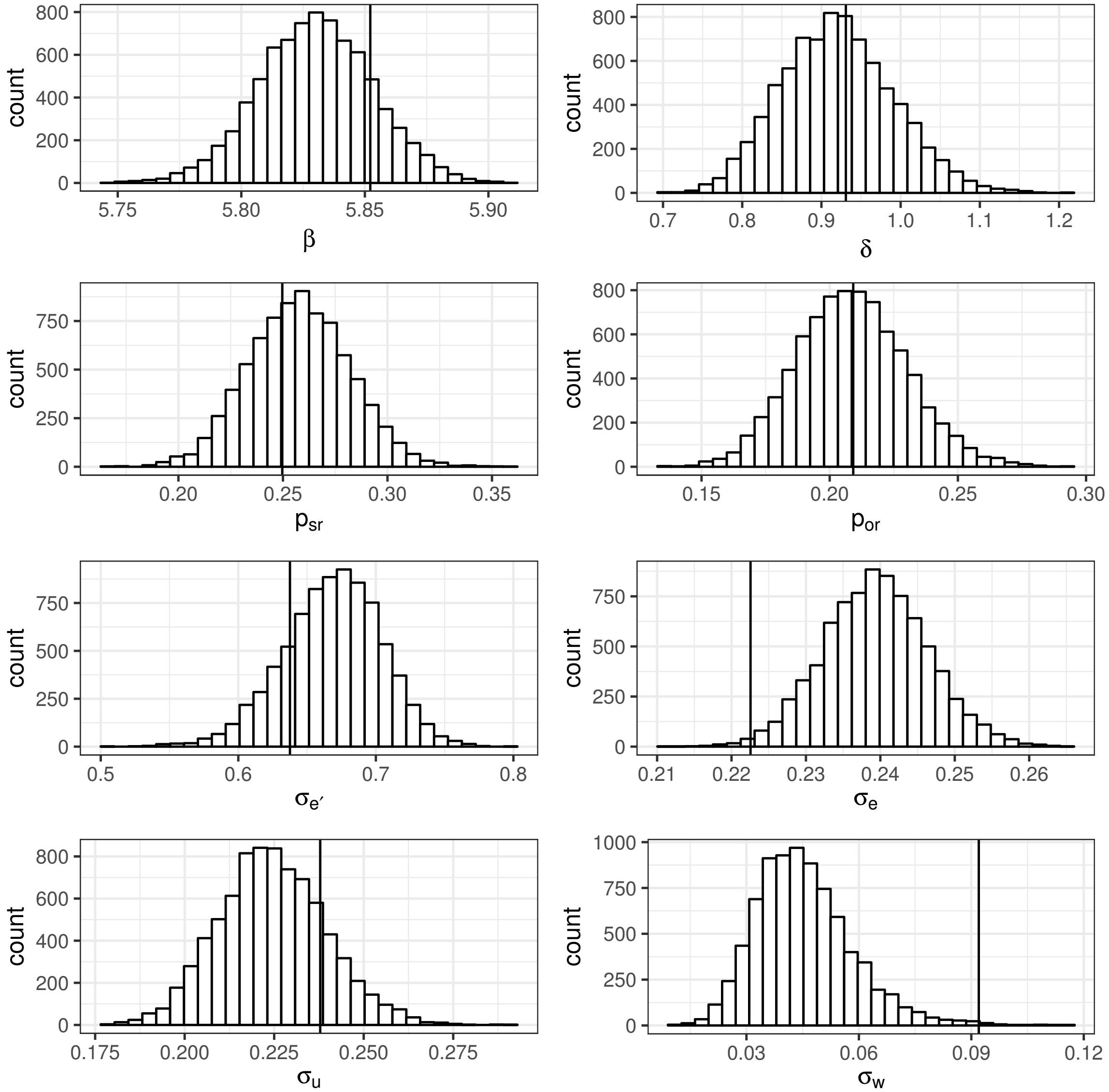

*Figure 7.* The vertical lines show the true point values that were used to generate simulated mixture distribution data (148 participants, and 15 items), and the posterior distributions of the parameters estimated from the hierarchical mixture model (M2).

by the DLT and the activation account. Rather, a more plausible explanation for these data may be in terms of the direct-access model of McElree (2000). Under this view, retrieval times are not affected by the distance between co-dependents; instead, a higher proportion of reanalyses occur in subject relatives compared to object relatives. A mixture distribution generates both subject and object relative clause data, but the proportion associated with the reanalysis distribution is higher in subject relatives.

