An Analysis of a Multi-Modal Repeated Exemplar Confound

Arunachalesh Swaminathan

MSc CS648 Project

Data Science and Analysis

Department of Computer Science
Maynooth University, Co. Kildare, Ireland.

A thesis submitted in partial fulfillment of the requirements for Data Science and Analysis.

Supervisor(s): Prof. Barak Pearlmutter, Mr. Jack Kilgallen

Date: July 29,2024

Acknowledgements

I would like to extend the utmost thanks to my supervisors Prof. Barak Pearlmutter and Mr. Jack Kilgallen who have been incredibly helpful during this project. Their unwavering assistance and encouragement have been instrumental in the successful completion of this work. I would like to thank them for all their support and their guidance, without which this would not be as successful. I would like to thank Max Planck Institute for Psycholinguistics, and members of Simanova (2010) who provided us with the dataset and provided us the opportunity to go ahead with the project. I would like to extend my heartfelt gratitude to Prof. Peter Mooney, the project course coordinator for CS648, for his exceptional support and understanding. Prof. Mooney's generosity and support, even from a distance, have been greatly appreciated and instrumental in the successful completion of this work.

Contents

List Of Figures

Abstract

In functional neuroimaging research, particularly in electroencephalography (EEG) classification tasks, repeated use of a single stimulus is used to enhance the signal-to-noise ratio. However, this approach risks a methodological confound when the same stimulus appears in both training and testing datasets, potentially over-estimating performance should classifiers learn exemplar-specific rather than category-specific features. This project aims to investigate this issue by reviewing the scientific paper “Identifying Object Categories from Event-Related EEG: Toward Decoding of Conceptual Representations" (Simonova, 2010), utilizing the Simonova (2010) dataset, which demonstrated the capability of decoding conceptual information from EEG data across different modalities such as visual, auditory, and orthographic representations of objects. The data was tested with and without the effect of the confound on four models and for all the three modalities. The models provided lower accuracy when we reduced the repeated exemplar confound. The average accuracies dropped by 2% to 23% (raw percentage values) for various models and modalities. Having done various statistical analysis, it is shown that the confound has a significant effect on the model’s performance.

Chapter 1: Introduction

1.1 Introduction and motivation

In cognitive neuroscience, recognizing conceptual representations from neuroimaging data is a significant challenge. Understanding how the human brain decodes, and processes different concepts can provide deep insights into human cognition, perception, and language. Conceptual representation is fundamental for categorizing and making sense of the world around us, and these representations are vital to various cognitive functions, such as reasoning, decision making, and memory.

Neuroimaging techniques like fMRI (Functional Magnetic Resonance Imaging) and EEG (Electroencephalography) are widely used to study the neural basis of conceptual representation. While fMRI can pinpoint specific brain regions involved in processing different concepts, it lacks the ability to capture rapid neural dynamics. To track the timing of neural processes with precision, EEG is employed, as it offers high temporal resolution. EEG is particularly suited for studying the fast and dynamic nature of conceptual processing.

Despite the advantages of using EEG, decoding conceptual representation from EEG data presents several challenges. EEG signals consist of overlapping neural activities from various sources in the brain, making the data complex to interpret. Additionally, the relationship between these neural activities and the high-level cognitive processes involved in conceptual representation is not straightforward. Advanced computational methods, such as Multi-Variate Pattern Analysis (MVPA) and machine learning algorithms, have been utilized to decode patterns in EEG data and identify responses to different stimuli, thus mapping EEG data to corresponding conceptual categories.

Identifying confounds is essential for maintaining the validity and integrity of almost every scientific research. Confounds can skew results and weaken hypotheses by misrepresenting the real relationship between variables. By properly identifying and managing confounds, researchers can isolate the variables of interest, obtaining more accurate and trustworthy results.

This approach not only enhances the validity of individual studies but also advances the overall field of research.

