# Modeling fMRI data generated by overlapping cognitive processes with unknown onsets using Hidden Process Models

Rebecca A. Hutchinson[a,*] Radu Stefan Niculescu[b,1] Timothy A. Keller[c] Indrayana Rustandi[a,d]
Tom M. Mitchell[a,e]

[a]*Computer Science Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213*
[b]*IKM-CKS Group, Siemens Medical Solutions, 51 Valley Stream Parkway, Malvern, PA 19355*
[c]*Department of Psychology, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213*
[d]*Center for the Neural Basis of Cognition, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213*
[e]*Machine Learning Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213*

**Abstract**

We present a new method for modeling fMRI time series data called Hidden Process Models (HPMs). Like several earlier models for fMRI analysis, Hidden Process Models assume the observed data is generated by a sequence of underlying mental processes that may be triggered by stimuli. HPMs go beyond these earlier models by allowing for processes whose timing may be unknown, and that might not be directly tied to specific stimuli. HPMs provide a principled, probabilistic framework for simultaneously learning the contribution of each process to the observed data, as well as the timing and identities of each instantiated process. They also provide a framework for evaluating and selecting among competing models that assume different numbers and types of underlying mental processes. We describe the HPM framework and its learning and inference algorithms, and present experimental results demonstrating its use on simulated and real fMRI data. Our experiments compare several models of the data using cross-validated data log-likelihood in an fMRI study involving overlapping mental processes whose timings are not fully known.

*Key words:* functional magnetic resonance imaging, statistical methods, machine learning, hemodynamic response, mental chronometry

## 1. Introduction

Hidden Process Models (HPMs) are a new method for the analysis of fMRI time series data based on the assumption that the fMRI data is generated by a set of mental processes. HPMs model each process with parameters that define both the spatial-temporal signature of fMRI activation generated by the process over some window of time (e.g. a hemodynamic response function (HRF)), and a probability distribution on the onset time of the process relative to some external event (e.g., a stimulus presentation or behavioral response). By including a probability distribution over process start times, HPMs allow the study of processes whose onset is uncertain and that may vary from trial to trial. HPMs assume these processes combine linearly when they overlap, and the sum of the HRFs from the active processes defines the mean of a Gaussian probability distribution over observed fMRI data. We present both a learning algorithm to train HPMs from fMRI data, and an inference algorithm that applies an HPM to infer the probable identities and start times of processes to explain additional observed data. The HPM learn-

[*] Corresponding author
*Email address:* rah@cs.cmu.edu (Rebecca A. Hutchinson).
*URL:* http://www.cs.cmu.edu/~rah (Rebecca A. Hutchinson).

[1] Supported by funding from Siemens, NSF grants CCR-0085982, CCR-0122581, Darpa grant HR0011-04-1-0041, and a grant from W.M. Keck Foundation.

ing algorithm resolves uncertainty about process identities and start times in the training data, while simultaneously estimating the parameters of the processes (the spatial-temporal signature and timing distribution). This probabilistic framework also provides an opportunity for principled comparison of competing models of the cognitive processes involved in a particular fMRI experiment.

Hidden Process Models build on a variety of work on fMRI analysis, combining aspects of hemodynamic response function estimation and mental chronometry. A linear systems approach for estimating HRFs is described in Boynton et al. (1996), and Dale & Buckner (1997) describes how to deal with overlapping hemodynamic responses when times and identities of the underlying processes are fully observed. More recently, the problem of asynchronous HRF estimation, in which the stimuli might not align with the image acquisition rate, was addressed in Ciuciu et al. (2003). The major difference between these approaches and HPMs is that all of these methods assume processes generating the HRFs are fully known in advance, as are their onset times, whereas HPMs can estimate HRFs even when there is uncertainty about when the HRF begins.

The field of mental chronometry is concerned with decomposing a cognitive task into its component processing stages. Traditionally, mental chronometry has relied on behavioral data such as measured reaction times, but studies have shown that functional MRI can also be used to address this problem (Menon et al. (1998); Formisano & Goebel (2003)). Henson et al. (2002) and Liao et al. (2002) have also proposed methods for estimating the HRF and its latency that use the temporal derivative of an assumed form for the HRF. While these works are not focused on identifying hidden processes, the estimation techniques they describe could potentially be incorporated into the HPM framework.

The use of classification techniques from the field of machine learning is becoming widespread in the fMRI domain (see Haynes & Rees (2006) for an overview). Classifiers have been used to predict group membership for particular participants (e.g. drug-addict vs. control, in Zhang et al. (2005)), and a variety of mental states (Kamitani & Tong (2005); Mitchell et al. (2004); Cox & Savoy (2003); Haxby et al. (2001)). HPMs can also be used for classification, but in a more general setting. Whereas the above classification methods assume that the mental processes being classified do not overlap in time, and that their timings are fully known during both training and testing, HPMs remove both of these restrictions. Finally, the current paper also extends preliminary work on HPMs reported in Hutchinson et al. (2006).

Dynamic Bayesian Networks (DBNs) (Murphy (2002)) are another class of models widely used in machine learning that can be applied to fMRI analysis. Hidden Markov models (HMMs) (Rabiner (1989)) are a type of DBN. HPMs are actually DBNs as well, although it is cumbersome to express HPMs in the standard DBN notation. An example of a DBN that has been used for the spatial-temporal analysis of fMRI is Faisan et al. (2007), which presents hidden Markov multiple event sequence models (HMMESMs). HMMESMs use data that has been pre-processed into a series of spikes for each voxel, which are candidates for association with hemodynamic events. In contrast, we estimate a spatial-temporal response to each process (similar to a hemodynamic event). Additionally, where we estimate a probability distribution over the lag between stimulus and activation, HMMESMs use an optimized, but fixed activation lag. Finally, HMMESMs have only been used to detect activation associated with stimuli, and not to investigate hidden processes.

One of the most widely used methods for analyzing fMRI data is SPM (Friston (2003)). An advantage of SPM is that it produces maps describing the activation throughout the brain in response to particular stimuli. A disadvantage of SPM is that it is massively univariate, performing independent statistical tests for each voxel. In fact, it has been shown that some mental states cannot be detected with univariate techniques, but require multivariate analysis instead (Kamitani & Tong (2005)). HPMs are a multivariate technique that employ data from all voxels, for example, to estimate the onset times of processes. The learned parameters of the HRF for each process can also generate maps to describe the activity over the brain when that process is active (e.g. the average over time of the response for each voxel). This type of map is different from the ones produced by SPM, but still potentially useful.

The ideas discussed in Poldrack (2006) are also relevant to HPMs, especially in interpreting models that include processes that are not well understood. That paper reviews the idea of ’reverse inference,’ and the caveats associated with it. Reverse inference occurs when activity of a brain region is taken to imply the engagement of a cognitive process shown in other studies to be associated with that brain region. This line of reasoning is more useful when the region of interest is highly selective for the cognitive process in question. This paper speaks directly to the problem of interpreting unsupervised processes from the parameters of their estimated hemodynamice response functions in HPMs. Also relevant is the companion paper to Poldrack (2006), Henson

(2006), which discusses conditions under which we can distinguish between competing cognitive theories that differ in the presence/absence of a cognitive process.

The major contributions of HPMs are the ability to study cognitive processes whose onset times are unknown, the ability to compare different theories of cognitive behavior in a principled way, and the ability to do classification in the presence of overlapping processes. In all of these tasks, HPM parameters can also be used to enhance our understanding of the model.

This paper proceeds as follows. Section 2 introduces HPMs formally, including the algorithms for inference and learning. Section 3 provides results on real and synthetic datasets, and Section 4 discusses these results.

## 2. Materials and methods

We first present real and synthetic datasets to which we have applied HPMs so that we may use them as examples in the following section. We then describe the HPM formalism and algorithms.

### 2.1. *Data acquisition and pre-processing*

#### 2.1.1. *Experiment 1: sentence-picture verification*

We applied HPMs to an fMRI dataset collected while participants viewed and compared pictures and sentences. In this dataset, 13 normal participants were presented with a sequence of 40 trials (Keller et al. (2001)). In half of the trials participants were shown a picture (involving vertical arrangements of the symbols *, +, and $) for 4 seconds followed by a blank screen for 4 seconds, followed by a sentence (e.g. “The star is above the plus.”) for 4 seconds. Participants were given 4 seconds to press a button indicating whether the sentence correctly described the picture. The participants then rested for 15 seconds before the next trial began. In the other half of the trials the sentence was presented first and the picture second, using the same timing.

Imaging was carried out on a 3.0 Tesla G.E. Signa scanner. A T2*-weighted, single-shot spiral pulse sequence was used with TR = 500 ms, TE= 18 ms, 50-degree flip angle. This sequence allowed us to acquire 8 oblique axial slices every 500 ms, with an in-plane resolution of 3.125 millimeters and slice thickness of 3.2 mm, resulting in approximately 5000 voxels per participant. The 8 slices were chosen to cover areas of the brain believed to be relevant to the task at hand. More specifically, the eight oblique-axial slices collected in each TR were acquired in two separate non-contiguous four-slice volumes, one positioned superiorly to cover superior parietal and prefrontal areas, and one positioned inferiorly to allow coverage of inferior frontal, posterior temporal, and occipital areas. The spacing between these volumes varied across participants and depended upon individual anatomy. Each participant was also annotated with anatomical regions of interest (ROIs). The data were preprocessed to remove artifacts due to head motion and signal drift using the FIASCO program by Eddy et al. (1998).

#### 2.1.2. *Experiment 2: synthetic data*

The synthetic data was created to roughly imitate the experiment described above, and was used to evaluate HPMs against ground truth. We created datasets using two synthetic processes (ViewPicture and ReadSentence), and three processes (adding Decide). For each experiment, all of the voxels responded to all of the processes. It is unlikely that all the voxels in the brain would respond to all processes in an experiment, but we wished to test HPMs in the most challenging setting possible: maximal spatial-temporal overap among the HRFs of different processes.

For each dataset, the HRF for each process was generated by convolving a boxcar function indicating the presence of the stimulus with a gamma function (following Boynton et al. (1996)) with parameters $\{a, \tau, n\}$ of the form

$$h(t) = a * \frac{(\frac{t}{\tau})^{n-1} exp(\frac{-t}{\tau})}{\tau(n-1)!}.$$

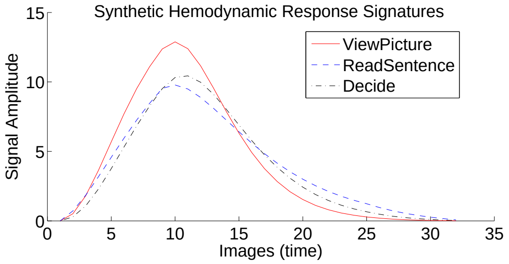

Fig. 1. Noise-free hemodynamic response functions of the processes in the synthetic data.

The parameters for the gamma functions were $\{8.22, 1.08, 3\}$ for ViewPicture, $\{8, 2.1, 2\}$ for ReadSentence, and $\{7.5, 1.3, 3\}$. The processes' gamma functions were convolved with a 4-second boxcar, matching the experiment timeline. These responses are shown in Figure 1.

The ViewPicture and ReadSentence processes had offset values of $\{0, 1\}$, meaning their onsets could be delayed 0 or 1 images (0 or 0.5 seconds) from their corresponding stimuli. The Decide process had offset values of $\{0, 1, 2, 3, 4, 5\}$, meaning its onset could be delayed 0-5 images (0-2.5 seconds) from its corresopnding stimulus, which in each case was the second stimulus presentation, whether it was a picture or a sentence.

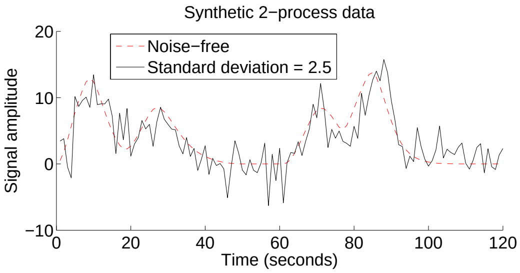

Fig. 2. Two-process synthetic data: a single voxel timecourse for two trials, with and without noise. The first trial is a picture followed by a sentence; the second trial is the reverse.

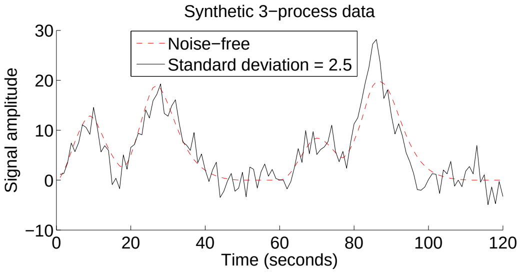

Fig. 3. Three-process synthetic data: a single voxel timecourse for two trials, with and without noise. The first trial is a picture followed by a sentence; the second trial is the reverse. The second peak in each trial is higher than in Figure 2 because the three-process data includes activation for the Decide process.

To generate the data for a new trial in the two-process dataset, we first chose which stimulus would come first (picture or sentence). We required that there be an equal number of picture-first and sentence-first trials. When a picture stimulus occurred, we randomly selected an offset from the ViewPicture offsets and added it to the stimulus onset time to get the process start time. Then we added the ViewPicture HRF (over all voxels) to the synthetic data beginning at that start time. Sentences were dealt with similarly, where overlapping HRFs summed linearly. Finally, we added Gaussian noise with mean 0 and standard deviation 2.5 to every voxel at every time point. The three-process dataset was generated similarly, except that each trial included a Decide process as well, whose onset was randomly chosen and added to the second stimulus onset time to get the process start time. Figure 2 shows the timecourse of a single voxel for two trials from the two-process dataset (with and without noise), and Figure 3 shows the same for the three-process dataset. In both cases, the first trial is a picture followed by a sentence; the second trial is the reverse. The same general process was used to generate four-process data.

