BCI practical course : Signal Processing

Jason Farquhar (coordinator)
<J.Farquhar@donders.ru.nl>
Philip van den Broek (Brainstream developer/tutorials)
<p.vandenbroek@donders.ru.nl>
Alex Brandmeyer (Teaching Assistant)
<A.Brandmeyer@donders.ru.nl>

Learning Goals : Last Time

- Understand:
  - why precise stimulus timing requirements mean we need to run the stimulus generation in a separate process/thread
- Know how to:
  - Specify some of your table to execute in a 2nd (or 3rd..) MATLAB process
  - Use a simple MATLAB only `runLoop` for more precise stimulus timing
    - With hardware/buffer markers to indicate stimulus events to other Brainstream processes
  - Use a Psychtoolbox based `runLoop` for more precise auditory/visual stimulus generation and timing
  - Use Stimbox2 as a Brainstream integerated Psychtoolbox based stimulus generation system

Today's Plan

- Review :Advanced Stimulus Presentation. Solutions and discussion of problems
- Lecture : Simple signal processing

Break

- Hands-on : ERSP classification – data gathering
- Hands-on : ERSP classification – analysis

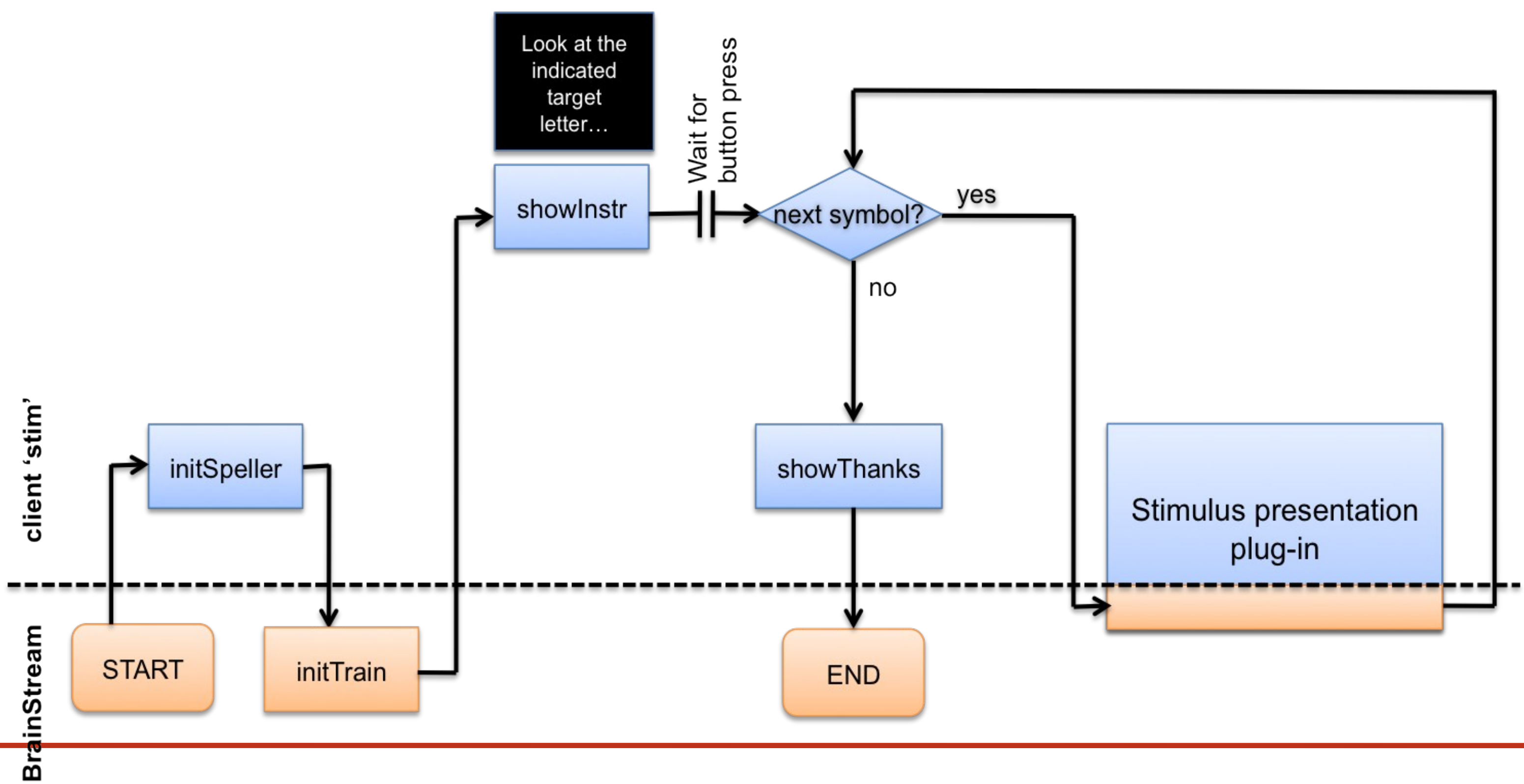
Break

- Hands-on : ERP classification – data gathering
- Hands-on : ERP classification – analysis

Hands On : Parallel stimulus presentation

Task

- Modify the simple speller example such that the stimulus presentation is run in a client process called 'stim'



Hands on: Using a runLoop

Task

- Write a simple loop to do one sequence of stimulus presentation with the visual speller
- Test if this improves the timing quality.

Hands on : Psychtoolbox

Task:

- Re-write your `runLoop` based visual speller BCI to use Psychtoolbox drawing functions
- Compare the timing performance of this version with the table-based, and runLoop based versions

Summary – Advanced stimulus presentation

- Parallel execution means we can (better) guarantee stimulus timing
- Parallel client can be based on (in order of increasing programing effort?):
  - BS tables – minimal code, (+/- 20ms)
  - matlab runLoop (+/-20ms)
  - Psychtoolbox runLoop (+/- 1ms)
  - StimBox2 (+/- 2ms)

Discussion

- Which approach did you prefer?
- Which was simplist to implement and test?
- Which was most flexible/clear?
- Which was most timing accurate?
- Which would/will you use for your project?

Simple Decoding

- We have used the functions:
  - `bs_train_erp_clsfr`, `apply_erp_clsfr` – to train and apply an ERP (i.e. evoked response) signal decoder
  - `bs_train_ersp_clsfr`, `apply_ersp_clsfr` – to train and apply an ERSP (i.e. Induced response) signal decoder
- Internally, these functions apply a simple 6-step default pre-processing, feature selection and classifier training/application procedure

Simple Decoding Steps

1) Detrending
2) Bad-channel identify and remove
3) Re-referencing/Spatial Filtering

ERP

4) Spectral Filtering
5) Classifier Training

ERSP

4) Feature extraction
5) Feature selection
6) Classifier Training

Key functions

- `clsfr=train_erp/ersp_clsfr(X,Y,...)`
  - train a linear classifier on the frequency power spectrum of the data
  - X - [ch x time x epochs] raw EEG data
  - Y - [epochs x 1] target labels for each epoch
  - `... –` lots of name,value option pairs to control the type of pre-processing/feature extraction done at each step.

