New frontiers for hierarchical Bayesian models in inverse problems

Daniela Calvetti

Case Western Reserve University

Department of Mathematics, Applied Mathematics and Statistics

ICERM, March 6, 2026

Reinterpreting inverse problems: Linear problems

- Consider a linear inverse problem

$$\text{data} = b = \mathrm{A}x + \text{noise}.$$

- Different point of view:

$$\mathrm{A} = \left[\begin{array}{cccc} a^{(1)} & a^{(2)} & \cdots & a^{(n)} \end{array} \right]$$

  Solving the inverse problems is tantamonut to finding a representation for the data in the form

$$b = x_1 a^{(1)} + x_2 a^{(2)} + \cdots + x_n a^{(n)},$$

  i.e., **decomposing the data into atoms** defined as columns of A.

- $$\mathrm{A} = \text{dictionary of atoms } a^{(j)},\ 1 \leq j \leq n.$$

Reinterpreting inverse problems: Non-linear problems

- Non-linear inverse problem,

$$b = f(\theta) + \text{noise}$$

- Sample the parameter space,

$$\theta_1, \theta_2, \ldots, \theta_n.$$

- Generate (or measure, if possible) the responses

$$d_1 = f(\theta_1),\ d_2 = f(\theta_2),\ \ldots, d_n = f(\theta_n).$$

- Matching the data:

$$b \approx x_1 d_1 + x_2 d_2 + \cdots + x_n d_n, \quad x \text{ sparse}.$$

- Ideally, $x_j \approx x_k \delta_{jk}$, or

$$b \approx x_k d_k, \quad \text{hence, we conclude } \theta \approx \theta_k.$$

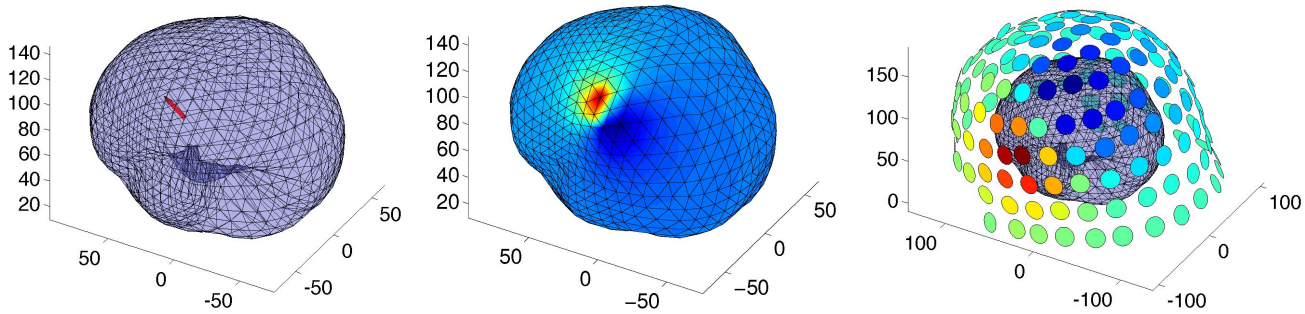
- More generally, interpolate or interpret x_js ar probability weights.

What are the atoms of a dictionary?

The atoms of a dictionary **depend on the application**.

- In **hyperspectral imaging** the atoms are measured annotated spectra
- In **MRI fingerprinting** the atoms are calculated model-based responses
- In **biomechanics** the atoms are MCMC generated time series of muscle activations
- In **LIGO/VIRGO** the atoms are simulated gravitational waves from different cosmic scenarios
- In **PDE parameter estimation**, the atoms are PDE-based simulated measurements with different parameters
- In **MEG** the atoms are computed magnetic fields at sensor induced by different source configurations
- In **SINDy** the atoms are candidate functions defining dynamics
- In **dynamic inverse problems** atoms are prior-based solution templates
- In **replicating portfolio design**, the atoms are return histories of assets

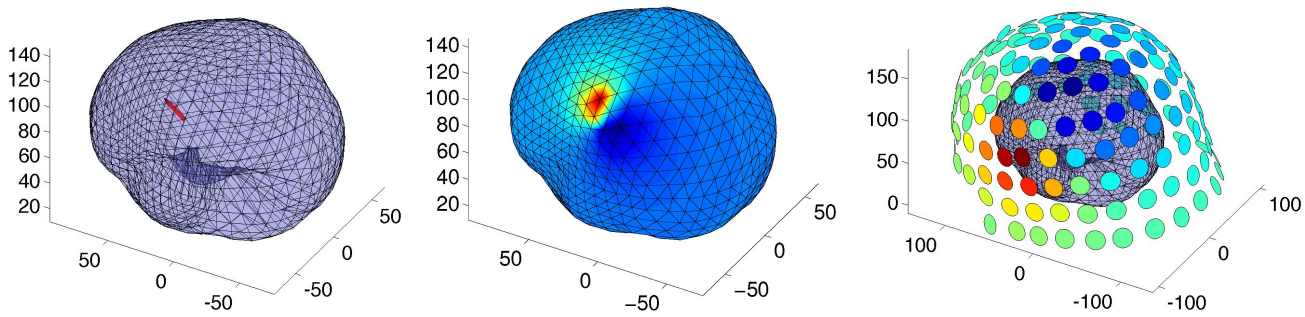
Applications



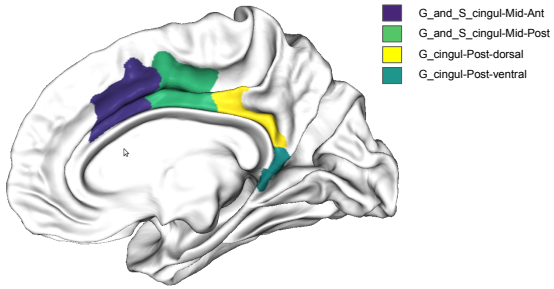
- **Brain activity imaging:** From MEG/EEG data, identify active brain regions.

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \underbrace{\begin{bmatrix} d^{(1)} & d^{(2)} & \cdots & d^{(n)} \end{bmatrix}}_{\sim 15\,000 \text{ dipole responses}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad m \ll n.$$

