Supporting Information

Gagnepain et al. 10.1073/pnas.1311468111

SI Data

This section reports a note on the outlier exclusion procedure during region of interest analyses.

Outliers were defined as 3 SDs above or below the mean of the difference between a priori contrasts of conditions of interest. One outlier participant was identified during the analysis of the think/no-think phase. This participant was an outlier in three regions of interest: the left and right hippocampus and the right fusiform. During the think/no-think phase, we observed increased activity for think relative to no-think items in both left [$t(22) = 2.75$, $P < 0.01$] and right [$t(22) = 1.79$, $P < 0.05$] hippocampus when this outlier was excluded from the analysis; when the outlier was included, this effect was marginal in the left hippocampus [$t(23) = 1.41$, $P = 0.086$] and absent in the right [$t(23) = 0.61$, $P = 0.27$]. In the right fusiform gyrus, we observed increased activity for think relative to no-think items [$t(22) = 1.95$, $P < 0.05$] when excluding this outlier; when included, this effect was no longer significant [$t(23) = 0.92$, $P = 0.18$]. Note that we did not exclude this outlier participant from the analyses reported in the manuscript. Rather, we report them here for reader consideration.

SI Simulation Methods

This section reports in detail (*i*) the models used to simulate fusiform activity under different suppression accounts, (*ii*) the Markov chain Monte Carlo (MCMC) algorithm to sample the entire space of parameter values for each simulated model and to approximate the posterior distribution of these parameters, and (*iii*) the method to test goodness of fit (GOF) of each model across participants, as well as (*iv*) evidence of MCMC convergence.

General Overview. The goal of these simulations was to compute the theoretical representational dissimilarity matrix ($\mathrm{RDM}_{\text{-t}}$) for a simulated fusiform cortex under different assumptions of about how memory is suppressed. A first factor distinguishing the models corresponded to how voxels were selected and modulated (i.e., voxel selection): (*i*) targeted (i.e., activity dependent, in which a subset of voxels is modulated for each item based on their higher degree of initial activation), (*ii*) random (in which a randomly chosen subset of voxels is selected for each item), and (*iii*) fixed (in which a randomly selected set of voxels is consistently modulated across items). A second factor captured how memory suppression was implemented via (*i*) inhibition (in which voxel activity is divided by some factor), (*ii*) truncated activation (in which memory reinstatement is stopped but not directly inhibited, resulting in fewer voxels remaining active), and (*iii*) retrieval alone (in which activity for no-think items is not modulated at all (and only think items were modulated; see *Model Construction*). After fitting model parameters for each of these models and each participant, and generating the corresponding $\mathrm{RDM}_{\text{-t}}$, we compared which $\mathrm{RDM}_{\text{-t}}$ provided the best fit to the RDM observed in the real fusiform gyrus ($\mathrm{RDM}_{\text{-fus}}$) for each participant. GOF values were then entered into a second-level analysis treating participants as a random effect variable.

To generate the critical theoretical $\mathrm{RDM}_{\text{-t}}$ for each account, we constructed a model M (e.g., targeted inhibition) given some parameters $\theta_1, \theta_2, ..., \theta_N$, which can be formulated as $\mathrm{RDM}_{\text{-t}} = \mathrm{M}(\theta_1, \theta_2, ..., \theta_N)$. For each generative model M, we estimated the values of the parameters that best fit the data. Here, we used a MCMC approach to sample the entire space of parameter values and to approximate the posterior distribution of each parameter. Then, the maximum a posteriori estimate (MAP) (i.e., the mode of the posterior distribution) was taken as the best fit of each parameter and these estimates used to establish the GOF of each model. Note that we repeatedly split the observed $\mathrm{RDM}_{\text{-fus}}$ into two halves so that one half provided a training set used to fit model parameters, and the other half provided a test set to calculate the GOF of the model (i.e., a cross-validation approach).

In this section, we first detail how $\mathrm{RDM}_{\text{-t}}$ was generated under the different theoretical accounts of memory suppression. We then present the MCMC algorithm used to fit model parameters. Finally, we report the cross-validation method used to estimate the GOF distribution of each theoretical model. All these simulations were performed in MATLAB (MathWorks).

Model Construction. To perform this simulation, we first created a G_{vi} grid (with rows corresponding to voxels, v, and columns to items, i) of random values drawn for each voxel from a multivariate normal distribution $x \sim \mathrm{N}(\mu, \Sigma)$, where μ was drawn from a standard uniform distribution across the open interval {0 1} for each item and the off-diagonal elements of $i \times i$ covariance matrix Σ were set to a free parameter c. G was divided into 12 think (T) and 12 no-think (NT) items such that $\mathrm{G}_{vi} = [\mathrm{T}_{vi}, \mathrm{NT}_{vi}]$. Note here that we used 12 items in each condition instead of 24 because of the cross-validation procedure, which assigned half of the items in the $\mathrm{RDM}_{\text{-fus}}$ to a training set and the other half to a test set. The parameter c determines the mean correlation across all patterns (i.e., items). Each column in this initial grid represents the initial pattern of activity triggered by a memory cue paired with a stored object. From this initial pattern, activity was then modulated differently for think and no-think items.