### 2.2. *HPM formalism*

A Hidden Process Model is a description of a probability distribution over an fMRI time series, represented in terms of a set of processes, and a specification of their instantiations. HPMs assume the observed time series data is generated by a collection of hidden process instances. Each process instance is active during some time interval, and influences the observed data only during this interval. Process instances inherit properties from general process descriptions. The timing of process instances depends on timing parameters of the general process it instantiates, plus a fixed timing landmark derived from input stimuli. If multiple process instances are simultaneously active at any point in time, then their contributions sum linearly to determine their joint influence on the observed data. Figure 4 depicts an HPM for synthetic sentence-picture data.

HPMs make a few key assumptions, some of which may be relaxed in future iterations of the framework. For instance, the current version of HPMs uses a prespecified and fixed duration for the response signature of a process. Additionally, the process offsets are discrete and tied to specific images. Finally, we adopt the commonly used linearity assumption in fMRI and sum the response signatures of overlapping process instances to predict the signal.

More formally, we consider the problem setting in which we are given observed data $\mathbf{Y}$, a $T \times V$ matrix consisting of $V$ time series, each of length $T$. For example, these may be the time series of fMRI activation at $V$ different voxels in the brain. The observed data $\mathbf{Y}$ is assumed to be generated nondeterministically by some system. We use an HPM to model this system. Let us begin by defining processes:

**Definition 1.** *A process $\pi$ is a tuple $\langle \mathbf{W}, \Theta, \Omega, d \rangle$. $d$ is a scalar called the duration of $\pi$, which specifies the length of the interval during which $\pi$ is active. $\mathbf{W}$ is a $d \times V$ matrix called the response signature of $\pi$, which specifies the influence of $\pi$ on the observed data at each of $d$ time points, in each of the $V$ observed time series. $\Theta$ is a vector of parameters that defines a multinomial distribution over a discrete-valued random variable which governs the timing of $\pi$, and which takes on values from a set of integers $\Omega$. The set of all processes is denoted by $\Pi$.*

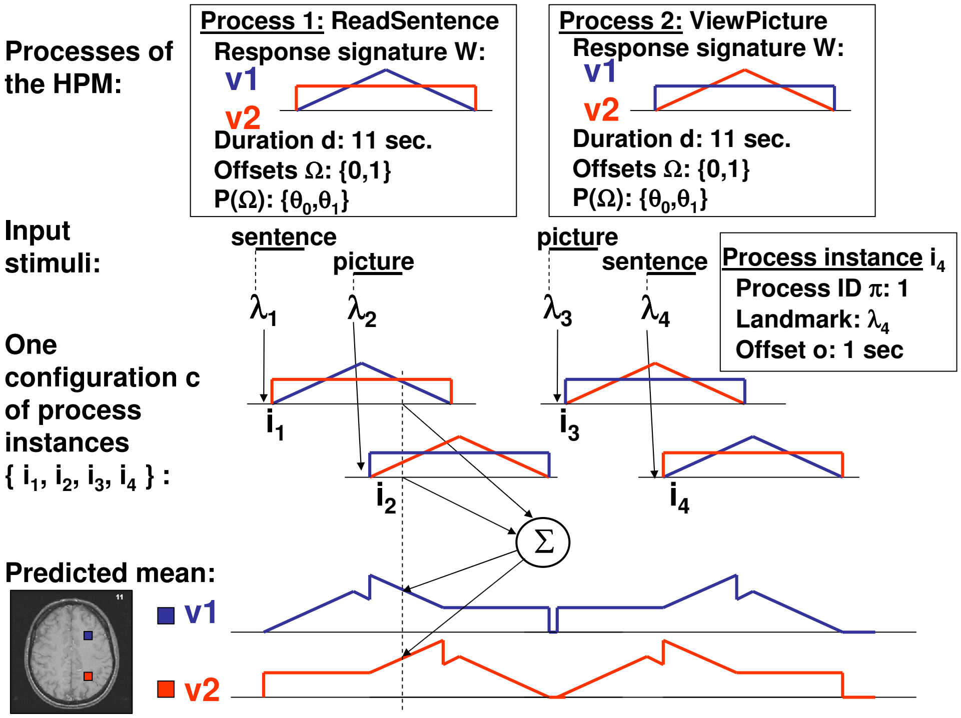

Fig. 4. Example HPM for synthetic 2-process, 2-voxel data. Each process has a duration $d$, possible offsets $\Omega$ and their probabilities $\Theta$, and a response signature $W$ over time (on the horizontal axis) and space (voxels $v1$ and $v2$). They are instantiated 4 times in the configuration $c$. The start time of a process instance $i$ is its landmark $\lambda$ plus its offset $o$. The predicted mean of the data is the sum of the contributions of each process instance at each time point. (The response signatures are contrived rather than realistic in order to more clearly show linearity.)

The set of values in $\Omega$ is defined by the designer of the HPM, and the length of $\Theta$ is the same as the chosen length of $\Omega$. As we will see below, the parameters $\Theta$ can be learned from data.

We will use the notation $\Omega(\pi)$ to refer to the parameter $\Omega$ for a particular process $\pi$. More generally, we adopt the convention that $f(x)$ refers to the parameter $f$ affiliated with entity $x$.

Each process represents a general procedure which may be instantiated multiple times over the time series. For example, in the sentence-picture fMRI study described above, we can hypothesize general cognitive processes such as ReadSentence, ViewPicture, and Decide, each of which is instantiated once for each trial. The instantiation of a process at a particular time is called a *process instance*, defined as follows:

**Definition 2.** *A process instance $i$ is a tuple $\langle \pi, \lambda, O \rangle$, where $\pi$ identifies a* process *as defined above, $\lambda$ is a known scalar called a timing landmark that refers to a particular point in the time series, and $O$ is an integer random variable called the offset, which takes on values in $\Omega(\pi)$. The time at which process instance $i$ begins is defined to be $\lambda + O$. The multinomial distribution governing $O$ is defined by $\Theta(\pi)$. The duration of $i$ is given by $d(\pi)$, and the response signature is $\mathbf{W}(\pi)$.*

The timing landmark $\lambda$ is a timepoint in the trial, typically associated with an event in the experiment design. For example, the timing landmark for a process instances in the sentence-picture verification task may be the times at which stimuli are presented. $\lambda$ specifies roughly when the process instance will begin, subject to some variability depending on its offset $O$, which in turn depends on its process idendity $\pi$. More specifically, the process starts at $t = \lambda + O$, where $O$ is a random variable whose distribution is a property of the process $\pi$. In contrast, the particular value of $O$ is a property of the instance. That is, while there may be slight variation in the offset times of ReadSentence instances, we assume that in general the amount of time between a sentence stimulus (the landmark) and the beginning of the ReadSentence cognitive process follows the same distribution for each instance of the ReadSentence process.

We consider process instances that may generate the data in ordered sets, rather than as individual entities. This allows us to use knowledge of the experiment design to constrain the model. We refer to each possible set of process instances as a *configuration*.

**Definition 3.** *A configuration $c$ of length $L$ is a set of process instances $\{i_1 \dots i_L\}$, in which all parameters for all instances ($\{\lambda, \pi, O\}$) are fully-specified.*

In HPMs, the latent random variable of interest is an indicator variable defining which of the set of configurations is correct. Given a configuration $c = \{i_1 \dots i_L\}$ the probability distribution over each observed data point $y_{tv}$ in the observed data $\mathbf{Y}$ is defined by the Normal distribution:

$$y_{tv} \sim \mathcal{N}(\mu_{tv}(c), \sigma_v^2) \tag{1}$$

where $\sigma_v^2$ is the variance characterizing the time-independent, voxel-dependent noise distribution associated with the $v^{th}$ time series, and where

$$\mu_{tv}(c) = \sum_{i \in c} \sum_{\tau=0}^{d(\pi(i))} \delta(\lambda(i) + O(i) = t - \tau)\ w_{\tau v}(\pi(i)) \tag{2}$$

Here $\delta(\cdot)$ is an indicator function whose value is 1 if its argument is true, and 0 otherwise. $w_{\tau v}(\pi(i))$ is the element of the response signature $\mathbf{W}$ associated with process $\pi(i)$, for data series $v$, and for the $\tau^{th}$ time step in the interval during which $i$ is instantiated.

Equation (2) says that the mean of the Normal distribution governing observed data point $y_{tv}$ is the sum of single contributions from each process instance whose interval of activation includes time $t$. In particular, the $\delta(\cdot)$ expression is non-zero only when the start time $(\lambda(i) + O(i))$ of process instance $i$ is exactly $\tau$ time steps before $t$, in which case we add the element of the response signature $\mathbf{W}(\pi(i))$ at the appropriate delay ($\tau$) to the mean at time $t$. This expression captures a linear system assumption that if multiple processes are simultaneously active, their contributions to the data sum linearly. To some extent, this assumption holds for fMRI data (Boynton et al. (1996)) and is widely used in fMRI data analysis.

We can now define Hidden Process Models:

**Definition 4.** *A Hidden Process Model,* HPM, *is a tuple* $\langle \Pi, \mathcal{C}, \langle \sigma_1^2 \dots \sigma_V^2 \rangle \rangle$, *where* $\Pi$ *is a set of processes,* $\mathcal{C}$ *is a set of candidate configurations, and* $\sigma_v^2$ *is the variance characterizing the noise in the* $v^{th}$ *time series of* $\mathbf{Y}$.

Note that the set of configurations $C$ is defined as part of the HPM and that it is fixed in advance. Each configuration is an assignment of timings and process types to some number of process instances. If we think of the hypothesis space of an HPM as the number of ways that its processes can be instantiated to influence the data, which is a very large number if any process can occur at any time, we can see that $C$ restricts the hypothesis space of the model by allowing the HPM to consider a smaller number of possibilities. These configurations facilitate the incorporation of prior knowledge about the experiment design like timing constraints as mentioned above. For example, if stimulus A was presented at $t = 5$ then there is no need to consider configurations that allow the cognitive process associated with this stimulus to begin at $t = 4$. This knowledge is contained in $C$ if none of the configurations have an instance of process A beginning at a time earlier than $t = 5$. The algorithms presented below can compute the probability of each configuration in several settings.

An *HPM* defines a probability distribution over the observed data $\mathbf{Y}$ as follows:

$$P(\mathbf{Y}|\textit{HPM}) = \sum_{c \in \mathcal{C}} P(\mathbf{Y}|\textit{HPM}, C = c) P(C = c|\textit{HPM}) \tag{3}$$

where $\mathcal{C}$ is the set of candidate configurations associated with the *HPM*, and $C$ is a random variable defined over $\mathcal{C}$. Notice the term $P(\mathbf{Y}|\textit{HPM}, C = c)$ is defined by equations (1) and (2) above. The second term is

$$P(C = c|\textit{HPM}) = \frac{P(\pi(c)|\textit{HPM}) \prod_{i \in c} P(O(i)|\pi(i), \textit{HPM})}{\sum_{c' \in \mathcal{C}} P(\pi(c')|\textit{HPM}) \prod_{i' \in c'} P(O(i')|\pi(i'), \textit{HPM})} \tag{4}$$

where $P(\pi(c)|\textit{HPM})$ is a uniform distribution over all possible combinations of process IDs. $P(O(i)|\pi(i), \textit{HPM})$ is the multinomial distribution defined by $\Theta(\pi(i))$.

Thus, the generative model for an *HPM* involves first choosing a configuration $c \in \mathcal{C}$, using the distribution given by equation (4), then generating values for each time series point using the configuration $c$ of process instances and the distribution for $P(\mathbf{Y}|\textit{HPM}, C = c)$ given by equations (1) and (2).

## 2.3. *HPM algorithms*

### 2.3.1. *Inference: classifying configurations and process instance identities and offsets*

The basic inference problem in HPMs is to infer the posterior distribution over the candidate configurations $\mathcal{C}$ of process instances, given the *HPM*, and observed data $\mathbf{Y}$. By Bayes theorem we have

$$P(C = c|\mathbf{Y}, \textit{HPM}) = \frac{P(\mathbf{Y}|C = c, \textit{HPM}) P(C = c|\textit{HPM})}{\sum_{c' \in \mathcal{C}} P(\mathbf{Y}|C = c', \textit{HPM}) P(C = c'|\textit{HPM})} \tag{5}$$

where the terms in this expression can be obtained using equations (1), (2), and (4).

Given the posterior probabilities over the configurations $P(C = c|\mathbf{Y}, \mathit{HPM})$, we can easily compute the marginal probabilities of the identities of the process instances by summing the probabilities of the configurations in which the process instance in question takes on the identity of interest. For instance, we can compute the probability that the second process instance $i_2$ in a particular trial has identity A by:

$$P(\pi(i_2) = A) = \sum_{c \in \mathcal{C}} \delta((\pi(i_2) = A)|c) P(C = c|\mathbf{Y}, \mathit{HPM}) \tag{6}$$

Note that other marginal probabilities can be obtained similarly from the posterior distribution, such as the probabilities of particular offsets for each process instance, or the joint probability of two process instances having a particular pair of identities.