    (To get information on the options available: >> `help train_erp_clsfr`)
  - `clsfr` - trained classifier structure

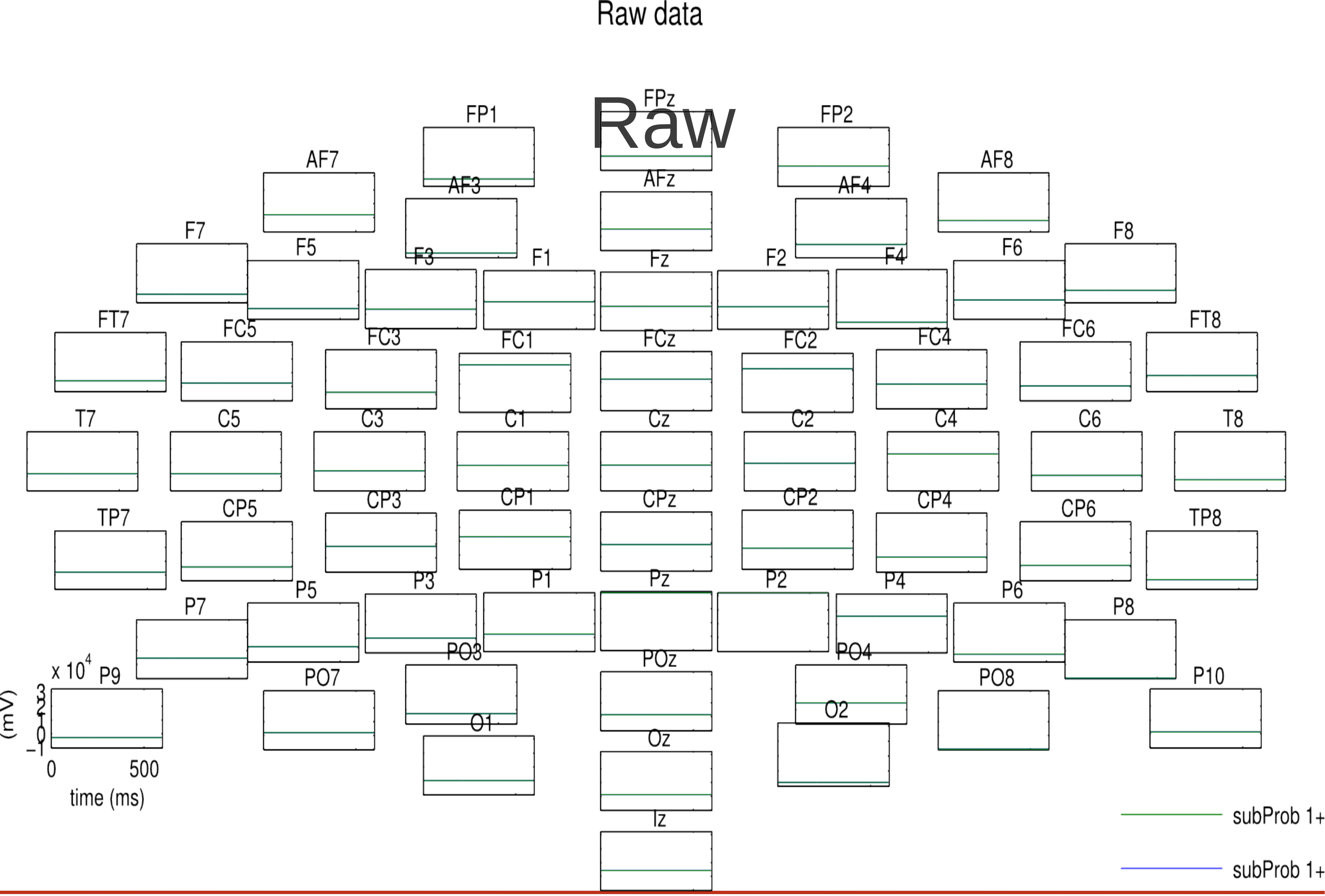
Key functions

- `[f,fraw,p]=apply_ersp/erp_clsfr(clsfr,X)`
  - Apply the trained pre-processing and classifier to the new data X
  - X - [ch x time x epochs] raw EEG data
  - f - [epochs x nCls] classifier decision value for each input epoch
  - fraw, p – [epochs x nSp] raw decision values/probabilities for each sub-problem

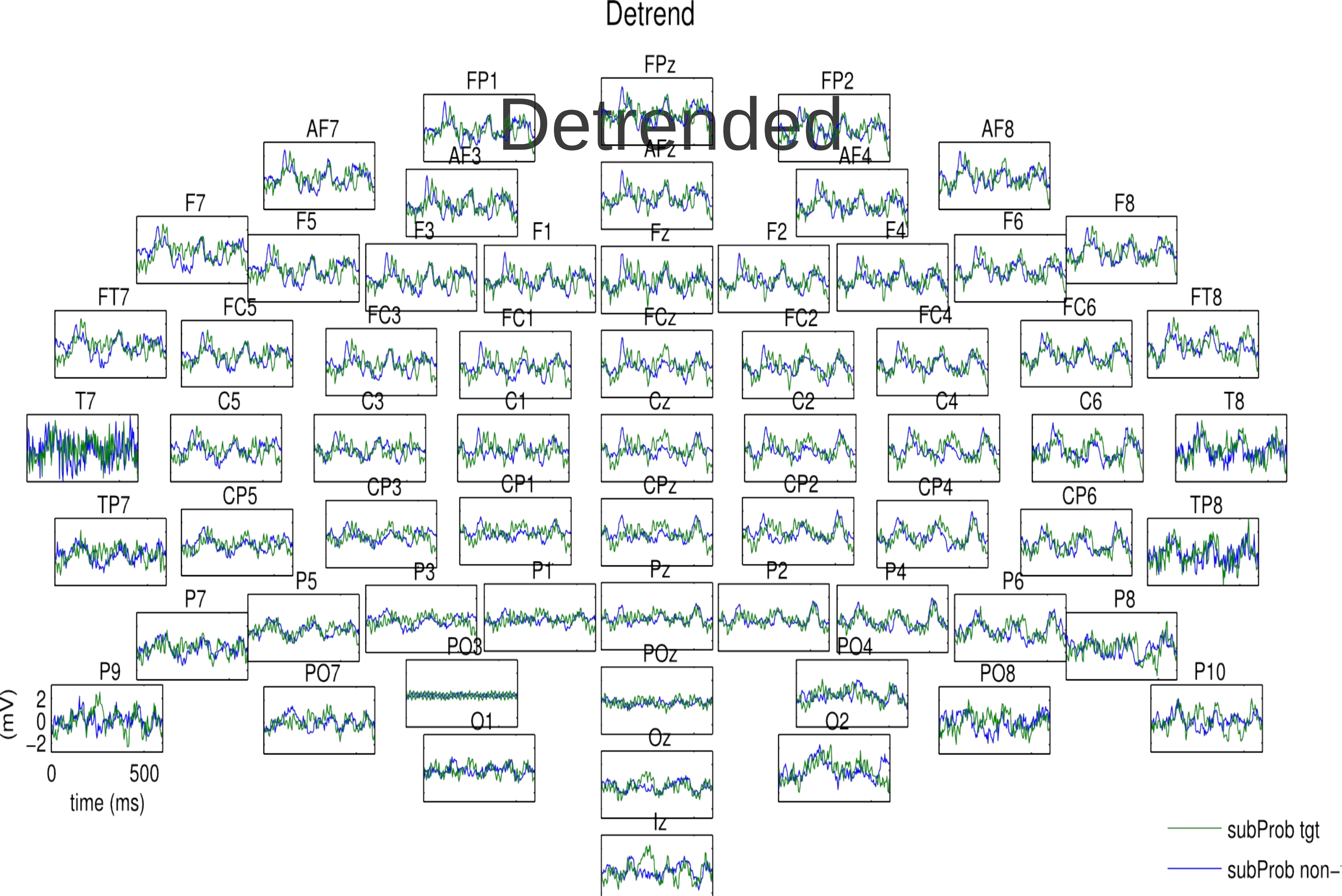
1) Detrending

- Why?
  - Remove slow-drifts and arbitary offsets in EEG data
- How?
  - Compute and subtract linear trends for each channel and epoch

