A Regularization Method for Linking Brain and Behavior

Inhan Kang, Woojong Yi, and Brandon M. Turner

The Ohio State University

Abstract

In a world of big data and computational resources, there has been a growing interest in further validating computational models of decision making by subjecting them to more rigorous constraints. One prominent area of study is model-based cognitive neuroscience, where measures of neural activity are explained and interpreted through the lens of a cognitive model. Although some early work has developed the statistical framework for exploiting the covariation between brain and behavior through factor analysis linking functions, current methods are still far from providing parsimonious accounts of high-dimensional (e.g., voxel-level) data. In this article, we contribute to this endeavor by investigating the fidelity of regularization methods such as the Lasso. Here, a combination of local and global penalty terms are applied to pressure elements of the factor loading matrix toward zero, reducing the false alarm rate. Such penalties facilitate the emergence of parsimonious network structure in the study of neural activation, giving way to clearer interpretations of high-dimensional data. We show through a set of three simulation studies and one application to real data that the Lasso can be an effective regularization method in the context of linking complex patterns of brain data to theoretical explanations of decisions. Although our analyses are specific to linking brain to behavior, the structure of the model is invariant to the type of high-dimensional data under investigation.

Keywords: Regularization, Lasso, model-based cognitive neuroscience, diffusion decision model

1 Introduction

The field of cognitive science is faced with two primary options for studying how experimentally-derived variables are related to the dynamics of cognitive processes. In the first approach, carefully constructed experimental designs map levels of an independent variable onto changes in brain activation, as measured through functional magnetic resonance imaging (fMRI), the electroencephalogram (EEG), or other modalities. Systematic changes in brain measurements as a function of important stimulus properties are meant to substantiate claims about the functional role of the associated brain regions. In the second approach, cognitive operations are abstracted away and treated as a set of statistical or mathematical processes, whose underlying dynamics are controlled by latent parameters. Once fit to data, the parameter estimates can be compared across the levels of the independent variable, where changes in the parameters are taken as an indicator that the associated cognitive processes have changed.

Both approaches have considerable strengths. In the first approach, identifying subcomponents of the organ housing mental operations that correspond to specific properties of stimuli facilitates localized interpretations of how the brain processes information. However, the localization of brain areas does not, by itself, permit interpretations of the functional role of those brain areas. There are potentially many reasons why a brain region would respond to a particular stimulus in a particular context. In the second approach, abstracting away complicated neural dynamics allows researchers to focus on high-level interpretations of how mechanisms and processes change in response to changes in the experimental design (e.g., instruction, stimuli). However, one can argue that the power of abstraction is also a curse: because the model is not sufficiently specified to reflect the biology of the system, one lacks the constraint that would permit an analysis of the associations between the mechanisms of the model and the supposed origins of those mechanisms in the brain.

Given the strengths and limitations of each approach, a growing number of researchers

have advocated for the advancement of cognitive neuroscience by blending cognitive models with neurophysiology. The goal is to use the abstractions provided by computational models of cognition to "steer" the interpretation of brain function. Although there are now many different ways of linking brain dynamics to model parameters (Turner, Forstmann, Love, Palmeri, & Maanen, 2017; Forstmann & Wagenmakers, 2015), in this article, we will focus on the joint modeling framework because it most naturally lends itself to the application of dimensionality reduction techniques, which is the primary focus of our research. In the joint modeling framework, fluctuations in neural data are statistically mapped to fluctuations in the parameters of a cognitive model. There are many types of statistical or mathematical maps that can be formed (see Turner, Palestro, Miletić, & Forstmann, 2019, for a review), and typically the map follows a parametric form (Turner, Forstmann, & Steyvers, 2019), although nonparametric functional forms are also possible (Bahg, Evans, Galdo, & Turner, in press). Predominantly, the probabilistic map that is used is a multivariate normal distribution (Turner, Forstmann, et al., 2013; Turner, van Maanen, & Forstmann, 2015; Turner, Rodriguez, Norcia, McClure, & Steyvers, 2016; Turner, Wang, & Merkle, 2017; Palestro et al., 2018). The multivariate normal distribution is a convenient choice because it allows every brain region to be associated with every model parameter in a pairwise fashion through the covariance matrix. However, as one might expect, such exhaustive associative techniques become computationally prohibitive as the number of brain areas increases, as it would when investigating activity at the voxel rather than region level.

A first attempt at reducing the dimensionality of probabilistic linking functions was considered in Turner, Wang, and Merkle (2017), where the covariance matrix between brain regions and model parameters was decomposed through a factor analysis linking function. This reduction technique dramatically altered the scalability of the joint models they investigated, such that increases in the number of brain regions had only a linear effect on the complexity of the covariance matrix, compared to a quadratic effect in

the standard approach. Despite this advancement, the linking functions used in Turner, Wang, and Merkle (2017) are most appropriate for problems of a confirmatory nature; that is, they are best suited for modeling brain-behavior dynamics when the pattern of factor loadings can be roughly prescribed prior to the analysis. However, in many cases, because we do not know the set of brain regions that will connect to each cognitive mechanism, we end up estimating every possible factor loading. What is needed is a technique that will allow us to unveil a simple structure of the links between brain and behavior at an affordable cost.

The goal of this article is to develop and apply dimensionality reduction techniques to factor analysis linking functions, with applications focused on the types of linking functions used in cognitive neuroscience. The method we focus on is the Lasso (Least Absolute Shrinkage and Selection Operator), which pushes weak associations between cognitive model parameters and brain regions toward zero, and allows strong associations to remain strong. We show, through three simulation studies and one application to real data, that such a regularization method allows for simple linking structures to emerge, while preserving the essential patterns of factor loadings induced in the simulations.

The outline of this article is as follows. First, we review regularization methods, and discuss the many ways in which the Lasso has been applied. We then discuss the effects that the Lasso can have on the estimated factor loadings, namely that its *shrinkage* effect can detect and remove small and unimportant loadings and accentuate the specific pattern of factor loadings. Next, we discuss the factor analysis linking function and the cognitive model used in Turner, Wang, and Merkle (2017), and then discuss how the Bayesian Lasso can be applied to such a model. We then explore the utility of the Lasso technique in a set of three simulation studies. Here, we compare a model that uses no Lasso to one that does in a variety of factor-loading structures ranging from simple to complex. Finally, we apply the Lasso method to data from a real experiment and show that simpler structures are found when using the Lasso that may facilitate clearer interpretations of

the links between brain and behavior.

2 Review of regularization methods

In this section, we review relevant statistical regularization methods while focusing our presentation on the Lasso method in two different contexts: univariate versus multivariate analyses, and frequentist versus Bayesian approaches.

2.1 Lasso in linear regression literature

The Lasso was first proposed by Tibshirani (1996) as a regularization method for a linear regression model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}$ is an $(N \times 1)$ vector of observations of a single dependent variable, $\boldsymbol{X}$ is an $(N \times (p+1))$ matrix of independent variables, $\boldsymbol{\beta}$ is a $((p+1)\times 1)$ vector of regression coefficients, and $\boldsymbol{\epsilon}$ is an $(N \times 1)$ vector of residuals assumed to follow a normal distribution with mean zero and variance σ^2. When using either ordinary least squares (OLS) or maximum likelihood estimation (MLE), parameters such as $\boldsymbol{\beta}$ and σ^2 are estimated by minimizing the following objective function, known as the sum of squared errors (SSE):

$$S(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{1}$$

Under some assumptions, Equation 1 can be solved analytically to produce estimates for the regression coefficients[1]:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

In regularization methods – including the Lasso – the objective function in Equation 1 is modified to

$$S(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \kappa||\boldsymbol{\beta}||_r^r, \tag{2}$$

[1]Note that this is the estimate without any constraint on the parameter space.

where an additional term

$$||\boldsymbol{\beta}||_r = \left(\sum_{j=1}^{p} |\beta_j|^r \right)^{\frac{1}{r}}$$

penalizes the SSE, based on the number and magnitudes of coefficients in the model. The intuition is that although adding more coefficients to a regression model can decrease the SSE, the decrease must be larger than the increase in the penalization term on the right to justify the increase in the number of parameters. To allow flexibility in the regularization, the penalty term is further scaled by the parameter κ. When $\kappa = 0$, no penalty is applied, and so the estimates would be equivalent to the OLS (or MLE) estimates. As κ increases, the penalty term has a larger shrinkage effect on the parameter estimates, and in some regularization methods, this yields fewer large coefficients. Because the value of the SSE depends on the data, it is difficult to specify the penalty parameter κ a priori. In practice, κ is tuned by a cross-validation (e.g., leave-one-out cross validation) procedure to reduce generalization error. For this reason, the parameter is also called the "tuning" parameter.

The penalty term can be interpreted as applying a constraint on the coefficient parameter space. The penalty term is defined as a r-norm of regression coefficients, which defines a subspace within the parameter space. Different values of r correspond to the restricted regions of different shapes. As examples, when setting $r = 1$, the Lasso regression is applied (Tibshirani, 1996), whereas if $r = 2$, the Ridge regression is applied (Hoerl & Kennard, 1970b, 1970a). Figure 1 illustrates how these two different settings of r adjust the OLS estimates. The left-most panel shows a standard estimation problem for two coefficients, β_1 (x-axis) and β_2 (y-axis). The contours show the shape of the OLS objective function, with the best estimate appearing in the center. The middle and left columns illustrate the effects of the Ridge and Lasso penalization terms, where the top row shows the influence of only the penalty term, and the bottom row shows the same penalty terms in the context of the OLS problem on the left. In the Lasso regression, the shape of this constraint is a diamond (right panel), whereas in the Ridge regression, the shape is circular. In each of the joint component plots (bottom middle and right), the

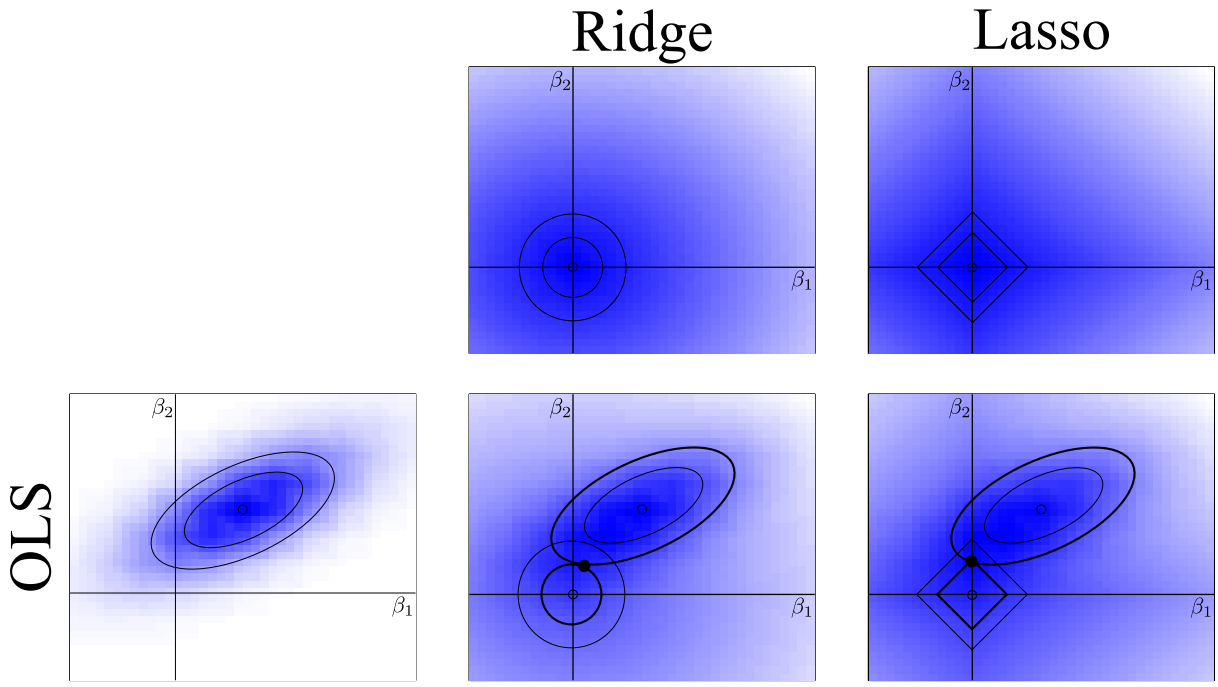

Figure 1: **Constraints imposed by regularization**. The left-most panel illustrates a parameter estimation problem for two parameter coefficients (axes), and the shape of the ordinary least squares solution (OLS) is represented as contours. The next columns illustrate the influence of regularization methods on the OLS, where the marginal (top row) and joint (bottom row) components show the Ridge (second column) and Lasso (third column) terms, respectively. Within the joint component plots, the black dot shows the best estimate of the model coefficients when regularization is applied.

best parameter estimate is represented as the black dot. In the Ridge regression, the location of the estimate would be any point such that the two circles meet, whereas in the Lasso regression, the estimate is likely to be found at the corner of the restricted region because the region is diamond-shaped. In this example, the Lasso estimate of the second regression coefficient β_2 is "shrunk" to zero (i.e., the vertical axis), effectively removing it from the resulting model. By removing unnecessary variables – that is, variables that do not contribute substantially to the model's fit to data – the Lasso ensures that the most parsimonious model is procured for a given phenomenon of interest. In this way, regularization methods simultaneously provide parameter estimates, selection among variables, and model selection.

Another strength of regularization methods is that their resulting estimates have smaller mean squared errors (MSE) and prediction errors (PE) (Tibshirani, 1996; Friedman, Hastie, & Tibshirani, 2001). Suppose $\hat{\beta}$ is an estimate of a regression coefficient β. The MSE of $\hat{\beta}$ is defined as

$$\text{MSE}(\hat{\beta}) = \text{E}[(\hat{\beta} - \beta)^2] = \text{Var}(\hat{\beta}) + (\text{E}(\hat{\beta}) - \beta)^2. \tag{3}$$

Here, MSE is defined as a measure of the distance between the estimate and the true coefficient value, which can be decomposed in Equation 3 to show that the MSE represents unbiasedness and stability of an estimator.

The OLS estimator of a regression coefficient is known as the best linear unbiased estimator (BLUE; Ravishanker & Dey, 2001), which achieves the minimum variance among the unbiased estimators. Due to shrinkage, the estimators of the regularization methods deviate from the OLS estimator, and thus they have a bias (i.e., they underestimate coefficients). However, by allowing this small bias, the shrinkage estimators can reduce variance to a large extent and achieve a smaller MSE. A similar statement holds for the PE as it can be shown that $\text{PE}(\hat{\beta}) = \text{MSE}(\hat{\beta}) + \sigma^2$ (Tibshirani, 1996; Friedman et al., 2001). Hence, regularization methods can provide more stable estimators with lower MSE and PE.

Although Equation 2 uses only a single penalty parameter, different regularization methods may have more than one penalty parameters in the objective function. For example, Elastic Net regularization (Zou & Hastie, 2005) uses two penalty parameters, one of which corresponds to the Ridge penalty and the other corresponds to the Lasso penalty. The technical details and differences among different regularization methods are beyond the scope of this article, and so we refer the interested readers to van Erp, Oberski, and Mulder (2019). The regularization methods have been applied to other models such as multivariate regression (Li, Nan, & Zhu, 2015; Peng et al., 2010; Rothman, Levina, & Zhu, 2010; Price & Sherwood, 2018), factor analysis (Choi, Zou, & Oehlert, 2010; Hirose & Konishi, 2012; Hirose & Yamamoto, 2015; Jung & Takane, 2008; Ning & Georgiou, 2011),

structural equation modeling (Jacobucci, Grimm, & McArdle, 2016), and item response theory modeling (Houseman, Marsit, Karagas, & Ryan, 2007; Magis, Tuerlinckx, & Boeck, 2015).

Our final note about regularization methods is their Bayesian interpretation (Tibshirani, 1996; Park & Casella, 2008). Within the Bayesian framework, parameters can be constrained by specifying a priori information about them, and this specification comes in the form of a prior distribution on the model parameters. For example, the constraint enforced by the Lasso can be represented by the following Laplace prior:

$$f(\beta_j) = \frac{\kappa}{2}\exp\Big(- \kappa|\beta_j|\Big). \tag{4}$$

Based on this interpretation, the Lasso has been extended to Bayesian models including linear regression and latent variable modeling, which are reviewed in the following sections.

2.2 Bayesian Lasso

As Tibshirani (1996) pointed out, the regularization penalty imposed by the Lasso regression is related to a Laplace prior on the regression coefficients, and this relationship has motivated several implementations of the Lasso when doing Bayesian linear regression (Bae & Mallick, 2004; Figueiredo, 2003; Park & Casella, 2008; Yuan & Lin, 2006). A notable example is Park and Casella's (2008) method in which they proposed to use a conditional Laplace prior of the following form:

$$\pi(\boldsymbol{\beta}|\sigma^2) = \prod_{j=1}^{p} \frac{\kappa}{2\sqrt{\sigma^2}} e^{-\kappa|\beta_j|/\sqrt{\sigma^2}}. \tag{5}$$

Conditioning on the residual variance σ^2 plays an important role as it ensures that the posterior distribution of $\boldsymbol{\beta}$ is unimodal. When this is the case, the posterior will have a unique maximum, which facilitates clear interpretations of the coefficients. Based on the

normal scale mixture representation of the Laplace distribution, Park and Casella derived the hierarchical representation of the Bayesian Lasso regression model. Namely,

$$\begin{aligned}
\boldsymbol{y}|\mu, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2 &\sim N_n(\mu \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n), \\
\boldsymbol{\beta}|\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim N_p(\mathbf{0}_p, \sigma^2 \boldsymbol{D}_\tau), \quad \boldsymbol{D}_\tau = \text{diag}(\tau_1^2, \ldots, \tau_p^2), \text{ and} \\
\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim \pi(\sigma^2) d\sigma^2 \prod_{j=1}^{p} \frac{\kappa^2}{2} e^{\kappa^2 \tau_j^2/2} d\tau_j^2, \quad \sigma^2, \tau_1^2, \ldots, \tau_p^2 > 0.
\end{aligned} \tag{6}$$

Park and Casella also derived the full conditional distribution of the model parameters so that estimation can be done using Gibbs sampler. In the above expression, new parameters $\tau_1^2, \ldots, \tau_p^2$ are introduced to connect the regression coefficients $\boldsymbol{\beta}$ to the tuning parameter κ. As different regression coefficients have different values of τ_j^2, these parameters modulate the penalizing effect of κ on the corresponding regression coefficients. To find the best value of κ, one can again use cross-validation methods, but in the Bayesian context, such procedures will be remarkably more computationally demanding than in the frequentist case. As alternatives, Park and Casella (2008) proposed to use either empirical Bayesian methods or a gamma distribution hyperprior for κ^2. Imposing a gamma prior for κ^2 results in a conjugate posterior, meaning that the model parameters can be efficiently estimated using Gibbs sampling techniques.