Memory suppression type: Inhibition versus truncated activation versus retrieval alone. For both think and no-think trials, a proportion (x and y, respectively; see voxel selection type) of voxels was first selected. Think trials were enhanced by an enhancement factor (e), such that $\mathrm{T}_{xi} = \mathrm{T}_{xi} \cdot e$ (e.g., doubled when $e = 2$).

- Inhibition: no-think selected voxels were down-scaled by a suppression factor (s) such that $\mathrm{NT}_{yi} = \mathrm{NT}_{yi} \cdot s$ (e.g., halved when $s = 0.5$).
- Truncated activation: the number of selected voxels whose reactivation was truncated in no-think trials corresponded to a ratio, r, of the number of selected voxels in the think condition, i.e., a proportion $x \cdot r$ of all voxels (e.g., 25% of voxels when $r = 0.5$ and $x = 0.5$). Activity of this subset of voxels during no-think trials was up-scaled by the same e as for think trials.
- Retrieval alone: the initial grid of activity was not modulated for no-think trials.

Voxel selection type: Targeted versus random versus fixed voxel selection.

- Targeted: for both think and no-think trials, a proportion (x and y, respectively) of voxels that were most highly activated were selected (e.g., the top 30% when $x = 0.3$). This selective mechanism was applied separately for each item.
- Random: for both think and no-think trials, a proportion (x and y, respectively) of voxels were randomly selected. This random selection was applied separately for each item.
- Fixed: the same proportions of voxels (x and y) were selected as in other models, in a random fashion (regardless of activity level), but this selection was fixed across items within a condition. Note that under this account, an additional overlapping factor (o) was introduced to control for the degree of overlap between voxels selected in the think and no-think conditions,

such that $o = 0.5$ means that half the voxels selected in the no-think condition were the same voxels as selected for the think condition.

Finally, for all models, once activity was modulated for think and no-think items, noise randomly drawn from a standard uniform distribution on the open interval $\{0\ 1^*n\}$, with $n < 1$, was added to each voxel and pattern such that $G_{vi} = G_{vi} + R_{vi}$.

The goal of the next MCMC step was then to sample the entire parameter space and to identify which parameter values were most likely to fit to the observed $\mathrm{RDM}_{\text{-fus}}$.

MCMC Algorithm. Our goal was to sample from the unknown target (i.e., posterior) distribution $p(\theta_j)$ of each of the $j = 1...N$ parameters presented above. Here we used a Metropolis sampler, which creates a Markov chain that produces a sequence of values

$$\theta_j^{(1)} \rightarrow \theta_j^{(2)} \rightarrow \theta_j^{(3)} \rightarrow \ldots \rightarrow \theta_j^{(t)},$$

where $\theta_j^{(t)}$ represents the state of a Markov chain at iteration t. In the Metropolis procedure, we initialize the first state, $\theta_j^{(1)}$ to some initial random value. For each parameter, we then used a standard uniform (see below) proposal distribution $q(\theta_j)$ to generate new candidate θ_j^*. The use of a uniform distribution is convenient as it makes no assumption about the shape of the target distribution, and it satisfies a key requirement of the Metropolis sampler, which is to have a symmetrical proposal distribution such that $q(\theta_j^*|\ \theta_j^{(t-1)}) = q(\theta_j^{(t-1)}|\ \theta_j^*)$. The next step is then to either accept or reject the new proposal θ_j^*, with the probability of accepting the new proposal being

$$\alpha = \min\left(1, \frac{\mathrm{p}\left(\theta_1^*, \theta_2^{t-1}, \ldots, \theta_N^{t-1}\right)}{\mathrm{p}\left(\theta_1^{t-1}, \theta_2^{t-1}, \ldots, \theta_N^{t-1}\right)}\right),$$

To compute this acceptance probability, we calculated for a given model M (see *General Overview*), $\mathrm{RDM}_{\text{-t}}$ with the new proposal, such that new $\mathrm{RDM}_{\text{-t}} = \mathrm{M}\ (\theta_1^*, \theta_2^{(t-1)}, ..., \theta_N^{(t-1)})$, as well as $\mathrm{RDM}_{\text{-t}}$ at the state of the chain $t - 1$, such that old $\mathrm{RDM}_{\text{-t}} = \mathrm{M}\ (\theta_1^{(t-1)}, \theta_2^{(t-1)}, ..., \theta_N^{(t-1)})$. Then we computed the cost of both the new proposal and old state such that new cost $= 1 - r$(new $\mathrm{RDM}_{\text{-t}}$, $\mathrm{RDM}_{\text{-fus}}$) and old cost $= 1 - r$ (old $\mathrm{RDM}_{\text{-t}}$, $\mathrm{RDM}_{\text{-fus}}$), with r being the Spearman rank correlation between the two vectorized RDMs. The probability of accepting the new proposal becomes then:

$$\alpha = \min(1, \exp(-(\text{new cost}/\text{old cost}))).$$

Hence, when a new cost value decreases relative to the old cost after a new proposed parameter (i.e., better fit), α increases toward 1 [i.e., new parameter θ_j^* is more likely than the old one $\theta_j^{(t-1)}$]. To make a decision on whether to accept or reject the new proposal, we draw a value, u, from a uniform standard distribution on the open interval $\{0\ 1\}$. If $u < \alpha$ or if the new cost value decreases relative to old cost, we accept the proposal θ_j^* and the next state is then set to $\theta_j^{(t)} = \theta_j^*$. If $u > \alpha$, we reject the new proposal and the next state is set to be equal to the old state, $\theta_j^{(t)} = \theta_j^{(t-1)}$.

At each iteration t, we generate independently a new proposal for each parameter entering our model M and either accept or reject the proposal. Here is a summary of the steps of the Metropolis sampler:

i) Set $t = 1$.
ii) Generate an initial value drawn from a uniform proposal distribution (see below) for each parameter $\theta_1, \theta_2, \theta_3, ..., \theta_N$.
iii) Generate a proposal θ_1^*, from $q(\theta_1)$ which is the uniform proposal distribution of θ_1, with $\theta_{1min} < \theta_1 < \theta_{1\ max}$.
iv) Evaluate the acceptance probability $\alpha = \min(1, \mathrm{p}(\theta_1^*, \theta_2^{t-1}, \ldots, \theta_N^{t-1})/\mathrm{p}(\theta_1^{t-1}, \theta_2^{t-1}, \ldots, \theta_N^{t-1}))$, with $\alpha = \min(1, \exp(-(\text{new cost}/\text{old cost})))$.
v) Generate u from a uniform $\{0\ 1\}$. If $u < \alpha$ or new cost < old cost, accept the proposal and set $\theta_1^{(t)} = \theta_1^*$; else set $\theta_1^{(t)} = \theta_1^{(t-1)}$. Apply the same process for $\theta_2, \ldots, \theta_N$.
vi) Repeat until $t = T$.

When t reaches the number of iterations specified (here $T = 5{,}000$), we then have an approximation of the posterior distribution of each parameter θ. Because this Metropolis algorithm always accepts a new proposal when it is more likely under the posterior distribution than the old state, the sampler will move toward the regions of the state space where the posterior distribution has high density (in other words, toward parameter values which are more likely to explain the data, i.e., $\mathrm{RDM}_{\text{-fus}}$; Fig. S5). However, even if the new proposal provides a worse fit to the data than the current state, it might still be accepted because $u < \alpha$ could arise by chance (if the drawn value is very low). This process of always accepting a new parameter value that improves model fit but occasionally accepting other values to ensure that the sampler explores the whole state space, i.e., samples all parts of the posterior distribution (including the tails).

However, this parameter space is limited by the open interval chosen for the uniform proposal distribution, so it is important that these proposal distributions cover the entire space of possible values, bound by some limits. Here we used the following uniform discrete distributions for the parameters described in the above model construction section:

- Average correlation across all patterns, $c = \mathrm{U}(0.1,0.9)$, step $= 0.05$.
- Number of voxels composing the grid G, $v = \mathrm{U}(20,1000)$, step $= 20$.
- Proportion of noise added to the data, $n = \mathrm{U}(0.05, 0.9)$, step $= 0.05$.
- Suppression factor, $s = \mathrm{U}(0.05, 0.9)$, step $= 0.05$ (inhibition accounts only).
- Retrieval factor, $e = \mathrm{U}(1/\max(s), 1/\min(s))$, i.e., $e = \mathrm{U}(1.1, 20)$, step $= 0.05$.
- Proportion of modulated voxels for think items, $x = \mathrm{U}(0.05, 0.9)$, step $= 0.05$.
- Proportion of modulated voxels for no-think items, $y = \mathrm{U}(0.05, 0.9)$, step $= 0.05$.
- Ratio of modulated voxels for no-think items compared with think items, $r = \mathrm{U}(0.05, 0.95)$, step $= 0.05$ (truncated activation accounts only).
- Proportion of overlapping voxels between think and no-think condition, $o = \mathrm{U}(0, 1)$, step $= 0.05$ (fixed voxel selection accounts only).

Note that for the retrieval alone model, s and y were not relevant and hence not sampled by the Metropolis algorithm.

