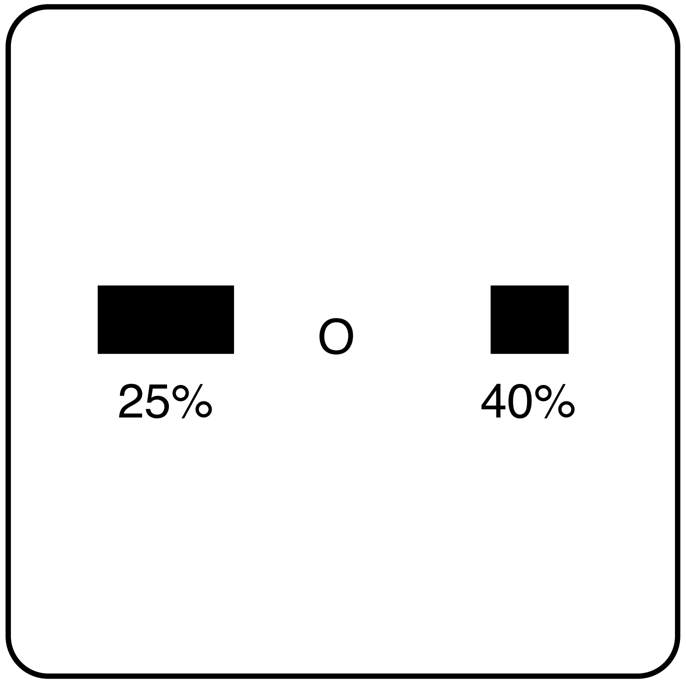
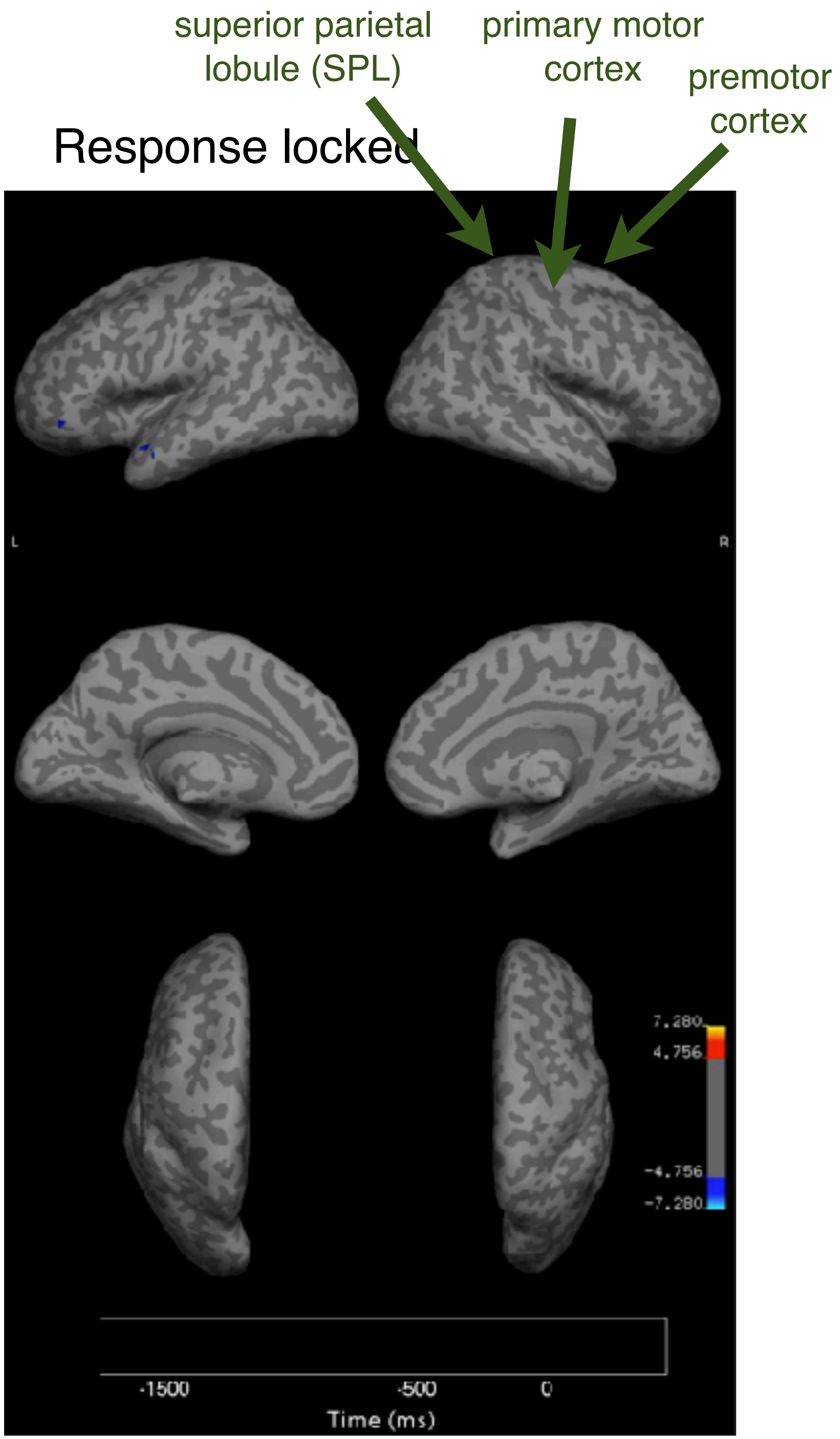
Sensor Space Analysis

OHBA MEG Analysis Workshop

- Induced analysis of the decision making period:
  - ***source reconstruction***
  - *epoching*: time-locked to when the response is given
  - compute the ***average evoked power*** (the induced response, ERD/ERS) from 1-12Hz
  - ***group averaged*** over 30 subjects



Hunt et al., Nature Neuroscience, 2012.