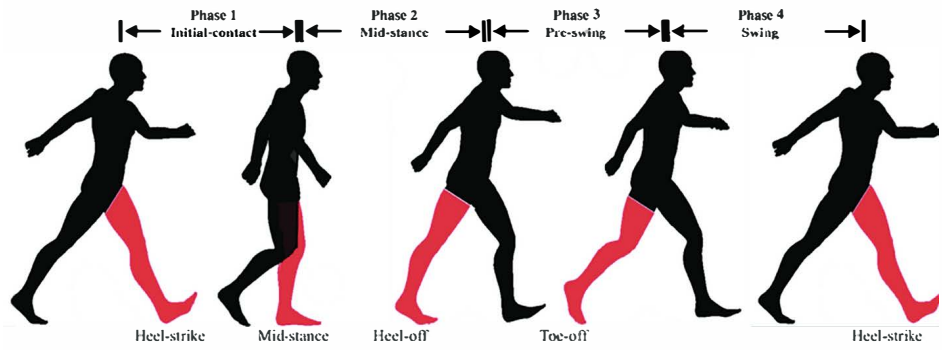
Applications

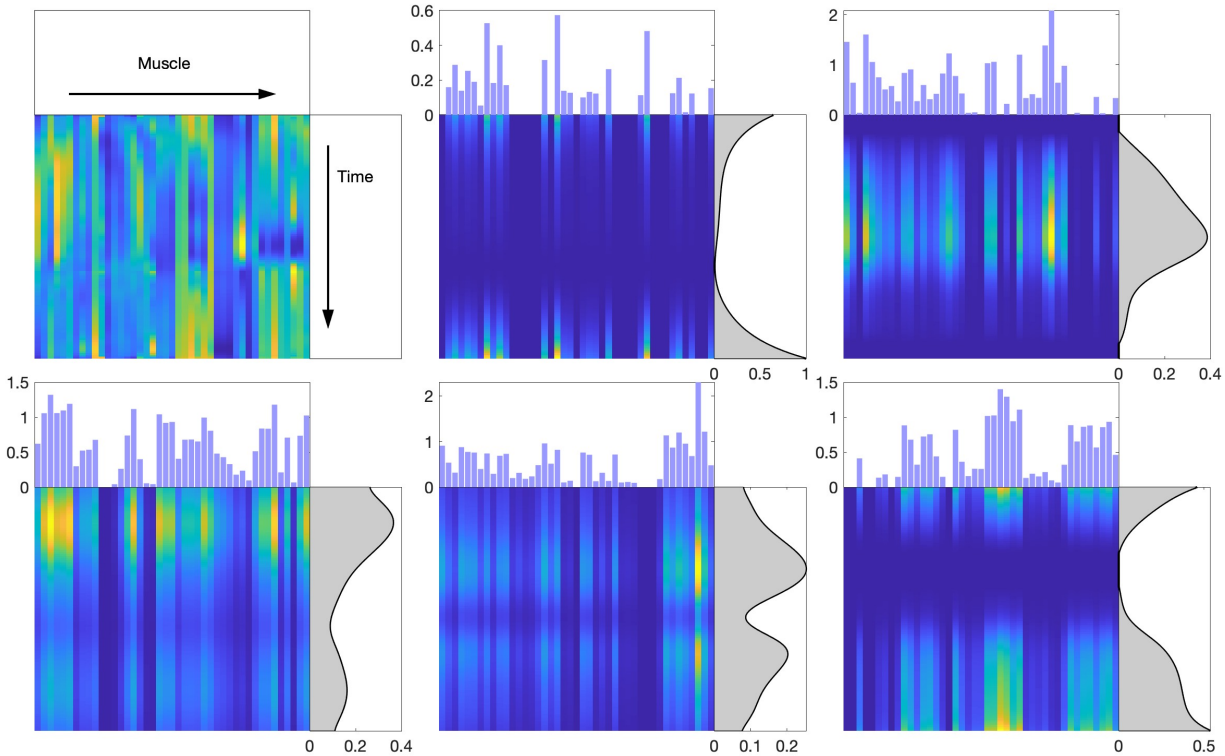


- **Brain activity imaging:** From MEG/EEG data, identify active brain regions.

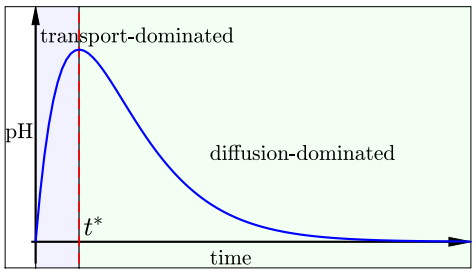


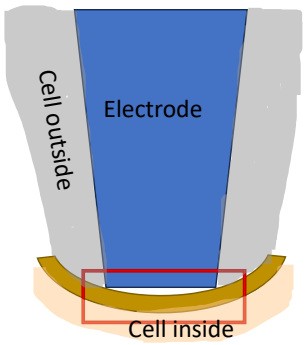
Muscle recruitment inverse problem and UQ