### 2.3.2. *Learning: estimating model parameters*

The learning problem in HPMs is: given an observed data sequence $\mathbf{Y}$ and a set of candidate configurations, we wish to learn maximum likelihood estimates of the HPM parameters. The set $\Psi$ of parameters to be learned include $\Theta(\pi)$ and $\mathbf{W}(\pi)$ for each process $\pi \in \Pi$, and $\sigma_v^2$ for each time series $v$.

### 2.3.3. *Learning from fully observed data*

First consider the case in which the configuration of process instances is fully observed in advance (i.e., all process instances, including their offset times and process IDs, are known, so there is only one configuration in the HPM). For example, in our sentence-picture brain imaging experiment, we might assume there are only two cognitive processes, ReadSentence and ViewPicture, and that a ReadSentence process instance begins at exactly the time when the sentence is presented to the participant, and ViewPicture begins exactly when the picture is presented.

In such fully observable settings the problem of learning $\Theta(\pi)$ reduces to a simple maximum likelihood estimate of multinomial parameters from observed data. The problem of learning the response signatures $\mathbf{W}(\pi)$ is more complex, because the $\mathbf{W}(\pi)$ terms from multiple process instances jointly influence the observed data at each time point (see equation (2)). Solving for $\mathbf{W}(\pi)$ reduces to solving a multiple linear regression problem to find a least squares solution, after which it is easy to find the maximum likelihood solution for the $\sigma_v^2$. Our multiple linear regression approach in this case is based on the GLM approach described in Dale (1999). One complication that arises is that the regression problem can be ill posed if the training data does not exhibit sufficient diversity in the relative onset times of different process instances. For example, if processes A and B always occur simultaneously with the same onset times, then it is impossible to distinguish their relative contributions to the observed data. In cases where the problem involves such singularities, we use the Moore-Penrose pseudoinverse to solve the regression problem.

### 2.3.4. *Learning from partially observed data*

In the more general case, the configuration of process instances may not be fully observed, and we face a problem of learning from incomplete data. Here we consider the general case in which we have a set of candidate configurations in the model and we do not know which one is correct. The configurations can have different numbers of process instances, and differing process instance identities and timings. If we have no knowledge at all, we can list all possible combinations of process instances. If we do have some knowledge, it can be incorporated into the set of configurations. For example, in the sentence-picture brain imaging experiment, if we assume there are three cognitive processes, ReadSentence, ViewPicture, and Decide, then while it is reasonable to assume known offset times for ReadSentence and ViewPicture, we must treat the offset time for Decide as unobserved. Therefore, we can set all of the configurations to have the correct identities and timings for the first two process instances, and the correct identity and unknown timing for the third.

In this case, we use an Expectation-Maximization algorithm (Dempster et al. (1977)) to obtain locally maximum likelihood estimates of the parameters, based on the following $Q$ function.

$$Q(\Psi, \Psi^{\text{old}}) = E_{C|\mathbf{Y},\Psi^{\text{old}}}[P(\mathbf{Y}, C|\Psi)] \tag{7}$$

The EM algorithm, which we will call Algorithm 1, finds parameters $\Psi$ that locally maximize the $Q$ function by iterating the following steps until convergence. Algorithm 1 is shown in Table 1.

The update to $\mathbf{W}$ is the solution to a weighted least squares problem minimizing the objective function

$$\sum_{v=1}^{V} \sum_{t=1}^{T} \sum_{c \in \mathcal{C}} -\frac{P(C = c|\mathbf{Y}, \Psi^{\text{old}})}{2\sigma_v^2} \; (y_{tv} - \mu_{tv}(c))^2 \tag{8}$$

where $\mu_{tv}(c)$ is defined in terms of $W$ as given in equation (2). We can optionally add a regularization term $r$ to this objective function to penalize undesirable properties of the the parameters:

**ALGORITHM 1**

Iterate over the following two steps until the change in $Q$ from one iteration to the next drops below a threshold, or until a maximum number of iterations is reached.

**E step:** Solve for the probability distribution over the configurations of process instances. The solution to this is given by Equation (5).

**M step:** Use the distribution over configurations from the E step to obtain new parameter estimates for $\mathbf{W}$, $\sigma_v^2 \forall v$, and $\theta_{\pi,o} \forall \pi, \forall o$ that maximize the expected log-likelihood of the full (observed and unobserved) data (Equation 7), using Equations 8 (or 9), 11, and 12.

Table 1
Algorithm 1 computes the maximum likelihood estimates of the HPM parameters in the case where the true configuration of process instances is unknown.

$$\sum_{v=1}^{V}\sum_{t=1}^{T}\sum_{c\in\mathcal{C}} -\frac{P(C=c|\mathbf{Y},\Psi^{\text{old}})}{2\sigma_v^2}\;(y_{tv}-\mu_{tv}(c))^2+\gamma r \tag{9}$$

Here $\gamma$ weights the influence of the regularizer $r$ in relation to the original objective function. In our experiments, we penalized deviations from temporal and spatial smoothness by setting $r$ to the squared difference between successive time points within the process response signatures, summed over voxels, plus the squared difference between adjacent voxels, summed over the time points of the process response signatures. That is:

$$r=\sum_{\pi\in\Pi}\sum_{v=1}^{V}\sum_{\tau=1}^{d(\pi)-1}(w_{v,\tau+1}(\pi)-w_{v,\tau}(\pi))^2+ \tag{10}$$
$$\sum_{\pi\in\Pi}\sum_{i=1}^{V-1}\sum_{j=i+1}^{V}\mathbf{A}(i,j)\sum_{\tau=1}^{d(\pi)}(w_{i,\tau}(\pi)-w_{j,\tau}(\pi))^2$$

where $\mathbf{A}$ is a binary $V\times V$ adjacency matrix ($\mathbf{A}(i,j)$ is 1 if and only if voxel $i$ is adjacent to voxel $j$, where we consider a voxel adjacent to the 26 surrounding voxels).

The updates to the remaining parameters are given by:

$$\sigma_v^2 \longleftarrow \frac{1}{T}\sum_{t=1}^{T}E_{C|\mathbf{Y},\Psi^{\text{old}}}[(y_{tv}-\mu_{tv}(C))^2] \tag{11}$$

$$\theta_{\pi,O=o}\longleftarrow \tag{12}$$
$$\frac{\sum_{c\in\mathcal{C},i\in c}\delta(\pi(i)=\pi\wedge O(i)=o)p(c)}{\sum_{c\in\mathcal{C},i\in c,o'\in\Omega(\pi(i))}\delta(\pi(i)=\pi\wedge O(i)=o')p(c)}$$

where $p(c)$ is shorthand for $P(C=c|\mathbf{Y},\Psi^{\text{old}})$.

the model is:

$$l(W,A,\sigma)=\frac{-NTVlog(2\pi)}{2}-NTVlog(\sigma)- \tag{15}$$
$$\frac{1}{2\sigma^2}l'(W,A)$$

where

$$l'(W,A)=\sum_{n,t,v}(y_{tv}^n-\sum_i a_{iv}w_{(t-\lambda^n(i))i})^2 \tag{16}$$

where $y_{tv}^n$ represents the value of in trial $n$ and $\lambda^n(i)$ represents the start of process $i$ in trial $n$.

**ALGORITHM 2**

Let $\bar{Y}$ be the column vector for the values in $\{y_{tv}^n\}$. Start with $(\hat{W},\hat{A})$ an initial random guess, then repeat Steps 1 and 2 until they converge to the minimum of the function $l'(\hat{W},\hat{A})$.

**STEP 1:** Write $l'(\hat{W},\hat{A})=||U\hat{W}-\bar{Y}||^2$ where $U$ is a $NTV$ by $Kd$ matrix depending on the current estimate $\hat{A}$ of the scaling constants. Minimize with respect to $\hat{W}$ using ordinary Least Squares to get a new estimate $\hat{P}=(U^TU)^{-1}U^T\bar{Y}$.

**STEP 2:** Minimize $l'$ with respect to $\hat{A}$ same as in Step 1.

**STEP 3:** Once convergence is reached by repeating the above two steps, let $\hat{\sigma}^2=\frac{l'(\hat{W},\hat{A})}{NVT}$.

Table 2
Algorithm 2 computes the maximum likelihood estimators for parameters in Shared HPMs.

It is easy to see that the function $l'$ does not depend on the variance $\sigma^2$ and it is a sum of squares, where the quantity inside each square is a linear function in both $W$ and $A$. Based on this observation, we describe a method to compute the maximum likelihood estimators for the parameters that are shared across the voxels in our set, which we call Algorithm 2, shown in Table 2.

In general it is difficult to specify a priori which voxels share their process response signature parameters. Algorithm 3, shown in Table 3 introduces Hierarchical Hidden Process Models, which use a nested cross-validation hierarchical approach to both partition the brain into clusters of voxels that will share parameters and simultaneously estimate those parameters. For each fold of training data, the algorithm starts by partitioning the voxels into their anatomical ROIs. It then iteratively evaluates the current partitioning scheme with an incremental modification to the partition created by splitting one existing subset into 16 smaller subsets using equally spaced planes in all three directions. If the average cross-validated log-likelihood of the model using the new partition is higher than that of the old, the new partition is kept. Otherwise the subset that the algorithm

**ALGORITHM 3**

**STEP 1:** Split the examples into $n$ folds $F = F_1, \ldots, F_n$.

**STEP 2:** For all $1 \leq k \leq n$, keep fold $F_k$ aside and learn a model from the remaining folds using Steps 3-5.

**STEP 3:** Start by partitioning all voxels in the brain into their ROIs and mark all subsets as *Not Final*.

**STEP 4:** While there are subsets in the partition that are *Not Final*, take any such subset and try to split it using equally spaced planes in all three directions (in our experiments we split each subset into 16 = 4 x 2 x 2 smaller subsets). If the cross-validation average log-likelihood of the model learned from these new subsets using Algorithm 2 (based on folds $F \setminus F_k$, where $\setminus$ denotes set minus) is lower than the cross-validation average log-likelihood of the initial subset for folds in $F \setminus F_k$, then mark the initial subset as *Final* and discard its subsets. Otherwise remove the initial subset from the partition and replace it with its subsets which then mark as *Not Final*.

**STEP 5:** Given the partition computed by Step 3 and 4, based on the data points in $F \setminus F_k$, learn a Hidden Process Model that is shared for all voxels inside each subset of the partition. Use this model to compute the log score for the examples/trials in $F_k$.

**STEP 6:** In Steps 2-4 we came up with a partition for each fold $F_k$. To come up with one single model, compute a partition using Steps 3 and 4 based on all $n$ folds together, then, based on this partition learn a model as in Step 5 using all examples. The average log score of this last model can be estimated by averaging the numbers obtained in Step 5 over the folds.

Table 3
Algorithm 3 performs nested cross-validation to partition the brain into clusters of voxels that share their process response signature parameters and simultaneously estimate those parameters.

attempted to split is placed into the final partition, and the algorithm iterates over the other subsets. The final partition is used to compute a score for the current fold of the training set, and then the process is repeated for the other folds. Finally, a model unifying all the folds can be computed, whose performance can be estimated from the cross-validation. (See Table 3 for details.)

## 3. Results

In this section, we present results on both synthetic data and fMRI data from the sentence-picture verification experiment. Our results on synthetic data show that we can accurately recover the true process response signatures from training data using Algorithm 1, that we can correctly classify which of a set of configurations truly generated the test data, and that we can reliably choose the number of processes underlying the data. Our results on the real sentence-picture fMRI data demonstrate the use of HPMs for model comparison in the sentence-picture study, and give examples of the results of the trained models, including spatial and temporal views of an estimated response signature, and an estimated timing distribution. Finally, we present results showing that Shared HPMs can improve data log-likelihood, especially when the training set is small, and show clusters of voxels learned by Algorithm 3.

Some of our results are compared in terms of the log-likelihood of the data under the model. To compute the log-likelihood, we use the log of Equation 3 with a slight modification. Equation 3 takes an average of the data likelihood under each configuration, weighted by the probability of that configuration. To compute the likelihood of the configuration, we use Equations 1 and 2. Since different configurations can be active for windows of time with different lengths, we replace inactive timepoints in $\mu_{tv}(c)$ with the mean over all trials of the training data for timepoint $t$ and voxel $v$, replacing zeros in the predicted mean where no process instances were active. This process helps to minimize the advantage of longer configurations over shorter ones in terms of data log-likelihood. These log-likelihood scores can be averaged over multiple test sets when performing cross-validation to get an estimate of the overall performance of the model.

### 3.1. *Synthetic data*

### 3.1.1. *Estimation of the hemodynamic responses*

To test whether HPMs can accurately recover the true process response signatures underlying the data, we compared the learned response signatures to the true responses in the two and three-process synthetic datasets. These datasets each had only 2 voxels (the small number of voxels was chosen to easily show the learned responses, even though the learning problem is easier with more informative voxels). In each case, we trained the HPM on 40 trials (the number of trials we have in the real data) using Algorithm 1. For the synthetic data experiments we did not use regularization.

In both datasets, the identity of the process instances in each trial were provided to the learning algorithm via the configurations, but their timings were not. This corresponds to reasonable assumptions about the real fMRI data. Since we know the sequence of the stimuli, it is reasonable to assume that the process instances match that sequence. However, we do not know the delay between the stimulus presentation and the beginning of the cognitive process(es) associated with it.

