Exploring MEG data in Continuous Speech

BRAINHACK SCHOOL TW-SG 2025

CHI WANG

National Taiwan University

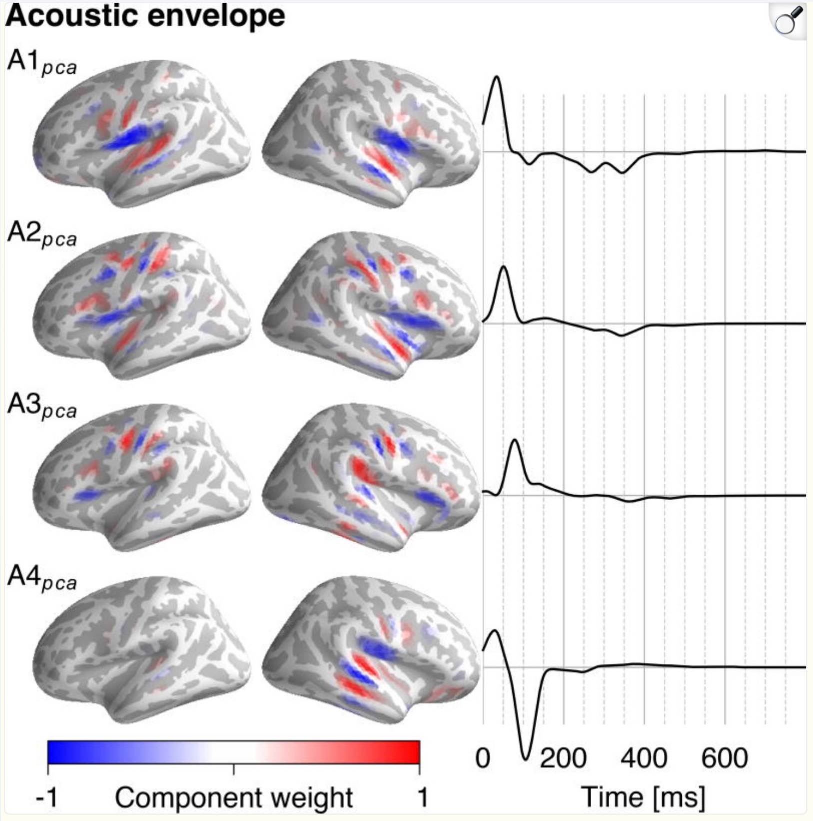
SMN4Lang Dataset / Wang /Nature data

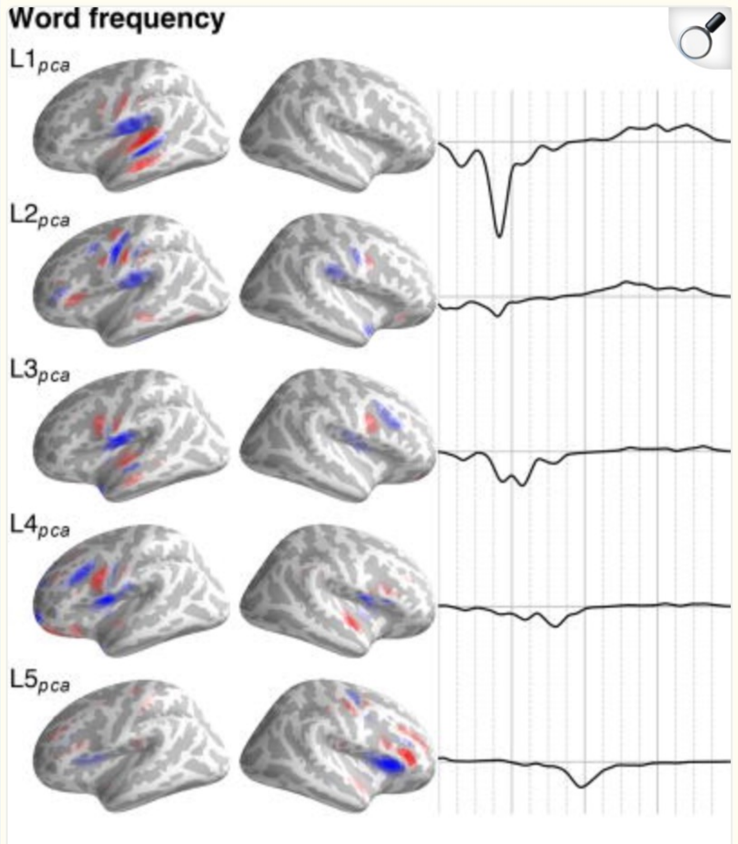
- **Participants:** 12 native Mandarin speakers
- **Recordings:** Both MEG and MRI (T1, T2, fMRI)
- **Stimuli:** ~6 hours of continuous naturalistic Chinese stories (60 stories, 4-7mi)
- **Annotations:**
  1. Acoustic features
  2. Word frequency
  3. Part-of-speech (POS) tags
  4. Syntactic structures
  5. Semantic embeddings (Word2Vec, BERT, GPT-2)
- **Highlights:**
  - Rich, naturalistic speech data for Mandarin Chinese
  - Enables in-depth exploration of brain language processing

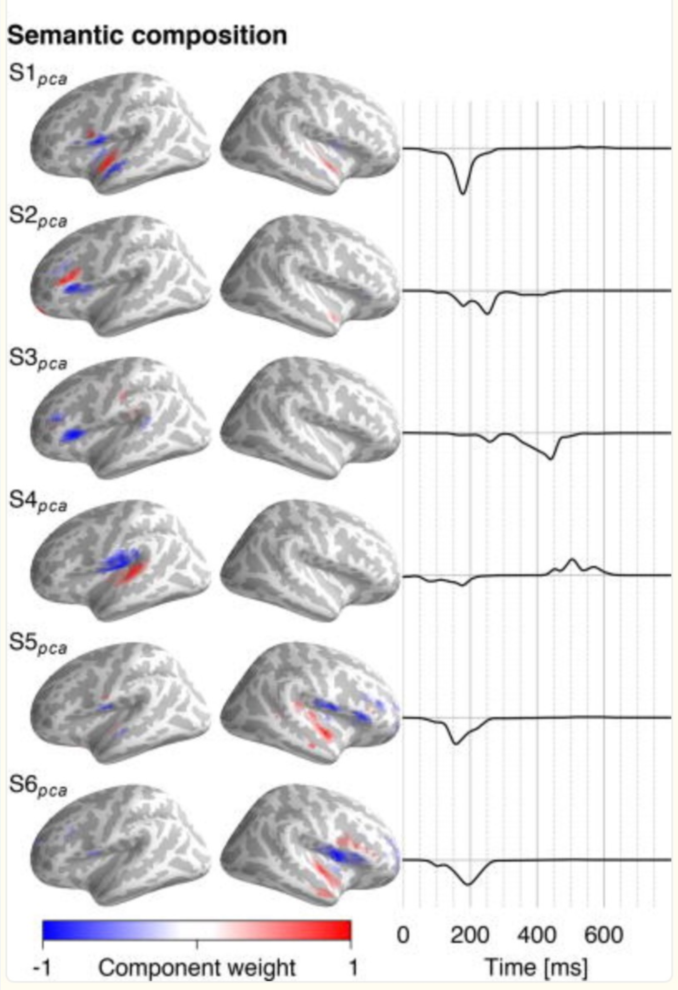
participants

participant_id	age	sex	hand
sub-01	25	M	R
sub-02	30	F	R
sub-03	26	M	R
sub-04	28	M	R
sub-05	26	M	R
sub-06	25	M	R
sub-07	27	F	R
sub-08	23	M	R
sub-09	24	F	R
sub-10	25	M	R
sub-11	23	F	R
sub-12	24	M	R

Brodbeck et al. (2018)







Background & Motivation

- [Brain Language Research] (mostly in English)

↓

- [Mandarin vs. English may activate different brain areas]

↓

- [SMN4Lang Dataset] → MEG + fMRI + Mandarin time-stamped data

↓

- ~~[fMRI Analysis by My Friend → Direct Comparison of Activation Patterns]~~

↓

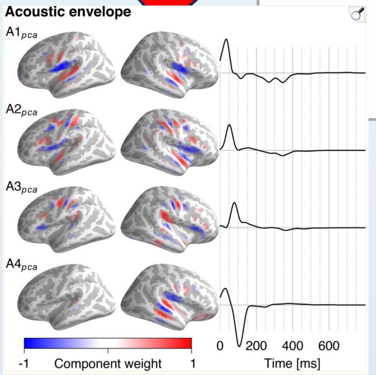
- ~~[MEG Analysis Method Selection] → Source, TRF, Frequency-Specific~~

My Original Three-Stage Analysis Plan

1. Re… ME…
- Brodbeck… et al.(2018)

2. + … al
- Frequen… Specific

3. … e … ing
- Explor… Speech Decoding



Research Aims (In Pitch)

1. How does the brain process continuous natural Mandarin speech in terms of temporal, spectral, and spatial dynamics?

2. Can temporal response functions (TRFs) and frequency-specific MEG features effectively capture Mandarin speech processing in the brain?

3. Can machine learning (SVM) classify brain activity patterns associated with continuous Mandarin speech based on MEG source space data?

Presentation Goals （Reality）

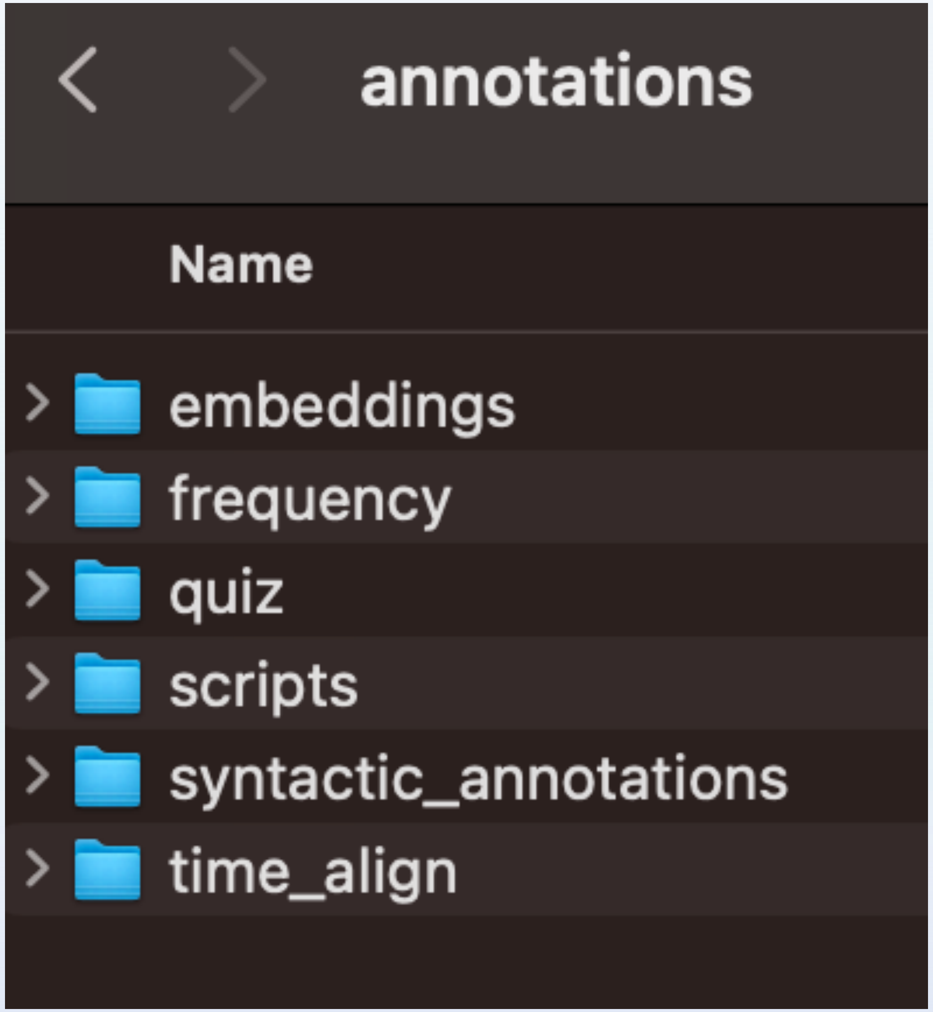
- No prior experience with hands-on MEG data analysis
- Focus: Learning process and exploration attempts

