



# Multivariate analysis

Romain Grandchamp, PhD  
Arnaud Delorme, PhD



# From ERPs to MVPA Using the Amsterdam Decoding and Modeling Toolbox (ADAM)

Johannes J. Fahrenfort<sup>1,2,3\*</sup>, Joram van Driel<sup>1</sup>, Simon van Gaal<sup>2,3</sup> and Christian N. L. Olivers<sup>1</sup>

<sup>1</sup> Department of Experimental and Applied Psychology, Institute Brain and Behavior Amsterdam (iBBA), VU University Amsterdam, Amsterdam, Netherlands, <sup>2</sup> Department of Psychology, University of Amsterdam, Amsterdam, Netherlands,

<sup>3</sup> Amsterdam Brain and Cognition (ABC), University of Amsterdam, Amsterdam, Netherlands

# Adam Features

- **Decoding (backward decoding, BDM):** Classifiers can be trained to discriminate between different experimental conditions based on EEG/MEG patterns.
- **Encoding modeling (forward encoding, FEM):** Allows reconstruction of “Channel Tuning Functions” (CTFs) or spatial maps of transformed weights (using the Haufe method) from the encoding model.
- **Temporal and generalization analyses (temporal generalization):** ADAM can compute “train vs. test at different times” matrices to examine the stability of neural representations over time.
- **Time–frequency domain decoding:** Before decoding, one can apply a time–frequency decomposition (TFR) and then perform decoding within separate frequency bands.
- **Statistics and multiple-comparison correction:** ADAM provides permutation testing, FDR correction, and group-level analyses to assess the statistical significance of results.
- **Visualization and publication-ready figures:** Generation of classifier performance time courses, topographical maps of weights, channel tuning functions, generalization matrices, and more.
- **Automatic control of potential biases (balanced design, AUC computation, etc.):** To prevent biases in the analyses, ADAM applies several default controls, such as balancing training classes and using AUC instead of raw accuracy.

*Import and pre-process*  
(not part of ADAM)

Import native EEG or MEG data into EEGLAB or FieldTrip format, pre-process, e.g. highpass filter, epoching, artefact rejection. Baseline correction and muscle artefact rejection can also be applied by ADAM during first-level analysis.

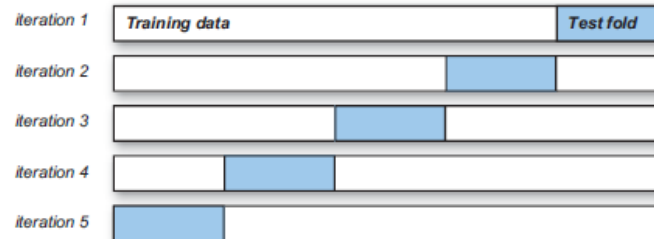
*adam\_MVPA\_firstlevel* In: Epochs files in either EEGLAB or FieldTrip format  
Out: ADAM result files (one for each subject), containing a performance metric for every train-test time sample (raw) or for every train-test sample of every frequency band (tfr)

Use RAW data

Compute time-frequency representations

For every time point, build a backward decoding model (BDM) or forward encoding model (FEM) using training data, and compute performance metric on testing data. Weights of BDMs are forward transformed.

*Option 1: K-fold cross-validation. Requires a single data file per subject.*



The final performance metric is computed by averaging over test folds (in this example,  $K=5$ ).

*Option 2: Requires separate data sets for training and testing (either using separate files or separate event values for train and test data)*

Training data

Testing data

The performance metric is computed over the testing data.

Several transformations can be performed on the training and testing data, e.g. binning, whitening, computing induced power, etc. These transformations are either performed separately on training and testing data, or they are performed indiscriminately across all stimulus classes.

*adam\_compute\_group\_MVPA* In: ADAM result files computed by *adam\_MVPA\_firstlevel*  
*adam\_compute\_group\_ERP* Out: ADAM stats variable(s) containing group statistics