More broadly, this paper demonstrates the importance of Bayesian finite mixture models for theory development. Had we followed the conventional procedure of fitting a canned linear mixed model to data without questioning the violations of model assumptions, we would have failed to notice theoretically important patterns in the data. The inherent flexibility of Bayesian methods allowed us to go beyond one variety of hierarchical model and to quantitatively explore alternative explanations for the data. This kind of modelling approach can be profitably used in many different research problems in cognitive science.

## Acknowledgments

We are very grateful to Ted Gibson for generously providing the raw data and the experimental items from Gibson and Wu (2013). Thanks also go to Lena Jäger for many insightful comments. Helpful observations by Aki Vehtari are also gratefully acknowledged. For partial support of this research, we thank the Volkswagen Foundation funding through grant 89 953, and the Deutsche Forschungsgemeinschaft for funding through the Collaborative Research Center (SFB) 1287.

## References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 211–252.

Draper, N., & Smith, H. (1998). *Applied regression analysis.* New York: Wiley.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayesian data analysis* (Third ed.). Chapman and Hall/CRC.

Gelman, A., Hwang, J., & Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, *24*(6), 997–1016.

Gibson, E. (2000). Dependency locality theory: A distance-based theory of linguistic complexity. In A. Marantz, Y. Miyashita, & W. O'Neil (Eds.), *Image, language, brain: Papers from the first mind articulation project symposium.* Cambridge, MA: MIT Press.

Gibson, E., & Wu, H.-H. I. (2013). Processing Chinese relative clauses in context. *Language and Cognitive Processes*, *28*(1-2), 125–155.

Hsiao, F. P.-F., & Gibson, E. (2003). Processing relative clauses in Chinese. *Cognition*, *90*, 3–27.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, *99(1)*, 122–149.

Kruschke, J. (2014). *Doing bayesian data analysis: A tutorial with R, JAGS, and Stan.* Academic Press.

Lee, M. D., & Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course.* Cambridge University Press.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 1–45.

McElree, B. (1993). The locus of lexical preference effects in sentence comprehension: A time-course analysis. *Journal of Memory and Language*, *32*(4), 536–571.

McElree, B. (2000). Sentence comprehension is mediated by content-addressable memory structures. *Journal of Psycholinguistic Research*, *29(2)*, 111–123.

Nicenboim, B., Engelmann, F., Suckow, K., & Vasishth, S. (2017). *Number interference in German: Evidence for cue-based retrieval.* Retrieved from https://osf.io/mmr7s/ (submitted to Cognitive Science)

Nicenboim, B., & Vasishth, S. (2017). *Models of retrieval in sentence comprehension: A computational evaluation using Bayesian hierarchical modeling.* Retrieved from https://arxiv.org/abs/1612.04174 (Under revision following review, Journal of Memory and Language)

Ratcliff, R. (1993). Methods for dealing with reaction time outliers. *Psychological Bulletin*, *114*(3), 510-532.

Rouder, J. N. (2005). Are unshifted distributional models appropriate for response time? *Psychometrika*, *70*(2), 377–381.

Safavi, M. S., Husain, S., & Vasishth, S. (2016). Dependency resolution difficulty increases with distance in Persian separable complex predicates: Implications for expectation and memory-based accounts. *Frontiers in Psychology*, *7*. doi: 10.3389/fpsyg.2016.00403

Simmons, J., Nelson, L., & Simonsohn, U. (2011). False-positive psychology. *Psychological Science*, *22*(11), 1359–1366.

Sorensen, T., Hohenstein, S., & Vasishth, S. (2016). Bayesian linear mixed models using Stan: A tutorial for psychologists, linguists, and cognitive scientists. *Quantitative Methods for Psychology*, *12*(3), 175-200. Retrieved from http://www.ling.uni-potsdam.de/~vasishth/statistics/BayesLMMs.html

Stan Development Team. (2013). *Stan: A C++ library for probability and sampling, version 2.1.* Retrieved from http://mc-stan.org/

Stan Development Team. (2016). Stan modeling language users guide and reference manual, version 2.12 [Computer software manual]. Retrieved from http://mc-stan.org/

Van Dyke, J., & McElree, B. (2006). Retrieval interference in sentence comprehension. *Journal of Memory and Language*, *55*, 157–166.