Two trials of the two-process data are shown in Figure 5, and the learned responses for each process in each voxel are shown in Figure 6. Note that the learned responses are reasonably smooth, even though this assumption was not provided to the learner. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.2647, and the estimated standard deviations for the voxels are 2.4182 and 2.4686 (compare with the true value, 2.5). The EM training procedure converged in 16 iterations. Figures 7 and 8 are the corresponding plots for the three-process data. In this case, the mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.4427, and the estimated standard deviations for the voxels are 2.4316 and 2.4226. The EM training procedure converged in 24 iterations.

The identifiability of the model from the data is a significant factor in estimating the HRFs of the processes. A system is identifiable if there is a unique set of parameters that optimally reconstruct the data. For instance, if the cognitive processes never overlap, we can easily deconvolve the contributions of each one. However, if two processes are fully-overlapping every time they occur in the data, there are an infinite number of response signature pairs that could explain the data equally well. Note that the two-process data is identifiable, whereas the three-process data is not.

### 3.1.2. *Classification of configurations*

To test whether these learned HPMs could correctly classify the process instance configurations in a new trial, we ran a similar experiment, again using synthetic data. In this case, the training set was again 40 trials, and the test set consisted of 100 trials generated independently from the same model. This time, we used 500 voxels, all of which responded to both stimuli, in both the training and test sets, which is a reasonable number of voxels in which we might be interested.

The uncertainty in the process configurations in the training set again reflected the real data. That is, the process identities were provided in the configurations for the training data, but the timing was unknown. In the test set however, we allowed more uncertainty, assuming that for a new trial, we did not know the sequence of the first two stimuli. Therefore, the third process was known to be Decide, but the first two could be either ReadSentence followed by ViewPicture, or ViewPicture followed by ReadSentence. Based on our knowledge of the experiment design, the first two instances could not have the same parent process (i.e. ReadSentence followed by ReadSentence was not allowed). For all three instances in the test set trials, the timing was unknown. For both the two-process and three-process data, HPMs predicted the correct configuration on the test data with 100% accuracy.

### 3.1.3. *Choosing the number of processes to model*

Another interesting question to approach with synthetic data is whether or not HPMs can be used to determine the number of processes underlying a dataset. One might be concerned that HPMs would be biased in favor of more and more processes by using extra processes to fit noise in the data more closely. We do expect to fit the training data better by using more processes, but we can use independent test data to evaluate whether the extra processes are indeed fitting noise in the training data (which we would not expect to help fit the test data) or whether the extra process contain actual signal that helps fit the test data. In fact, there can be a slight bias, even when using separate test data, in favor of models that allow a larger number of possible configurations. As we show below, the impact of this bias in our synthetic data experiments does not prevent HPMs from recovering the correct model.

To investigate this question, we generated training sets of 40 examples each using 2, 3, and 4 processes in the same fashion as the previous experiments. For each training set, we trained HPMs with 2, 3, and 4 processes each. Additionally, we generated test sets of 100 examples each using 2, 3, and 4 processes. Every training and test set had 100 voxels. For each test set, we used each of the HPMs trained on its corresponding training set to evaluate the log-likelihood of the test data under the model. In each case, the model selected by the algorithm based on this score was the one with the correct number of processes. We performed this experiment with independently generated training and test sets 30 times with consistent results. The average test-set log-likelihoods are shown in Table 4, along with further repetitions of this experiment with decreasing training set sizes. As expected, we see increased variability with smaller training sets, but even for small numbers of examples, the HPM with the highest test data log-likelihood has the same number of processes as were used to generate the data.

When using real data instead of synthetic data, we do not have an independent test set on which to verify

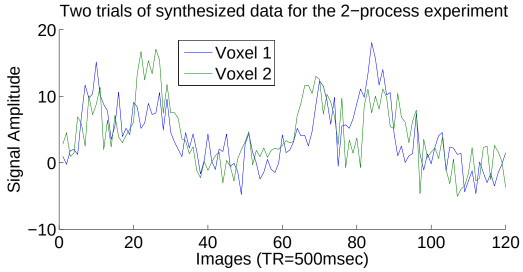

Fig. 5. Two trials of synthetic data for the 2-process experiment using 2 voxels. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial is the reverse.

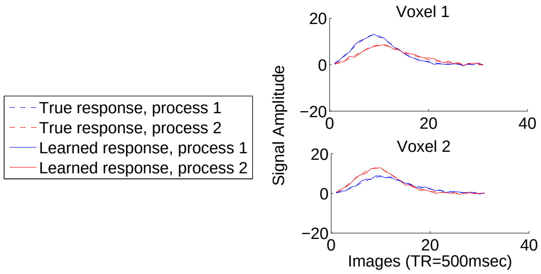

Fig. 6. Comparison of the learned vs. true process response signatures in the two-process data for two voxels. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.2647.

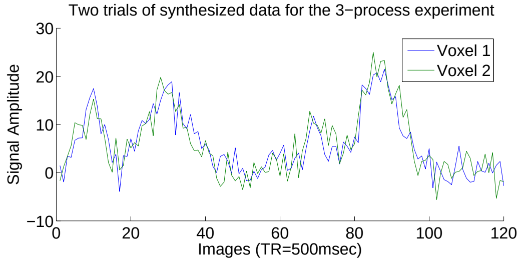

Fig. 7. Two trials of synthetic data for the 3-process experiment using 2 voxels. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial is the reverse.

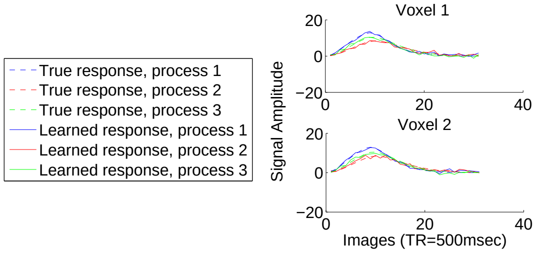

Fig. 8. Comparison of the learned vs. true process response signatures in the synthetic two-process data for two voxels. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.4427.

the number of processes to use. Instead, we can modify our method by using cross-validation. We can separate the dataset into $n$ non-overlapping pieces, leave one of them out as the independent test set, and train HPMs with different numbers of processes on the remaining pieces. For each left-out piece, we can compute the log-likelihood of the data for each HPM, and average these log-likelihoods over the left-out pieces. This average log-likelihood is then a measure of how well each HPM performs on unseen data. This process avoids overfitting

| Number of Training Trials | Number of Processes in HPM | 2 Process Data | 3 Process Data | 4 Process Data |
|---|---|---|---|---|
| 40 | 2 | $\mathbf{-5.64 \pm 0.00444}$ | $-7.93 \pm 0.0779$ | $-7.72 \pm 0.0715$ |
| 40 | 3 | $-7.47 \pm 0.183$ | $\mathbf{-5.66 \pm 0.00391}$ | $-5.72 \pm 0.00504$ |
| 40 | 4 | $-7.19 \pm 0.0776$ | $-5.687 \pm 0.00482$ | $\mathbf{-5.65 \pm 0.00381}$ |
| 20 | 2 | $\mathbf{-2.87 \pm 0.204}$ | $-3.80 \pm 0.192$ | $-3.70 \pm 0.606$ |
| 20 | 3 | $-4.00 \pm 0.0461$ | $\mathbf{-2.86 \pm 0.00597}$ | $-2.87 \pm 0.00276$ |
| 20 | 4 | $-3.91 \pm 0.0319$ | $-2.89 \pm 0.00320$ | $\mathbf{-2.85 \pm 0.00364}$ |
| 10 | 2 | $\mathbf{-1.44 \pm 0.245}$ | $-2.07 \pm 0.0653$ | $-1.96 \pm 0.0665$ |
| 10 | 3 | $-1.99 \pm 0.119$ | $\mathbf{-1.47 \pm 0.0231}$ | $-1.47 \pm 0.00654$ |
| 10 | 4 | $-1.95 \pm 0.0872$ | $-1.49 \pm 0.0195$ | $\mathbf{-1.46 \pm 0.00427}$ |
| 6 | 2 | $\mathbf{-2.87 \pm 0.204}$ | $-1.32 \pm 0.0363$ | $-1.34 \pm 0.297$ |
| 6 | 3 | $-4.00 \pm 0.0461$ | $\mathbf{-0.923 \pm 0.0130}$ | $-0.928 \pm 0.0126$ |
| 6 | 4 | $-3.91 \pm 0.0319$ | $-0.933 \pm 0.0149$ | $\mathbf{-0.921 \pm 0.00976}$ |
| 2 | 2 | $\mathbf{-3.75 \pm 0.00710}$ | $-7.23 \pm 0.0456$ | $-4.62 \pm 0.0383$ |
| 2 | 3 | $-5.36 \pm 0.0689$ | $\mathbf{-6.99 \pm 0.0823}$ | $-3.96 \pm 0.0252$ |
| 2 | 4 | $-5.08 \pm 0.0704$ | $-7.02 \pm 0.0853$ | $\mathbf{-3.91 \pm 0.0241}$ |

Table 4
For each training set, the table shows the average (over 30 runs) test set log-likelihood of each of 3 HPMs (with 2, 3, and 4 processes) on each of 3 synthetic data sets (generated with 2, 3, and 4 processes). Each cell is reported as mean ± standard deviation. NOTE: All values in this table are $*10^5$.

the noise in the training set because the model must do well on data it has not seen in training.

## 3.2. *Real data*

### 3.2.1. *HPMs for model comparison*

In this section, we create several HPMs for the sentence-picture fMRI data and evaluate them in terms of the log-likelihood of the data under each model. This evaluation is done in the context of 5-fold cross-validation, meaning that the 40 trials are split into 5 folds of 8 trials each. Each fold is held out in turn as a test set while the models are trained on the other 32 trials. The training was done using Algorithm 1 with regularization, with weight $\gamma = 20$. Since log-likelihood does not have concrete units, we compare the score of each model to the log-likelihood score achieved by the naive method of averaging all training trials together and predicting this average trial for each test trial.

Table 5 presents the improvement of 4 models over a baseline, averaged over 5 folds of cross-validation. Figure 9 presents the same information graphically. The baseline is computed by predicting the mean of the training trials for each test trial. The 4 models are an HPM with 2 processes (ReadSentence and ViewPicture) with non-overlapping responses (HPM-GNB), HPMs with 2 processes (ReadSentence and ViewPicture) with known and unknown timing for ReadSentence and ViewPicture (HPM-2-K and HPM-2-U respectively), and an HPM with 3 processes (ReadSentence, ViewPicture, and Decide) with known timing for ReadSentence and ViewPicture (HPM-3-K). Note that all of the models used all available voxels for each participant. More details of each model are explained below.

The HPM-GNB model approximates a Gaussian Naive Bayes model (Mitchell (1997))through the HPM framework. It has two processes: ReadSentence and ViewPicture. The duration of each process is 8 seconds (16 images) so that they may not overlap. HPM-GNB has a variance for each voxel, and incorporates the knowledge that each trial has exactly one instance of ReadSentence and exactly one of ViewPicture. It is the same as HPM-2-K (presented in the next section) except that the processes are shorter so that they may not overlap.

HPM-2-K is a 2-process HPM containing just ReadSentence and ViewPicture, each of which has a dura-

| Participant | HPM-GNB | HPM-2-K | HPM-2-U | HPM-3-K |
|---|---|---|---|---|
| A | $14040 \pm 3304$ | $12520 \pm 1535$ | $14720 \pm 1970$ | $11780 \pm 3497$ |
| B | $14460 \pm 2555$ | $14580 \pm 1011$ | $15960 \pm 1790$ | $7380 \pm 2439$ |
| C | $12860 \pm 4039$ | $14080 \pm 1794$ | $15460 \pm 2542$ | $7500 \pm 1329$ |
| D | $12140 \pm 1276$ | $13700 \pm 943$ | $15720 \pm 1264$ | $10360 \pm 1408$ |
| E | $14340 \pm 1941$ | $17140 \pm 1236$ | $17560 \pm 1484$ | $10100 \pm 4909$ |
| F | $16180 \pm 3671$ | $17400 \pm 4064$ | $18040 \pm 4060$ | $8180 \pm 1886$ |
| G | $12160 \pm 680$ | $14680 \pm 942$ | $14940 \pm 1358$ | $5740 \pm 1820$ |
| H | $14120 \pm 1281$ | $14860 \pm 811$ | $16160 \pm 1699$ | $7360 \pm 4634$ |
| I | $11460 \pm 2201$ | $13080 \pm 2572$ | $14260 \pm 2384$ | $10420 \pm 2046$ |
| J | $12140 \pm 2509$ | $12840 \pm 1301$ | $14420 \pm 3391$ | $7960 \pm 2907$ |
| K | $14080 \pm 2983$ | $14620 \pm 2190$ | $17120 \pm 2410$ | $8800 \pm 3044$ |
| L | $17820 \pm 2716$ | $20200 \pm 1580$ | $19980 \pm 2494$ | $13700 \pm 3519$ |
| M | $12940 \pm 1205$ | $12680 \pm 1796$ | $14560 \pm 1236$ | $9240 \pm 1677$ |

Table 5
Improvement of test-set log-likelihood for 4 models over predicting the average training trial computed using 5-fold cross validation for 13 subjects. Each cell reports the mean plus or minus 1 standard deviation of the model log-likelihood minus the naive method log-likelihood over the 5 folds.