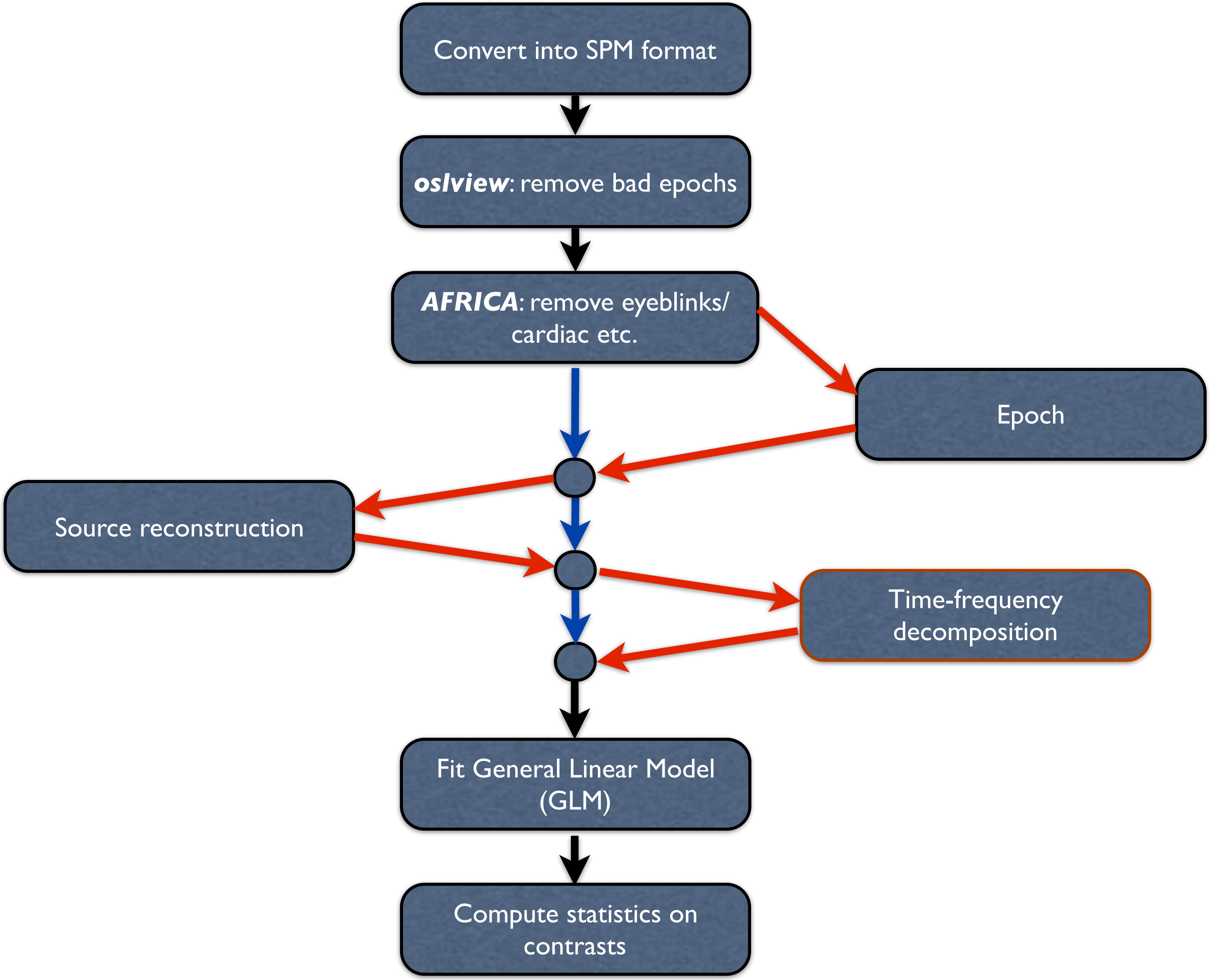


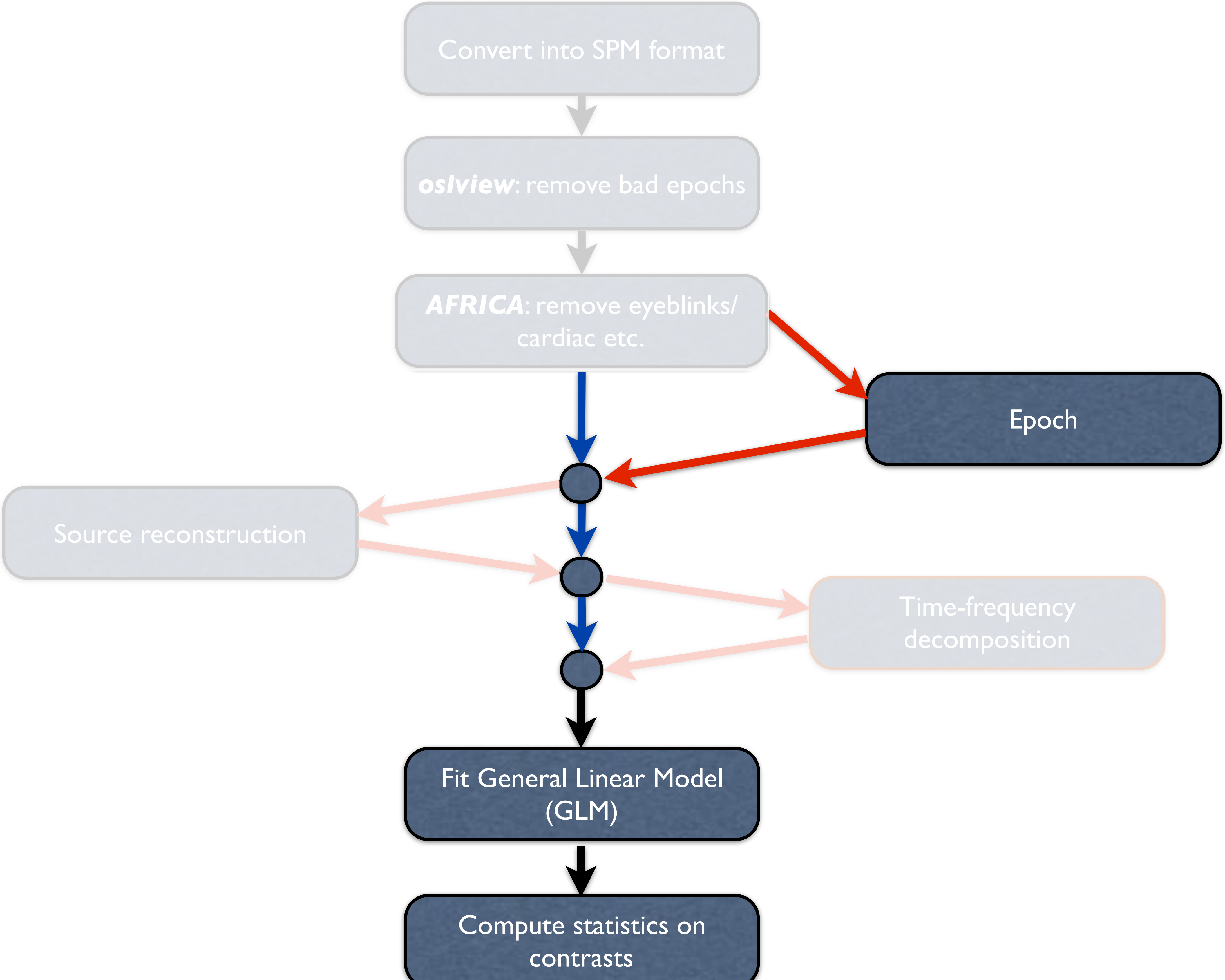
Talk Outline

- Analysing epoched task data in sensor space
- OSL (OHBA's Software Library):
  - OAT (OSL's Analysis Tool)

Epoched Data Example

- Faces versus motorbikes
  - ➡ 240 trials (epochs) of presenting pictures of faces
  - ➡ 120 trials (epochs) of presenting pictures of motorbikes
- We want to compare the responses time-locked to stimulus presentation (i.e. the Event-Related Fields (ERFs))

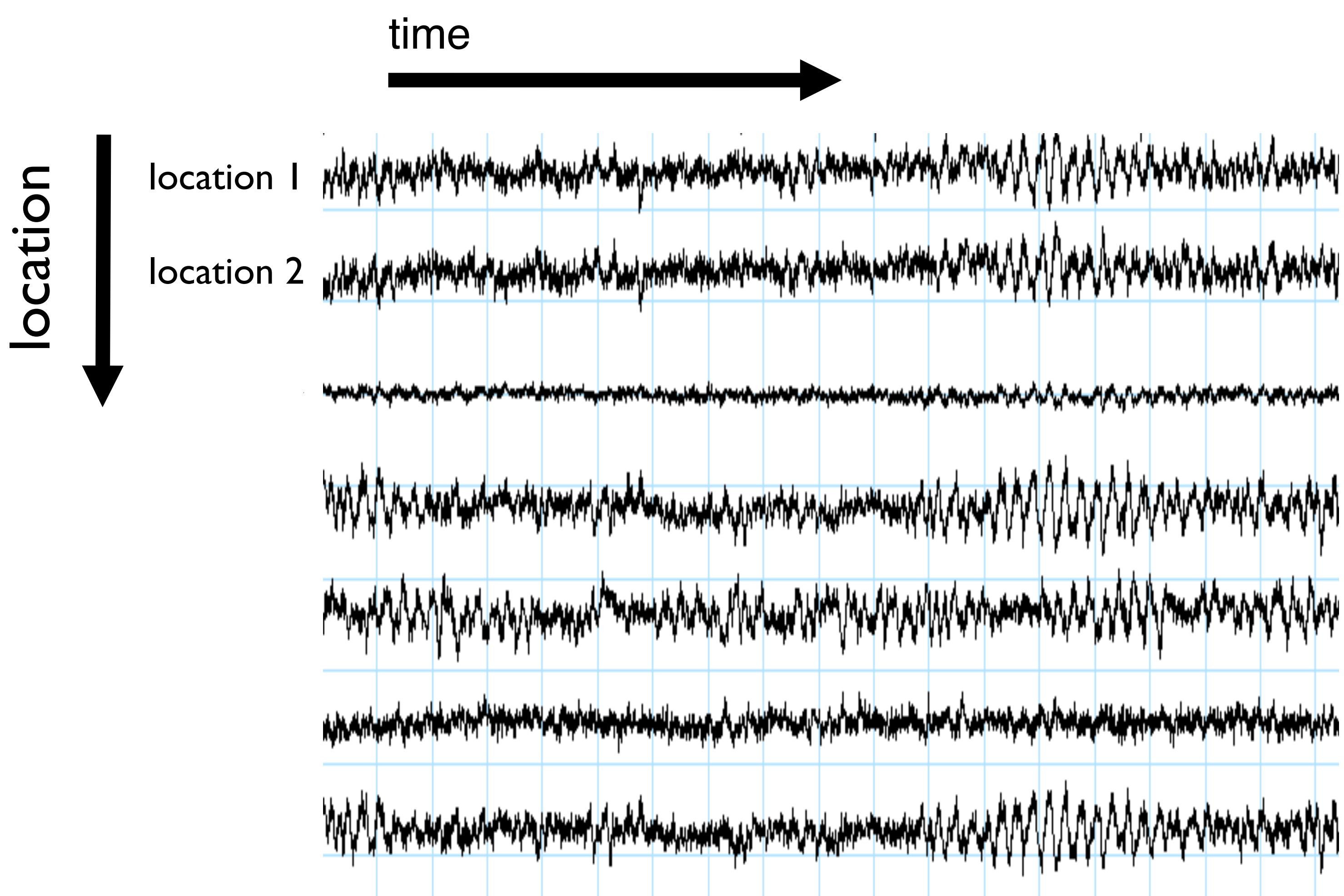

Convert into SPM format
oslview: remove bad epochs
AFRICA: remove eyeblinks/ cardiac etc.
Epoch
Source reconstruction
Time-frequency decomposition
Fit General Linear Model (GLM)
Compute statistics on contrasts


Convert into SPM format
oslview: remove bad epochs
AFRICA: remove eyeblinks/ cardiac etc.
Epoch
Source reconstruction
Time-frequency decomposition
Fit General Linear Model (GLM)
Compute statistics on contrasts

Epoching and ERFs

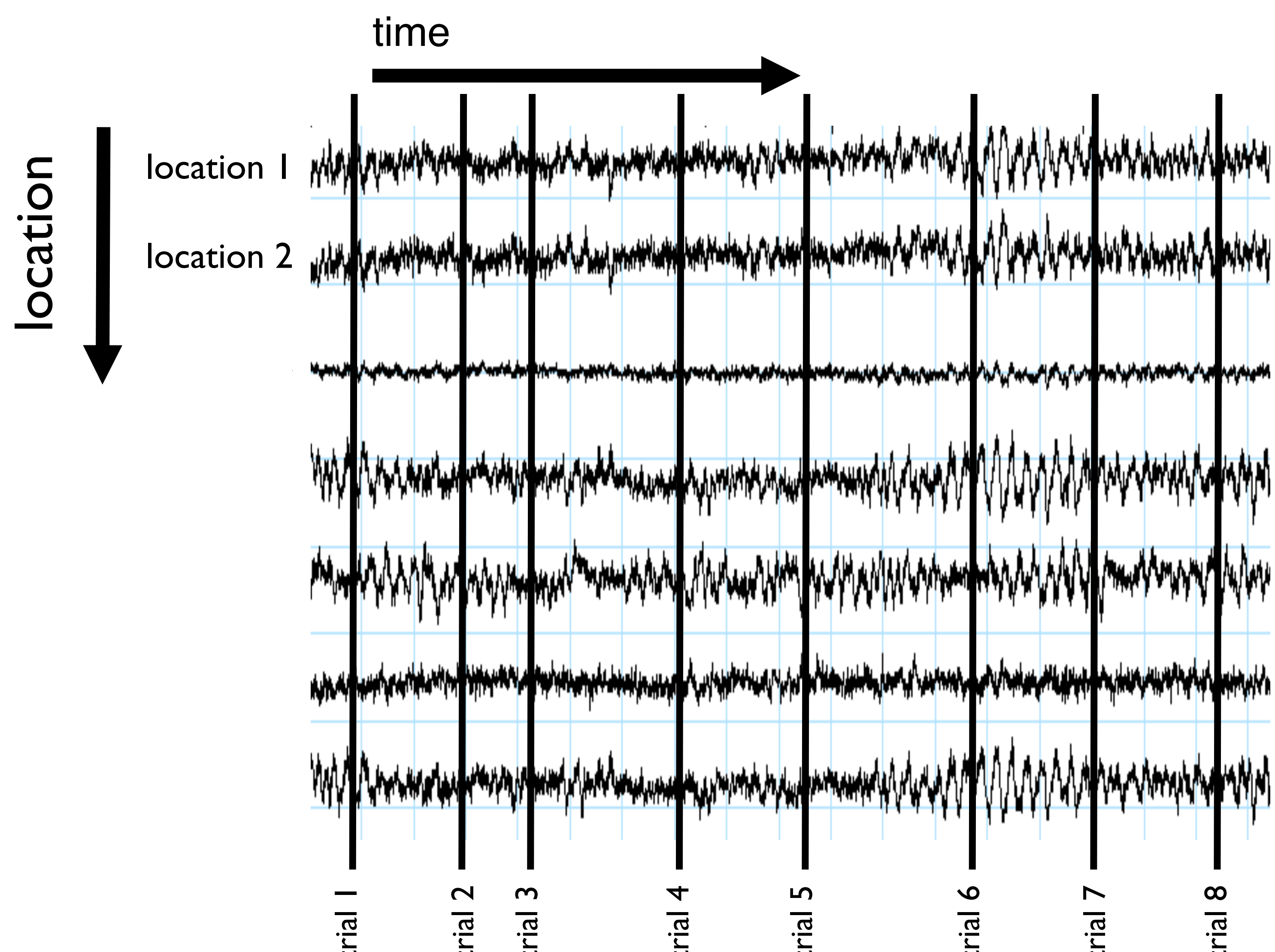
Epoching takes in source or sensor data:

locations x timepoints



Epoching and ERFs

Epoching takes in source or sensor data:
locations x timepoints

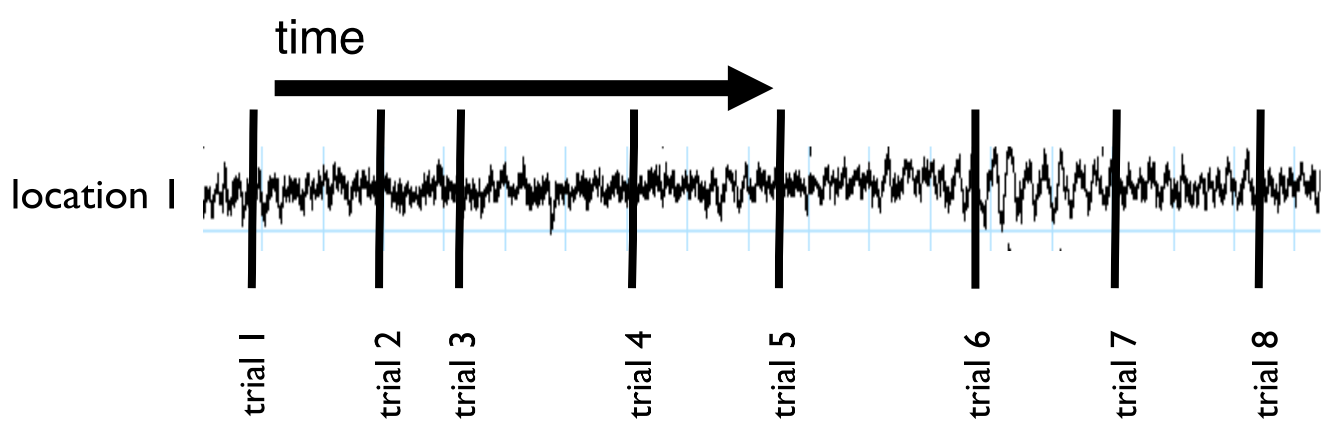


1. Identify when trial events occurred (e.g. the time of stimulus presentation in each trial)

Epoching and ERFs

Epoching takes in source or sensor data:
locations x timepoints

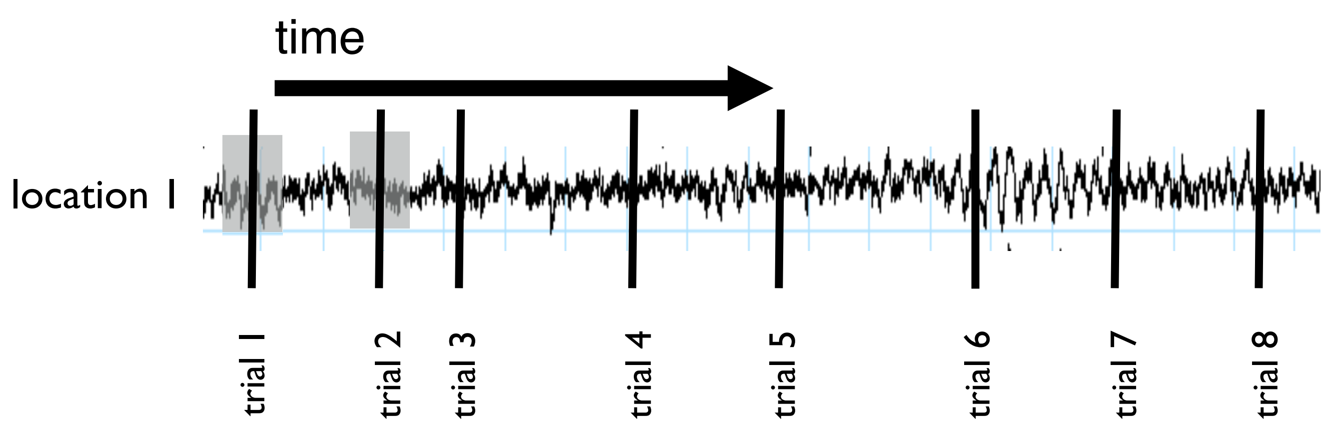
1. Identify when trial events occurred (e.g. the time of stimulus presentation in each trial)



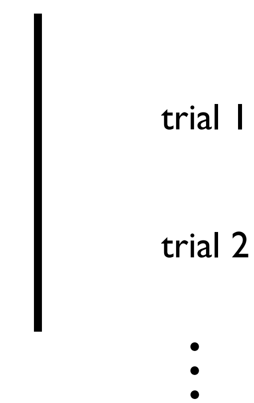
Epoching and ERFs

Epoching takes in source or sensor data:
locations x timepoints

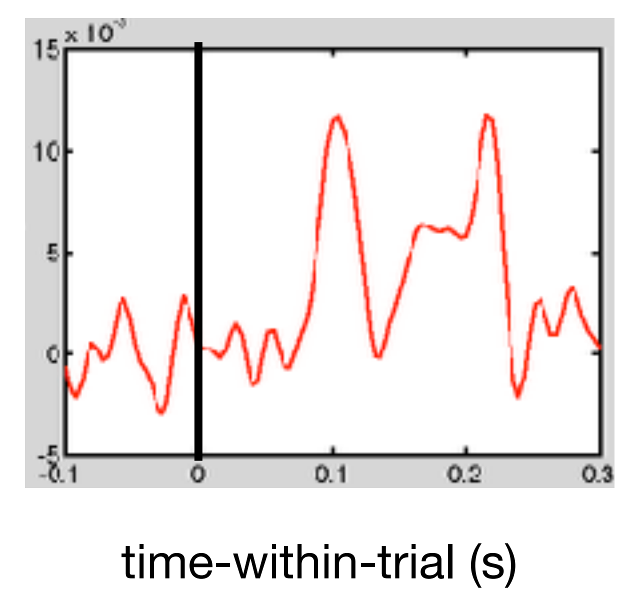
1. Identify when trial events occurred (e.g. the time of stimulus presentation in each trial)



2. Extract time-locked "epochs" of data

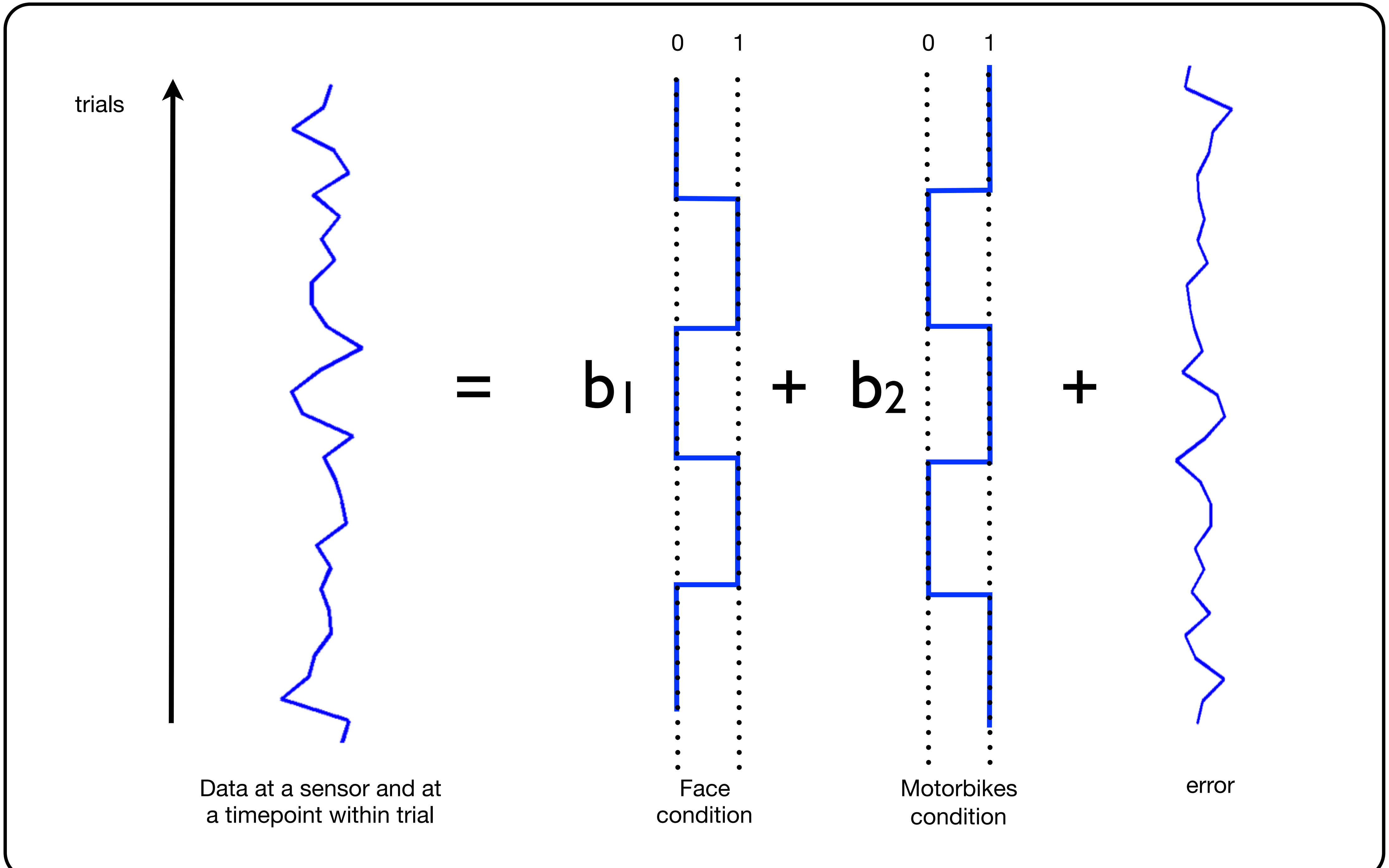


3. Average over all trials to compute an average stimulus response, known as an ERF (Event Related Field)



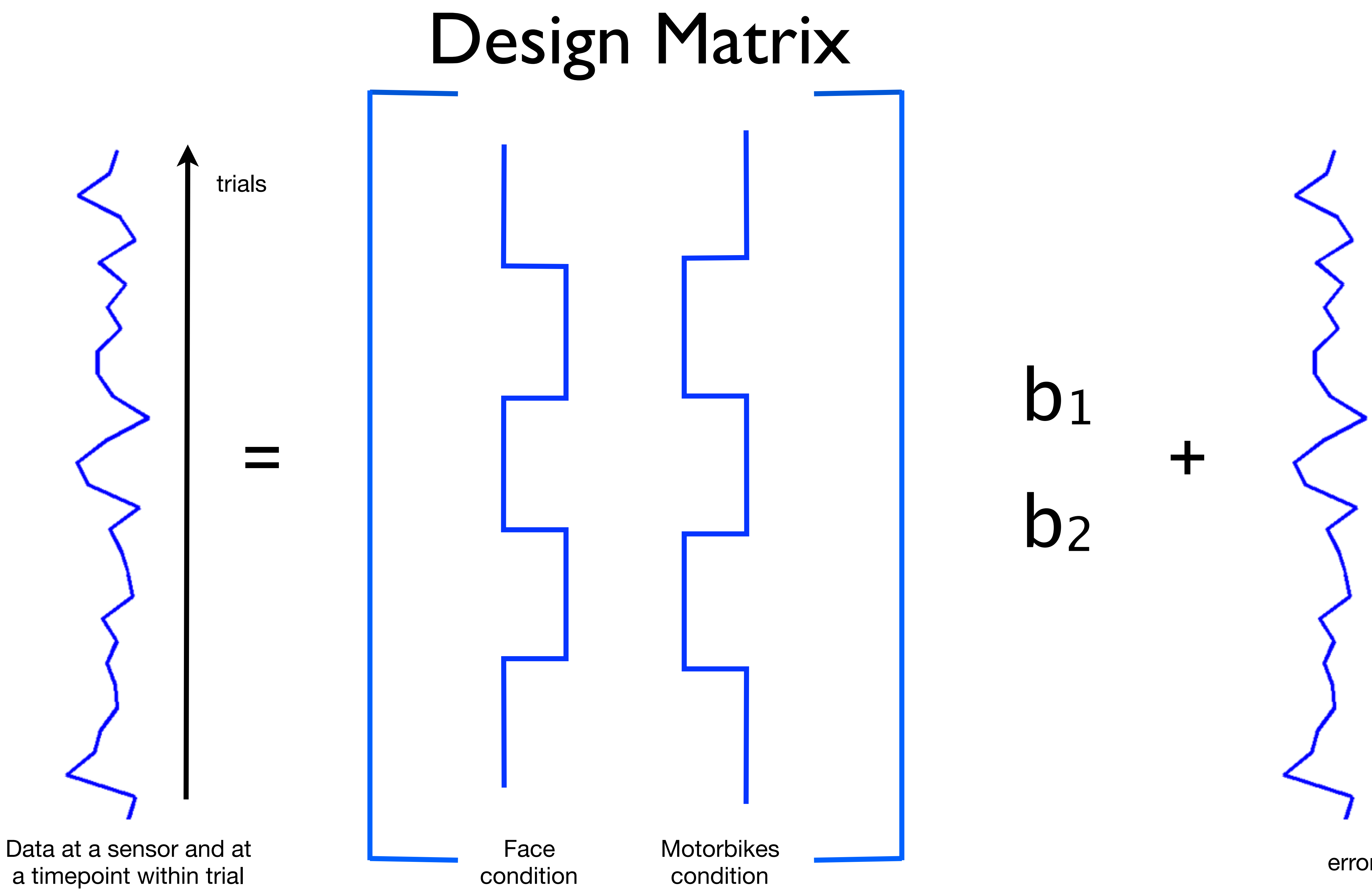
Trial-wise multiple regression

ERFs can be also computed using separate multiple regressions at each sensor and timepoint