This study builds on the work of Simanova et al. (2010), who used Bayesian Logistic Regression to classify conceptual categories from EEG recordings. Their research demonstrated the potential to decode semantic categories presented in various modalities, including visual, auditory, and orthographic stimuli. However, an important consideration in such studies, which is often overlooked, is the repeated exemplar leakage. This data leakage occurs when the same or similar examples are presented multiple times (same stimulus types), leading to familiarity effects. These effects can influence neural responses and classification performance, reflecting repeated exposure rather than the experimental variables of interest.

1.2 Overview of dissertation

This study aims to analyze the effects of these repeated exemplar leakages on their EEG data. By using two different data splitting methods which help us identify the effect of the confound, we can provide an accurate assessment of the model's performance. The primary objectives are to measure classifiers' performance across various modalities, statistically analyze the temporal features contributing to the performance, and propose methods to address the repeated exemplar leakage.

Chapter 2: Background

2.1 Literature

The base paper for our analysis is Simanova et al. (2010), and they utilized Bayesian logistic regression to classify EEG responses to stimuli presented in visual, auditory, and orthographic modalities. Their study provided support for the hypothesis that EEG data could be used to decode conceptual categories, establishing a basis for further exploration into the neural representations of concepts.

Other related studies have employed multivariate pattern analysis (MVPA) and machine learning techniques to decode neural responses from fMRI and EEG data. For instance, Mitchell et al. (2008) used fMRI data to predict word meanings based on neural activation patterns elicited by different semantic categories. Similarly, EEG-based studies (Huth et al. 2012 and Sudre et al. 2012) have explored the temporal dynamics of semantic processing, highlighting the potential of EEG for real-time applications.

Electroencephalography (EEG) is a non-invasive method used to record electrical activity of the brain through electrodes placed on the scalp. It provides a high temporal resolution view of neural activity, ideal for studying the dynamics of neural processes and capturing rapid changes. However, EEG has relatively low spatial resolution and can be contaminated by various sources of noise, such as muscle movements and eye blinks. Advanced preprocessing techniques are required to clean the data and isolate relevant neural signals. EEG signals are complex, consisting of multiple overlapping neural activities, necessitating sophisticated analytical methods for decoding.

Single-trial decoding is a process that involves analyzing EEG data on a trial-by-trial basis, capturing the variability and richness of neural responses to individual stimuli. This method is crucial for understanding the dynamics of brain activity and cognitive processes. Single-trial analysis allows researchers to study the variability in neural responses, providing insights into different brain states or cognitive strategies. It is essential for real-time applications such as brain-computer interfaces (BCIs) and neurofeedback systems, enabling immediate interpretation of neural signals. However, single-trial EEG data often have a low signal-to-noise ratio, requiring advanced techniques like machine learning algorithms and multivariate pattern analysis (MVPA) to improve decoding accuracy. Ensuring that decoding models generalize well to new, unseen trials is a critical challenge.

Repeated exemplar leakage occurs when participants are exposed to the same or similar stimuli multiple times, leading to familiarity effects that can skew the results of neural decoding studies. A reverse citation search for Simanova et al. (2010) found that many research papers that cited Simanova et al. (2010) have often utilized repeated exemplars in their designs.

Looking up research studies that were done before Simanova, we found that many studies faced the same confound. Haxby et al. (2001) used a set of 8 object categories with 8 exemplars per category in their fMRI study to investigate distributed representations of object categories in the ventral temporal cortex. Cox and Savoy (2003) used a set of 10 object categories with 6 exemplars per category in their fMRI study to classify object categories. Polyn et al. (2005) used 6 semantic categories with 6 exemplars per category in their fMRI study to decode the recalled category during a free recall memory task. Reddy et al. (2010) used 8 object categories with 8 exemplars per category in their fMRI study to investigate invariant object representation.

Grootswagers et al. (2016) used 8 object categories with 8 exemplars per category in their EEG study to investigate methods for decoding object representations. It is a paper that cited Simanova et al. (2010). Fahrenfort et al. (2012) used multiple object categories with repeated presentation s of exemplars in their EEG study to investigate object category decoding. Cichy et al. (2014) used 92 object images from 4 categories with multiple exemplars per category in their MEG and fMRI study to decode object representations. It is suspected of having repeated exemplar leakage. Isik et al. (2014) used 92 object images from 4 categories with multiple exemplars per category in their MEG study to investigate the time course of object recognition.