Random-Effect Analysis and Cross-Validation. The MCMC algorithm presented above allows us to sample from the posterior distribution of each parameter and to identify the regions of the state space where the posterior distribution has high density for the RDM of a given participant. Once the initial samples of the MCMC algorithm have been discarded (the burn-in period was set to 250 samples; see *MCMC Convergence*), the mode of this posterior distribution hence reflects a reasonable estimate of the most likely parameter values under the posterior distribution (MAP estimation), i.e., providing the best fit to the data. However, with so many parameters to each model, and relatively few data, there is a danger that the models will overfit the data (i.e., fit the noise in the data, rather than the true signal). To evaluate this, we used cross-validation to select the model that best generalizes from one half of the data (training set) to the other half of the data (test set). We randomly split the RDM$_{\text{-fus}}$ of each participant into two independent halves 100 times, each time fitting the training half using the above MCMC algorithm, and using the posterior mode of each parameter to estimate the GOF for the test half. GOF was defined as r(RDM$_{\text{-t}}$, RDM$_{\text{-fus}}$), where r is the Spearman correlation between the two vectorized RDMs. These 100 GOF values for each participant and the models were then averaged, resulting in a 24-participant × 9-model data set. Statistical differences between models were then tested with a bootstrap with replacement approach on the mean difference between pairs or families of models (using 2,000 bootstraps), allowing us to compute the confidence intervals for the differences between models (corresponding to bias-corrected and accelerated percentile method).

Fig. 5C in the main text reports the mean GOF across participants. The cross-validation approach used for model fitting and testing is illustrated in Fig. S4.

MCMC Convergence. The first 250 samples of the MCMC chains were discarded and not collected. Different diagnostic tests were performed to check whether the chains have converged to their stationary distributions. Those tests were performed on each sampled parameter for each model and each participant, discarding the first 250 initial samples. One way to assess convergence is to compute the autocorrelations between the draws of the Markov chain. The lag k autocorrelation ρ_k is the correlation between every draw i of the chain x and its k_{th} lag:

$$\rho_k = \frac{\sum_{i=1}^{n-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

Fig. S6 illustrates how k_{th} lag autocorrelation is smaller as k increases for a given participant and random split, indicating that the chains have mixed quickly to their stationary distribution. This pattern was true across all participants and random splits.

Another assessment of stationary distribution is the Gelman–Rubin diagnostic which can be performed by running the same Markov chain multiple times (as was done for the cross-validation approach above) and to estimate the variance of the parameter as a weighted sum of the within-chain and between-chain variance. The within variance (W) is the mean of the variance of m chains, such that

$$W = \frac{1}{m}\sum_{j=1}^{m} s_j^2,$$

where s_j^2 is the variance of the j_{th} chain x with n samples, such that

$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_{ij} - \bar{x}_j\right)^2.$$

The between variance (B) is given by

$$B = \frac{n}{m-1}\sum_{j=1}^{m}\left(\bar{x}_j - \bar{\bar{x}}\right)^2, \text{ where } \bar{\bar{x}} = \frac{1}{m}\sum_{j=1}^{m}\bar{x}_j.$$

We can then estimate the variance of the stationary distribution as a weighted average of W and B:

$$\widehat{Var}(x) = \left(1 - \frac{1}{n}\right)W + \frac{1}{n}B.$$

The estimated potential scale reduction factor (EPSR) corresponds then to

$$\hat{R} = \sqrt{\frac{\widehat{Var}(x)}{W}}.$$

EPSR measures the degree to which the posterior variance would decrease if we were to continue sampling to infinity. If EPSR ≈ 1, then that estimate is reliable, meaning the variance between the chains is similar to the variance within each chain, and that the chains have converged to the stationary distribution.

Here EPSR < 1.06 for all parameters of each model tested for each participant and random split, indicating that the MCMC algorithm converged well.

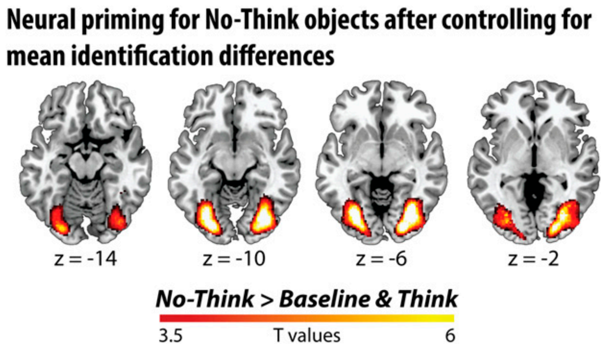

Neural priming for No-Think objects after controlling for mean identification differences
z = -14
z = -10
z = -6
z = -2
No-Think > Baseline & Think
3.5
T values
6

Fig. S1. Memory inhibition effect during the final priming test phase after controlling for mean identification time differences across conditions. Note that, contrary to Fig. 3B, we did not mask this effect with the main effect of neural priming.