Trial-wise GLM

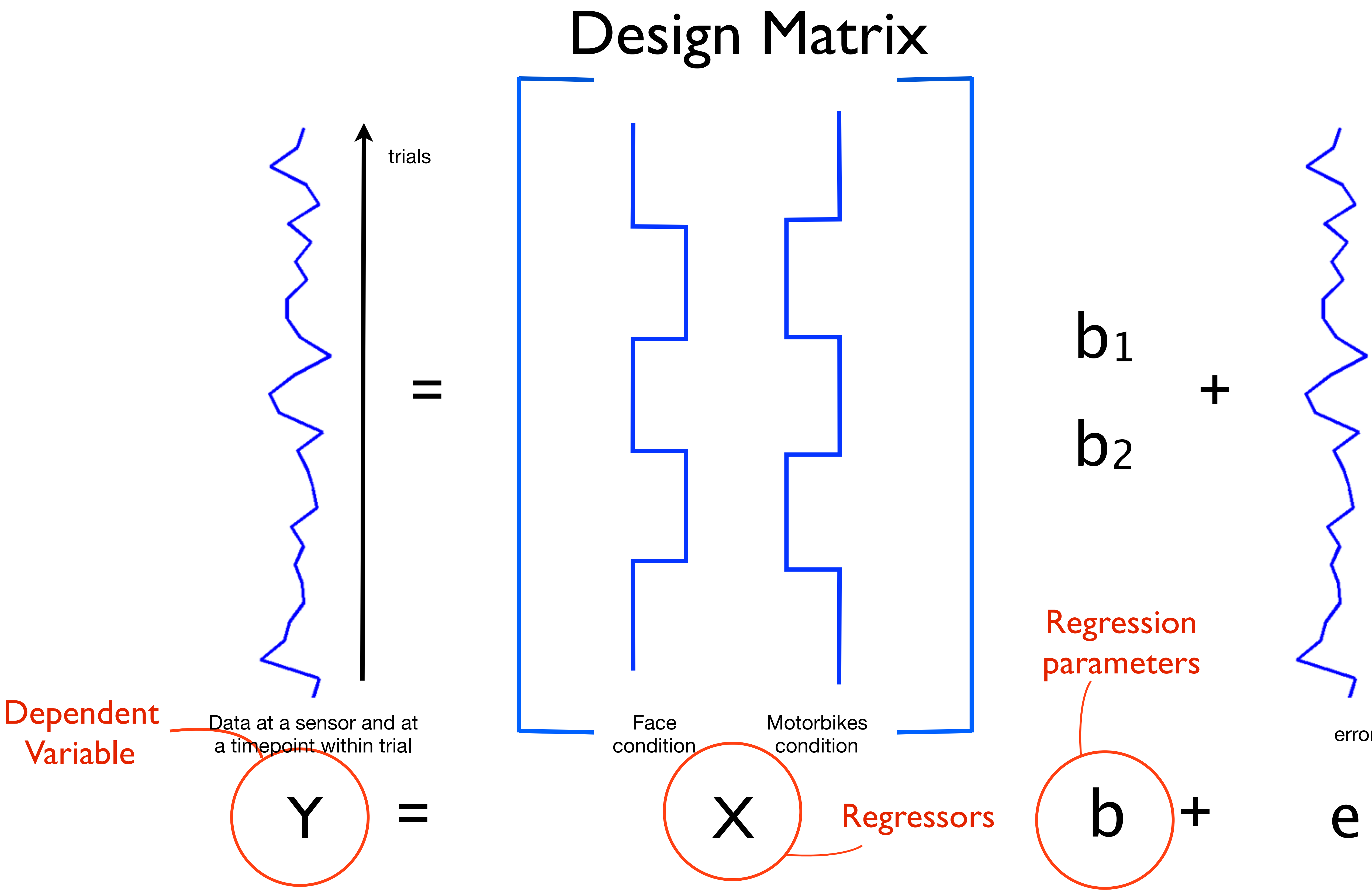
More formally we fit a separate General Linear Model (GLM) at each sensor and timepoint



$$Y = X b + e$$

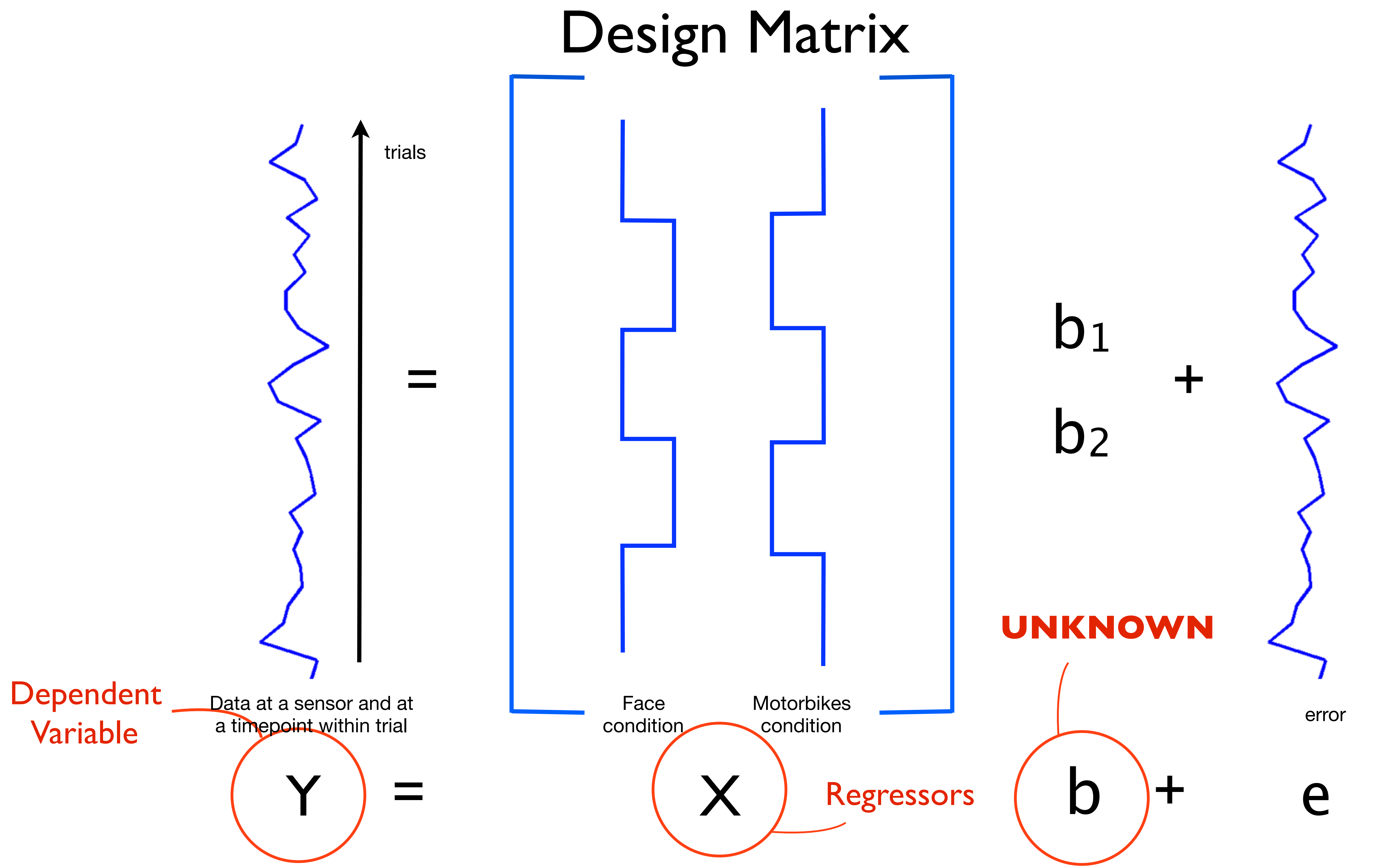
Trial-wise GLM

More formally we fit a separate General Linear Model (GLM) at each sensor and timepoint