Raw


Raw data
FP1
FPz
FP2
AF7
AF3
AFz
AF4
AF8
F7
F5
F3
F1
Fz
F2
F4
F6
F8
FT7
FC5
FC3
FC1
FCz
FC2
FC4
FC6
FT8
T7
C5
C3
C1
Cz
C2
C4
C6
T8
TP7
CP5
CP3
CP1
CPz
CP2
CP4
CP6
TP8
P7
P5
P3
P1
Pz
P2
P4
P6
P8
x 10^4
P9
PO7
PO3
POz
PO4
PO8
P10
O1
Oz
O2
Iz
(mV)
3
2
1
0
−1
0
500
time (ms)
subProb 1+
subProb 1+

Detrended

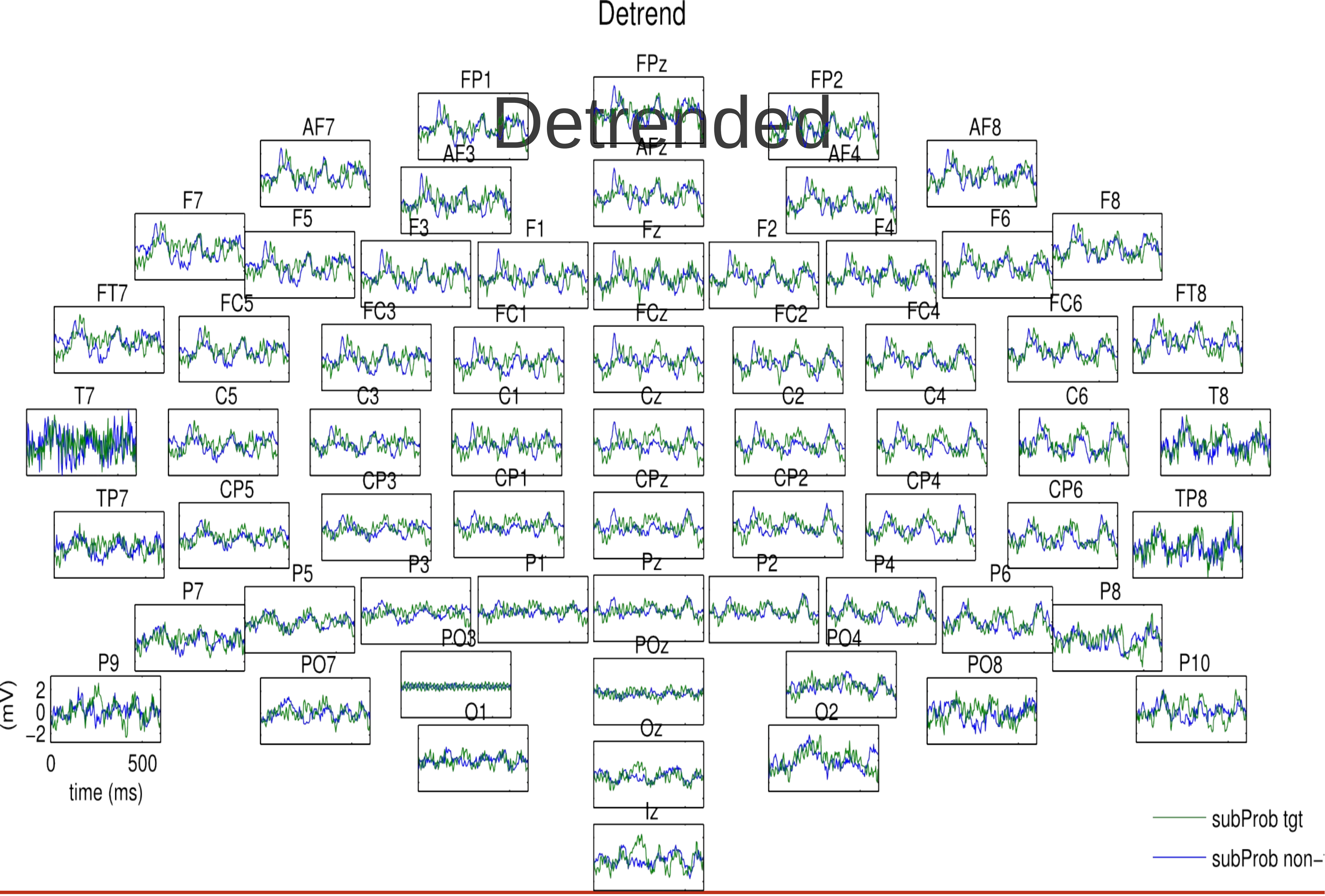


Detrend
FPz
FP1
FP2
AF7
AF3
AFz
AF4
AF8
F7
F5
F3
F1
Fz
F2
F4
F6
F8
FT7
FC5
FC3
FC1
FCz
FC2
FC4
FC6
FT8
T7
C5
C3
C1
Cz
C2
C4
C6
T8
TP7
CP5
CP3
CP1
CPz
CP2
CP4
CP6
TP8
P7
P5
P3
P1
Pz
P2
P4
P6
P8
P9
PO7
PO3
POz
PO4
PO8
P10
O1
Oz
O2
Iz
(mV)
2
0
−2
0
500
time (ms)
subProb tgt
subProb non−