2.3 Bayesian regularized latent variable modeling

Park and Casella (2008)'s hierarchical framework for the Bayesian Lasso and other regularization methods has been extended to latent variable models in a variety of applications (Feng, Wu, & Song, 2017a, 2017b; Guo, Zhu, Chow, & Ibrahim, 2012; Song, Lu, & Feng, 2014; Wang, Lane, Chakraborty, & Wood, 2013). For example, Guo et al. (2012) proposed the Bayesian Lasso for the semi-parametric structural equation model (SEM). The SEM consists of a measurement model (a factor analysis model for the measurement of latent variables) and a structural model (a set of regression models of latent variables

models) and Guo et al. applied Park and Casella's method to the structural model but not to the measurement model (*c.f.,* Lu, Chow, & Loken, 2016). Guo et al. used separate penalty parameters (i.e., multiple κ parameters) for endogenous independent variables and nonparametric functions of exogenous variables. Thus, the proposed method is closer to the group Lasso (Yuan & Lin, 2006) in which a pre-determined group of regression coefficients is penalized by a common penalty term and different groups are penalized by different penalty terms. Furthermore, Guo et al. allowed different penalty parameters for different endogenous latent variables so that regression coefficients for different dependent variables in the model are penalized by different penalty terms. This is different from the frequentist approach to multivariate regression and SEM in which the same penalty term is applied to all model equations for different dependent variables (Jacobucci et al., 2016; Li et al., 2015; Peng et al., 2010; Rothman et al., 2010; Price & Sherwood, 2018). Having different penalty terms for different groups of parameters can be useful when the groups should be penalized by different degrees. However, when Park and Casella's hierarchical representation of the Bayesian Lasso is applied, multiple penalty terms may not be required due to the effect of τ parameters, which will be described in detail in Section 2.4. In contrast to Guo et al.'s approach, the Bayesian Lasso will be applied to the measurement model with a single κ parameter in our method.

Similarly to Guo et al.'s proposal, Bayesian regularization methods have been extended to models with latent variables including semi-parametric SEM (the Elastic net and the fused Lasso, Wang et al., 2013), univariate and bivariate nonparametric functions of latent variables (the Lasso, Song et al., 2014), univariate ordinal regression (the adaptive Lasso, Feng et al., 2017a), and multivariate generalized latent variable models (the adaptive Lasso, Feng et al., 2017b). In all of these cases, different penalty terms have been used for different dependent variables.

2.4 Global and local shrinkage

In the hierarchical representation of the Bayesian Lasso, additional parameters, usually denoted as $\boldsymbol{\tau}$, are introduced. These parameters produce an interesting difference from the frequentist regularization methods in that they modulate the penalizing effect of the global penalty term $\boldsymbol{\kappa}$ on each of the regression coefficients. Polson and Scott (2011, 2012) studied a group of shrinkage priors with such parameterization, which they called global and local shrinkage priors. Under these priors, estimation of regression coefficients depends on both global and local shrinkage parameters. The global parameter controls a general magnitude of penalization on all regression coefficients within a model, just as the penalty parameter κ does in the frequentist regularization methods (e.g., Equations 2). The local parameters, each of which corresponds to a single regression coefficient, modulate the effect of global penalization on each coefficient. In other words, the magnitude of the shrinkage effect differs by coefficient when global and local shrinkage priors are used.

Ideally, small coefficients will be greatly penalized, and large coefficients will be only weakly penalized. This pattern will ensure that small coefficients will be removed from the model whereas the estimates for large coefficients will be less biased by the penalization terms. In this sense, the global and local priors are advantageous in that they find a sparse model while simultaneously estimating large coefficients with less bias compared to the frequentist methods. The same advantage might be manufactured in the frequentist method by allowing many tuning parameters (e.g., the group Lasso and the adaptive Lasso), but it would be more difficult to update all tuning parameters via cross-validation. By contrast, global and local shrinkage parameters are just a part of a Bayesian hierarchical model, and can be estimated via standard sampling methods when performing Bayesian estimation. Furthermore, many of the global and local shrinkage priors have conjugate posteriors, which facilitates estimation.

The Bayesian Lasso proposed by Park and Casella uses the same parameterization and enjoys the global-local shrinkage effect (Polson & Scott, 2011). Figure 2 illustrates how the

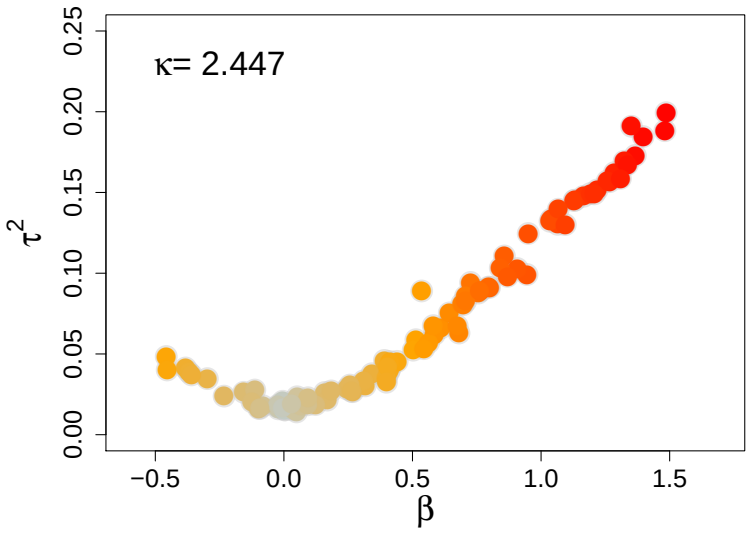

Figure 2: **Modulation of the Shrinkage Effect under a Global-Local Prior.** The Bayesian Lasso regression (Park & Casella, 2008) is fit to simulated data and the estimated regression coefficients β (x-axis) are shown along with their corresponding local penalty parameters τ^2 (y-axis). The global penalty parameter was estimated to be $\kappa = 2.447$.

Bayesian Lasso modulates shrinkage of the regression coefficients. Data were simulated with sample size of $N = 500$ and $p = 100$ covariates. We randomly sampled values for the regression coefficients from a uniform distribution on the interval $(-0.5, 1.5)$. Although the precise interval was arbitrarily selected, our main goal was to cover zero with a reasonable amount of overlap. We then constructed a dependent variable by multiplying the regression coefficients with the simulated data, and then adding random noise from a normal distribution with mean zero and standard deviation of 2. We then fit the Bayesian Lasso regression model to the simulated data.

Figure 2 shows the estimated regression coefficient β (x-axis) against the corresponding local penalty parameters τ (y-axis). The global penalty parameter was estimated to be $\kappa = 2.447$. However, its effect is modulated by different values of the local parameters. Figure 2 shows that the local penalty parameters are large for large coefficients, but are small for small coefficients. This particular pattern makes the posterior variances of the regression coefficients smaller as the coefficients are estimated to be nearer to zero

(see Equation 6). As a result, smaller coefficients will shrink to zero whereas only weak shrinking effects will be imposed on larger coefficients.

3 The factor analysis neural drift diffusion model

Turner, Wang, and Merkle (2017) proposed the factor analysis neural drift diffusion model (FA NDDM) as an extension of the previously proposed multivariate normal neural drift diffusion model (MVN NDDM). Both are considered joint models (Palestro et al., 2018) in which behavioral data $\boldsymbol{B}$ and neural data $\boldsymbol{N}$ are analyzed simultaneously by connecting the parameters of appropriate statistical or computational models. For example, one may assume a computational model with parameters $\boldsymbol{\theta}$ for the behavioral data, and a statistical model with parameters $\boldsymbol{\delta}$ for the neural data. In both the FA and MVN NDDMs, a standard statistical model is used to analyze the pattern of neural data, and a diffusion decision model (Ratcliff, 1978; Ratcliff & McKoon, 2008) is used to explain both choice and response time from a psychological experiment involving a simple two-choice perceptual task. To complete the model, the joint relationship between $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ must be specified. In both the FA and MVN NDDMs, a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is assumed to create a linking function for the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$, such that

$$(\boldsymbol{\theta}_i, \boldsymbol{\delta}_i) \sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where the subscript i indicates the i-th subject or i-th trial.

While the mean $\boldsymbol{\mu}$ is freely estimated in both of the FA and MVN NDDMs, the two models differ in how the covariance matrix $\boldsymbol{\Sigma}$ is structured. The MVN NDDM assumes a full rank covariance matrix $\boldsymbol{\Sigma}$, where each element is estimated when fit to data. As shown in Turner, Wang, and Merkle (2017), the complexity of such a model grows quadratically with increases in the number of neural features, which prohibits being able to fit the model to neural data at the voxel level, for example. By contrast, in the FA NDDM, the

covariance matrix ($\boldsymbol{\Sigma}$) is decomposed by a factor analysis model

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi},$$

where $\boldsymbol{\Lambda}$ is a factor loading matrix, $\boldsymbol{\Phi}$ is a factor variance-covariance matrix, and $\boldsymbol{\Psi}$ is a diagonal matrix of residual variances. Parameters for these matrices are freely estimated except for those on which constraints are imposed. With the factor analytic structure, elements within the matrix $\boldsymbol{\Sigma}$ are not freely estimated as they are now functions of the factor loadings, factor variances and covariances, and residual variances. The dimensions of the matrices and the multivariate normal distribution in the NDDM models are determined by the number of behavioral and neural parameters.

Mathematically, the FA NDDM can be fully expressed as

$$\begin{gathered}
\boldsymbol{b}_i | a, t_i, z_i, d_i \sim \textit{Diffusion}(a, t_i, z_i, d_i), \\
(\log(t_i), \text{logit}(\frac{z_i}{a}), d_i, \boldsymbol{n}_i^T)^T \sim MVN_{p+q}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}, \\
\text{where } \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{I}_q \\ \boldsymbol{\Lambda}^*_{p\times q} \end{bmatrix} \text{ and } \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{0}_q & \boldsymbol{0}_{q\times p} \\ \boldsymbol{0}_{p\times q} & \boldsymbol{\Psi}^*_p \end{bmatrix}.
\end{gathered} \tag{7}$$

where $\boldsymbol{b}_i$ is a (2×1) vector of choice and RT in the i-th trial, $\boldsymbol{n}_i$ is a $(p \times 1)$ vector of all p neural covariate in the i-th trial, $\boldsymbol{I}_q$ is the q-dimensional identity matrix and $\boldsymbol{0}_q$ and $\boldsymbol{0}_{q\times p}$ are matrices of zeroes of size $(q \times q)$ and $(q \times p)$, respectively. '*Diffusion*' indicates the probability density function of the Wiener diffusion process, which is a popular model in psychology for explaining the joint distribution of choice and response time (Diffusion Decision Model or DDM; Ratcliff, 1978; Ratcliff & McKoon, 2008; Wabersich & Vandekerckhove, 2014). For example, subjects may be asked to decide which direction most of the dots are moving within a cloud of randomly moving dots. The Wiener process model assumes that, when a stimulus is presented, a subject starts to accumulate evidence at a starting point z_i at a mean rate of d_i over time. This evidence continues to accumulate

until it reaches one of two boundaries, each of which represents a choice option (e.g., "left" and "right"). At this time, a choice C_i is made. The separation between these two boundaries is represented as a. The time at which the process terminates determines a decision time (DT_i). The model also assumes the presence of nondecision processes such as visual encoding time and motor response time, which are captured by the nondecision time parameter t_i. Hence, the model's predicted response time is $RT_i = DT_i + t_i$. The parameters represent cognitive components of decision-making processes. The drift rate d_i represents the quality of evidence accumulation and the decision threshold a measures the amount of information required to make a decision. Also, the starting point z_i is a measure of an initial bias toward one of the two choice options. Nondecision processes can be collectively represented by the nondecision time t_i.

Because the diffusion model is used as the behavioral submodel within the FA NDDM, the behavioral parameters $\boldsymbol{\theta}$ include all the diffusion model parameters across trials, such that $\boldsymbol{\theta} = (a, \boldsymbol{t}, \boldsymbol{z}, \boldsymbol{d}))$, where $\boldsymbol{t}$, $\boldsymbol{z}$, and $\boldsymbol{d}$ are the vectors of nondecision times, starting points, and drift rates over all trials, respectively. Because the nondecision time cannot be negative, and the starting point is bounded between 0 and a, they are first transformed to match the support of the multivariate normal distribution and then associated with the neural part of the FA NDDM via the overarching linking function. While any neural model can be used for the neural data, it is also possible to directly link the neural features to behavioral parameters so that neural sources can be mapped directly to components of a cognitive process (e.g., the diffusion model parameters in this example). We adopt the latter approach and in this case, the neural parameters $\boldsymbol{\delta}$ are just set equal to the neural features (i.e., $\boldsymbol{\delta} = \boldsymbol{N}$). With this specification, the number of latent variables in the linking function for the i-th trial is $q = 3$, and the number of neural sources p is equal to the number of neural features.

By imposing the factor structure on the covariance matrix, the FA NDDM provides a remarkable reduction in the total number of parameters that needs to be estimated, going

from $\frac{1}{2}(p+q)(p+q+1)$ in the MVN structure to $pq + \frac{1}{2}q(q+1) + p = (p + \frac{1}{2}q)(q+1)$ in the FA structure. Because the number of manifest variables p is usually large (e.g., 32, 64, or 128 if the neural data are from EEG recording) while the number of factors q is much smaller (e.g., 3 in both the previous NDDM approaches in Turner, Wang, & Merkle, 2017), the total number of parameters is greatly reduced in the FA NDDM.

A further strength of the FA NDDM is that it allows for a systematic study of how a brain network is related to the factors employed in the model. Factor loadings represent the strength of linkage between corresponding factors and manifest variables. If a neural feature has a high factor loading for one factor but not for the others, then it can be concluded that the brain region measured by that neural feature is largely related to the factor with the high loading. In the FA NDDM, the latent variables are single-trial diffusion model parameters (i.e., drift rate, starting point, and nondecision time). Thus, the FA NDDM promotes the identification of which brain regions are related to cognitive constructs such as the quality of evidence accumulation, a state of initial activation, and nondecision processes such as stimulus encoding and response production, via factor loadings. Furthermore, if some of the brain regions are collectively related to a given factor, we can consider it model-based evidence for the functional connectivity of those regions. For example, if several neural features load highly on the drift rate factor, we could conclude that those features are components in the brain network engaged in evidence accumulation during decision-making tasks. Although both the MVN NDDM and FA NDDM can be used to identify brain networks, the latter method can provide a simpler way to investigate networks through the factor structure whereas the former requires that all variances and covariances be estimated before a network analysis can be performed.

Equation 7 shows that the FA NDDM is a hybrid of confirmatory and exploratory factor analysis. On the one hand, it is confirmatory because the factors in the model are fixed to be the theoretical components of the DDM (i.e., nondecision time, bias, and drift rate). Given that the first three rows of the factor loading matrix comprise the behavioral model

parameters, Turner, Wang, and Merkle imposed the constraint that the first $q = 3$ rows of the factor loading matrix should be fixed to an identity matrix ($\boldsymbol{I}_q$), and that the residual variances for the first three rows should be zero ($\mathbf{0}_q$ on the upper-left side of $\boldsymbol{\Psi}$). On the other hand, the model is exploratory in the sense that all other factor loadings linking the factors and the manifest variables (i.e., neural features) can be freely estimated ($\boldsymbol{\Lambda}^*_{p\times q}$). In addition, the corresponding factor variances, covariances, and residual variances in $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are also freely estimated. This approach differs from other hybrid approaches (e.g., Lu et al., 2016) in which only minimal identification constraints are imposed on a CFA factor loading matrix to identify factors, while others are estimated freely as in EFA. Instead, the factors used within the FA NDDM are presupposed to be mechanisms assumed by the DDM and well constrained by choice and response time data.

3.1 Factor loading constraints

As the FA NDDM is typically fit in a Bayesian framework, an issue arises during posterior sampling called the "sign-switching" problem. Because a factor does not have a specific scale, the sign of the factor loading is also arbitrary. In the estimated results, the signs of the estimated factor loadings can be reversed in a column-wise manner because reversing the signs of the factor scores and those of the corresponding factor loadings does not change the model fit result. However, the instability in the signs of the factor loadings can be problematic when a sampling method is applied to estimate posterior distributions because the signs of the factor loadings can oscillate throughout the sampling process; for example, if the true posterior mean is 1, chains may oscillate between -1 and 1 or some of the chains can be centered at 1 while the other can be centered at -1. Thus, sometimes an estimated posterior distribution of a specific factor loading can be bimodal even when the true posterior distribution is unimodal.

When performing exploratory factor analyses with q factors in the Bayesian framework, it has been shown that constraining the signs of q diagonal elements in the factor

loading matrix to be positive can address the sign-switching problem, and fixing at least $q(q-1)/2$ upper diagonal loadings to zero can resolve the rotational indeterminacy of latent variables (Geweke & Zhou, 1996; Erosheva & Curtis, 2017).

In the FA NDDM, the first q rows of the factor loading matrix are set to be an identity matrix (Equation 7) and it may appear as though this constraint addresses both of the sign-switching problem and the rotational indeterminacy. However, it can be deduced that the identity matrix in the factor loading structure only addresses the rotational indeterminacy and a solution to the sign-switching problem is required for the estimation of the FA NDDM. From Equation 7, it can be derived that:

$$\left(\log(t_i), \text{logit}\left(\frac{z_i}{a}\right), d_i\right)^T = \boldsymbol{\mu}_{1:q} + \boldsymbol{I}_q\boldsymbol{\omega}_i = \boldsymbol{\mu}_{1:q} + \boldsymbol{\omega}_i \tag{8}$$

where $\boldsymbol{\mu}_{1:q}$ represents the first q elements of the mean vector of the multivariate normal distribution with $q = 3$ for the specification of the FA NDDM in the current study, and $\boldsymbol{\omega}_i$ is a vector of factor scores for the i-th trial (see also the Appendix of Turner, Wang, & Merkle, 2017). This formula shows that the latent variables in the FA NDDM are set equal to the mean-centered and transformed single-trial diffusion model parameters. The main role of this formula is to link the behavioral model parameters to the factor analysis model implemented for the neural data. By defining the meaning of the latent variables, the formula successfully removes the rotational indeterminacy issue.

Little can be said about the sign-switching problem with only Equation 8. If the single-trial diffusion model parameters and their means can be fully determined by the diffusion model fit to the behavioral data, the signs of $\boldsymbol{\omega}_i$ can also be determined. While the means of transformed nondecision time, transformed starting point, and drift rate can be estimated from the choice and RT data across all the trials, choice and RT for a single trial are not sufficient to fully constrain the single-trial parameters. In fact, the estimation and measurement of $\boldsymbol{\omega}_i$ rely heavily on the factor analysis model applied to the neural data,

which is specified by the bottom p rows of the factor loading matrix and the bottom-right $(p \times p)$ submatrix of the residual variance matrix ($\mathbf{\Lambda}^*_{p\times q}$ and $\mathbf{\Psi}^*_p$ in Equation 7). As all factor loadings of neural features on q latent variables should be estimated, the estimation of these loadings can suffer from the sign-switching problem and some appropriate solution should be applied.