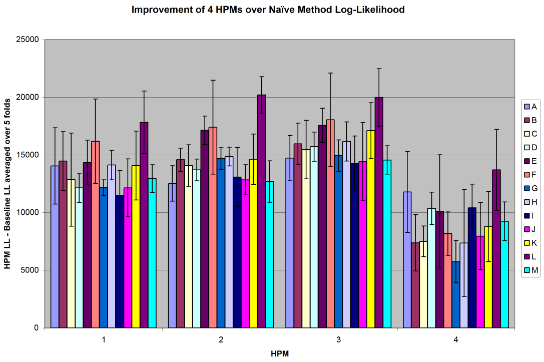

Fig. 9. Improvement in all 13 participants (A-M in different colors) of 4 HPMs over a baseline method that predicts the average training trial for every test trial. The values plotted are the mean over 5 folds of cross validation of the model log-likelihood minus the baseline log-likelihood, and the error bars represent one standard deviation. From left to right on the x-axis, the models are HPM-GNB, HPM-2-K, HPM-2-U, HPM-3-K.

tion of 12 seconds (a reasonable length for the hemodynamic response function). The 'K' stands for 'known,' meaning that the timing of these two processes is assumed to be known. More specifically, it is assumed that each process starts exactly when its corresponding stimulus is presented. In other words, $\Omega = \{0\}$ for both processes. For a given test set trial, the configurations for HPM-2-K are shown in Table 6. The configurations for a training set trial are restricted to those containing the correct order of the processes (based on the order of the stimuli).

| $c$ | $\pi(i_1)$ | $\lambda(i_1)$ | $O(i_1)$ | $\pi(i_2)$ | $\lambda(i_2)$ | $O(i_2)$ |
|---|---|---|---|---|---|---|
| 1 | S | 1 | 0 | P | 17 | 0 |
| 2 | P | 1 | 0 | S | 17 | 0 |

Table 6
Configurations for a test set trial under HPM-2-K. $i_1$ is the first process instance in the trial, and $i_2$ is the second process instance. $\pi(i)$, $\lambda(i)$, and $O(i)$ are the process ID, landmark, and offset, respectively, for process instance $i$. For supervised training, there is only one configuration for each training trial.

HPM-2-U is also a 2-process HPM containing just ReadSentence and ViewPicture. In this case though, the 'U' stands for 'unknown,' meaning that we allow a small amount of uncertainty about the start times of the processes. In HPM-2-U, the $\Omega$ for each process is $\{0, 1\}$, which translates to delays of 0 or 0.5 seconds (0 or 1 images) following the relevant stimulus presentation. The configurations for the test set in this case are in Table 7. Again, during supervised training, the configurations for a given training trial are limited to those that have the correct process ordering. Since the true offset for the processes are unknown even during training, there are still 4 configurations for each training trial.

| $c$ | $\pi(i_1)$ | $\lambda(i_1)$ | $O(i_1)$ | $\pi(i_2)$ | $\lambda(i_2)$ | $O(i_2)$ |
|---|---|---|---|---|---|---|
| 1 | S | 1 | 0 | P | 17 | 0 |
| 1 | S | 1 | 1 | P | 17 | 0 |
| 1 | S | 1 | 0 | P | 17 | 1 |
| 1 | S | 1 | 1 | P | 17 | 1 |
| 2 | P | 1 | 0 | S | 17 | 0 |
| 2 | P | 1 | 1 | S | 17 | 0 |
| 2 | P | 1 | 0 | S | 17 | 1 |
| 2 | P | 1 | 1 | S | 17 | 1 |

Table 7
Configurations for a test set trial using HPM-2-U. For supervised training where we assume the order of the stimuli is known, there will be only four configurations for each training trial.

HPM-3-K adds a Decide process to the previous 2 models, also with a duration of 12 seconds. The Decide process has $\Omega = \{0, 1, 2, 3, 4, 5, 6, 7\}$, allowing the process to start with a delay of 0 to 3.5 seconds from the second stimulus presentation. For these HPMs, we can use the information we have about the participant's reaction time by assuming that the Decide process must begin before the participant pushes the button. This means that for each trial, we can also eliminate any configurations in which the Decide process starts after the button press; that is, any configurations for which $o_3 > \mathit{reaction\_time}$. Note that this implies that different trials can have different sets of configurations since reaction times varied from trial to trial. If the participant did not press the button for some trial, all offsets are considered for the Decide process. The test set configurations for HPM-3-K for a trial with a reaction time of 2.6 seconds are given in Table 8. Since the nearest preceding image to the button press corresponds to offset 5, we have removed configurations with $o_3 \in \{6, 7\}$. To do supervised training, we only use the configurations with the correct ordering of ReadSentence and ViewPicture.

The first thing to note about Figure 9 is that all 4 HPMs show significant improvement over the naive baseline method of predicting the mean training trial for all test trials. While the differences between models for each individual subject are not all significant, the cross-subject trend indicates that HPM-2-K slightly outperforms HPM-GNB, implying that modeling the overlap of the ReadSentence and ViewPicture process response signatures can be advantageous. HPM-2-U generally performs even better, indicating that it can be helpful to allow some uncertainty as to the onset of the process instances. Interestingly, HPM-3-K shows the least improvement over the naive method, despite the fact that some kind of third process must be occurring in this experiment. One possible explanation for the relatively poor performance of this model could be that the possible offsets for Decide are not modeled properly. Another possibility is that the added complexity of HPM-3-K requires more training data than is available.

| $c$ | $\pi(i_1)$ | $\lambda(i_1)$ | $O(i_1)$ | $\pi(i_2)$ | $\lambda(i_2)$ | $O(i_2)$ | $\pi(i_3)$ | $\lambda(i_3)$ | $O(i_3)$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | S | 1 | 0 | P | 17 | 0 | D | 17 | 0 |
| 2 | S | 1 | 0 | P | 17 | 0 | D | 17 | 1 |
| 3 | S | 1 | 0 | P | 17 | 0 | D | 17 | 2 |
| 4 | S | 1 | 0 | P | 17 | 0 | D | 17 | 3 |
| 5 | S | 1 | 0 | P | 17 | 0 | D | 17 | 4 |
| 6 | S | 1 | 0 | P | 17 | 0 | D | 17 | 5 |
| 7 | P | 1 | 0 | S | 17 | 0 | D | 17 | 0 |
| 8 | P | 1 | 0 | S | 17 | 0 | D | 17 | 1 |
| 9 | P | 1 | 0 | S | 17 | 0 | D | 17 | 2 |
| 10 | P | 1 | 0 | S | 17 | 0 | D | 17 | 3 |
| 11 | P | 1 | 0 | S | 17 | 0 | D | 17 | 4 |
| 12 | P | 1 | 0 | S | 17 | 0 | D | 17 | 5 |

Table 8
Configurations for a test set trial under HPM-3-K. The reaction time for this trial is 2.6 seconds, which corresponds to offset 5 for the Decide process, so all configurations with offsets greater than 5 for this process have been eliminated for this trial.

### 3.2.2. *Interpreting HPMs*

In the previous section, we estimated how well different HPMs would predict unseen test data using cross-validation. In this section, we examine the parameters of the HPMs to try to understand what the models are learning better. For the results below, we trained HPM-3-K on all trials for Participant L.

The learned timing distribution for the Decide process is shown in Table 9. The distribution is similar to the distribution of reaction times for this subject. For trials where the subject did not press the button, the program tended to choose the last possible offset for the Decide process.

We can also visualize the model by looking at the

| Offset | Θ |
|---|---|
| 0 | 0.2713 |
| 1 | 0.0762 |
| 2 | 0.1006 |
| 3 | 0.1006 |
| 4 | 0.0762 |
| 5 | 0.1250 |
| 6 | 0.0030 |
| 7 | 0.2470 |

Table 9
Learned timing distribution for the Decide process for Participant L using HPM-3-K, for offsets from the second stimulus presentation 0-7 images (0-3.5 seconds). The Θ values are the parameters of the multinomial distribution over these offsets.

learned process response signatures. Since a response signature contains parameters over space and time, one option is to average the parameters of each voxel over time. This value with respect to the ReadSentence process is plotted for each voxel (in that voxel's location) in Figure 10, for the View Picture process in Figure 11, and for the Decide process in Figure 12. We can also plot the time courses for individual voxels. For instance, the time course of the darkest red voxel from Figure 12 (in the top middle slice) is shown in Figure 13.

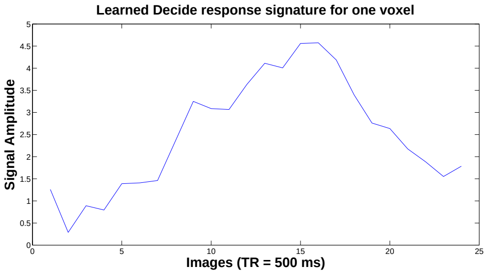

Fig. 13. The time course of the darkest red voxel in Figure 12, for the Decide process in participant L under HPM-3-K.

### 3.2.3. *Improvement of data log-likelihood with Shared HPMs*

In this section we report experiments on the same sentence-picture dataset using the methods in Section 2.4, which attempt to share, when appropriate, process parameters across neighboring voxels. Since the results reported in this section are intended to be only a proof of concept for incorporating sharing of HPM parameters across spatially contiguous voxels in HPM learning, they are limited to data from a single human participant, consisting of 4698 voxels (participant D). For each trial, we considered only the first 32 images (16 seconds) of brain activity.

We model the activity in the brain using a Hidden Process Model with two processes, ReadSentence and ViewPicture. The start time of each process instance is assumed to be known in advance (i.e., the offset values for each process are $\{0\}$, so each process instance begins exactly at the landmark corresponding to the relevant stimulus). This is equivalent to HPM-GNB from the previous section. We evaluated the performance of the models using the average log-likelihood over the held-out trials in a leave-two-out cross-validation approach, where each fold contains one example in which the sentence is presented first, and one example in which the picture is presented first.

Our experiments compared three HPM models. The first model, which we consider a baseline, consists of a standard Hidden Process Model (StHPM) learned independently for each voxel (or equivalently, a standard HPM with no parameter sharing). The second model is a Hidden Process Model where all voxels in an ROI share their Hidden Process parameters (ShHPM). ShHPM is learned using Algorithm 2. The third model is a Hierarchical Hidden Process Model (HieHPM) learned using Algorithm 3.

The first set of experiments, summarized in Table 10, compares the three models based on their performance in the calcarine sulcus of the visual cortex (CALC). This is one of the ROIs actively involved in this cognitive task, containing 318 voxels for this participant. The training set size was varied from 6 examples to all 40 examples, in increments of two. Sharing parameters for groups of voxels proved beneficial, especially for small training sets. As the training set size increased, the performance of ShHPM degraded because the bias in the inaccurate assumption that all voxels in CALC share parameters was no longer outweighed by the corresponding reduction in the variance of the parameter estimates. However, we will see that in other ROIs, this assumption holds and in those cases the gains in performance may be quite large.

As expected, the hierarchical model HieHPM performed better than both StHPM and ShHPM because it takes advantage of shared Hidden Process Model parameters while not making the restrictive assumption of sharing across all voxels in a ROI. The largest difference in performance between HieHPM and StHPM is observed at 6 examples, in which case StHPM fails to learn a reasonable model while the highest difference between HieHPM and ShHPM occurs at the maximum number of examples, presumably when the bias of ShHPM is most harmful. As the number of training ex-