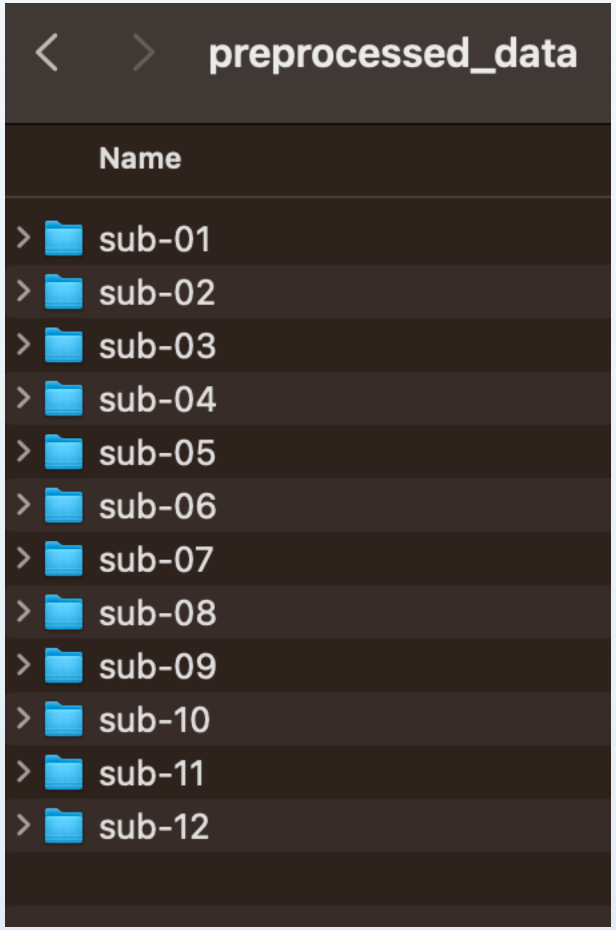
BHS Module -

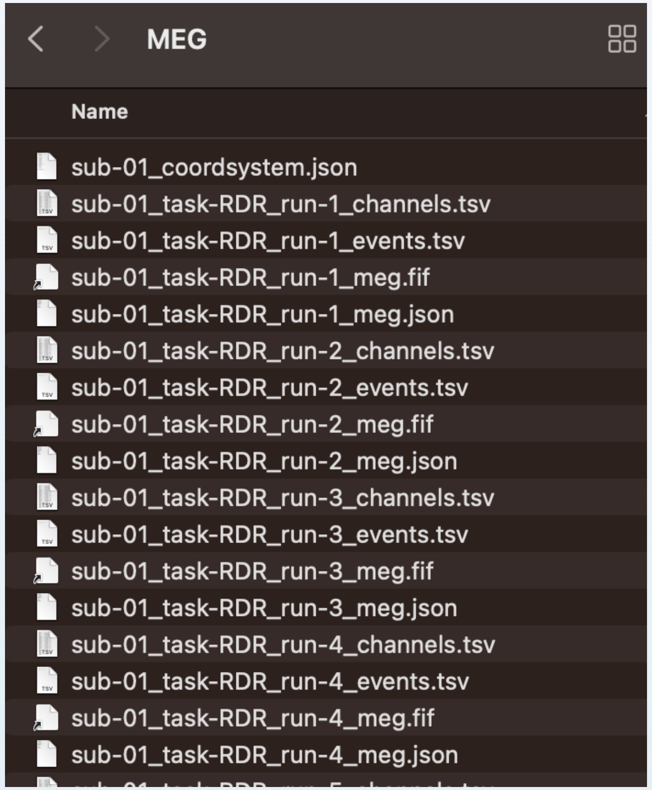
- BASH - A brief introduction to the bash shell
- MNE-Python: Working with MNE-Python and EEG-BIDS
- …Each module has taught me something new and unexpected — things I had never thought about before.

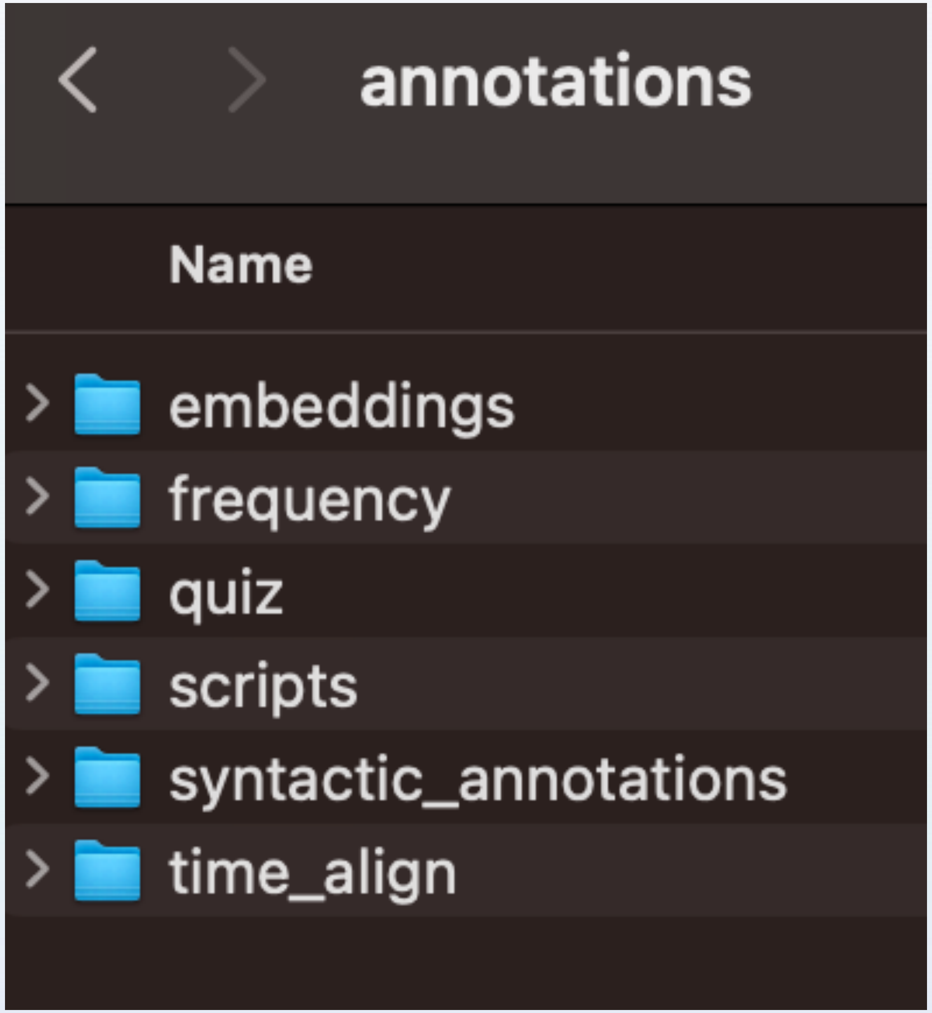
```
(base) meguser@wangqis-MacBook-Pro ~ % conda info --envs

# conda environments:
#
base                  * /Users/meguser/miniconda3
datalad-env             /Users/meguser/miniconda3/envs/datalad-env
eelbrain-env            /Users/meguser/miniconda3/envs/eelbrain-env
main_edu_brain_decoding   /Users/meguser/miniconda3/envs/main_edu_brain_decoding
meg-analysis-env        /Users/meguser/miniconda3/envs/meg-analysis-env
mne_eeg_workshop        /Users/meguser/miniconda3/envs/mne_eeg_workshop
pilot                   /Users/meguser/miniconda3/envs/pilot
pythonpackage           /Users/meguser/miniconda3/envs/pythonpackage
```

Got it by Datalad!









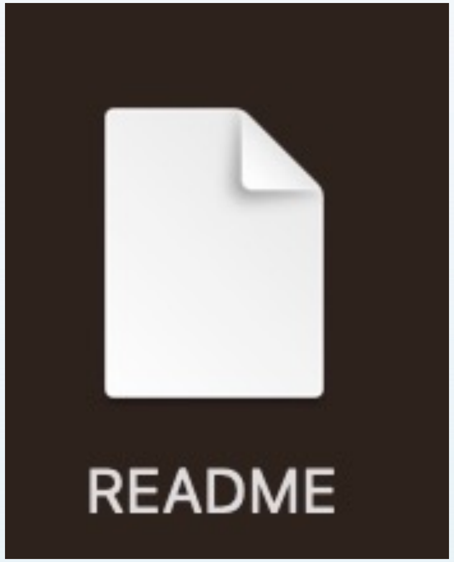

```
- embeddings
  - word2vec
    - word-level
      - 100d
      - 300d
    - char-level
      - 100d
      - 300d
  - gpt
    - word-level
  - bert
    - word-level
    - char-level
- frequency
  - word-level
  - char-level
- quiz
- scripts
- time_align
  - word-level
  - char-level
- syntactic_annotations
  - dependency_parsing
  - constituency_parsing
  - part_of_speech
```

Overcoming Basic Challenges

```
--- TIME ALIGN (word-level) ---
[11.28 11.45 11.68 11.82 12.1  12.65 13.07 14.12 14.25 1
4.57 14.86 14.98
 15.61 15.76 15.84 16.23 16.56 17.04 17.6  17.91]
```

```
--- WORD FREQUENCY (word-level) ---
[15.41563968 12.55887159 15.04833405 15.15246419 14.94689
299 13.53182715
 10.28530866 19.29483723 13.90786273 14.94689299 18.64457
06  12.41426904
 13.44249279 14.67172597 12.83701724 14.70894281 13.87025
534 18.49681653
 12.47815111 19.29483723]
```

```
--- PART OF SPEECH ---
0       我们      PN
1       经常      AD
2       会        VV
3       说        VV
4       教育      NN
5       关系      VV
6       千家万户  NN
7       ，        PU
8       有关      VV
9       教育      NN
10      的        DEC
11      讨论      NN
12      总        AD
13      能        VV
14      引发      VV
15      社会      NN
16      关注      VV
17      。        PU

0       最近      NT
```

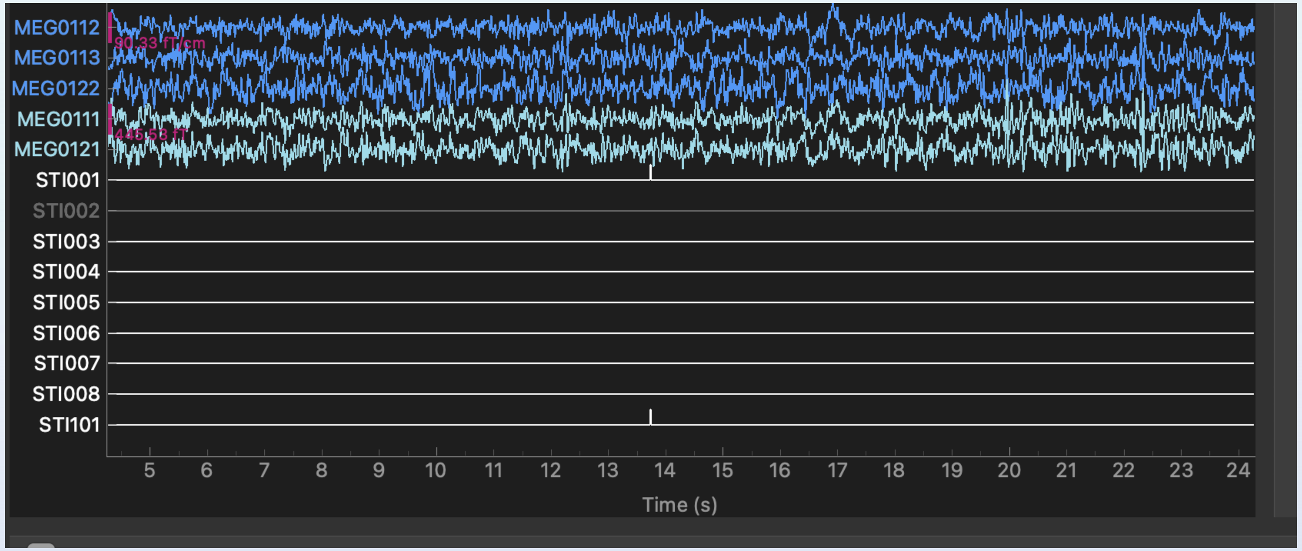


1. Speech to text alignment: The onset and offset time of each character and words in the audio are
provided in the "stimuli/time_align" folder. Note that the onset and offset time were added by 10.65
seconds to align with the time of fMRI images because the fMRI scan was started 10.65 seconds before
playing the audio.