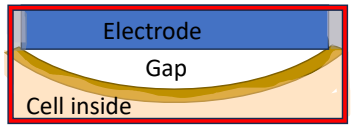


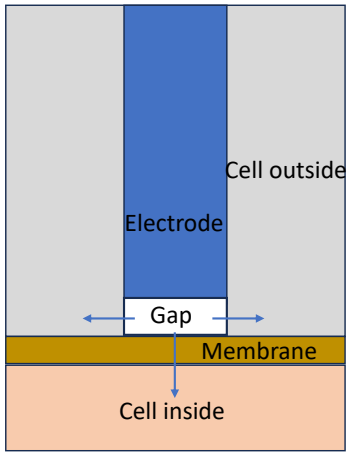


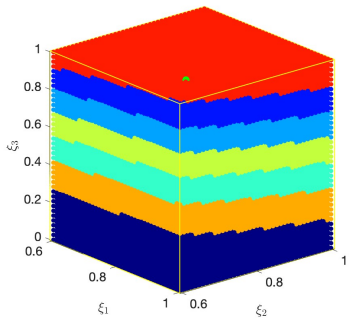
Parameter estimation in PDEs

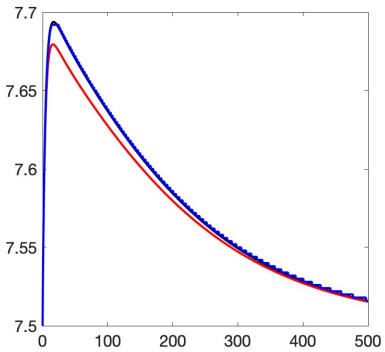


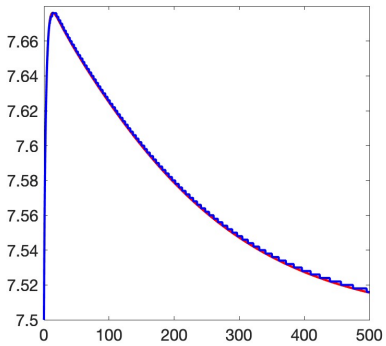


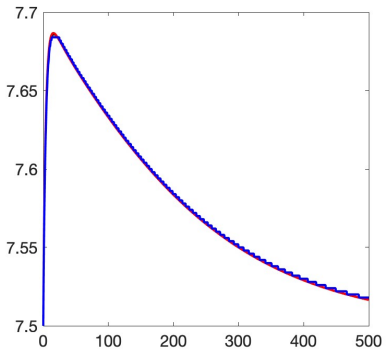




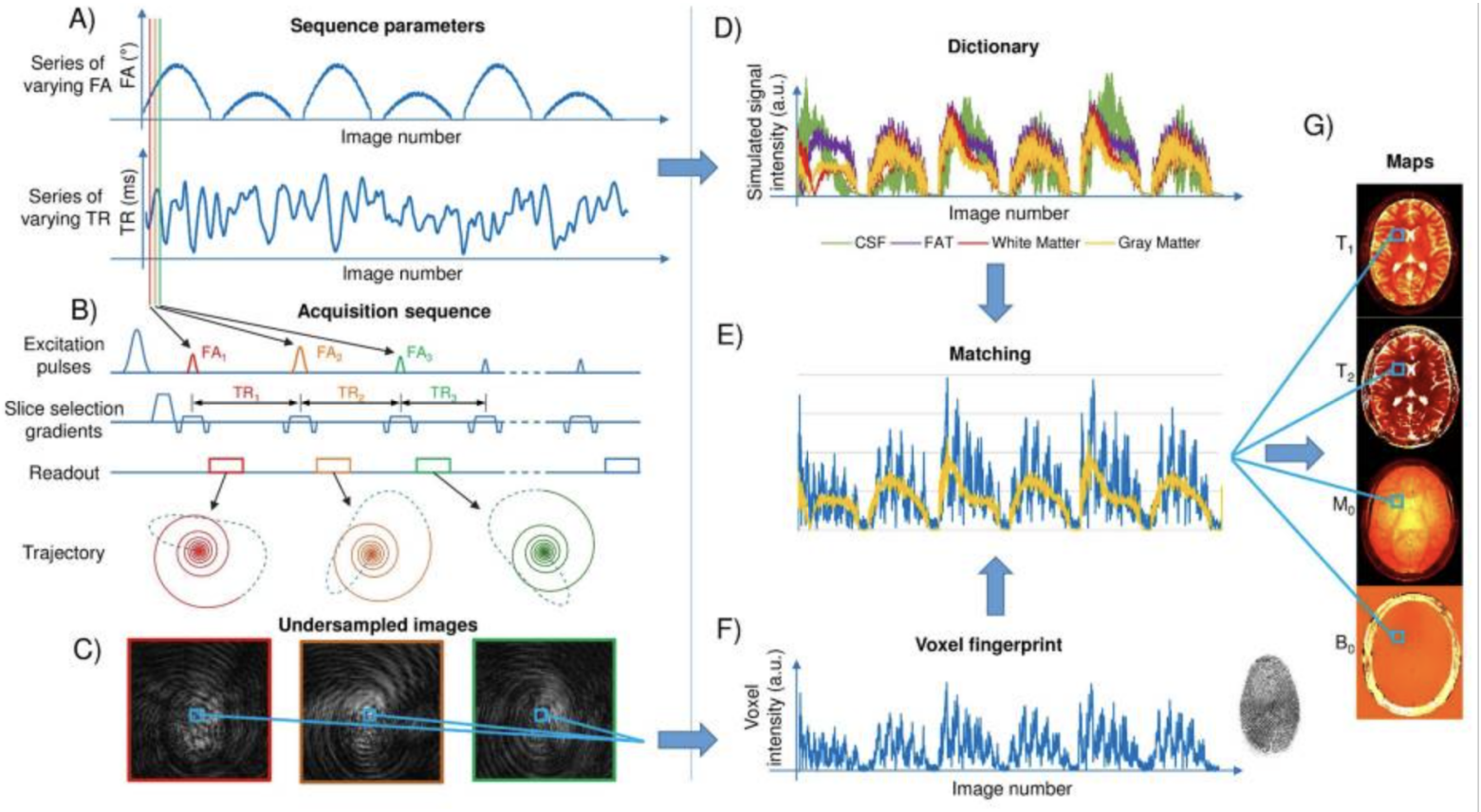








MRI Fingerprinting



Curr Opin Biomed Eng. 2017 Sep;3:56–66. doi: 10.1016/j.cobme.2017.11.001

Dictionaries: between data science and inverse problems

In many applications we have data, we can generate synthetic data, and we want to solve inverse problems for which

- The forward model f is known but too complex for traditional methods;
- There is no formal forward problems but can generate data corresponding to given input;
- The inverse problem is an intermediate step to a classification or design task.

In the dictionary framework, the solution is given in terms of the dictionary atoms (templates).

Often, the fewer items are needed, the better.

In this data-centric approach to inverse problems, it is natural to employ tools from data science.

Dictionary-based inverse problem

Problem: Estimate
a time series of unknown unobservable vectors $X = [x^{(1)}, \ldots, x^{(N_T)}]$ based on a time series of indirect observations $B = [b^{(1)}, \ldots, b^{(N_T)}]$, where

$$b^{(t)} = F^{(t)} x^{(t)} + \epsilon^{(t)}$$

.
Letting $F = \mathrm{diag}\{\mathrm{F}^{(1)}, \ldots, \mathrm{F}^{(\mathrm{N_T})}\}$, $b = \mathrm{vec}(\mathrm{B})$, $\mathrm{x} = \mathrm{vec}(\mathrm{X})$, we can write

$$b = Fx + \epsilon.$$

Stochastic extension: if noise is white Gaussian we have the **likelihood**

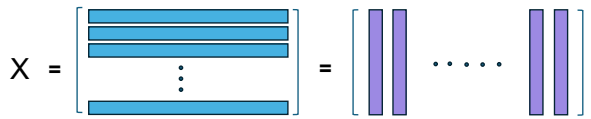
$$\pi_{B|X}(b \mid x) \propto \exp\Big(-\frac{1}{2}\|b - Fx\|_2^2 \Big).$$

Assuming that x admits sparse representations in a dictionary W, we have

$$\pi_{B|Z}(b \mid z) \propto \exp\Big(-\frac{1}{2}\|b - FWz\|_2^2 \Big).$$

Integrating spatio-temporal priors in dictionary

- Row i of X: time evolution of location i
- Column j of X: snapshot of unknown (state) at jth time instance



Express $x^{(t)}$ in terms of a spatial dictionary $S \in \mathbb{R}^{N_s \times N_T}$, e.g., Haar wavelets, Gabor filters, etc, i.e. $x^{(t)} = S\omega^{(t)}$, $\quad X = S\Omega$
Express $x_{(j)}$ in terms of a temporal dictionary $E \in \mathbb{R}^{N_s \times N_T}$, i.e., $x_{(j)} = E\xi_j$,

$$\mathsf{X}^{\mathsf{T}} = \mathsf{E}\Xi \qquad \Longrightarrow \qquad \mathsf{X} = \Xi^{\mathsf{T}}\mathsf{E}^{\mathsf{T}}$$

If Z is a matrix such that $\mathsf{S}W = \Xi$ and $\mathsf{Z}\mathsf{E}^{\mathsf{T}} = \Omega$, it follows that

$$\boxed{\mathsf{X} = \mathsf{S}\mathsf{Z}\mathsf{E}^{\mathsf{T}}}.$$

The Kronecker connection

To arrive at standard dictionary formulation notice that if $z = \mathrm{vec}(\mathsf{Z})$

$$\boxed{x = (\mathsf{E} \otimes \mathsf{S})\, z}$$

thus

$$\mathsf{W} = \mathsf{E} \otimes \mathsf{S}.$$

Can multiply a vector z by $\mathsf{W} = \mathsf{E} \otimes \mathsf{S}$ without forming W by switching to from vector to matrix representation

$$z \overset{\mathrm{vec}^{-1}}{\longrightarrow} \mathsf{Z} \longrightarrow \mathsf{S}\mathsf{Z}\mathsf{E}^{\mathsf{T}} \overset{\mathrm{vec}}{\longrightarrow} \mathsf{W}z$$

and a vector y by $\mathsf{W}^{\mathsf{T}} = \mathsf{E}^{\mathsf{T}} \otimes \mathsf{S}^{\mathsf{T}}$

$$y \overset{\mathrm{vec}^{-1}}{\longrightarrow} \mathsf{Y} \longrightarrow \mathsf{S}^{\mathsf{T}}\mathsf{Y}\mathsf{E} \overset{\mathrm{vec}}{\longrightarrow} \mathsf{W}^{\mathsf{T}}y$$

Dictionary coding: a data-centric inverse problem

Given:

- A dictionary D of p atoms in $\mathbb{R}^n$ (data version of forward model),

$$\mathrm{D} = \left[\begin{array}{cccc} d^{(1)} & d^{(2)} & \dots & d^{(p)} \end{array} \right],$$

- to explain $b \in \mathbb{R}^n$ in terms of the dictionary atoms solve the linear system

$$b = \mathrm{D}x \quad \mathrm{D} \in \mathbb{R}^{n \times p}, \quad x \geq 0.$$

The entry x_j of the solution vector can be regarded as the weight of the contribution of jth atom to b.

In this setting the dictionary D plays the role of the forward model.