Despite the advancements in EEG-based semantic decoding, the issue of repeated exemplar leakage remains underexplored. While some studies acknowledge the potential for familiarity effects, few have systematically addressed this confound in the context of EEG-based semantic decoding. This project seeks to fill this gap by explicitly analyzing the impact of repeated exemplars on classification performance and proposing methods to mitigate this confound. By doing so, it aims to provide a more accurate assessment of model performance in classifying conceptual categories from EEG data. By addressing this repeated exemplar leakage, this project aims to enhance the accuracy and reliability of EEG-based decoding studies, ultimately providing a clearer understanding of how the brain processes and represents conceptual information.

2.2 Environment and Tools

To conduct this project, a combination of programming languages, software tools, and libraries were used. We used **Python** as our programming language**, Jupyter Notebook** and **Google Collab** as our IDE, various important libraries like **Scikit-learn, TensorFlow, MNE** for processing the data, training and testing the models.

Python was chosen for its versatility, extensive libraries, and community support. Python is widely used in data science and machine learning, making it suitable for any project. Jupyter notebook provided an interactive environment for coding, data analysis, and visualization. It allowed for iterative development and easy sharing of the code and results. However, we faced a versioning problem for the TensorFlow probability package which held me back from running the Bayesian Logistic Regression Model in Jupyter, that is where Google Collab came into picture. Google Collab is known to offer the computational resources we need to run models, particularly for handling large EEG datasets and running complex machine learning models. Google Collab also supports collaboration and sharing. It was primarily useful for running Bayesian Logistic Regression models with the compatible TensorFlow version and is recommended for quick execution.

MNE Library was useful in reading and processing EEG data. MNE is a powerful library for EEG and MEG analysis, providing functions for preprocessing, visualization, and statistical analysis of neuroimaging data. TensorFlow is a robust library for deep learning, offering flexibility and scalability for several types of neural network architectures. So it was utilized for building and training the Bayesian Logistic Regression model. Scikit-Learn is a widely used library in the machine learning community, providing easy-to-use tools for data mining and data analysis. It was used for tasks such as model training, evaluation, and analysis. They are employed for implementing traditional machine learning algorithms and various utility functions.

Throughout the project, there was a need to learn and master these tools, particularly the MNE library for EEG data processing and TensorFlow for model training. This involved gaining an understanding of EEG data structures, preprocessing techniques, and the implementation of machine learning algorithms. The learning process was supported by online resources, tutorials, and community forums (like GitHub and reddit), which provided valuable insights and solutions to challenges encountered during the project.

2.3 Dataset and Ethical Considerations

The dataset used in this project was provided by the Max Planck Institute for Psycholinguistics, with permission from member(s) of the Simanova et al. (2010) paper. The dataset included EEG recordings of participants responding to stimuli in visual, auditory, and orthographic modalities. Ethical considerations were adhered to, ensuring that the data was used in accordance with the permissions granted and maintaining the confidentiality and anonymity of the participants.

2.4 Contribution to the Field

This project contributes to the field of cognitive neuroscience by addressing the repeated exemplar confound in EEG-based semantic decoding studies. By implementing and comparing two data-splitting methods, the study provides insights into the impact of repeated exemplars on classification performance and proposes strategies to mitigate this confound. The findings enhance the reliability of EEG-based decoding studies and support the development of more accurate and generalizable models for brain-computer interface applications.

Chapter 3: Experimental Setup

3.1 Data Description

The dataset held by Max Planck's Institute of Psycholinguistics as per the paper Simanova et al (2010) was reported to comprise of EEG recordings from 24 native Dutch speakers who reported no psychological or neurological disorders. The dataset captures EEG responses to stimuli from three different modalities auditory, visual, and orthographic and three semantic categories namely Animals (non-target), Tools (non-target) and Task (target). The stimuli were repeated 80 times for each modality and were reported to be balanced. The visual and orthographic stimuli were reported to be shown for 300 ms followed by a blank screen for 1000-1200ms, whereas auditory stimuli were followed by a fixation cross screen for 1000-1200ms. The data was captured using a 64 channel ACTi Cap system at a sampling rate of 500 Hz filtering between 0.2-200Hz.