Vasishth, S., Chen, Z., Li, Q., & Guo, G. (2013, 10). Processing Chinese relative clauses: Evidence for the subject-relative advantage. *PLoS ONE*, *8*(10), 1–14.

Vasishth, S., Jäger, L. A., & Nicenboim, B. (2017). Feature overwriting as a finite mixture process: Evidence from comprehension data. In *Proceedings of MathPsych/ICCM.* Warwick, UK. Retrieved from https://arxiv.org/abs/1703.04081

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*(5), 1413–1432.

Vehtari, A., Ojanen, J., et al. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, *6*, 142–228.

# Appendix A
## K-fold cross-validation

The K-fold cross-validation algorithm works as follows:

1. Split data pseudo-randomly into $K$ *held-out* sets $\mathbf{y}_{(k)}$, where $k = 1, \ldots, K$ that are a fraction of the original data, and $K$ *training sets*, $\mathbf{y}_{(-k)}$. Here, we use $K = 10$, and the length of the held-out data-vector $\mathbf{y}_{(k)}$ is approximately $1/K$-th the size of the full data-set. We ensure that each participant's data appears in the training set and contains an approximately balanced number of data points for each condition.

2. Sample from the model using each of the $K$ training sets, and obtain posterior

distributions $p_{\text{post(-k)}}(\theta) = p(\theta \mid \mathbf{y}_{(-k)})$, where $\theta$ is the vector of model parameters.

3. The posterior distributions $p(\theta \mid \mathbf{y}_{(-k)})$ are used to compute predictive accuracy for each held-out data-point $y_i$:

$$\log p(y_i \mid \mathbf{y}_{(-k)}) = \log \int p(y_i \mid \theta) p(\theta \mid \mathbf{y}_{(-k)})\, d\theta \tag{6}$$

4. Given that the posterior distributions $p(\theta \mid \mathbf{y}_{(-k)})$ are summarized by $s = 1, \ldots, S$ simulations, i.e., $\theta^{k,s}$, the log predictive density for each data point $y_i$ in subset $k$ is computed as

$$\widehat{elpd}_i = \log \left( \frac{1}{S} \sum_{s=1}^{S} p(y_i \mid \theta^{k,s}) \right) \tag{7}$$

5. Given that all the held-out data in the $K$ subsets are $y_i$, where $i = 1, \ldots, n$, we obtain the $\widehat{elpd}$ for all the held-out data points by summing up the $\widehat{elpd}_i$ for each held-out data point:

$$\widehat{elpd} = \sum_{i=1}^{n} \widehat{elpd}_i \tag{8}$$

The difference between the $\widehat{elpd}$'s of two competing models is a measure of relative predictive performance. We can also compute the standard deviation of the sampling distribution (the standard error) of the difference in $\widehat{elpd}$ using the formula discussed in Vehtari et al. (2017). Letting $\widehat{elpd}_{i,m0}$ be the estimated elpd for the i-th data point from model M0, we can write:

$$se(\widehat{elpd}_{m0} - \widehat{elpd}_{m1}) = \sqrt{nVar(\widehat{elpd}_{i,m0} - \widehat{elpd}_{i,m1})} \tag{9}$$

## Appendix B
## Stan code for implementing mixture models

Here, we present the essential code chunks necessary to implement the models. We focus on the Stan code for fitting the models M0 and M2. Listing 1 shows the standard hierarchical linear model M0. Stan syntax requires separate code blocks. The data block defines the types of the different variables in the data that is fed into the `stan` function available in the library `rstan`. The parameters block declares the types of the parameters involved in the model. The model block defines priors and describes how the data are assumed to be generated. The generated quantities block can be used to record the log likelihood (for model comparison) and for generating posterior predictive values.

Listings 2-4 show, in three parts, the code for the mixture model M2. New in this code (Listing 2) is the transformed parameters block. Here, we reparameterize some parameters in order to make the sampling more efficient (see Stan Development Team, 2016).

Listing 5 shows how the models can be fit using the R package `rstan`, and how the models can be compared using the `loo` package. The latter, an approximation of leave-one-out cross-validation, is a faster alternative to the K-fold cross-validation we used. Leave-one-out cross-validation compares the expected predictive performance of alternative models by subsetting the data into a training set (for estimating parameters) by excluding one observation. The difference between the predicted and observed held-out value can then be used to quantify model quality by successively holding out each observation. As in K-fold, the sum of the expected log pointwise predictive density, $\widehat{elpd}$, can be used as a measure of predictive accuracy.