Turner, Wang, and Merkle (2017) proposed two solutions for the FA NDDM to address the sign-switching problem. The first solution is a column-wise solution in which we specify the sign of some pre-determined factor loadings (Ghosh & Dunson, 2009). For example, if one can assume that the j-th manifest variable is positively related to the k-th factor, we may take the constraint $\lambda^*_{j,k} > 0$ as a reference for the k-th column of the factor loading matrix. During a Bayesian sampling procedure, samples of $\lambda^*_{j,k}$ are then monitored such that negative posterior draws result in a reversal of the sign of the factor loadings in the k-th column of the matrix. Thus, given a positive assertion on $\lambda^*_{j,k}$, the sampling algorithm would run with the following constraint:

$$\lambda^*_{1:p,k} = \begin{cases} \lambda^*_{1:p,k} & \text{if } \lambda^*_{j,k} > 0, \\ -\lambda^*_{1:p,k} & \text{if } \lambda^*_{j,k} < 0. \end{cases} \tag{9}$$

To avoid the sign-switching problem while estimating the joint posterior distribution, q constraints should be applied, one for each factor. In Turner, Wang, and Merkle (2017), this solution is combined with giving the rescaled Beta prior $\lambda^*_{j,k} \sim \text{Beta}(1, 1, -1, 1)$ to all the factor loadings (except for those in the first three rows) where the first two hyperparameters are shape parameters and the last two hyperparameters are lower and upper bounds of the distribution. Other priors that are well-suited to represent the support of the factor loading values can also be used within this solution.

The second solution is an element-wise solution in which the sign of all the factor loadings are collectively constrained. For example, by imposing $\lambda^*_{j,k} \sim \text{Beta}(1, 1, 0, 1)$ for

all j and k, all the factor loadings are constrained to have positive signs. Although this is an unlikely case for many real-world problems, it is worth considering as an option for estimation of the joint posterior distribution. It is also possible to use different priors for different factor loadings. For example, if it is reasonable to assume that $\lambda^*_{j,k} > 0$ while $\lambda^*_{j,l} < 0$, we may give $\lambda^*_{j,k} \sim \text{Beta}(1, 1, 0, 1)$ and $\lambda^*_{j,l} \sim \text{Beta}(1, 1, -1, 0)$, respectively, as in Turner, Wang, and Merkle's application. However, this is not plausible without a strong theoretical background to pre-determine the signs of all factor loadings.

3.2 Estimating the FA NDDM using conjugate priors

Although not covered in Turner, Wang, and Merkle (2017), it is also possible to implement conjugate priors on the factor analysis parameters in the FA NDDM to estimate the joint posterior distribution. For example, we can specify the following prior distributions as used in Hoff (2009); Song and Lee (2012):

$$\begin{aligned}
\boldsymbol{\mu} &\sim MVN_{p+q}(\boldsymbol{\upsilon}_0, \boldsymbol{\Delta}_0), \\
\psi_{\epsilon j}^{-1} &\sim Gamma(\alpha_{0\epsilon j},\ \beta_{0\epsilon j}), \quad j = (q+1), (q+2), \ldots, (q+p), \\
\boldsymbol{\Lambda}_j |\ \psi_{\epsilon j} &\sim MVN_q(\boldsymbol{\Lambda}_{0j},\ \psi_{\epsilon j}\boldsymbol{H}_{0j}), \text{ and} \\
\boldsymbol{\Phi} &\sim IW_q(\boldsymbol{R}_0^{-1},\ \rho_0)
\end{aligned} \tag{10}$$

where p is the number of manifest variables (neural features in the current application), $\boldsymbol{\Lambda}_j$ is the j-th row of the factor loading matrix, $\psi_{\epsilon j}$ is the residual term in the j-th measurement equation, and Φ is the factor covariance matrix. The quantities with the subscript 0 ($\boldsymbol{\upsilon}_0$, $\alpha_{0\epsilon j}$, $\beta_{0\epsilon j}$, $\boldsymbol{\Lambda}_{0j}$, ρ_0, and positive definite matrices $\boldsymbol{\Delta}_0$, $\boldsymbol{H}_{0j}$, and $\boldsymbol{R}_0$) are hyperparameters for the prior distributions. Given these priors, the conditional posterior distributions

are:

$$
\begin{aligned}
&\boldsymbol{\mu} \mid \boldsymbol{Y}, \boldsymbol{\Lambda}, \boldsymbol{\Phi}, \boldsymbol{\Psi} \sim MVN_{p+q}(\boldsymbol{\upsilon}_n, \boldsymbol{\Delta}_n), \\
&\quad \boldsymbol{\upsilon}_n = \boldsymbol{\Delta}_n(\boldsymbol{\Delta}_0^{-1}\boldsymbol{\upsilon}_0 + n\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{y}}), \quad \boldsymbol{\Delta}_n = (\boldsymbol{\Delta}_0^{-1} + n\boldsymbol{\Sigma}^{-1})^{-1}, \\
&\psi_{\epsilon j}^{-1} \mid \boldsymbol{y}_j, \boldsymbol{\Omega} \sim Gamma(\alpha_{n\epsilon j}, \beta_{n\epsilon j}), \quad j = (q+1), (q+2), \ldots, (q+p), \\
&\quad \alpha_{n\epsilon j} = \alpha_{0\epsilon j} + \frac{n}{2}, \quad \beta_{n\epsilon j} = \beta_{0\epsilon j} + \frac{1}{2}(\boldsymbol{y}_j^T\boldsymbol{y}_j + \boldsymbol{\Lambda}_{0j}^T\boldsymbol{H}_{0j}^{-1}\boldsymbol{\Lambda}_{0j} - \boldsymbol{\Lambda}_{nj}^T\boldsymbol{H}_{nj}^{-1}\boldsymbol{\Lambda}_{nj}), \qquad (11) \\
&\boldsymbol{\Lambda}_j \mid \boldsymbol{y}_j, \boldsymbol{\Omega}, \psi_{\epsilon j} \sim MVN_q(\boldsymbol{\Lambda}_{nj},\ \psi_{\epsilon j}\boldsymbol{H}_{nj}), \\
&\quad \boldsymbol{\Lambda}_{nj} = \boldsymbol{H}_{nj}(\boldsymbol{H}_{0j}^{-1}\boldsymbol{\Lambda}_{0j} + \boldsymbol{\Omega}\boldsymbol{y_j}), \quad \boldsymbol{H}_{nj} = (\boldsymbol{H}_{0j}^{-1} + \boldsymbol{\Omega}\boldsymbol{\Omega}^T)^{-1}, \text{ and} \\
&\boldsymbol{\Phi} \mid \boldsymbol{\Omega} \sim IW_q(\boldsymbol{R}_0^{-1} + \boldsymbol{\Omega}\boldsymbol{\Omega}^T,\ n + \rho_0).
\end{aligned}
$$

where n is the number of observations, $\boldsymbol{y}_j$ be a vector of all n observations of the j-th manifest variable, and $\boldsymbol{Y}$ is a matrix of all observations. $\boldsymbol{\Omega}$ is a matrix of all factor scores and $\boldsymbol{\Sigma}$ is the implied covariance matrix of the factor analysis model, calculated from the posterior samples during a sampling procedure (Equation 7). The quantities with the subscript n ($\boldsymbol{\upsilon}_n$, $\alpha_{n\epsilon j}$, $\beta_{n\epsilon j}$, $\boldsymbol{\Lambda}_{nj}$, and positive definite matrices $\boldsymbol{\Delta}_n$ and $\boldsymbol{H}_{nj}$) represent the parameters for the posterior distributions updated by the n observations of the data.

These conditional posterior can be applied to Equation 7. Due to the constraints to link the single-trial diffusion model parameters and the factor analysis model, the conditional posterior distributions of factor loadings should be applied to $\boldsymbol{\Lambda}_{p\times q}^*$, the bottom p rows of the whole factor loading matrix in Equation 7, because the first q rows should be an identity matrix as constraints. Similarly, the conditional posterior distributions of residual variances should be applied to $\boldsymbol{\Psi}_p^*$, the bottom left $p \times p$ submatrix of the whole residual variance matrix. For these two sets of posterior distributions, $\boldsymbol{y}_j = \boldsymbol{N}_j$ in Equation 11 where $\boldsymbol{N}_j$ is a vector of the j-th neural covariate containing all n observations (over trials or subjects). To update $\boldsymbol{\mu}$ for all $p + q$ variates in Equation 7, $\boldsymbol{Y}$ in Equation 11 should be replaced by a matrix of $(\log(t_i), \text{logit}(\frac{z_i}{a}), d_i, \boldsymbol{n}_i^T)^T$ containing all observations over $i = 1, \cdots, n$.

A prior for $\boldsymbol{\Omega}$ needs not be specified because the factor scores are defined based on the single-trial diffusion model parameters. As shown in Equation 8, they are set to be equal to the transformed single-trial diffusion model parameters centered by the corresponding means in $\boldsymbol{\mu}$ (Appendix in Turner, Wang, & Merkle, 2017). Thus, the factors are measured by neural data (the factor analysis) and constrained again by the behavioral data (the diffusion model). This is another strength of the joint modeling framework in that an integrative analysis of multiple data modalities (e.g., behavioral and neural data) provides more information on the latent construct of interest. Although the conjugate approach reduces the computational burden of posterior sampling to a large extent, it does not, by itself, solve the sign-switching problem. This method should be accompanied by an appropriate sign-switching solution. Equation 9 can be easily implemented during the conjugate posterior sampling.

The diffusion model parameters (a, t_i, z_i, and d_i) do not have conjugate priors and thus their posterior distributions should be estimated by Monte Carlo sampling methods. The prior and likelihood of these parameters are from the multivariate normal distribution implemented as the overarching linking function and the diffusion model likelihood, respectively, as stated in Equation 7. Due to potentially high correlations among the parameters in the behavioral model, we recommend techniques that can manage said parameter dependencies such as Differential Evolution Markov Chain Monte Carlo (DE-MCMC; Ter Braak, 2006; Turner, Sederberg, Brown, & Steyvers, 2013). The central idea of this method is to allow sets of chains to approximate the shape of the target distribution (e.g., the posterior) by calculating differences among the chains. These differences are used to create a vector that defines movement within the parameter space, from which proposal parameter values can be generated. For example, one can calculate the difference between two randomly selected chains, and scale this difference by a tuning parameter $\gamma \sim U(0.5, 1)$. As the sampling process progresses, this difference vector naturally resembles the shape of the target distribution, and so it can be used to generate a candidate proposal by simply

adding it – along with some small amount of random noise – to the chain that is being currently updated (see Steps 22-24 and 35-37 in Figure 3). The standard Metropolis-Hastings acceptance probability of the target distribution is calculated by combining the likelihood function and prior distribution to determine if the new proposal should be accepted or rejected (Steps 25 and 38 in Figure 3). As presented in Equation 7, because the decision threshold parameter (a) only depends on the behavioral data, it is updated separately from the parameters that are used in the factor analysis component of the model. The other diffusion model parameters depend on both aspects of the data, and so acceptance probabilities should be calculated using the likelihood functions of both the diffusion model and the factor analysis model. A detailed, algorithmic description of the sampling procedure is shown in Figure 3. Although this figure includes details of how to apply the Lasso to the FA NDDM, we will save discussion of the Lasso until the next section.

In the FA NDDM, all factor loadings except for those in the first q rows are freely estimated which is an exploratory aspect of the model. Although this enables us to study a factor structure of the manifest variables, this approach will produce complex structures, whereas often parsimonious solutions are desired to facilitate interpretations of the factor structure. In a typical EFA, rotation techniques are applied to find a simple structure by minimizing a complexity function. The central focus of our article is that statistical regularization can also be applied to remove small and unimportant factor loadings, while detecting important loadings (Choi et al., 2010; Hirose & Konishi, 2012; Hirose & Yamamoto, 2015; Lu et al., 2016; Muthén & Asparouhov, 2012; Ning & Georgiou, 2011). Given the conjugate priors and posteriors we reviewed, it is straightforward to apply the hierarchical representation of the Bayesian Lasso (Park & Casella, 2008) to the FA NDDM, mirroring previous studies of the structural component within SEM (Feng et al., 2017b; Guo et al., 2012; Wang et al., 2013).

4 Applying the Bayesian Lasso to the FA NDDM

In this section, we extend the FA NDDM to the Lasso FA NDDM by applying the hierarchical representation of the Bayesian Lasso proposed by Park and Casella (2008). The priors and posteriors for intercepts, residual variances and factor covariance matrix remain the same as in Equation 10 and Equation 11. For the factor loadings, the following hierarchical priors are applied to implement shrinkage effects.

$$\begin{aligned}
\boldsymbol{\Lambda}_j|\ \psi_{\epsilon j}, \boldsymbol{\tau}_j &\sim MVN_q(\boldsymbol{\Lambda}_{0j},\ \psi_{\epsilon j}\boldsymbol{H}_{0j}),\\
\boldsymbol{H}_{0j} = diag(\boldsymbol{\tau}_j),\ \boldsymbol{\tau}_j &= (\tau_{j1}^2, \ldots,\ \tau_{jq}^2)^T,\\
\pi(\boldsymbol{\tau}_j|\ \kappa^2) &\propto \prod_{k=1}^{q} \frac{\kappa^2}{2}\text{exp}(-\kappa^2\tau_{jk}^2/2),\\
\kappa^2 &\sim Gamma(\alpha_{0\kappa},\ \beta_{0\kappa}).
\end{aligned} \tag{12}$$

As a global and local shrinkage prior, κ controls the global shrinkage effect on all the factor loadings Λ_j, $j = 1, \ldots, p$, but the effect is modulated by $\boldsymbol{\tau}_j$. Also, usually $\boldsymbol{\Lambda}_{0j} = 0$ in regularization methods. To implement the Lasso to the FA NDDM, this prior should be applied to $\Lambda^*_{p\times q}$ in Equation 7, the bottom p rows of the factor loading matrix.

Given $\boldsymbol{N}_j$, a vector of the j-th neural covariate containing all n observations, the corresponding conjugate posterior distributions are

$$\begin{aligned}
\boldsymbol{\Lambda}_j|\ \boldsymbol{N}_j, \Omega, \psi_{\epsilon j}, \boldsymbol{\tau}_j &\sim MVN_q(\boldsymbol{\Lambda}_{nj},\ \psi_{\epsilon j}\boldsymbol{H}_{nj}),\\
\boldsymbol{\Lambda}_{nj} = \boldsymbol{H}_{nj}(\boldsymbol{H}_{0j}^{-1}\boldsymbol{\Lambda}_{0j} + \boldsymbol{\Omega}\boldsymbol{N}_j), &\quad \boldsymbol{H}_{nj} = (\boldsymbol{H}_{0j}^{-1} + \boldsymbol{\Omega}\boldsymbol{\Omega}^T)^{-1},\\
(1/\tau_{jk}^2)|\ \lambda_{jk}, \psi_{\epsilon j}, \kappa^2 \sim IG(\mu_{\tau_{jk}}, \kappa^2), &\quad \mu_{\tau_{jk}} = \sqrt{\frac{\kappa^2}{(\lambda_{jk} - \lambda_{0jk})^2}\psi_{\epsilon j}},\\
\kappa^2|\ \boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_p &\sim Gamma(\alpha_{n\kappa},\ \beta_{n\kappa}),\\
\alpha_{n\kappa} = \alpha_{0\kappa} + qk, &\quad \beta_{n\kappa} = \beta_{0\kappa} + \frac{1}{2}\sum_{j}^{p}\sum_{k}^{q}\tau_{jk}^2.
\end{aligned} \tag{13}$$

where λ_{jk} and λ_{0jk} are (j,k) elements of $\mathbf{\Lambda}$ and $\mathbf{\Lambda}_0$, respectively, and IG indicates an inverse Gaussian distribution:

$$f(x \mid \mu, \kappa^2) = \left(\frac{\kappa^2}{2\pi x^3}\right)^{\frac{1}{2}} \exp\left(\frac{-\kappa^2(x-\mu)^2}{2\mu^2 x}\right). \tag{14}$$

With these prior distributions for factor loadings, the conditional distributions for intercepts, residual variances, and factor covariance matrix, and the sampling method for the diffusion model parameters, the full joint posterior distributions can be estimated. The algorithm to estimate the joint posterior distribution of the parameters in the FA NDDM with the Lasso is shown in Figure 3. The sampling method in the algorithm is a combination of the conditional posterior sampling and the DE-MCMC.

The Bayesian model proposed here differs from the previous Bayesian Lasso applications to latent variable models (Feng et al., 2017a, 2017b; Guo et al., 2012; Song et al., 2014; Wang et al., 2013). First, the Bayesian Lasso is applied to the factor analysis model (i.e., the measurement model in the SEM) in the current approach, whereas previous studies applied regularization methods to the structural model of the SEM. In doing so, the model is able to explore a factor structure of the manifest variables regarding the latent variables defined based on the cognitive components in the behavioral model employed in the joint modeling framework. As the Lasso FA NDDM exploratorily estimates factor loadings, the solution in Equation 9 should be implemented during the posterior sampling to address the sign-switching problem. Second, a single global parameter κ is used in the proposed prior, whereas this parameter is allowed to vary by different dependent variables in the previous studies. If we were to allow multiple global parameters in the current approach, the model should simultaneously update $\kappa_1, \ldots, \kappa_p$, one for each dependent (manifest) variable, instead of a single κ which exerts its effect on all of the variables. In this case, each global parameters $\kappa_j, j = 1, \ldots, p$ and corresponding local shrinkage parameters $\tau_{j1}, \ldots, \tau_{jq}$ would be updated on the basis of only the $q = 3$ latent variables of the DDM. Having only a little information to constrain penalty parameters would severely limit our

Initiate $\boldsymbol{\theta}^{(0)}$ for all chains k: $1 \leq k \leq K$ (e.g., sample from priors).

for Iteration s: $0 \leq s \leq S-1$ **do**

  for Chain k: $1 \leq k \leq K$ **do**

    Calculate factor scores $\boldsymbol{\omega}_i^{(s)} = (\tilde{t}_i^{(s)}, \tilde{z}_i^{(s)}, d_i^{(s)})^T - \boldsymbol{\mu}_{1:3}^{(s)}$ for all $i : 1 \leq i \leq n$ where $\tilde{t}_i^{(s)} = \log(t_i^{(s)})$ and $\tilde{z}_i^{(s)} = \text{logit}(\frac{z_i^{(s)}}{a^{(s)}})$. Let $\boldsymbol{\Omega}^{(s)} = (\boldsymbol{\omega}_1^{(s)}, \cdots, \boldsymbol{\omega}_n^{(s)})$.

    Update $\boldsymbol{\Lambda}^*$ and $\boldsymbol{\Psi}^*$ (Eq. 11 and 13):

      for j: $1 \leq j \leq p$ **do**

        Define $\boldsymbol{H}_{nj} = diag(\boldsymbol{\tau}_j^{(s)})$ and **Calculate** $\boldsymbol{\Lambda}_{nj}$, $\boldsymbol{H}_{nj}$, $\alpha_{n\epsilon j}$, and $\beta_{n\epsilon j}$.

        Sample $\boldsymbol{\Lambda}_j^{(s+1)} \sim MVN_q(\boldsymbol{\Lambda}_{nj},\ \psi_{\epsilon j}^{(s)} \boldsymbol{H}_{nj})$ and $(\psi_{\epsilon j}^{(s+1)})^{-1} \sim Gamma(\alpha_{n\epsilon j}, \beta_{n\epsilon j})$.

      end for

    Update $\boldsymbol{\Phi}$ (Eq. 11): **Sample** $\boldsymbol{\Phi}^{(s+1)} \sim IW_q(R_0^{-1} + (\boldsymbol{\Omega}^{(s)})(\boldsymbol{\Omega}^{(s)})^T, n + \rho_0)$.