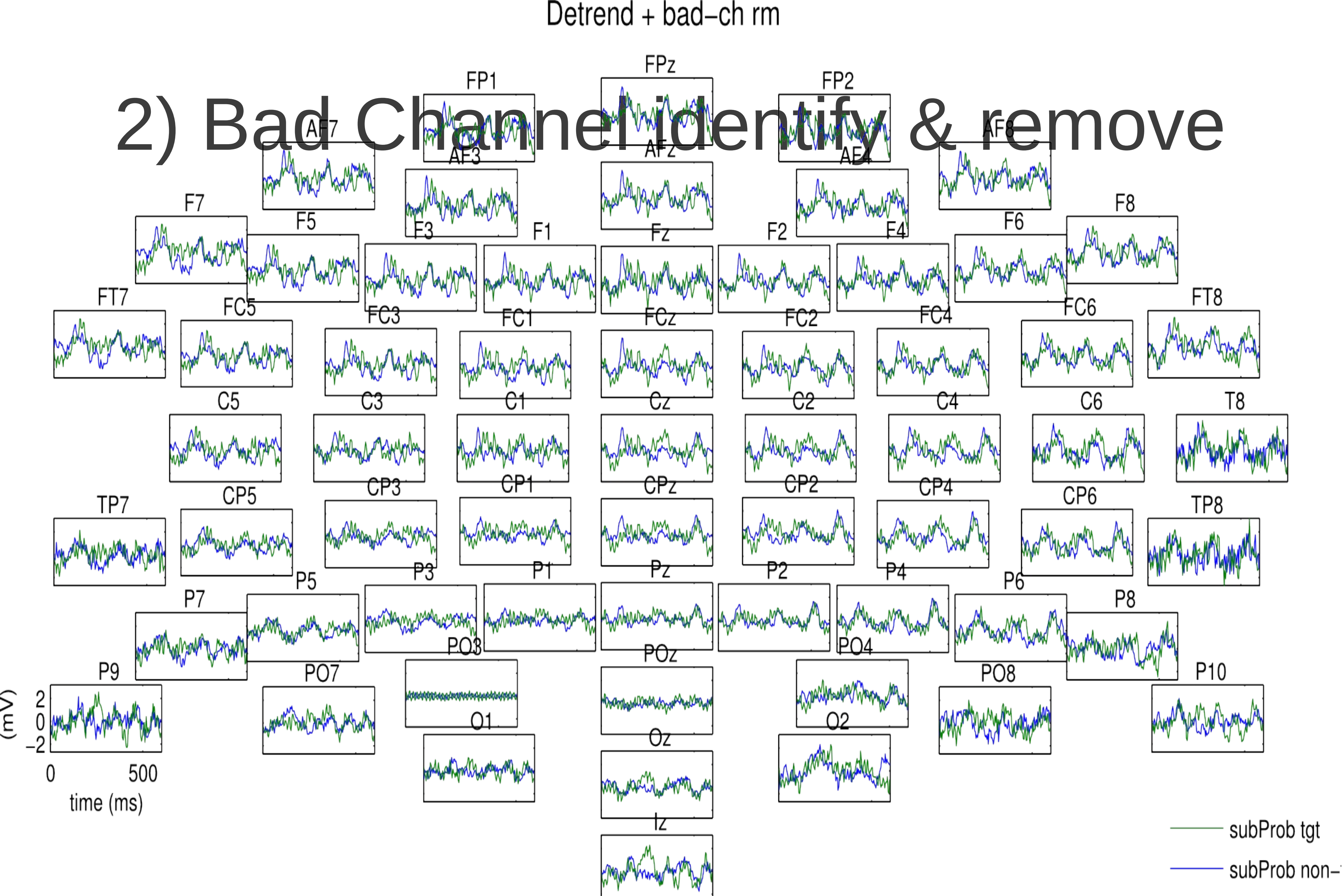
2) Bad channel identify and remove

- Why?
  - Some channels don't connect well so only pickup noise. Leaving them in messes up the reset when spatial filtering.
- How?
  - Identify channels with excessively high power.
  - Specificially:
    - compute total power for each channel over all epochs
    - Compute mean channel power and variance in channel power
    - Remove any channels with power more than 3 std-dev from mean

Detrended


Detrend
FPz
FP1
FP2
AF7
AF3
AFz
AF4
AF8
F7
F5
F3
F1
Fz
F2
F4
F6
F8
FT7
FC5
FC3
FC1
FCz
FC2
FC4
FC6
FT8
T7
C5
C3
C1
Cz
C2
C4
C6
T8
TP7
CP5
CP3
CP1
CPz
CP2
CP4
CP6
TP8
P7
P5
P3
P1
Pz
P2
P4
P6
P8
P9
PO7
PO3
POz
PO4
PO8
P10
O1
Oz
O2
Iz
(mV)
2
0
−2
0
500
time (ms)
subProb tgt
subProb non−