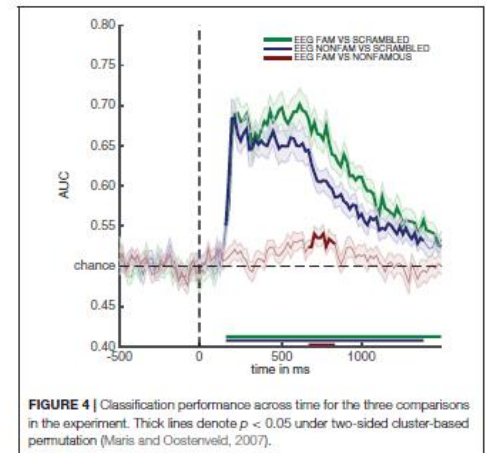
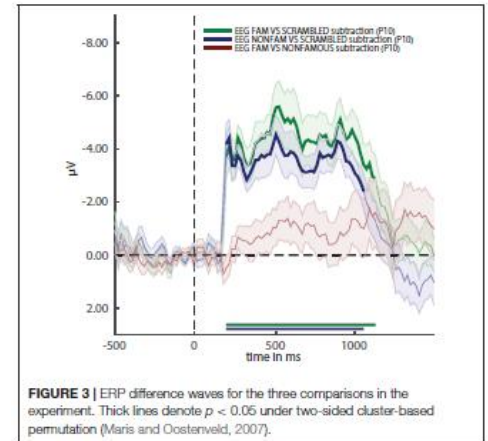
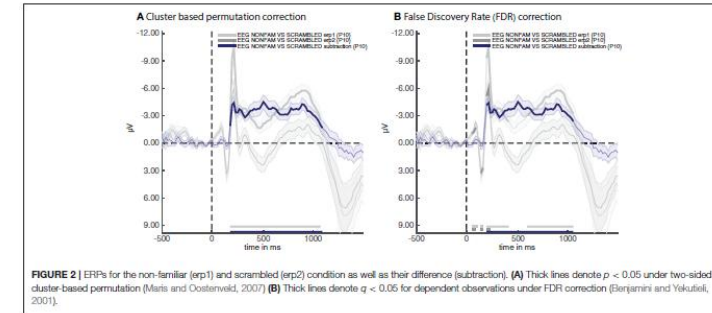
*adam\_compare\_MVPA\_stats* In: ADAM stats variable(s) containing group statistics  
Out: ADAM stats variable(s) containing group statistics

*adam\_plot\_MVPA* In: ADAM stats variable(s) containing group statistics  
*adam\_plot\_BDM\_weights* Out: publication-ready graphs of performance metrics and/or topographical maps of forward transformed weights

**FIGURE 1 |** ADAM processing pipeline, from top to bottom. The top left corner of each box states the ADAM function that performs the transformations that are described in the box. The top right describes the input-output transformation that the function performs. The output of each function serves as input for the function described in the box below it. The *adam\_MVPA\_firstlevel* box contains more detailed information about train-test algorithms. Further details about functionality and how to execute functions are provided in the main text.

# Main functions

- `adam_MVPA_firstlevel` (computes and stores first level / single subject results)
- `adam_compute_group_ERP` (reads single subject ERPs and computes group statistics which can be plotted using `adam_plot_MVPA`)
- `adam_compute_group_MVPA` (reads single subject classification performance and computes group statistics which can be plotted using `adam_plot_MVPA`)
- `adam_compare_MVPA_stats` (compares outcomes of group analyses, which can be plotted using `adam_plot_MVPA`)
- `adam_plot_MVPA` (plots the outcome of the `adam_compute_group_` or the `adam_compare_MVPA_stats` functions)
- `adam_plot_BDM_weights` (plots topomaps of the classifier weights or forward transformed weights, the latter of which are equivalent to univariate difference maps and are interpretable as neural sources, see (Haufe et al., 2014)).



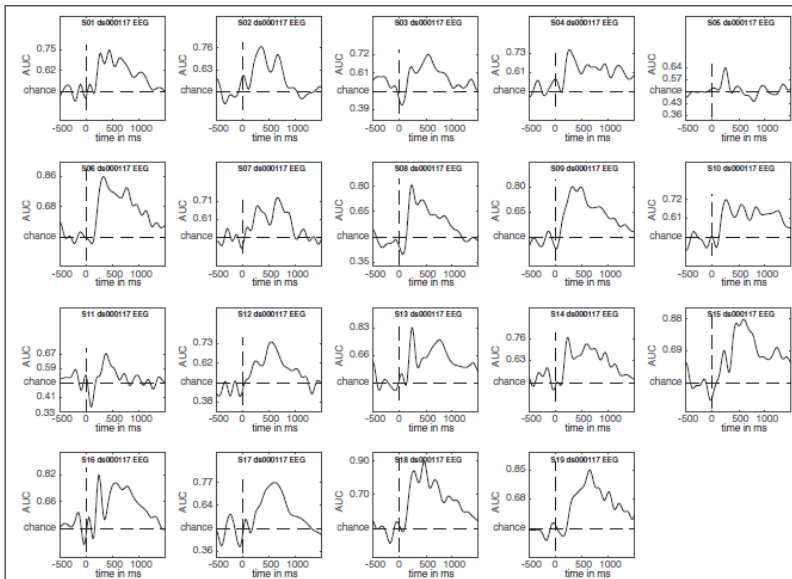


FIGURE 5 | Decoding results of individual subjects for the famous vs. scrambled faces comparison.

