Computational Brain & Behavior (2018) 1:184–213
https://doi.org/10.1007/s42113-018-0013-5

Hierarchical Bayesian Analyses for Modeling BOLD Time Series Data

M. Fiona Molloy[1] · Giwon Bahg[1] · Xiangrui Li[1] · Mark Steyvers[2] · Zhong-Lin Lu[1] · Brandon M. Turner[1]

Published online: 27 September 201 8
© Springer Nature Switzerland AG 2018

Abstract

Hierarchical Bayesian analyses have become a popular technique for analyzing complex interactions of important experimental variables. One application where these analyses have great potential is in analyzing neural data. However, estimating parameters for these models can be complicated. Although many software programs facilitate the estimation of parameters within hierarchical Bayesian models, due to some restrictions, complicated workarounds are sometimes necessary to implement a model within the software. One such restriction is convolution, a technique often used in neuroimaging analyses to relate experimental variables to models describing neural activation. Here, we show how to perform convolution within the R programming environment. The strategy here is to pass the convolved neural signal to existing software package for fitting hierarchical Bayesian models to data such as JAGS (Plummer 2003) or Stan (Carpenter et al. 2017). We use the convolution technique as a basis for describing neural time series data and develop five models to describe how subject-, condition-, and brain-area-specific effects interact. To provide a concrete example, we apply these models to fMRI data from a stop-signal task. The models are assessed in terms of model fit, parameter constraint, and generalizability. For these data, our results suggest that while subject and condition constraints are important for both fit and generalization, region of interest constraints did not substantially improve performance.

Keywords Stop-signal · Hierarchical · Bayesian · Modeling · Functional magnetic resonance imaging

Introduction

Analyzing neural data can be quite complicated as its quality can be affected by a variety of confounding factors, such as motion, technological limitations, and patterns of activation that are unrelated to the experimenter's interests. As a result, many data analytic strategies have been developed to sieve the important characteristics (i.e., "signal") from neural data, which are often obscured by sources of variance that are unsystematically related to the independent variables of the experiment (i.e., "noise"). It is particularly challenging to purify the signal within functional magnetic resonance imaging (fMRI) data. For example, preprocessing fMRI data often includes motion correction, slice timing correction, mapping to a standardized space, and spatial smoothing. Even after all of these preprocessing steps, it can be difficult to extract meaningful signal from fMRI data, due to noise and perhaps more importantly, the slow nature of the hemodynamic lag. As a consequence, it can be difficult to quantify statistical evidence or precisely identify the neural basis of cognitive functions.

One convenient way to purify data quality while simultaneously quantifying statistical evidence is through hierarchical Bayesian modeling (Rouder and Lu 2005; Lee 2008; Shiffrin et al. 2008; Ahn et al. 2011; Turner et al. 2013). Within a hierarchical model, experimental effects can be conceptualized as having a common basis, yet still allowing for departures that are intrinsic to particular factors. For example, subjects performing a task in a typical experiment will undoubtedly have some commonalities: they may have similar age ranges, they are all performing the same task, and they may all be mentally healthy. Yet, they may also have individual characteristics that allow them to perform systematically differently from all other subjects in our task. Hierarchical models allow us to balance these different factors in a statistically principled manner. Also, hierarchical Bayesian models improve parameter estimates for a single trial by introducing additional constraints on the estimates and reducing uncertainty. Furthermore, Bayesian statistics allow us to quantify the

✉ Brandon M. Turner
turner.826@gmail.com

[1] Department of Psychology, The Ohio State University, Columbus, USA

[2] Department of Cognitive Sciences, University of California, Irvine, USA

effects in a way that is exclusively influenced by the data from our current experiment, as well as our prior beliefs about the relative magnitudes of the effects.

Despite the many advantages that hierarchical Bayesian methods provide, they can be complicated to implement. Fortunately, many statistical software packages facilitate Bayesian estimation, enabling Bayesian solutions to otherwise impenetrable analysis questions. However, because many software packages are highly restrictive in the models they can fit to data, sometimes complicated workarounds are necessary to perform more complex analyses. One area where extant software packages have not yet been successfully applied is in the analysis of fMRI data. Although there are potentially a few different reasons (see General Discussion), we believe that a major limiting factor is the lack of accessibility and clarity about how to relate the experimental design to the neural data acquired from the scanner. To accomplish this, a mathematical process called convolution is needed to combine the details of the stimulus presentation with mathematical models that describe hypothesized neurophysiological responses to the stimuli. In many papers, the equations detailing how to perform convolution are somewhat obscured, making it difficult to imagine what the data generating model is and how we can fit the model to data. Although there are software packages available for implementing Bayesian analyses for fMRI data, such as those built into FSL (Chappell et al. 2009) or SPM (Han and Park 2018), these packages are difficult to customize and are unfortunately not hierarchical. All of these complications make it difficult for any aspiring modeler to analyze neural data, and for example, link the parameters of interest to a cognitive model (Turner et al. 2013, 2015, 2016, 2018).

Here, we present a more accessible introduction to modeling neural activation by providing concrete convolution equations, and connecting these equations with computer code written in R. The benefit of this approach is that once the design matrix for the experimental data is computed, it can easily be passed to user-friendly Bayesian software programs such as Just Another Gibbs Sampler (Plummer 2003, JAGS;) or Stan (Carpenter et al. 2017). These programs make Bayesian inference quite convenient, as several algorithms facilitate efficient and accurate estimation of the model parameters (e.g., Markov chain Monte Carlo). We then develop five models of neural time series data that increase in complexity in terms of the number of effects that they are able to accommodate. Although the models are intended to be generally applicable, we apply them to data from a stop-signal task to provide a concrete example of implementation. Within our task, we assess the models' ability to both fit data and to generalize to new data through out-of-sample prediction analyses. Hence, the present goal of this article is to show the importance of condition- and subject-level constraints in the stop-signal task, while providing the tools necessary for others to carry out hierarchical Bayesian analyses of fMRI time series data in their own research.

The outline of the article is as follows. First, we give a brief overview of the literature on Bayesian analysis of fMRI data, and of the literature on response inhibition. Second, we explain how the general linear model analysis used in fMRI can be implemented in a Bayesian framework. Here, we explain how convolution is done effectively for two types of experimental designs. Third, we develop five models of neural time series data. An advantage of these models, and a deviation from typical fMRI analyses, is that they extract estimates of neural activation for each trial/ stimulus, as opposed to a single estimate averaged across trials for each condition. Fourth, we describe our stop-signal task and the fMRI procedures. Fifth, we present model fitting results from one run of the stop-signal task. Sixth, we use an out-of-sample prediction analysis to test the generalizability of the models across different runs of the stop-signal task. We conclude with a summary of the results and a discussion of limitations and future directions.

Bayesian Analyses of Neural Activation

One of the most standard methods in fMRI data analysis is the general linear model (GLM). The GLM can be considered a multiple linear regression model applied to fMRI time series data, where the data are modeled as a linear combination of factors such as the condition, stimulus, and baseline level of activation. In addition, task-irrelevant factors such as scanner drift, physiological noise, and autocorrelation due to hemodynamic properties are sometimes modeled (Friston et al. 1995). Although there are scenarios where the linear modeling assumptions necessary for the GLM do not hold for fMRI data (Monti 2011; Poline and Brett 2012), the GLM approach still remains the most frequently used data analysis method due to its simplicity and approachability. However, fMRI analysis—particularly the GLM—could be improved using Bayesian statistics, where the uncertainty of model parameters must be quantified in terms of posterior distributions. Although fMRI analysis has been predominantly frequentist, we certainly are not the first to suggest using Bayesian statistics to improve fMRI analysis. Bayesian techniques such as spatial priors, adaptive priors, and model comparisons have been applied to fMRI research in various ways over the past few decades (for a review, see Zhang et al. (2015)).

A natural extension of a Bayesian GLM applied to a single subject is to introduce a hierarchical structure. Although it is a technical point, extensions to hierarchical models are relatively easy in the Bayesian framework, compared to frequentist statistics. This is because Bayesian statistics factorize multi-dimensional model parameters into a series of conditional probabilities that depend on the structure of the model (Kruschke 2014; Woolrich 2012). By building a hierarchical layer, the information extracted from one subject's data helps to constrain the inference process for other subjects. This information sharing allows subject-specific effect parameters to be less affected by random noise, because each parameter is affected by both the subject's data and the information learned across the group (Kruschke 2014). Given these properties, one can view the addition of a hierarchical structure for an fMRI time series as an extension of the standard GLM; whereas the standard GLM may only estimate condition-level effects, adding a hierarchy enables the model to capture both condition-level effects and stimulus-specific effects.

We are also not the first to use *hierarchical* Bayesian techniques to analyze fMRI data. For example, hierarchical Bayesian analyses have been used to impose spatial constraints on voxel-level analyses (Bowman et al. 2008). Although the analyses presented in this article are based on clusters of voxels that comprise regions of interest (ROIs), the methods presented here can easily be integrated into existing pipelines for voxel-based fMRI analyses (see the General Discussion).

Unfortunately, Bayesian approaches to neuroscience are complex. With such complicated methods, it is difficult to find a balance between ease of implementation and ability to customize. Many Bayesian estimation programs are home-brewed, which are complicated to write. Others are deeply imbedded within software packages, and do not allow the user to tailor the models to their experimental design and analytic goals. Another barrier of using Bayesian statistics is the computational burden. Programs such as JAGS and Stan have helped ease this burden, and also increased accessibility by creating a user-friendly working environment that can be managed within R. In this paper, we show how one can first perform convolution in R (which be translated to other programming languages such as Matlab or Python) and then pass the resulting convolved signal to your preferred Bayesian software package.

Measuring Response Inhibition

One domain where hierarchical Bayesian analysis can be helpful is when measuring response inhibition. Response inhibition is considered an important component of cognitive control (Miyake et al. 2000; Aron 2007; Logan 1985) and has interesting applications to individual differences (Miyake and Friedman 2012), attention deficit hyperactivity disorder (Nigg 2001; Schachar and Logan 1990), obsessive-compulsive disorder (Bannon et al. 2002; Penadés et al. 2007), and substance use disorders (Monterosso et al. 2005; Nigg et al. 2006). Additionally, tasks designed to measure response inhibition often produce many missing behavioral observations. For example, if a subject was instructed to withhold their response, a correct action would result in no behavioral data. To gain a better perspective on the cognitive processes that allowed for successful response inhibition, we can look to patterns of neural measures during the trial. In this sense, the issue of sparse or missing data can be circumvented by thorough analysis of brain data.

Response inhibition has been studied extensively in neuroscience and mathematical psychology. Considerable research has led to well-developed theories of response inhibition, in both of these fields. It is important to note that most of this research, with a few exceptions, is either focused entirely on neuroscience or modeling, without much overlap between the two fields. We begin a brief review of the response inhibition theories in neuroscience, primarily within the context of fMRI experiments.

Within neuroscience, cognitive control theories, in general, are based on the idea that fronto-parietal connectivity allows for cognitively regulatory abilities (Miller and Cohen 2001; Jung and Haier 2007). Additionally, individual differences found in these tasks arise from differences in fronto-parietal connectivity, or whole-brain connectivity to the prefrontal cortex (Cole et al. 2012). Response inhibition, including both stopping and not going, has been found to involve the right inferior frontal gyrus, presupplementary motor area, and the basal ganglia, although the distinct role of each area in the network is contested (Aron et al. 2014; Chikazoe et al. 2009; Sebastian et al. 2016; Verbruggen and Logan 2008).

In addition to these neural theories, response inhibition theories have been extensively modeled to examine the mechanisms of the behavior. Behavioral models of the stop-signal task are generally based on a race process, where go processes and stop processes are different "racing" accumulators (Logan and Cowan 1984). More recent models add features such as stochastic accumulators (Logan et al. 2014) and the ability to estimate entire stop signal reaction time distributions (Matzke et al. 2013). While many of these models focus on purely behavioral data, some have incorporated neuroscience by using single-unit neurophysiology in experiments involving nonhuman primates to constrain behavioral models and differentiate between competing theories (Logan et al. 2015; Boucher et al. 2007).

Two tasks commonly used to measure response inhibition are the go/no-go and stop-signal tasks. In the go/no-go task, subjects are instructed to respond to one stimulus (or set of stimuli), often by invoking a motor response (i.e., pressing a button), and not to respond to a different stimulus or set of stimuli. The stop-signal task extends this basic paradigm by adding a stopping condition, where a go signal is presented, but after a set delay, a "stop-signal" is presented. While the tasks both look at response inhibition, not going (in the go/no-go task) and stopping (in the stop-signal task) have overlapping, but not identical networks (Rubia et al. 2001; Swick et al. 2011). The development of a task that has both stopping and not going components extended upon this theory by confirming that both types of inhibition share a common network, but stopping and not going have distinct subprocesses within that network (Sebastian et al. 2013). The task we use in this paper has components of a go/no-go task built into the stop-signal task to further analyze these nuances in response inhibition. In this next section, we begin by detailing how to construct a time series of neural activity that can later be constrained by task-specific details.

Bayesian Implementation of the General Linear Model Analysis

Unlike many applications of Bayesian statistics for cognitive models, to perform Bayesian inference on neural activation, we must relate the stimulus effects to an entire time series worth of neural activity throughout the experiment. The reason for this is due to what is called *hemodynamic lag*, where the effects of an individual stimulus may linger for up to 30 s after stimulus presentation. These lingering effects can have a major impact on our ability to understand the systematic relationship that our independent variables have on brain activity for most realistic experimental designs where stimuli are presented within 30 s of one another. Hence, to properly estimate stimulus effects, we must carefully consider the specific sequence of stimulus presentations.

Consistent with other Bayesian cognitive modeling applications (Turner et al. 2017, 2018), we first define a generative model that describes how stimuli provoke neural activation for a given region of interest. This generative model is used to describe the effects of each stimulus in the experimental design through a set of activation parameters. Once each individual stimulus is described, we must combine these descriptions to form a *convolved* neural signal. The convolved neural signal is a prediction from our generative model, and so it can be compared to data observed from an experiment to assess how well our model prediction matches the observed data. Close matches indicate that the set of activation parameters are accurate, whereas poor matches indicate that at least a subset of activation parameters need to be adjusted. The process of inference is to adjust all of the activation parameters such that close matches can be obtained for all neural time series of interest.

Before providing all the details of our hierarchical models, we felt it is necessary to detail the core component of these models that relates each individual stimulus to a predicted time series of neural activity. While we have provided similar descriptions elsewhere (Palestro et al. 2018), the experimental design and purpose of the analyses below are both quite different. Furthermore, despite its simplicity, we find that most descriptions of convolution are somewhat difficult to apply generally. Thus, we provide a description of convolution in our specific task so that other researchers wishing to use these tools may have them readily accessible.

In this section, we describe how to construct a time series of neural activity in three parts. First, we describe the generative model of neural activity for a single stimulus. Second, we describe how critical experimental design variables should be collected and organized, to allow for formal mathematical descriptions of neural activity. Third, we describe how the experimental design variables can be used with the generative model for each specific stimulus to produce a predicted time series for neural activity. This operation is known as convolution, and we discuss two cases of convolution: finite impulse and the more general boxcar convolution. To aid in this discussion, we provide code in the Appendices that can implement the analyses we report here. Appendix A provides the means to calculate the canonical hemodynamic response function, Appendix B enables the user to specify the stimulus presentation for a specific experimental design, Appendix C provides the function for performing convolution, and finally, Appendix D provides the JAGS code for one of the five hierarchical models we investigate here.

Hemodynamic Responses and Convolution

Unfortunately, scientists have yet to develop a way of directly observing neural activity within the brain in a noninvasive way that is safe for humans. One of the current best approaches for measuring said neural activity is through indirect observation of metabolic changes in blood flow so that neurons consuming oxygen and glucose can be replenished. The onset of neural activity leads to a systematic series of local physiological changes, most importantly the change in concentration of oxyhemoglobin and deoxyhemoglobin (Poldrack et al. 2011). Ultimately, fMRI experiments measure these concentrations through the blood-oxygenation-level-dependent (BOLD) response. The BOLD response is intended to measure changes in brain activity as a dependent variable in our experiments: if the BOLD response in a given brain area is systematically related to the independent variable in our experiment, then it follows that the area is somehow being recruited to process the stimuli.