    Update $\boldsymbol{\mu}$ (Eq. 7 and 11):

      Calculate sample means $\bar{\boldsymbol{y}}$ of $(\tilde{t}_i^{(s)}, \tilde{z}_i^{(s)}, d_i^{(s)}, \boldsymbol{n}_i)$ over $i : 1 \leq i \leq n$, and then $\boldsymbol{\upsilon}_n$ and $\boldsymbol{\Delta}_n$.

      Sample $\boldsymbol{\mu}^{(s+1)} \sim MVN_{p+q}(\boldsymbol{\upsilon}_n, \boldsymbol{\Delta}_n)$.

    Update $\kappa, \tau_1, \cdots, \tau_p$ (Eq. 13):

      for j: $1 \leq j \leq p$ **do**

        for l: $1 \leq l \leq q$ **do**

          Calculate $\mu_{\tau_{jk}} = \sqrt{\frac{(\kappa^{(s)})^2}{(\lambda_{jk}^{(s+1)} - \lambda_{0jk})^2} \psi_{\epsilon j}^{(s+1)}}$ and then **Sample** $(\tau_{jk}^{(s+1)})^{-2} \sim IG(\mu_{\tau_{jk}}, (\kappa^{(s)})^2)$.

        end for

      end for

      Calculate $\alpha_{n\kappa}$ and $\beta_{n\kappa}$ and then **Sample** $\kappa^2 \sim Gamma(\alpha_{n\kappa},\ \beta_{n\kappa})$.

    Update a (Eq. 7):

      Sample two chains except for k. Let a_1 and a_2 be their last samples of a.

      Sample $\gamma \sim U(0.5, 1)$ and $e \sim U(-b, b)$.

      Propose by crossover: $a^* \leftarrow a^{(s)} + \gamma(a_1 - a_2) + e$.

      Calculate p_A with $\prod_i^n [\textit{Diffusion}(a^*, t_i^{(s)}, z_i^{(s)}, d_i^{(s)})]$ and $\prod_i^n [\textit{Diffusion}(a^{(s)}, t_i^{(s)}, z_i^{(s)}, d_i^{(s)})]$.

      Generate $p^* \sim U(0, 1)$

      if $p^* < p_A$ **then**

        Store $a^{(s+1)} \leftarrow a^*$

      else

        Store $a^{(s+1)} \leftarrow a^{(s)}$

      end if

    Update $\boldsymbol{\eta}_i = (\tilde{t}_i, \tilde{z}_i, d_i)^T$ (Eq. 7):

      Calculate $\boldsymbol{\Sigma}^{(s+1)} = (\boldsymbol{\Lambda}^{(s+1)})(\boldsymbol{\Phi}^{(s+1)})(\boldsymbol{\Lambda}^{(s+1)})^T + \boldsymbol{\Psi}^{(s+1)}$

      for j: $1 \leq j \leq p$ **do**

        Sample two chains except for k. Let $\boldsymbol{\eta}_{1i}$ and $\boldsymbol{\eta}_{2i}$ be vectors of their last samples of $(\tilde{t}_i, \tilde{z}_i, d_i)^T$.

        Sample $\gamma \sim U(0.5, 1)$ and $e \sim U(-b, b)$.

        Propose by crossover: $\boldsymbol{\eta}^* = (\tilde{t}_i^*, \tilde{z}_i^*, d_i^*)^T \leftarrow \boldsymbol{\eta}^{(s)} + \gamma(\boldsymbol{\eta}_{1i} - \boldsymbol{\eta}_{2i}) + e$ and inverse-transform $(\tilde{t}_i^*, \tilde{z}_i^*)$ into (t_i^*, z_i^*): $t_i^* = \exp(\tilde{t}_i^*)$ and $z_i^* = a^{(s+1)} \cdot \left(\frac{\exp(\tilde{z}_i^*)}{1+\exp(\tilde{z}_i^*)}\right)$.

38: **Calculate** p_A with *Diffusion*$(a^{(s+1)}, t_i^*, z_i^*, d_i^*)$, *Diffusion*$(a^{(s+1)}, t_i^{(s)}, z_i^{(s)}, d_i^{(s)})$, $(\tilde{t}_i^*, \tilde{z}_i^*, d_i^*, \boldsymbol{n}_i^T)^T \sim MVN_{p+q}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\Sigma}^{(s+1)})$, and $(\tilde{t}_i^{(s)}, \tilde{z}_i^{(s)}, d_i^{(s)}, \boldsymbol{n}_i^T)^T \sim MVN_{p+q}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\Sigma}^{(s+1)})$.

39: **Generate** $p^* \sim U(0,1)$

40: **if** $p^* < p_A$ **then**

41: **Store** $\boldsymbol{\eta}^{(s+1)} \leftarrow \boldsymbol{\eta}^*$

42: **else**

43: **Store** $\boldsymbol{\eta}^{(s+1)} \leftarrow \boldsymbol{\eta}^{(s)}$

44: **end if**

45: **end for**

46: **end for**

47: **end for**

Figure 3: **Pseudocode for Performing Posterior Sampling.** The pseudocode here can be used to draw samples from the joint posterior distribution of the factor analysis neural drift diffusion model with the Lasso application. Except where stated, the last posterior sample in the chain should be used in all calculations above. Although not specified, prior distributions should be taken into account when calculating the acceptance probability p_A (as defined in Metropolis-Hasting algorithms) in Steps 25 and 38. Some calculations can be grouped, or the order of updating steps may be rearranged to increase computational efficiency.

Notations) n: number of trials, p: number of neural covariates, q: number of behavioral factors, K: number of chains, S: number of iterations, $\boldsymbol{\theta}$: vector of all parameters, b: small noise parameter for the crossover step (e.g., $b = 0.005$), p_A: acceptance probability.

ability to estimate all model parameters, and so we chose to explore only a single global penalty parameter in this article.

5 Simulation study

In this section, we present three simulation studies meant to systematically examine the effectiveness of the Bayesian Lasso in three factor loading settings: simple, overlapping, and complex structures. In each simulation, we generated data by assuming there are $p = 128$ neural features, and the number of trials was $N = 300$ as in Turner, Wang, and Merkle (2017). We set the number of factors equal to $q = 3$, representing each of the single-trial DDM parameters (i.e., nondecision time, bias, and drift rate). To induce different complexities, we specified three different settings of the factor loading matrix (reported below).

After generating data in each simulation, we fit the FA NDDM and the Lasso FA NDDM to the simulated data. We used a combination of the conditional posterior sampling (for the factor analysis parameters) and the DE-MCMC (Ter Braak, 2006; Turner, Sederberg, et al., 2013) sampling (for single-trial DDM parameters). Each time, the algorithm was run for 20,000 iterations with 12 chains. The first 2,000 samples were treated as the burn-in period and were discarded, resulting in 216,000 samples.

5.1 Simulation 1: Simple structure

In Simulation 1, we assumed a factor loading matrix with a simple structure, such that each observed variable loaded onto a single factor. Except for the first three rows, which were constrained to be diagonal as discussed above, the true factor loading matrix had the following form

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 & 0 & 0 \\ 0 & \mathbf{\Lambda}_2 & 0 \\ 0 & 0 & \mathbf{\Lambda}_3 \end{bmatrix}$$

where $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, and $\mathbf{\Lambda}_3$ are $(q_1 \times 1)$, $(q_2 \times 1)$, and $(q_3 \times 1)$ vectors of factor loadings, respectively, and $q_k, (k = 1, 2, 3)$ indicates the number of manifest variables related to the k-th factor. To generate the factor loading shape, we set $q_1 = 64$, $q_2 = 32$, and $q_3 = 32$, respectively. We based this decision on Turner, Wang, and Merkle (2017)'s factor loading results from real data analysis, where the nondecision time factor typically had more large factor loadings than the other two factors.

Following the simulation study in Turner, Wang, and Merkle (2017), we randomly generated parameter values for the FA NDDM according to the following specifications:

- $\lambda_{jk} \sim TN(0.5, 0.2, 0, \infty)$
- $\psi_j \sim U(0.03, 0.3)$
- Φ: $\phi_{jj} = 1$ and $\phi_{jk} = 0.3$
- $\mu_{1:3} = (-0.5, 0.1, 1.5)$ and $\mu_{4:131} \sim N(0.3, 0.05)$
- $a = 2$

When fitting the FA NDDM to the data, regardless of the factor loading structure, we initialized all parameter values by randomly sampling values from Unif$(0.2, 0.8)$. The goal of the simulation study was to assess whether or not the Lasso FA NDDM could detect and shrink factor loadings with zero true values while estimating the other loadings well.

5.2 Results

The top panel of Figure 4 shows the parameter recovery results for the factor loading matrix. On the left, the maximum a posteriori (MAP) estimates of the factor loadings whose true values are nonzero are plotted on the x-axis with their true values plotted on the y-axis. On the right, the histogram shows the estimates of the factor loadings whose

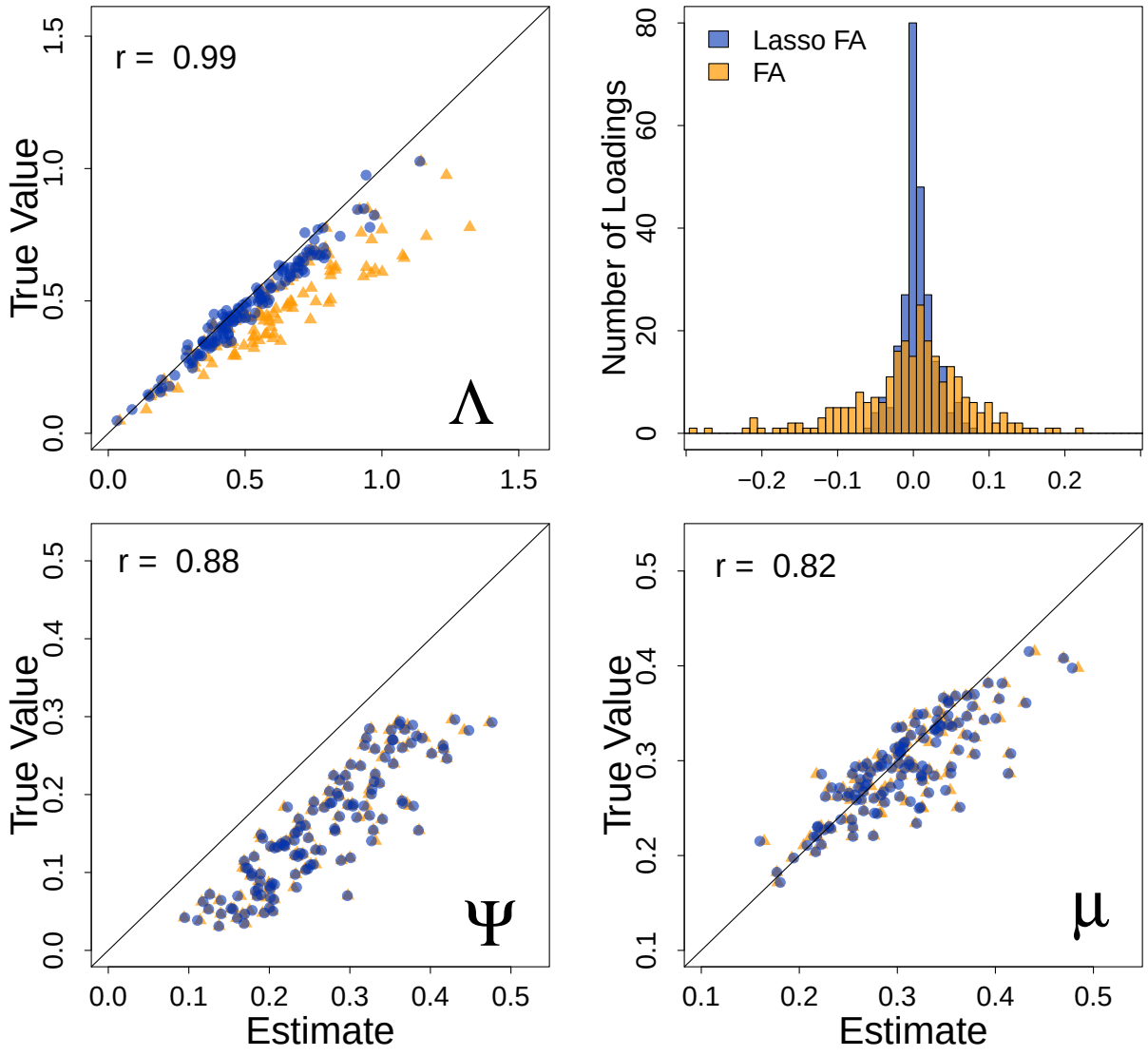

Figure 4: **Structural Recovery Results, Simple Structure.** In each panel, parameter estimates (x-axis) obtained by the Lasso FA NDDM (blue) and the FA NDDM (orange) are shown against the true parameter values (y-axis). The top right panel shows the distribution of estimated factor loadings with zero true values. Where appropriate, Pearson correlations are reported for the Lasso FA NDDM results within panels. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

true values are zero (i.e., zero loadings). On each panel, the Lasso estimates (blue) and plain FA NDDM estimates (orange) are plotted together for comparison.

For the factor loadings with nonzero true values, the Pearson correlation between the true values and their estimates is 0.994 in the Lasso FA NDDM and 0.964 in the FA NDDM.

Although the correlation is high in both methods, the factor loadings are slightly overestimated in the plain FA NDDM result which is consistent with Turner, Wang, and Merkle (2017)'s result. By contrast, there is no systematic bias in the Lasso result. It can be speculated that the overestimation of the nonzero factor loadings is corrected due to the shrinkage effect of the Lasso. For factor loadings with zero true values, estimates are near zero in the both methods. However, the Lasso tends to produce estimates that are nearer to zero than plain FA. Unlike the frequentist Lasso, the Bayesian Lasso cannot produce zero estimates exactly as it estimates the posterior distributions of the coefficients. To examine the false alarm rate of the Lasso and plain FA NDDMs, we can specify an arbitrary criterion to determine if the estimated factor loadings are meaningfully large. A cutoff of 0.1 is used following the previous studies (Feng et al., 2017a, 2017b; Guo et al., 2012; Hoti & Sillanpää, 2006) and thus if MAP estimates of the factor loadings fall within the cutoff region $|\lambda_{jk}| < 0.1$, they are considered too small to be meaningful ('unimportant'). If the MAP estimates fall outside of the cutoff region ($|\lambda_{jk}| > 0.1$), they are considered large and meaningful. Under the simple structure, all of the Lasso estimates corresponding to the 256 truly zero loadings fall under this cutoff value (they range between [-0.062, 0.081]), producing a false alarm rate of zero. For the plain FA NDDM, 44 of 256 estimates are detected as large, and thus the false alarm rate is 0.172. The Lasso application increases the miss rate from 0.008 to 0.016, but this effect is relatively minor compared to the improvement of the false alarm rate. These results demonstrate that applying the Lasso to the FA NDDM can be useful in better discriminating large meaningful factor loadings from small and unimportant ones.

Figure 5 provides another look at the parameter recovery for the factor loading matrix. In each panel, a matrix is shown whose elements are colored according to the legend on the right-hand side. The left panel shows the true factor loading matrix used to generate the data. Here, the simple structure is evident, as each element (e.g., neural feature) loads onto only a single factor (columns). The second panel shows how each parameter was

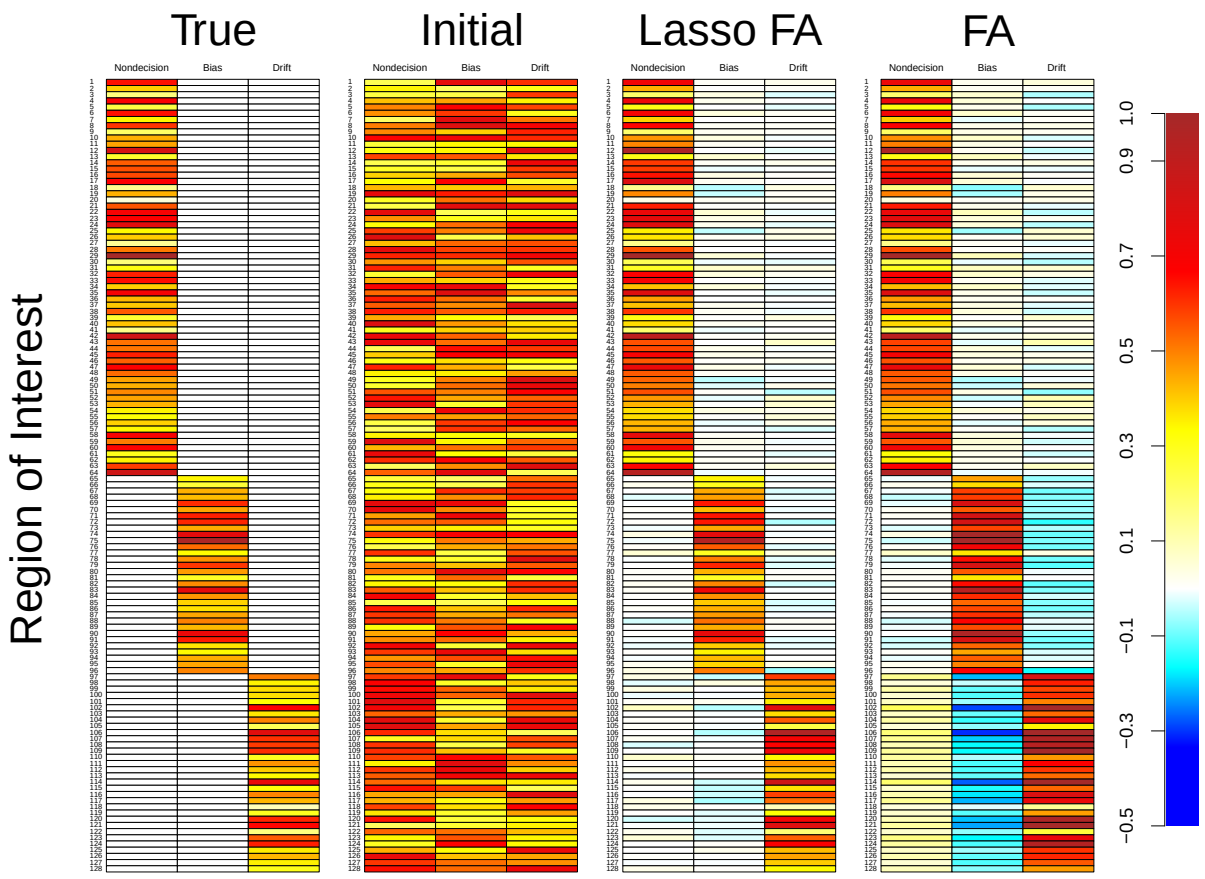

Figure 5: **Factor Loading Recovery Results, Simple Structure.** The left panel shows the true factor loading structure used to generate the data. The second panel shows how both the Lasso FA and FA NDDMs were initialized with random starting values. The third and fourth panels show the recovered factor loading matrix when using the Lasso FA NDDM and FA NDDM, respectively. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

initialized when fitting the models to data. The third and fourth panels show the recovery results obtained when fitting the Lasso FA NDDM and FA NDDM to the generated data, respectively. The right two panels show two principle results. First, they illustrate that the estimates are clearly different from their initialized values, suggesting that the fitting algorithm is not biasing our results. Second, they both illustrate accurate recovery of the true, data-generating factor structure to used to generate the data. Comparing across the two rightmost panels, Figure 5 shows that the Lasso FA NDDM provides more accurate parameter estimates compared to the FA NDDM.