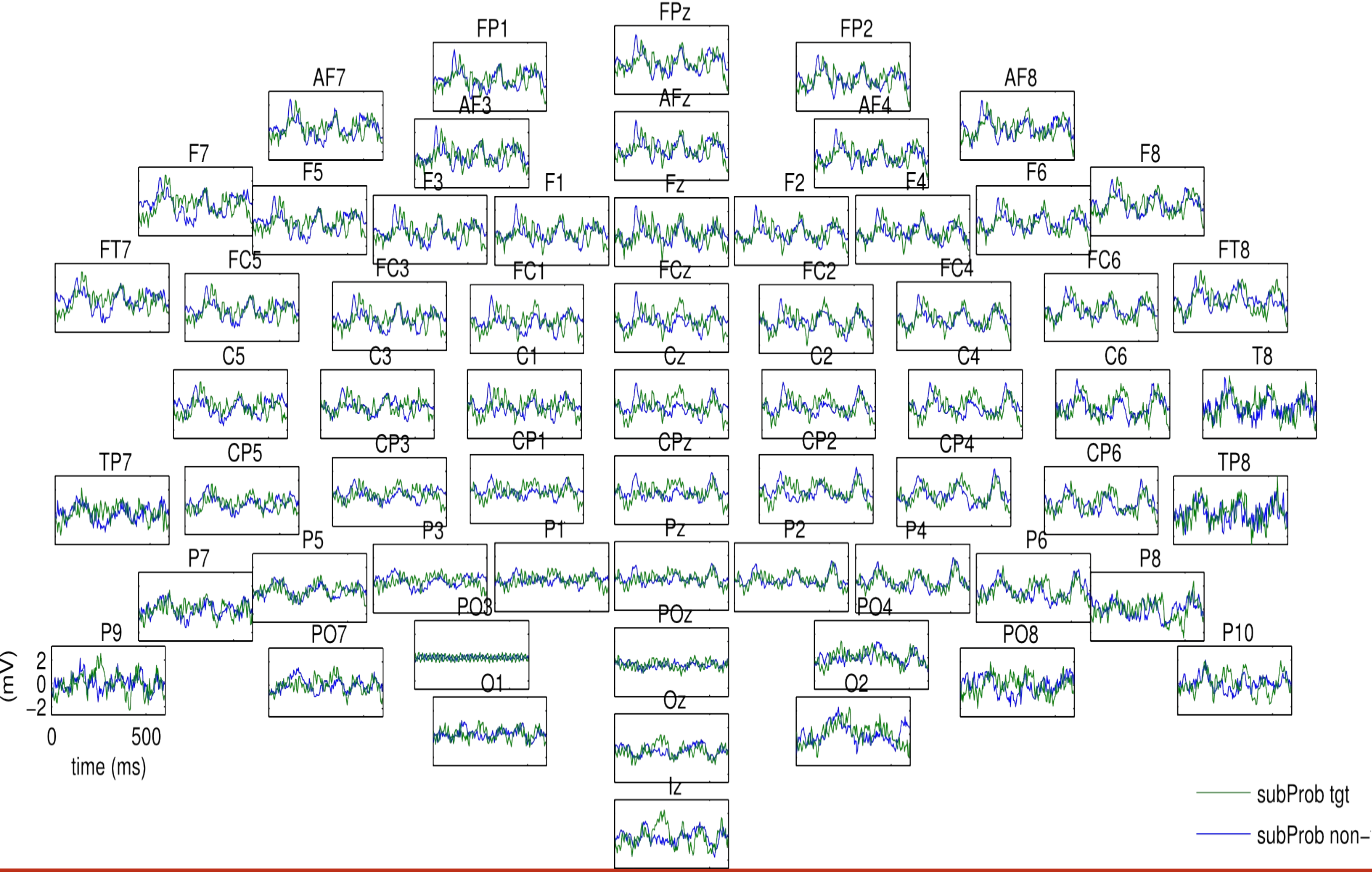
2) Bad Channel identify & remove



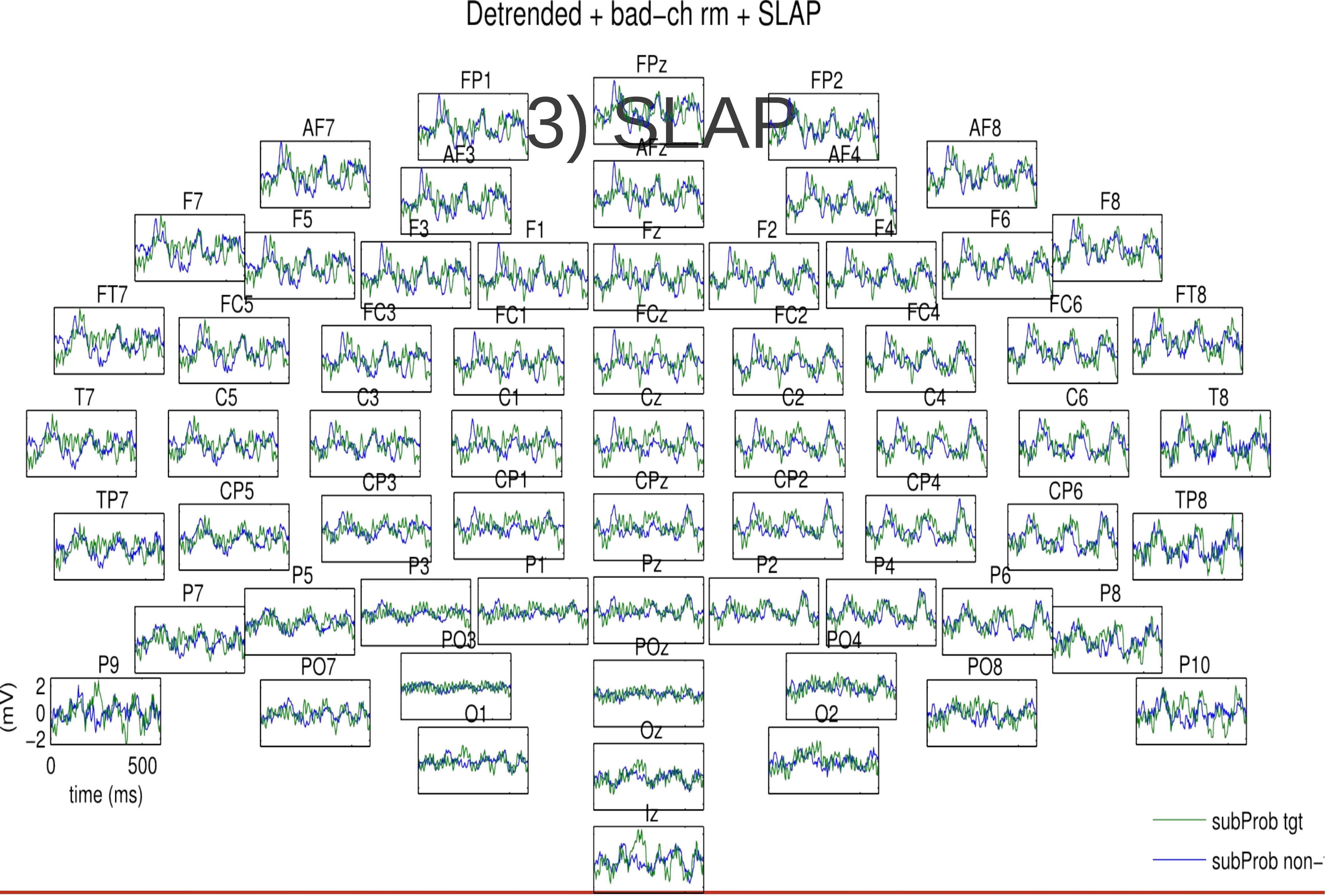
3) Spatial Filter

- Why?
  - EEG contains lots of signal from external noise sources – which is visible as a signal common to all channels
  - Volume conduction means nearby channels have high correlation
- How?
  - Common Average Reference (CAR) -- Remove common external signal by subtracting the average signal over channels from all channels
  - Surface Laplacian – Remove channel correlation (and common signal!) subtracting a local average signal (default)
  - Spatial Whiten – Remove channel correlation (and common signal!) using PCA to map to a co-ordinate set where channels are uncorrelated.