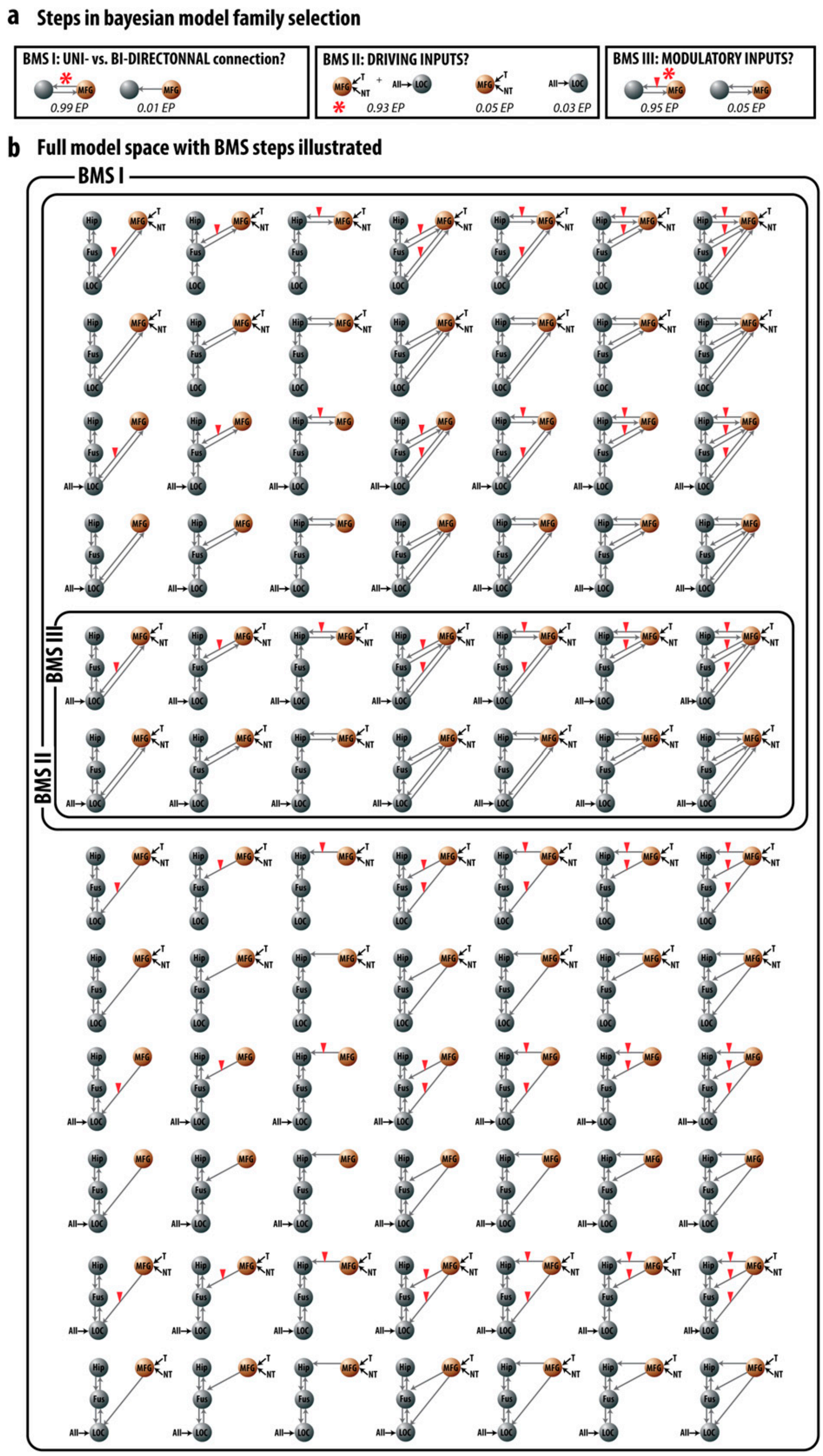

a Steps in bayesian model family selection
BMS I: UNI- vs. BI-DIRECTONNAL connection?
0.99 EP
0.01 EP
BMS II: DRIVING INPUTS?
0.93 EP
0.05 EP
0.03 EP
BMS III: MODULATORY INPUTS?
0.95 EP
0.05 EP
b Full model space with BMS steps illustrated
BMS I
BMS II
BMS III
Hip
Fus
LOC
MFG
T
NT
All

Fig. S2. Dynamic causal modeling (DCM) model space and Bayesian model selection (BMS) procedures. (*A*) BMS was first applied on the direction (bilateral versus unilateral intrinsic connections) family. The bilateral subgroup won (as indicated by red asterisk) against the unilateral subgroup with an exceedance probability of 0.99, and an expected posterior probability of 0.77. Within the bilateral family of models, we then compared which driving input was more likely. Models including a driving input in both the lateral occipital complex (LOC) and the middle frontal gyrus (MFG) won with an exceedance probability of 0.926 (against 0.0532 for the MFG only and 0.0208 for the LOC only), and an expected posterior probability of 0.5538 (against 0.2476 for the MFG only and 0.1986 for the LOC only). Finally, we compared the remaining seven modulatory models (i.e., including a top-down modulation of the coupling between the

Legend continued on following page

MFG and posterior regions during no-think trials) to the seven null models that did not include any modulatory input on top-down connections. This analysis overwhelmingly favored models with modulation over models without modulation (exceedance probability = 0.95, expected posterior probability = 0.73). (*B*) The full space of DCM models. Gray arrows represent connections between nodes. Black arrows correspond to the driving inputs of the models. The red arrowheads illustrate the presence of a modulatory input for no-think items on connection strength.