Overcoming Basic Challenges

```python
import mne
import numpy as np

rawpath='../SMN4Lang_data/ds004078/derivatives/preprocessed_data/sub-01/MEG/sub-01_task-RDR_run-1_meg.fif'
raw = mne.io.read_raw_fif(rawpath, preload=True)
raw
```

```
Opening raw data file ../SMN4Lang_data/ds004078/derivatives/preprocessed_data/sub-01/MEG/sub-01_task-RDR_run-1_meg.fif...
    Range : 13000 ... 458999 =     13.000 ...   458.999 secs
Ready.
Reading 0 ... 445999  =      0.000 ...   445.999 secs...
```



- Shift issues
  (alignment between words and MEG signals)
- Range adjustments
  (impact of segment selection)

```
--- TIME ALIGN (word-level) ---
[11.28 11.45 11.68 11.82 12.1  12.65 13.07 14.12 14.25 1
4.57 14.86 14.98
 15.61 15.76 15.84 16.23 16.56 17.04 17.6  17.91]
```

alignstimuli.ipynb

Name	Modified	Size
story_1_stimulus_table.csv	5 min. ago	39.1 KB
story_2_stimulus_table.csv	5 min. ago	37.1 KB
story_3_stimulus_table.csv	5 min. ago	34.1 KB
story_4_stimulus_table.csv	5 min. ago	34.5 KB
story_5_stimulus_table.csv	5 min. ago	33.8 KB
story_6_stimulus_table.csv	5 min. ago	33.2 KB
story_7_stimulus_table.csv	5 min. ago	31 KB
story_8_stimulus_table.csv	5 min. ago	28.9 KB
story_9_stimulus_table.csv	5 min. ago	29.3 KB
story_10_stimulus_table.csv	5 min. ago	26 KB
story_11_stimulus_table.csv	5 min. ago	39.3 KB
story_12_stimulus_table.csv	5 min. ago	37.7 KB
story_13_stimulus_table.csv	5 min. ago	31.4 KB
story_14_stimulus_table.csv	5 min. ago	33.9 KB
story_15_stimulus_table.csv	5 min. ago	32.6 KB
story_16_stimulus_table.csv	5 min. ago	30.6 KB
story_17_stimulus_table.csv	5 min. ago	32.9 KB
story_18_stimulus_table.csv	5 min. ago	31.4 KB
story_19_stimulus_table.csv	5 min. ago	31.8 KB
story_20_stimulus_table.csv	5 min. ago	25.4 KB
story_21_stimulus_table.csv	5 min. ago	38.1 KB
story_22_stimulus_table.csv	5 min. ago	35.5 KB
story_23_stimulus_table.csv	5 min. ago	30.9 KB
story_24_stimulus_table.csv	5 min. ago	32.9 KB
story_25_stimulus_table.csv	5 min. ago	30.1 KB
story_26_stimulus_table.csv	5 min. ago	30.1 KB

```python
#讀取存好的檔案
import pandas as pd

story_id = 1
csv_path = f'StimulusTables/story_{story_id}_stimulus_table.csv'

df = pd.read_csv(csv_path)

# 看前幾筆確認
print(df.head(20))
```

```
     word  pos  word_freq_log  word_onset_sec
0    我们    PN      15.415640            0.63
1    经常    AD      12.558872            0.80
2    会      VV      15.048334            1.03
3    说      VV      15.152464            1.17
4    教育    NN      14.946893            1.45
5    关系    VV      13.531827            2.00
6    千家万户  NN      10.285309            2.42
7    ，      PU      19.294837            3.47
8    有关    VV      13.907863            3.60
9    教育    NN      14.946893            3.92
10   的     DEC      18.644571            4.21
11   讨论    NN      12.414269            4.33
12   总      AD      13.442493            4.96
13   能      VV      14.671726            5.11
14   引发    VV      12.837017            5.19
15   社会    NN      14.708943            5.58
16   关注    VV      13.870255            5.91
17   。      PU      18.496817            6.39
18   最近    NT      12.478151            6.95
19   ，      PU      19.294837            7.26
```

```python
rawpath='ds004078/derivatives/preprocessed_data/sub-01/MEG/sub-01_task-RDR_run-1_meg.fif'
raw = mne.io.read_raw_fif(rawpath, preload=True)
raw
```

Acquisition

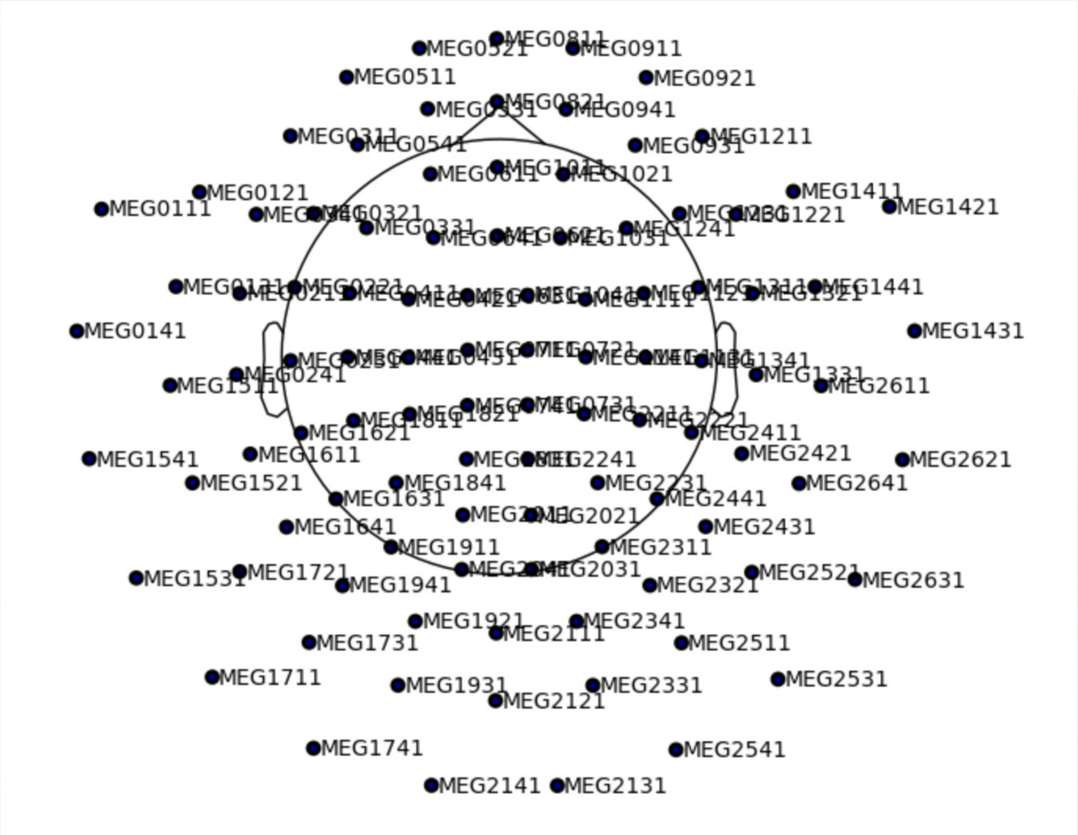
Duration	00:07:26 (HH:MM:SS)
Sampling frequency	1000.00 Hz
Time points	446,000

Channels

Magnetometers	102
Gradiometers	204
BIO	1
Stimulus	9
Internal Active Shielding data (Triux systems)	11
System status channel information (Triux systems)	1
Head & sensor digitization	127 points

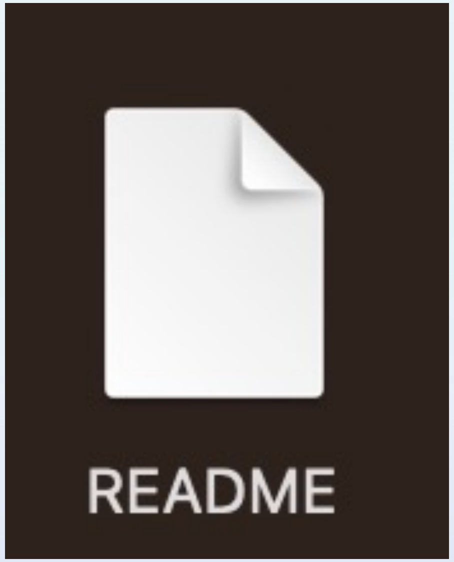
Filters

Highpass	1.00 Hz
Lowpass	40.00 Hz

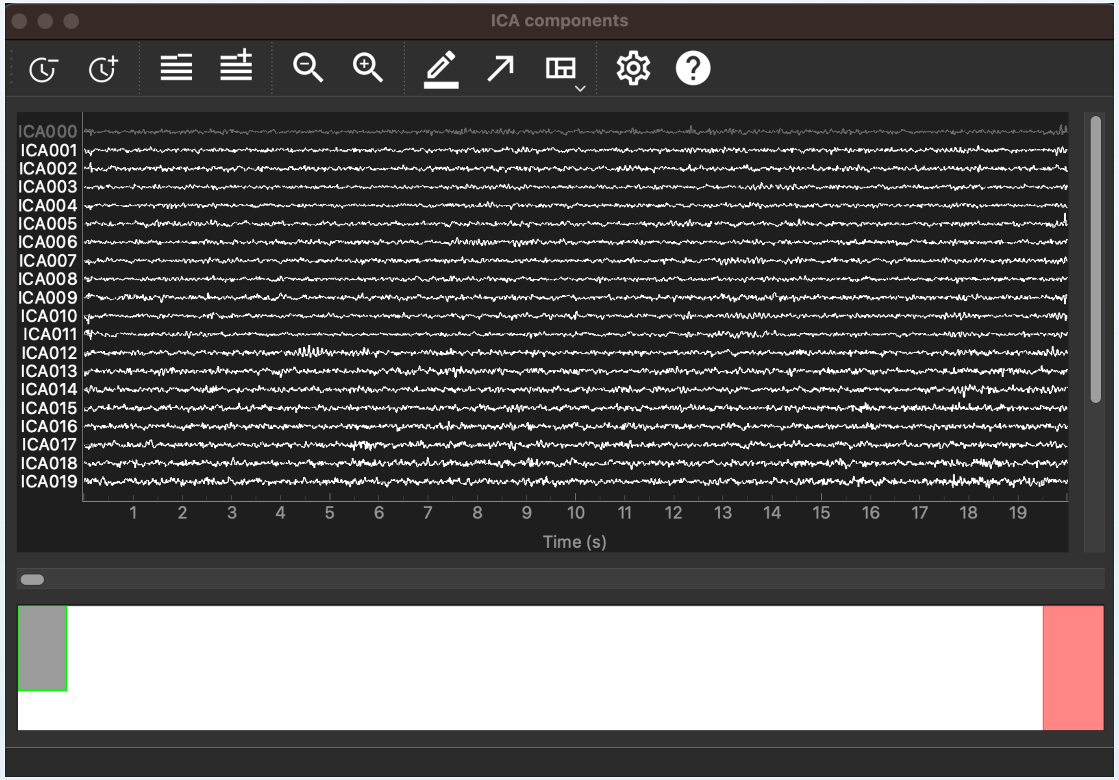


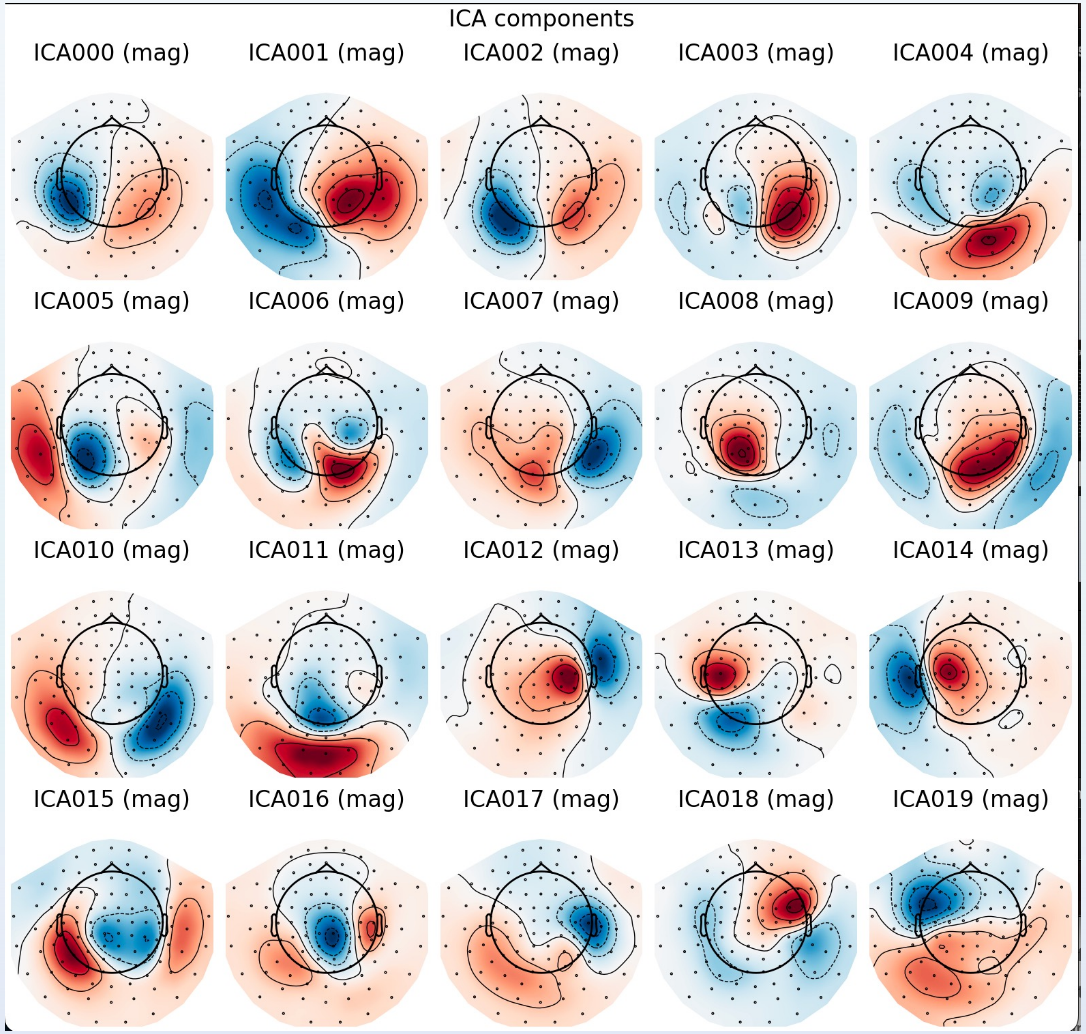
Preprocessed Data

- The MEG data was first preprocessed using the temporal Signal Space Separation (tSSS) method and the bad channels were excluded.
- And then the independent component analysis (ICA) method was applied to remove the ocular artefacts using the MNE software.