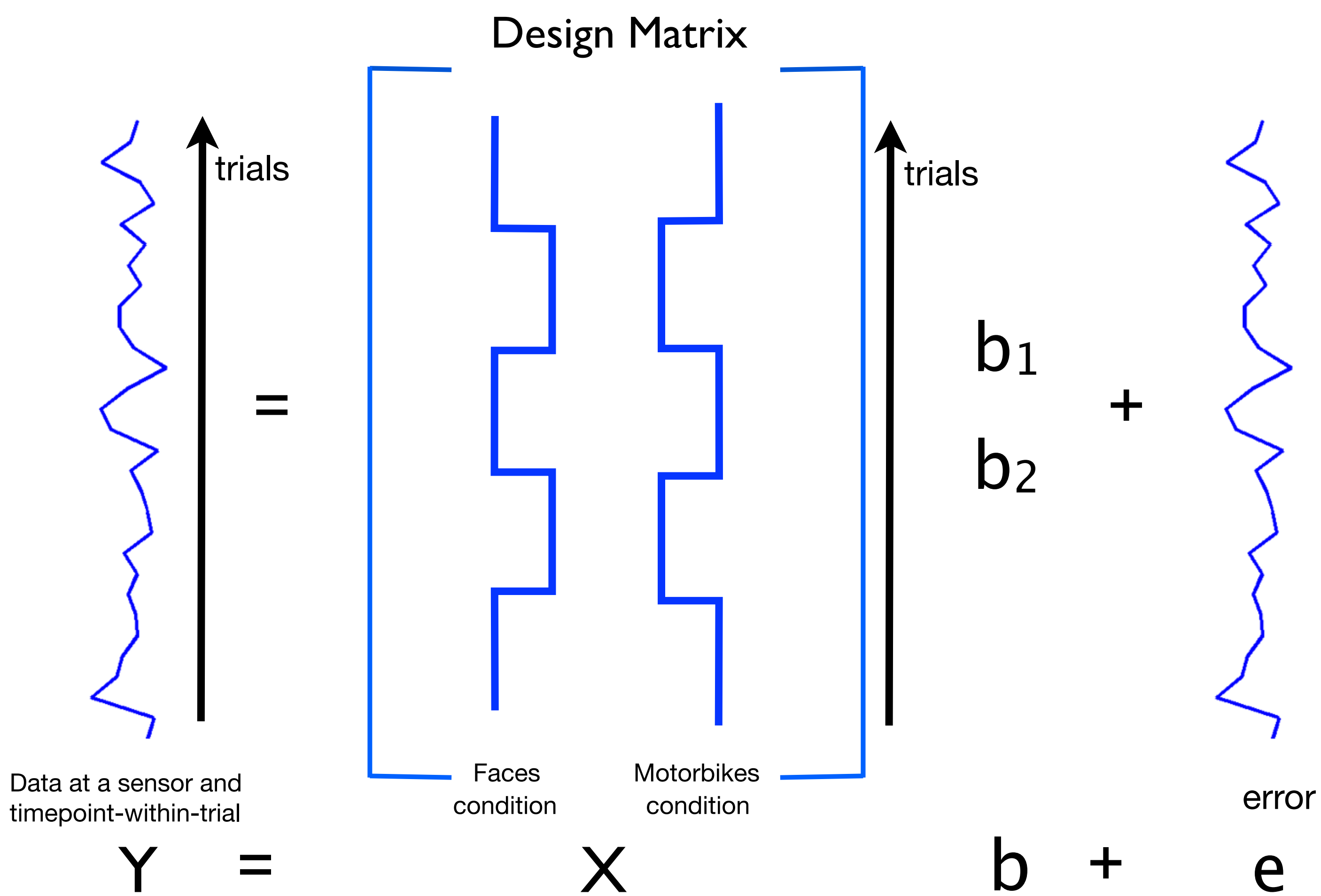


Trial-wise GLM

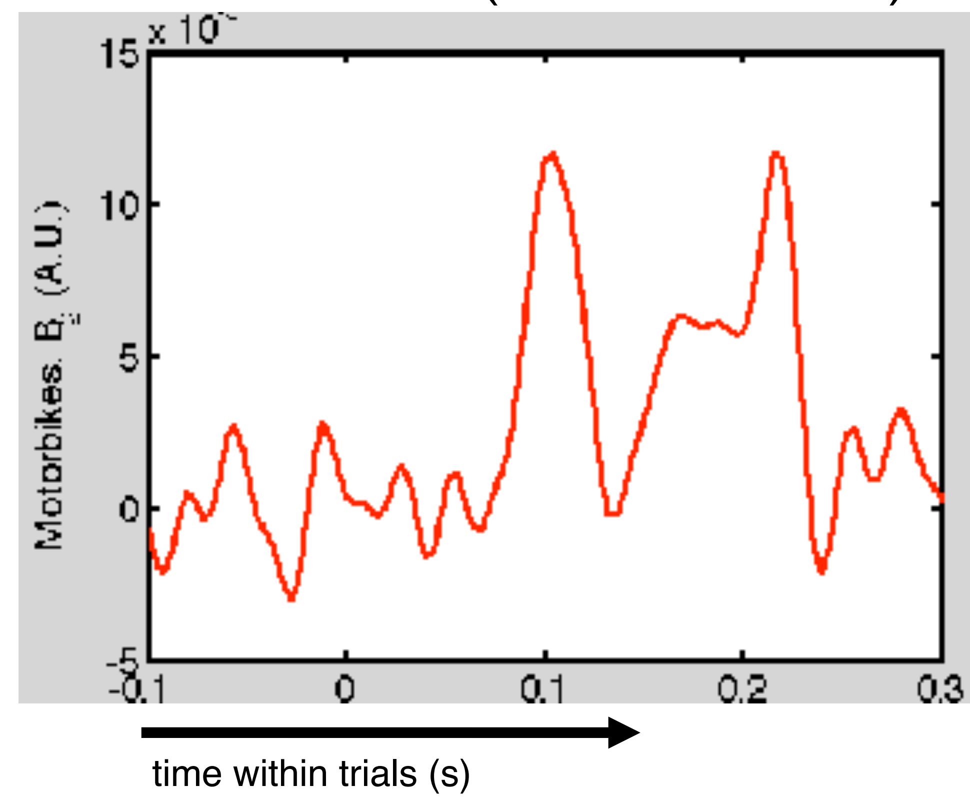
More formally we fit a separate General Linear Model (GLM) at each sensor and timepoint



Fitting over time = ERF



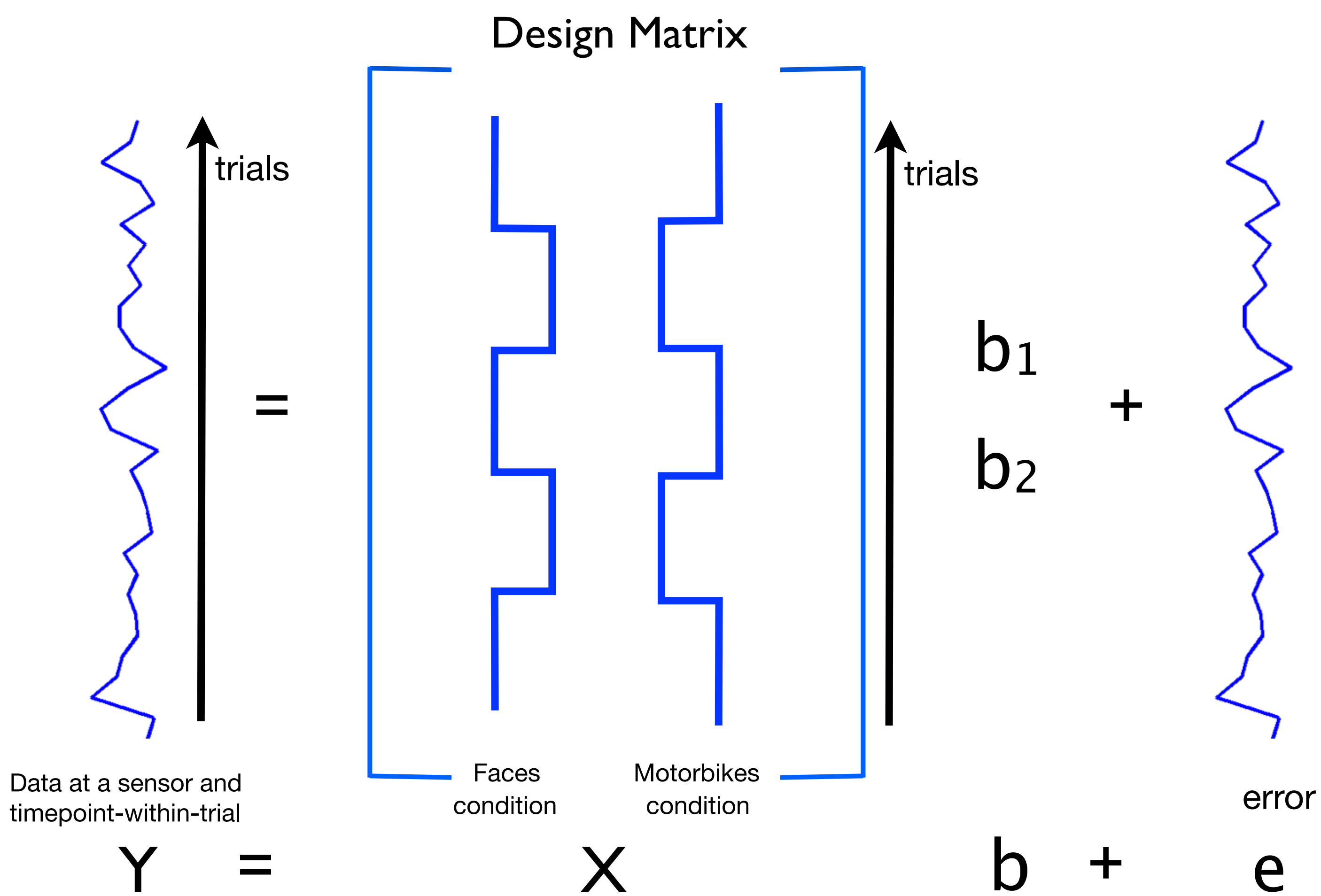
Motorbikes (B_2) at a sensor near the visual cortex (motorbike ERF)



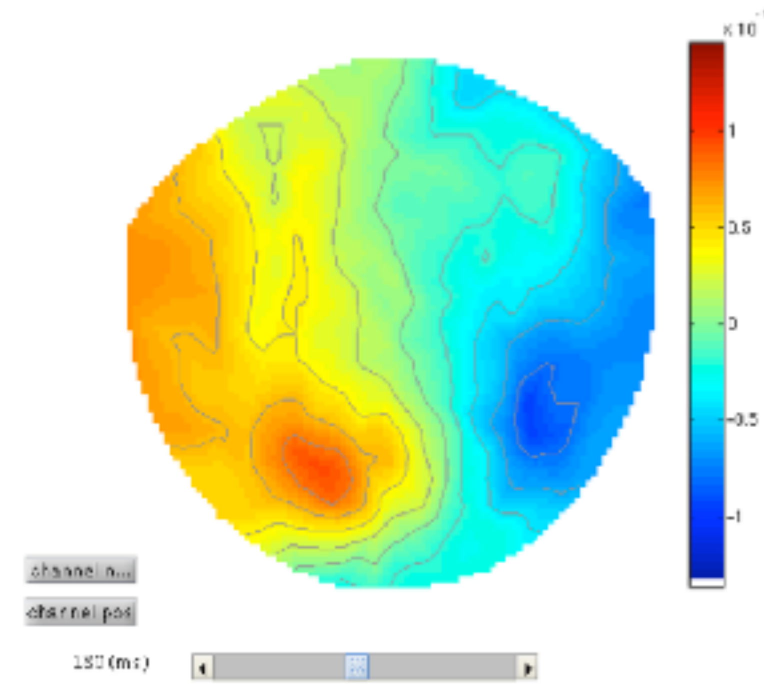
Repeat for all timepoints within-trial at a sensor