Detrend + bad-ch rm



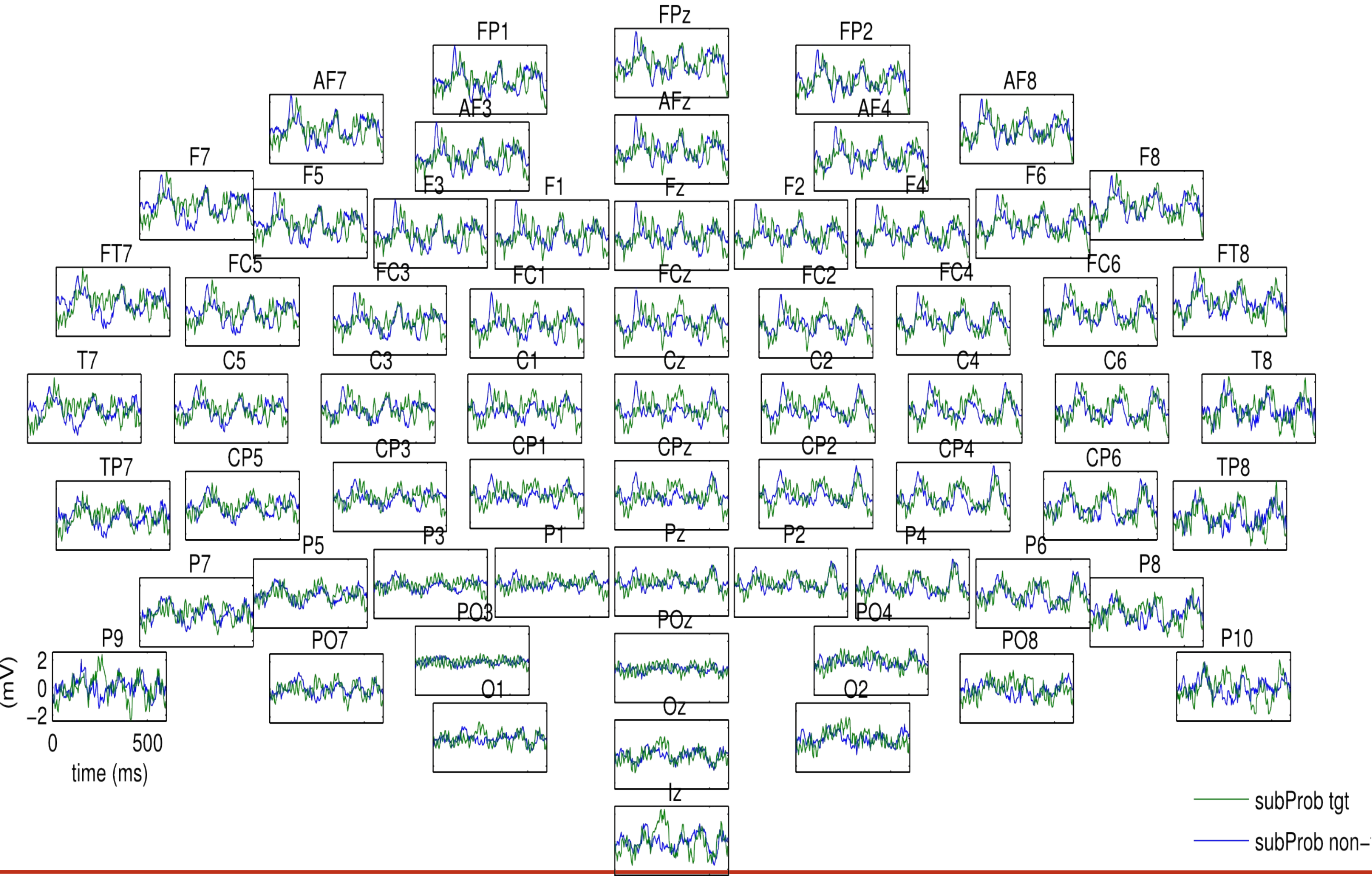
3) SLAP


Detrended + bad–ch rm + SLAP
FP1
FPz
FP2
AF7
AF3
AFz
AF4
AF8
F7
F5
F3
F1
Fz
F2
F4
F6
F8
FT7
FC5
FC3
FC1
FCz
FC2
FC4
FC6
FT8
T7
C5
C3
C1
Cz
C2
C4
C6
T8
TP7
CP5
CP3
CP1
CPz
CP2
CP4
CP6
TP8
P7
P5
P3
P1
Pz
P2
P4
P6
P8
P9
PO7
PO3
POz
PO4
PO8
P10
O1
Oz
O2
Iz
(mV)
2
0
−2
0
500
time (ms)
subProb tgt
subProb non–

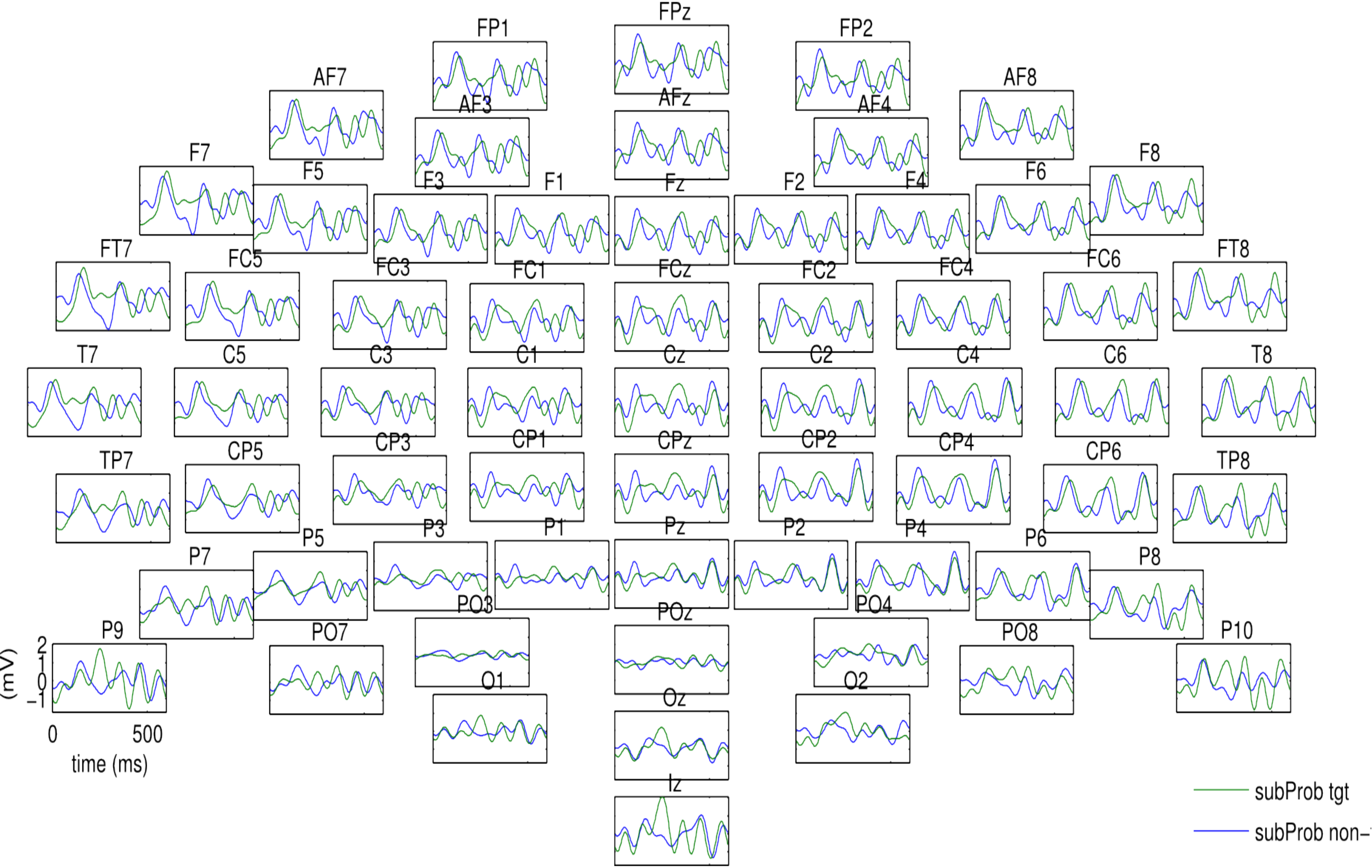
4) ERP Spectrally Filter

- Why?
  - Our signal of interest only occurs in a particular frequency range
  - Thus, any signal which occurs outside this frequency range **must** be noise, and should be removed
- How?
  - Apply a spectral filter to remove frequencies outside the range of interest