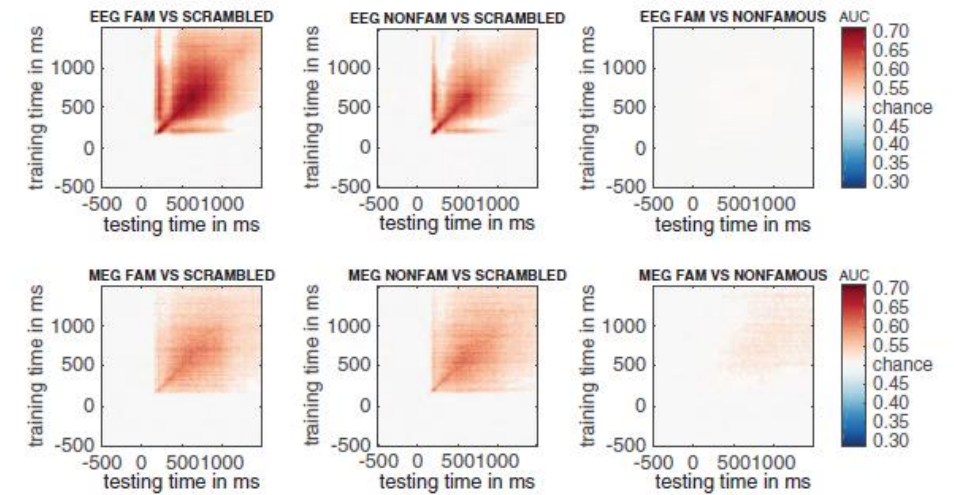


FIGURE 7 | Temporal generalization plots for all six analyses. The plots show the degree to which the classifier when trained on a given time point (on the y-axis), generalizes to other time points in the trial (on the x-axis). Color indicates classifier performance using AUC. The diagonal (from the left bottom to the top right) shows classification performance when the classifier is trained and tested on the same time point. More off-diagonal activity indicates stronger temporal generalization. (top row: EEG, bottom row: MEG).

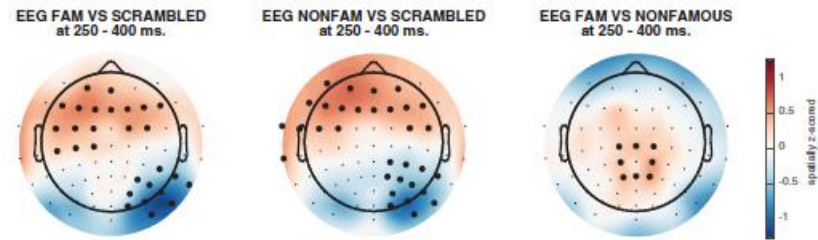


FIGURE 6 | Activation patterns from 250 to 400 ms, spatially normalized (z-scored) for every subject. Thick electrodes denote  $p < 0.05$  under two-sided cluster-based permutation (Maris and Oostenveld, 2007).

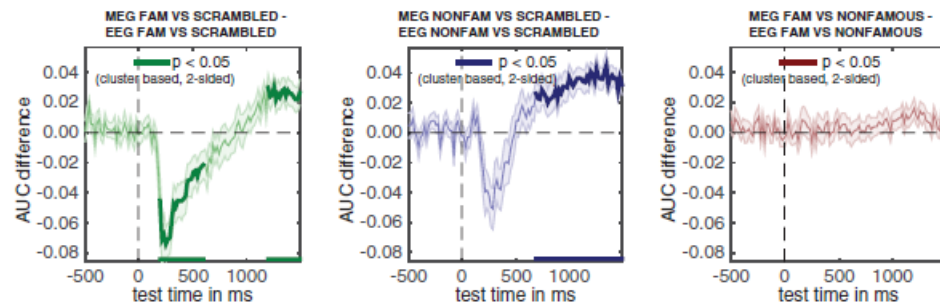


FIGURE 9 | Directly testing the difference in temporal generalization between MEG and EEG given the 250–400 ms training window.