$Y = Xb + e$ → B are the Parameter Estimates (PEs) of b

Fitting over sensors



Motorbikes (B_2) at 100ms post-stimulus

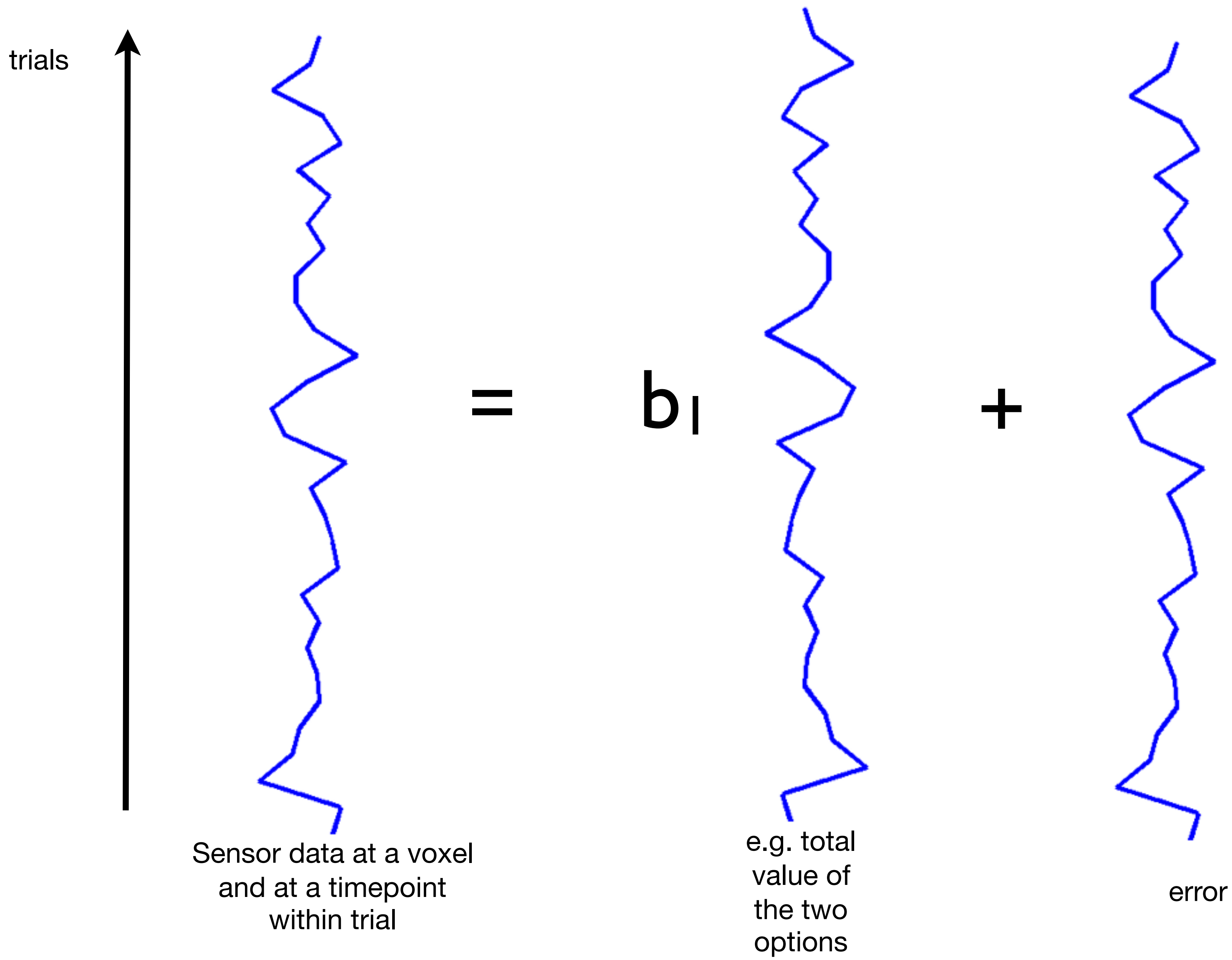


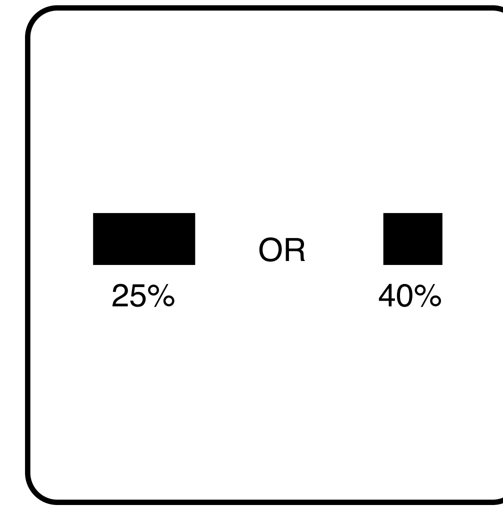
Repeat for all sensors at a time-within-trial

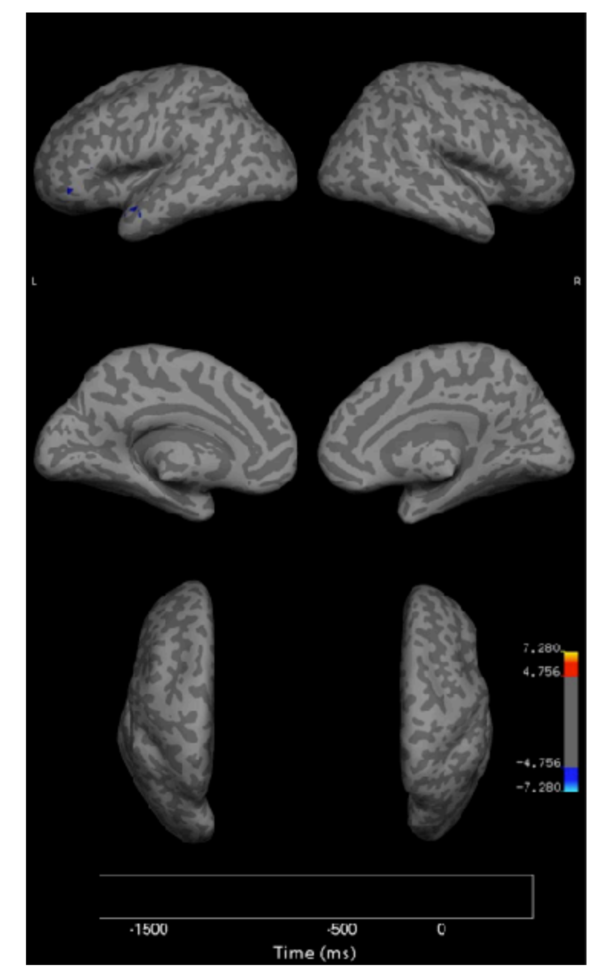
$Y=Xb+e$ → B are the Parameter Estimates (PEs) of b

GLM

Note that the GLM is a general framework, e.g. in which we can also fit continuous variables:







Hunt, Nature Neuroscience, 2012

Other stuff the GLM can do

- Continuous (e.g. behavioural) variables
- Time-frequency (induced response) analysis
- Linear, and higher order, trends between conditions
- Factorial designs (interaction effects)
- F-tests (combined explanatory power over multiple contrasts)
- Subject-wise GLMs at the group level (e.g. patients vs controls)
- See the FSL course FEAT/FMRI Preprocessing and Model-Based slides at:

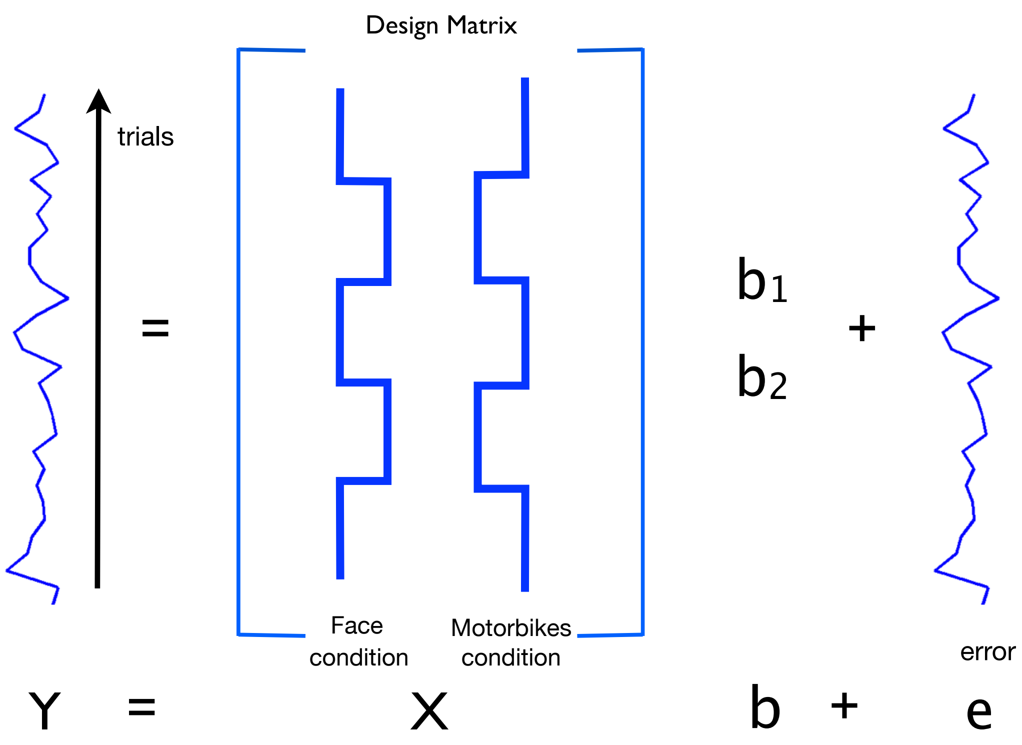
http://www.fmrib.ox.ac.uk/fslcourse

Contrasts

A COntrast of Parameter Estimates (COPE) is a linear combination of the regression parameter estimates, e.g.

Contrast [1 0] gives a COPE $= 1 x B_1 + 0 x B_2$
$= B_1$

Contrast [1 -1] gives a COPE $= 1 x B_1 - 1 x B_2$
$= B_1 - B_2$

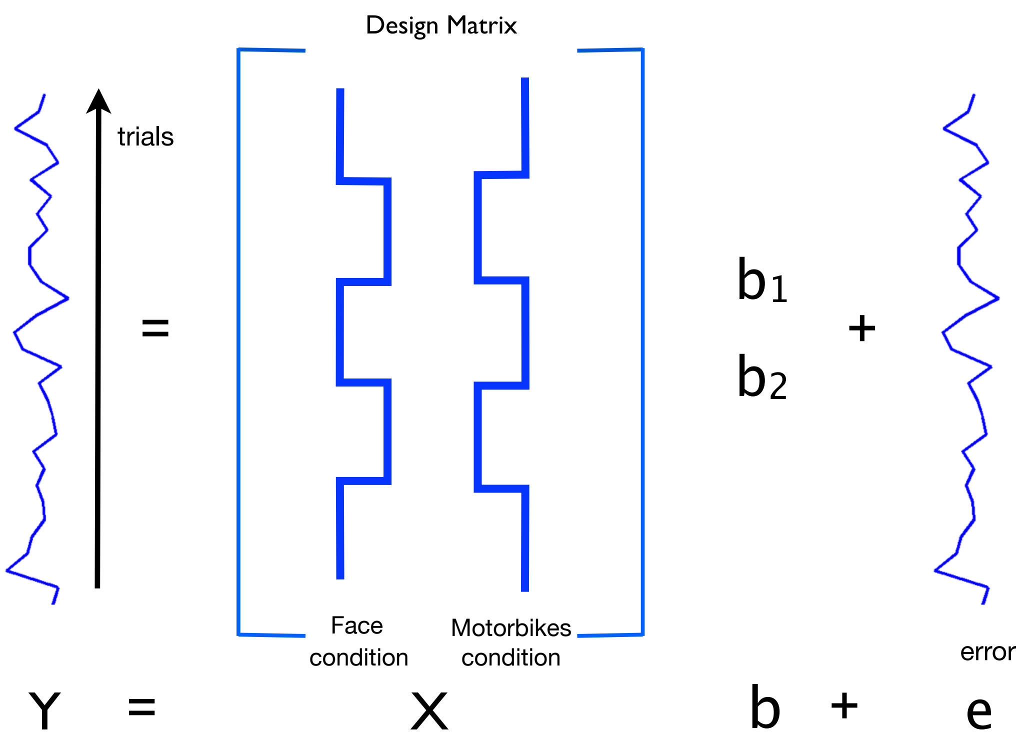


Contrasts

A COntrast of Parameter Estimates (COPE) is a linear combination of the regression parameter estimates, e.g.