The bottom panel of Figure 4 shows the parameter recovery results for residual variances (Ψ) and intercepts (μ), whereas Figure 6 shows the results for factor variances and covariances (Φ). For the residual variances, the Pearson correlations between the true values and the estimates are 0.879 and 0.877, and for the intercepts, the correlations are

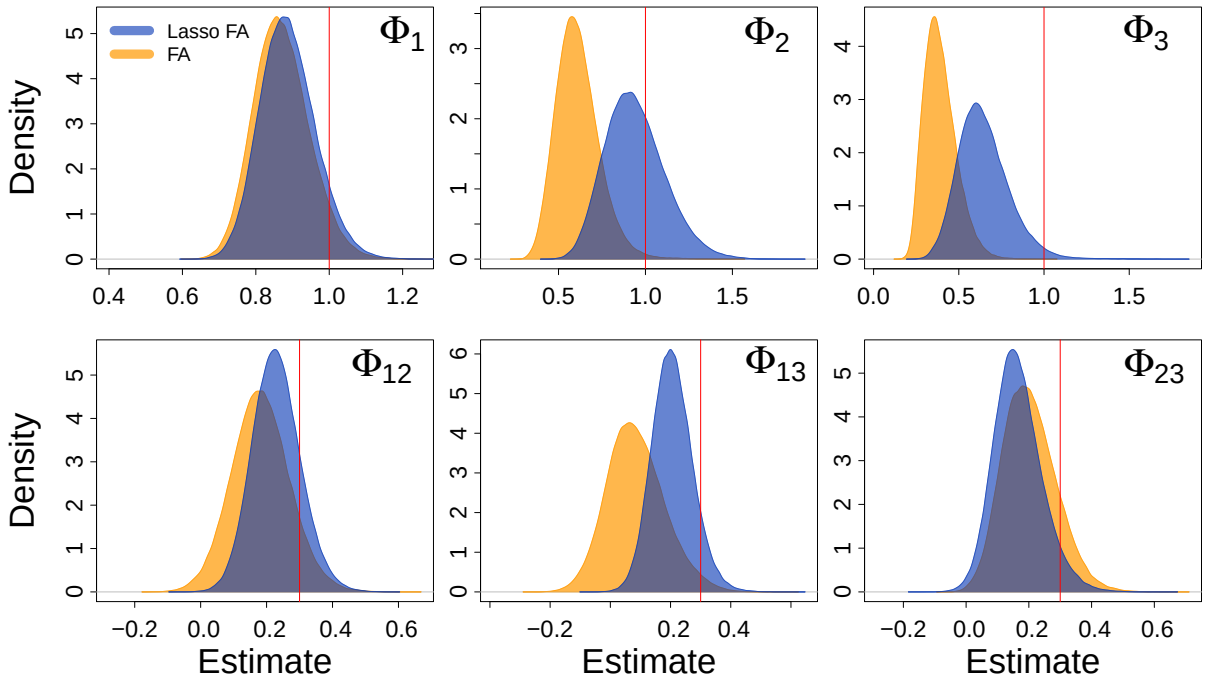

Figure 6: **Factor Variance Estimates, Simple Structure.** Each panel shows the estimated posterior distributions for each element of the factor variance matrix obtained by either the Lasso FA NDDM (blue) or the plain FA NDDM (orange). The red vertical lines indicate the true values. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

0.821 and 0.822, in the Lasso FA NDDM and in the plain FA NDDM, respectively. In this recovery, factor variances are underestimated and residual variances are overestimated, whereas intercepts do not show any systematic bias. Comparing the Lasso and plain FA NDDM, the factor variance for nondecision time (Φ_1) is estimated similarly while the variances for the other diffusion model parameters (Φ_2 and Φ_3) are better estimated when the Lasso is implemented. Biases in the estimation result are due to the systematic relationship that exists among the parameters. In factor analysis, the model estimates parameters by reducing the discrepancy between a sample covariance matrix $\boldsymbol{S}$ and a model-predicted covariance matrix $\hat{\Sigma} = \hat{\Lambda}\hat{\Phi}\hat{\Lambda}^T + \hat{\Psi}$ (i.e., implied covariance matrix). Given a fixed amount of common variances explained by the factor model ($\hat{\Lambda}\hat{\Phi}\hat{\Lambda}^T$), overestimation of the factor loading matrix is accompanied by underestimation of the factor variances. Also, if the common variances are obtained smaller than the optimum, the residual variances ($\hat{\Psi}$) should be overestimated to better match the sample covariance matrix. The FA NDDM tends to overestimate the factor loadings and the bias propagates to the factor variances so that they are underestimated. The Lasso FA NDDM corrects the overestimation bias in the factor loading estimates and thus the variances can also be better estimated. One might expect that the shrinkage effect of the Lasso or other regularization methods will reduce the common variances and increase the residual variances. However, estimates for residual variances remain almost the same even though the regularization is applied. Thus, it can be concluded that the Lasso does not change the total amount of variances that can be explained by the single-trial diffusion model parameters. Instead, the Lasso exerts its shrinkage effect by modulating values of factor loadings and factor variances.

Figure 7 shows the parameter recovery results for the single-trial DDM parameters. Nondecision time (t_i), bias (z_i), and drift rate (d_i) are plotted in the left, middle, and right panels, respectively. The MAP estimates are shown on the x-axis whereas the true values are shown on the y-axis. In each panel, estimates obtained from the Lasso FA NDDM

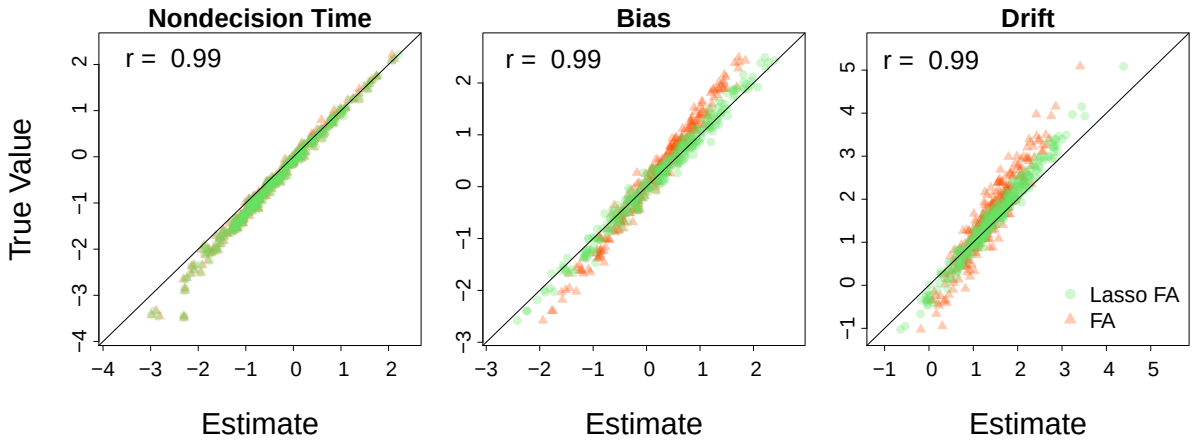

Figure 7: **Single-trial Parameter Estimates, Simple Structure.** Each panel shows the maximum a posteriori estimate of the single-trial parameters for the diffusion decision model component of the model: nondecision time (left), starting point bias (middle), and drift rate (right). In each panel, estimates are shown on the x-axis, whereas the true parameter value is shown on the y-axis, where Lasso FA NDDM results are shown in green and FA NDDM results are shown in red. Pearson correlations for the Lasso FA NDDM are shown in the top left region for each panel. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

are shown in green, whereas estimates of the FA NDDM are shown in red. In general, the parameter estimates are accurate, with the most inaccurate estimates occurring at the most extreme values of the parameter ranges. This is a typical result in a Bayesian framework due to shrinkage effects that occur in hierarchical modeling (Turner, Wang, & Merkle, 2017; Turner et al., 2015). The underestimation bias in the FA NDDM result is the reason for the factor loadings being overestimated. One interesting effect is that the Lasso FA NDDM reduces this shrinkage effect, at least for the bias and drift rate parameters (middle and right panels). This is consistent with the finding that the Lasso corrects the overestimation bias in the factor loadings. The Pearson correlations between the true values and the estimates are 0.995 and 0.994 for nondecision time, 0.991 and 0.989 for bias, 0.988 and 0.948 for drift rate, in the Lasso FA NDDM and in the FA NDDM, respectively.

5.3 Simulation 2: Overlapping structure

Although the simple structure from the previous section is useful from a pedagogical perspective, it is very unlikely that such a simple structure would emerge in real-world data. It is also unlikely that every feature will load onto any factor. For example, Turner, Wang, and Merkle (2017) analyzed data from an experiment and showed that (1) many features were sparse in the sense that they did not have large loadings onto any factor, and (2) several features loaded largely onto more than one factor. Given this possibility, Simulation 2 was designed to evaluate the ability of the Lasso to (1) identify features whose factor loading overlaps across factors, and (2) identify features who exhibit small and trivial factor loading structure. Also, unlike in Simulation 1, nonzero factor loadings were scattered across the matrix rather than being organized according to the simple structure. The left panel of Figure 9 illustrates the true factor loading matrix we used to generate the data. In this figure, the rows of the factor loading matrices are sorted for visual clarity (see the row numbers). Figure 9 shows a more complex pattern of factor loadings where features can load onto either zero, one, two or three factors. All other details of this simulation study, unless otherwise noted, were identical to those presented in Simulation 1.

5.4 Results

The top panel of Figure 8 shows the parameter recovery for the factor loading matrix. As in Figure 4, the left panel shows the MAP estimates for the nonzero loadings whereas the right panel shows the histogram of zero loadings. In general, the results are similar to those in Simulation 1. For the factor loadings with nonzero true values, the Pearson correlation between the true values and their estimates is 0.974 in the Lasso FA NDDM and 0.803 in the FA NDDM. As in Simulation 1, the FA NDDM overestimates some factor loading but the degree of bias becomes larger due to the overlapping structure. These positive biases disappear when the Lasso is applied. For the factor loadings with zero

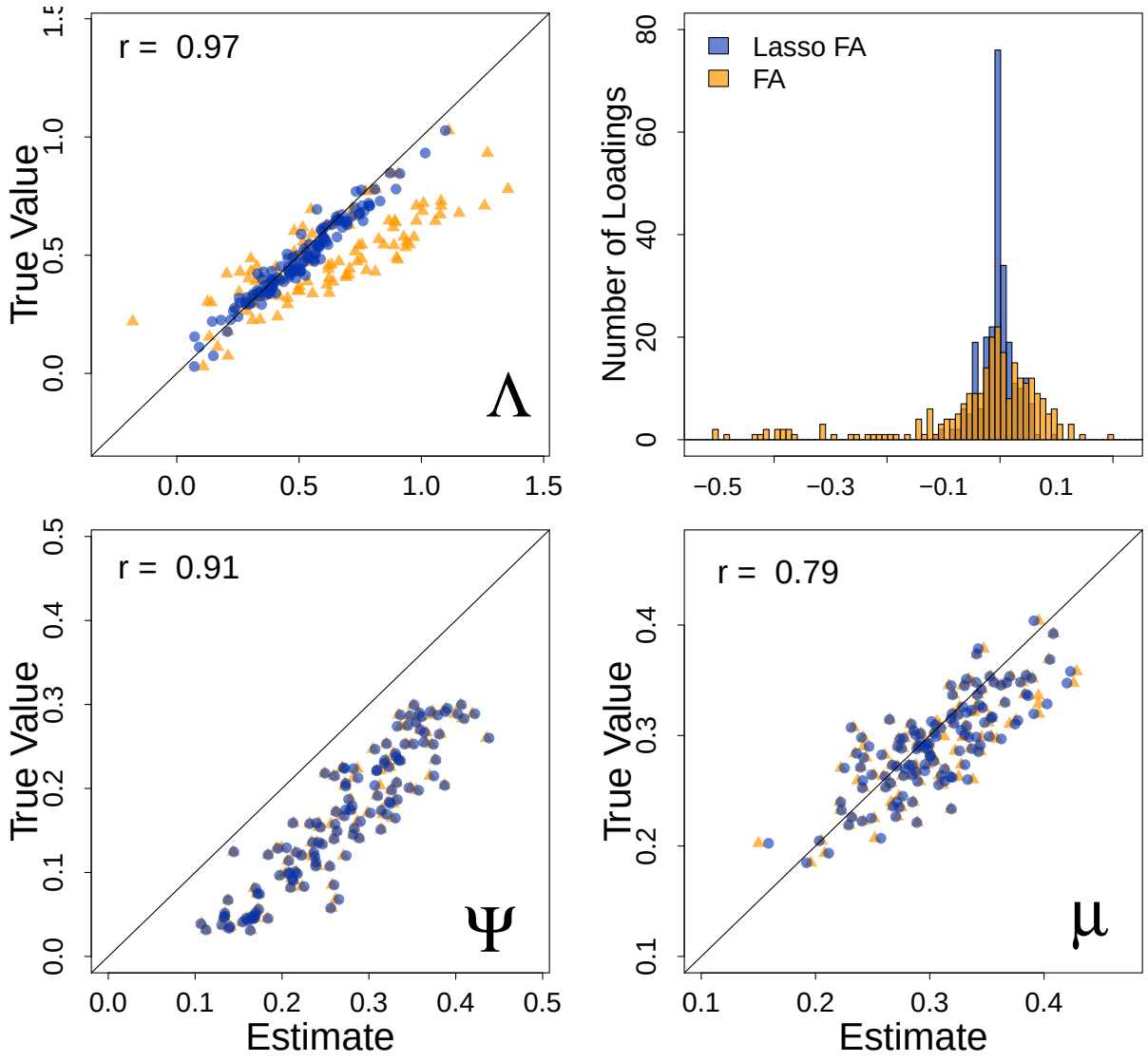

Figure 8: **Structural Recovery Results, Overlapping Structure.** In each panel, parameter estimates (x-axis) obtained by the Lasso FA NDDM (blue) and the FA NDDM (orange) are shown against the true parameter values (y-axis). The top right panel shows the distribution of estimated factor loadings for features with imposed zero-value true loadings. Where appropriate, Pearson correlations are reported for the Lasso FA NDDM results within panels. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

true values, the FA NDDM without the Lasso produces more variable MAP estimates and many of the values are highly negative. As a result, 49 out of 256 zero loadings fall outside of the cutoff region ($|\lambda_{jk}| < 0.1$). In contrast, the Lasso estimates the zero loadings fairly close to zero and 253 of the loadings are identified as too small to be meaning-

ful. The false alarm rates are 0.191 and 0.012 for the plain FA NDDM and the proposed method, respectively, which demonstrates that implementing the Lasso can help the FA NDDM improve its performance in identifying meaningful brain-behavior relations. The shrinkage effect induced by the Lasso increases the miss rate from 0 (in the FA NDDM) to 0.023 (in the Lasso), which is a minor difference compared to the effect on the false alarm rate.

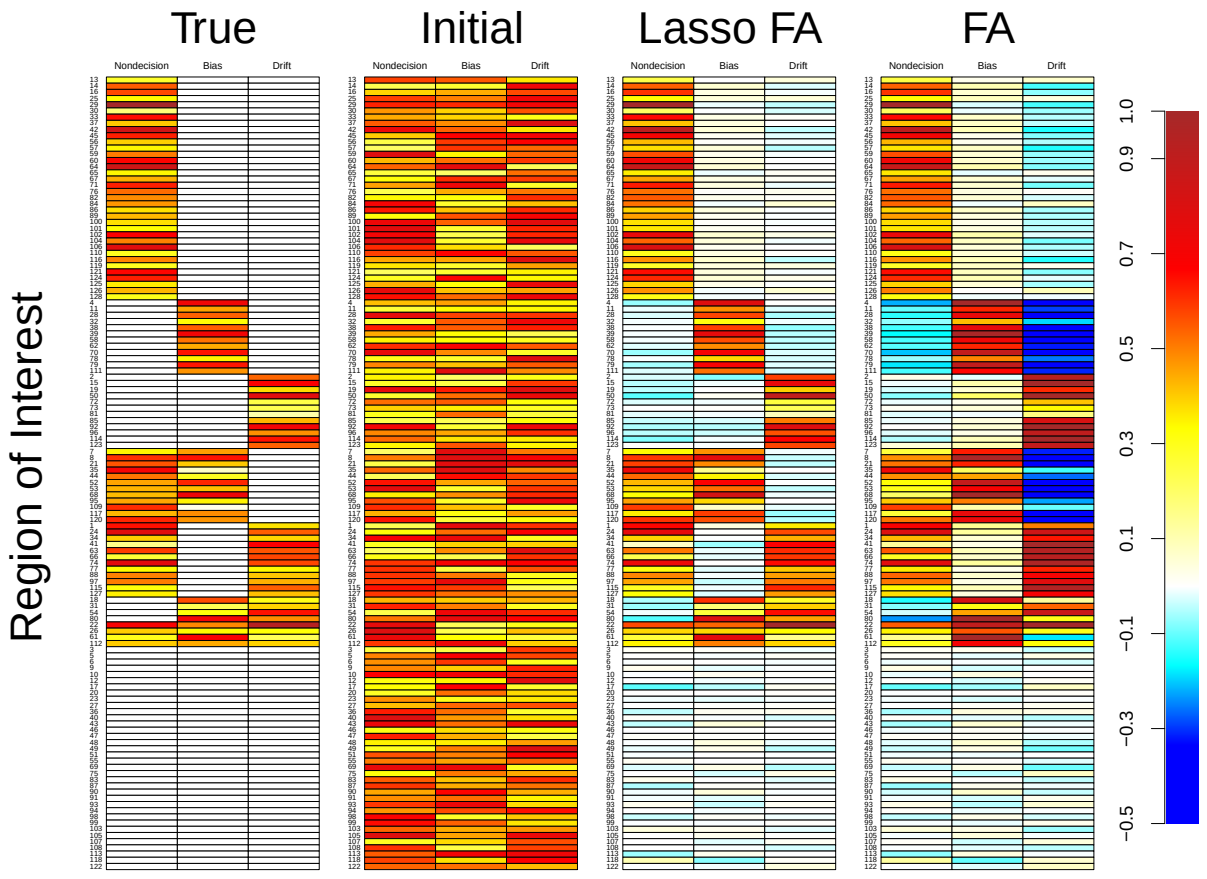

Figure 9: **Factor Loading Recovery Results, Overlapping Structure.** The first (i.e., left) panel shows the true factor loading structure used to generate the data. The second panel shows how both the Lasso FA NDDM and FA NDDM were initialized with random starting values. The third and fourth panels show the recovered factor loading matrix when using the Lasso FA NDDM and FA NDDM, respectively. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

Figure 9 shows another view of the factor loading structure recovery. In each panel, the values of the factor loadings are color coded according to the legend on the far right-

hand side. Although the nonzero factor loadings were randomly arranged within the true factor loading matrix, they are sorted in this figure for visual clarity (see the index on the left side of the panels for the original row numbers). As in Simulation 1, both the Lasso and plain FA NDDMs performed well in retrieving the true structure of the factor loading matrix. However, the plain model produces some high negative values for the zero loadings, many of which are diagnosed as large and meaningful according to our criterion. Furthermore, many of the nonzero loadings are estimated as having higher factor loadings compared to their true values. By contrast, MAP estimates from the Lasso FA NDDM are generally closer to the true values. Although there are some truly zero loadings that are estimated as having small negative loadings (i.e., the pale cyan blocks), most of these estimated loadings are too small to be considered meaningful according to our criterion.