Sparsity and nonnegativity

In many application, e.g., for interpretability or classification, a representation of b in term of few dictionary atoms is desired.

We want a sparse (few nonzero entries) solution with nonnegative entries.

Sparse nonnegative code formulation:

Find weights $\widehat{x}$ s.t.

$$\widehat{x} = \arg\min_{x} \|Dx - b\| , \quad \widehat{\mathbf{x}} \geq \mathbf{0}, \quad \|\widehat{\mathbf{x}}\|_{\mathbf{0}} \ll \mathbf{p},$$

where $\|\widehat{x}\|_0$ is the number of nonzero components of $\widehat{x}$.

A four step approach

The entries of the dictionary may have a natural clustering into subdictionaries, or may acquire it via unsupervised clustering.

1. **Partition**: cluster atoms of D into L subdictionaries $\mathrm{D}^{(\ell)}$, $1 \leq \ell \leq L$.
2. **Compress**: compress each subdictionary into low rank code book,

$$\mathrm{D}^{(\ell)} = \mathrm{W}^{(\ell)}\mathrm{H}^{(\ell)} + \mathrm{E}^{(\ell)}.$$

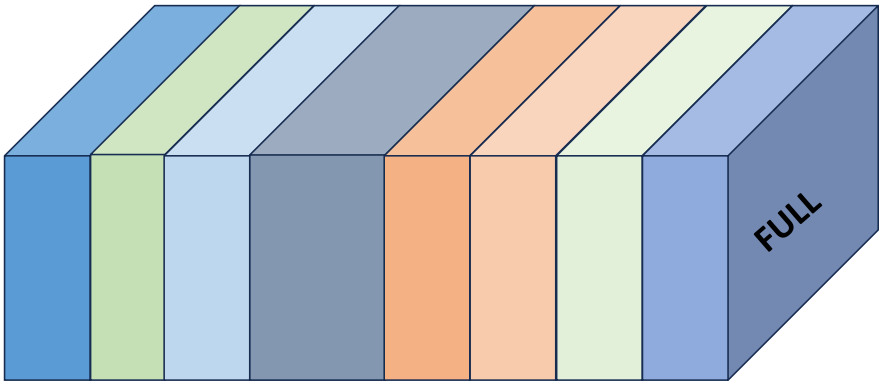
3. **Select**: Identify the few subdictionaries needed to explain b,

$$b = \mathrm{W}^{(j_1)}h^{(j_1)} + \ldots + \mathrm{W}^{(j_r)}h^{(j_r)} + \epsilon'.$$

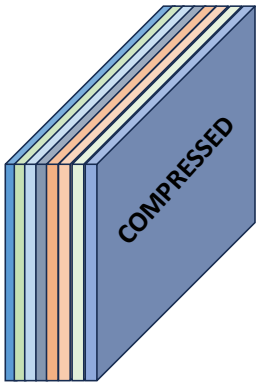
4. **Match** : Explain b in terms of these few subdictionaries,

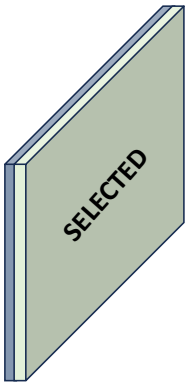
$$b = \mathrm{D}^{(j_1)}x^{(j_1)} + \ldots + \mathrm{D}^{(j_r)}x^{(j_r)} + \tilde{\epsilon}'.$$

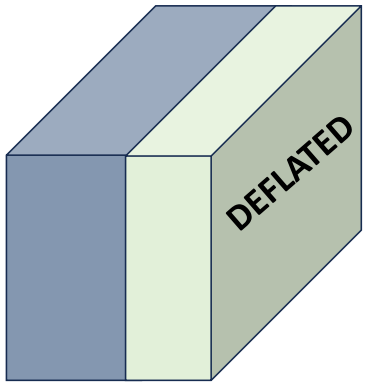
In steps 2-4 use Hierarchical Bayesian Models (HBM)



DICTIONARY COMPRESSION
SELECTION DEFLATION







Step 2: Compress

Compute favorite low-rank approximation of each subdictionary (PCA, NMF, etc),

$$\mathsf{D}^{(\ell)} = \mathsf{W}^{(\ell)}\mathsf{H}^{(\ell)} + \mathsf{E}^{(\ell)}, \quad \mathsf{W}^{(\ell)} \in \mathbb{R}^{n \times k_\ell}, \ \mathsf{H}^{(\ell)} \in \mathbb{R}^{k_\ell \times n_\ell}, \ k_\ell \ll n_\ell.$$

- $\mathsf{W}^{(\ell)} =$ **local code book**, summarizes characteristics of the atoms in $\mathsf{D}^{(\ell)}$
- $\mathsf{E}^{(\ell)} =$ **compression error**
- The rank may differ according to complexity of subdictionary.

Combine:

$$\begin{aligned} \mathsf{D} &= \begin{bmatrix} \mathsf{W}^{(1)}\mathsf{H}^{(1)} & \mathsf{W}^{(2)}\mathsf{H}^{(2)} & \dots & \mathsf{W}^{(L)}\mathsf{H}^{(L)} \end{bmatrix} + \begin{bmatrix} \mathsf{E}^{(1)} & \mathsf{E}^{(2)} & \dots & \mathsf{E}^{(L)} \end{bmatrix} \\ &= \mathsf{W}\mathsf{H} + \mathsf{E}, \\ \mathsf{W} &= \begin{bmatrix} \mathsf{W}^{(1)} & \dots & \mathsf{W}^{(L)} \end{bmatrix}, \quad \mathsf{H} = \mathrm{diag}\{\mathsf{H}^{(1)}, \dots, \mathsf{H}^{(\mathrm{L})}\} \end{aligned}$$

Dictionary compression error (DCE)

The compression introduces a "modeling error" (DCE) $\mathsf{E} = \mathsf{D} - \mathsf{W}\mathsf{H}$.

In the Bayesian framework, the DCE is a random variable. To approximate its distribution we write the ℓth dictionary entry as

$$d^{(\ell)} = \mathsf{W}h^{(\ell)} + \underbrace{(\mathsf{D}_{(:,\ell)} - \mathsf{W}h^{(\ell)})}_{e^{(\ell)}}, \quad 1 \le \ell \le p,$$

where $h^{(\ell)}$ is the solution of the compressed dictionary matching.

Estimation of DCE density

1. For each subdictionary $\mathsf{W} = \mathsf{W}^{(\ell)}$, the computed

$$e = d - \mathsf{W}h, \quad d \in \{d^{(1)}, d^{(2)}, \ldots, d^{(p)}\}$$

is a sample from an underlying probability density π_{DCE}.

2. $$\mu_{\mathrm{DCE}} = \frac{1}{m}\sum e^{(j)}, \quad \mathsf{C}_{\mathrm{DCE}} = \frac{1}{m-1}\sum (e^{(j)} - \mu_{\mathrm{DCE}})(e^{(j)} - \mu_{\mathrm{DCE}})^{\mathsf{T}} + \epsilon \mathsf{I}.$$

3. Use Laplace approximation

$$e \sim \mathcal{N}(\mu_{\mathrm{DCE}}, \mathsf{C}_{\mathrm{DCE}}).$$

Likelihood model:

$$\pi_{b|h}(b \mid h) \propto \exp\left(-\frac{1}{2}(b - \mathsf{W}h - \mu_{\mathrm{DCE}})^{\mathsf{T}} \mathsf{C}_{\mathrm{DCE}}^{-1}(b - \mathsf{W}h - \mu_{\mathrm{DCE}})\right).$$

Step 3: Group sparsity for compressed dictionary matching

Given b and code book $\mathsf{W} = [\mathsf{W}_1 \dots \mathsf{W}_L]$ find nonnegative group-sparse h

$$b \approx \mathsf{W}h = \sum_{\ell=1}^{L} \mathsf{W}^{(\ell)} h^{(\ell)}, \quad h^{(\ell)} \geq 0, \quad h^{(\ell)} \approx 0 \text{ for most } \ell, .$$

Group sparsity via HBM

- The $h^{(\ell)}$ are mutually independent
- Each $h^{(\ell)}$ is zero-mean conditionally Gaussian
- The magnitude of $h^{(\ell)}$ is scaled by the reciprocal of $\theta_\ell > 0$.
- Most θ_ℓ should be small, but some can be large.
- The direction of $h^{(\ell)}$ should be guided by the columns of $\mathsf{H}^{(\ell)}$.