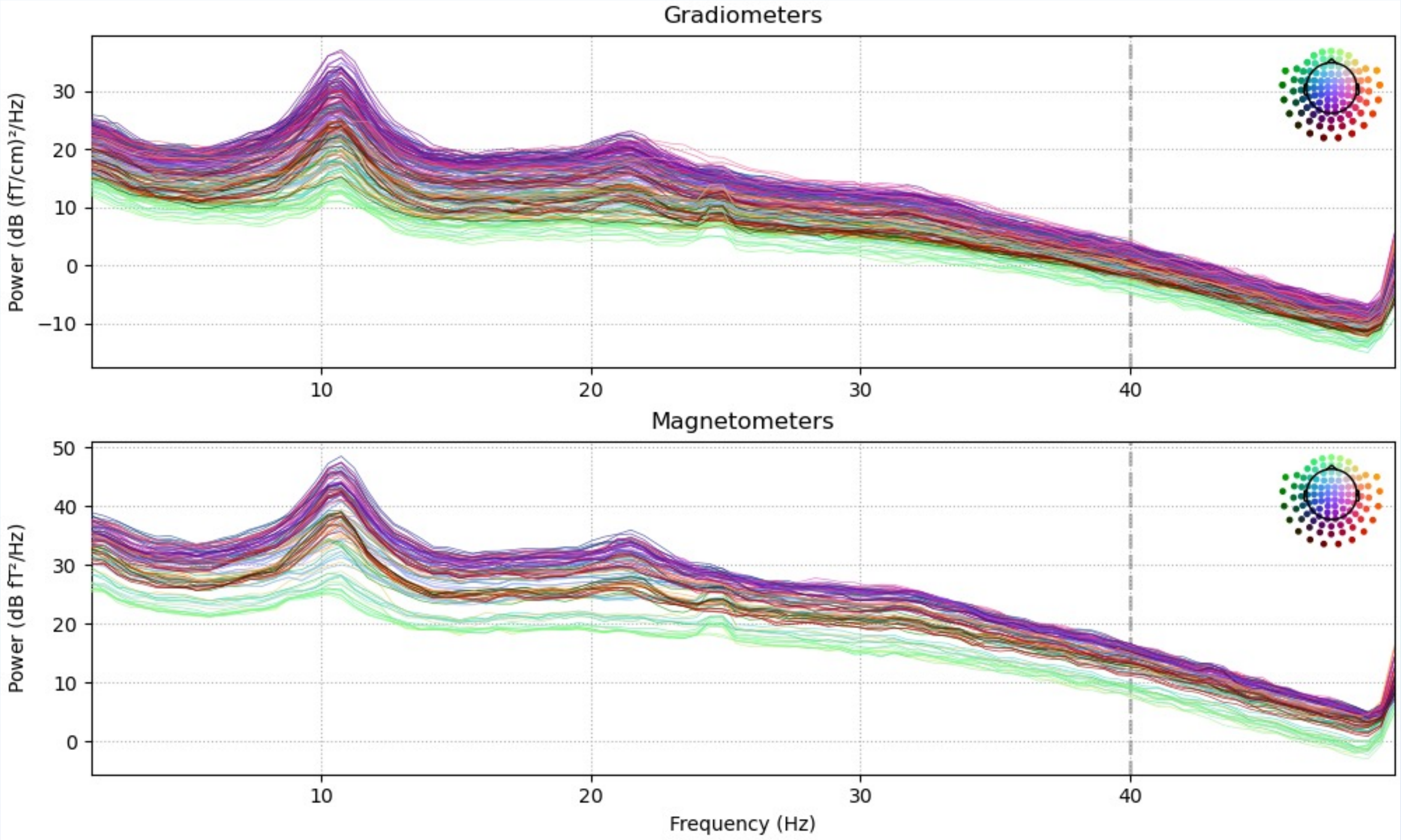


ICA (I didn't exclude any ICA)





PSD_sub01_run1



Epoch

```
FileNotFoundError: fname does not exist: "/Users/meguser/Desktop/Project for Brainhack/domybest/../SMN4Lang_da
ta/ds004078/derivatives/preprocessed_data/sub-01/MEG/sub-01_task-RDR_run-16_meg.fif"
```

/ PreprocessedEpochs /

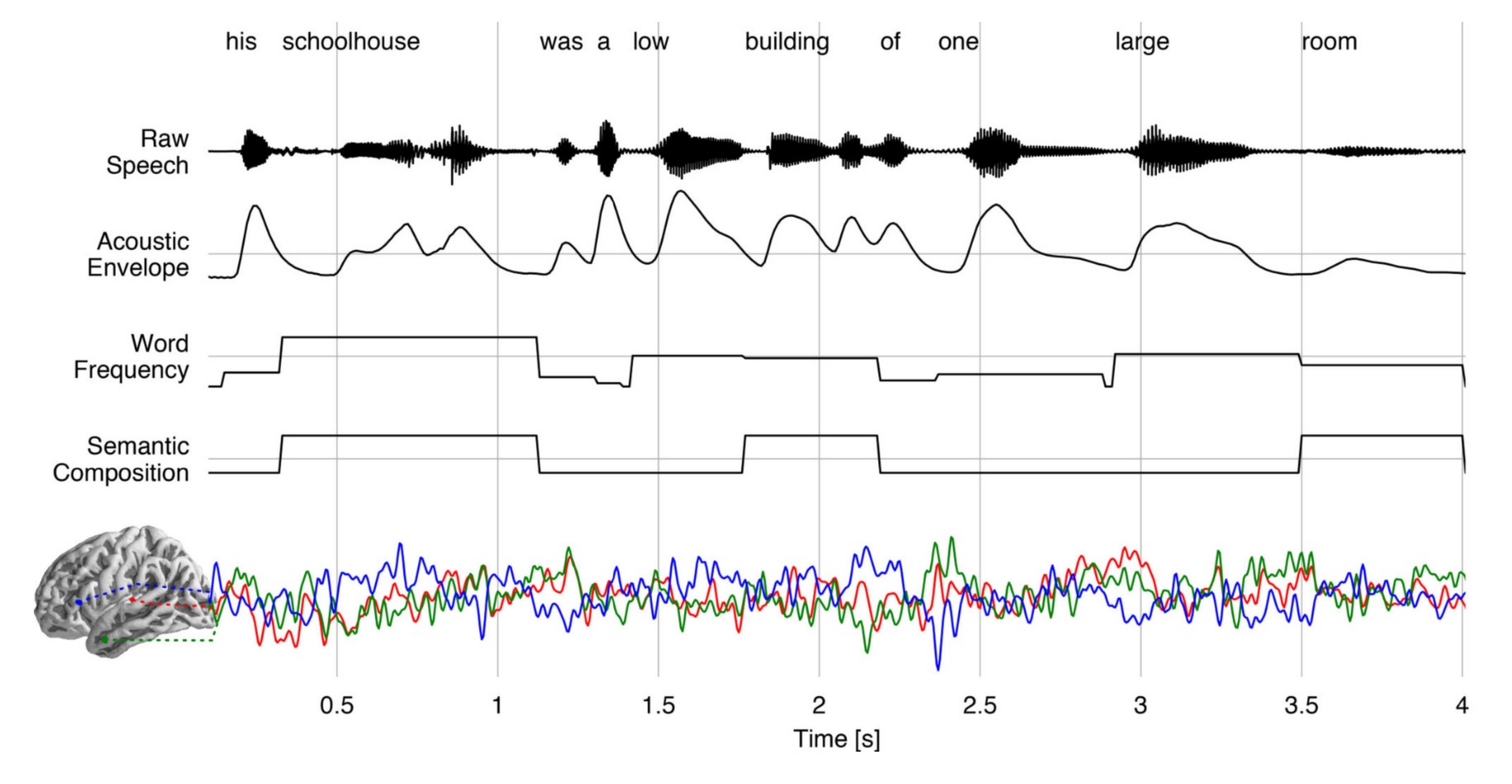
Name
sub-01_run-1
sub-01_run-2
sub-01_run-3
sub-01_run-4
sub-01_run-5
sub-01_run-6
sub-01_run-7
sub-01_run-8
sub-01_run-9
sub-01_run-10
sub-01_run-11
sub-01_run-12
sub-01_run-13

/ PreprocessedEpochs / sub-01_run-1 /

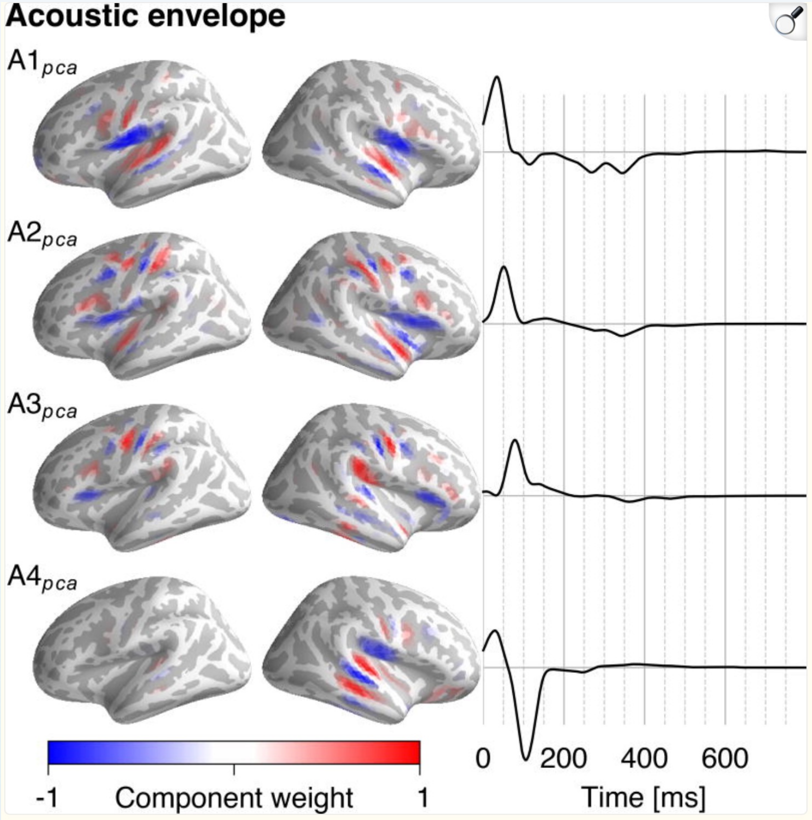
Name
sub-01_run-1_epo.fif
sub-01_run-1_meta.csv

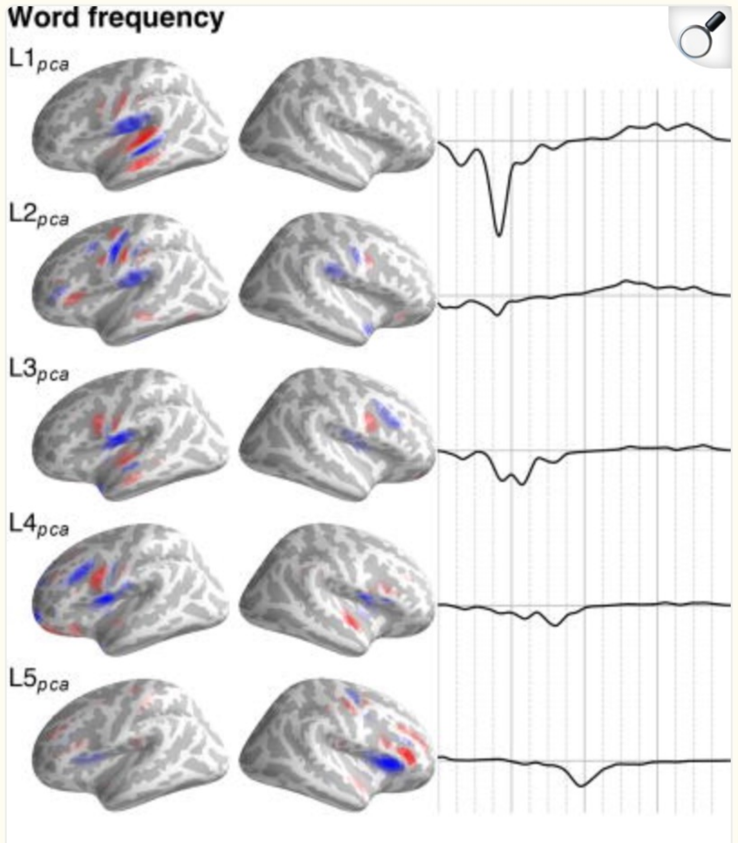
Replicate Brodbeck et al. (2018) Approach