Contrast [1 0] gives a COPE $=1\text{x}B_1+0\text{x}B_2$
$= B_1$

Contrast [1 -1] gives a COPE $=1\text{x}B_1-1\text{x}B_2$
$= B_1 - B_2$



Use a t-test to test the null hypothesis that COPE=0:

t-statistic:

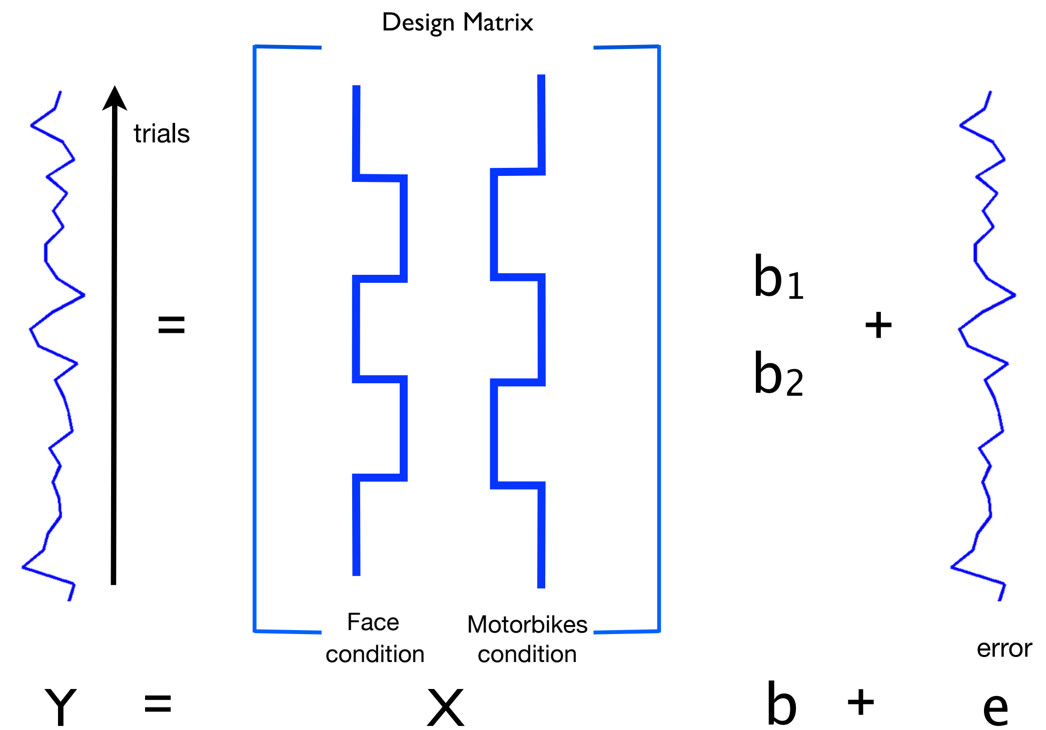
$$t = \frac{COPE}{std(COPE)}$$

Contrasts

A COntrast of Parameter Estimates (COPE) is a linear combination of parameter estimates, e.g.

Contrast [0 1] gives a COPE $= 0 \times B_1 + 1 \times B_2$

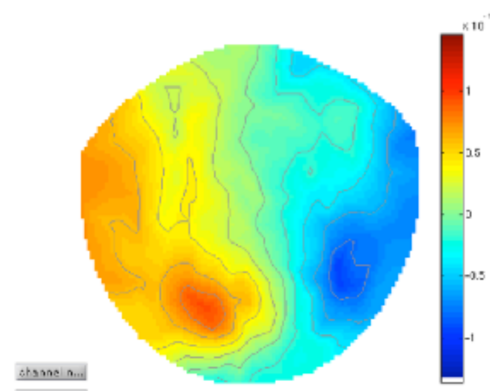
$= B_2$

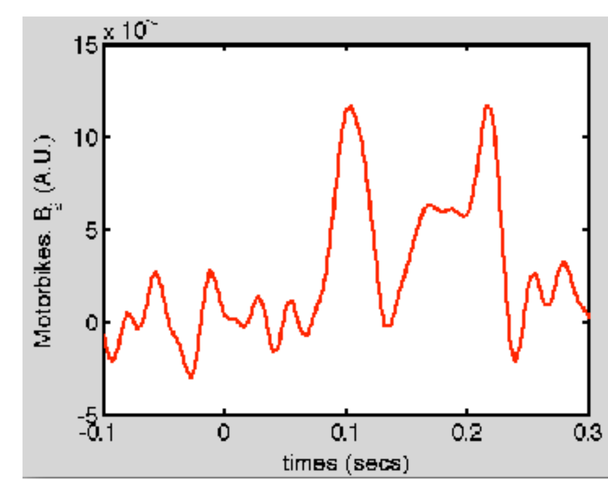


Test the null hypothesis that $B_2=0$

e.g. where in time and space is there significant positive* activity in response to the motorbike condition?

** as we are doing a one-tailed t-test*



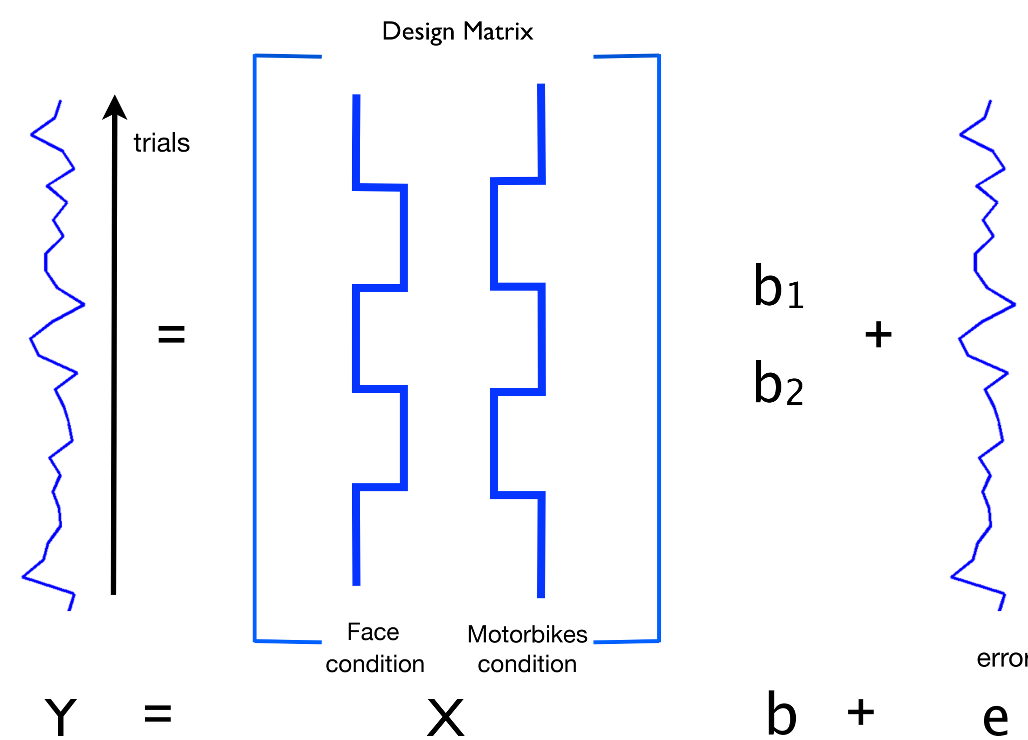


Contrasts

A COntrast of Parameter Estimates (COPE) is a linear combination of parameter estimates, e.g.

Contrast [1 -1] gives a COPE $= 1 \times B_1 - 1 \times B_2$

$= B_1 - B_2$



Test the null hypothesis that $B_1 - B_2 = 0$

e.g. where in time and space is there more* activity in response to the faces than the motorbike condition?

* *as we are doing a one-tailed t-test*

OAT - OSL's Analysis Tool

- **Task-based** analysis in:
  - ➡ sensor space, or
  - ➡ source space (e.g. via beamforming)

OAT - OSL's Analysis Tool

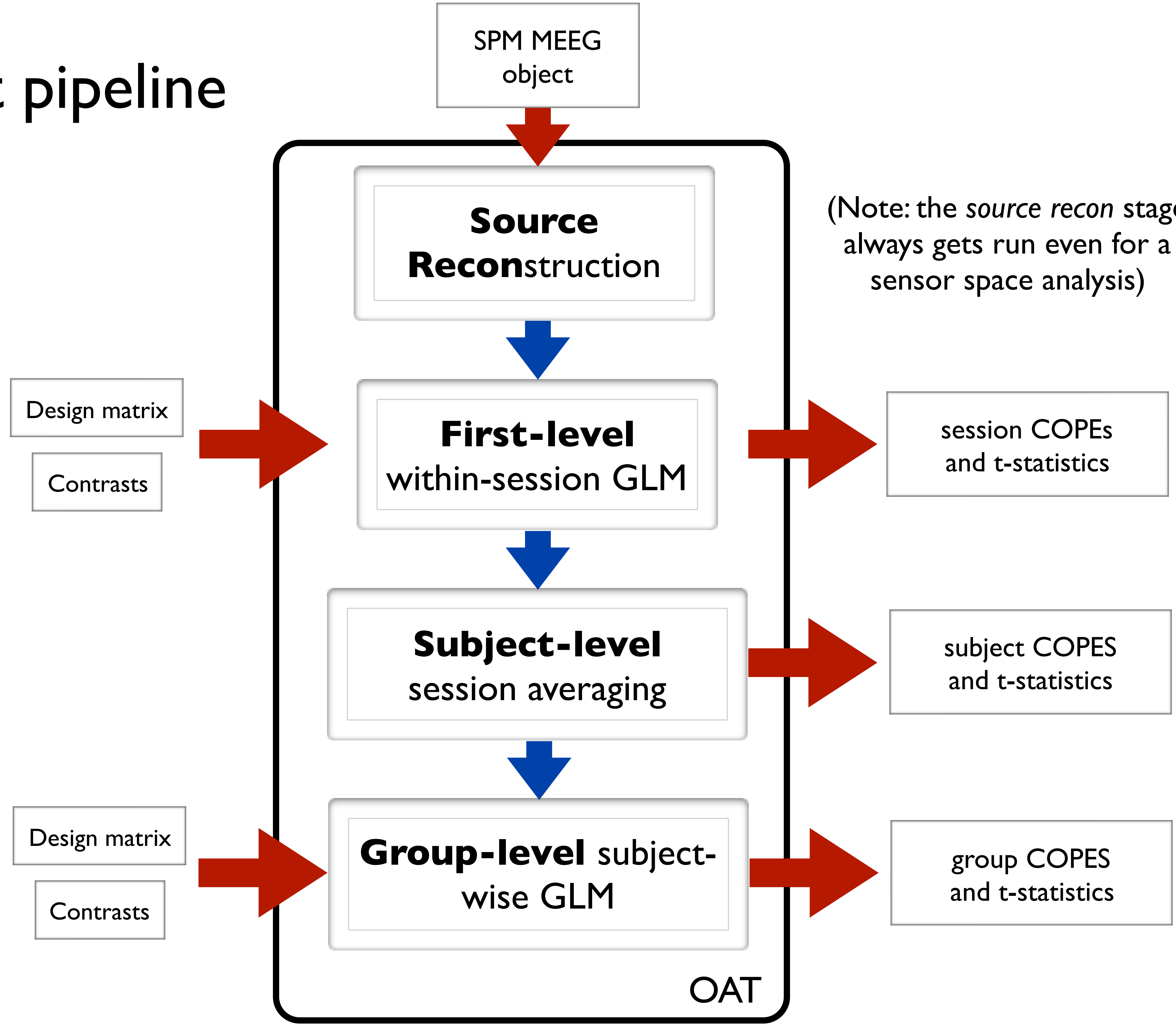
- **Task-based** analysis in:
  - ➡ sensor space, or
  - ➡ source space (e.g. via beamforming)
- In:
  - ➡ time domain (e.g. ERF-style), or
  - ➡ in time-frequency domain (e.g. induced responses)