The code for K-fold cross validation is slightly more involved as it requires fitting the data only on the data that is not held out (the training set). A new indicator variable `heldout` is therefore added to the data, which has value 0 when a data-point is from the training set, and value 1 when it is from the held-out set. The modifications needed to the model block of the code for models M0 and M2 are shown in Listing 6; here, we just use the indicator variable to estimate parameters using only the training data. Listing 7 shows code for creating the K subsets of the data and Listing 8 shows the code for carrying out model comparison.

```
data {
  int<lower=1> N;                  //number of data points
  real rt[N];                      //reading time
  real<lower=-1,upper=1> so[N];    //predictor
  int<lower=1> J;                  //number of subjects
  int<lower=1> K;                  //number of items
  int<lower=1, upper=J> subj[N];   //subject id
  int<lower=1, upper=K> item[N];   //item id
}

parameters {
  vector[2] beta;            //fixed intercept and slope
  vector[J] u;               //subject intercepts
  vector[K] w;               //item intercepts
  real<lower=0> sigma_e;     //error sd
  real<lower=0> sigma_u;     //subj sd
  real<lower=0> sigma_w;     //item sd
}

model {
  real mu;
  //priors
  beta ~ cauchy(0,2.5);
  sigma_e ~ cauchy(0,2.5);
  sigma_u ~ cauchy(0,2.5);
  sigma_w ~ cauchy(0,2.5);
  u ~ normal(0,sigma_u);    //subj random effects
  w ~ normal(0,sigma_w);    //item random effects
  // likelihood
  for (i in 1:N){
    mu = beta[1] + u[subj[i]] + w[item[i]] + beta[2]*so[i];
    rt[i] ~ lognormal(mu,sigma_e);
  }
}
generated quantities{
  real log_lik[N];
  vector[N] rt_tilde;
 for (i in 1:N)
    log_lik[i]=lognormal_lpdf(rt[i] | beta[1] +
                                u[subj[i]] + w[item[i]] + beta[2]*so[i],sigma_e);
for(i in 1:N){
  rt_tilde[i] = lognormal_rng(beta[1] +
                                u[subj[i]] + w[item[i]] + beta[2]*so[i],sigma_e);
  }
}
```

Listing 1: Code for fitting the standard hierarchical linear model M0.

```
//Part 1:
data {
  int<lower=1> N;                  //number of data points
  real rt[N];                      //reading time
  int<lower=0,upper=1> SO[N];      //predictor, treatment contrasts
  int<lower=1> J;                  //number of subjects
  int<lower=1> K;                  //number of items
  int<lower=1, upper=J> subj[N];  //subject id
  int<lower=1, upper=K> item[N];  //item id
}

parameters {
  real<lower=0> beta;            // only one intercept for both conditions
  vector[J] u;               //subject intercepts
  vector[K] w;               //item intercepts
  real<lower=0> sigma;     //error sd
  real<lower=0> sigma_diff;
  real<lower=0> sigma_u;     //subj sd
  real<lower=0> sigma_w;     //item sd
  real<lower=0,upper=1> prob_sr; //probability of extreme values
  real<lower=0,upper=1> prob_or; //probability of extreme values
  real<lower=0> delta;
}
transformed parameters{
  // reparameterization
  real sigmap_e;
  real sigma_e;
  sigmap_e = sigma + sigma_diff;
  sigma_e = sigma - sigma_diff;
}
```

Listing 2: Part 1 of the code for fitting the hierarchical mixture model M2.

```
// Part 2:
model {
  //priors
  prob_sr ~ beta(1,1);
  prob_or ~ beta(1,1);
  delta ~ cauchy(0,2.5);
  beta ~ cauchy(0,2.5);
  sigma ~ cauchy(0,2.5);
  sigma_diff ~ normal(0,1);
  sigma_u ~ cauchy(0,2.5);
  sigma_w ~ cauchy(0,2.5);
  u ~ normal(0,sigma_u);     //subj random effects
  w ~ normal(0,sigma_w);     //item random effects
  // log likelihood
  for (i in 1:N){
      if(heldout[i]==0){
      if(SO[i]==1){
      target += log_sum_exp(log(prob_sr) + lognormal_lpdf(rt[i] | beta +
      u[subj[i]] + w[item[i]] + delta, sigmap_e), log1m(prob_sr) +
      lognormal_lpdf(rt[i] | beta + u[subj[i]] + w[item[i]], sigma_e) );
      }
      if(SO[i]==0){
      target += log_sum_exp(log(prob_or) + lognormal_lpdf(rt[i] | beta +
      u[subj[i]] + w[item[i]] + delta, sigmap_e), log1m(prob_or) +
      lognormal_lpdf(rt[i] | beta + u[subj[i]] + w[item[i]], sigma_e) );
      }
      }
  }
}
```