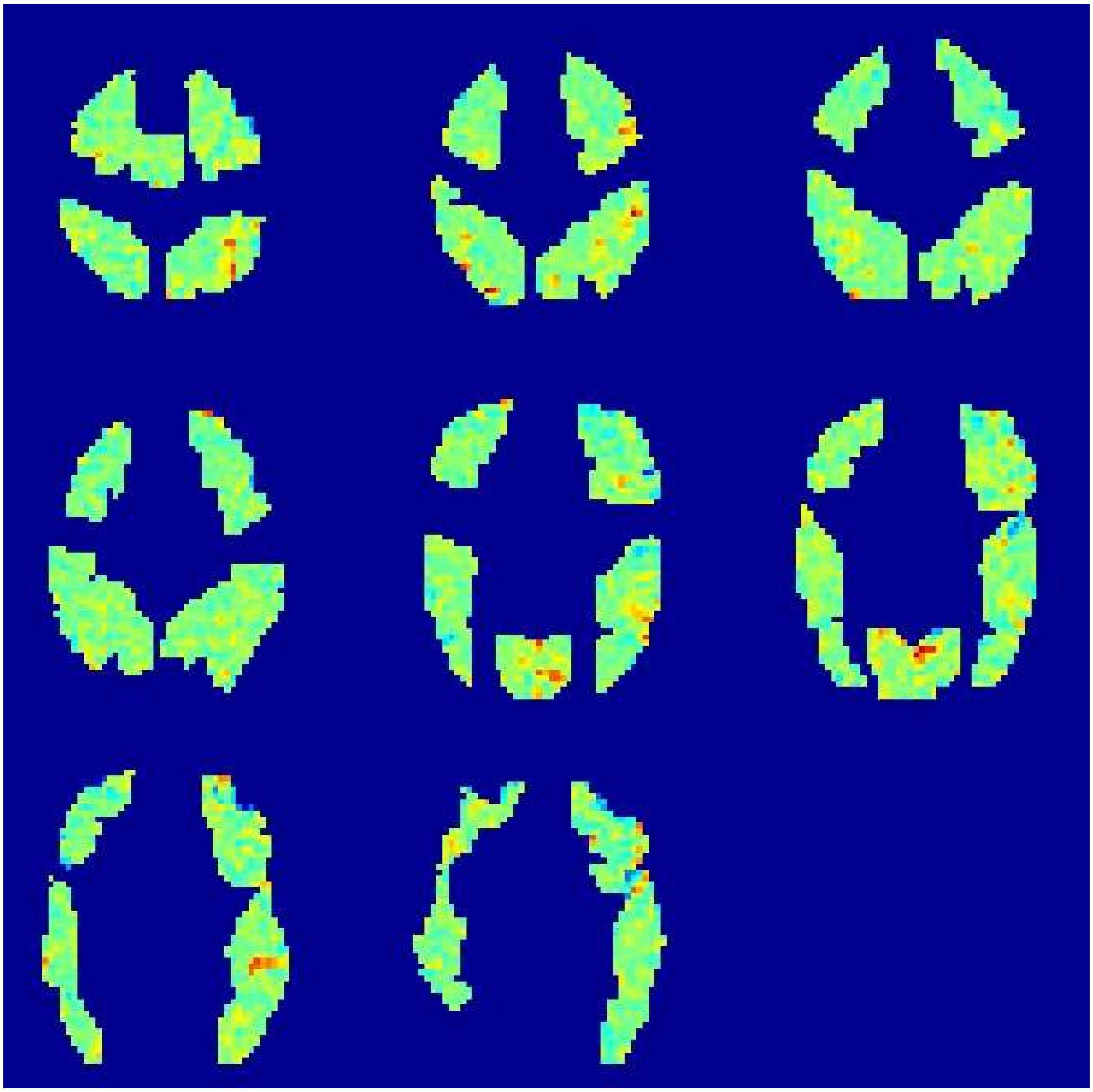

Fig. 10. The learned spatial-temporal response for the ReadSentence process in participant L under HPM-3-K, averaged over time. Red corresponds to higher values, blue to lower values. Images are displayed in radiological convention (the left hemisphere is on the right of each slice). Anterior is higher, posterior is lower.

amples increases, both StHPM and HieHPM tend to perform better and better and one can see that the marginal improvement in performance obtained by the addition of two new examples tends to shrink as both models approach convergence. While with an infinite amount of data, one would expect both StHPM and HieHPM to converge to the true model, at 40 examples, HieHPM still outperforms the baseline model StHPM by a difference of 106 in terms of average log-likelihood, which is an improvement of $e^{106}$ in terms of data likelihood.

Perhaps the best measure of the improvement of HieHPM over the baseline StHPM is the number of examples needed by StHPM to achieve the same performance as HieHPM. These results show that on average, StHPM needs roughly 2.9 times the number of examples needed by HieHPM in order to achieve the same level of performance in CALC.

The last column of Table 10 displays the number of clusters into which HieHPM partitioned CALC. At small sample sizes HieHPM uses only one cluster of voxels and improves performance by reducing the variance in the parameter estimates. However, as the train-

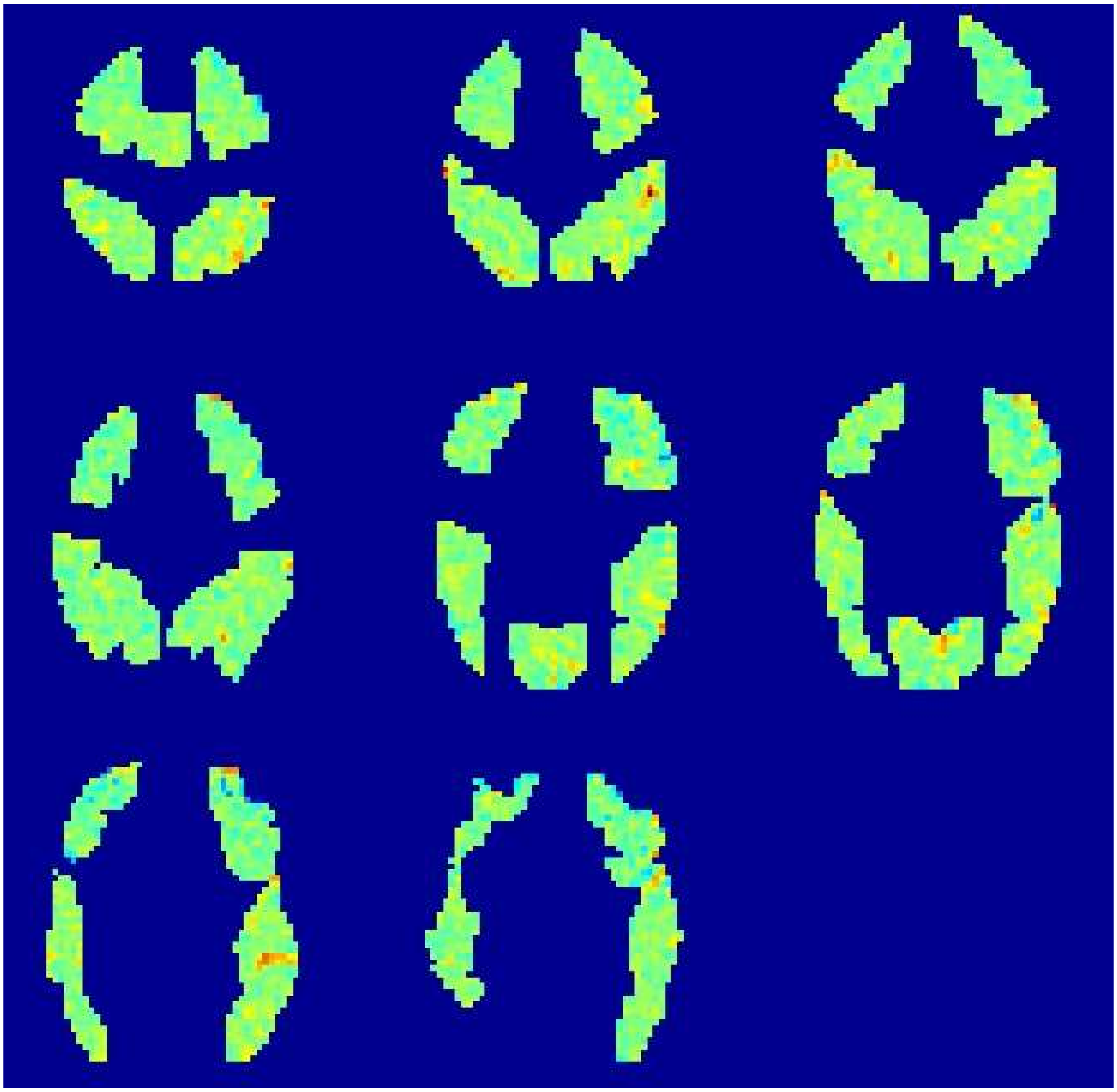

Fig. 11. The learned spatial-temporal response for the ViewPicture process in participant L under HPM-3-K, averaged over time. Red corresponds to higher values, blue to lower values. Images are displayed in radiological convention (the left hemisphere is on the right of each slice). Anterior is higher, posterior is lower.

ing set size increases, HieHPM improves by finding more and more refined partitions. This number of shared voxel sets tends to stabilize around 60 clusters once the number of examples reaches 30, which yields an average of more than 5 voxels per cluster in this case. For a training set of 40 examples, the largest cluster has 41 voxels while many clusters consist of only one voxel.

The second set of experiments (see Table 11) describes the performance of the three models for each of the 24 individual ROIs of the brain, and also when trained over the entire brain. While ShHPM was biased in CALC, we see here that there are several ROIs in which it makes sense to assume that all voxels share parameters. In fact, in most of these regions, HieHPM finds only one cluster of voxels. ShHPM outperforms the baseline model StHPM in 18 out of 24 ROIs while HieHPM outperforms StHPM in 23 ROIs. One may ask how StHPM can possibly outperform HieHPM on any ROI, since StHPM is a special case of HieHPM. The explanation is that the hierarchical approach is subject to local minima since it is a greedy process that does not look beyond the current potential split for a finer grained partition. Fortunately, this problem is very rare in our experiments.

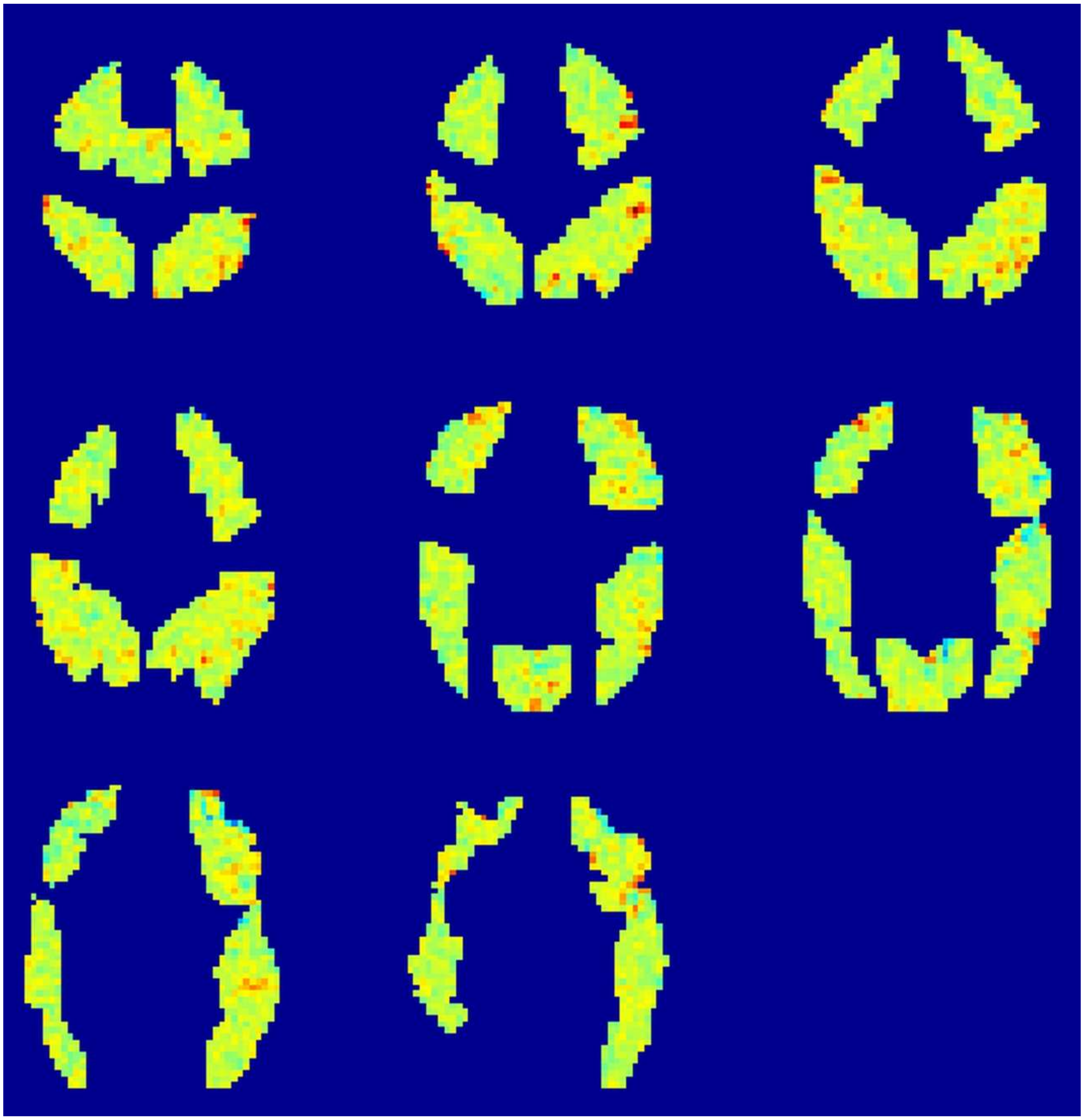

Fig. 12. The learned spatial-temporal response for the Decide process in participant L under HPM-3-K, averaged over time. Red corresponds to higher values, blue to lower values. Images are displayed in radiological convention (the left hemisphere is on the right of each slice). Anterior is higher, posterior is lower.

Over the whole brain, HieHPM outperforms StHPM by 1792 in terms of log-likelihood while ShHPM outperforms StHPM only by 464. ShHPM also makes a restrictive sharing assumption and therefore HieHPM emerges as the recommended approach.

As mentioned above, HieHPM automatically learns clusters of voxels that share their process response signature parameters. Figure 14 shows these learned clusters in slice five of the eight slices. Neighboring voxels that were assigned by HieHPM to the same cluster are pictured in the same color. Note that there are several very large clusters in this picture. This may be because it makes sense to share voxel parameters at the level of an entire ROI if the cognitive process does not activate voxels in this ROI. However, large clusters are also found in areas like CALC, which we know is directly involved in visual processing.