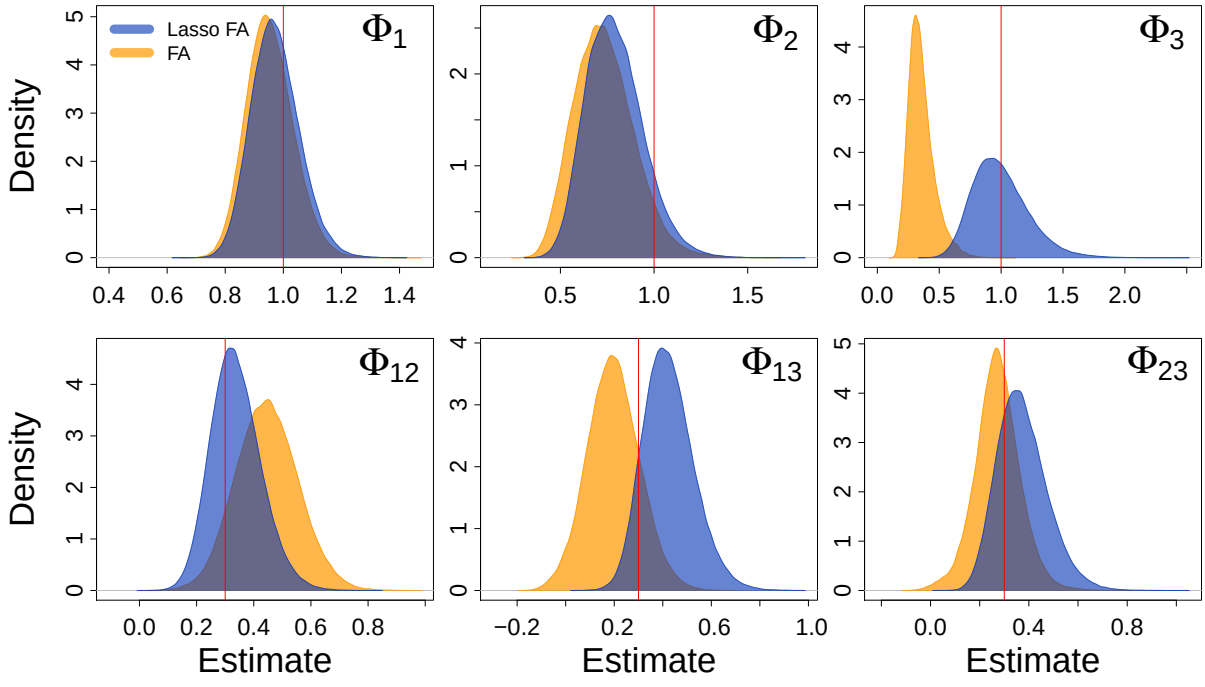

Figure 10: **Factor Variance Estimates, Overlapping Structure.** Each panel shows the estimated posterior distributions for each element of the factor variance matrix obtained by either the Lasso FA NDDM (blue) or the plain FA NDDM (orange). The red vertical lines indicate the true values. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

The bottom two panels of Figure 8 and Figure 10 show the parameter recovery for the residual variances, intercepts, and factor variances and covariances. The MAP estimates for residual variances and intercepts are consistent with the previous simulation and the FA NDDM with and without the Lasso produce very similar results. The Pearson correlations between the true values and the estimates are 0.913 and 0.914 for the residual variances, and 0.785 and 0.786 for the intercepts, in the Lasso FA NDDM and in the FA NDDM, respectively. The factor variances have negative biases in the FA NDDM results as in Simulation 1. When the Lasso is applied, these biases are reduced and the posterior distributions of the variance terms are centered more closely to their corresponding true values.

The parameter recovery results for the single-trial DDM parameters are very similar to those in Simulation 1 (with correlations ranging from 0.98 to 0.99). As such, we do not present those results here; instead, the recovery plot can be found in the supplementary materials.

5.5 Simulation 3: Complex structure

In the two previous simulation studies, all factor loadings with nonzero true values were generated from a truncated normal distribution with the same mean and standard deviation. However, the distribution of factor loadings can exhibit considerable variance and multimodality, in practice. Also, even if there is no strong and meaningful relationship between a manifest variable and a factor, their corresponding factor loading may have some nonzero value due to noise. For the final simulation, we vary factor loading value to a large degree and test the Lasso's ability to detect different degrees of factor loadings. From the buildup of simulation complexity, we consider Simulation 3 to be a realistic case that matches patterns of factor loadings we might expect to see in our real-world application below.

For this simulation, we used the same (overlapping) factor loading structure as in

Simulation 2, but varied factor loading values. Nonzero factor loadings were sampled from two truncated normal distributions with different means and standard deviations:

- $\lambda_{jk} \sim TN(0.8, 0.2; 0.6, 1.0)$ for high loadings
- $\lambda_{jk} \sim TN(0.2, 0.1; 0.15, 0.3)$ for low loadings

This produced a pattern of bimodal factor loadings such that some loadings had high values, whereas others were low.

After the loadings were sampled, we added some small perturbation noise by sampling a random deviate from the normal distribution with mean zero and standard deviation equal to 0.05 and adding this deviate to the factor loading matrix. Although the amount of noise was not large, it further blended the pattern of loadings from the structural constraints such that high factor loadings, low factor loadings, and zero factor loadings would be difficult to discern. In other words, this additional variability allowed us to assess whether the Lasso can discriminate small but meaningful loadings from noise. All other settings and procedures were identical to the previous simulations.

5.6 Results

Figure 11 shows the parameter recovery results for the factor loading matrix. The left panel shows MAP estimates for the large factor loadings, the middle panel shows MAP estimates for the small loadings, and the right panel shows the histogram of MAP estimates for factor loadings centered at zero. Because of the additional noise added to the true values, the distribution of parameter estimates in the right panel vary more than those in the previous simulations. For the high and low factor loadings, the recovery result is similar to the results from previous simulations. The Pearson correlation between the true values and their estimates is 0.968 in the Lasso FA NDDM but 0.587 in the FA NDDM. The large decrease in the FA NDDM is because the factor loadings corresponding to bias and drift rate are largely overestimated. In contrast, the Lasso FA NDDM exhibits

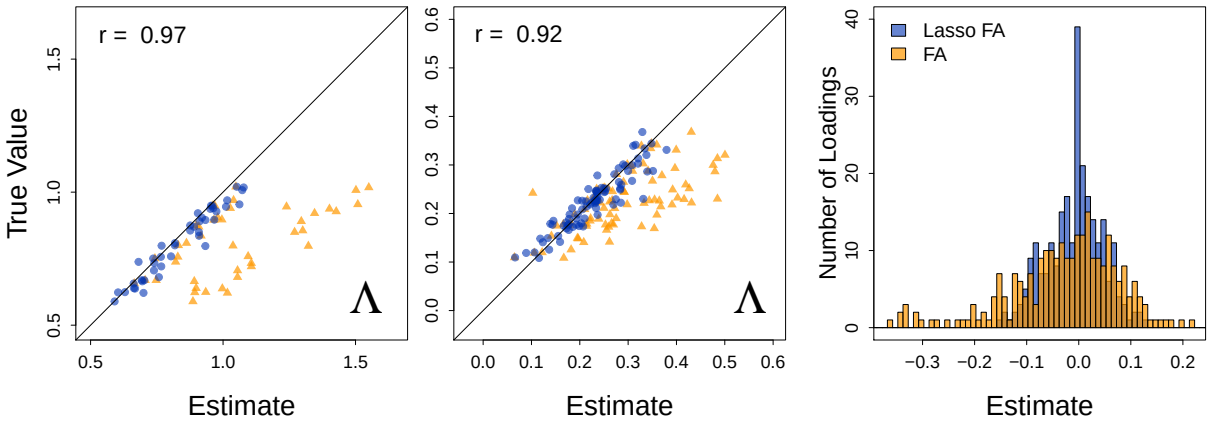

Figure 11: **Structural Recovery Results, Complex Structure.** In the first two panels, parameter estimates (x-axis) obtained by the Lasso FA NDDM (blue) and the FA NDDM (orange) are shown against the true parameter values (y-axis). Factor loadings with high true values are shown on the left and factor loadings with low true values are shown in the middle. The right panel shows the distribution of estimated factor loadings whose true values are centered at zero. Where appropriate, Pearson correlations are reported for the Lasso FA NDDM results within panels. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

no systematic bias and correctly estimates the loadings. Despite the noise, the Lasso still performs well in detecting unimportant loadings. Given our cutoff of 0.1, 238 out of 256 zero loadings are determined as small and unimportant, resulting in a false alarm rate of 0.070. By contrast, the plain model excludes 181 zero factor loadings and so its false alarm rate is 0.293. Although the false alarm rates are slightly higher in this simulation compared to the previous ones, the reduction of the false alarm rate due to the Lasso is remarkable as the miss rate does not change that much across the two methods (0.016 and 0.008 for the FA NDDM with and without the Lasso, respectively). Therefore, we conclude that the noise decreases the accuracy of detecting zero loadings in both methods, but the Lasso FA NDDM is robust and it outperforms the plain model.

Figure 12 shows the recovery of the factor loading structure. The factor loadings are sorted in the same way as in Simulation 2 for visual clarity. The pale yellow and cyan colors in the true factor loading matrix represent factor loadings due to noise. The result is consistent with those of the previous simulations: the Lasso FA NDDM outperforms the

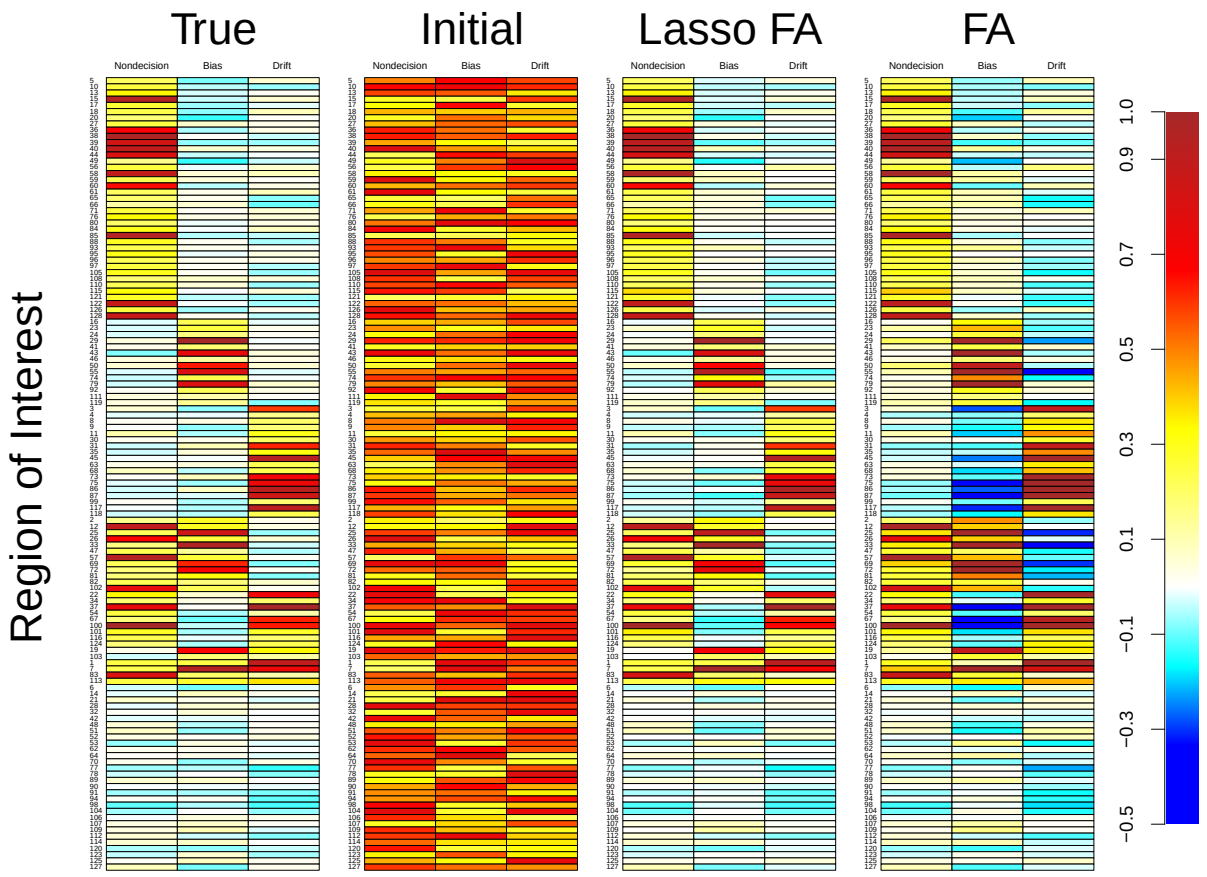

Figure 12: **Factor Loading Recovery Results, Complex Structure.** The left panel shows the true factor loading structure used to generate the data. The second panel shows how both the Lasso FA NDDM and FA NDDM were initialized with random starting values. The third and fourth panels show the recovered factor loading matrix when using the Lasso FA NDDM and FA NDDM, respectively. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

plain model as the MAP estimates with the Lasso reproduce the true values well whereas those without the Lasso suffer from biases.

The parameter recovery results for the other parameters (residual variances, intercepts, factor variances and covariances, and single-trail DDM parameters) are very similar to the previous simulations and the corresponding plots can be found in the supplementary materials. Despite the large variability in the factor loading values and the perturbation noise, the Lasso FA NDDM performs similarly well in this simulation, further demonstrating the robustness of the proposed method.

5.7 Evaluation of the estimation

As discussed in Section 2.1, regularization methods can decrease the MSE of estimators by reducing the variances at the expense of some biases. Because this is a simulation study, we know the true values of the parameters that generated data, and so we can compare the estimates obtained by FA NDDM and the Lasso FA NDDM on the basis of their relative amounts of bias. Similarly, we can evaluate the standard errors of the factor loading estimates by calculating the posterior standard deviations. Although we could have calculated these quantities in the previous two simulation studies, for brevity, we only report the standard errors and MSE for Simulation 3.

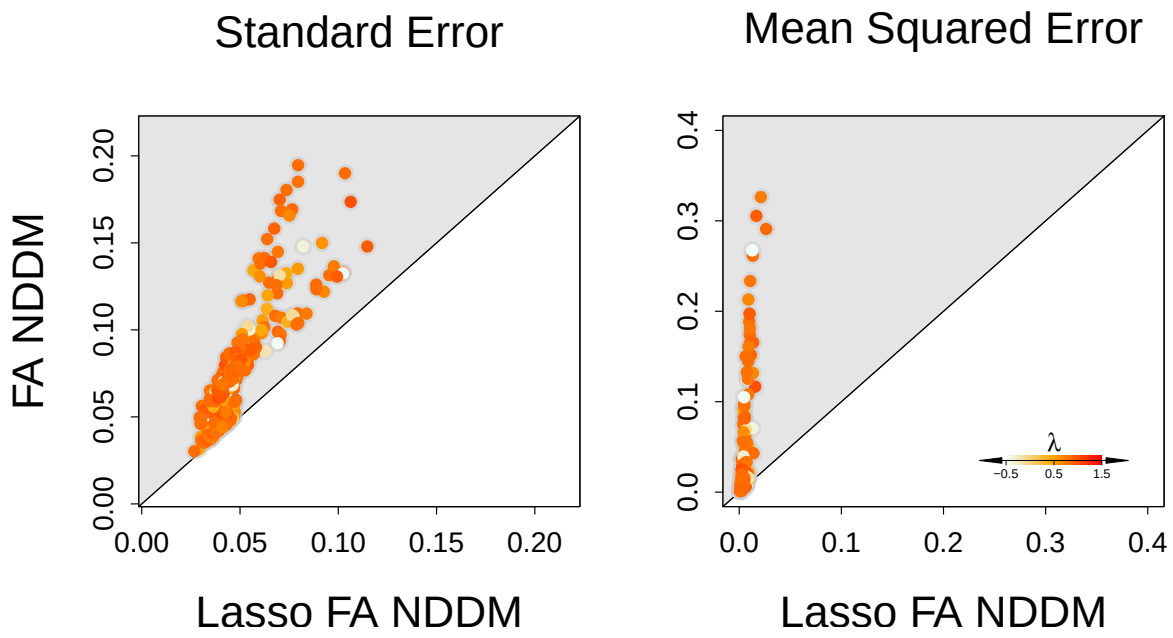

Figure 13: **Standard Errors and Mean Squared Errors, Complex Structure.** The left panel shows the estimated standard errors and the right panel shows the estimated mean squared errors for the FA NDDM with (x-axis) and without (y-axis) the Lasso. The gray-shaded area represents regions in which the Lasso FA NDDM outperforms the plain FA NDDM. The estimates are colored according to the corresponding factor loading estimates from the Lasso FA NDDM. FA NDDM = Factor Analysis Neural Drift Diffusion Model.

Figure 13 shows the standard errors (left) calculated from the standard deviations of the estimated factor loading posterior distributions, and the MSEs (right) calculated from

Equation 3 for the FA NDDM (y-axis) and the Lasso FA NDDM (x-axis). In each panel, the gray-shaded area designates regions in which the Lasso FA NDDM outperforms the plain FA NDDM. As expected, every standard error of the factor loading estimate is smaller when the Lasso is applied. In addition, the MSE is considerably smaller when the Lasso is applied. Specifically, of the factor loadings with truly zero, small, and large values, 98.4%, 95.3%, and 100% of the estimates have smaller MSEs when the Lasso is applied. This pattern of results emerges because the original FA NDDM tends to overestimate the factor loadings and the Lasso corrects this bias (see the structural recovery results in Figure 4, 8, and 11). Hence, unlike what is typically expected from regularization methods, the Lasso FA NDDM estimates of the factor loadings have smaller biases despite the shrinkage induced by the Lasso. The reduction of the MSE combined with the reduced standard error leads us to the conclusion that the Lasso FA NDDM produces better estimates of the factor loadings.

5.8 Effective sample size and convergence

In our simulation study, we ran a combination of the conditional posterior sampling and the DE-MCMC sampling (Figure 3) for 20,000 iterations (the first 2,000 discarded as burn-in) with 12 chains. Having such large numbers of samples is recommended because the number of parameters of the model can easily be large as we attempt to jointly model multiple data modalities and estimate single-trial parameters. Also, some parameters have high autocorrelations due to the complex structure of the behavioral models used in psychology. Specifically, it is often found that the values of parameters in accumulator models trade off with one another, which results in a highly correlated parameter space (Turner, Sederberg, et al., 2013).