The BOLD response to a given stimulus is generally characterized by an increase to a peak level of activation, a decrease in activation below a baseline value, and then an asymptotic return to the baseline value (e.g., Fig. 1). To model the shape of these changes in the BOLD response through time, we can define what is called a hemodynamic response function (HRF). Fortunately, after many experiments, the field has settled on a few functional forms, each having advantages and disadvantages (Poldrack et al. 2011; Glover 1999; Boynton et al. 1996). One particular functional form that is commonly implemented in standard brain analysis software packages such as SPM 12 (http://www.fil.ion.ucl.ac.uk/spm/software/spm12) is the double-gamma model (Glover 1999; Boynton et al. 1996), given by

$$\begin{aligned} h(t) &= \beta h_0(t) \\ &= \beta\left(\frac{t^{a_1-1}{b_1}^{a_1}\exp(-b_1 t)}{\Gamma(a_1)} - c\frac{t^{a_2-1}{b_2}^{a_2}\exp(-b_2 t)}{\Gamma(a_2)}\right), \qquad (1) \end{aligned}$$

where t is time, β is the amplitude of the response, and $\Gamma(x) = (x-1)!$ denotes the gamma function. Many of the parameters in Eq. 1 are fixed to specific values by convention: $a_1 = 6$, $a_2 = 16$, $b_1 = 1$, $b_2 = 1$, and $c = 1/6$. These parameters define the shape of the HRF. While we could freely estimate these to reflect their variability across the brain or with stimulus types (Aguirre et al. 1998; Buckner 1998), they are often assumed to have a canonical form by using specific parameter values for simplicity in computation. Hence, the key parameter in Eq. 1 is β, as it defines how active a given voxel or region of interest is following a stimulus presentation. Appendix A provides R code that can be used to implement Eq. 1.

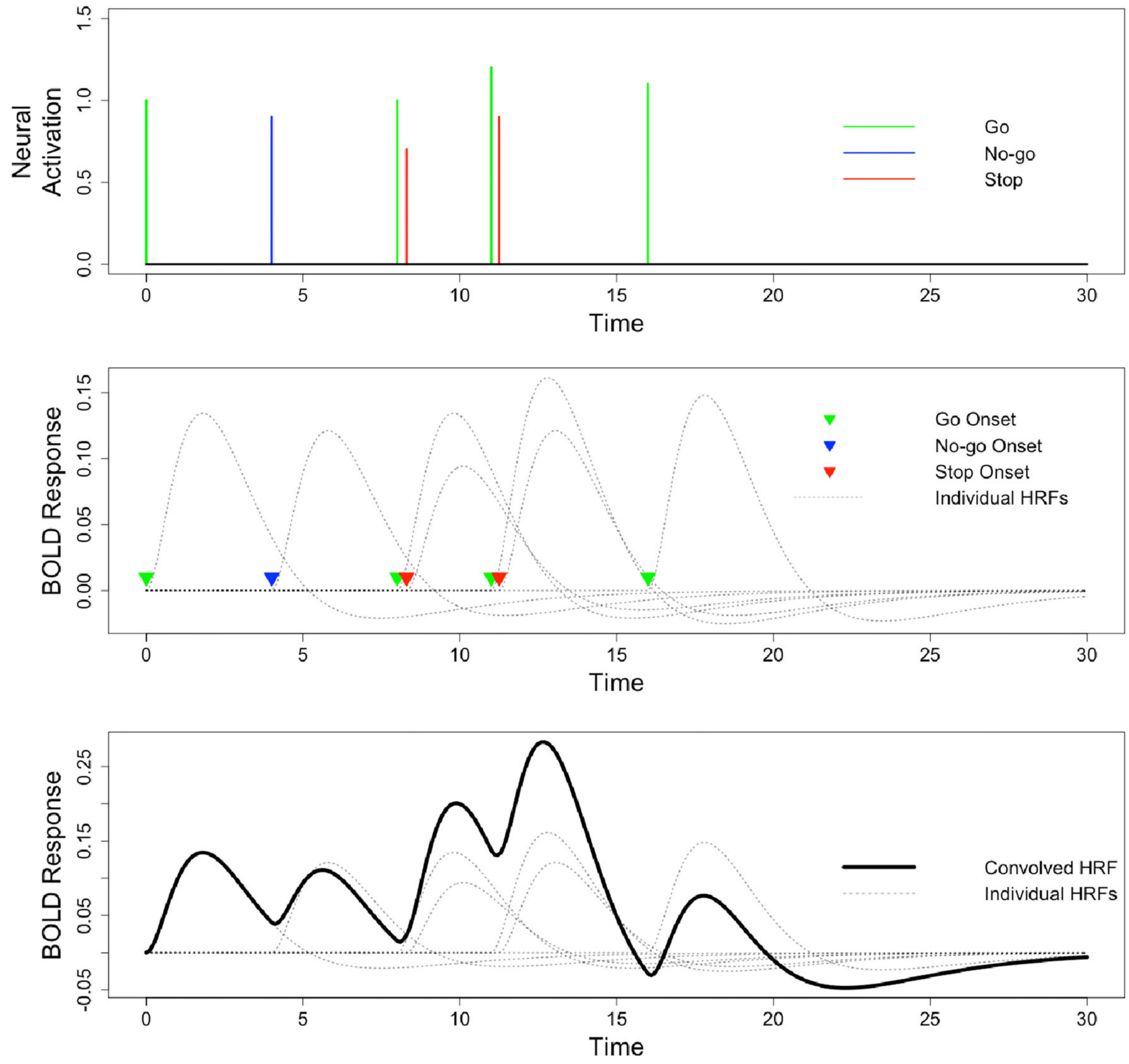

Fig. 1 Convolution of the hemodynamic response function (HRF). The process of generating a convolved signal of neural activity is separated into three panels for a hypothetical stop-signal task: parameters and design matrix (top row), individual HRFs for each stimulus (middle row), and a convolved HRF (bottom row). The top row shows neural activation for seven stimuli of three different trial types: a go trial (green), a no-go trial (blue), and a stop signal (red). The middle row shows how individual HRFs (dotted lines) are aligned to the specific sequence of stimuli (triangles of corresponding color), where each HRF is scaled according the β parameter represented in the top row (i.e., the heights of each bar). The bottom row shows how the convolved predicted signal (solid black line) is formed, with the individual HRFs from the middle row shown as a reference

Construction of a Design Matrix

To construct a prediction of how neural activity changes as a consequence of the stimulus variables, we must first organize variables that define the experimental environment. Specifically, it is important that we know (1) which stimulus was presented at what point in time and (2) how that particular stimulus should affect the neural activity. As experimenters, we easily have access to (1), and we can organize this information within a "design" matrix **X**. The design matrix **X** must be constructed by setting individual regressors for each level of the independent variable, and some specification must be made for each trial (Rissman et al. 2004; Mumford et al. 2012). **X** has a row for each point in time a neural measure is collected (i.e., repetition time), a column for each stimulus effect, and a column for baseline activation (analogous to a y-intercept term in linear regression). Within **X**, each column is constructed by shifting the template HRF in Eq. 1 to the point in time at which a stimulus was presented. More technically, the HRF must be convolved with what is called an impulse (response) function that defines the points in time a stimulus was presented.

While **X** can be defined from the experimental design and a template HRF, we now must specify the magnitude of each stimulus effect on neural activity. To do this, we can define an amplitude coefficient vector β such that each stimulus can be scaled according to Eq. 1. In general, we will not know the values of β that will scale each HRF appropriate to perfectly match our data. Instead, we must estimate the the values in β using, in our applications, Bayesian inference. In particular, we will show how the single-stimulus estimates can be constrained by building up hierarchical structures to explain neural activity.

The top and middle panels of Fig. 1 illustrate how the design matrix and amplitude coefficient vectors interact within 30 s of a hypothetical stop-signal task. The top row shows points in time at which seven stimuli of three different trial types were presented: go trials are shown in green, no-go trials are shown in blue, and stop-signals are shown in red. Each bar in the top panel corresponds to a column within **X**, and the location of each bar with respect to time corresponds to the row within **X** where an individual HRF would eventually be placed. The height of each bar corresponds to the values of the parameters in β. For this example, β includes stimulus-wise activation coefficients:

$$\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_7]^\mathsf{T} .$$

The middle panel of Fig. 1 illustrates how the HRFs in Eq. 1 are shifted according to the stimulus presentation sequence. Each bar in the top panel has a corresponding triangle, as a reference, in the middle panel. In this example, each stimulus has a corresponding HRF in the middle panel, and each HRF is scaled according to the heights of the bars in the top panel (i.e., the values of β). The middle panel of Fig. 1 shows that the individual HRFs for this particular stimulus sequence—which was actually taken from our experimental design below—clearly overlap through the experimental session. Therefore, any attempt to estimate individual β parameters would be hopeless without considering the entire time series.

Fortunately, the BOLD response exhibits a linear time invariant (LTI) property that enables us to isolate the effects of each individual stimulus presentation on the observed neural time series (Boynton et al. 1996). The first result implied by the LTI property is that if neural activation follows a stimulus presentation but peak activation is delayed by some arbitrary amount, the BOLD response will also start at the time of the stimulus presentation and be delayed by the same amount. This implies that if an appropriate function is specified for neural activation, it can be shifted to the time at which a given stimulus is presented. This property is illustrated in the middle panel of Fig. 1 where each HRF is shifted to the stimulus onset. The second result implied by the LTI property is that the scale parameter β (height of each HRF in the second row) is linearly related to amplitude of the neural activation (height of each line in the first row). In other words, trials with larger neural activation will also have larger β values. The final implication of the LTI property is in regards to how the individual HRFs are combined to form one convolved signal, which we now discuss.

Convolution

To impose the HRF expected for our particular experimental design, we require a mathematical operation called convolution. Convolution is usually described as an integration of two functions, one of which defines the shape of the HRF $h(t)$, and the other defines the times at which the stimuli are presented $f(t)$. The shape of the HRF $h(t)$ was defined in Eq. 1. Conceptually, it is easy to define a function $f(t)$ as an indicator function that specifies when a stimulus is present as

$$f(t) = \begin{cases} 1 & \text{when a stimulus is presented at time } t \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The convolution of $f(t)$ with $h(t)$ is defined as

$$(f * h)(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau. \tag{3}$$

Here, the dummy variable τ allows us to slide the HRF along the time axis until we reach the value of τ such that $f(\tau) = 1$. In other words, the variable τ is completely inconsequential as it is integrated out over the temporal axis. At this location, $(f * h)(t)$ becomes the HRF shifted by τ, as shown for each individual HRF in the middle panel of Fig. 1.

Equation 3 considers the case of any generic function for $f(t)$ and all continuous values of time t. However, there are a few things we can do to simplify Eq. 3 for the purposes of pedagogy and pragmatic applications. First, we can assume a discrete representation for time, rather than a continuous one. This assumption is justified because fMRI scanners are incapable of producing measures of brain activity that are practically continuous through time (i.e., they usually provide a set of measures from the whole brain every 1.5 or 2 s). With a discrete representation in place, we can use summation rather than integration in Eq. 3. Second, depending on the duration of stimulus presentation, $f(t)$ may take different forms. The simplest case is when $f(t) = 1$ for only one point in time t for a given stimulus, which is called an impulse function (e.g., the top row of Fig. 1). The more general case is when $f(t) = 1$ for a period of time, such as a few seconds. In this case, the function is often called a boxcar function. We begin with a description of convolution using an impulse function because it is conceptually simpler, and then move to the more general form of boxcar convolution.

Impulse Convolution

The top panel of Fig. 1 shows the simplest case of $f(t)$, where stimuli are shown and removed immediately, such that $f(t) = 1$ for a single point in time t for each stimulus. Because of the impulse function, the convolution operation in Eq. 3 is equivalent to simply shifting the starting point of the HRF $h(t)$, without additional scaling by β. For convenience, we define $h_{0,i}(t)$ as the HRF corresponding to the ith stimulus presentation; that is, $h_{0,i}(t)$ contains the same information in $h(t)$, but is shifted in time to correspond to the ith stimulus presentation. Using a discrete representation of time and an impulse function, we can perform convolution by combining these individual HRFs $h_{0,i}(t)$ in the following way:

$$\begin{aligned}\mathbf{N}(t) &= \beta^0 + \sum_{i=1}^{R} \beta_i h_{0,i}(t) \\ &= \beta^0 + (\mathbf{X}\boldsymbol{\beta})(t), \end{aligned} \tag{4}$$

where R is the total number of stimulus presentations, and β^0 denotes an intercept term that shifts the time series throughout the experiment. The bottom panel of Fig. 1 shows the predicted convolved neural activity $\mathbf{N}(t)$ from the model across the 30-s time window (solid black line), along with the individual HRFs from the middle panel as a reference. Because the third result of the LTI property suggests that the aggregated neural signal is a linear combination of the individual stimulus effects, we can use Eq. 4 to generate the convolved signal. Here, each individual effect is described by a separate HRF, and these individual HRFs from the middle panel are integrated together (i.e., through summation) to form the convolved signal.

If we wish to estimate the parameters β with software packages such as JAGS or Stan, one strategy is to first construct the design matrix $\mathbf{X}$ in our home environment (e.g., R or MATLAB) so that we can pass the constructed variable to the Bayesian software package of our choice. Because the design matrix only needs to be computed once, we can use advanced techniques (e.g., the fast Fourier transform) to aid in the computation. Although we do not investigate these methods here, this may be an important consideration for different problems. Once the functions in Appendices A-C are loaded into our R working environment, the following block of code shows how we can perform convolution with an impulse response function:

```r
# t.start is the sequence of onset times.
# duration is the stimulus duration (in seconds).
# For convolving the HRF with impulse response functions, set duration to 0.
# measurement is the number of measurements.
# TR is the repetition time of the experiments.
# resolution is temporal resolution used for upsampling in seconds
# Syntax
X = sapply(t.start, hrf.conv, duration, measurement, TR, resolution)
# Example: Convolving with impulse responses
X = sapply(t.start = c(4, 7, 12, 20), hrf.conv, duration = 0, measurement = 30,
   TR = 2, resolution = 0.01)
```

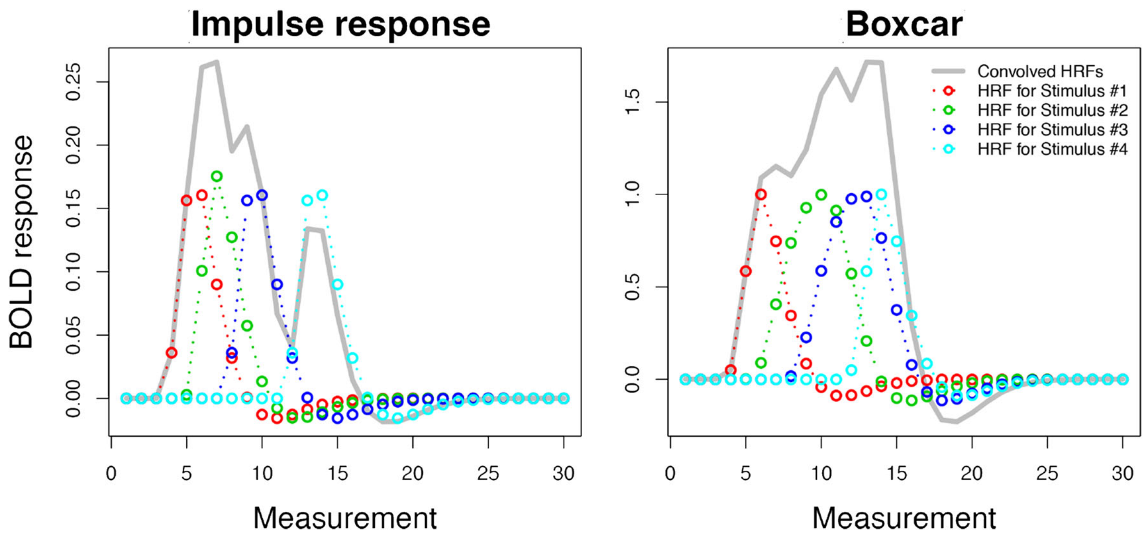

Fig. 2 Examples of convolution. Each panel shows an example of convolution using either an impulse function (left) or a boxcar function (right). The gray solid line shows the convolved hemodynamic response functions (HRFs) for the four stimuli. Red, green, blue, and cyan dotted lines correspond to the individual HRFs corresponding to each of the four stimuli, respectively

The function `hrf.conv` takes the HRF with a unit-level amplitude (i.e., $h_0(x)$; Eq. 1, see Appendix A) and returns convolved HRFs using design variables such as stimulus onset times (`t.start`), stimulus duration (`duration`), the number of measurements (`measurement`), and TR (`TR`). The last argument `resolution` is only necessary for boxcar convolution as the function `hrf.conv` was intended to subsume impulse convolution as a special case of boxcar convolution. Accordingly, to perform convolution with an impulse function, simply set `duration` to zero. We also decided to exclude the baseline activation column from $\mathbf{X}$ and add the baseline activation separately to simplify computation.