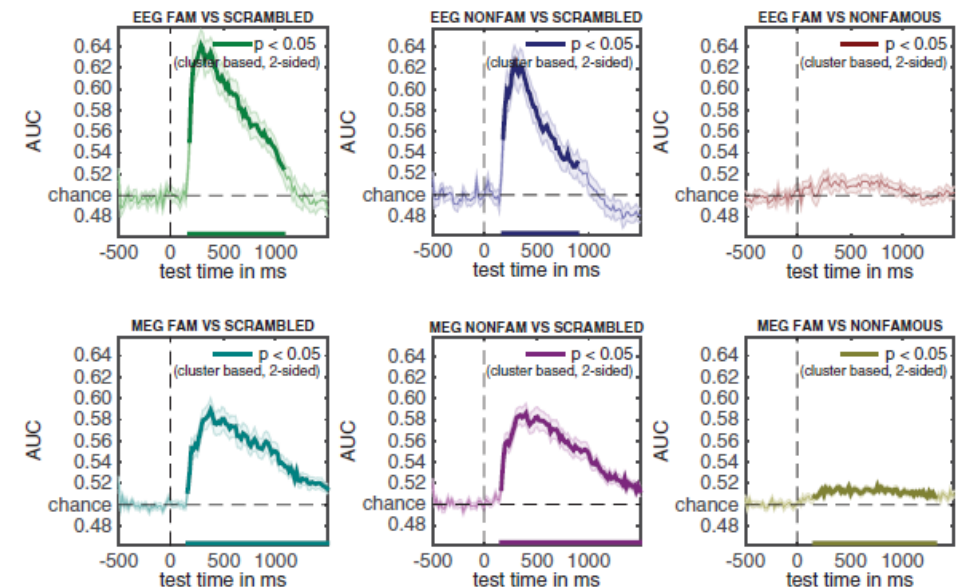


FIGURE 8 | Assessing cortical stability in EEG and MEG through generalization across time given a 250–400 ms training window.