OAT - OSL's Analysis Tool

- **Task-based** analysis in:
  - ➡ sensor space, or
  - ➡ source space (e.g. via beamforming)
- In:
  - ➡ time domain (e.g. ERF-style), or
  - ➡ in time-frequency domain (e.g. induced responses)
- First-level (within-session) analysis, using:
  - ➡ trial-wise GLM on epoched data
  - ➡ time-wise GLM on continuous data
- Group-level (between-subject) subject-wise GLM analysis

OAT Pipeline Stages

- 4 distinct pipeline stages:

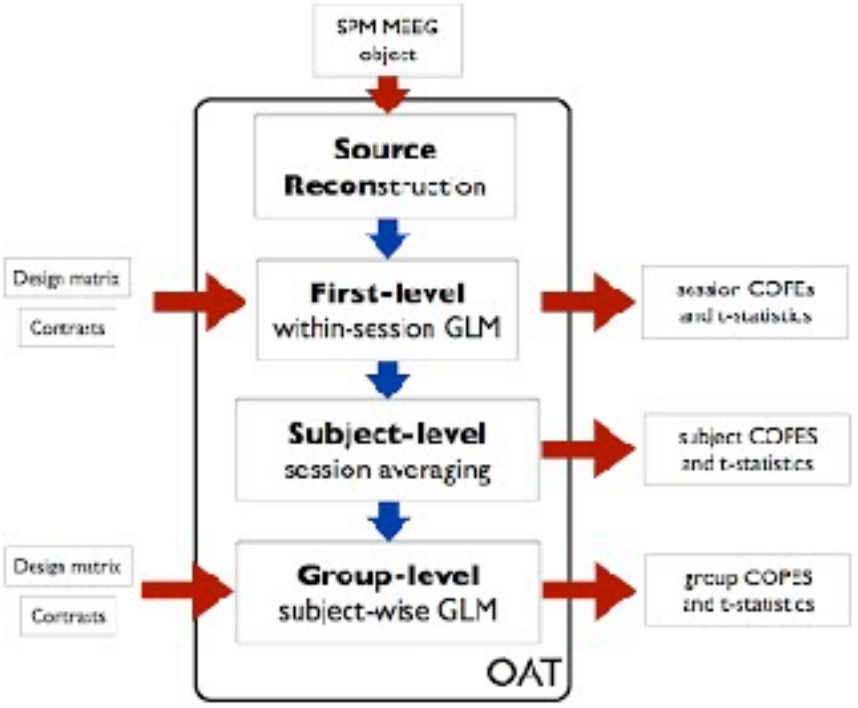


OAT Setup

- Set some mandatory fields, and then use osl_check_oat call to setup an OAT struct:
  - ➡ *oat= osl_check_oat(oat);*

OAT Setup

- Set some mandatory fields, and then use osl_check_oat call to setup an OAT struct:
  - ➡ *oat= osl_check_oat(oat);*
- Settings are organised by the 4 distinct stages of the pipeline:
  - ➡ *oat.source_recon, e.g.*
  - ➡ *oat.first_level* (GLM within-session analysis)
  - ➡ *oat.subject_level* (within-subject averaging)
  - ➡ *oat.group_level* (GLM subject-wise analysis)



OAT Setup

- The *oat* gets stored in the directory specified in *oat.source_recon.dirname*, with a '*.oat*' suffix
- A previously setup/run *oat* can be loaded into Matlab with:
  - *oat.source_recon.dirname='/path/oatname';*
  - *oat=osl_load_oat(oat);*

Some oat.first_level settings

- Set time range and freq range using:
  - *oat.first_level.time_range = [-1 2] % secs around stimulus onset*
  - *oat.first_level.tf_freq_range = [1 45] % Hz*

Some oat.first_level settings

- Set time range and freq range using:
  - *oat.first_level.time_range = [-1 2] % secs around stimulus onset*
  - *oat.first_level.tf_freq_range = [1 45] % Hz*

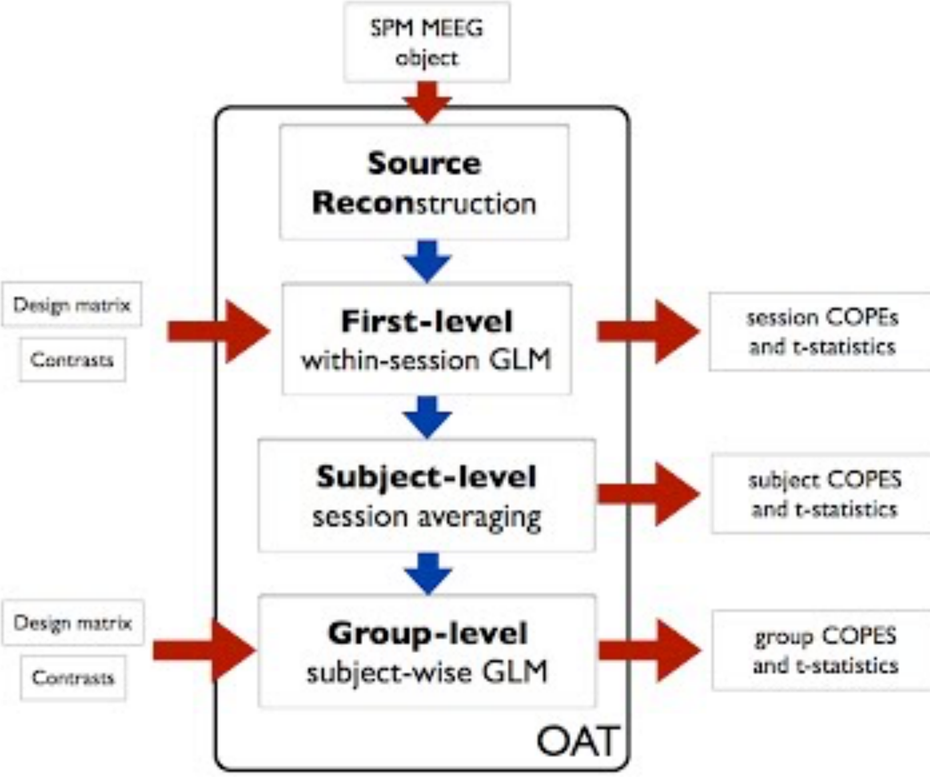
- To do an ERF analysis set *oat.first_level.tf_method='none'*
- To do a Time-Frequency (TF) induced response analysis set *oat.first_level.tf_method='hilbert' % or 'morlet'*

Running OAT

- Use osl_run_oat to run an OAT:
  - ➡ *oat=osl_run_oat(oat);*

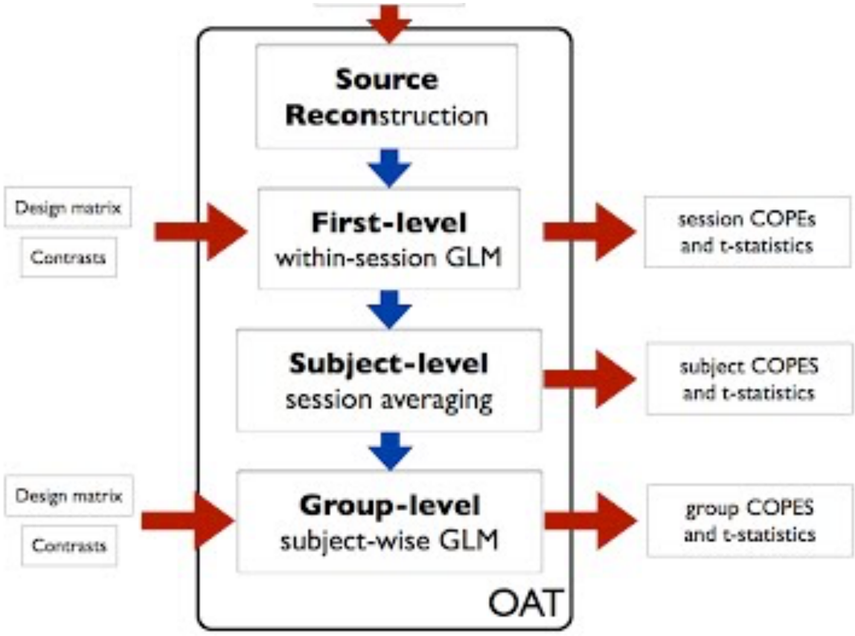
Running OAT

- Use osl_run_oat to run an OAT:
  - ➡ *oat=osl_run_oat(oat);*
- This only runs the stages specified in ***oat.to_do***, e.g.:
  - ➡ *oat.to_do=[1 1 0 0];* only runs source_recon and first-level stages



OAT output

- After running, the *oat* struct contains filenames of the outputs for each stage of the pipeline:
  - ➡ *oat.source_recon.results_fnames*
  - ➡ *oat.first_level.results_fnames*
  - ➡ *oat.subject_level.results_fnames*
  - ➡ *oat.group_level.results_fnames*

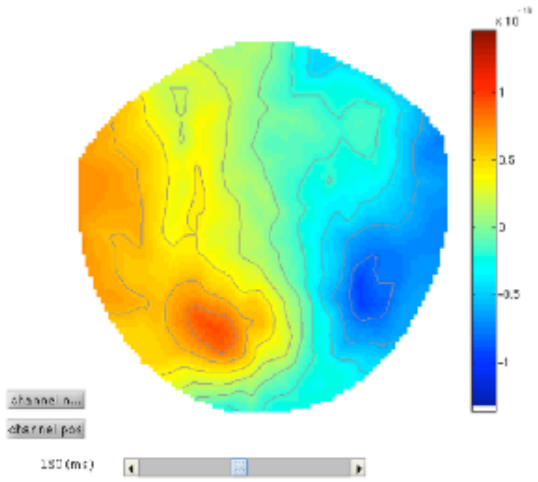


- *These can be loaded into Matlab, e.g. to load session 2's first level results use the call:*
  - ➡ *res=osl_load_oat_results(oat, oat.first_level.results_fnames{2})*

Viewing OAT output

- It is highly recommended that you inspect *oat.results.report* (an HTML page), to ensure that OAT has run successfully (See the practical)

- In sensor space, use:
  - Use *osl_stats_multiplotER* and *osl_stats_multiplotTFR* to call Fieldtrip interactive topoplots
  - The two orientations of the *MEGPLANARs* are combined (in the first_level stage) by rectifying and adding



Practical

Sensor space trial-wise GLM using OAT on **epoched** task data:

a) Time-domain (ERF) analysis

b) Time-frequency (induced response) analysis