Listing 3: Part 2 of the code for fitting the hierarchical mixture model M2.

```
// Part 3:
generated quantities{
  vector[N] rt_tilde;
  real<lower=0,upper=1> reanalysis_sr;
  real<lower=0,upper=1> reanalysis_or;
  real log_lik[N];
  real diffprob;
  real beta2;
  beta2=beta+delta;
  diffprob=prob_sr-prob_or;
  // likelihood:
  for(i in 1:N){
     if(SO[i]==1){
      log_lik[i] = log_sum_exp(log(prob_sr) + lognormal_lpdf(rt[i] | beta +
      u[subj[i]] + w[item[i]] + delta, sigmap_e), log1m(prob_sr) +
      lognormal_lpdf(rt[i] | beta + u[subj[i]] + w[item[i]], sigma_e) );
      }
      if(SO[i]==0){
      log_lik[i] = log_sum_exp(log(prob_or) + lognormal_lpdf(rt[i] | beta +
      u[subj[i]] + w[item[i]] + delta, sigmap_e), log1m(prob_or) +
      lognormal_lpdf(rt[i] | beta + u[subj[i]] + w[item[i]], sigma_e) );
      }
  }
//posterior predictive values:
    for(i in 1:N){
  // SR:
  if(SO[i]==1){
         reanalysis_sr = bernoulli_rng(prob_sr);
         if(reanalysis_sr) {
         rt_tilde[i] =  lognormal_rng(beta +
        u[subj[i]] + w[item[i]] + delta, sigmap_e);
         } else {
            rt_tilde[i] = lognormal_rng(beta +
            u[subj[i]] + w[item[i]], sigma_e);
                }
         }
  // OR:
  else {
         reanalysis_or = bernoulli_rng(prob_or);
         if(reanalysis_or) {
         rt_tilde[i] =  lognormal_rng(beta +
         u[subj[i]] + w[item[i]] + delta, sigmap_e);
         } else {
            rt_tilde[i] = lognormal_rng(beta +
            u[subj[i]] + w[item[i]], sigma_e);
                }
       }
}
}
```

Listing 4: Part 3 of the code for fitting the hierarchical mixture model M2.

```
library(rstan)
m0 <- stan(file = "models/m0.stan",
                      data = headnoun.dat,
                      iter = 2000, chains = 4,
           refresh=0)

paramnames<-c("beta[1]","beta[2]","sigma_e",
              "sigma_u","sigma_w")
## summarize results:
(m0_smry<-print(m0,pars=paramnames))

m2 <- stan(file = "models/m2postpred.stan",
                  data = headnoun.dat,
                  iter =  2000,
                  chains = 4,control =
             list(adapt_delta = 0.999),
           refresh=0)

paramnames<-c("beta","delta","diffprob",
            "prob_sr","prob_or","sigmap_e",
            "sigma_e",
            "sigma_u","sigma_w")
(m2_smry<-print(m2,pars=paramnames))

#model comparison:
library(loo)
loglikm0<-extract_log_lik(m0)
loom0<-loo(loglikm0)
loglikm2<-extract_log_lik(m2)
loom2<-loo(loglikm2)
compare(loom0,loom2)
```