To generate a complete design matrix `X`, we use the R function `sapply` to make `hrf.conv` return multiple sequences of convolved HRFs according to each stimulus onset in `t.start` as in line 8. Line 10 shows an example of impulse convolution with four stimuli presented at t =

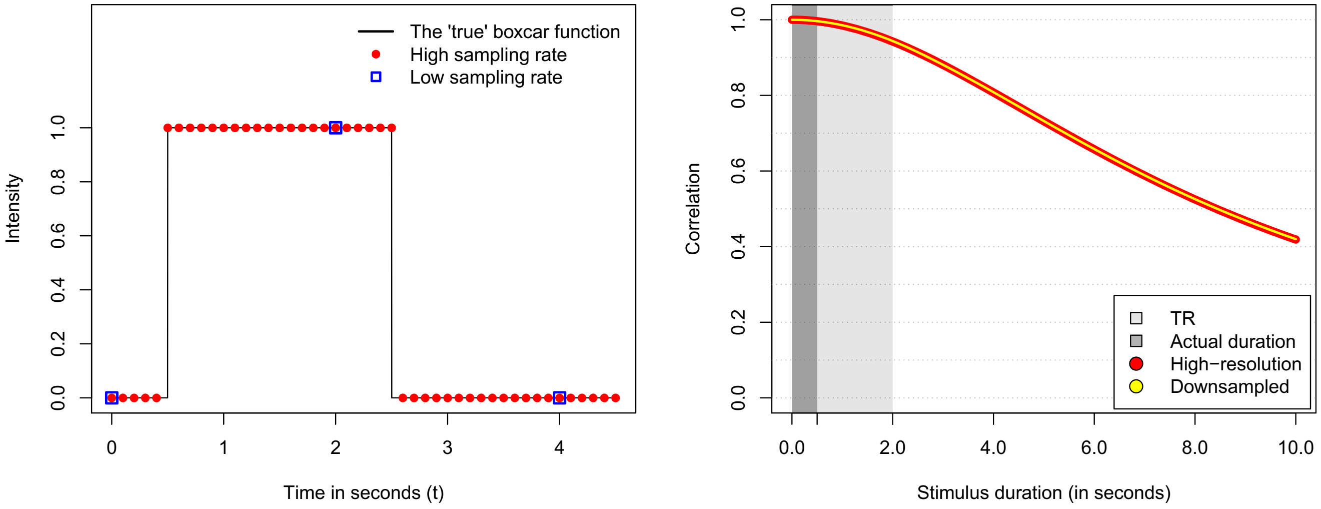

Fig. 3 Effects of resolution on the convolution operation. The left panel shows how the resolution of the temporal grid affects approximation quality. The stimulus intensity is shown over 4 s, with a stimulus presented for 2 s at time $t = 0.5$. The red dots indicate a boxcar function defined at a high temporal resolution, the blue squares indicate a boxcar function defined by a grid of low temporal resolution, and the black line indicates the "true" boxcar function we would expect with the stimulus onset and duration. The right panel shows the correlation (y-axis) between convolved signals assuming an impulse function (yellow) and a boxcar function (red) as a function of the length of stimulus presentation (x-axis)

4, 7, 12, and 20. Here, we assume that we acquired 30 measurements with $TR = 2$. The resulting convolution used in line 10 is shown in the left panel of Fig. 2. For more in-depth details of how `hrf.conv` works, we refer the reader to Appendix C where the full code is provided.

Boxcar Convolution

Although using impulse functions makes the conceptualization of convolution significantly easier, it may not serve some readers well. For example, impulse functions would not be appropriate in experiments that use block designs, or slow event-related designs where stimuli are presented for many seconds. In these cases, we could not simply shift the common HRF template to one specific time. Instead, boxcar convolution would be needed.

Fortunately, we can maintain our discrete representation of time when approximating the integration in Eq. 3. In this case, the resolution of the temporal grid becomes vitally important so that the predicted neural time series can be approximated well. The left panel of Fig. 3 illustrates how the resolution of the temporal grid will affect the quality of our approximation. In this example, we have assumed that the BOLD response is acquired every 2 s, and a stimulus is presented for 2 s at the time $t = 0.5$. If we define a boxcar function with extremely high resolution (red circles), the true stimulus presentation details are well approximated. However, if we use a grid with low temporal resolution, such as the same resolution as our scan acquisition (blue squares), our boxcar function would consist of a single point at $t = 2$. This low resolution would not approximate the true stimulus presentation details, and so the ensuing convolution is unlikely to allow us to investigate the systematic effects the stimuli have on the neural time series.

The left panel of Fig. 3 illustrates that one potential strategy for solving the resolution problem is to implement an "upsampling-convolving-downsampling" (UCD) cycle. First, we could increase the temporal resolution of the boxcar function (i.e., upsampling) within R, so that the stimulus presentation details can be approximated well. Second, we could use the high-resolution boxcar function to convolve it with the HRF template to produce a prediction for the neural time series. Using a discrete, albeit high-resolution, grid would still enable us to approximate the integral in Eq. 3 with the summation in Eq. 4. Third, we could decrease the temporal resolution of the convolved signal (i.e., downsampling) to match the fMRI acquisition. This last step would allow for an easy comparison to the observed neural time series through the GLM procedure described above, preserving our ability to simply pass the convolved design matrix X to any software package we wish to use (e.g., JAGS, Stan).

At this point, one may wonder for which stimulus durations a UCD process would be necessary. To evaluate the effects of resolution on the accuracy of the impulse function, the right panel of Fig. 3 shows the correlation (i.e., y-axis) between the predicted neural time series using impulse convolution (i.e., left panel of Fig. 2) and boxcar convolution (i.e., right panel of Fig. 2) for a single stimulus as a function of stimulus duration (i.e., x-axis), using the canonical HRF. The dark gray-shaded area represents the stimulus duration and the light gray-shaded area represents the acquisition time for our particular experiment below. The right panel shows that the correlation remains high (e.g., greater than 0.9) for stimulus presentations that are shorter than 2 s. This indicates that even for somewhat slow stimulus presentations, impulse convolution is nearly as effective as the UCD cycle with a high-resolution grid (red lines). The yellow lines show that downsampling from the high-resolution grid has no effect on the correlations, which is expected given that downsampling occurs after the convolution step.

As in the example of impulse convolution, the following block of code shows how we can convolve the HRF with a boxcar function using R:

```
# Syntax: When stimulus duration differs across stimuli
X = mapply(t.start, duration, hrf.conv, MoreArgs = list(measurement, TR,
    resolution))
# Example: Different stimulus duration values
X = mapply(t.start = c(4, 7, 12, 20), duration = c(2, 10, 10, 2), hrf.conv,
    MoreArgs = list(measurement = 30, TR = 2, resolution = 0.01))
```

As mentioned above, the function `hrf.conv` is made to subsume impulse convolution. For boxcar convolution, we need to specify a stimulus duration that is greater than zero in seconds. Because the result of boxcar convolution can arbitrarily scale a convolved HRF according to how we set `duration` and `resolution`, the function normalizes the height of the convolved HRF (i.e., the maximum value of the convolved HRF) to 1.

Line 2 provides an example of the syntax of boxcar convolution within `hrf.conv` when stimulus duration is nonzero. In this case, we should define `duration` as a vector representing stimulus duration for each stimulus, and use the R function `mapply` so that `hrf.conv` can call the information in the onset and duration vectors for each stimulus simultaneously. Line 4 presents a concrete example when we show four stimuli at $t = 4, 7, 12$, and 20, and their durations are 2, 10, 10, and 2 s, respectively. The right panel of Fig. 2 shows the resulting convolved neural signal. Here, the HRFs for the second (green) and third (blue) stimuli have extended durations compared to the HRFs for the first (red) and fourth (cyan) stimuli due to longer stimulus durations. As described above, the heights of each individual HRF are scaled to have a maximum value of 1, arbitrarily.

Summary

In this section, we described how to convolve a canonical HRF with the design matrix for one's specific experiment. As convolution can be abstruse, we discussed how to perform this operation in the simple case that stimuli are shown briefly (i.e., an impulse function), and then the more general case of stimuli presented for a longer duration. Although the computer code for implementing impulse convolution is relatively simple (see above), performing convolution for the more general case is more difficult. To facilitate this, Appendix A provides a function for the HRF template, Appendix B provides a function that defines a boxcar function, and Appendix C provides a function for general convolution in R. These functions make Bayesian analysis of fMRI time series data quite convenient as they enable the design matrix **X** to be precomputed in R and simply passed to either JAGS or Stan. Within these

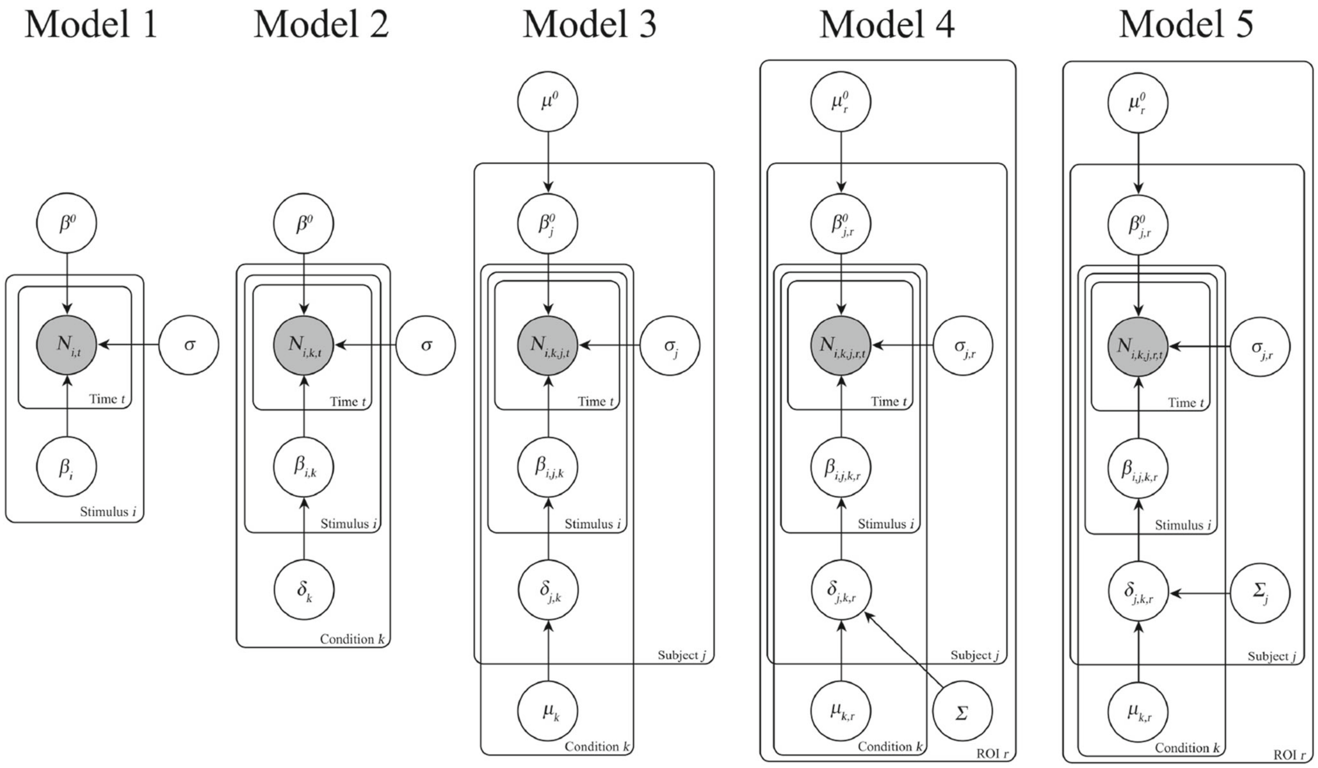

Fig. 4 Graphical Diagrams for Each Model. Each panel illustrates a graphical diagram for each model used in our analysis. Each node represents a variable in the model, where the filled nodes are the observed neural time series from the experiment and empty nodes correspond to latent variables. The design matrix (information about stimulus condition and onset time) was not included in this diagram for visual clarity. Arrows represent relationships between variables and plates represent replications across dimensions (e.g., conditions or subjects). Model 1 has no hierarchical component, model 2 constructs a hierarchy on the condition-level, and model 3 constructs a hierarchy on the subject-level. Models 4 and 5 both construct a hierarchy on the ROI-level: model 4 assumes a common covariance matrix for the entire group of subjects, whereas model 5 assumes one covariance matrix for each subject

packages, one can estimate the influence of an individual stimulus (i.e., the β parameter) by using the dot product of a vector of coefficients with the design matrix.

Model Specification

We developed five models of increasing complexity to investigate which types of constraints should be used to best capture neural data. Figure 4 shows a graphical diagram of the five models. Each graphical diagram illustrates how the model parameters, shown as empty nodes, are connected to the neural data, shown as a gray-filled circle. Beneath the nodes are a set of plates, which represent loops that are needed to capture multiple features of the data. For example, one plate corresponds to the different stimuli that are presented throughout the experiment, as was illustrated in the top panel of Fig. 1. The complexity of the models increases in a progressive manner, where each model adds an additional constraint based on either the experimental condition, subject-to-subject variability, or different brain regions of interest (ROI). As the section above detailed, the design matrix is also part of the model structure as it relates the experimental design to the neural time series, but we have removed it for visual clarity. An additional parameter, σ^{β}, included in models 2–5, is also not pictured as it is nonessential, but its role will be described in detail in later sections. We now discuss each of the five models in turn.

Model 1

The first model is the simplest and has no hierarchical component. $N_{i,t}$ represents the observed neural data at time t for a given region of interest in response to the presentation of a go, no-go, or stop-signal stimulus, referenced as stimulus j. The shape of the BOLD response is constructed from convolved hemodynamic response functions discussed in the previous section.

The observed neural data (i.e., BOLD responses) are assumed to be derived by the sum of baseline activation (β^0) and the convolved hemodynamic responses, with an assumption that the measurement error follows a normal distribution:

$$\begin{aligned}\mathbf{N}(t) &= \beta^0 + \sum_{i=1}^{R} h_i(t) + \epsilon(t)\\ &= \beta^0 + \sum_{i=1}^{R} \beta_i h_{0,i}(t) + \epsilon(t). \qquad (5)\end{aligned}$$

Here β_i and $h_{0,i}$ refer to the neural activation amplitude and nonscaled HRF for the ith stimulus, respectively. R is the number of stimulus presentations, which for one run of our stop-signal task is 240. The notation $\epsilon(t)$ denotes residual noise in the neural activity at each point in time that is not accounted for by the model,

$$\epsilon(t) \sim \mathcal{N}(0, \sigma),$$

where $\mathcal{N}(a, b)$ denotes a normal distribution with mean a and standard deviation b. Given this distributional assumption, we can express the likelihood of the neural data $\mathbf{N}$ using matrix notation as

$$\mathbf{N} \sim \mathcal{N}(\beta^0 + \mathbf{X}\boldsymbol{\beta}, \sigma), \qquad (6)$$

where $\mathbf{X}$ and $\boldsymbol{\beta}$ are the design matrix with stimulus-wise regressors and corresponding activation coefficient vector, respectively.

In this model variant, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{240}]^{\mathsf{T}}$ (stimulus-wise activation), β^0 (baseline activation), and σ (measurement noise) were freely estimated. For the activation parameters, we imposed diffuse normal priors:

$$\begin{aligned}\beta^0 &\sim \mathcal{N}(0, \sqrt{1000})\\ \beta_i &\sim \mathcal{N}(0, \sqrt{1000}) \;\forall\, i = \{1, 2, \ldots, 240\}.\end{aligned}$$

Note that JAGS uses precision defined as an inverse of variance (i.e., $1/\sigma^2$) instead of standard deviation (i.e., σ) for parameterizing the uncertainty of estimates. However, in this paper, we will report the normal priors in terms of standard deviation.

For the measurement noise parameter, we used a diffuse inverse gamma prior

$$\sigma^2 \sim \text{InvGamma}(0.001, 0.001),$$

where InvGamma(r, λ) is the inverse gamma distribution with shape r and rate λ. As stated previously, JAGS uses precision, but for reporting priors in this paper, we use the more conventional notation of standard deviation. We report the noise parameters in terms of variance, as the inverse gamma distribution is conjugate for the variance parameter in this model, which speeds the sampling procedure in JAGS. We note that applying an inverse gamma prior with shape r and rate λ for variance is equivalent to applying a gamma prior with shape r and rate λ for precision.

Model 2

The second model adds a hierarchical structure across conditions. For the stop-signal task, there were four conditions: go, no-go, stop-signal presented before a response was made, and a nuisance regressor. Here, the nuisance regressor refers to trials in which a stop-signal trial was presented *after* a response was made. As the response was already made, we considered the late stop-signal to be cognitively unimportant as it is clearly not related to response inhibition.

Despite this, we included a parameter to model the effects of this late stop-signal as it may have had some unintended influence on the neural signal on subsequent trials.