Structural prior model for $h^{(\ell)}$

Define a zero mean (expected to vanish) conditionally Gaussian prior for $h^{(\ell)}$

$$\pi_{h^{(\ell)}|\theta_\ell}(h^{(\ell)} \mid \theta_\ell) \propto \theta_\ell{}^{-r_\ell/2} \exp\left(-\frac{1}{2}\frac{\|h^{(\ell)}\|^2_{(\ell)}}{\theta_\ell}\right), \quad \|h^{(\ell)}\|^2_{(\ell)} = \left(h^{(\ell)}\right)^{\mathsf{T}} \left(\mathsf{G}^{(\ell)}\right)^{-1} h^{(\ell)}.$$

The direction-sensitive prior covariance matrix $\mathsf{G}^{(\ell)}$ guides $h^{(\ell)}$ in the overall direction determined by the columns of $\mathsf{H}^{(\ell)}$.

If $\mathsf{H}^{(\ell)} = \mathsf{Q}^{(\ell)}\Lambda^{(\ell)}(\mathsf{V}^{(\ell)})^{\mathsf{T}}$, then

$$\mathsf{G}^{(\ell)} = \left(\frac{1}{\lambda_1}\right)^2 \mathsf{Q}^{(\ell)}\Lambda^{(\ell)}[\Lambda^{(\ell)}]^{\mathsf{T}}[\mathsf{Q}^{(\ell)}]^{\mathsf{T}} + \epsilon\,\mathsf{I} = q_1 q_1^{\mathsf{T}} + \sum_{j=2}^{r_\ell}\left(\frac{\lambda_j}{\lambda_1}\right)^2 q_j q_j^{\mathsf{T}} + \varepsilon\,\mathsf{I}_{r_\ell}.$$

Generalized gamma distributions are convenient hyperprior choices for the θ_ℓ:

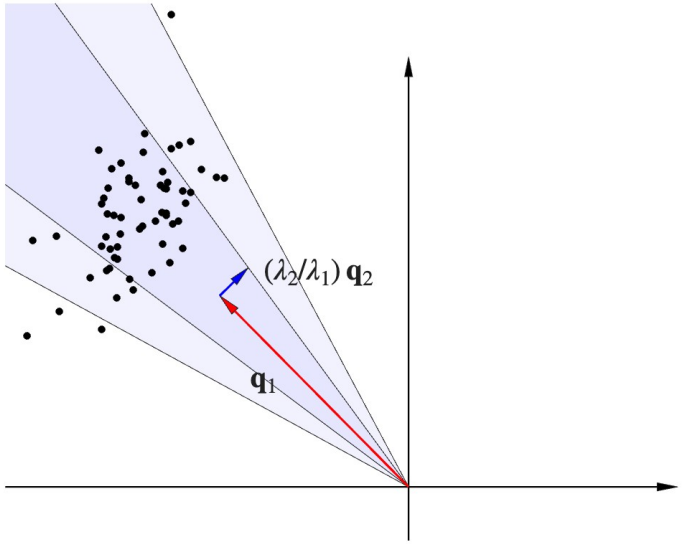
$$\theta_\ell \sim \theta^{r\beta_\ell - 1} \exp\left(-\left(\frac{\theta_\ell}{\vartheta_\ell}\right)^r\right).$$

Structural prior interpretation

Let $H^{(\ell)} = Q^{(\ell)}\Lambda^{(\ell)}(V^{\ell})^{T}$ be the SVI
$H^{(\ell)}$

The first singular vector q_1 (red ar
points in the overall direction of
columns of $H^{(\ell)}$

The next scaled singular vectors (blu
rows) indicate the next most spread,



$G^{(\ell)}$ favors vectors $h)^{\ell)}$ in the direction of the range of $H^{(\ell)}$, the magnitude being penalized by $\theta_{\ell}^{-\frac{1}{2}}$.

MAP for the compressed dictionary matching problem

It follows from Bayes' formula that the posterior for the pair (h, θ) is

$$\pi_{h,\theta|b}(h,\theta) \propto \exp\Bigg(\underbrace{-\frac{1}{2}\|\mathsf{C}_{\mathrm{DCE}}^{-1/2}(\mathsf{W}h + \mu_{\mathrm{DCE}} - b)\|^2}_{\textit{likelihood}} - \underbrace{\frac{1}{2}\sum_{\ell=1}^{L} \frac{\|h^{(\ell)}\|^2_{(\ell)}}{\theta_\ell}}_{\textit{coupling}} - \underbrace{\Phi(\theta)}_{\textit{prior}} \Bigg),$$

where

$$\Phi(\theta) = \sum_{\ell=1}^{L} \left(\frac{\theta_\ell}{\vartheta_\ell}\right)^r + \sum_{\ell=1}^{L} \left(\frac{r_\ell}{2} + 1 - r\beta_\ell\right) \log \theta_\ell.$$

The MAP estimate is the solution of the compressed dictionary matching problem.

MAP solution by IAS for group sparsity

Iterative Alternating Sequential (IAS) optimizer is based on the two following steps:

- With current value $\theta = \theta_c$, update h by minimizing

$$\frac{1}{2}\|\mathsf{C}_{\mathrm{DCE}}^{-1/2}(\mathsf{W}h + \mu_{\mathrm{DCE}} - b)\|^2 + \frac{1}{2}\sum_{\ell=1}^{L}\frac{\|h^{(\ell)}\|^2_{(\ell)}}{\theta_\ell},$$

  (linear least squares problem).

- With current value of $h = h_c$, update θ by minimizing

$$\frac{1}{2}\sum_{\ell=1}^{L}\frac{\|h^{(\ell)}\|^2_{(\ell)}}{\theta_\ell} + \Phi(\theta),$$

  which is a straightforward component-wise optimization problem.

Back to the dictionary matching problems

Recapping:

- The MAP estimate of (h, θ) can be used to determine which subdictionaries are relevant to explain b (those with larger θ_ℓ).
- The pair (h_{MAP}, θ_{MAP}) only used information in compressed subdictionaties
- This may be sufficient for classification task – or not.
- The last step, matching with reduced dictionary, which may also improve classification results.