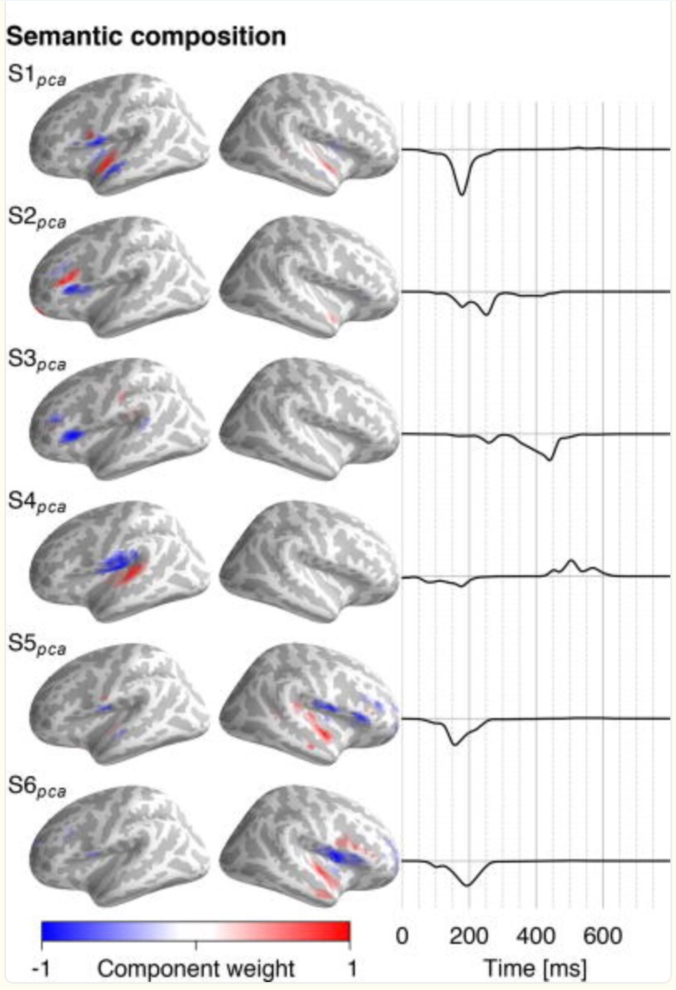
- **Stimulus:** Continuous natural speech (stories)
- **Method:**
  - **Source Localization:** Minimum norm estimate in MEG
  - **Linear kernel estimation** (estimates Temporal Response Function, TRF)
    - Models brain's response over time to continuous stimuli, mapping how features drive neural activity with temporal delay.
    - **Input Features:** Acoustic envelope, word frequency, semantic features
    - **Output:** Brain responses in source space, indicating when and where activity occurs
- **Why this approach?**
  - Preserves the richness of continuous speech, avoids trial averaging
- **Expected Results:**
  - Temporal response curves (timing of brain reactions)
  - Brain regions responsive to specific speech features



Brodbeck et al. (2018)







Brodbeck et al. (2018)

Initial Attempt: Using Eelbrain

- Original plan:
  - Reproduce Brodbeck et al. (2018) results
  - Tool: Eelbrain toolbox + public code
- Challenges:
  - Eelbrain installation successful but some functions failed
  - Data format incompatibility issues → extensive debugging
- Outcome:
  - Gained understanding of Eelbrain concepts and TRF pipeline
  - <u>Switched to using MNE + manual TRF implementation</u>

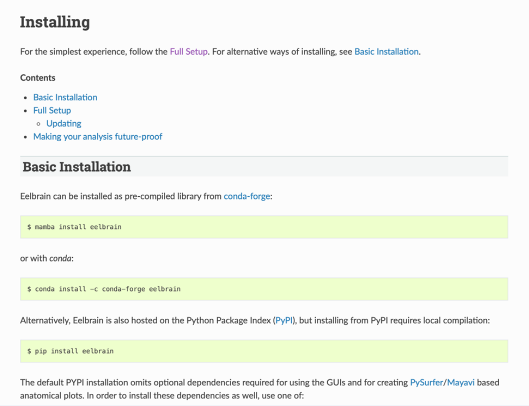
› Elife. 2023 Nov 29:12:e85012. doi: 10.7554/eLife.85012.

Eelbrain, a Python toolkit for time-continuous analysis with temporal response functions

Christian Brodbeck [1], Proloy Das [2], Marlies Gillis [3], Joshua P Kulasingham [4], Shohini Bhattasali [5], Phoebe Gaston [1], Philip Resnik [6], Jonathan Z Simon [6]

Affiliations + expand

PMID: 38018501 PMCID: PMC10783870 DOI: 10.7554/eLife.85012



Installing

For the simplest experience, follow the Full Setup. For alternative ways of installing, see Basic Installation.

Contents

- Basic Installation
- Full Setup
  - Updating
- Making your analysis future-proof

Basic Installation

Eelbrain can be installed as pre-compiled library from conda-forge:

```
$ mamba install eelbrain
```

or with *conda*:

```
$ conda install -c conda-forge eelbrain
```

Alternatively, Eelbrain is also hosted on the Python Package Index (PyPI), but installing from PyPI requires local compilation:

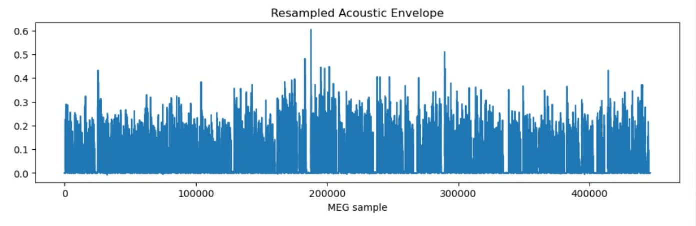
```
$ pip install eelbrain
```

The default PYPI installation omits optional dependencies required for using the GUIs and for creating PySurfer/Mayavi based anatomical plots. In order to install these dependencies as well, use one of:

Temporal Response Function (TRF)

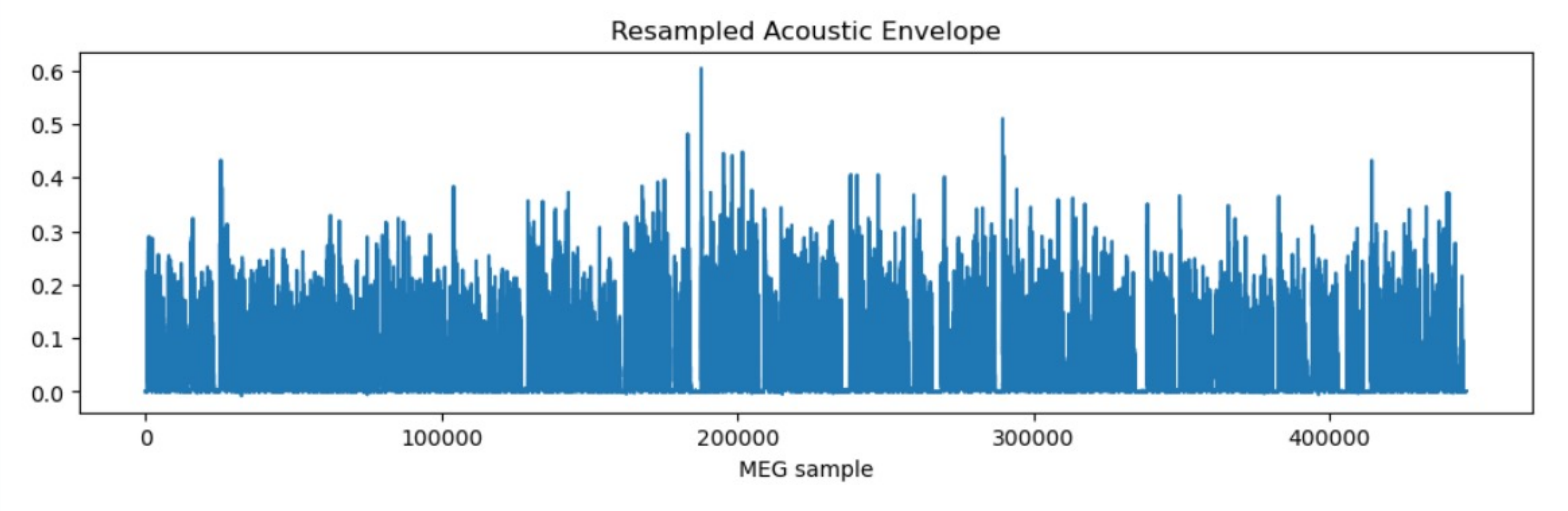
- Linear model of brain responses to continuous stimuli
- Maps stimulus features → neural activity over time
- Captures timing and strength of stimulus-driven neural responses

TRF Workflow (Acoustic Envelope Tracking)

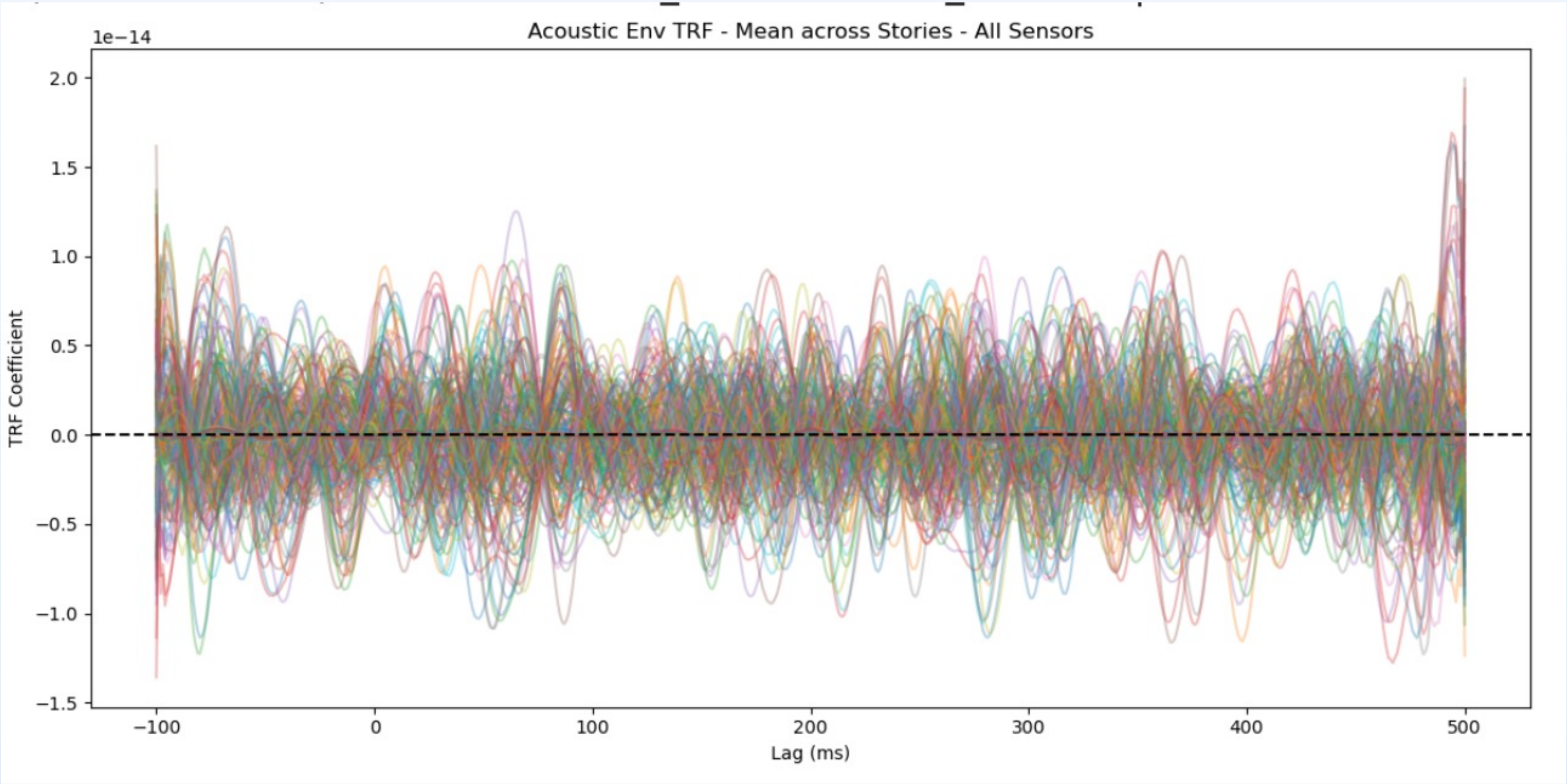


1. Audio (.wav)
2. Hilbert Transform → Envelope
3. Resample Envelope to MEG frequency sampling rate
4. Standardize (Z-score)
5. Build Lagged Matrix (X) Lags: -100 ms → +500 ms
6. MEG data (Y)
7. Ridge Regression (per ch.) $\alpha = 1.0$
8. Save TRF weights + info (coefs, lags, picks, meg_info)