In Figure 15 we present the learned ReadSentence process for the voxels in slice five of CALC. The graphs corresponding to voxels that belong to the same cluster

| Trials | StHPM | ShHPM | HieHPM | Cells |
|---|---|---|---|---|
| 6 | -30497 | -24020 | -24020 | 1 |
| 8 | -26631 | -23983 | -23983 | 1 |
| 10 | -25548 | -24018 | -24018 | 1 |
| 12 | -25085 | -24079 | -24084 | 1 |
| 14 | -24817 | -24172 | -24081 | 21 |
| 16 | -24658 | -24287 | -24048 | 36 |
| 18 | -24554 | -24329 | -24061 | 37 |
| 20 | -24474 | -24359 | -24073 | 37 |
| 22 | -24393 | -24365 | -24062 | 38 |
| 24 | -24326 | -24351 | -24047 | 40 |
| 26 | -24268 | -24337 | -24032 | 44 |
| 28 | -24212 | -24307 | -24012 | 50 |
| 30 | -24164 | -24274 | -23984 | 60 |
| 32 | -24121 | -24246 | -23958 | 58 |
| 34 | -24097 | -24237 | -23952 | 61 |
| 36 | -24063 | -24207 | -23931 | 59 |
| 38 | -24035 | -24188 | -23921 | 59 |
| 40 | -24024 | -24182 | -23918 | 59 |

Table 10
The effect of training set size on the average log-likelihood (higher numbers/lower absolute values are better) of the three models in the calcarine sulcus of the visual cortex (CALC), and the number of clusters into which the algorithm segmented the ROI.

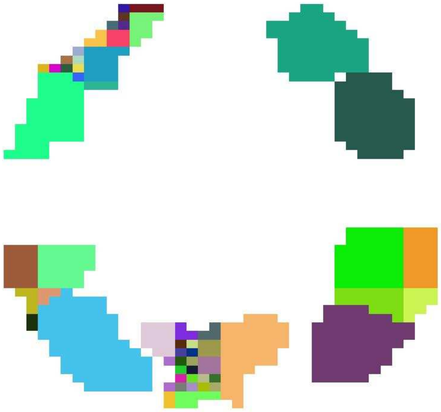

Fig. 14. Clusters learned by HieHPM in slice 5 of the brain

have the same color as in Figure 14. For readability, we only plot the base processes, disregarding the learned scaling constants which specify the amplitude of the response in each voxel within a given cluster. The increased smoothness of these learned time courses suggest that the discontinuities from figure 13 may be partially due to the sparsity of data that occurs when each voxel is estimated independently.

| ROI | Voxels | StHPM | ShHPM | HieHPM | Cells |
|---|---|---|---|---|---|
| CALC | 318 | -24024 | -24182 | -23918 | 59 |
| LDLPFC | 440 | -32918 | -32876 | -32694 | 11 |
| LFEF | 109 | -8346 | -8299 | -8281 | 6 |
| LIPL | 134 | -9889 | -9820 | -9820 | 1 |
| LIPS | 236 | -17305 | -17187 | -17180 | 8 |
| LIT | 287 | -21545 | -21387 | -21387 | 1 |
| LOPER | 169 | -12959 | -12909 | -12909 | 1 |
| LPPREC | 153 | -11246 | -11145 | -11145 | 1 |
| LSGA | 6 | -441 | -441 | -441 | 1 |
| LSPL | 308 | -22637 | -22735 | -22516 | 4 |
| LT | 305 | -22365 | -22547 | -22408 | 18 |
| LTRIA | 113 | -8436 | -8385 | -8385 | 1 |
| RDLPFC | 349 | -26390 | -26401 | -26272 | 40 |
| RFEF | 68 | -5258 | -5223 | -5223 | 1 |
| RIPL | 92 | -7311 | -7315 | -7296 | 11 |
| RIPS | 166 | -12559 | -12543 | -12522 | 20 |
| RIT | 278 | -21707 | -21720 | -21619 | 42 |
| ROPER | 181 | -13661 | -13584 | -13584 | 1 |
| RPPREC | 144 | -10623 | -10558 | -10560 | 1 |
| RSGA | 34 | -2658 | -2654 | -2654 | 1 |
| RSPL | 252 | -18572 | -18511 | -18434 | 35 |
| RT | 284 | -21322 | -21349 | -21226 | 24 |
| RTRIA | 57 | -4230 | -4208 | -4208 | 1 |
| SMA | 215 | -15830 | -15788 | -15757 | 10 |
| Full Brain | 4698 | -352234 | -351770 | -350441 | 299 |

Table 11
Performance of the three models over whole brain and in several ROIs when learned using all 40 examples

### 3.2.4. *SPM analysis*

For comparison, we provide an SPM analysis of the sentence-picture dataset. To detect activation in each of the conditions the data for each participant were fit to a general linear model using SPM2 software (`http://www.fil.ion.ucl.ac.uk/spm/spm2.html`). The data were first corrected for slice acquisition timing using sinc-interpolation, and then modeled using regressors for each stimulus condition and run consisting of a box-car representing the onset and duration of each stimulus presentation convolved with the canonical double-gamma HRF model available in SPM2. Additional covariates implemented a high-pass filter

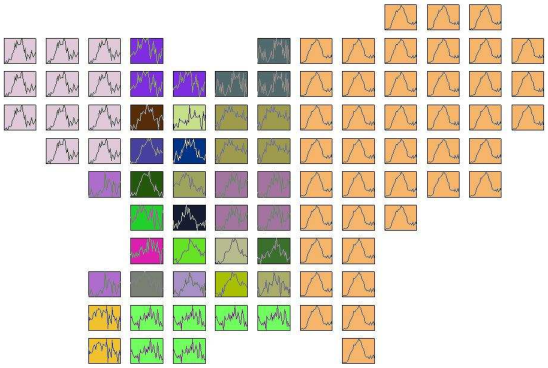

Fig. 15. Learned ReadSentence process for all voxels in slice 5 of the Visual Cortex

(cutoff at 128 s) and serial correlations in the errors were modeled with an AR(1) process. No spatial smoothing, resampling, or 3D motion correction of the data were employed in SPM2, for the sake of consistency with the real and synthetic data that were used in the Hidden Process Models. Parameter estimates and their contrasts were evaluated on a voxel-wise basis for statistcal significance by t-tests, at a threshold of $p < 0.05$, corrected for multiple comparisons using Gaussian random field theory. The activation map for Participant L for the sentence stimuli (minus fixation) is shown in Figure 16 and the corresponding map for the picture stimuli is in Figure 17.

Consistent with previous imaging studies of sentence-picture verification (Carpenter et al. (1999); Reichle et al. (2000)), this participant showed activation reliably related to sentence presentation in left parietal cortex (Fig. 16, slices 1-4), consistent with the requirement to process the visuo-spatial content of the sentences, as well as in a network of reading-related areas including occipital cortex (slices 5-8), left temporal cortex (Wernicke's area, slices 5-7), and left inferior frontal gyrus (Broca's area, slices 7-8). Activation reliably related to picture presentation was found in the same areas of left parietal cortex in this participant (Fig. 17, slices 1-4), but there was less evidence of activation in the left occipital-temporal-frontal reading network for this condition.

Note the differences among the figures we have presented in Section 3. Figures 10, 11, and 12 show the parameters of the process response signatures of the three processes in HPM-3-K averaged over time for participant L. Figures 14 and 15 show the clustering and time courses of the process response signatures learned for participant D under HPM-GNB. These signatures are combined based on the timing of the process instances to predict fMRI data. Figures 16 and 17 show activity maps for responses to the sentence and picture stimuli for participant L. These maps show the voxels whose data are significantly correlated with each type of stimuli. The differences among these figures reflect the different objectives of the methods that produced them.

## 4. Discussion

In this paper, we have presented Hidden Process Models, a generative model for fMRI data based on a set of assumed mental processes. We have given the formalism for the model, algorithms to infer characteristics of the processes underlying a dataset, and algorithms for estimating the parameters of the model from data under various levels of uncertainty. Additionally, we presented a preliminary approach to improve HPMs by taking advantage of the assumption that some neighboring voxels share parameters, including an algorithm to automatically discover sets of spatially contiguous voxels whose HPM parameters can be shared. We presented experimental results on synthetic data demonstrating that our algorithms can estimate HPM parameters, classify new data, and identify the number of processes underlying the data. We explored a real sentence-picture verification dataset using HPMs to demonstrate that HPMs can be used to compare different cognitive models, and to show that sharing parameters can improve our results.

HPMs facilitate the comparison of multiple theories of the cognitive processing underlying an fMRI experiment. Each theory can be specified as an HPM: a set of processes, each with a timing distribution and response signature parameters, and a set of configurations, each specifying a possible instantiation of the processes that may explain a window of data. The parameters of the processes may be learned from an fMRI dataset. The models may then be compared in terms of their data log-likelihood on a separate test set, or using cross-validation.

Another contribution of Hidden Process Models is the ability to estimate the spatial-temporal response to events whose timing is uncertain. While the responses for simple processes that are closely tied to known stimulus timings can be reasonably estimated with the General Linear Model (Dale & Buckner (1997)) (like ViewPicture and ReadSentence), HPMs allow processes whose onset is uncertain (like Decide). Note that this is not equivalent to letting the GLM search for the Decide process by running it once for each start time in

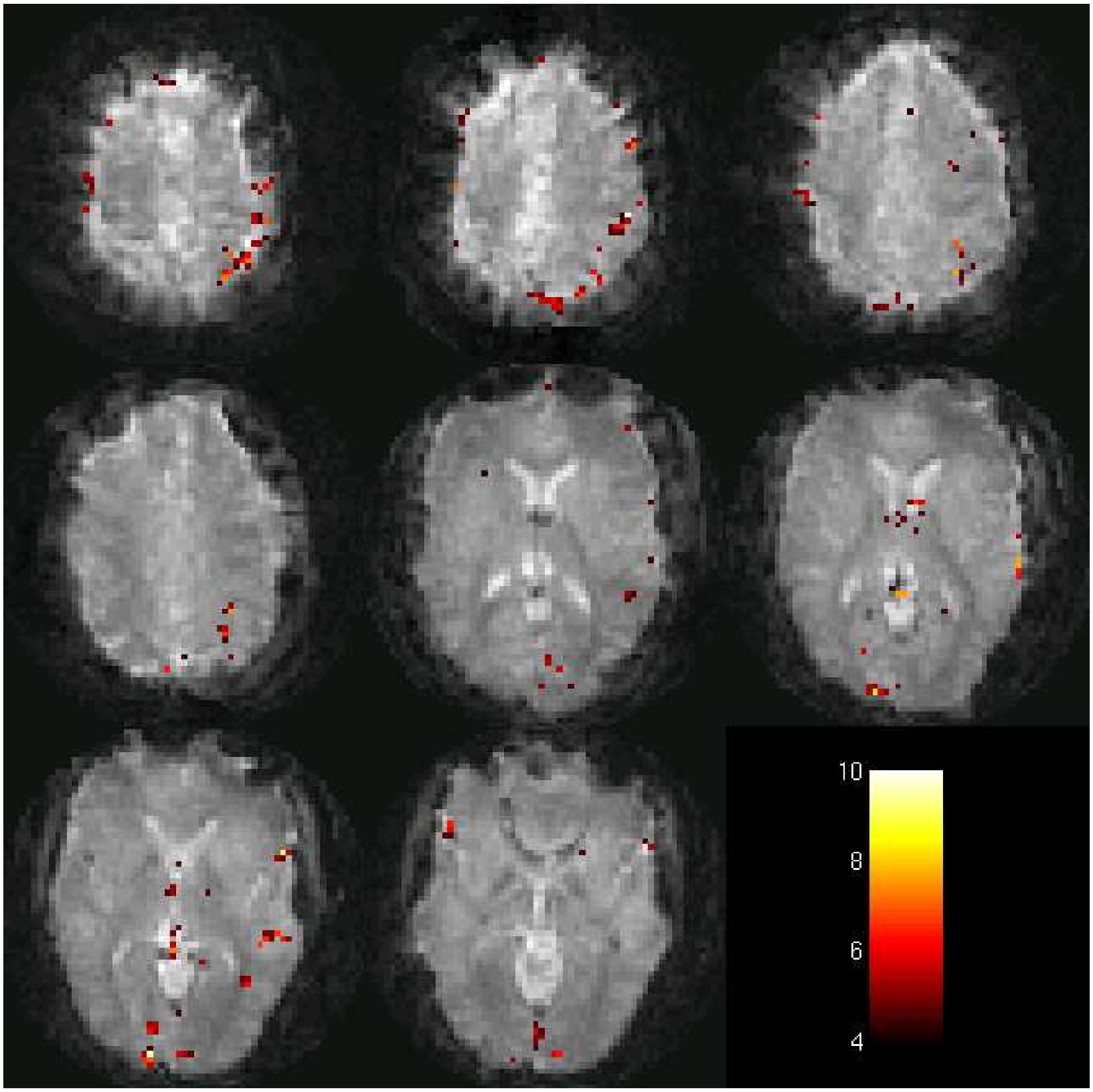

Fig. 16. The SPM activation map for Participant L for the sentence stimuli minus fixation. Images are displayed in radiological convention (the left hemisphere is on the right of each slice).

the window. HPMs allow the start time to vary according to a probability distribution; this means that the start time can change from trial to trial within the experiment to help model trials with varying difficulty or strategy differences for example. In practice, we have observed that HPMs do indeed assign different onsets to process instances in different trials.