Preprocessing: identify indices $j \in I_{\rm defl}$ such that, for some p, $0 < p < 1$,

$$\theta_j > p \max_{1 \le \ell \le L} \theta_\ell,$$

and define the reduced dictionary

$$\mathrm{D}^{\rm defl} = [D^{(j_1)}, \ldots, D^{(j_r)}], \quad j_\ell \in I_{\rm defl}.$$

Step 4: Reduced Dictionary matching

- Use HBM to solve the linear inverse problem

$$b = \mathsf{D}^{\mathrm{defl}} x + \eta$$

- To promote matching with few dictionary atoms, we use an element-wise sparsity promoting prior for x. Assume mutually independent $x_j \sim \mathcal{N}(0, \theta_j)$
- The variances θ_j, mutually independent, are distributed according to generalized gamma.
- The IAS algorithm is used to find the MAP estimate

$$(x_{\mathrm{MAP}}, \theta_{\mathrm{MAP}}) = \mathrm{argmin}\left(\|b - \mathsf{D}^{\mathrm{defl}} x\|^2 + \sum_{j=1}^{n_d} \frac{x_j^2}{\theta_j} + \Psi(\theta) \right).$$

Brain region identification from MEG data

1. Dictionary: Simulated MEG magnetometer and gradiometer measurement in correspondence to activity in a patch in a given ROI. (100 random patches per region)
2. Dictionary partitioned in 148 subdictionaries (= number of regions in Destrieux atlas)
3. Test how accurately each region can be identified from MEG data.
4. Build confusion matrix: scale columns to have unit 1-norm and use to quantify source ROI-fication uncertainty

Reinterpretation of confusion matrix

- Plain vanilla confusion matrix

$$C_{ij} = \#\text{ cases classified in class } j \text{ when true calss is } i.$$

- Column scaling:

$$P_{ij} = \frac{C_{ij}}{\sum_{i'=1}^{L} C_{i'j},} \quad j \text{ given,}$$

Interpreted as

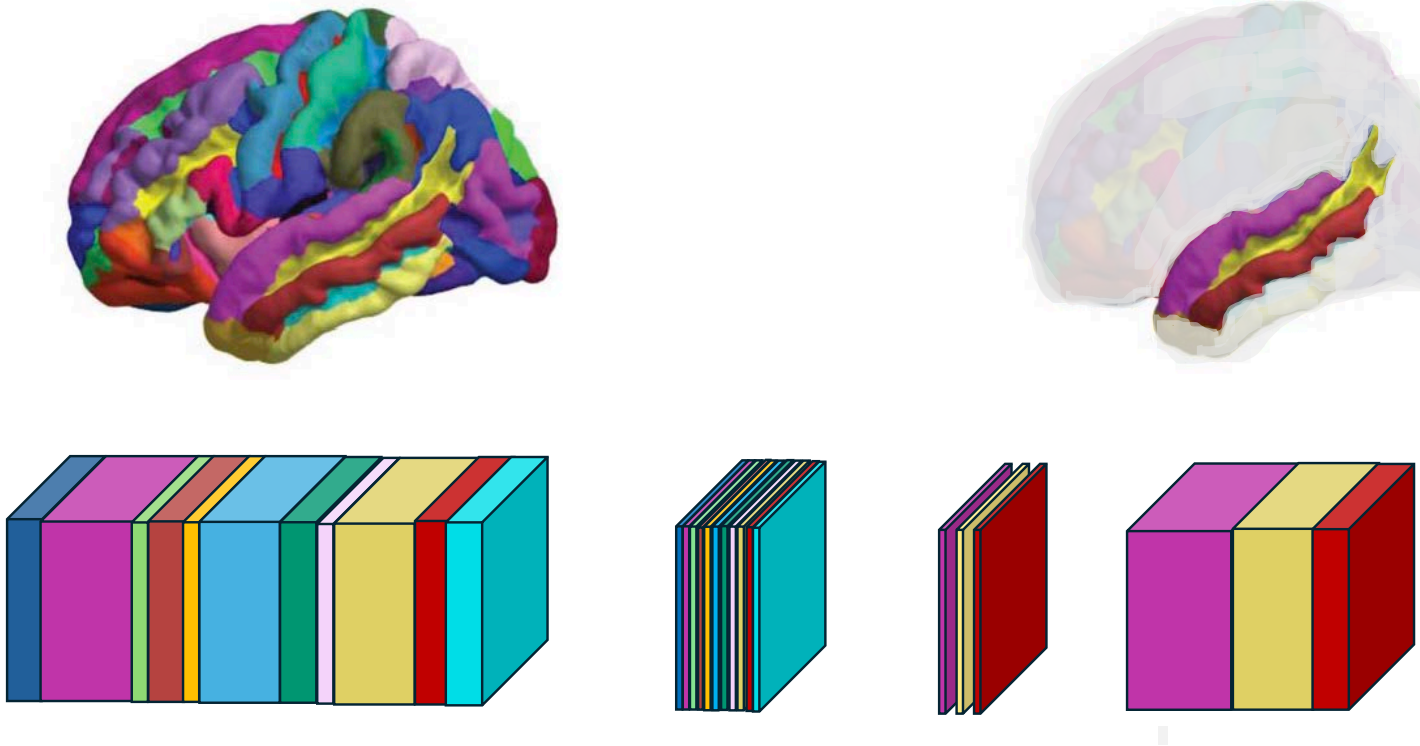
$$P_{ij} = \mathbb{P}(\text{region } i \text{ is active} \mid \text{region } j \text{ was identified}), \ \textbf{(recall)}$$

- Row scaling:

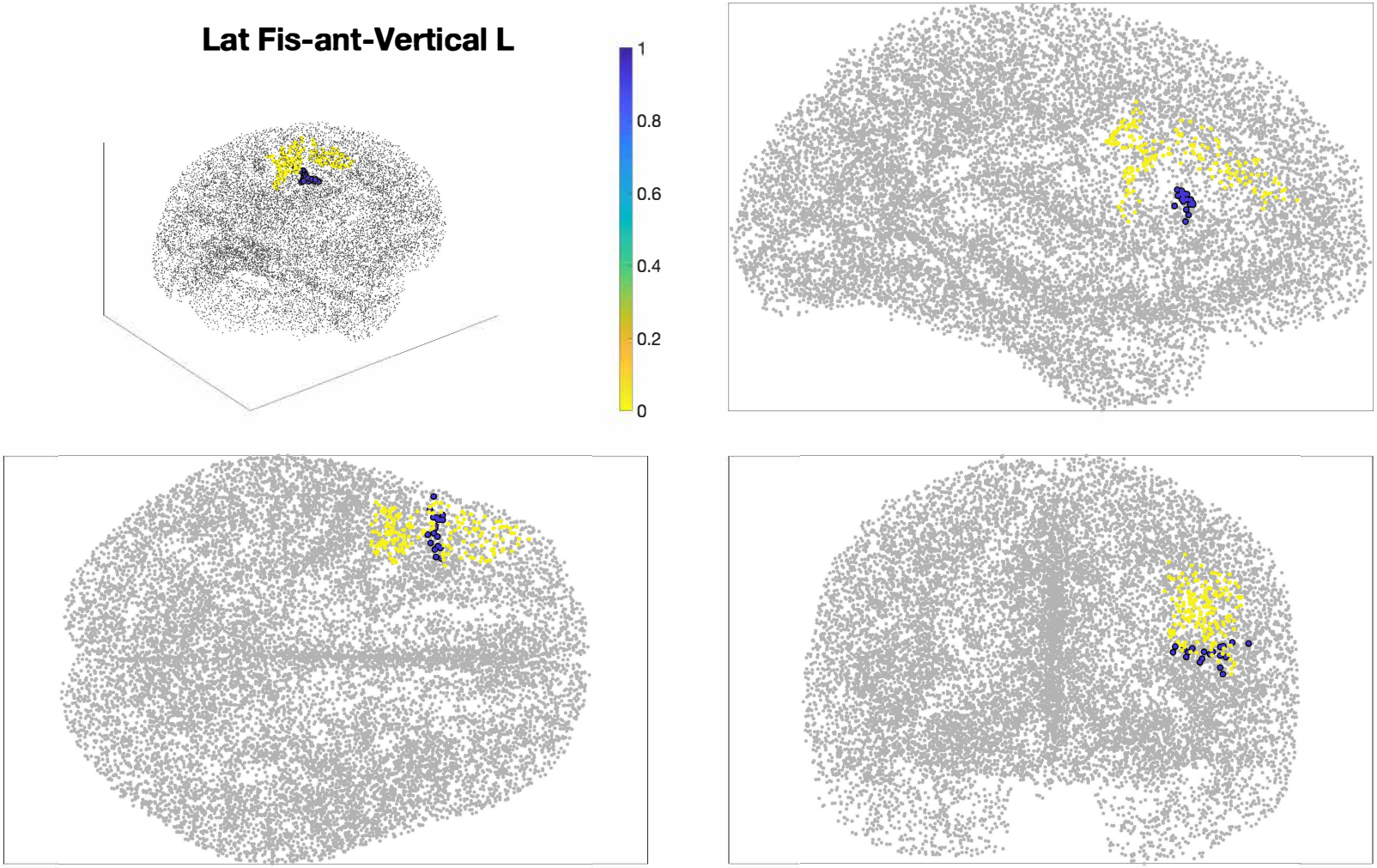
$$Q_{ij} = \frac{C_{ij}}{\sum_{j'=1}^{L} C_{ij'},} \quad i \text{ given,}$$