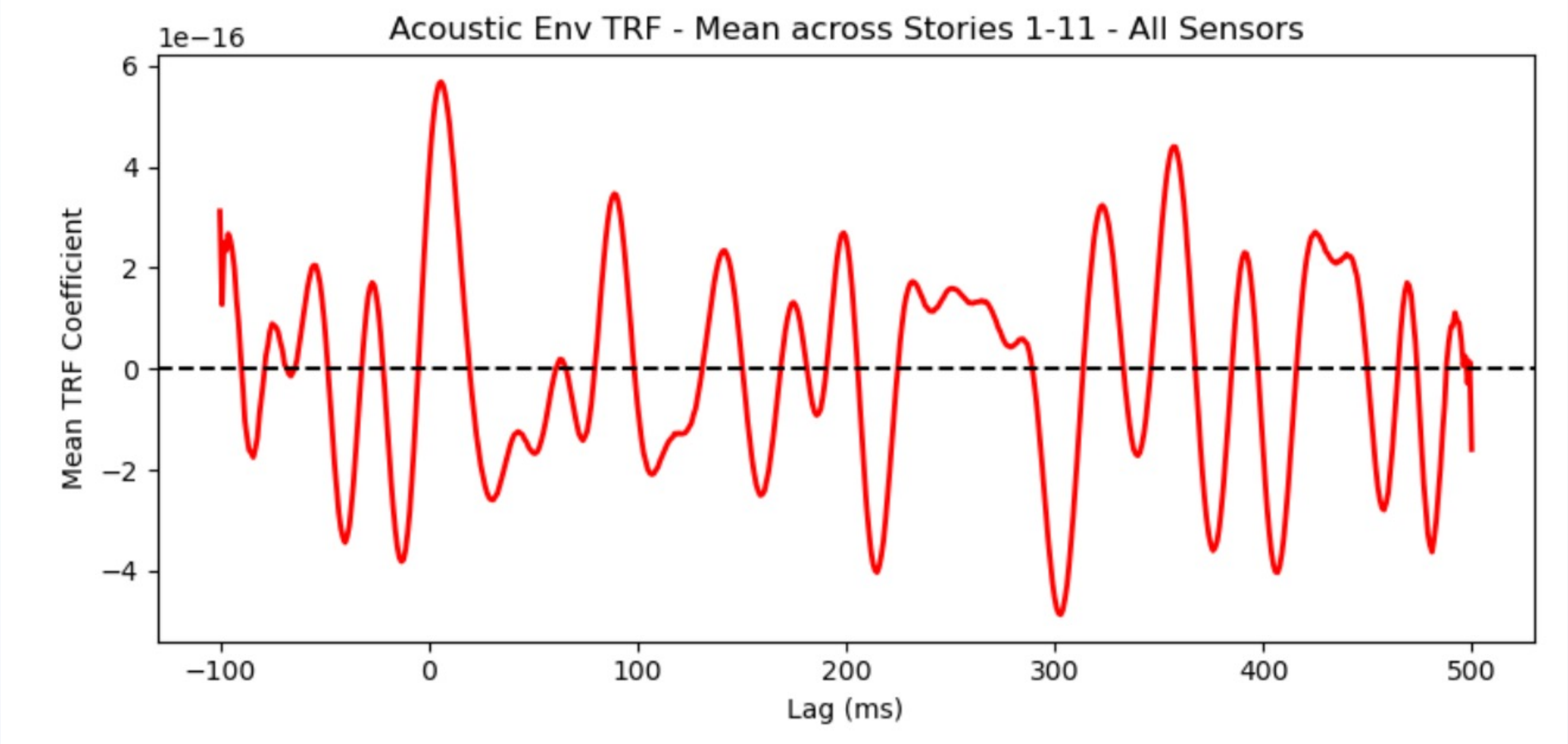
Acoustic Envelope_sub01_run1



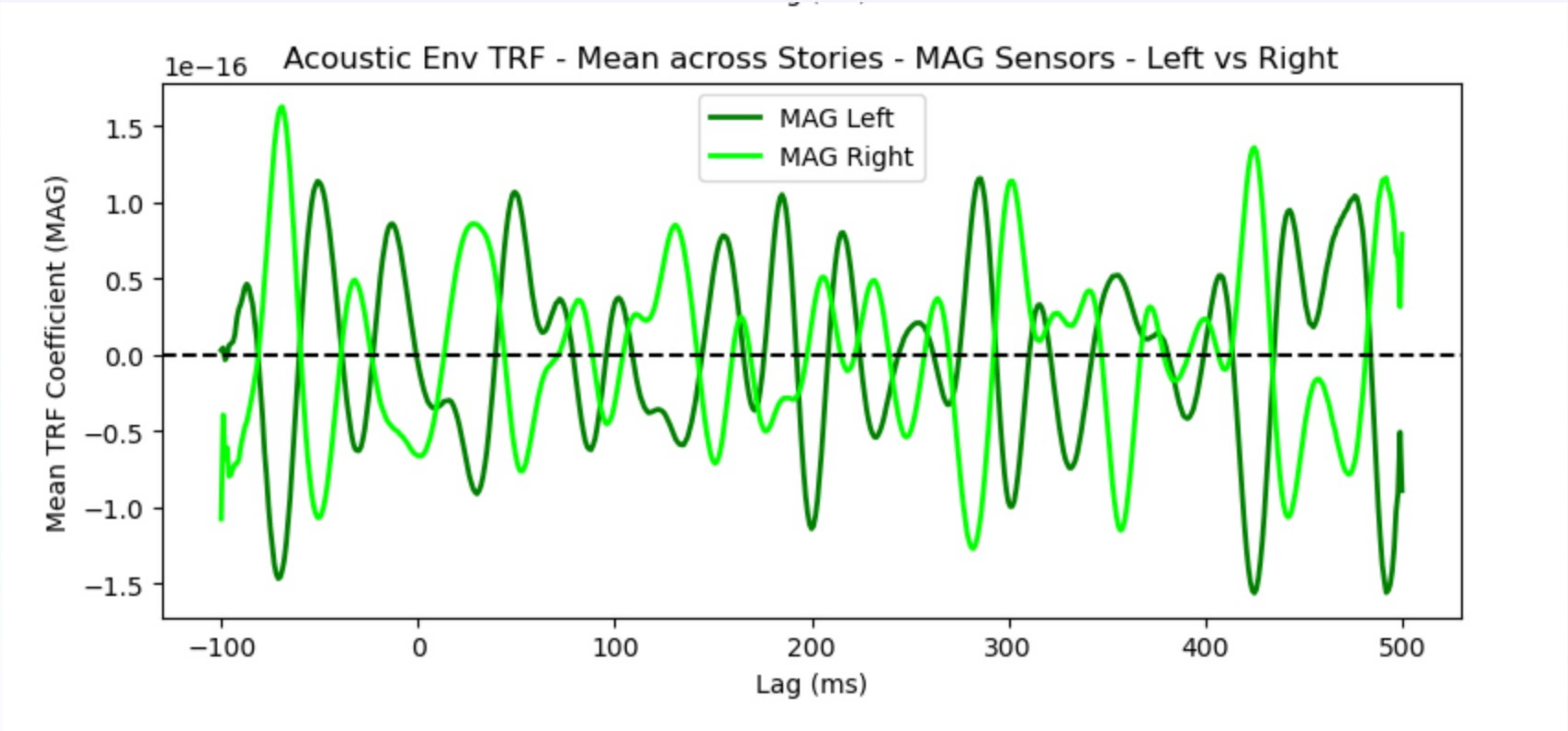
Acoustic Envelope TRF



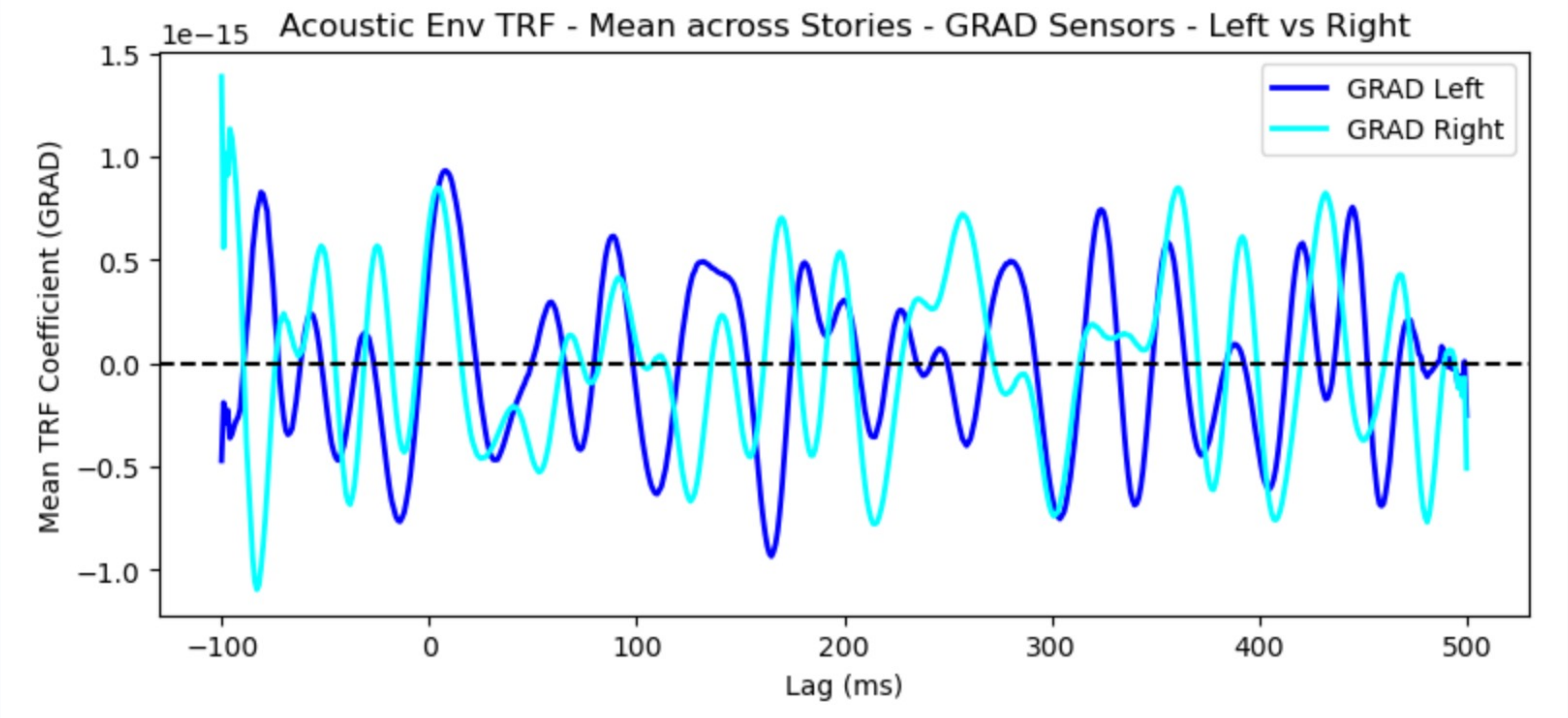
Acoustic Envelope TRF



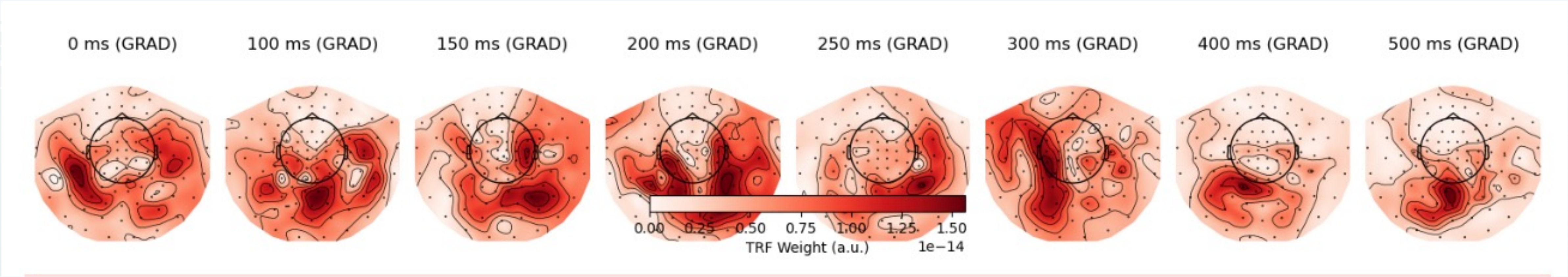
Acoustic Envelope TRF

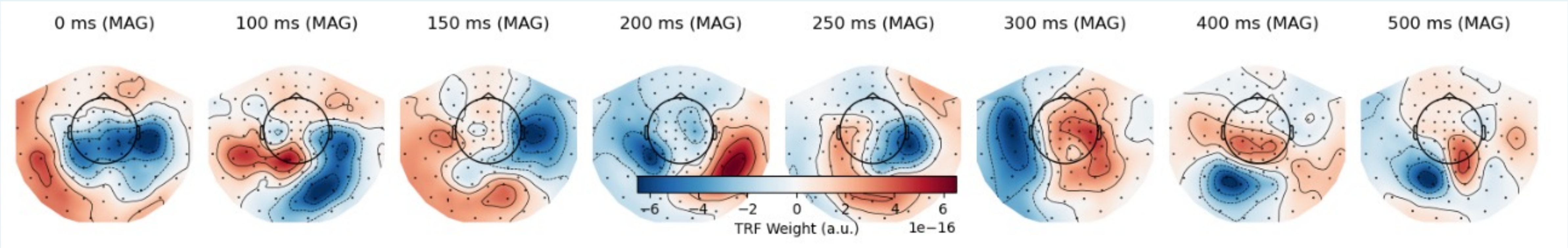


Acoustic Envelope TRF

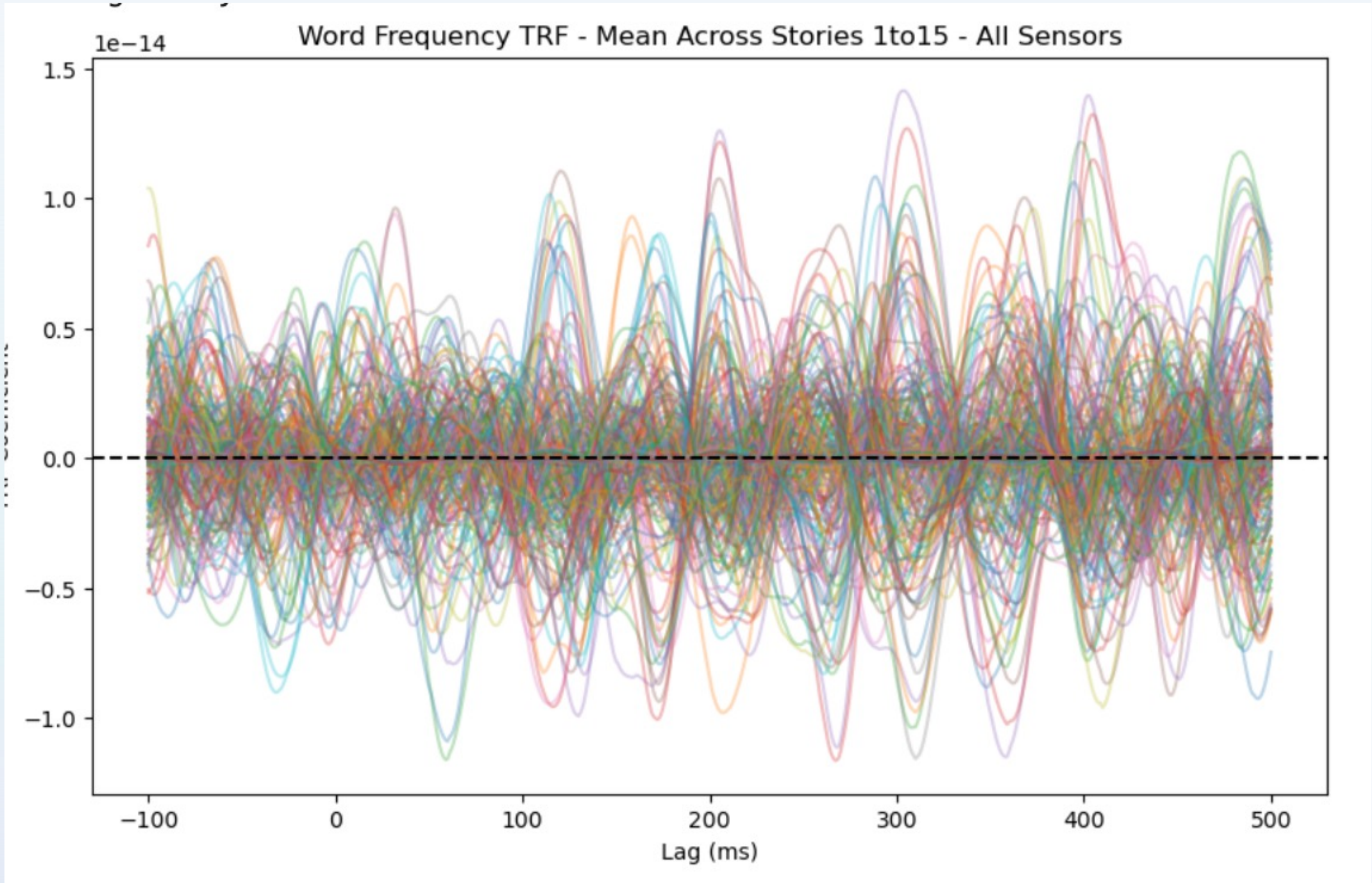


Acoustic Envelope TRF

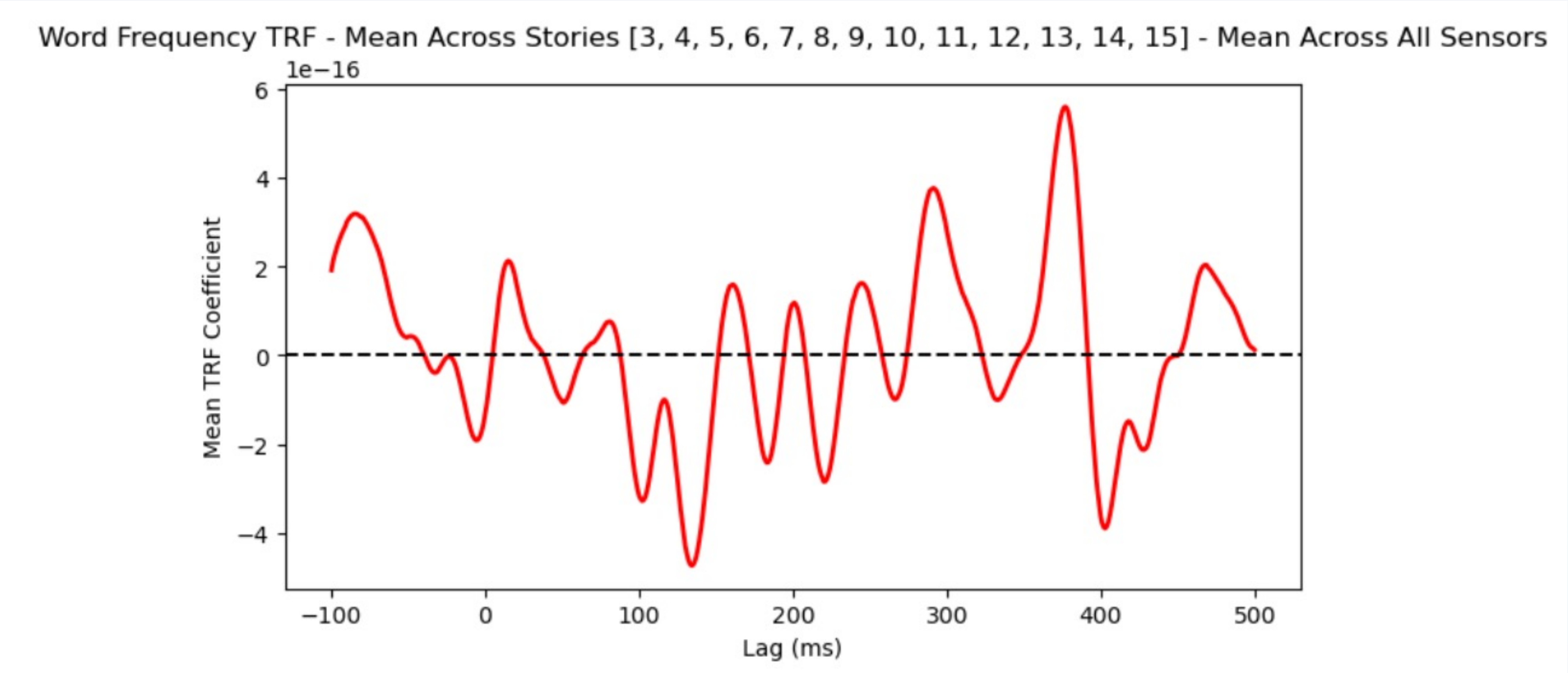




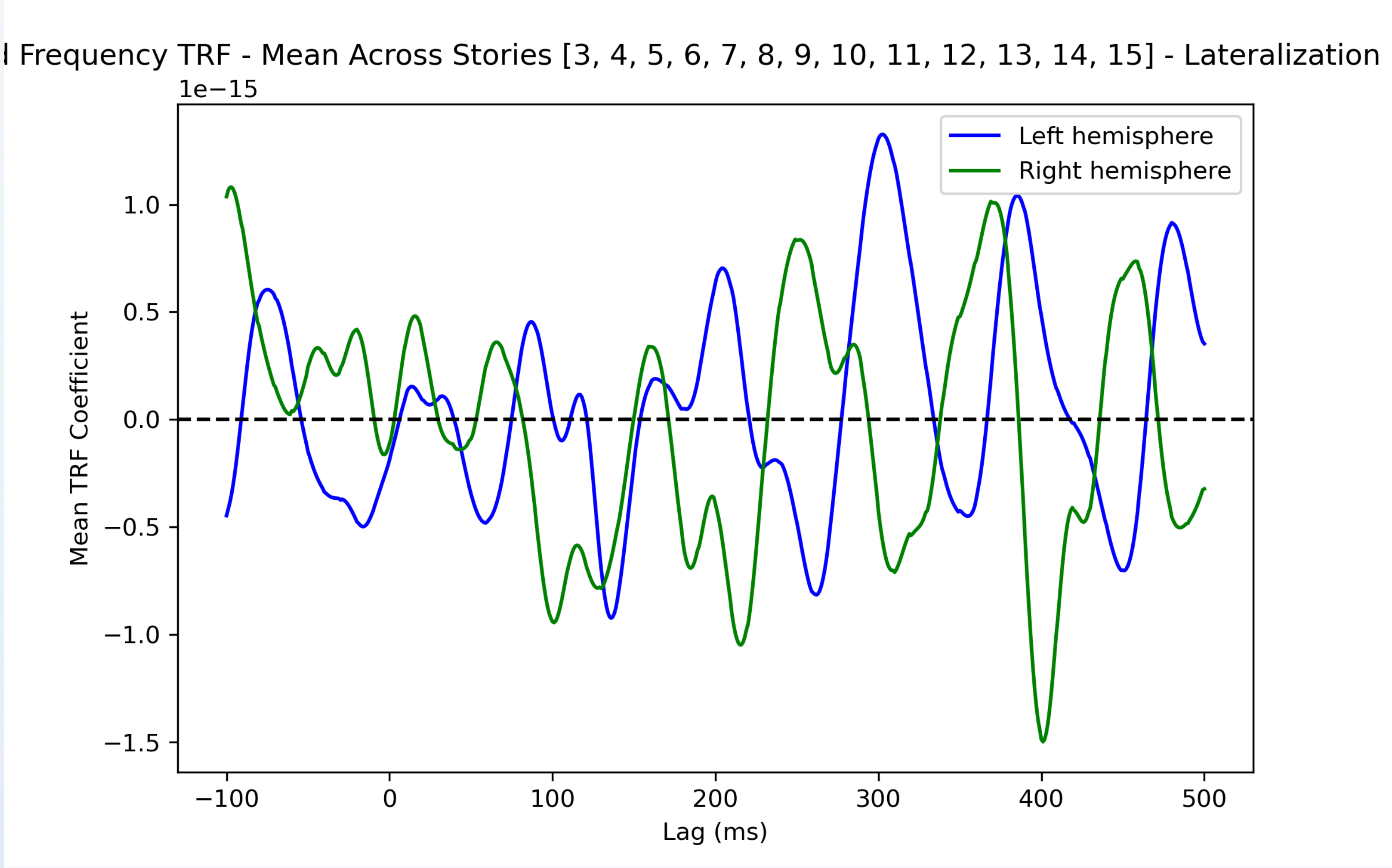
Word Frequency TRF



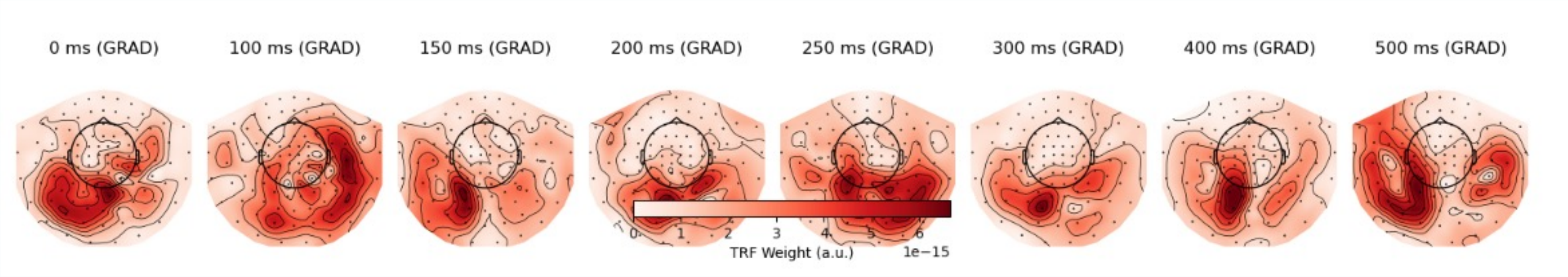
Word Frequency TRF

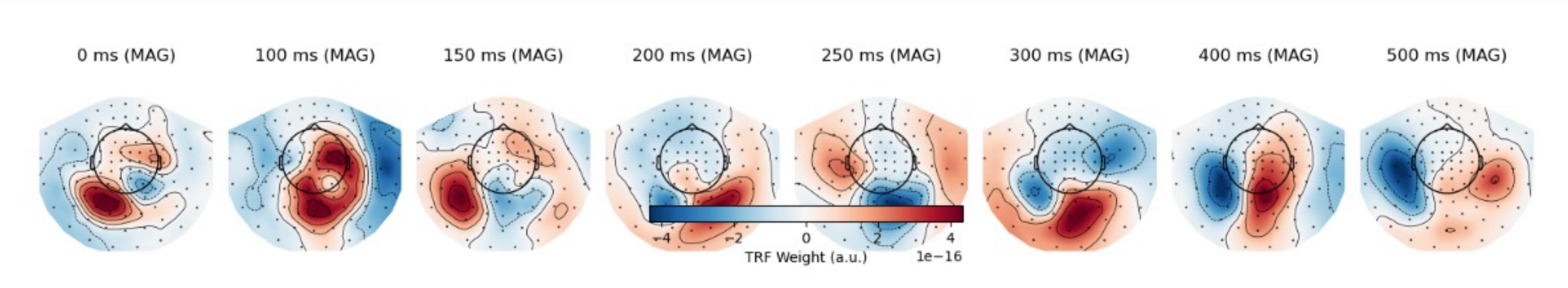


Word Frequency TRF



Word Frequency TRF





Current Limitations & Future Directions

- **Current Status**
  - At this point, I'm not able to interpret the TRF results yet.
  - Some patterns emerge, but their meaning is still unclear.
- **Next Steps**
  - Localize responses to brain regions consistent with the literature.
  - Compute TRF specifically for sensors / channels corresponding to these regions.

SVM (Support Vector Machine

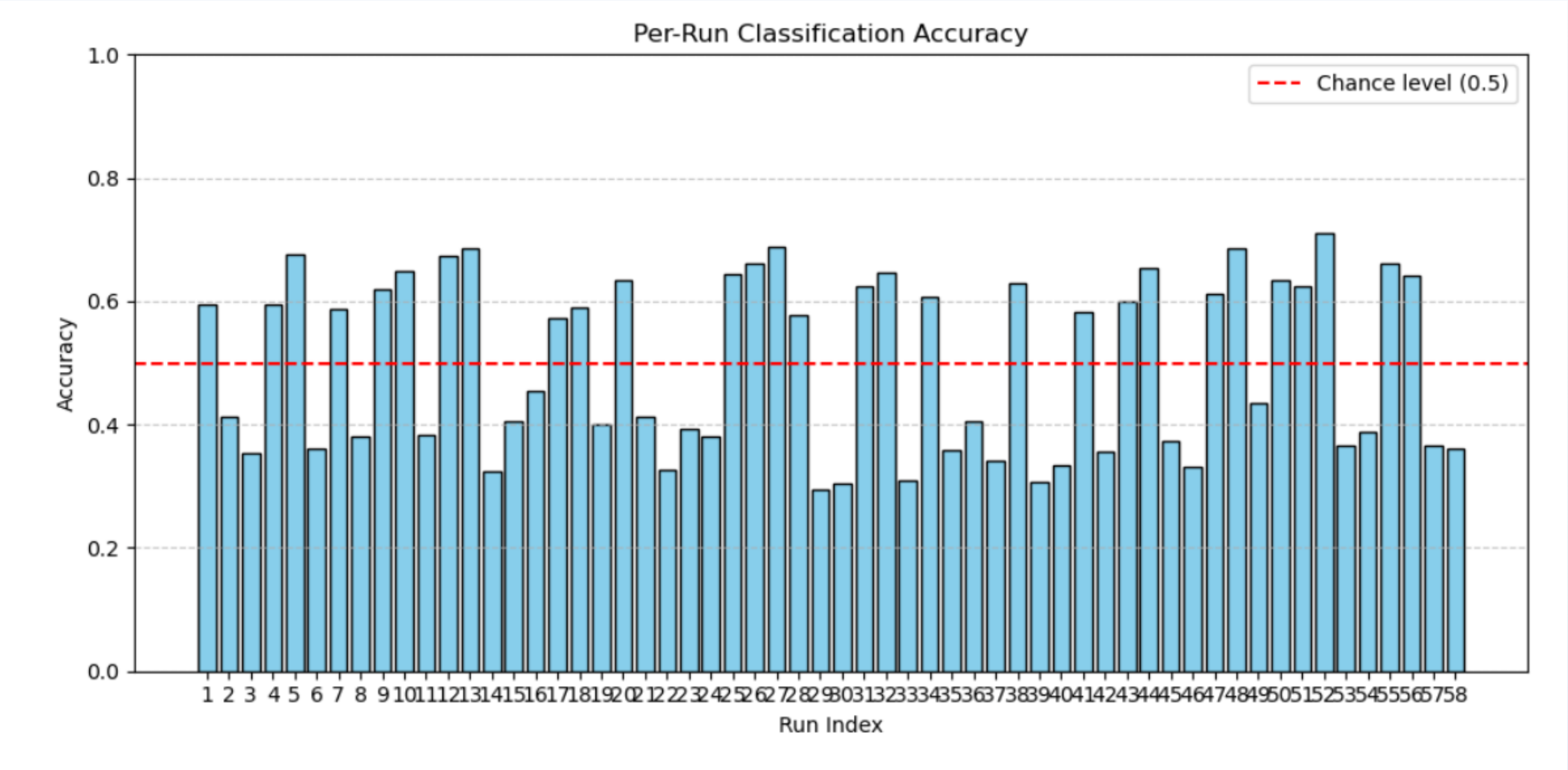
- A **supervised learning algorithm** for classification and regression
- Finds the **optimal hyperplane** that best separates classes in feature space
- **Margin maximization**
  - → chooses the boundary that maximizes distance to nearest samples (support vectors)