In the interest of comparison and placing HPMs in the field, let us highlight three major differences between analysis with HPMs and analysis with SPM (Friston (2003)). A conventional analysis of the sentence-picture data using SPM convolves the timecourse of the sentence and picture stimuli with a canonical hemodynamic response, evaluating the correlation of the timecourse of each voxel with that convolved signal, and thresholding the correlations at an appropriate level. This analysis would support claims about which regions were activated for each type of stimulus (e.g. region R is more active under condition S than condition P). It is not clear how an SPM analysis would treat any of the cognitive processes presented above with uncertain timings. The first major difference between this SPM approach and HPM analysis is that HPMs estimate a hemodynamic response instead of assuming a canonical form. Sec-

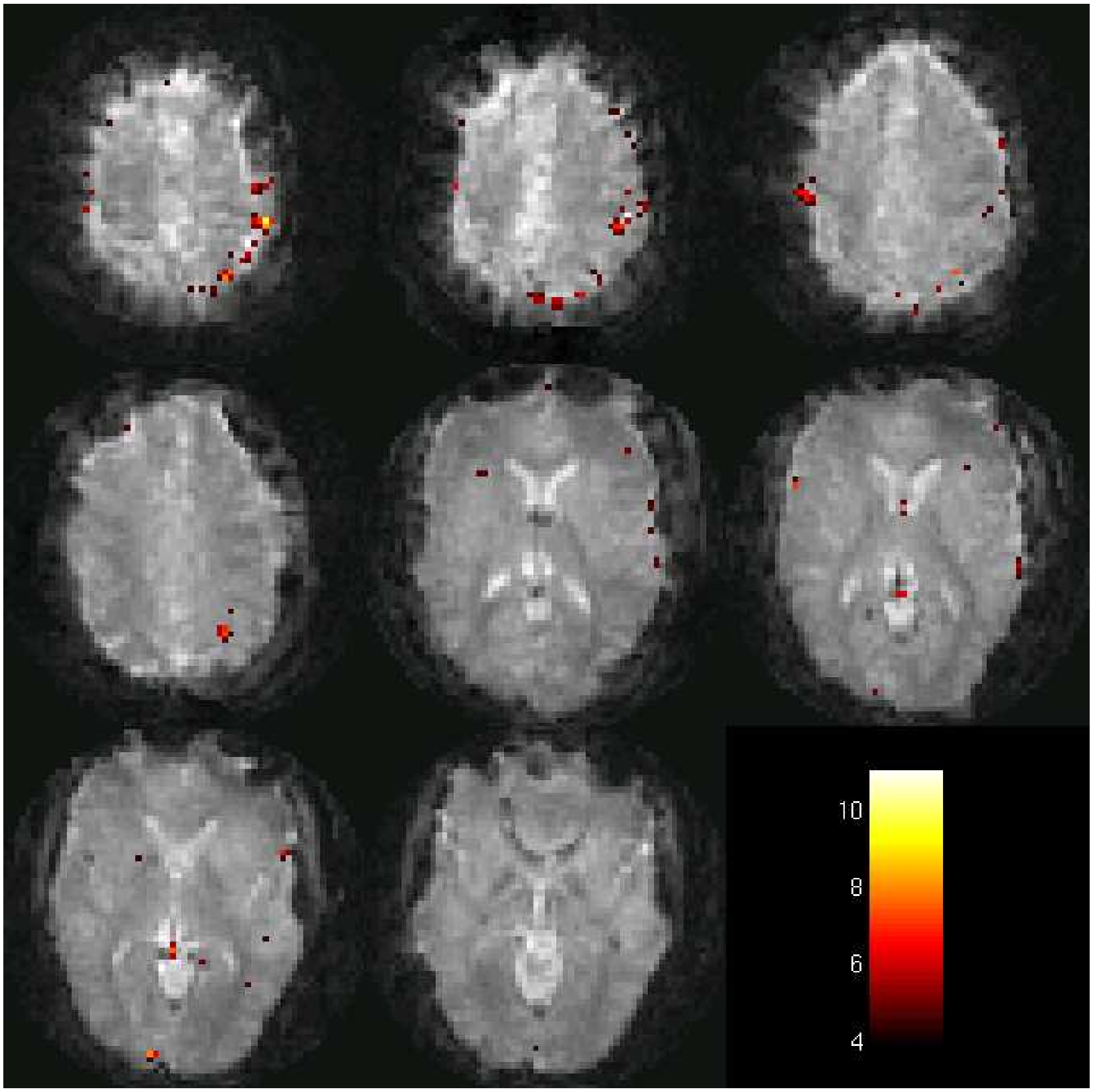

Fig. 17. The SPM activation map for Participant L for the picture stimuli minus fixation. Images are displayed in radiological convention (the left hemisphere is on the right of each slice).

ondly, HPMs allow unknown timing for cognitive processes, and estimate parameters of a probability distribution describing the timing. Finally, the key claim that HPMs attempt to make is fundamentally different from that of SPM. HPMs provide evidence for claims about competing models of the mental processes underlying the observed sequence of fMRI activity (e.g. model M outperforms model N). We believe that HPMs and SPM can be complementary approaches for fMRI data analysis.

Based on our experience, we can provide a few pieces of advice to those interested in using HPMs. First, since HPMs are an exploratory tool, one must be careful not to jump to conclusions about the processes learned from them. In this paper, we have named a number of processes (ViewPicture, Decide, etc.). These names are only for convenience. Processes are defined by their offset values, the probability distribution over the offset values, and their response signatures. Furthermore, the parameters learned also depend on the configurations used for the training data. The set of configurations associated with an HPM reflects the prior knowledge and assumptions built into the learning algorithm; this set is part of the bias of the model. Changing the set of configurations could result in learning different parameters for each process. The response signature for each process should be carefully examined to determine whether the name it was given for convenience is a good descriptor.

A second suggestion for researchers interested in applying HPMs to fMRI data regards the benefit of designing experiments with HPMs in mind. HPMs perform best if the arrangement of process instances in the data renders the processes identifiable by isolating and/or varying the overlap between processes. For example, two processes that are always instantiated simultaneously cannot be uniquely separated, so this case should be avoided in the experiment design. Experiment designs that provide natural choices for landmarks like stimulus presentation times and behavioral data are also helpful for limiting the window of possible start times for processes. For example, without recording participants' reaction times in the picture and sentence data, we might model the button press with a similar range of offsets to the Decide process instead of just two offsets corresponding to the button press, which would result in more configurations, and thus more complexity. Finally, HPMs are of most interest for studying processes of uncertain onset. If the timing of all processes of interest is known to be directly tied to stimuli, there are a number of other analysis methods that will perform just as well as HPMs.

A third note about the application of HPMs is that the main computational complexity issue to address is the number of configurations that must be created to encode the prior knowledge we have from the experiment design, since adding configurations increases the size of the linear system to be solved in training HPMs. The number of configurations can be lessened by limiting the number of offsets for any given process and providing the sequence of some subset of processes (as we have done with ViewPicture and ReadSentence).

Finally, we conclude with a discussion of future work. Hidden Process Models show promise as a new method for fMRI data analysis, and will become even more useful as they become more general. We are working to provide more modeling options for scientists using HPMs, including regularization options for allowing priors on the areas of the brain in which we expect a process to be active. We are trying to reduce the sample complexity of the model by allowing the process response signatures to be modeled using weights on a set of basis functions, which could significantly reduce the number of parameters to be estimated in training an HPM. Modelling the process response signatures with continuous functions could also allow us to model process instance onsets at a finer temporal resolution than the experiment TR, and potentially relax the assumption of fixed-duration responses. We are also looking into ways to relax the linearity assumption present in the current version of HPMs so that models can reasonably incorporate more overlapping processes. We would of course like to explore ways of combining data from multiple participants, and finally, we hope to relax the simplifying assumptions of the parameter sharing approach to accommodate uncertainty in the onset times of the processes.

## 5. Acknowledgements

We thank Marcel Just for providing the data and Francisco Pereira for help pre-processing the data. We also thank them both for helpful discussions about the work.

## References

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

Carpenter, P.A., Just, M.A., Keller, T.A., Eddy, W.F., & Thulborn, K. R., Time course of fMRI-Activation in language and spatial networks during sentence comprehension, s1999, NeuroImage 10(2), 216-224.

Ciuciu, P., Poline, J., Marrelc, G., Idier, J., Pallier, C., & Benali, H., Unsupervised robust nonparametric estimation of the hemodynamic response function for any fMRI experiment, 2003, IEEE Transactions on Medical Imaging, 22(10), 1235-1251.

Cox, D.D., & Savoy, R.L., Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex, 2003, Neuroimage, 19, 261-270.

Dale, A.M., & Buckner, R.L., Selective averaging of rapidly presented individual trials using fMRI, 1997, Human Brain Mapping, 5, 329-340.

Dale, A.M., Optimal experiment design for event-related fMRI, 1999, Human Brain Mapping, 8, 109-114.

Dempster, A., Laird, N., & Rubin, D., Maximum likelihood from incomplete data via the EM algorithm, 1977, Journal of the Royal Statistical Society Series B, 39(1), 1-38.

Eddy, W.F., Fitzgerald, M., Genovese, C., Lazar, N., Mockus, A., & Welling, J., The challenge of functional magnetic resonance imaging, 1998, Journal of Computational and Graphical Statistics, 8(3), 545-558.

Faisan, S., Thoraval, L., Armspach, J., & Heitz, F., Hidden Markov multiple event sequence models: A paradigm for the spatio-temporal analysis of fMRI data, 2007, Medical Image Analysis, 11, 1-20.

Formisano, E., & Goebel, R., Tracking cognitive processes with functional MRI mental chronometry, 2003, Current Opinion in Neurobiology, 13, 174-181.

Friston, K.J., Introduction to statistical parametric mapping, 2003, in Human brain function (editors Frackowiak et al.).

Haxby, J.V., Gobbini, M.I., Furey, M.L, Ishai, A., Schouten, J.L., & Pietrini, P., Distributed and overlapping representations of faces and objects in ventral temporal cortex, 2001, Science, 293, 2425-2430.

Haynes, J., & Rees, G., Decoding mental states from brain activity in humans, 2006, Nature Reviews Neuroscience, 7, 523-534.

Henson, R.N.A., Forward inference using functional neuroimaging: dissociations versus associations, 2006, Trends in Cognitive Sciences, 10(2), 64-69.

Henson, R.N.A., Price, C., Rugg, M.D., Turner, R., & Friston, K., Detecting latency differences in event-related BOLD responses: application to words versus nonwords, and initial versus repeated face presentations, 2002, Neuroimage, 15, 83-97.

Højen-Sørensen, P., Hansen, L.K., & Rasmussen, C.E., Bayesian modelling of fMRI time series, 2000, Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS), 12, 754-760.

Hutchinson, R.A., Mitchell, T.M., & Rustandi, I., Hidden Process Models, 2006, Proceedings of the 23rd International Conference on Machine Learning, 433-440.

Kamitani, Y. & Tong, F., Decoding the visual and subjective contents of the human brain, 2005, Nature Neuroscience, 8, 679-685.

Keller, T.A., Just, M.A., & Stenger, V.A., Reading span and the time-course of cortical activation in sentence-picture verification, 2001, Annual Convention of the Psychonomic Society.

Liao, C.H., Worsley, K.J., Poline, P-B., Aston, J.A.D., Duncan, G.H., & Evans, A.C., Estimating the delay of the fMRI response, 2002, Neuroimage, 16(3), 593-606.

Menon, R.S., Luknowsky, D.C., & Gati, J.S., Mental chronometry using latency-resolved functional MRI, 1998, Proceedings of the National Academy of Sciences of the United States of America, 95(18), 10902-10907.

Mitchell, T.M., Hutchinson, R.A., Niculescu, R.S., Pereira, F., Wang, X., Just, M., & Newman, S., Learning to decode cognitive states from brain images, 2004, Machine Learning, 57, 145-175.

Mitchell, T.M., Machine Learning, 1997, McGraw-Hill.

Mitra, P.S., Automated knowledge discovery from functional magnetic images using spatial coherence, 2006, University of Pittsburgh Ph.D. Thesis.

Murphy, K.P., Dynamic Bayesian Networks, 2002, in Probabilistic Graphical Models (ed. Jordan, M.).

Niculescu, R.S., Mitchell, T.M., & Rao, R.B., Bayesian network learning with parameter constraints, 2006, Journal of Machine Learning Research, Special Topic on Machine Learning and Large Scale Optimization, 7, 1357-1383.

Niculescu, R.S., Exploiting parameter domain knowledge for learning in Bayesian networks, 2005, Carnegie Mellon University Ph.D. Thesis, CMU-CS-05-147.

Norman, K.A., Polyn, S.M., Detre, G.J., & Haxby, J.V., Beyond mind-reading: multi-voxel pattern analysis of fMRIdata, 2006, Trends in Cognitive Sciences, 10(9), 424-430.

Palatucci, M.M., & Mitchell, T.M., Classification in very high dimensional problems with handfuls of examples, 2007, Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Springer-Verlag, September, 2007.

Poldrack, R.A., Can cognitive processes be inferred from neuroimaging data?, 2006, Trends in Cognitive Sciences, 10(2), 59-63.

Rabiner, L.R., A tutorial on Hidden Markov Models and selected applications in speech recognition, 1989, Proceedings of the IEEE, 77(2), 257-286.

Reichle, E.D., Carpenter, P.A., & Just, M.A., The neural bases of strategy and skill in sentence-picture verification, 2000, Cognitive Psychology, 40, 261-295.

Zhang, L., Samaras, D., Tomasi, D., Alia-Klein, N., Cottone, L., Leskovjan, A., Volkow, N., & Goldstein, R., Exploiting temporal information in functional magnetic resonance imaging brain data, 2005, Medical Image Computing and Computer-Assisted Intervention, 679-687.