To build a hierarchy across conditions, we added hyperparameters δ_k where $k = \{1, \ldots, 4\}$ and σ^β to capture the mean and standard deviation of the single-stimulus βs, respectively. Formally, we specified that

$$\beta_{i,k} \sim \mathcal{N}(\delta_k, \sigma^\beta),$$

where i corresponds to the individual stimuli, and k corresponds to the four types of stimuli (i.e., go, no-go, stop-signal, and nuisance stop-signal). For δ_k we imposed a diffuse normal prior

$$\delta_k \sim \mathcal{N}(0, \sqrt{1000}),$$

and specified a diffuse inverse gamma prior on σ^β such that

$$(\sigma^\beta)^2 \sim \text{InvGamma}(0.001, 0.001).$$

We did not assume baseline activation or noise was constrained by condition, so the priors for β^0 and σ are equivalent to their specification in model 1.

Out of the five models, model 2 most closely resembles the standard GLM analysis that would only consider condition-level effects. Compared to the standard GLM, one key difference in our analysis is that we obtain parameter estimates for every stimulus presented in the experiment, instead of averaging this effect out across trials (i.e., pooling trials within the same condition). Because the δ_k parameter captures the central tendency of the single-stimulus βs within a given condition k, the δ_k parameter conveys the same information as a β estimate would in the standard GLM. However, one key difference is that the variance of the posterior of δ_k will likely be larger than the standard GLM β.

Model 3

The third model adds an additional hierarchical structure across subjects. The primary difference between models 2 and 3 is that model 3 assumes that each subject may have a different generative model to describe their brain activation. Formally, this assumption relates to adding subject-specific parameters for the single-stimulus βs and the baseline activation levels β^0.

To build the hierarchy in baseline activation, we added the hyperparameter μ^0 to describe the mean baseline for each of the j terms of β^0. We specified that each subject's baseline activation is sampled from a common normal distribution such that

$$\beta_j^0 \sim \mathcal{N}(\mu^0, \sqrt{1000}),$$

where j indexes the subject number. For the hypermean μ^0, we imposed a diffuse normal prior such that

$$\mu^0 \sim \mathcal{N}(0, \sqrt{1000}).$$

To build the hierarchy in stimulus activations, we added the hypermean parameter μ_j to center each activation parameter $\delta_{j,k}$. Similarly with μ^0, we assume that the condition-level mean activation $\delta_{j,k}$s differ among subjects and are sampled from a normal distribution, which incorporates individual differences in stimulus-wise brain activation in terms of $\beta_{i,j,k}$:

$$\delta_{j,k} \sim \mathcal{N}(\mu_j, \sqrt{1000}), \text{ and}$$
$$\beta_{i,j,k} \sim \mathcal{N}(\delta_{j,k}, \sigma^\beta).$$

Here, we have added the index j to refer to the jth subject. We specified a similarly diffuse prior for μ_j, such that

$$\mu_j \sim \mathcal{N}(0, \sqrt{1000}).$$

For the variability of single-stimulus βs (i.e., σ^β), we do not assume individual differences and therefore the prior for this parameter is defined in the same way as for model 2.

A final addition was the assumption that measurement noise σ should also vary freely across subjects. However, we did not build a hierarchy on the resulting σ_js as it was not clear what distribution they should take. For each σ_j, we specified a similar prior as was declared in model 2:

$$\sigma_j^2 \sim \text{InvGamma}(0.001, 0.001).$$

Model 4

The fourth model investigates patterns of coactivation in the ROIs across the brain. The rationale for including patterns of coactivity was to build in *functional* constraint such that ROIs that had similar profiles of activity through time could constrain one another. In modeling perceptual decision-making in a random dot motion task, Turner et al. (2015) found that models including patterns of single-trial coactivation performed better than models that did not, suggesting that information in different ROIs can be used effectively to constrain estimates of single-trial activation. This result naturally follows from the conditional distribution of a multivariate normal distribution when at least two ROIs have nonzero functional correlation (Turner 2015).

To capture the pattern of coactivation, we used a hyper variance-covariance matrix Σ, a (24 × 24) matrix. For this first model of coactivity, we assumed that coactivation would be similar across all subjects, and so we did not allow it to vary across any dimension of our experiment. The matrix Σ works together with the hypermean vector $\boldsymbol{\mu}$ to control the distribution of the activation parameters δ. Specifically, letting the subscripts j, k, and r represent subject, condition, and ROI, respectively, we assumed

$$\boldsymbol{\delta_{j,k,1:R}} \sim \mathcal{N}_{24}(\boldsymbol{\mu_{k,1:R}}, \Sigma),$$

where $\mathcal{N}_p(a, b)$ denotes a p-dimensional multivariate normal distribution with mean vector a and variance-covariance matrix b. We introduce the notation $\boldsymbol{\mu_{k,1:R}}$ to refer to the kth row of $\boldsymbol{\mu}$ such that

$$\boldsymbol{\mu_{k,1:R}} = \left[\mu_{k,1}, \mu_{k,2}, \cdots, \mu_{k,24}\right]^{\mathsf{T}},$$

and this notation is used similarly on $\boldsymbol{\delta}$.

With this additional hierarchical structure in place, the notation for the parameters β was changed to index each ROI:

$$\beta_{i,j,k,r} \sim \mathcal{N}(\delta_{j,k,r}, \sigma_r^{\beta}).$$

Note that σ_r^{β} no longer varies across subjects, but does vary across ROIs. We specified the following vague prior for each σ_r^{β}:

$$(\sigma_r^{\beta})^2 \sim \text{InvGamma}(0.001, 0.001).$$

The hyperprior for $\boldsymbol{\mu_{k,1:R}}$ is a 24-dimensional multivariate normal distribution

$$\boldsymbol{\mu_{k,1:R}} \sim \mathcal{N}_{24}\left(\boldsymbol{\phi_0}, s_0\right),$$

where $\boldsymbol{\phi_0}$ is a 24-dimensional vector of zeros and s_0 is a (24 × 24) identity matrix (i.e., a diagonal matrix with ones on the diagonal). We specified an inverse Wishart distribution as the prior on Σ such that

$$\Sigma \sim \mathcal{W}^{-1}(I_0, n_0), \tag{7}$$

where I_0 is a (24 × 24) identity matrix representing the scale of the distribution, and $n_0 = 24$ is the degrees of freedom of the inverse Wishart. It should be noted that many different priors for the variance-covariance matrix are possible. We chose the inverse-Wishart distribution because of its mathematical convenience: it is a conjugate prior for the variance-covariance matrix Σ. However, the inverse-Wishart distribution is quite inflexible in that it only contains a precision matrix I_0 and a degrees of freedom parameter n_0. Other, more flexible, priors decompose the Σ matrix and place priors on the resulting components (see Gelman et al. 2004 for more details). Fortunately, software packages, such as JAGS and Stan, make modification of the prior on Σ straightforward, and so we recommend a sensitivity analysis for this particular set of priors if one is interested in their influence on the estimates of Σ.

For the baseline activation parameter β^0, we added an index for the rth ROI, so that

$$\beta_{j,r}^0 \sim \mathcal{N}(\mu_r^0, \sqrt{1000}), \text{ and}$$
$$\mu_r^0 \sim \mathcal{N}(0, \sqrt{1000}).$$

We also added an index to the measurement noise parameter σ to index each subject and ROI, but did not constrain these hierarchically for similar reasons as in model 3:

$$(\sigma_{j,r})^2 \sim \text{InvGamma}(0.001, 0.001).$$

Model 5

Model 5 is identical to model 4, except it allows each subject to have their own variance-covariance matrix Σ. This assumption is based off of the idea that individual differences in connectivity could lead to differences in coactivation. Thus, the prior for δ in model 5 is

$$\boldsymbol{\delta_{j,k,1:R}} \sim \mathcal{N}_{24}(\boldsymbol{\mu_{j,k,1:R}}, \Sigma_j)$$

for the jth subject and kth condition. As in model 4, $\boldsymbol{\mu_{j,k,1:R}}$ and $\boldsymbol{\delta_{j,k,1:R}}$ are 24-dimensional vectors where each element represents a different ROI. The hyperprior for Σ_j is equivalent to the specification in model 4 (i.e., Eq. 7), except for the new index on Σ_j:

$$\Sigma_j \sim \mathcal{W}^{-1}(I_0, n_0).$$

All other priors are defined the same way as in model 4.

Model Summary

Table 1 provides a conceptual summary of the different types of hierarchical constraints used in each of the five models. The first model has no hierarchical component. The second model constructs a hierarchy on the condition-level, constraining single-stimulus βs by the type of stimulus that was presented (i.e., go, no-go, stop-signal, or nuisance stop-signal). The third model constructs a hierarchy on the subject-level, assuming individual differences in measurement noise, baseline neural activity, and the condition-level constraint on single-stimulus βs.

Table 1 Conceptual summary of the different types of hierarchical constraints used in each of the five models

Model	Hierarchy			ROI covariance matrix
	Condition-level	Subject-level	ROI-level	
Model 1				
Model 2	X			
Model 3	X	X		
Model 4	X	X	X	Collapsed across subjects
Model 5	X	X	X	One per individual

The fourth and fifth models construct a hierarchy on the ROI-level, building off of all previous assumptions, but also add the additional feature of ROI coactivation. Model 5 differs from model 4 in that it has a covariance matrix for each subject, whereas model 4 has one group-level covariance matrix. As the five models have somewhat similar JAGS code, Appendix A provides JAGS code for model 3 as an example.

Assessing which hierarchical constraints are appropriate is a tricky problem: we want the models to be complex enough to fit data well, but we also want the models to be able to generalize to new data accurately (Pitt and Myung 2002). In order to explore these considerations, we fit the models to real experimental data. We chose to use the stop-signal task because of its interesting cognitive applications and relevance to missing data problems present in a variety of tasks. Before discussing model fit and generalizability, we review our experimental design and MRI procedures.

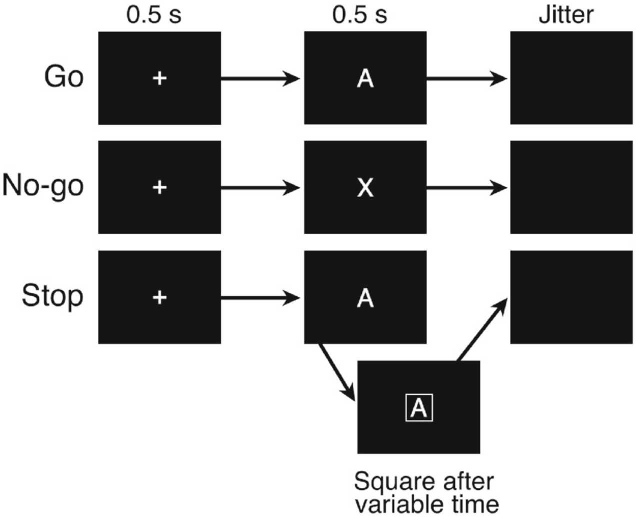

Fig. 5 Example trials. Diagram showing the example stimulus within a trial. Each row corresponds to a trial type from the stop-signal task (one go trial, one no-go trial and one stop trial). For a stop trial, a square around the letter appears after variable time to indicate to inhibit response

The Stop-signal Task

As discussed in the introduction, the stop-signal task is a widely used paradigm in studying response inhibition (Logan and Cowan 1984). Our task design differs from the standard stop-signal task in that it also incorporates aspects of the go/no-go task. The purpose of this incorporation is to be able to distinguish between different types of response inhibition: not going and stopping. In this section, we describe the experimental methods of the task as well as the initial fMRI processing.

Participants

The eleven participants analyzed in this study completed the stop-signal task in the MRI scanner. All participants were recruited from The Ohio State University and its surrounding community and provided informed consent. The study was approved by the Institutional Review Board of the university. Among the eleven participants (mean age = 24.6 years; range from 18 to 48) included in the analysis, there were five females and six males.

Stimuli

All stimuli were programmed in Matlab using Psychtoolbox extensions (http://psychtoolbox.org/) on a Windows PC. The participant laid supine on the scanner bed and viewed the visual stimuli back-projected onto a screen through a mirror attached onto the head coil. Subjects were instructed to press a button when they viewed an A, B, C, D, or E, and to not press any button when they viewed an X, Y, or Z. These trials resemble "go" and "no-go" trials of a go/no-go task, but additionally on some trials a go signal was presented but then after a delay, a stop signal (square around the letter) appeared on the screen. The stimuli remained on screen for 500 m/s. The task consisted of 64 "go" trials, 16 "no-go" trials, and 80 "stop" trials of 3 different delays (individually fit for each subject, based on response time distributions). There were 160 trials per run, and each subject completed three runs of the stop-signal task, so there were 480 trials total. The model fitting focuses on the first run, whereas the validation study uses all three runs. Figure 5 shows example stimuli making up each trial. The jitter in each trial was designed in such a way that the trial duration ranged from 3 to 7 s, with an increment of 1 s. The length of trial duration was optimized by optseq (https://surfer.nmr.mgh.harvard.edu/optseq/). The button response was collected using an MRI compatible fiber optical device (fOPR; https://www.curdes.com/). The TTL output from the fOPRP was fed into the RTBox (Li et al. 2010) to measure response time with high accuracy.

Behavioral Results

Although our analysis does not incorporate a subject's behavior (e.g., response times, accuracy), it is still useful to look at the behavioral results to better understand the data. Figure 6 summarizes some behavioral aspects that are important in the stop-signal task. The left panel of the figure shows a histogram of the response times shown in seconds. The response times are from all conditions and

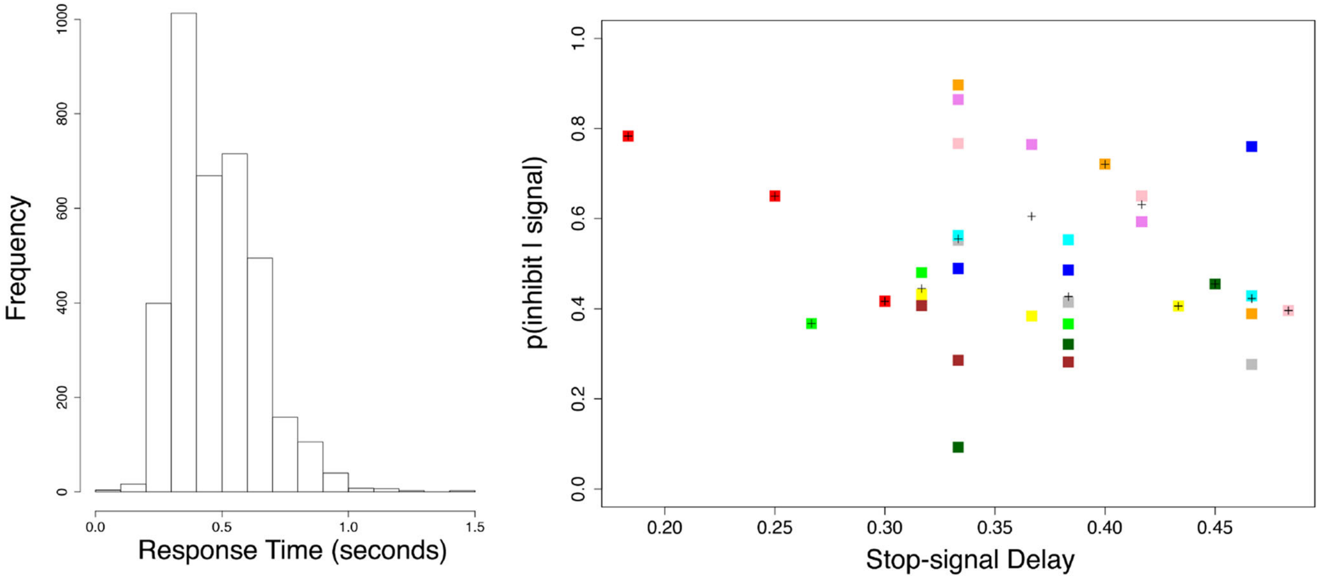

Fig. 6 Behavioral results. Figure showing response time distribution from the task (left) and the relationship between stop-signal delay and the probability a response was successfully inhibited (right). The histogram on the left shows the response time distribution in seconds across subjects and conditions. The scatter plot on the right shows the relationship between stop-signal delay (x-axis) and $p(inhibit|signal)$, or the probability a response was successfully inhibited (y-axis). The squares indicate an individual subjects's $p(inhibit|signal)$ for a given stop-signal delay, where every subject has a distinct color. The black "+" signs refer to the group mean $p(inhibit|signal)$ at that given stop-signal delay

all subjects regardless of errors. The median response time for errors where a button was pressed during a no-go/stop trial is 0.44 s, and the median response time for correct go trials where a button was correctly pressed is 0.46 s. The right panel of Fig. 6 shows the relationship between the length of a stop-signal delay (SSD; in seconds) and the probability that the signal was successfully inhibited, denoted $p(inhibit|signal)$. The black "+" signs refer to the group mean $p(inhibit|signal)$ at that given SSD. The squares indicate an individual subjects' $p(inhibit|signal)$ for a given SSD, where every subject has a distinct color. Each subject had three different SSD conditions and thus three points on the plot. For the majority of subjects, as SSD increases, $p(inhibit|signal)$ decreases; in other words, the longer the delay, the less likely the subject was to successfully inhibit the response. It is important to emphasize that while we show accuracy in these figures, incorrect trials were modeled the same way as correct trials in our analyses—we neither treated errors differently nor removed error trials.