As a consequence of the high autocorrelations, the effective sample sizes (ESSs) are not expected to be large relative to the product of the number of iterations and chains. In the current study, the ESSs vary largely across different parameters. The factor load-

ings, which are of our main interest, have ESSs of $3315.9 - 199142.5$ in the most complex condition (Simulation 3: Complex structure). In our result, the high autocorrelations and some small ESSs do not signal a convergence issue of the Markov chains of the Bayesian samples. Figure 14 shows the posterior densities of factor loadings with different ESSs. The posterior densities of a single factor loading were estimated by individual chains and then plotted in the same panel to check the convergence of the chains. The factor loading is displayed along with its associated ESS on the top of each panel. The top left and bottom right panels show the posterior densities of the factor loadings with the minimum and maximum ESSs, respectively. Across all ranges of the ESSs, the posterior densities are well centered around the true values of the factor loadings which are indicated by the red vertical lines. This shows that the chains converged well and the high autocorrelations result from the correlated parameter space, not from any convergence issue. In general, the factor loadings with large true values are loosely constrained and so their posterior densities vary more while those with small and unimportant true values have highly constrained posterior densities. This is due to the main feature of the global and local shrinkage priors (Section 2.4). Convergence of the other parameters were assessed with their posterior samples in the same manner. Also, the estimated values of Gelman-Rubin convergence diagnostic ($\hat{R}$) were smaller than 1.1 for all parameters (Gelman, 1996; Gelman, Carlin, Stern, Dunson, & A. Vehtari, 2013).

6 Application: Brain networks under the speed-accuracy trade-off

Having assured ourselves of the performance of the Lasso under a variety of complexity circumstances, we now apply the Lasso FA NDDM to fMRI data from a perceptual decision making task. The data are first reported in van Maanen et al. (2011), and they consist of choice and response time from a simple, two-choice decision-making task. The fMRI data are obtained in an 8-second scan that preceeded the stimulus presentation, because the scanning protocol was designed to assess off-task behavior, and because de-

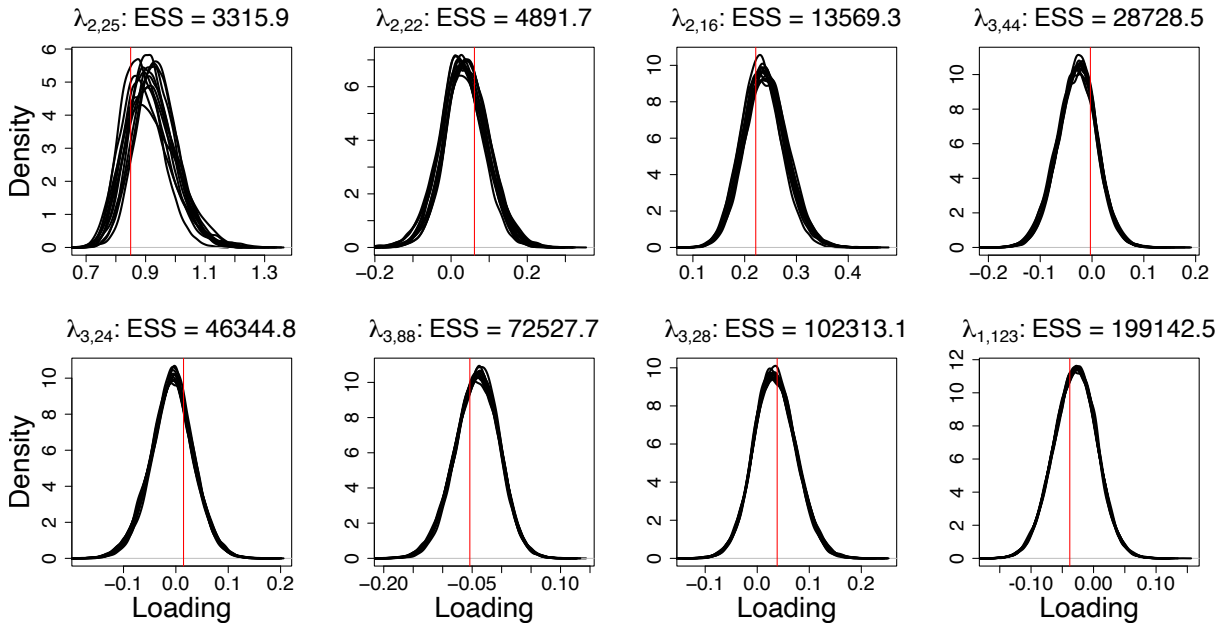

Figure 14: **Posterior Densities of Factor Loadings.** Some factor loadings with different effective sample sizes (ESSs) are selected for the plotting purpose. Their posterior densities are calculated by individual chains separately and then plotted in the same panel. The factor loading plotted and its ESS are displayed on top of each panel. The red vertical lines indicate the true values of the factor loadings.

cisions within the task were typically far faster than the temporal resolution of the blood oxygenated level dependent response used in fMRI. The task was a random dot motion task where subjects were presented with 120 dots in a display, 60 of which move to the left or to the right while the others move in random directions. The subject was instructed to report the direction that most of the dots were moving toward. Additionally, the instructions were manipulated to either emphasize speed or accuracy by telling subjects to either respond "as quickly as possible" or "as accurately as possible". In the reported data, there were 17 subjects (7 female, mean age = 23.1 SD age = 3.1) who participated in the task. More details of the experiments, procedures, and preprocessing of the fMRI data can be found in Turner, Wang, and Merkle (2017) and van Maanen et al. (2011).

For the purposes of comparison, we fit both the Lasso FA NDDM and the FA NDDM separately to trials involving a speed emphasis and those involving an accuracy emphasis. The purpose of fitting the models to separate data streams was to examine whether

brain networks exhibited different functional properties when processing stimuli for speed rather than accuracy. We fit them separately because, despite many theoretical arguments that only a threshold should change across instruction, recent evidence suggested that other parameters may also change as a function of task instruction (Rae, Heathcote, Donkin, Averell, & Brown, 2014). Because we wanted to avoid any undue specification of which parameters should change across task instruction, we chose to allow all parameters to vary. In each of the four fits (i.e., model by instruction), we specified the same prior distributions for each model parameter, and used a combination of conjugate posterior sampling and DE-MCMC sampling to sample from their joint posterior distribution. We ran this algorithm for 20,000 iterations with 18 chains, and discarded the first 5,000 samples as a burnin period. Hence, our parameter estimates are based on 270,000 samples of the joint posterior distribution.

As discussed in the introduction, fitting either FA NDDM model to data requires a constraint to prevent the factor loadings from switching their signs. Turner, Wang, and Merkle (2017) used an element-wise constraint by specifying a uniform prior from zero to one. By doing so, all the loadings were constrained to be positive, circumventing the sign-switching problem altogether. A possible side-effect of this constraint is that, if some factor loadings have true values outside of $[0, 1]$, they cannot be estimated close to the true values, resulting in some biases. As previously explained, these biases can propagate to other factor loadings or variances as the FA NDDM attempts to find a set of estimates that can best approximate the covariance structure in the data.

To avoid these issues, we fit both the Lasso FA NDDM and the FA NDDM using the conjugate prior distributions derived in Equation 10, and the column-wise constraint suggested in Turner, Wang, and Merkle (2017) (and discussed above). Here, the solution relies on our ability to fix the sign of one factor loading in each column, resulting in three constraints for the FA NDDM. Fortunately, considerable progress has been made in the field of model-based cognitive neuroscience in mapping parameters of the DDM to regions

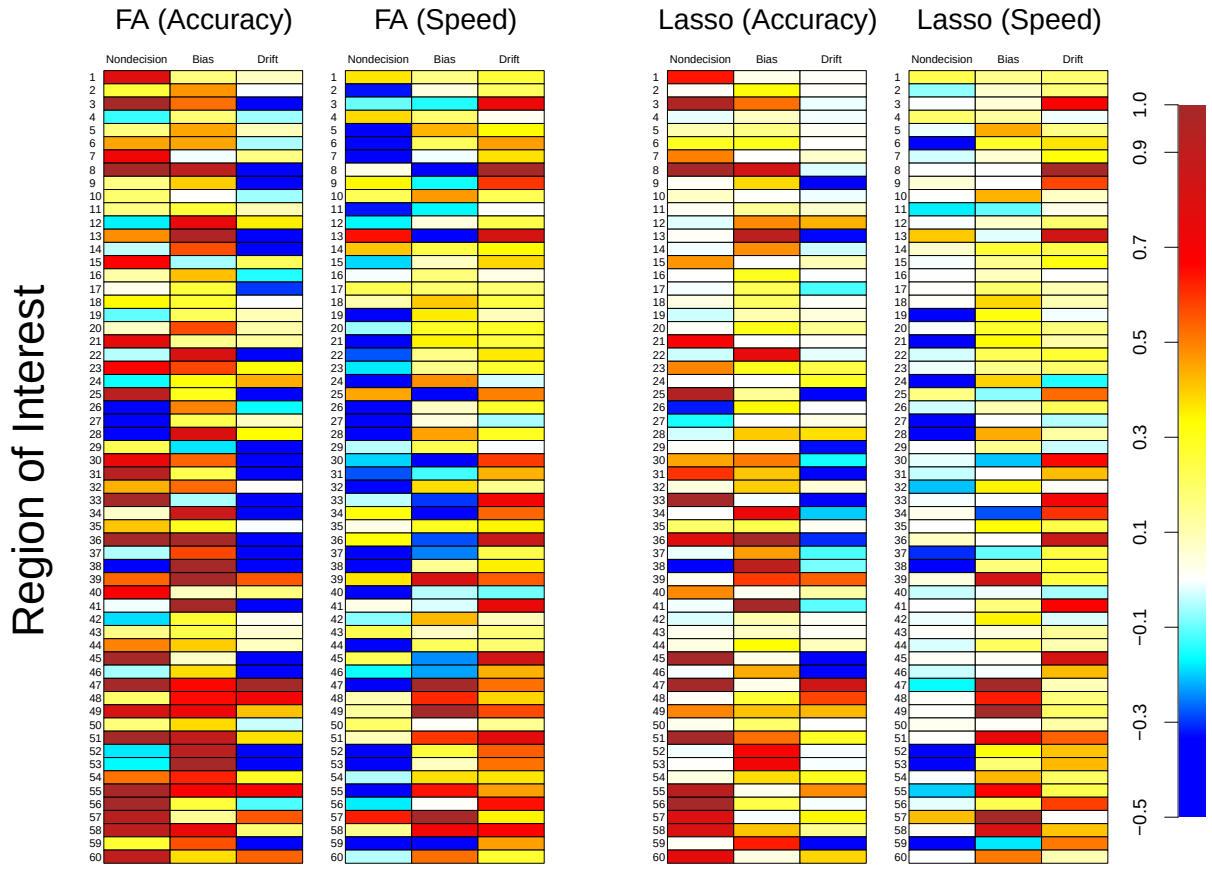

Figure 15: **Factor Loading Matrices for Experimental Data.** The left and right panels show the factor loading matrices estimated for the FA NDDM (left two) and the Lasso FA NDDM (right two) after fits to data from van Maanen et al. (2011). Here, the rows represent different brain regions of interest, whereas the columns correspond to mechanisms of the diffusion decision model (left to right: nondecision time, bias, and drift rate). Each element within the factor loading matrices is color coded according to the legend on the right-hand side.

of the brain (Keuken et al., 2014; Mulder, Van Maanen, & Forstmann, 2014; Forstmann & Wagenmakers, 2015; Turner et al., 2015; Turner, Wang, & Merkle, 2017). According to Turner, Wang, and Merkle (2017)'s result, nondecision time, bias, and drift rate had high positive loadings for brain regions of interest (ROIs) 33, 59, and 57, respectively, in the speed condition (Figure 7 in Turner, Wang, and Merkle). Hence, we fixed the signs of the corresponding factor loadings to be positive when fitting both models to data. The same constraints were imposed on ROIs 39, 24, and 33 for nondecision time, bias, and drift rate, respectively, in the accuracy condition.

Figure 15 shows the MAP estimates for the factor loading matrices from the plain FA NDDM (the left two panels) and the Lasso FA NDDM (the right two panels) for the accuracy (the first and third panels) and speed (the second and fourth panels) conditions. The MAP estimates are color-coded according to the legend on the right-hand side. There are several – but not many – estimated factor loadings with values larger than one, but they are color-coded as if their loadings are exactly one for visual clarity. In each matrix panel, the rows correspond to brain ROIs whereas the columns correspond to the factors of nondecision time, bias, and drift rate, respectively.

Figure 15 allows us to compare the factor loadings across the two instruction conditions, as well as compare the estimates obtained from the FA NDDM to those from the Lasso FA NDDM. In comparing the two methods, Figure 15 shows that several ROIs have high connectivity with the three single-trial parameters. To more specifically examine the difference between the two methods, we applied a threshold of 0.6 to the factor loading values as used in Turner, Wang, and Merkle (2017). In the FA NDDM results, 59 and 24 factor loadings are higher than the threshold in the accuracy and speed conditions, respectively. The application of the Lasso further simplifies this result, producing 32 and 20 large factor loadings in those two conditions, respectively. Also, small loadings are estimated much closer to zero by the Lasso (more whitish cells in Figure 15). Thus, the Lasso produces a more parsimonious structure, allowing us to clearly explore meaningfully related brain networks.

In the Lasso FA NDDM result, there are 17, 10, and 5 ROIs with large loadings on the nondecision time, bias, and drift rate, respectively, in the accuracy condition. Specifically, the ROIs with high factor loadings on the nondecision time factor include the calcarine sulcus (ROIs 1, 3, 21), cerebellum (8), precuneus (31, 33, 55, 56), splenium (31), posterior intraparietal sulcus (38), thalamus (45), superior frontomedian cortex (47), cingulate sulcus (51), rolandic operculum (57), superior temporal gyrus (58) and medial temporal gyrus (60). For the bias factor, ROIs such as the cerebellum (8, 13), medial temporal gyrus

(22), superior frontomedian cortex (34), posterior intraparietal sulcus (38), cingulate gyrus (41), anterior insula (52), and frontopolar cortex (53, 59) have high factor loadings. The drift rate factor has high loadings for the following ROIs: precuneus (33), thalamus (45), parahippocampus (46), superior frontomedian cortex (47), and frontopolar (59).

In the speed condition, a number of regions changes their activation compared to the accuracy condition and there are 3, 8, and 9 ROIs with large loadings on the nondecision time, bias, and drift rate, respectively. For the nondecision time factor, ROIs such as calcarine (21), ventromedial orbitofrontal cortex / precuneus (24), and fusiform gyrus (28) have high factor loadings. Some areas such as the middle frontal gyrus (ROI 39), superior frontomedian cortex (47), mid occipital gyrus (48), cingulate sulcus (51), precuneus (55), rolandic operculum (57), and superior temporal gyrus (58) have high factor loadings on the bias factor. For the drift rate factor, areas including the calcarine sulcus (3), cerebellum (8, 13), thalamus (30, 45), precuneus (33, 56), superior frontomedian cortex (34), cingulate gyrus (41) have high factor loadings.

Together, our results imply that some brain regions are highly related to more than one or all cognitive components of interest, whereas others do not show a noticeable relationship with them. For example, precuneus has high factor loading values on the nondecision time and drift rate in the accuracy condition. In contrast, superior temporal gyrus (ROIs 18 and 19) does not have large factor loadings across all the conditions in both methods. Furthermore, the activation pattern changes considerably across the conditions. For example, posterior intraparietal sulcus (38) does not have high factor loadings in the speed condition, while it does for the nondecision time and bias factors in the accuracy condition. The two ROIs related to cerebellum, 8 and 13, have high factor loadings in the accuracy condition, but their activation switches to the drift rate in the speed condition. Some differences in the activated areas between the current and Turner, Wang, and Merkle (2017)'s results can be attributed to the choice of the boundary constraints of the factor loading values: the parameter range was $[0, 1]$ in the previous study whereas there is no

boundary in the current analysis.

The regularization results may depend on the global penalty parameter κ, and thus different prior specifications for κ can produce considerably different MAP estimates of the factor loading matrix. In the sensitivity analysis (Appendix A), it turned out that our results are robust to the different choices for hyperparameters of κ.

7 Discussion

In this article, we have demonstrated that dimensionality reduction techniques such as the Lasso can be combined with recent endeavors of simultaneously modeling high-dimensional neural data and computational theories of behavioral data. We have illustrated the benefits of using our approach in the context of identifying brain networks that correlate with model mechanisms, such as the drift rate, starting point, and nondecision time parameters assumed by the diffusion decision model. We have shown that the Lasso was successful in decreasing the false alarm rate and producing a sparse composition of brain regions without compromising the miss rate. Although the application of the Lasso has been successful here, there are a few issues that merit further discussion.

7.1 Alternative regularization techniques

Although the current article focused on the Lasso technique, there are many other regularization methods that could be applied to arrive at more parsimonious brain networks. Luckily, many of these methods can be applied using the same framework presented here by simply altering the priors within the Bayesian hierarchical model. For example, Lu et al. (2016) proposed to use the slab-and-spike prior (Mitchell & Beauchamp, 1988) in Bayesian factor analysis. This prior is a discrete mixture of an uninformative normal distribution (the slab), and either the Dirac delta function concentrated at zero or a normal distribution with small variance centered around zero (the spike; Lu et al., 2016; van Erp et al., 2019). The slab-and-spike prior works to remove small factor loadings by shift-

ing their corresponding estimates toward zero (i.e., the spike), and retaining large factor loadings by shifting them toward the part of the prior containing the slab. When using the Dirac delta function instead of the mixture of normals, it is possible to completely eliminate coefficients with small and unimportant factor loadings (Lu et al., 2016). Eliminating features can be advantageous in the case of uncertainty about the estimated factor loading values. For example, in our applications we applied an arbitrary criterion of 0.1 (although this value has been widely accepted; Feng et al., 2017a, 2017b; Guo et al., 2012; Hoti & Sillanpää, 2006), but setting the criterion to zero would avoid the arbitrary nature of choosing a criterion. In summary, many other regularization methods could be investigated using the model structure presented here by simply adjusting the prior appropriately. The proposed Lasso application should be understood as a first attempt to apply a shrinkage estimation technique to the joint modeling approach of linking brain and behavior, and we look forward to further applications of other regularization methods.

7.2 The paradoxical advantage of shrinkage

In the Bayesian hierarchical modeling, shrinkage is the tendency for a parameter to more closely resemble the prior information imposed by the upper-level structure rather than that of the likelihood. Of course, an ideal method of inference would allow the data to speak prominently in the parameter estimation process, and so shrinkage due to the hierarchical structure would appear to be a negative aspect of our research. However, hierarchical methods allow the data to control the prior through the hyperstructure that is informed by jointly estimating all model parameters. Hence, while shrinkage may appear to be problematic at the level of an individual subject, it is statistically optimal when considering the full set of subjects, and the relative importance of different levels of the hierarchical model (e.g., subject- versus group-level effects).