Detrended + bad–ch rm + SLAP



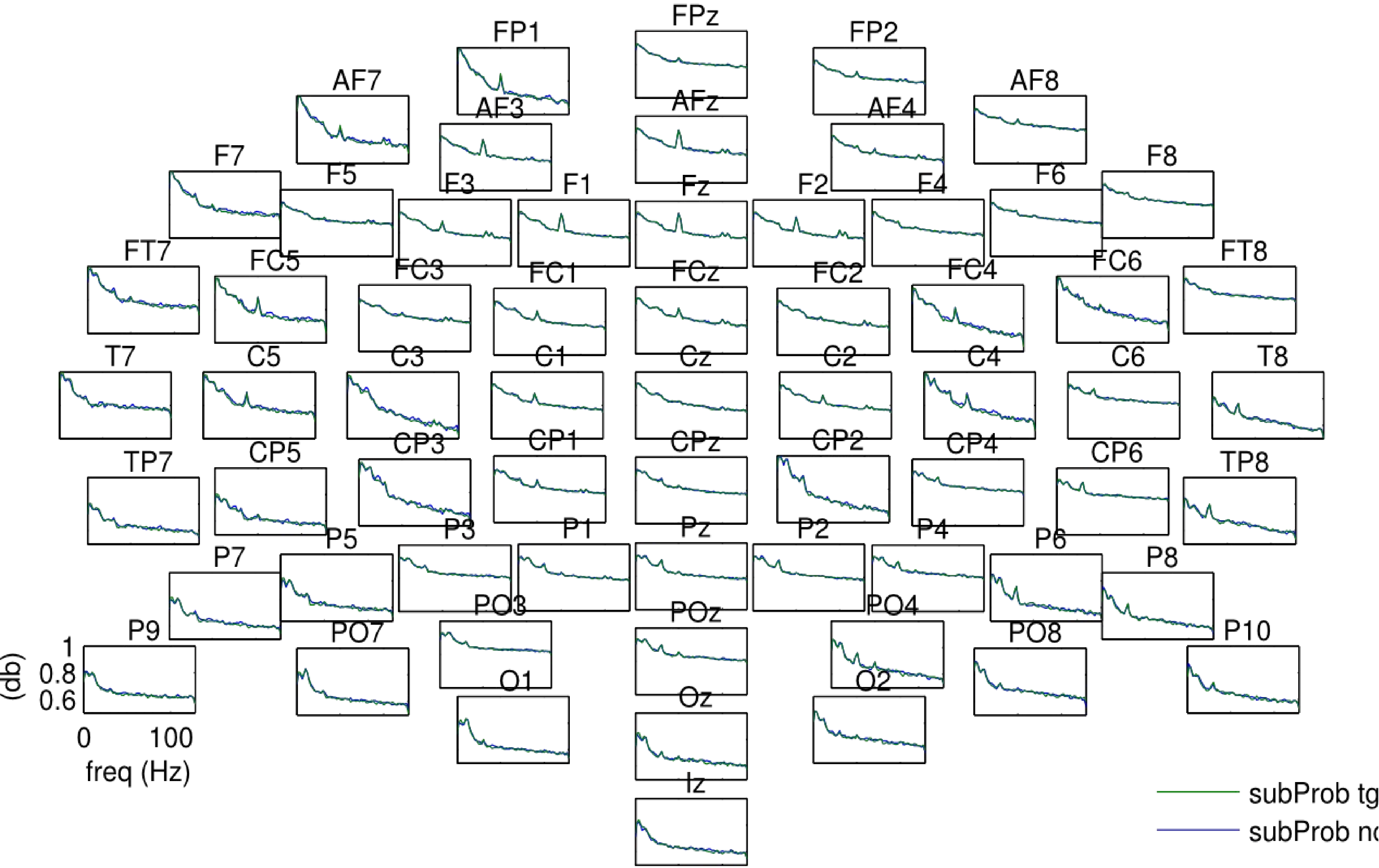
Detrended + bad–ch rm + SLAP + spectral filter



4) ERSP Feature Extraction

- Why?
  - Our signal of interest is **not** time locked, but a change in power in a particular frequency range
  - Thus, it cannot be detected from the raw time-domain features by a linear classifier
- How?
  - Use Welch's method to compute a high quality estimate of the signals power spectrum for each epoch, i.e. power in each frequency bin
  - N.B. To make the distribution of powers more Gaussian distributed, we use **log** power as the classification feature