MRI Data Acquisition

MRI recording was performed using a 12-channel head coil in a Siemens 3T Trio Magnetic Resonance Imaging System with TIM, housed in the Center for Cognitive and Behavioral Brain Imaging at The Ohio State University. BOLD functional activations were measured with a T2*-weighted EPI sequence (repetition time = 2000 m/s, echo time = 28 m/s, flip angle = 72°, field of view = 222×222 mm, in-plane resolution = 74×74 pixels or 3×3 mm, and 38 axial slices with 3-mm thickness to cover the entire cerebral cortex and most of the cerebellum). In addition, the anatomical structure of the brain was acquired with the three-dimensional MPRAGE sequence (1×1×1 mm^3 resolution, inversion time = 950 m/s, repetition time = 1950 m/s, echo time = 4.44 m/s, flip angle = 12°, matrix size = 256×224, 176 sagittal slices per slab; scan time 7.5 min) for each participant.

Image Preprocessing and Analysis

The fMRI preprocessing was carried out using FEAT (FMRI Expert Analysis Tool) in FSL (FMRIB software library, version 5.0.8; Smith et al. 2004). The first six volumes were discarded to allow for T1 equilibrium. The remaining images were then realigned to correct for head motion. Data were spatially smoothed using a 6-mm full-width-half maximum Gaussian kernel. The data were filtered in the temporal domain using a nonlinear high-pass filter with a 90-s cutoff. A two-step registration procedure was used whereby EPI images were first registered to the MPRAGE structural image, and then into the standard (MNI) space, using affine transformations. Registration from the MPRAGE structural image to the standard space was further refined using FNIRT nonlinear registration.

After the neural data was preprocessed, the time series from 24 regions of interest (ROIs) were extracted. The selection of ROIs was based on related literature (Dunovan et al. 2015). Table 2 shows information about ROIs and their corresponding indices, used in later figures. The MNI coordinates in the table defined the center of each ROI, and the radius of a sphere ROI was estimated from the number of voxels provided in Dunovan et al. (2015).

As stated previously, the ideal model should be able to fit data well, and be generalizable. To address both concerns, we present our modeling analyses in two stages. First, we fit the five models to one run of data from the stop-signal task, assess the models' ability to fit data from the first run, and examine how the model structure constrains parameter estimates, in particular the neural activation parameter β. Second, we use the fitted models to generate predictions for two additional runs of the same task, using the parameters estimated from the first run, and the experimental design variables from the remaining two runs. Together, the results from these two analyses should help to identify models that not only fit data well, but are also not too complex relative to the data.

Run 1 Model Fitting

The first step of our analysis is to see how well the five models can fit the data. The models increase in complexity, and so we might expect from the outset that the more complex model will provide the best fits to data. However, this is not always the case, depending on how the model is structured, and which aspects of the data inform the parameter estimation procedure. In this section, we fit all five models to the first run of the stop-signal experiment (runs 2 and 3 will be used in the validation analysis in the following section). We compare the neural predictions to the actual observed data, the single-stimulus β estimates, and, for models 2 through 5, the constraint on the hyperparameters. Within this analysis, an "ideal" model would have neural predictions that closely resemble the observed data and provide constrained and reasonable estimates for β.

Methods

Fitting Details

We used JAGS to fit all five models. All of the models had three chains, but took on one of two combinations of adaptation, burn-in, and sampling iterations. The first, longer procedure, was used for just model 1. In this procedure, model initialization ran for 2000 adaptations. After initialization, 4000 samples were discarded as burn-in. Then, the posterior sampling ran for 6000 iterations. Thus,

Table 2 Regions of interest. The table shows the number index, name of each region of interest, MNI coordinates, and the number of voxels. *MNI*: Montreal Neurological Institute

Number	Name	MNI [x y z]	nVox (< 40)
1.	Callosum	[3 −23 29]	208
2.	PCC (posterior cingulate cortex)	[−2 −56 22]	957
3.	preSMA (presupplementary motor area)	[4 21 47]	1952
4.	Left angular gyrus	[−44 −72 30]	328
5.	Left fusiform gyrus	[−43 −60 −17]	84
6.	Left IFG-1 (inferior frontal gyrus 1)	[−37 18 −4]	912
7.	Left IFG-2 (inferior frontal gyrus 2)	[−44 9 29]	426
8.	Left IPL (left inferior parietal lobe)	[−34 −52 46]	459
9.	Left ITG (left inferior temporal gyrus)	[−56 −10 −20]	44
10.	Left insula	[−39 −3 7]	41
11.	Left MFG (left middle frontal gyrus)	[−3 50 −9]	477
12.	Left putamen	[−27 −13 7]	48
13.	Left SFG (left superior frontal gyrus)	[−9 57 35]	128
14.	Left thalamus	[−6 −16 −2]	72
15.	Left ventral striatum	[−1 16 −9]	100
16.	Right caudate	[13 10 6]	55
17.	Right IFG (right inferior frontal gyrus)	[43 20 12]	2830
18.	Right IPL (right inferior parietal lobe)	[48 −44 43]	1400
19.	Right MFG (right middle frontal gyrus)	[38 48 −10]	83
20.	Right MTG (right middle temporal gyrus)	[49 −66 26]	60
21.	Right precuneus	[12 −67 42]	83
22.	Right putamen	[31 −11 4]	44
23.	Right SFG (right superior frontal gyrus)	[21 49 31]	45
24.	Right thalamus	[9 −16 3]	154

with three chains, there was a total of 18,000 samples for each parameter. Because of computational complications, the sampling lengths were shortened for models 2 through 5. In this procedure, model initialization ran for 1000 adaptations, followed by a burn-in period of 2000 iterations. The posterior sampling then ran for 3000 iterations. With the three chains again, there were a total of 9000 samples for each parameter. For all models, the chains were plotted and visually checked for convergence.

Results

To assess the models' ability to fit data, we provide three types of analyses. First, we examine the models' predictions of neural responses across time. Because all models were fitted to the same time series, we can generate predictions from the model and assess how similar they are to the data to which they were applied. Second, we compare estimates of the trial-level activation parameter β across the five models.

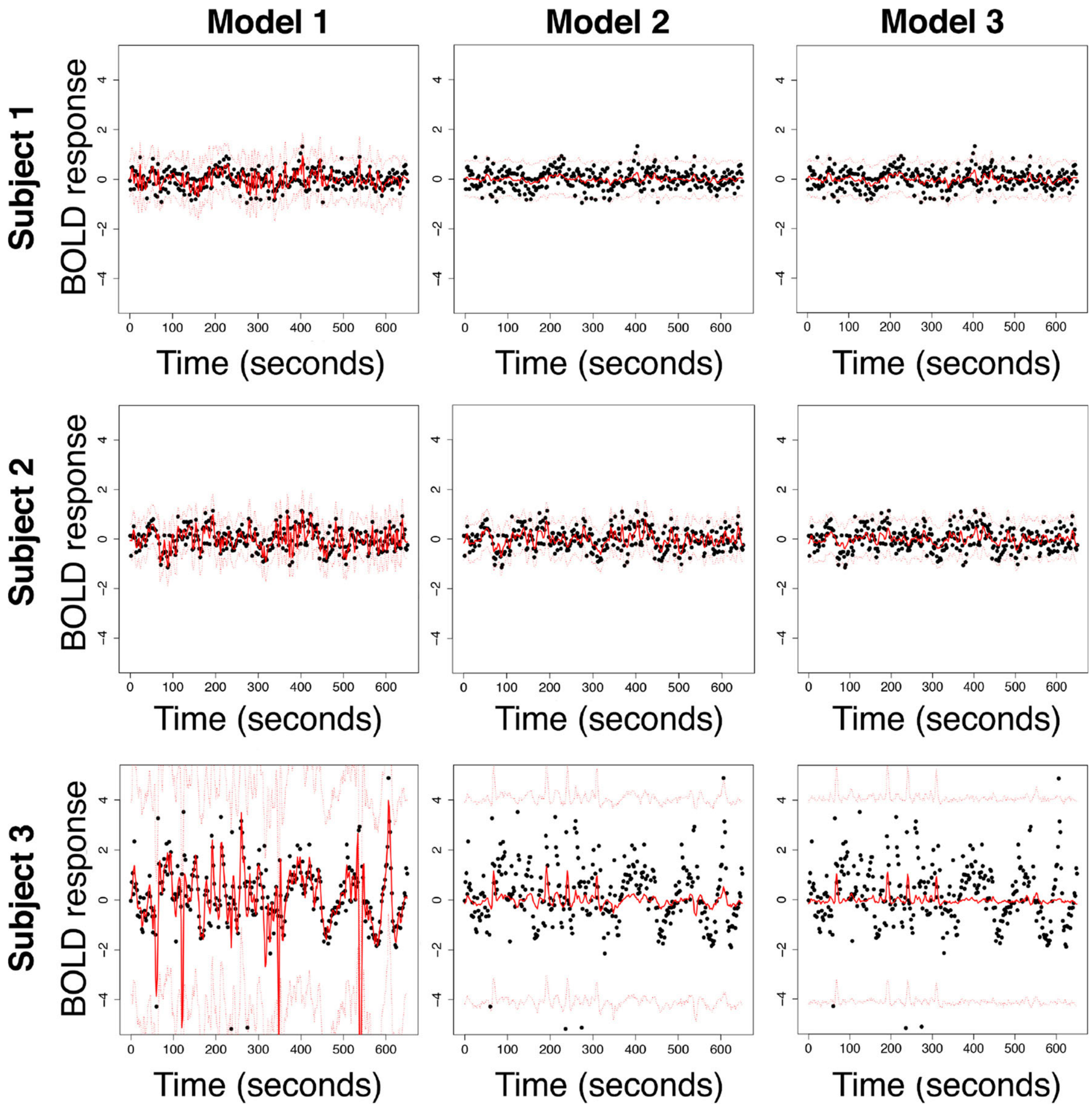

Fig. 7 Time series fits. Model predictions of neural activity of the left ventral striatum. Rows correspond to subjects and columns correspond to model. The black dots in each subplot are the real, observed BOLD response. The solid red line is the mean of the posterior predicted neural data across the time series and the dotted red lines represent the 95% posterior predictive interval

Unlike in the first comparison, β is latent and thus cannot be examined in relation to any true metric. Third, we compare estimates of the condition-level activation parameter δ. Because δ is the hyperparameter of β, it is only present in the models with a hierarchical component (i.e., only models 2–5 can be compared). We conclude by showing how these δ estimates can be used to understand neural activation in the stop-signal task.

Time Series Predictions

All five of the models predicted neural activation at every point of the time series, which allowed us to compare the quality of their fits to data. Figure 7 provides an example of the first three models in predicting the neural activity of the left ventral striatum (ROI 15) across the time series. Each column corresponds to a model and each row corresponds to a subject. The first three subjects were chosen for illustrative purposes, but as we show next, they are representative of the general trends with respect to mean and variability across subjects. The black dots in each subplot represent the observed BOLD responses from the data, whereas the solid red line is the mean predicted neural data from the model, and the dotted red lines represent the 95% posterior predictive interval. The neural predictions from models 4 and 5 did not visually differ from the predictions from model 3 and are not included in this figure.

Visually comparing across panels, Fig. 7 shows that model 1 provides predictions that more closely match the observed data. These results are exemplified for subject 3, whose variability is largest among the three representative subjects. Although Fig. 7 provides a close look at how model predictions compare to the true time series, it shows only a few subjects for only one out of many ROIs. To show that the general conclusions from Fig. 7 apply to the entirety of subjects and ROIs, we correlated each model's predicted time series with the observed time series for each subject and ROI from the first run of the task. Figure 8 shows these correlations. Each panel corresponds to a different model, where the correlation values between predicted and observed time series data are organized by ROI (rows) and subject (columns). The colors indicate the value of the correlation, where cooler (blue-green) colors indicate low correlations and hotter (red-orange) colors indicate higher correlations. Across panels, there is substantial variability in the correlation values, where the lowest correlation across models is 0.11 (in model 2), and the highest correlation is 0.97 (in model 1). Overall, model 1 provided predictions that were the most highly correlated with the observed data. Models 4 and 5 were the next most highly correlated, closely followed by model 3. Model 2 had the lowest correlations out of the five models. Additionally, in all five models, subject 3 showed the most variability and lowest correlation values overall.

As seen in Fig. 7, and more generally in Fig. 8, model 1 makes predictions for the neural time series that clearly fluctuate in ways that nearly match the data, whereas models 2 and 3 make predictions that are relatively stationary across time. This difference across model predictions can be explained by considering the model structure. Model 1

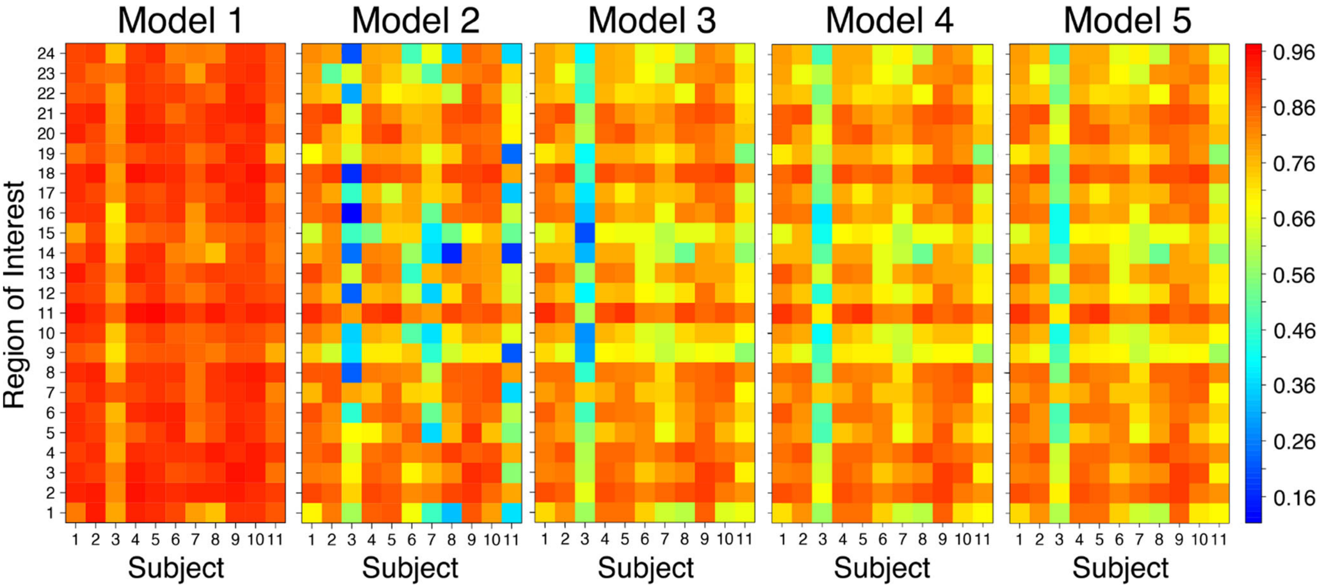

Fig. 8 Time series correlations. Each panel shows the correlations between a given model's predictions for the neural time series and the observed time series data for each ROI (rows) and subject (columns) combination. Colors indicate the correlation between the observed time series data for each subject and ROI combination (i.e., across time). Cooler colors (i.e., blue/green) indicate lower correlations, whereas warmer colors (i.e., orange/red) indicate higher correlations