Regularization methods that exploit shrinkage trade a small amount of bias and a

large standard error to obtain a large reduction in the mean squared error (MSE). Hence, regularization methods generally produce more stable and reliable estimators with reasonably small biases. The Lasso method in this article lowered the MSE to a large extent compared to the FA NDDM without the Lasso. This was an expected outcome. However, it was unexpected that the Lasso FA NDDM would also have smaller biases and smaller standard errors in the factor loading estimates. This result occurs because the original FA NDDM imposed shrinkage effects on the single-trial diffusion model parameters (i.e., factor scores) from the model's hierarchical structure. The shrinkage of these model parameters then propagated to the factor loading estimates, biasing the results. Applying the Lasso to the FA NDDM seems to correct the biases of the factor loading estimates. Although this result may be limited to the current study, it is possible that regularization methods can remedy the bias that shrinkage causes within the hierarchical models. We save the formalization of a precise statistical and mathematical mechanism that explains this result for future research.

7.3 Generalization of Lasso FA NDDM

Although the present article considers the use of regularization techniques on the identification of brain networks, this particular application was only a case study. In general, any type of covariate can be used within the FA NDDM, and the Lasso method can also be applied. Furthermore, the application of FA NDDM is not restricted to a single covariate. For example, Turner et al. used a joint model to simultaneously model EEG, fMRI, and choice response time data from an intertemporal choice task. Although Turner, Wang, and Merkle's approach did not use a factor structure, such modeling efforts are now well within reach. As another example, subjects' personal attributes and clinical measures could also be exploited to further constrain the model (e.g., dysphoria and tendency to ruminate, Vandekerckhove, 2014). Because linking more data sets requires more complex models to deal with information from multiple sources, it is possible that

these other sources could obscure networks of interesting variables, hindering our understanding of the underlying latent processes. The Lasso method proposed here should be a useful tool to help identify important features of psychological phenomena from noisy conglomerates of data.

8 Conclusions

There is a tremendous amount of data in the fields of psychology and neuroscience, but very little theory that can be used to explain why patterns of (especially neural) data are dramatically altered across experimental conditions, individual subjects, or even across time. Until such a unified theory of the brain exists, we can use computational models to examine how experimental manipulations affect neural and behavioral data. In the field of mathematical psychology, the DDM (and other sequential sampling models) has provided the theoretical groundwork necessary to explain how differences in behavioral data emerge as a function of individual differences, intelligence quotients, or experimental manipulations. Our goal was to build a computational framework that would enable such a model to transcend behavioral data and establish context when interpreting complicated patterns in high-dimensional neural data. Such a framework should help to isolate the aforementioned effects and build toward a unified theory of brain-behavior dynamics.

References

Bae, K., & Mallick, B. K. (2004). Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, *20*(18), 3423-3430. doi: 10.1093/bioinformatics/bth419

Bahg, G., Evans, D. G., Galdo, M., & Turner, B. M. (in press). *Gaussian process linking functions for mind, brain, and behavior.* (In press at *Proceedings of the National Academy of Sciences*)

Choi, J., Zou, H., & Oehlert, G. (2010). A penalized maximum likelihood approach to sparse factor analysis. *Statistics and its Interface*, *3*, 429–436.

Erosheva, E. A., & Curtis, S. M. (2017). Dealing with Reflection Invariance in Bayesian Factor Analysis. *Psychometrika*, *82*(2), 295-307. doi: 10.1007/s11336-017-9564-y

Feng, X.-N., Wu, H.-T., & Song, X.-Y. (2017a). Bayesian adaptive lasso for ordinal regression with latent variables. *Sociological Methods & Research*, *46*(4), 926-953. doi: 10.1177/0049124115610349

Feng, X.-N., Wu, H.-T., & Song, X.-Y. (2017b). Bayesian regularized multivariate generalized latent variable models. *Structural Equation Modeling: A Multidisciplinary Journal*, *24*(3), 341-358. doi: 10.1080/10705511.2016.1257353

Figueiredo, M. A. T. (2003). Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(9), 1150-1159. doi: 10.1109/TPAMI.2003.1227989

Forstmann, B. U., & Wagenmakers, E. J. (2015). *An introduction to model-based cognitive neuroscience*. Springer.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *Elements of statistical learning*. Springer.

Gelman, A. (1996). Inference and monitoring convergence. In W. R. Gilks, S. Richardson, & D. J. Spiegelhalter (Eds.), *Markov chain monte carlo in practice* (p. 131-143). CRC Press.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., & A. Vehtari, D. B. R. (2013). *Bayesian data analysis* (3rd ed.). CRC Press.

Geweke, J., & Zhou, G. (1996). Measuring the Pricing Error of the Arbitrage Pricing Theory. *The Review of Financial Studies*, *9*(2), 557-587. doi: 10.1093/rfs/9.2.557

Ghosh, J., & Dunson, D. B. (2009). Default prior distributions and efficient posterior computation in bayesian factor analysis. *Journal of Computational and Graphical Statistics*, *18*(2), 306-320. (PMID: 23997568) doi: 10.1198/jcgs.2009.07145

Guo, R., Zhu, H., Chow, S.-M., & Ibrahim, J. G. (2012). Bayesian lasso for semiparametric structural equation models. *Biometrics*, *68*(2), 567-577. doi: 10.1111/j.1541-0420.2012.01751.x

Hirose, K., & Konishi, S. (2012). Variable selection via the weighted group lasso for factor analysis models. *Canadian Journal of Statistics*, *40*(2), 345-361. doi: 10.1002/cjs.11129

Hirose, K., & Yamamoto, M. (2015). Sparse estimation via nonconcave penalized likelihood in factor analysis model. *Statistics and Computing*, *25*(5), 863-875.

Hoerl, A. E., & Kennard, R. W. (1970a). Ridge regression: Applications to nonorthogonal problems. *Technometrics*, *12*(1), 69-82. doi: 10.1080/00401706.1970.10488635

Hoerl, A. E., & Kennard, R. W. (1970b). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67. doi: 10.1080/00401706.1970.10488634

Hoff, P. D. (2009). *A first course in bayesian statistical methods.* Springer.

Hoti, F., & Sillanpää, M. (2006). Bayesian mapping of genotype x expression interactions in quantitative and qualitative traits. *Heredity*, *97*(1), 4-18.

Houseman, E. A., Marsit, C., Karagas, M., & Ryan, L. M. (2007). Penalized item response theory models: Application to epigenetic alterations in bladder cancer. *Biometrics*, *63*(4), 1269-1277.

Jacobucci, R., Grimm, K., & McArdle, J. (2016, 13). Regularized structural equation modeling. *Structural Equation Modeling*, *23*(4), 1-12. doi: 10.1080/10705511.2016.1154793

Jung, S., & Takane, Y. (2008). Regularized exploratory factor analysis. In K. Shigemasu, A. Okada, T. Imaizumi, & T. Hoshino. (Eds.), *New trends in psychometrics* (Vol. 141-

149). Tokyo: University Academic Press.

Keuken, M. C., Müller-Axt, C., Langner, R., Eickhoff, S. B., Forstmann, B. U., & Neumann, J. (2014). Brain networks of perceptual decision-making: an fmri ale meta-analysis. *Frontiers in human neuroscience*, *8*, 445.

Li, Y., Nan, B., & Zhu, J. (2015). Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. *Biometrics*, *71*, 354-363.

Lu, Z.-H., Chow, S.-M., & Loken, E. (2016). Bayesian factor analysis as a variable-selection problem: Alternative priors and consequences. *Multivariate Behavioral Research*, *51*(4), 519-539. (PMID: 27314566) doi: 10.1080/00273171.2016.1168279

Magis, D., Tuerlinckx, F., & Boeck, P. D. (2015). Detection of differential item functioning using the lasso approach. *Journal of Educational and Behavioral Statistics*, *40*(2), 111-135. doi: 10.3102/1076998614559747

Mitchell, T. J., & Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, *83*(404), 1023-1032.

Mulder, M., Van Maanen, L., & Forstmann, B. (2014). Perceptual decision neurosciences–a model-based review. *Neuroscience*, *277*, 872–884.

Muthén, B., & Asparouhov, T. (2012). Bayesian structural equation modeling: A more flexible representation of substantive theory. *Psychological Methods*, *17*(3), 313-335.

Ning, L., & Georgiou, T. T. (2011). Sparse factor analysis via likelihood and l_1-regularization. In *2011 50th ieee conference on decision and control and european control conference* (p. 5188-5192). doi: 10.1109/CDC.2011.6161415

Palestro, J. J., Bahg, G., Sederberg, P. B., Lu, Z.-L., Steyvers, M., & Turner, B. M. (2018). A tutorial on joint models of neural and behavioral measures of cognition. *Journal of Mathematical Psychology*, *84*, 20 - 48. doi: 10.1016/j.jmp.2018.03.003

Park, T., & Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, *103*(482), 681-686. doi: 10.1198/016214508000000337

Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D.-Y., Pollack, J. R., & Wang, P. (2010).

Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Ann. Appl. Stat.*, *4*(1), 53–77. doi: 10.1214/09-AOAS271

Polson, N. G., & Scott, J. G. (2011, 19). Shrink globally, act locally: Sparse bayesian regularization and prediction. In *Bayesian statistics 9.* Oxford University Press. doi: 10.1093/acprof:oso/9780199694587.003.0017

Polson, N. G., & Scott, J. G. (2012). On the half-cauchy prior for a global scale parameter. *Bayesian Analysis*, *7*(4), 887-902.

Price, B. S., & Sherwood, B. (2018). A cluster elastic net for multivariate regression. *Journal of Machine Learning Research*, *18*(232), 1-39.

Rae, B., Heathcote, A., Donkin, C., Averell, L., & Brown, S. (2014). The hare and the tortoise: Emphasizing speed can change the evidence used to make decisions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *40*(5), 1226–1243. doi: /10.1037/a0036801

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*(2), 59-108.

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: Theory and data for two-choice decision tasks the diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, *20*(4), 873-922.

Ravishanker, N., & Dey, D. K. (2001). *A first course in linear model theory*. Chapman & Hall.

Rothman, A. J., Levina, E., & Zhu, J. (2010). Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, *19*(4), 947-962. (PMID: 24963268) doi: 10.1198/jcgs.2010.09188

Song, X., & Lee, S.-Y. (2012). A tutorial on the bayesian approach for analyzing structural equation models. *Journal of Mathematical Psychology*, *56*(3), 135 - 148. doi: https://doi.org/10.1016/j.jmp.2012.02.001

Song, X., Lu, Z., & Feng, X. (2014). Latent variable models with nonparametric interaction effects of latent variables. *Statistics in Medicine*, *33*(10), 1723-1737. doi: 10.1002/

sim.6065

Ter Braak, C. J. F. (2006). A markov chain monte carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces. *Statistics and Computing*, *16*, 239-249.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.

Turner, B. M., Forstmann, B. U., Love, B. C., Palmeri, T. J., & Maanen, L. V. (2017). Approaches to analysis in model-based cognitive neuroscience. *Journal of Mathematical Psychology*, *76*, 65 - 79. (Model-based Cognitive Neuroscience) doi: 10.1016/j.jmp.2016.01.001

Turner, B. M., Forstmann, B. U., & Steyvers, M. (2019). *Joint models of neural and behavioral data*. Springer.

Turner, B. M., Forstmann, B. U., Wagenmakers, E.-J., Brown, S. D., Sederberg, P. B., & Steyvers, M. (2013). A bayesian framework for simultaneously modeling neural and behavioral data. *NeuroImage*, *72*, 193 - 206. doi: https://doi.org/10.1016/j.neuroimage.2013.01.048

Turner, B. M., Palestro, J. J., Miletić, S., & Forstmann, B. U. (2019). Advances in techniques for imposing reciprocity in brain-behavior relations. *Neuroscience & Biobehavioral Reviews*, *102*, 327 - 336. doi: 10.1016/j.neubiorev.2019.04.018

Turner, B. M., Rodriguez, C. A., Norcia, T. M., McClure, S. M., & Steyvers, M. (2016). Why more is better: Simultaneous modeling of eeg, fmri, and behavioral data. *NeuroImage*, *128*, 96 - 115. doi: 10.1016/j.neuroimage.2015.12.030

Turner, B. M., Sederberg, P. B., Brown, S. D., & Steyvers, M. (2013). A method for efficiently sampling from distributions with correlated dimensions. *Psychological Methods*, *18*(3), 368-384.

Turner, B. M., van Maanen, L., & Forstmann, B. U. (2015). Informing cognitive abstractions through neuroimaging: The neural drift diffusion model. *Psychological Review*,

122(2), 312-336.

Turner, B. M., Wang, T., & Merkle, E. C. (2017). Factor analysis linking functions for simultaneously modeling neural and behavioral data. *NeuroImage*, *153*, 28-48. doi: 10.1016/j.neuroimage.2017.03.044

Vandekerckhove, J. (2014). A cognitive latent variable model for the simultaneous analysis of behavioral and personality data. *Journal of Mathematical Psychology*, *60*, 58-71. doi: 10.1016/j.jmp.2014.06.004

van Erp, S., Oberski, D. L., & Mulder, J. (2019). Shrinkage priors for bayesian penalized regression. *Journal of Mathematical Psychology*, *89*, 31 - 50. doi: 10.1016/j.jmp.2018.12.004

van Maanen, L., Brown, S. D., Eichele, T., Wagenmakers, E.-J., Ho, T., Serences, J., & Forstmann, B. U. (2011). Neural correlates of trial-to-trial fluctuations in response caution. *Journal of Neuroscience*, *31*(48), 17488–17495. doi: 10.1523/JNEUROSCI.2924-11.2011

Wabersich, D., & Vandekerckhove, J. (2014). Extending jags: A tutorial on adding custom distributions to jags (with a diffusion model example). *Behavior Research Methods*, *46*, 15-28.

Wang, Z., Lane, A., Chakraborty, S., & Wood, P. (2013). Bayesian elastic-net and fused lasso for semiparametric structural equation models. *NBER-NSF Seminar on Bayesian Inference in Econometrics and Statistics*, *Conference paper*, 3-4.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49-67. doi: 10.1111/j.1467-9868.2005.00532.x

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *67*(2), 301–320.

Appendix A: Robustness check for different prior specifications of the global penalty parameter

The effectiveness of the Lasso depends on the global penalty parameter κ, and different priors for κ can produce considerably different MAP estimates of the factor loading matrix. To ensure that our results in Section 6 were not overly sensitive to the specification for the prior on κ, we also conducted a sensitivity analysis by specifying four different priors for κ, and refitting the Lasso FA NDDM. In the previous applications (Feng et al., 2017a, 2017b; Guo et al., 2012; Song et al., 2014; Wang et al., 2013), a gamma prior has been used successfully such that $\kappa^2 \sim Gamma(\alpha_{0\kappa},\ \beta_{0\kappa})$, where $\alpha_{0\kappa}$ is commonly set to 1 as a means to specify an uninformative prior. Previous sensitivity results have examined

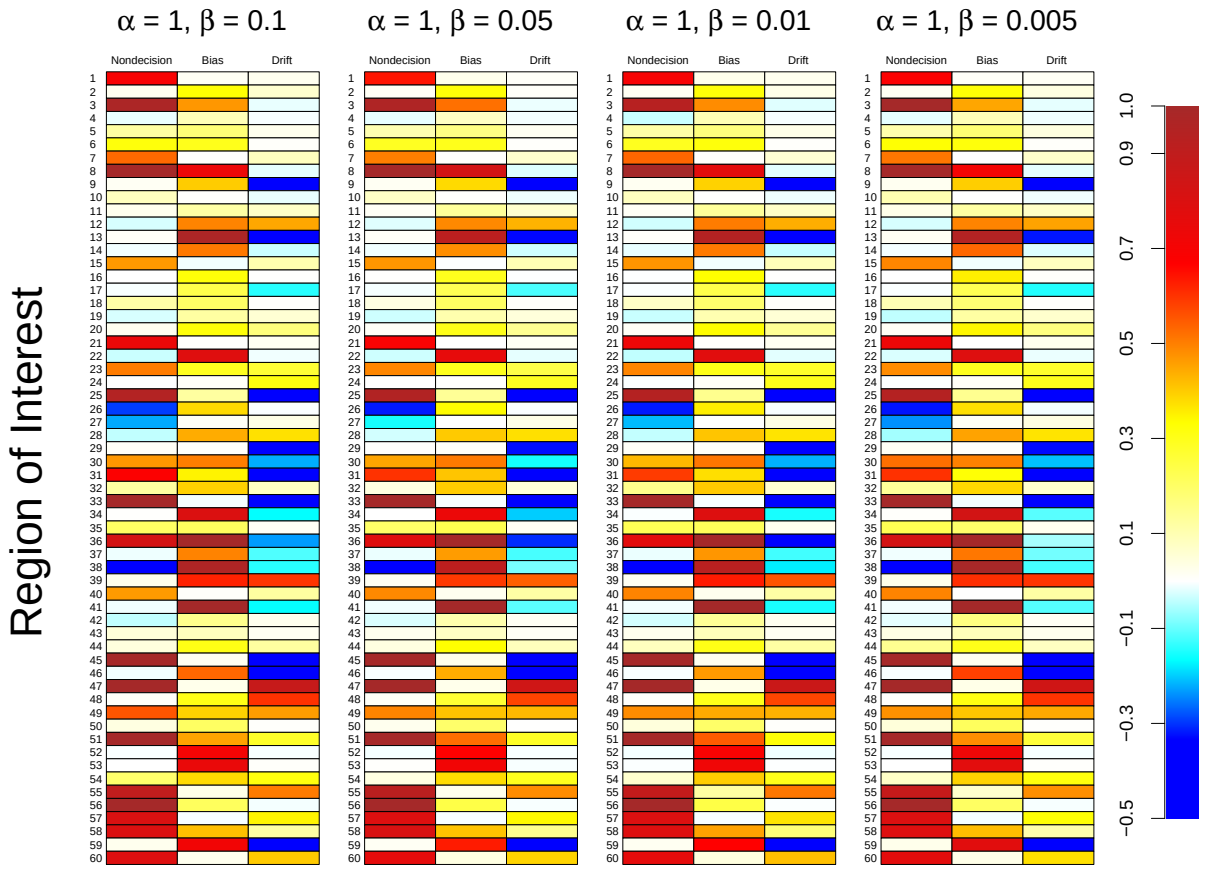

Figure 16: **Effects of Penalty Hyperparameters.** Each panel shows the estimated factor loading matrix obtained from the accuracy condition under four different hyperparameter settings for the penalty term.

the role of $\beta_{0\kappa}$ by setting it to the following values: 0.1, 0.05, 0.01, and 0.005.

Figure 16 shows the estimated factor loading matrices under four different settings for $\beta_{0\kappa}$ in the gamma prior: $\beta_{0\kappa} = \{0.1, 0.05, 0.01, 0.005\}$. As in previous figures, each value of the factor loading matrix is color coded according to the legend on the right-hand side. Across all settings for $\beta_{0\kappa}$, Figure 16 shows that the estimated factor loading matrices are quite similar. In large part, these factor loading results are due to only small differences in the MAP estimates obtained for κ, which were 5.603 ($\beta_{0\kappa} = 0.1$), 5.393 ($\beta_{0\kappa} = 0.05$), 5.320 ($\beta_{0\kappa} = 0.01$), and 5.301 ($\beta_{0\kappa} = 0.005$). These results assure us that our conclusions are insensitive to our choice of prior, and that the estimate of the penalty term κ is determined largely from the data rather than the prior.

A Copyright Notice

Supplementary Materials

Supplementary materials to this article can be found online at `https://github.com/MbCN-lab/LassoFANDDM`.