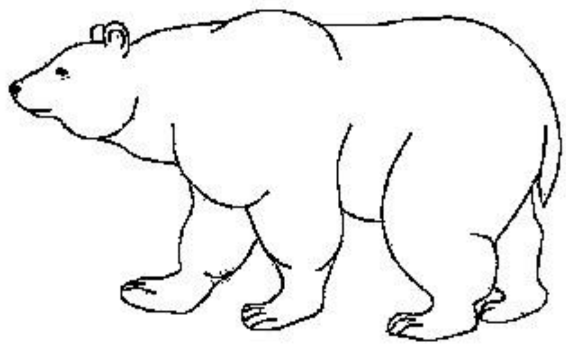


Fig. 1: Stimuli: Picture of a Bear shown to the subjects.

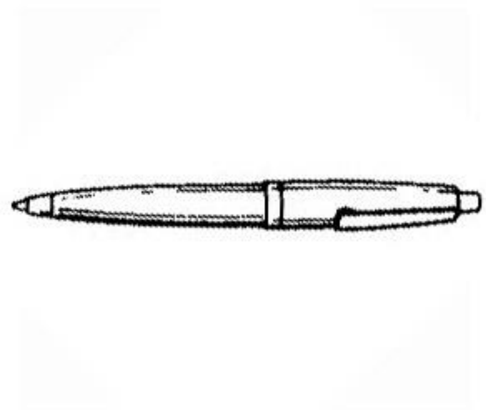


Fig. 2: Stimuli: Picture of a Pen shown to the subjects.

3.2 Data Preprocessing

Initially the EEG data files were not in a readable format, so we had to rename the files to their correct file format and then had repeated data (in multiple files for a single subject) which had to dealt with. After doing the basic cleaning and reading in the file, it came to our attention that the EEG data needed preprocessing as it contained noise and other artifacts.

3.2.1 Setting EEG Montage

The dataset included 64 channels, with 60 equidistant electrodes on an electrode cap, a reference electrode on the right mastoid, and an additional electrode on the left mastoid (excluding the ground electrode). After evaluating various inbuilt montages, we selected the Easycap-M10 montage for its closest match to the electrode positioning used by Simanova et al. (2010). The electrode positions were assigned according to the montage, and the linked mastoids reference and EOG channels were configured. Unwanted channels were removed, leaving only 60 channels and 1 EOG channel. The sensor positions were then plotted to verify the setup.

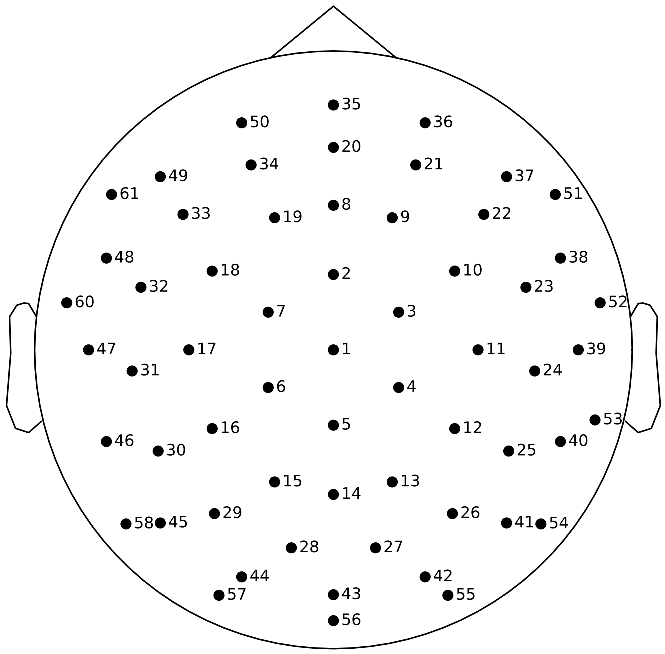


Fig.3: Easycap-M10 montage

The paper did not specify the correct montage, so we had to manually assign the closest montage (Easycap-M10) that we could find for the electrode placement that was followed by Simanova et al. (2010).