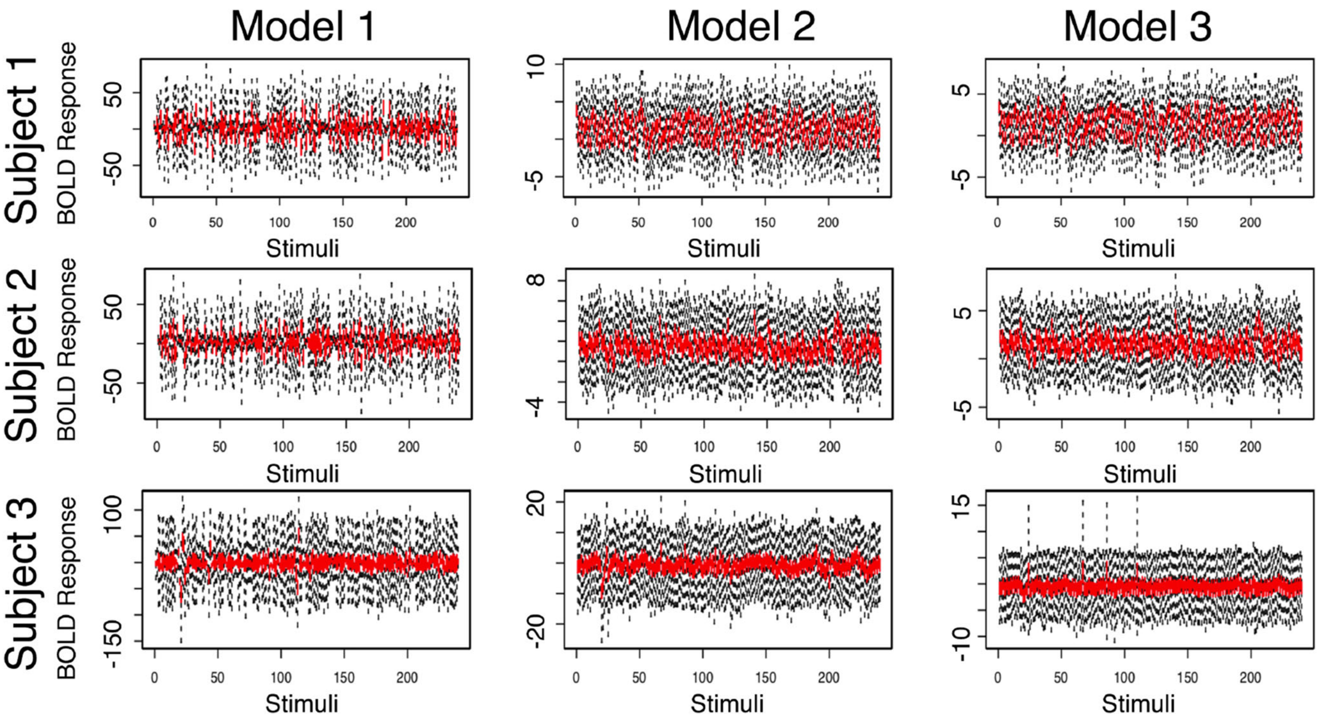

Fig. 9 Constraint on beta estimates. Representative plots of the constraint introduced when constructing a hierarchical component into a model. These single-trial estimates are all for ROI 5, the left fusiform gyrus. Each row corresponds to a different subject and each column corresponds to a model. The red box shows the interquartile range and the dotted lines show the range of the posterior

considers each neural time series as a separate entity, where the estimates for each β parameter are purely influenced by the neural time series shown in Fig. 7. Having only one source of influence allows the estimates to be very sensitive to the particularities of each time series. By contrast, the β parameters for models 2 and 3 are influenced by multiple time series, either across conditions (i.e., model 2), or both condition and subject (i.e., model 3). In this way, models 2 and 3 are more constrained, and this constraint influences the quality of the fits to data. For some aspects of the data, such as predicting the neural time series, this added constraint manifests in negative ways (e.g., as shown by the quality of the fits). However, as the results in this article will make clear, these constraints can manifest in positive ways, too.

Constraint on Beta Estimates

While incorporating the details of the experiment into the model structure did not improve the quality of the predictions for the neural time series, it had a profound effect on the quality of the single-trial beta estimates β. Figure 9 compares estimates of β from the first three models, using the left fusiform gyrus as an example. Similar to Fig. 7, the rows correspond to the first three subjects and the columns correspond to the first three models. Unlike in Fig. 7, however, there is no evaluative metric for comparing the accuracy of the estimated β values, as the parameter is latent, or unobserved. The dotted lines refer to the range of the posterior estimates and the red boxes denote the interquartile range.

Figure 9 shows the effects of shrinkage on the estimates of the parameter β, especially when going from a nonhierarchical model (model 1) to a hierarchical model (model 2). For example, the differences in the variance of the estimated posteriors are so extreme that they distort the y-axis (which changes across columns for visual clarity). However, the improvement in the reduction of parameter uncertainty has its limitations, as the estimates for models 4 and 5 did not drastically improve upon the estimates shown for model 3, so they are not pictured. Importantly, models 4 and 5 provide more information regarding correlations between ROIs. Additionally, models 4 and 5 provide more constraint on the posteriors of other parameters, such as the β hyperparameters (δ), which we now discuss.

Constraint on Delta Estimates

Another parameter of interest is the hyper mean for ROI activation δ. As additional details of the experiment are incorporated into the model, we should expect the estimates for δ to improve because the amount of data it is directly affected by continue to increase. Figure 10 shows the joint distribution of δ_{Go} and δ_{Stop} for the left fusiform gyrus

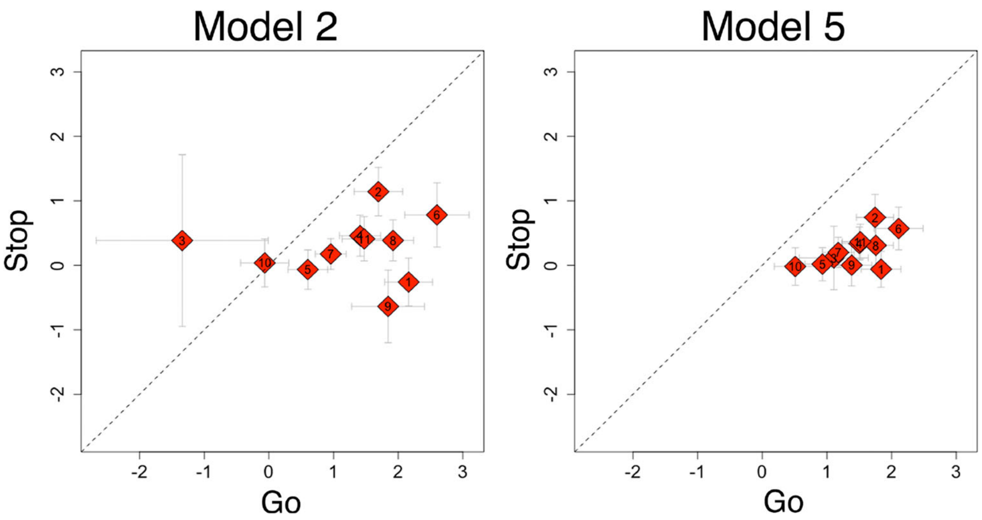

Fig. 10 Constraint on β hyperparameters in SS. The joint distribution of δ_{Go} and δ_{Stop} for the left fusiform gyrus for model 2 (left) and model 5 (right). Each red point corresponds to a different subject. The x-coordinate of the point is the mean of the δ posterior for the go condition and the y-coordinate is the mean of the δ posterior for the stop condition. The error bars represent two standard deviations from either the δ_{Go} mean if oriented horizontally or the δ_{Stop} mean if oriented vertically

in model 2 (left panel) and model 5 (right panel). In this figure, each red point corresponds to a different subject. The x-coordinate of the point is the mean of the posterior estimate for δ in the go condition and the y-coordinate is the mean of the posterior estimate for δ in the stop condition. The error bars represent two standard deviations away from either the δ_{Go} mean if oriented horizontally, or the δ_{Stop} mean if oriented vertically. By plotting the mean activation of δ by trial type, we are better able to visually assess which subjects had activation that was differentially activated across the two conditions. For example, if a point appears in the bottom right triangular area, it suggests that the mean activation in go trials is higher than during stop trials.

In Model 5, all eleven subjects show more activation in the go condition than in the stop condition. In model 2, however, only nine subjects show this pattern of activation. Model 2 also inferred that subject 3 exhibited greater activation for δ_{Stop} than δ_{Go}, with a relatively larger difference (or distance from the line of indifference) than

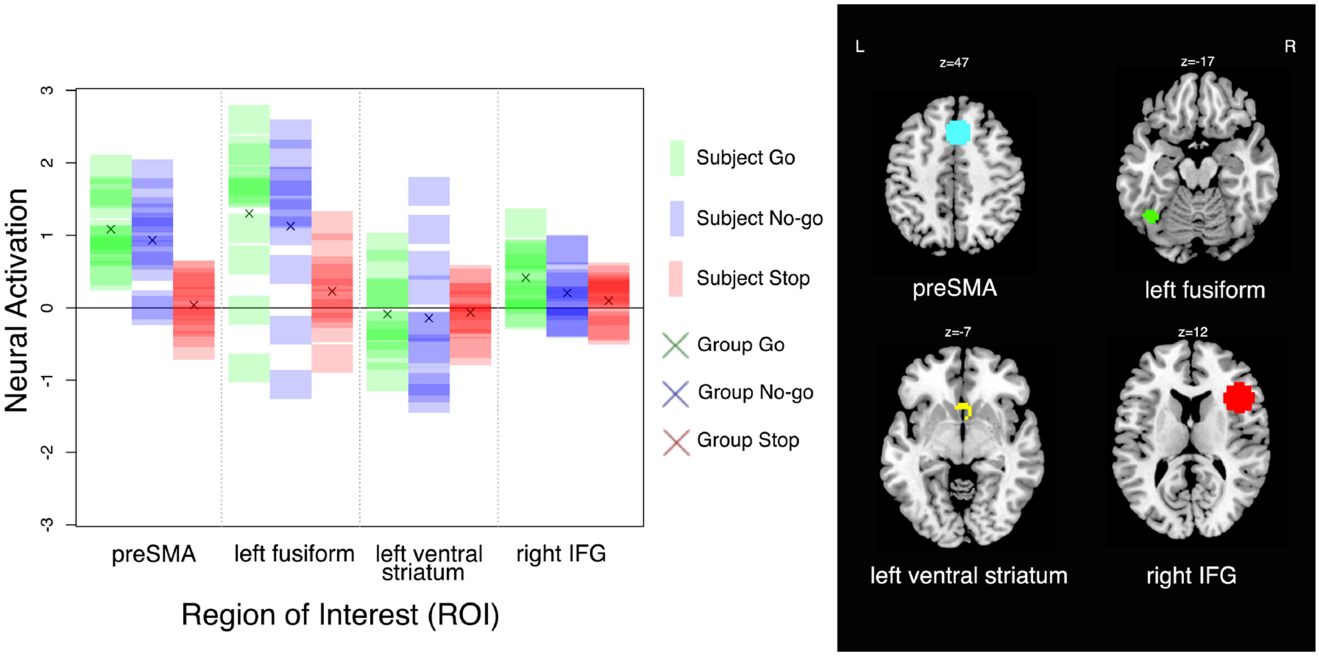

Fig. 11 Delta results for key ROIs. Figure showing the condition-level neural activation (δ) for four regions of interest (left) and the locations of those regions of interest in the brain (right). On the left, each ROI has a color-coordinated column corresponding to condition where green is go, blue is no-go, and red is stop. The alpha-blended rectangles denote the mean of the δ distribution for a single subject, and the crosses denote the group mean. The right shows masks of the four key ROIs: the preSMA (presupplementary motor area; [4, 21, 47]), left fusiform ([− 43, − 60, − 17]), left ventral striatum ([− 1 ,16, − 7]), and right IFC (right inferior frontal cortex; [43, 20, 12])

observed in other subjects. However, in the model 5 subplot, subject 3 shows the opposite pattern, with more activation in δ_{Go} than in δ_{Stop}, and is closer to the group mean. Furthermore, in model 5, the posteriors for the δ parameters for each subject are more constrained than the posteriors from fitting model 2.

Delta and Response Inhibition

The additional constraint of the hierarchy on δ has important implications in understanding the task that is being modeled, because we are often interested in condition-level differences. For example, in the stop-signal task, researchers are interested in understanding the differences between go and stop trials. Because δ is a conditional-level hierarchy built on single-stimulus β estimates, we can use δ estimates to compare activation levels in ROIs and subjects in different conditions. Further constraining δ with other hierarchical levels, such as in models 3, 4, and 5, help us to make more informed conclusions on condition-level differences.

Figure 11 shows how δ (estimated, in this case, using model 3) can be used to understand conditional differences in four key ROIs: presupplementary motor area (preSMA; ROI 3), left fusiform gyrus (ROI 5), left ventral striatum (ROI 15), and right inferior frontal gyrus (rIFG; ROI 17). The left panel of Fig. 11 shows the mean of δ for go stimuli (denoted by green), no-go stimuli (denoted by blue), and stop-signal stimuli (denoted by red). The rectangles show the mean for each subject, and the cross shows the mean for the group. For the preSMA and left fusiform, there is less activation in the stop condition than in either the go or no-go conditions. Additionally, the subject-to-subject variability within an ROI differs. For example, the variability in the left fusiform is larger, whereas the variability is smallest for the right IFG. The right panel of Fig. 11 visualizes the locations of each ROI in the brain. ROI 5 (left fusiform gyrus) and ROI 15 (left ventral striatum) were chosen because they were used earlier in this paper as illustrative examples to show how the subject-level hierarchy constrains neural predictions and single-stimulus β estimates. ROI 3 (presupplementary motor area) and ROI 17 (right inferior frontal cortex) were included because of their relevance to the go/no-go and stop-signal tasks in the cognitive neuroscience literature (Simmonds et al. 2008; Aron et al. 2014).

Validation Analysis on Runs 2 and 3

The model-fit analysis in the last section evaluated how well a particular model fit data from one run of the experiment. However, it is also important that our models generally characterize how the stimuli interact with mental operations. To evaluate which models generalize to new data well, we can use out-of-sample prediction on the other runs of the stop-signal task. The validation analysis could help explain why model 1 closely captured the trend of the neural data, yet did not provide much constraint on the estimates of the single-stimulus activation parameters β. In this section, we use out-of-sample prediction to evaluate each model's ability to generalize to new data.

Methods

As each subject in our task provided data from three different experimental runs, we used the estimated posterior distributions from the first run (i.e., the previous section) to generate predictions from the second and third runs. Hence, the model predictions for the neural time series (i.e., the BOLD response) for both runs are not only based on the parameter estimates from the first run, but the design matrices $\mathbf{X}$ from runs 2 and 3. If a model variant can generate out-of-sample predictions that more closely match the data from runs 2 and 3 in terms of both central tendency and variance, it can be said that the variant generalizes well.

Generating Out-of-sample Predictions

As we used Bayesian statistics to estimate the model parameters from run 1, we need to generalize the information contained within the estimated posterior distributions to runs 2 and 3. To do this, we must construct a posterior predictive distribution (PPD), which quantifies the probability of observing new (i.e., out of sample) data y^* based on the posterior distributions of model parameters θ estimated on data $\mathbf{y}$. Mathematically, the PPD is expressed as

$$f(y^*|\mathbf{y}) = \int f(y^*|\theta)\pi(\theta|\mathbf{y})d\theta. \tag{8}$$

The term $\pi(\theta|\mathbf{y})$ denotes the posterior distribution—the estimates derived from fitting the model to run 1—and the term $f(y^*|\theta)$ denotes a prediction from the model for new data y^*, using the parameters θ and the design matrix corresponding to each run. By integrating over all possible values of θ in the parameter space, we can incorporate the relative probabilities of θ in our predictions of y^*. In other words, values of θ that are more likely (i.e., values of θ with more posterior mass) will place higher predictive mass in the data space corresponding to a prediction from the model at the location of θ.

The key parameter used to generate the neural time series in our model are the stimulus-wise neural activation parameters β_i. To generate predictions about β_i, we must refer to the hyper structure that governs the shape of each β_i across trials. For models 2–5, the hyper structure is defined by the condition-level hyper mean $\delta_{j,k,r}$ and the hyper standard deviation σ^{β}. Using the design matrix for runs 2 and 3, we can generate samples from the distribution of predicted single-stimulus β_is by sampling

$$\beta^*_{i,j,k,r} \sim N(\delta_{j,k,r}, \sigma^{\beta}).$$

Here, the design matrix for runs 2 and 3 provide the index of which $\delta_{j,k,r}$ to use as the hyper mean, namely the jth subject, kth condition, and rth ROI.

Unlike the other models, model 1 does not have a condition-level hierarchy, and so we have no guide in selecting single-stimulus βs when generating out-of-sample predictions. As a remedy, we "pooled" all of the estimated β_is from run 1, and selected one at random when generating each predictive distribution. Because the β_is are pooled and the experimental design is not used when generating predictions for model 1, its predictions should remain fixed regardless of stimulus type, subject, and ROI.

Once single-stimulus β predictions were sampled, they were convolved with a canonical HRF to generate a mean neural time series (i.e., $\mathbf{X}\boldsymbol{\beta}$ in Eq. 5) to match the BOLD response from the experiment. Because the residual noise term σ in Eq. 5 was also estimated when fitting the models to data, we could calculate the probability density of observing the withheld data by evaluating the density of the PPD at the location of each data point at each point in time analytically. For every sampled posterior value, we computed the density of the withheld data, and repeated this process for every posterior sample acquired after burn-in: 18,000 samples per time point in model 1, and 9000 samples per time point in models 2–5.