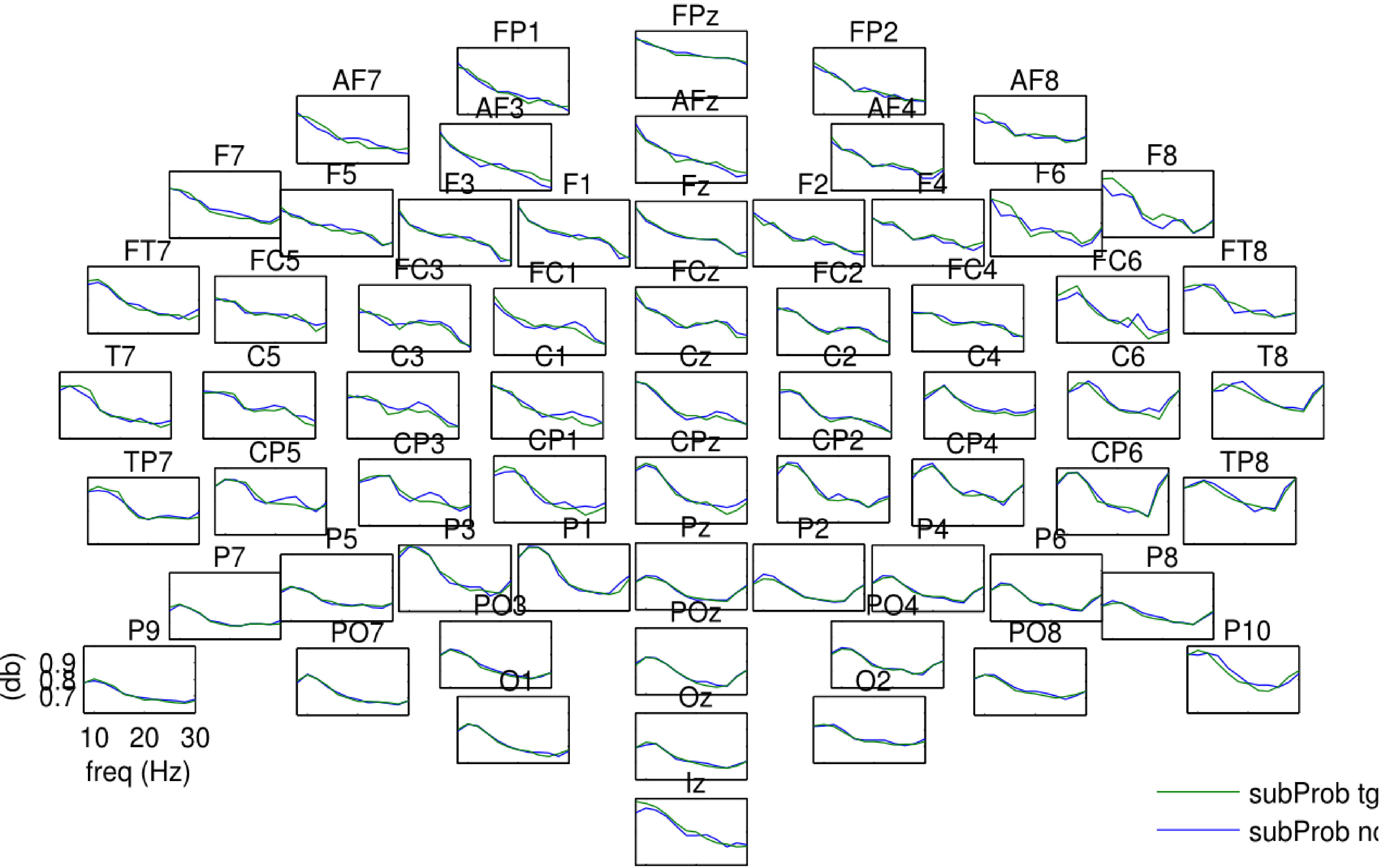
Detrended + bad–ch rm + SLAP + welch



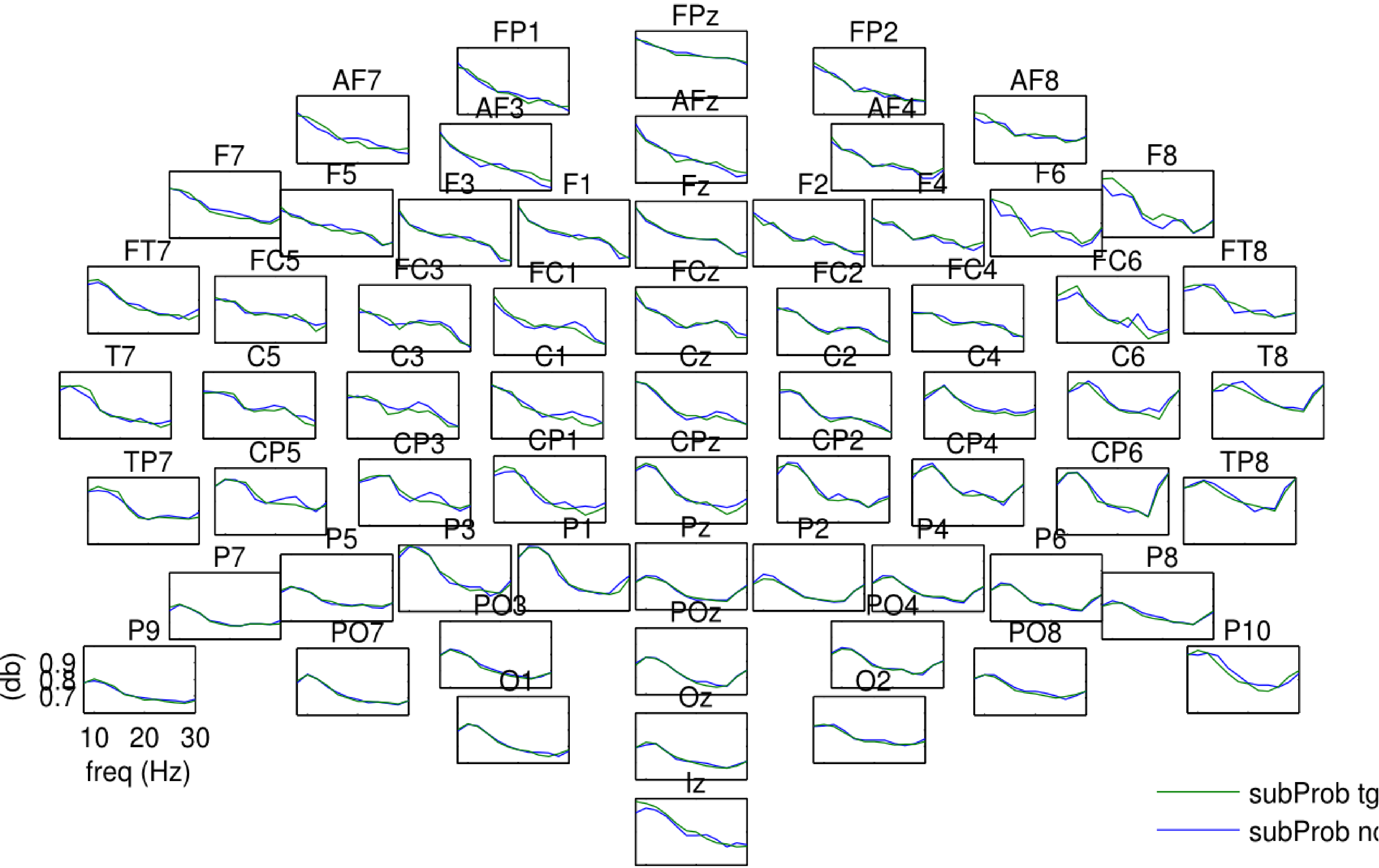
5) ERSP Feature Selection

- Why?
  - Our signal of interest only occurs in a particular frequency range
  - Thus, any signal which occurs outside this frequency range **must** be noise, and should be removed
- How?
  - Discard frequency bins from the power spectrum which occur outside the range of interest

Detrended + bad–ch rm + SLAP + welch + feat seln



Detrended + bad–ch rm + SLAP + welch + feat seln



5/6) Train Classifier

- Why?
  - Account for high variability in signal and noise properties, by building a special purpose detector for this subject on this day.
- How?
  - Train a linear logistic regression classifier with quadratic regularisation on the gathered examples
  - With 10-fold cross validation to select the optimal regularisation strength
  - N.B. Available alternative classifiers:
    - Linear Support Vector Machine: 'objFn','l2svm_cg'
    - Regularised Linear Discriminants: 'objFn','rkls'

Detrended + bad–ch rm + SLAP + spectral filter

Classifier Output

Regularisation Strength

High ... low

		High					low
outer fold, (all data)	(out)	0.51/NA	0.93/NA	0.95/NA	0.95/NA	0.95/NA	0.95/NA
Folds	(1)	0.50/0.50	0.93/0.91	0.95/0.95	0.95/0.95	0.95/0.95	0.95/0.95
	(2)	0.50/0.50	0.93/0.92	0.95/0.94	0.95/0.94	0.96/0.94	0.96/0.94
	(3)	0.50/0.50	0.92/0.96	0.95/0.99	0.95/0.99	0.95/1.00	0.95/1.00
	(4)	0.50/0.50	0.93/0.93	0.95/0.97	0.95/0.97	0.95/0.98	0.95/0.98
	(5)	0.50/0.50	0.92/0.93	0.96/0.94	0.95/0.93	0.96/0.93	0.96/0.93
	(6)	0.50/0.50	0.93/0.91	0.95/0.96	0.95/0.97	0.95/0.97	0.95/0.97
	(7)	0.50/0.50	0.92/0.94	0.95/0.96	0.95/0.96	0.95/0.96	0.95/0.96
	(8)	0.50/0.50	0.93/0.91	0.96/0.94	0.95/0.94	0.95/0.94	0.95/0.94
	(9)	0.50/0.50	0.92/0.94	0.95/0.98	0.95/0.97	0.95/0.97	0.95/0.97
	(10)	0.50/0.50	0.94/0.89	0.96/0.89	0.96/0.89	0.96/0.89	0.96/0.89

	(ave)	0.50/0.50	0.93/0.92	0.95/0.95	0.95/0.95	0.95/0.95	0.95/0.95

Training Performance

Testing Performance

Hands on : ERSP classification – data gathering

- Task
  - Get some ERSP data to compare classification pipelines on.
  - Use the imaginedmovement_ans example to gather some brain data

Hands on : ERSP classification – decoding comparsion

- Task
  - Use the training data saved by BS to test and compare different decoding pipeline options
    - Performance when we use the ERP classification pipeline?
    - Effect of feature selection range
    - Effect of spatial filter?

Discussion: ERSP Classification

- So, what matters what doesnt?

Hands on : ERP classification – data gathering

- Task
  - Get some ERP data to compare classification pipelines on.
  - Use the simplespeller_client_ans example training block to gather some brain data

Hands on : ERP classification – decoding comparsion

- Task
  - Use the training data saved by BS to test and compare different decoding pipeline options:
    - Effect of detrending
    - Effect of spectral filter range
    - Effect of different spatial filters

Discussion: ERP Classification

- So, what matters what doesnt?

Summary : Decoding

- 5/6 Steps sufficient for simple ERP/ERSP BCIs
  - Detrend
  - Identify and remove bad channels
  - Spatial filter / reference
  - Spectral filter (ERP)
  - Compute power spectrum and select frequencies (ERSP)
  - Regularised linear classifier training