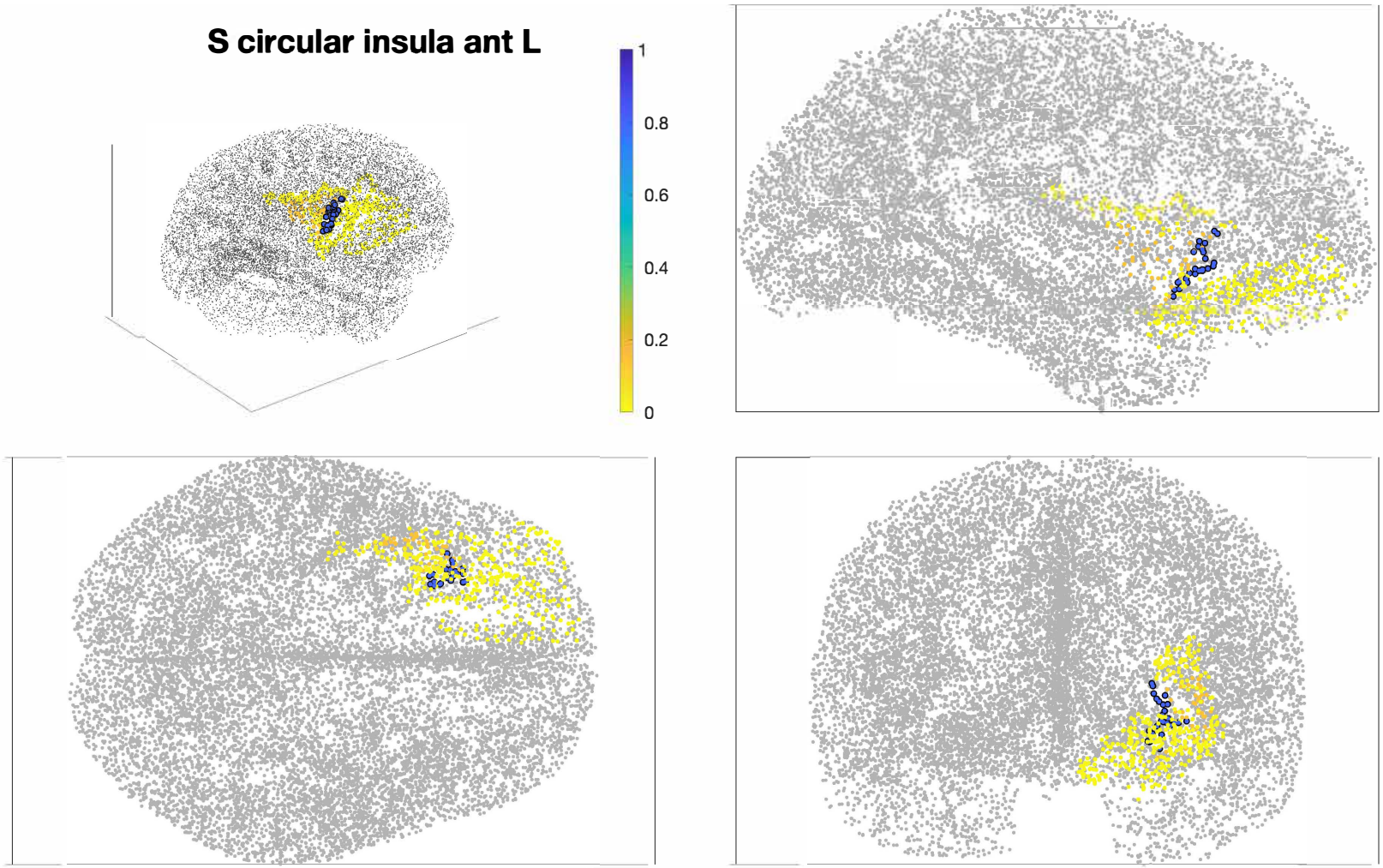
interpreted as

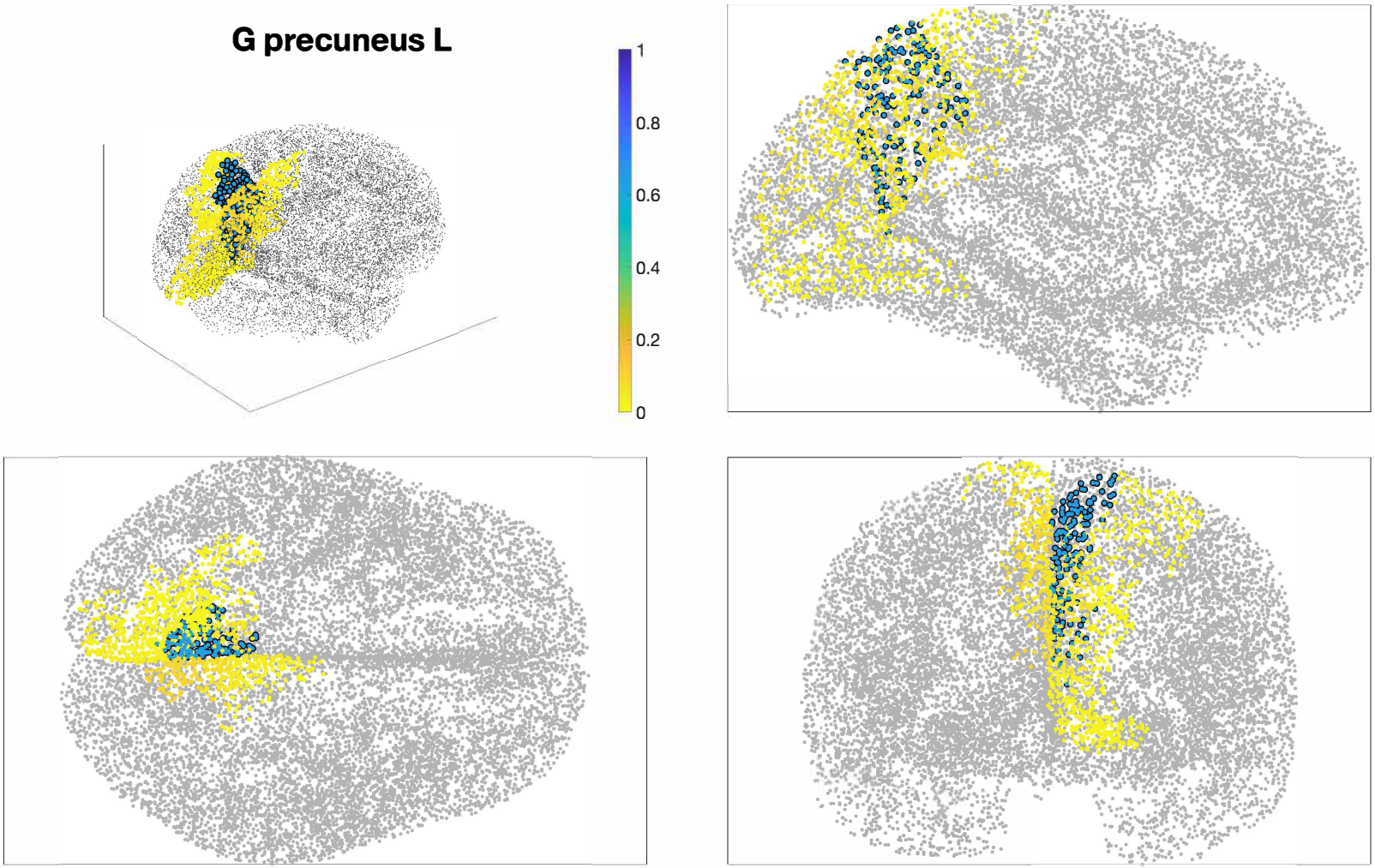
$$Q_{ij} = \mathbb{P}(\text{region } j \text{ is identified} \mid \text{region } i \text{ is active}), \ \textbf{(precision)}.$$