Listing 5: Example code showing how the Stan models can be fit to the data, and how model comparison can be carried out using an approximation of leave-one-out cross-validation.

```
// Modification to data block:
data {
 //... other declarations as before
 real<lower=0,upper=1> heldout[N];// 0 = not held out: training data
}
// Modifications to model block for M0:
 // likelihood
 for (i in 1:N){
  if(heldout[i]==0){
  mu = beta[1] + u[subj[i]] + w[item[i]] + beta[2]*so[i];
  rt[i] ~ lognormal(mu,sigma_e);
  }
 }
//Modifications to model block for M2:
// likelihood:
 for (i in 1:N){
   if(heldout[i]==0){
   if(SO[i]==1){
   target += log_sum_exp(log(prob_sr) + lognormal_lpdf(rt[i] | beta +
   u[subj[i]] + w[item[i]] + delta, sigmap_e), log1m(prob_sr) +
  lognormal_lpdf(rt[i] | beta + u[subj[i]] + w[item[i]], sigma_e) );
   }
   if(SO[i]==0){
   target += log_sum_exp(log(prob_or) + lognormal_lpdf(rt[i] | beta +
   u[subj[i]] + w[item[i]] + delta, sigmap_e), log1m(prob_or) +
   lognormal_lpdf(rt[i] | beta + u[subj[i]] + w[item[i]], sigma_e) );
   }
   }
 }
```

Listing 6: Modifications needed to the M0 and M2 code in order to carry out K-fold cross-validation.

```
# given data in the following format:
#> head(headnoun[,c(1,2,3,7,10,11)])
#    subj item     type   rt   so SO
#94     1   13  obj-ext 1561  0.5  0
#221    1    6 subj-ext  959 -0.5  1
#341    1    5  obj-ext  582  0.5  0
row.names(headnoun)<-1:dim(headnoun)[1]
headnoun$row<-row.names(headnoun)
K <- 10
d <- headnoun
G <- list()
for (i in 1:K) {
        G[[i]] <- sample_frac(group_by(d, subj),
                              (1/(K + 1 - i)))
        G[[i]]$k <- i
        d <<- anti_join(d, G[[i]],
                        by = c("subj", "item",
                               "type", "rt","so","SO"))
}
# We create a data-frame again:
dK <- bind_rows(G)
# We save the order of the dataframe
dK <- dK[order(dK$row), ]
ldata <- plyr::llply(1:K, function(i) {
        list(N = nrow(dK), rt = dK$rt,
             so = dK$so,
             SO = dK$SO,
            subj = as.numeric(as.factor(dK$subj)),
            J = length(unique(dK$subj)),
            item = as.numeric(as.factor(dK$item)),
            K = length(unique(dK$item)),
            heldout = ifelse(dK$k == i, 1, 0))
})
```

Listing 7: Code for generating K cross-validation sets.

```
## M0:
pointwisem0 <- list()
for(i in 1:10){
  hnxval.dat<-ldata[[i]]
  m0xval <- stan(file = "models/m0xval.stan",
                      data = hnxval.dat,
                      iter = 4000, chains = 4,
           refresh=0)
  m0xval<-extract(m0xval,pars="log_lik")
  loglik<-m0xval$log_lik
  hldout<-which(hnxval.dat$heldout==1)
  logmeans<-rep(NA,length(hldout))
  for(j in 1:length(hldout)){
     logmeans[j]<-log(mean(exp(loglik[,hldout[j]])))
  }
pointwisem0[[i]]<-logmeans
}
pointwisem0_flat<-Reduce(c,pointwisem0)
(elpdm0<-sum(pointwisem0_flat))
(elpdm0_se<-sqrt(length(pointwisem0_flat)*
                   var(pointwisem0_flat)))
## M2:
pointwisem4 <- list()
for(i in 1:10){
  hnxval.dat<-ldata[[i]]
  m2xval <- stan(file = "models/m2postpredxval.stan",
                      data = hnxval.dat,
                      iter = 4000, chains = 4,
           refresh=0)
  m4xval<-extract(m2xval,pars="log_lik")
  loglik<-m2xval$log_lik
  hldout<-which(hnxval.dat$heldout==1)
  logmeans<-rep(NA,length(hldout))
  for(j in 1:length(hldout)){
     logmeans[j]<-log(mean(exp(loglik[,hldout[j]])))
  }
pointwisem2[[i]]<-logmeans
}
pointwisem2_flat<-Reduce(c,pointwisem2)
elpdm2<-sum(pointwisem2_flat)
elpdm2_se<-sqrt(length(pointwisem2_flat)*
                  var(pointwisem2_flat))
## k-fold cv:
(kfoldelpddiff<-elpdm2-elpdm0)
(kfoldelpdse<-sqrt(length(pointwisem0_flat)*
                     var(pointwisem0_flat-
                           pointwisem2_flat)))
```

Listing 8: Code for carrying out the cross-validation.