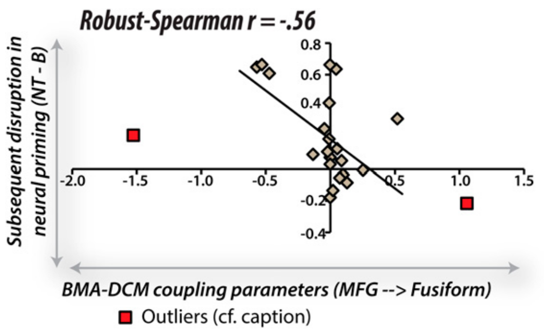
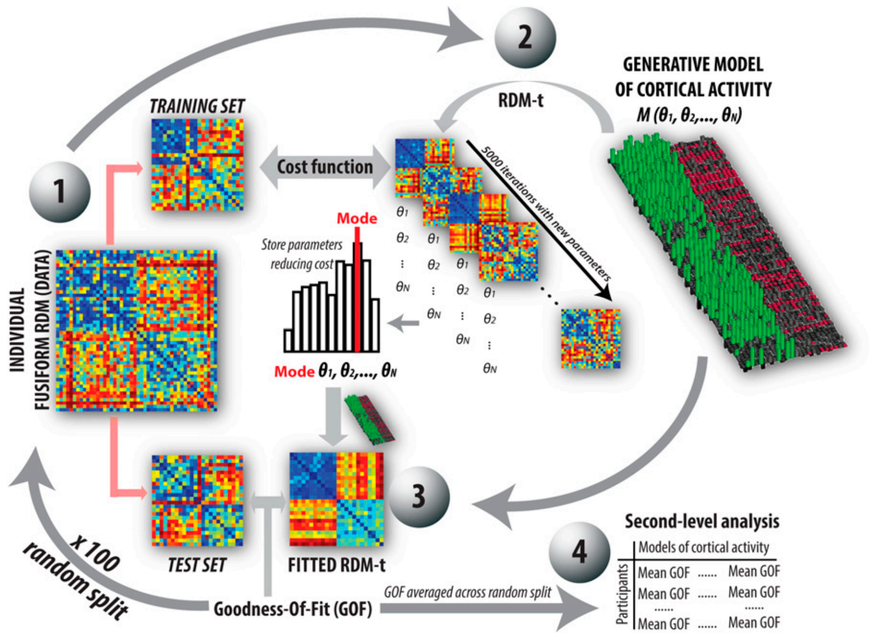


Fig. S3. Relationship between modulatory parameters for the MFG→fusiform DCM model and disrupted neural priming for no-think items (i.e., no-think − baseline). Robust Spearman correlation (1) revealed the presence of two bivariate outliers consistently identified using box-plot rule, the median absolute deviation to the median, and the median of absolute distances (S outlier; cf. ref. 1).

1. Pernet CR, Wilcox R, Rousselet GA (2013) Robust correlation analyses: False positive and power validation using a new open source Matlab toolbox. *Front Psychol* 3:a606.



TRAINING SET
Cost function
RDM-t
GENERATIVE MODEL OF CORTICAL ACTIVITY
M (θ1, θ2,..., θN)
5000 iterations with new parameters
Mode
Store parameters reducing cost
Mode θ1, θ2,..., θN
INDIVIDUAL FUSIFORM RDM (DATA)
TEST SET
FITTED RDM-t
x 100 random split
Goodness-Of-Fit (GOF)
GOF averaged across random split
Second-level analysis
Models of cortical activity
Participants
Mean GOF ...... Mean GOF

Fig. S4. A visual illustration depicting the model fitting and testing procedures based on MCMC: a cross-validation approach. We randomly split the RDM$_{\text{-fus}}$ of each participant into two independent halves 100 times, each time fitting the training half using the MCMC algorithm. The MCMC algorithm allows us to sample from the posterior distribution of each parameter and to identify the regions of the state space where the posterior distribution has high density. Once the initial samples of the MCMC algorithm have been discarded (the burn-in period was set to 250 samples, see *MCMC Convergence*), the mode of this posterior distribution hence reflects a reasonable estimate of the parameter values providing the best fit to the data. We used the posterior mode of each parameter to test how model fitting generalizes to the other half of the data (test set) and to estimate the GOF (i.e., Spearman correlation between the two vectorized RDM$_{\text{-fus}}$ and RDM$_{\text{-t}}$). The 100 GOF values computed were then averaged for each model and each participant, and entered into a second-level analysis treating participants as a random effect variable.