- Can handle **linear and nonlinear separation** (via kernel trick)

Methods used

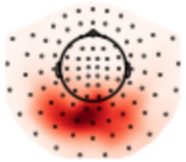
```python
import os
from sklearn.svm import SVC
from sklearn.model_selection import LeaveOneGroupOut, cross_val_score
import matplotlib.pyplot as plt
```

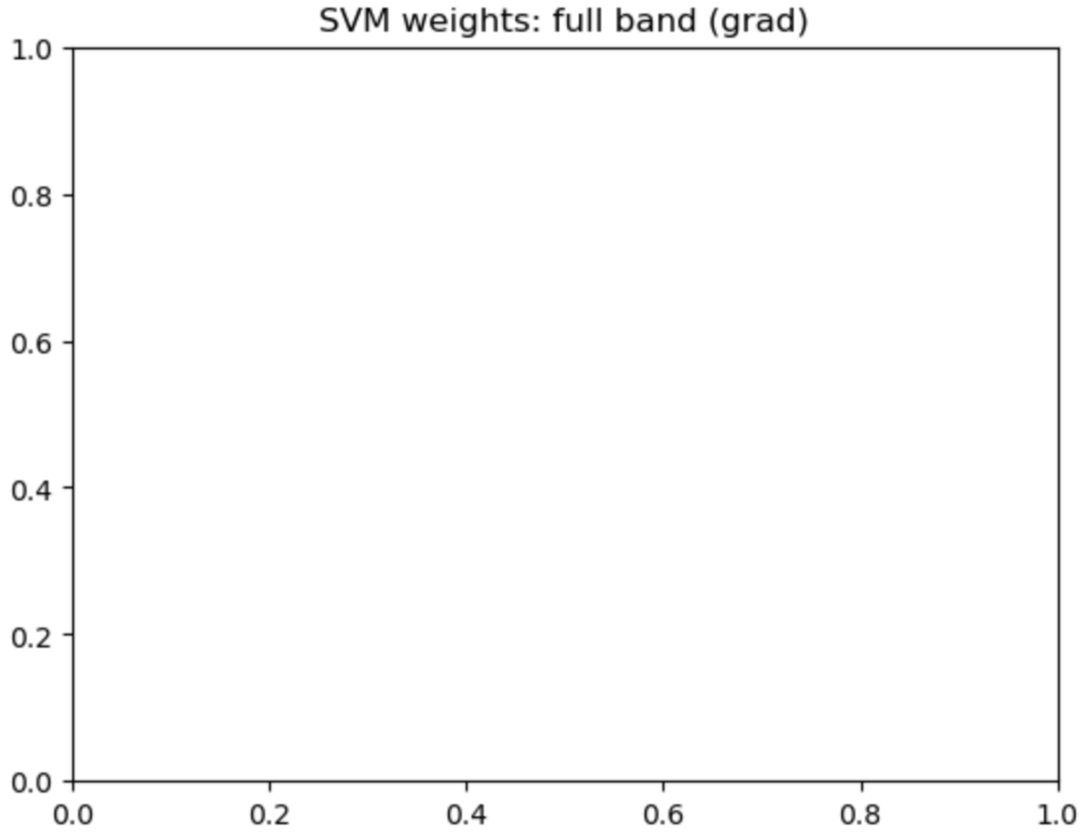
- **Target**: Classify **Part-of-Speech (POS)** → Noun vs. Verb
- **Features**:
  - Calculated using **PSD (power spectral density)** in each epoch
- **Epoch window**: 0–500 ms post word onset
- **Classifier**:
  - Linear SVM → SVC(kernel='linear', class_weight='balanced')
- **Cross-validation**:
  - Leave-One-Story-Out (LOSO-CV) → test generalization across stories
- **Evaluation**:
  - Classification accuracy per run
  - SVM weights → visualized as topographic maps → interpret spatial patterns

```
✅ Loaded data: X shape (20877, 306), y shape (20877,)
Number of NN: 13245, VV: 7632

LOSO-CV accuracy (balanced=True): 0.50 ± 0.14
Per run accuracy: [0.59590793 0.41239892 0.35443038 0.59413203 0.67559524 0.36085627
 0.58858859 0.37951807 0.61818182 0.64983165 0.38369305 0.67356322
 0.68539326 0.3253012  0.4056338  0.45348837 0.5738255  0.5887574
 0.4        0.63480392 0.41242938 0.32601881 0.39295393 0.38170347
 0.64482759 0.66233766 0.68852459 0.42156863 0.29545455 0.30350195
 0.62532982 0.64588529 0.30870712 0.60613811 0.35962145 0.40625