Results

Figure 12 shows the summary of log-transformed posterior predictive densities (log PPDs) for each subject. Empty and

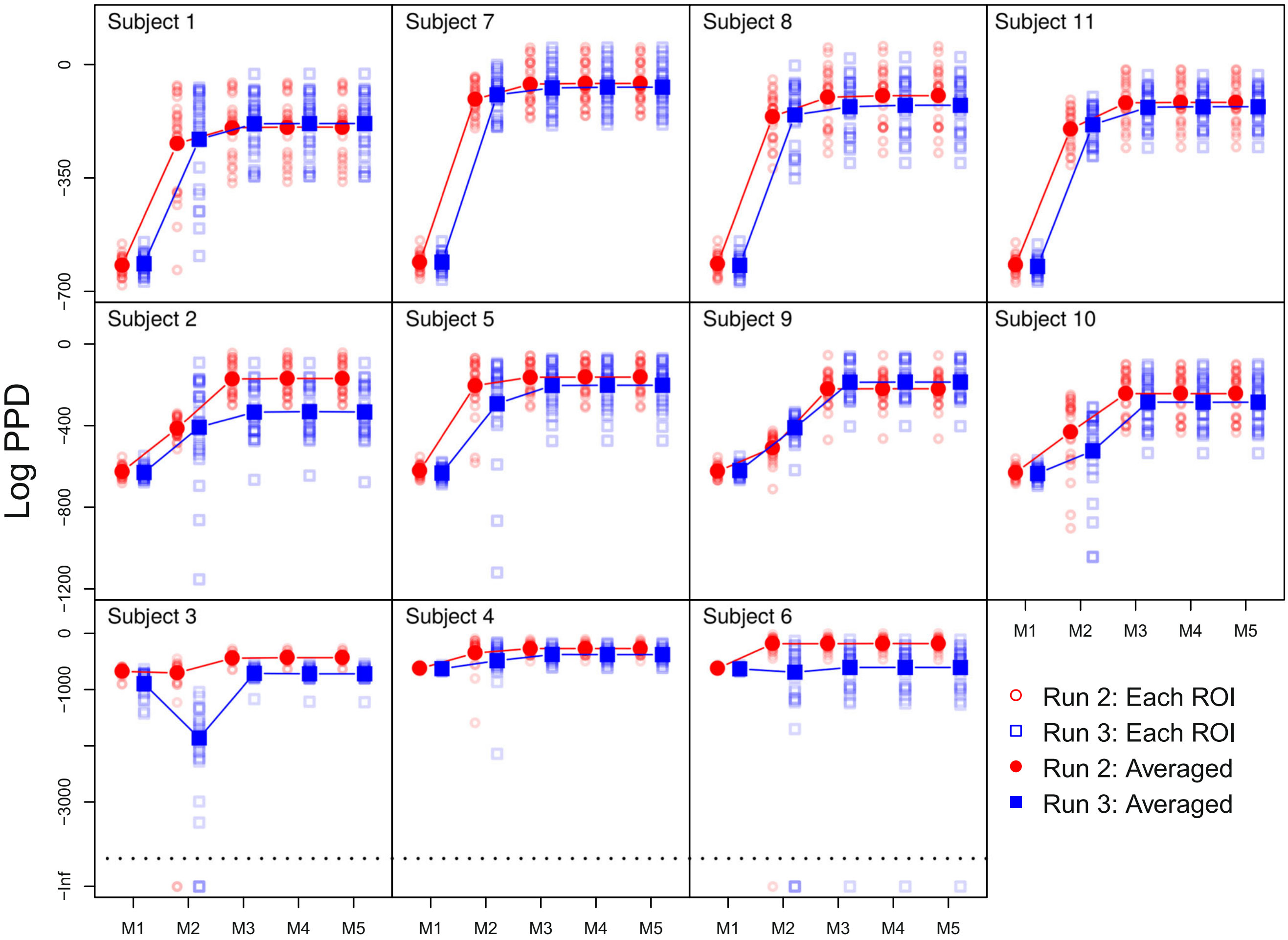

Fig. 12 Validation analyses results. A scatterplot of log-transformed posterior predictive densities (log PPDs) computed for the data of the second (red) and third runs (blue) from each of the 11 participants. Empty squares represent each ROI, while filled squares represent the mean of log posterior predictive density. For each subject and experimental run, ROIs with log PPDs of $-\infty$ due to limited numerical precision are separately plotted below the black dotted line and were excluded when computing the mean of log density

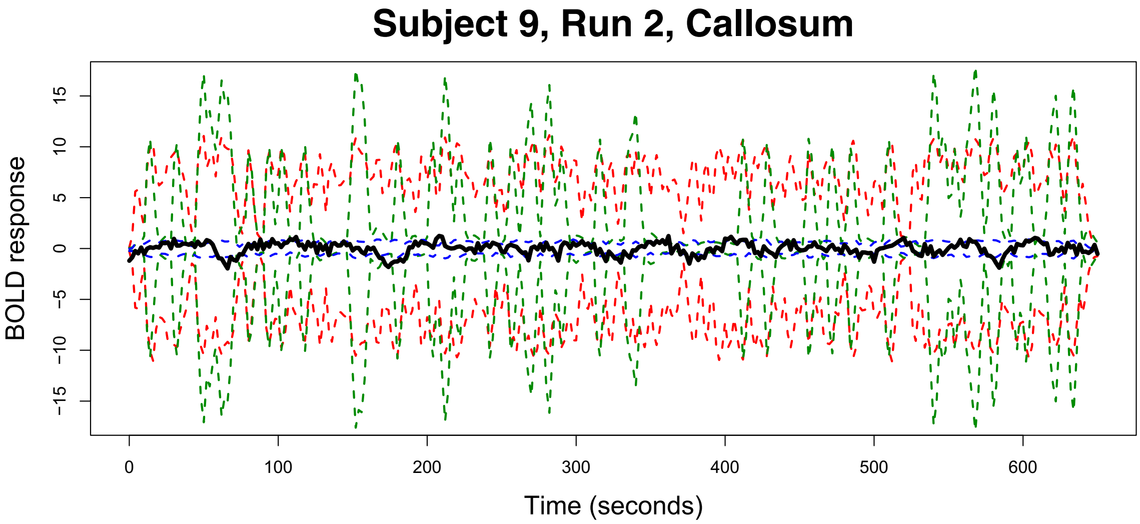

Fig. 13 Comparison of model predictions. 95% predictive intervals are shown for the corpus callosum of subject 9 for three different models: model 1 (red), model 2 (green), and model 5 (blue). A black solid line represents the actual (withheld) BOLD response

filled points represent the log PPD from each ROI and the average log PPD, respectively. Red circles and blue squares refer to the result from the second and third runs, respectively. In a few cases, some models were incapable of predicting any density at the location of the withheld data (e.g., subjects 3 and 5 for model 2). In these cases, data points with "invalid" predictions were ignored and are plotted separately under the black dotted line.

In general, predictions from model 1 show the lowest log PPDs compared to predictions from models 2–5. This is because the prediction of model 1 is too broadly distributed, which results in relatively low log PPD. Figure 13 illustrates an example from the data of the callosum from the ninth participant. The 95% predictive interval from model 1 (red dotted lines) covers a broad range of BOLD responses and practically provides no information about what we can expect based on the model fit from the first run. However, the predictions from model 5 (blue dotted lines) are better constrained and therefore offer informative predictions about the neural signal from the second run.

Across almost every subject, it is difficult to differentiate the prediction performance between models 3, 4, and 5 by the log PPD. Model 2 seems to suffer from a similar problem as model 1, where its predictions are often too under constrained to make consistently accurate predictions for the entire time series (e.g., see the green lines in Fig. 13). Together, these results suggest that models 3–5 have more flexibility when explaining unobserved data by embedding individual- and ROI-level covariance structure.

Unfortunately, it does not appear as though our validation analysis provides any justification for the additional ROI-based constraint used in models 4 and 5. This suggests that while ROIs are clearly correlated within the task[1], the information does not seem to provide additional constraints that are generalizable to new data.

Discussion

We used hierarchical Bayesian modeling to improve the constraint and generalizability of time series models of fMRI data collected in a stop-signal task. First, we found evidence that hierarchical Bayesian modeling improves single-stimulus β estimates by reducing the variability of their posterior distributions. Increasing the levels of the hierarchy improved estimation of these parameters, both at the hyper parameter level and the single-stimulus level. However, we also found some contradictory results from model 1. Of all the models, model 1 best fit the observed neural time series data, but had the least constrained single-stimulus β estimates by far. This is contradictory, because the β estimates inform the neural predictions, so we hypothesized that if the neural predictions are similar to the real data (i.e., they are accurate), then β estimates

[1] We began our analysis by first examining pairwise functional correlations of each time series across all ROIs. These analyses revealed a potential need for ROI-based constraints, motivating the development of models 4 and 5.

would be constrained appropriately. These two aspects of the parameter estimates for model 1 signify the hallmark of overfitting: the parameters are weakly constrained by the data, and so the model is too flexible.

To elucidate this property of the model, we performed a validation analysis. We took the parameter estimates from the first set of data (i.e., the first run) and used them to generate predictions for two new time series of the same task. Here, the models were only guided by the information in the design matrix, i.e., the timing of the stimulus presentation and the type of stimulus presented (e.g., a go or stop signal). The results of the validation analysis clarify the apparent contradiction found in the first set of analyses. Model 1 focused on providing the best fit to the neural data as no formal constraint was imposed to single-stimulus βs. However, in the validation analyses, it allowed an overly broad range of predictions for the new time series data, making its predictions less accurate overall. On the contrary, model 2 did not allow any variability in the single-stimulus β estimates for different individuals or ROIs, solely focusing on experimental conditions. As a consequence, it was also unable to generalize well, although its performance in the validation analysis was consistently better than that of model 1.

This pattern of results can be viewed as symptomatic of overfitting. Model 1 overfits the neural data (time-series predictions) because the single-stimulus βs are freely estimated for every stimulus presentation. This gives the model practically unlimited ability to match the precise shape of the neural time series. Meanwhile, model 2 tends to underfit the neural data because the condition-level hierarchy, collapsing across individual or ROI-level differences, does not allow for flexible predictions. While both of these models fit the time series data well, they were unable to generalize to new data well because they did not learn features of the neural data that were consistent from one scanning run to the next. By contrast, models 3–5 imposed several different constraints that made their particular fit to neural data appear visually worse (compared to models 1 and 2), yet they were able to generalize well because of the neural features they learned in the fitting process. To summarize, by applying the right type of constraints (condition, subject, and ROI), better models were developed that both provided adequate fits to data and generalized well to new data.

Establishing Brain-Behavior Relations

While our analyses provide insight into neural dynamics, they do not connect patterns of brain activation to behavioral responses. To do this, one would have to sort estimates for the neural activation parameter β not only by the type of condition (e.g., go trials), but also by the type of behavior that was observed. For example, with the stop trials, one could manually divide the estimates for β into trials in which the subject correctly inhibited their response as instructed, and into another group where the subject failed to do so. The distribution of βs across these two groups could then reveal how the pattern of ROI activations was different across the two response contingencies, and one could subsequently speculate about the importance of each ROI in successfully implementing executive control.

As mentioned in the introduction, there are several computational models of executive control, designed specifically for the stop-signal task (Logan and Cowan 1984; Logan et al. 2014; Matzke et al. 2013). These models make strong commitments to particular theories about control by instantiating theoretical assumptions within a complete computational model. The advantage of using a computational model is that the patterns of behavioral data—both choice and response time—can be understood through theoretically motivated mechanisms in the model. As many authors have argued, including both choice and response time can provide stringent tests of the suitability of various mechanisms of such models (Ratcliff 1978; Luce 1986; Ratcliff and Rouder 1998; Ratcliff et al. 1999; Van Zandt 2000; Molloy et al. in press). Furthermore, new techniques have been established for linking the mechanisms in these computational models to patterns of neural data, providing even greater constraints on the model (Turner et al. 2013, 2015, 2016, 2017, 2018). An important next step will be to investigate the role that neural data play in constraining computational models of the stop-signal task.

Limitations

Although the analyses presented here are suitable for understanding complex patterns of brain activation, they are not without their limitations. In our view, there are four limitations that merit further consideration: handling autocorrelation of the BOLD response, implementing "boxcar" convolution, using JAGS as a sampler, and performing voxel-based analyses. Below, we provide a discussion of these limitations, as well as some strategies for improving upon the models we have presented here.

Autocorrelation of BOLD Response

One potentially problematic assumption of the five models we have discussed here is the assumption of independent, identically distributed Gaussian noise surrounding the time course of each time series. We made this assumption as it is conventional in both frequentist and Bayesian applications of the general linear model (GLM) to neural time series data (Friston et al. 2002; Penny and Friston 2004). However, BOLD responses measured in fMRI experiments are known to have temporal autocorrelation due to task-irrelevant factors such as thermal noises from the scanner, and physiological noises from participants (Lindquist 2008). Ignoring the temporal autocorrelation has been shown to cause higher false-positive rates (Zarahn et al. 1997; Purdon and Weisskoff 1998) and overestimation of power (Mumford and Nichols 2008).

Fortunately, the autocorrelation of BOLD responses can be accommodated by incorporating autoregressive error models into the typical GLM structure (Purdon et al. 2001; Lindquist 2008; Leonski et al. 2008). For example, a simple autoregressive error model can be constructed such that measurement noise at a given time point t is based on (i.e., correlated with) measurement noise at time $t - p$. These models are denoted AR(p), where p denotes the time dependence in the noise process. More advanced models use a combination of autoregressive structures and moving averages to allow for known physical properties of fMRI scanners, such as the so-called "scanner drift," where the average BOLD response gradually changes throughout the duration of the experiment (Poldrack et al. 2011).

JAGS as a Sampler

One of the major goals of this paper is to present hierarchical Bayesian modeling on time series data in an accessible way. To achieve this, we used JAGS to perform the difficult process of collecting posterior samples. While JAGS is user-friendly and widely used, it is not without limitations. One of the major limitations is that JAGS uses Gibbs sampling which can sometimes produce large autocorrelations. JAGS has built-in functions to plot or obtain a numerical estimate of autocorrelation, as a check on the quality of the samples. In cases where autocorrelation is too large, we recommend using advanced techniques such as differential evolution MCMC (DE-MCMC; ter Braak 2006; Turner et al. 2013), as it is especially powerful in reducing autocorrelation among posterior samples.

Another limitation in the results presented here was the computational burden for the more complicated models. Because of the size of the data and the variables being estimated, we experienced limitations of memory on our system, especially when we saved the mean activation and posterior predictive samples for the raw BOLD responses. Here, custom-built samplers have another advantage over programs like JAGS because the user can define when samples are stored, such as allocating them to memory or writing them out to a file during the sampling procedure. Finally, while JAGS is not directly parallelizable, a custom-build function can dramatically increase the efficiency of collecting posterior samples such as with the R package snow or snowfall.

Voxel-based Versus ROI-based

A final limitation of the current analyses presented here is the focus on ROIs rather than individual voxels. We decided to focus on ROI-based analyses out of practical concerns. While it is not impossible to perform voxel-based analyses, it dramatically increases the computational burden. To implement a voxel-based analysis, one simply needs to replace the time series from a given ROI with the time series data from each individual voxel. Unfortunately, JAGS has memory limitations that make it difficult to store all of the results from extensive estimation procedures. One approach may be to model each voxel independently, although we discourage this approach as it neglects important spatial autocorrelations that exist between nearby voxels. An ideal solution is to apply priors to the parameters governing the time series of each voxel, where the priors reflect information about the distance between voxels (Bowman et al. 2008). To circumvent the memory capacity limitations of JAGS, one could program up the sampler as we suggested above. Although this approach is more difficult to program, it allows the user to print out the samples from each iteration, and so memory would only need to be allocated to store the current position (i.e., within one iteration) of each chain.

Conclusions

This article has illustrated the importance of including appropriate constraints on models of BOLD time series data. We found that applying no constraints allowed models to be too flexible where they fit data well, but performed poorly in validation analyses. On the other hand, imposing constraints on the bases of only condition-level variables masked important features of the data that prohibited the model's ability to capture data well and perform well in validation analyses. The best constraints for our data imposed structure that modulated parameters across both condition and subject variables. While it is certainly true that BOLD measurements from fMRI experiments are often noisy, the results presented here reveal that the data can be purified by applying appropriately constrained hierarchical (Bayesian) models.