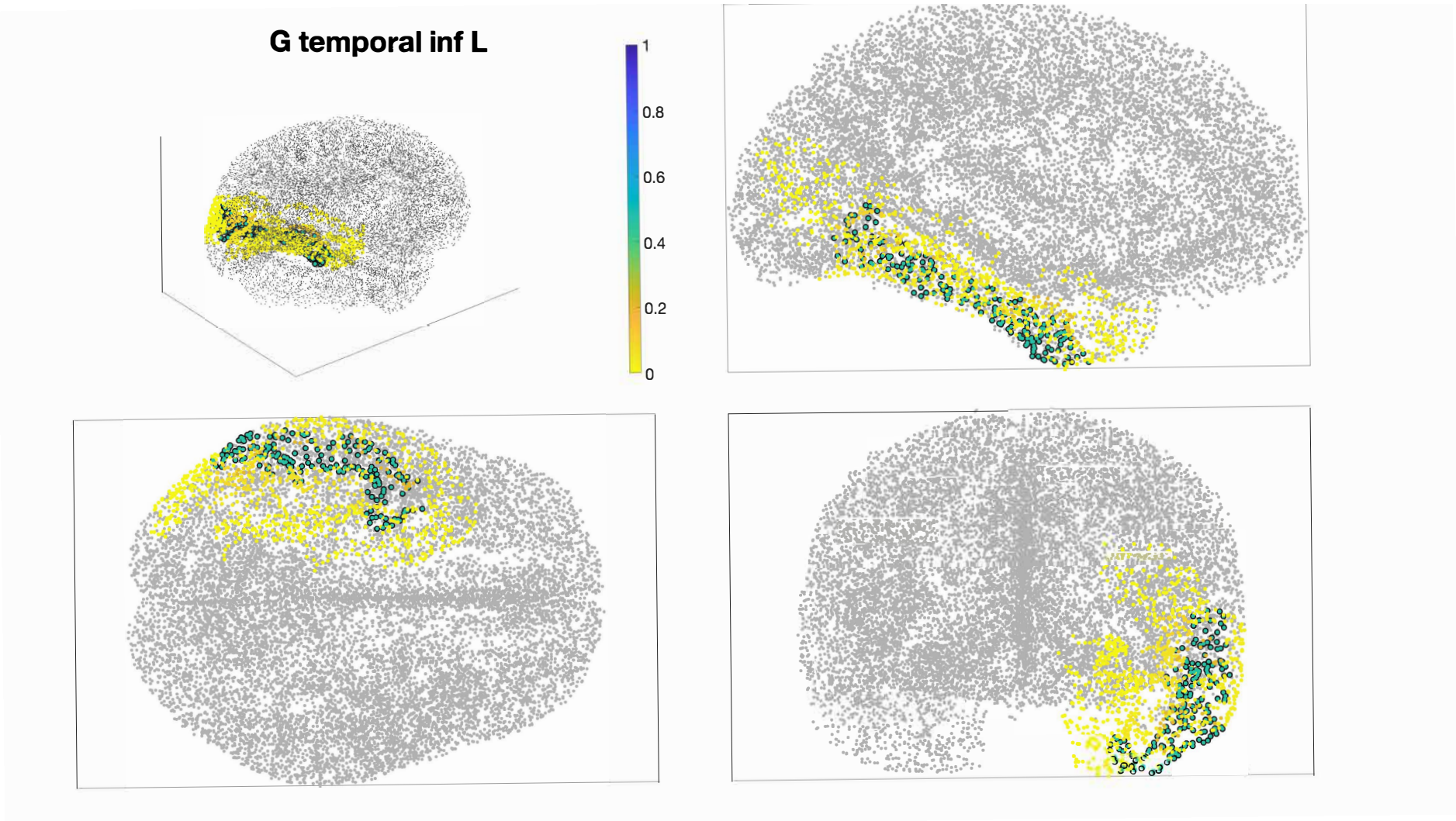


UQ via confusion matrix




S circular insula ant L
1
0.8
0.6
0.4
0.2
0


G precuneus L
1
0.8
0.6
0.4
0.2
0


G temporal inf L
1
0.8
0.6
0.4
0.2
0

Main References

1. Calvetti D and Somersalo E (2023) Bayesian Scientific Computing. Springer Berlin.
2. Calvetti D and Somersalo E (2021) Mathematics of Data Science: a Computational Approach to Clustering and Classification SIAM Philadelphia.
3. Bocchinfuso A, Calvetti D and Somersalo E (2024) Bayesian sparsity and class sparsity priors for dictionary learning and coding. Journal of Computational Mathematics and Data Science, 11, p.100094.
4. Bocchinfuso A, Calvetti D and Somersalo E (2025) Bayesian dictionary learning estimation of cell membrane permeability from surface pH data. SIAM Life Science, to appear. arXiv:2507.09651 2025
5. Pragliola M, Calvetti D and Somersalo E (2022) Overcomplete representation in a hierarchical Bayesian framework. Inverse Problems and Imaging **16**: 19-38. doi: 10.3934/ipi.2021039
6. Waniorek N, Somersalo E and Calvetti D (2023) Bayesian hierarchical dictionary learning. Inverse Problems, **39**: p.024006.

References

7. Calvetti D and Somersalo E (2025) Subspace splitting fast sampling from Gaussian posterior distributions of linear inverse problems. SIAM/ASA Journal on Uncertainty Quantification 14 (1), 111-141. arXiv preprint arXiv:2502.05703
8. Calvetti D and Somersalo E (2025) Dictionary learning methods for brain activity mapping with MEG data. arXiv preprint arXiv:2510.19702
9. Mason-Mackay, A etc al (2026) Sparse Dictionary-Based Solution of Dynamic Inverse Problems. arXiv preprint arXiv:2602.18593, year=2026