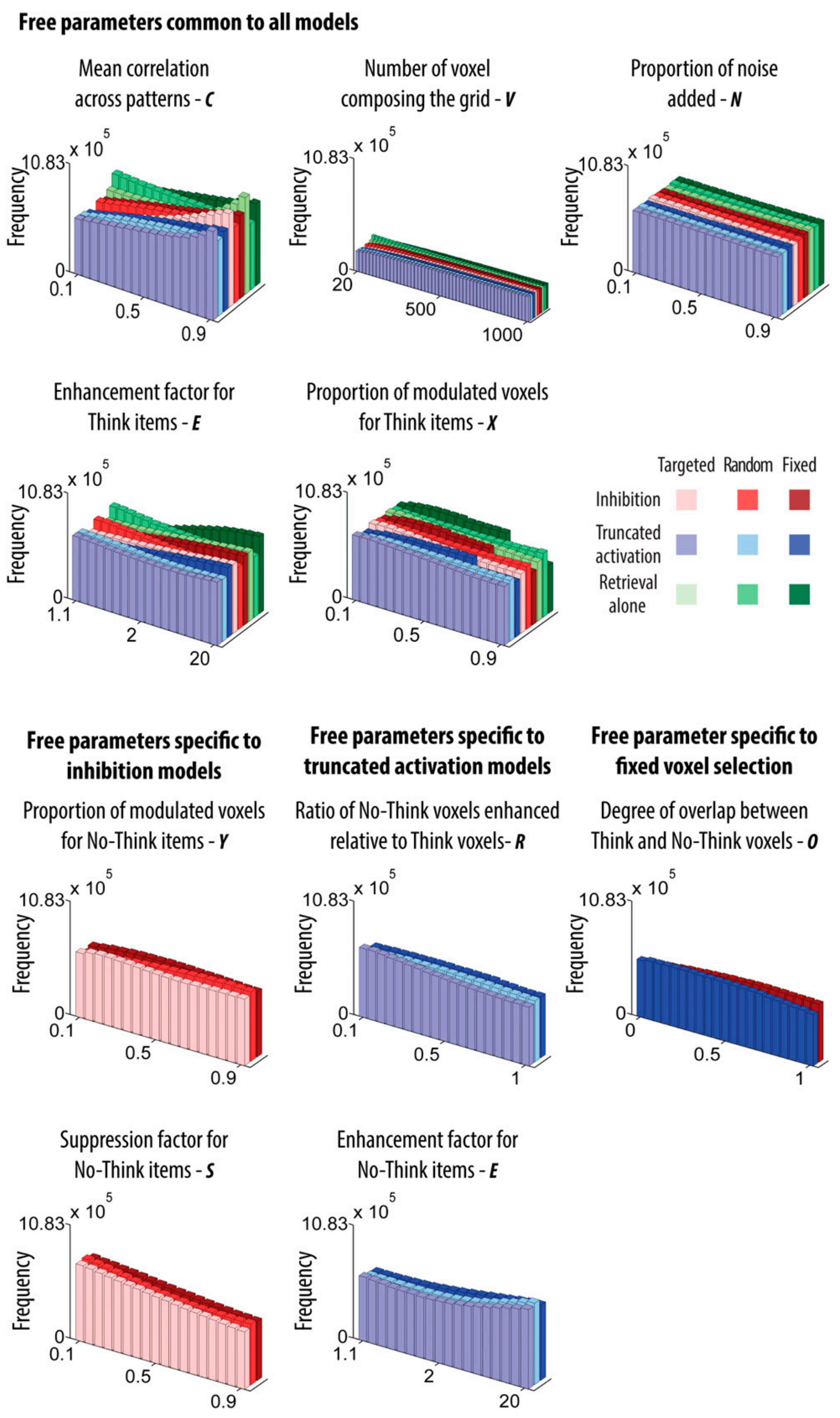


Fig. S5. Histogram of the sample distribution for each model free parameter obtained after MCMC convergence. The MCMC Metropolis algorithm ensures that the whole state space of parameters is sampled, and the sampler will move toward the regions of the state space that provide a better fit to the data. Hence, more frequent values represent a critical feature that is necessary to explain the data under a given model, while distributions that stay largely uniform indicate that these parameters do not have much impact on model fit. For instance, the distribution of the suppression factor (*S*) is skewed toward lower values, indicating that suppression has an impact on model fit. In contrast, the distribution of the number of voxels remains flat showing that our findings generalize well across a range of voxel number values. These histograms were plotted after 100 repetitions of the MCMC Metropolis sampler comprising 4,750 iterations (i.e., discarding the first 250 burn-in samples) across 24 participants.