Appendix A: R Code for the canonical hemodynamic response function

```r
hrf = function(t.start, length = 30, resolution = 0.01){
  # Shape parameters: assumed to be fixed as used in SPM 12
  a1 = 6; a2 = 16; b1 = 1; b2 = 1; c = 1/6

  # t.start is the stimulus onset time
  # ts is the generated time vector (seconds)
  ts = seq(0, length, resolution) - t.start
  ts[ts<0] = 0

  # ys is the hemodynamic responses
  ys = ( ((ts^(a1-1) * b1^a1 * exp(-b1 * ts) )/gamma(a1) )
         - c * (((ts^(a2 - 1) * b2^a2 * exp(-b2 * ts) )/gamma(a2))) )
  ys
}
```

Appendix B: R Code for the boxcar function

```r
boxcar = function(length, t.start, t.end, resolution = 0.01){
  # the default resolution of the upsampled time series: 10msec
  if (length %% resolution != 0){
    print("length %% resolution must be zero")
  }
  else{
    # xs is the upsampled time vector
    # ts is the boxcar vector
    xs = seq(0, length, resolution)
    ts = rep(0, length(xs))
    ts[(xs >= t.start) & (xs <= t.end)] = 1

    # Return both outputs
    list(xs = xs, ts = ts)
  }
}
```

Appendix C: R Code for boxcar convolution using discrete approximation

```r
hrf.conv = function(t.start, duration, measurement, TR = 2, resolution=0.01){
  # Uses the functions ’hrf’ and ’boxcar’ defined at Appendices A and B.
  # Measurement is the number of TRs
  # length is the total time for data acquisition in seconds
  # t.end is the stimulus offset time
  length = (measurement-1)*TR
  t.end = t.start + duration

  # Define the boxcar functions
  bc = boxcar(length, t.start, t.end, resolution)
  boxcars = bc$ts
  ts = bc$xs
  # Find the time that a stimulus is presented
  stim.on = ts[boxcars == 1]

  # Convolve the HRF with a boxcar function:
  # This part works for the procedure described in Figure 1,
  # but with a boxcar function that specifies the stimulus onset and duration.
  # (1) apply the ’hrf’ function to all time points that the boxcar function
    returns 1.
  hrfs = sapply(stim.on, hrf, length=length)
  # (2) Sum the generated hemodynamic responses.
  conv = rowSums(hrfs)
  # (3) For boxcar convolution, scale the result
  if (duration != 0) {conv = conv / max(conv)}

  # Downsample the result
  temp.idx = which(ts%%TR == 0)
  ts.TR = ts[temp.idx]
  conv.hrfs.TR = conv[temp.idx]

  # Return the result
  res = conv.hrfs.TR
  res
}
```

Appendix D: JAGS Code for Model 3

```
# lenN is length of time series.
# lenS is length of stimuli.
# numS is number of subjects.
# numCond is number of conditions.
# N is a (lenN x numS) of neural data
# cond is a (lenS x numS) of conditions

# Likelihood
for(j in 1:numS){
for (i in 1:lenN) {
N[i,j] ~ dnorm(muN[i,j], sigma[j])
Npred[i,j] ~ dnorm(muN[i,j], sigma[j])
muN[i,j] = beta0[j] + inprod(beta[1:lenS,j], X[i,j])
}}

# Prior
mu0 ~ dnorm(0, 0.001)

for(j in 1:numS){
sigma[j] ~ dgamma(.001, .001)
beta0[j] ~ dnorm(mu0, 0.001)
}

for(j in 1:numS){
for (i in 1:numCond){
delta[i,j] ~ dnorm(mu[i], 0.001)
}}

for(i in 1:numCond){
mu[i] ~ dnorm(0, 0.001)
}

sigmabeta~ dgamma(0.001,0.001)

for(j in 1:numS){
for (k in 1:lenS){
beta[k,j] ~ dnorm(delta[cond[k,j],j], sigmabeta)
}}
```

Compliance with Ethical Standards The study was approved by the Institutional Review Board of the university.

References

Aguirre, G.K., Zarahn, E., D'Esposito, M. (1998). The variability of human, BOLD hemodynamic responses. *NeuroImage*, *8*, 360–369.

Ahn, W.-Y., Krawitz, A., Kim, W., Busemeyer, J.R., Brown, J.W. (2011). A model-based fMRI analysis with hierarchical Bayesian parameter estimation. *Journal of Neuroscience, Psychology, and Economics*, *4*, 95–110.

Aron, A.R. (2007). The neural basis of inhibition in cognitive control. *Neuroscientist*, *13*, 214–228.

Aron, A.R., Robbins, T.W., Poldrack, R.A. (2014). Inhibition and the right inferior frontal cortex: one decade on. *Trends in Cognitive Sciences*, *18*, 177–185.

Bannon, S., Gonsalvez, C.J., Croft, R.J., Boyce, P.M. (2002). Response inhibition deficits in obsessive-compulsive disorder. *Psychiatry Research*, *110*, 165–174.

Boucher, L., Palmeri, T.J., Logan, G.D., Schall, J.D. (2007). Inhibitory control in mind and brain: an interactive race model of countermanding saccades. *Psychological Review*, *114*, 376–397.

Bowman, F.D., Caffo, B., Bassett, S.S., Kilts, C. (2008). A Bayesian hierarchical framework for spatial modeling of fMRI data. *NeuroImage*, *39*, 146–156.

Boynton, G.M., Engel, S.A., Glover, G.H., Heeger, D.J. (1996). Linear systems analysis of functional magnetic resonance imaging in human V1. *Journal of Neuroscience*, *16*(13), 4207–4221.

Buckner, R.L. (1998). Event-related fMRI and the hemodynamic response. *Human Brain Mapping*, *6*, 373–377.

Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Riddell, A. (2017). Stan: a probabilistic programming language. *Journal of Statistical Software*, *76*(1), 1–32.

Chappell, M., Groves, A., Whitcher, B., Woolrich, M. (2009). Variational Bayesian inference for a non-linear forward model. *IEEE Transactions on Signal Processing*, *57*, 223–236.

Chikazoe, J., Jimura, K., Asari, T., Yamashita, i.K., Morimoto, H., Hirose, S., Konishi, S. (2009). Functional dissociation in right inferior frontal cortex during performance of go/no-go task. *Cerebral Cortex*, *19*, 146–152.

Cole, M.W., Yarkoni, T., Repovs, G., Anticevic, A., Braver, T.S. (2012). Global connectivity of prefrontal cortex predicts cognitive control and intelligence. *The Journal of Neuroscience*, *32*, 8988–8999.

Dunovan, K., Lynch, B., Molesworth, T., Verstynen, T. (2015). Competing basal ganglia pathways determine the difference between stopping and deciding not to go. *eLife*, *4*, e08723.

Friston, K., Holmes, A.P., Poline, J., Grasby, P., Williams, S., Frackowiak, R.S., Turner, R. (1995). Analysis of fMRI time-series revisited. *NeuroImage*, *2*(1), 45–53.

Friston, K., Penny, W., Phillips, C., Kiebel, S., Hinton, G., Ashburner, J. (2002). Classical and Bayesian inference in neuroimaging. *NeuroImage*, *16*, 465–483.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B. (2004). *Bayesian data analysis*. New York: Chapman and Hall.

Glover, G. (1999). Deconvolution of impulse response in event-related BOLD fMRI. *NeuroImage*, *9*, 416–429.

Han, H., & Park, J. (2018). Using SPM 12's second-level Bayesian inference procedure for fMRI analysis: practical guidelines for end users. *Frontiers in Neuroinformatics*, *12*, 1.

Jung, R., & Haier, R. (2007). The parieto-frontal integration theory (P-FIT) of intelligence: converging neuroimaging evidence. *Behavioral and Brain Sciences*, *30*, 135–154.

Kruschke, J.K. (2014). *Doing Bayesian data analysis: a tutorial with R, JAGS, and Stan*. Burlington: Academic Press.

Lee, M.D. (2008). Three case studies in the Bayesian analysis of cognitive models. *Psychonomic Bulletin and Review*, *15*, 1–15.

Leonski, B., Baxter, L.C., Karam, L.J., Maisog, J., Debbins, J. (2008). On the performance of autocorrelation estimation algorithms for fMRI analysis. *IEEE Journal of Selected Topics in Signal Processing*, *2*, 828–838.

Li, X., Liang, Z., Kleiner, M., Lu, Z.-L. (2010). RTbox: a device for highly accurate response time measurements. *Behavioral Research Methods*, *42*, 212–225.

Lindquist, M.A. (2008). The statistical analysis of fMRI data. *Statistical Science*, *23*, 439–464.

Logan, G.D. (1985). Executive control of through and action. *Acta Psychologica*, *60*, 193–210.

Logan, G.D., & Cowan, W. (1984). On the ability to inhibit thought and action: a theory of an act of control. *Psychological Review*, *91*, 295–327.

Logan, G.D., Van Zandt, T., Verbruggen, F., Wagenmakers, E.J. (2014). On the ability to inhibit thought and action: general and special theories of an act of control. *Psychological Review*, *121*, 66–95.

Logan, G.D., Yamaguchi, M., Schall, J.D., Palmeri, T.J. (2015). Inhibitory control in mind and brain 2.0: blocked-input models of saccadic countermanding. *Psychological Review*, *122*, 115–147.

Luce, R.D. (1986). *Response times: their role in inferring elementary mental organization*. New York: Oxford University Press.

Matzke, D., Dolan, C.V., Logan, G.D., Brown, S.D., Wagenmakers, E.-J. (2013). Bayesian parametric estimation of stop-signal reaction time distributions. *Journal of Experimental Psychology: General*, *142*, 1047–1073.

Miller, E.K., & Cohen, J.D. (2001). An integrative theory of the prefrontal cortex. *Annual Review of Neuroscience*, *24*, 167–202.

Miyake, A., Friedman, N., Emerson, M., Witzki, A., Howerter, A., Wager, T. (2000). The unity and diversity of executive functions and their contributions to complex "frontal lobe" tasks: a latent variable analysis. *Cognitive Psychology*, *41*, 49–100.

Miyake, A., & Friedman, N.P. (2012). The nature and organization of individual differences in executive functions: four general conclusions. *Current Directions in Psychological Science*, *21*, 8–14.

Molloy, M.F., Galdo, M., Bahg, G., Liu, Q., Turner, B.M. (in press). What's in a response time?: on the importance of response time measures in constraining models of context effects.

Monterosso, J.R., Aron, A.R., Cordova, X., Xu, J., London, E.D. (2005). Deficits in response inhibition associated with chronic methamphetamine abuse. *Drug and Alcohol Dependence*, *79*, 273–277.

Monti, M.M. (2011). Statistical analysis of fMRI time-series: a critical review of the GLM approach. *Frontiers in Human Neuroscience*, *5*, 28.

Mumford, J.A., & Nichols, T.E. (2008). Power calculation for group fMRI studies accounting for arbitrary design and temporal autocorrelation. *NeuroImage*, *39*, 261–268.

Mumford, J.A., Turner, B.O., Ashby, F.G., Poldrack, R.A. (2012). Deconvolving BOLD activation in event-related designs for multivoxel pattern classification analyses. *NeuroImage*, *59*, 2636–2643.

Nigg, J. (2001). Is ADHD a disinhibitory disorder? *Psychological Bulletin*, *127*, 571–598.

Nigg, J., Wong, M., Martel, M., Jester, J., Puttler, L., Glass, J., Zucker, R. (2006). Poor response inhibition as a predictor of problem drinking and illicit drug use in adolescents at risk for alcoholism and other substance use disorders. *Journal of the American Academy of Child and Adolescent Psychiatry*, *45*, 468–475.

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.

Penadés, R., Catalán, R., Rubia, K., Andrés, S., Salamero, M., Gastó, C. (2007). Impaired response inhibition in obsessive compulsive disorder. *European Psychiatry*, *22*, 404-410.

Penny, W., & Friston, K. (2004). Classical and Bayesian inference in fMRI. In Landini, L. (Ed.) *Advanced image processing in magnetic resonance imaging*. New York: Marcel Dekker.

Pitt, M.A., & Myung, I.J. (2002). When a good fit can be bad. *Trends in Cognitive Sciences*, *6*, 421–425.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*.

Poldrack, R.A., Mumford, J.A., Nichols, T. (2011). *Handbook of functional MRI data analysis*. Cambridge: Cambridge University Press.

Poline, J.-B., & Brett, M. (2012). The general linear model and fMRI: does love last forever? *NeuroImage*, *62*(2), 871–880.

Purdon, P.L., Solo, V., Weisskoff, R.M., Brown, E.N. (2001). Locally regularized spatiotemporal modeling and model comparison for functional MRI. *NeuroImage*, *14*, 912–923.

Purdon, P.L., & Weisskoff, R.M. (1998). Effect of temporal autocorrelation due to physiological noise and stimulus paradigm on voxel-level false-positive rates in fMRI. *Human Brain Mapping*, *6*, 239–249.

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

Ratcliff, R., & Rouder, J.N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, *9*, 347–356.

Ratcliff, R., Van Zandt, T., McKoon, G. (1999). Comparing connectionist and diffusion models of reaction time. *Psychological Review*, *106*, 261–300.

Rissman, J., Gazzaley, A., D'Esposito, M. (2004). Measuring functional connectivity during distinct stages of a cognitive task. *NeuroImage*, *23*, 752–763.

Rouder, J.N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin and Review*, *12*, 573–604.

Rubia, K., Russell, T., Overmeyer, S., Brammer, M.J., Bullmore, E.T., Sharma, T., Taylor, E. (2001). Mapping motor inhibition: conjunctive brain activations across different versions of go/no-go and stop tasks. *NeuroImage*, *13*, 250–261.

Schachar, R., & Logan, G.D. (1990). Impulsivity and inhibitory control in normal development and childhood psychopathology. *Developmental Psychology*, *23*, 710–720.

Sebastian, A., Jung, P., Neuhoff, J., Wibral, M., Fox, P., Lieb, K., Mobascher, A. (2016). Dissociable attentional and inhibitory networks of dorsal and ventral areas of the right inferior frontal cortex: a combined task-specific and coordinate-based meta-analytic fMRI study. *Brain Structure and Function*, *221*, 1635–1651.

Sebastian, A., Pohl, M., Klöpper, S., Feige, B., Lange, T., Stahl, C., Tüscher, O. (2013). Disentangling common and specific neural subprocesses of response inhibition. *NeuroImage*, *64*, 601–615.

Shiffrin, R.M., Lee, M.D., Kim, W., Wagenmakers, E.-J. (2008). A survey of model evaluation approaches with a tutorial on hierarchical Bayesian methods. *Cognitive Science*, *32*, 1248–1284.

Simmonds, D.J., Pekar, J.J., Mostofsky, S.H. (2008). Meta-analysis of go/no-go tasks demonstrating that fMRI activation associated with response inhibition is task-dependent. *Neuropsychologia*, *46*, 224–232.

Smith, S.M., Jenkinson, M., Woolrich, M.W., Beckmann, C.F., Behrens, T.E., Johansen-Berg, H., Matthews, P.M. (2004). Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, *23*, S208–S219.

Swick, D., Ashley, V., Turken, U. (2011). Are the neural correlates of stopping and not going identical? Quantitative meta-analysis of two response inhibition tasks. *NeuroImage*, *56*, 1655–1665.

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.

Turner, B.M. (2015). Constraining cognitive abstractions through Bayesian modeling. In Forstmann, B.U., & Wagenmakers, E.-J. (Eds.) *An introduction to model-based cognitive neuroscience* (pp. 199–220). New York: Springer.

Turner, B.M., Forstmann, B.U., Love, B.U., Palmeri, T.J., Van Maanen, L. (2017). Approaches to analysis in model-based cognitive neuroscience. *Journal of Mathematical Psychology*, *76*, 65–79.

Turner, B.M., Forstmann, B.U., Steyvers, M. (2018). Computational approaches to cognition and perception. In Criss, A.H. (Ed.) *Simultaneous modeling of neural and behavioral data*. Switzerland: Springer International Publishing.

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.

Turner, B.M., Rodriguez, C.A., Norcia, T., Steyvers, M., McClure, S.M. (2016). Why more is better: a method for simultaneously modeling EEG, fMRI, and behavior. *NeuroImage*, *128*, 96–115.

Turner, B.M., Sederberg, P.B., Brown, S., Steyvers, M. (2013). A method for efficiently sampling from distributions with correlated dimensions. *Psychological Methods*, *18*, 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*, 312–336.

Van Zandt, T. (2000). How to fit a response time distribution. *Psychonomic Bulletin and Review*, *7*, 424–465.

Verbruggen, F., & Logan, G. (2008). Response inhibition in the stop-signal paradigm. *Trends in Cognitive Sciences*, *12*, 418–424.

Woolrich, M.W. (2012). Bayesian inference in FMRI. *NeuroImage*, *62*, 801–810.

Zarahn, E., Aguirre, G.K., D'Esposito, M. (1997). Empirical analyses of BOLD fMRI statistics. *NeuroImage*, *5*, 179–197.

Zhang, L., Guindani, M., Vannucci, M. (2015). Bayesian models for fMRI data analysis. *Wiley Interdisciplinary Reviews. Computational Statistics*, *7*, 21–41.