 0.34242424 0.62912088 0.30666667 0.33481153 0.58333333 0.35519126
 0.60057471 0.65337423 0.37393768 0.33136095 0.61142857 0.6870229
 0.43425076 0.63355408 0.62533693 0.71078431 0.3648294  0.38860104
 0.6601467  0.6408046  0.36513158 0.36050157]
```


Per-Run Classification Accuracy
Chance level (0.5)
Accuracy
Run Index
0.0
0.2
0.4
0.6
0.8
1.0

```
=== SVM weight summary ===
full band → mean weight: 0.0000, std: 0.0000, max: 0.0000, min: -0.0000
```





Summary of Results

- Part-of-Speech (POS) decoding from PSD appears challenging in this dataset
- Possible contributing factors:
  - Feature space may require further optimization
    → band selection, time window, feature reduction
  - Natural speech evokes complex, variable neural responses
  - POS categories (NN / VV) may not map to clear oscillatory patterns
- SVM in this form may not be sufficient → further tuning or alternative approaches needed

Summary & Reflections

- Learned to process and explore **MEG data**
- Built a pipeline for **data preparation** and **annotation alignment**
- Implemented **TRF analysis** → initial patterns observed
- Tried **SVM POS decoding** → currently at chance level
- Gained hands-on experience in handling complex data & testing models
- Future work:
  - optimize features, explore advanced analysis methods
  - Source analysis?

Thank BHS for fostering such an open and supportive ecosystem