eeg\_adam plugin

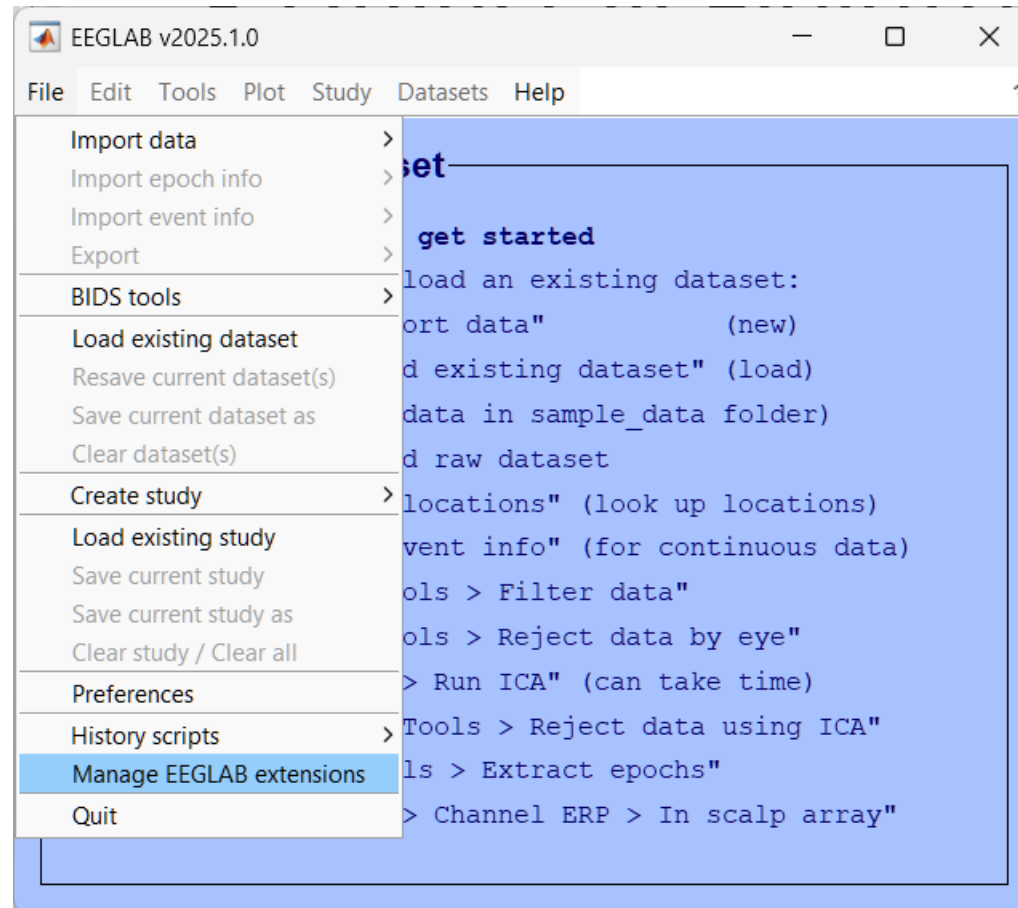
# Manually install eeglab\_adam plugin

- Unzip eeglab\_adam.zip in eeglab/plugins folder
- Contains:
  - Fieldtrip special brew
  - ADAM toolbox
  - EEGLAB Study
  - Paper results



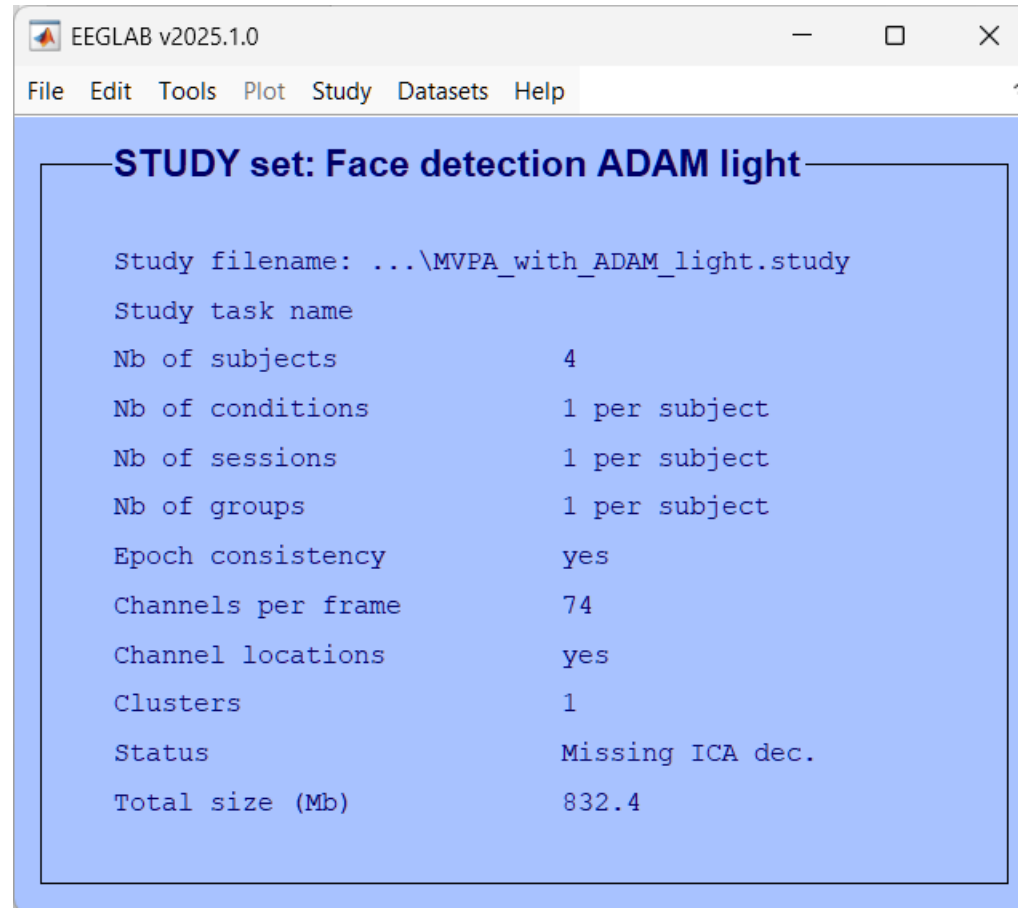
# Setup : Step 1

- Uninstall Fieldtrip plugin using EEGLAB Extension Manager



# Setup : Step 2

- Load STUDY eeglab\_adam\ds002718\MVPA\_with\_ADAM\_light.study



# Setup : Step 3

- Set eeglab\_adam plugin Preferences to setup proper paths to required toolboxes