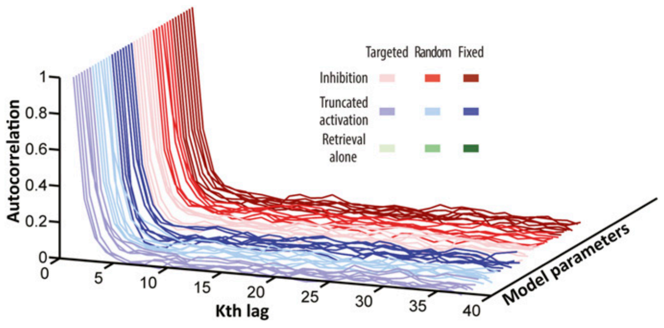


Fig. S6. Autocorrelation between the draws of the Markov chains showing that the MCMC algorithm converged well. Autocorrelation is a cross-correlation of the sample time series with itself as a function of a time separation (i.e., k_{th} lag). A decrease in autocorrelation when *k* lag increases indicates a fast mixing of the chain and a convergence to a stationary distribution. Here, these autocorrelations were plotted for one participant and for each model parameter after a single random split of the MCMC Metropolis sampler comprising 4,750 iterations (i.e., discarding the first 250 burn-in samples), to illustrate that the MCMC algorithm converged well. Note that the Gelman–Rubin diagnostic test also indicated a convergence of the Markov chains to stationary distribution (*SI Simulation Methods*).

Table S1. Peak coordinates of the regions showing a think versus no-think difference at $P_{FWE} < 0.05$ (whole brain)

		MNI coordinates, mm				
Anatomical description	No. of voxels	*x*	*y*	*z*	*T*	P_{FWE}
No-think > think						
Right SFG	377	20	16	58	8.96	<0.001
Right IPC	705	44	−46	36	8.5	<0.001
Right MFG	331	44	24	46	8.37	<0.001
Right IFG	254	50	20	8	8.26	<0.01
Left IPC	68	−60	−52	38	6.8	<0.01
Right inferior orbitofrontal gyrus	25	42	46	−8	6.58	<0.05
Right SFG (anterior)	67	22	52	18	6.29	<0.05
Right medial SFG	45	10	36	42	6.23	<0.05
Left LOC	7	−46	−80	−4	6.22	<0.05
Left inferior temporal gyrus	7	−58	−28	−20	6.24	<0.05
Left MFG	5	−40	28	44	6.01	<0.05
Right superior parietal gyrus	7	34	−52	58	5.98	<0.05
Think > no-think						
Left fusiform gyrus	52	−32	−32	−24	7.09	<0.01
Left IFG	49	−42	32	14	7.09	<0.01

The think > no-think difference observed in the hippocampus survived correction when the search volume was restricted to the left [$t(23) = 4.01$, $P_{FWE} < 0.05$, $x = -32$, $y = -26$, $z = -14$] and to the right [$t(23) = 3.65$, $P_{FWE} < 0.05$, $x = 34$, $y = -8$, $z = -26$] hippocampus. IFG, inferior frontal gyrus; IPC, inferior parietal cortex; MNI, Montreal Neurological Institute; P_{FWE}, *P* family-wise error; SFG, superior frontal gyrus.

Table S2. Peak coordinates of the regions showing neural priming (think + no-think + baseline < unprimed) and memory inhibition (no-think > think + baseline) during the final priming test phase at $P_{FWE} < 0.05$

		MNI coordinates, mm				
Anatomical description	No. of voxels	*x*	*y*	*z*	*T*	P_{FWE}
Think + no-think + baseline < unprimed (whole-brain correction)						
Right inferior temporal and fusiform gyri	442	46	−54	12	7.3	<0.001
Left LOC	679	−44	−68	−6	7.24	<0.001
Left fusiform gyrus		−40	−54	−10	6.13	<0.01
Left fusiform gyrus		−34	−44	−18	5.74	<0.01
Right inferior temporal gyrus	183	48	8	26	6.36	<0.01
No-think > think + baseline (main effect of neural priming as restricted search volume)						
Left LOC	243	−38	−76	−8	5.28	<0.001
Left fusiform gyrus		−36	−60	−6	4.96	<0.01
Right fusiform gyrus	69	40	−58	−10	4.65	<0.01
Left fusiform gyrus	3	−34	−52	−16	3.42	0.059

An additional whole-brain correction showed a memory inhibition effect (no-think > think + baseline) in the right LOC [$t(23) = 10.88$, $P_{FWE} < 0.05$, $x = 26$, $y = -84$, $z = -6$], although this region did not show an initial neural priming effect (at least when *P* values were whole-brain corrected). Fig. S1 also reports whole-brain memory inhibition effect (no-think > think + baseline) during the final priming test phase after controlling for mean identification differences across conditions. Regions in italics correspond to submaxima peak coordinates in the cluster.