3.2.2 Noise Removal

Ocular artifacts, such as blinks, were removed using Independent Component Analysis (ICA) with the reference of EOG channels. First, EOG epochs were created and plotted to visualize the artifacts. Then, the data was passed through a bandpass filter with a passband of 1-30 Hz to remove high-frequency noise. ICA was performed to filtered data to identify and remove components corresponding to ocular artifacts.

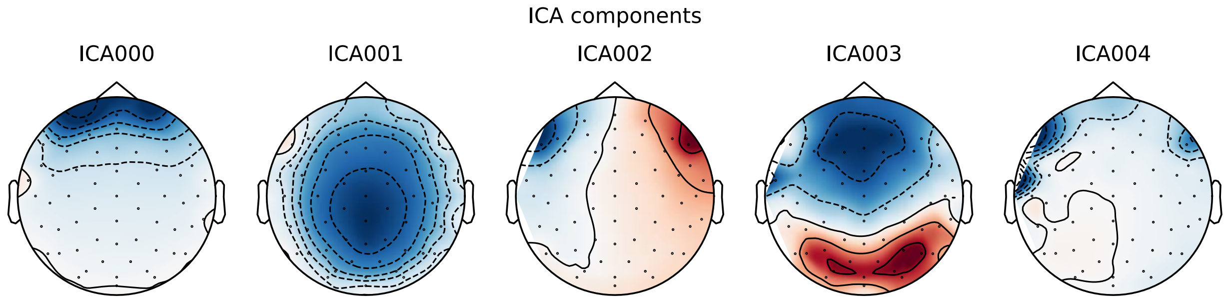


Fig.4: ICA Components

The figure shows the topographical maps of the first five Independent Components (ICs) identified through ICA. Each map represents the spatial distribution of the component across the scalp. These components are used to isolate specific patterns in the EEG data, such as ocular artifacts, which can then be removed from the signal to enhance data quality.

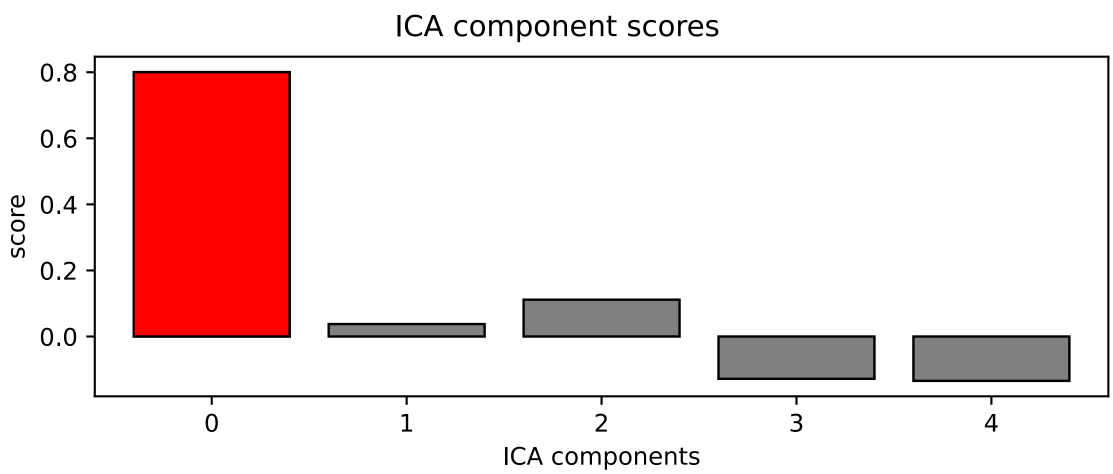


Fig.5: ICA Component Scores

The bar chart presents the scores of the ICA components. The red bar indicates the component with the highest score, likely corresponding to the most significant artifact, such as eye blinks. The gray bars represent other components' scores.

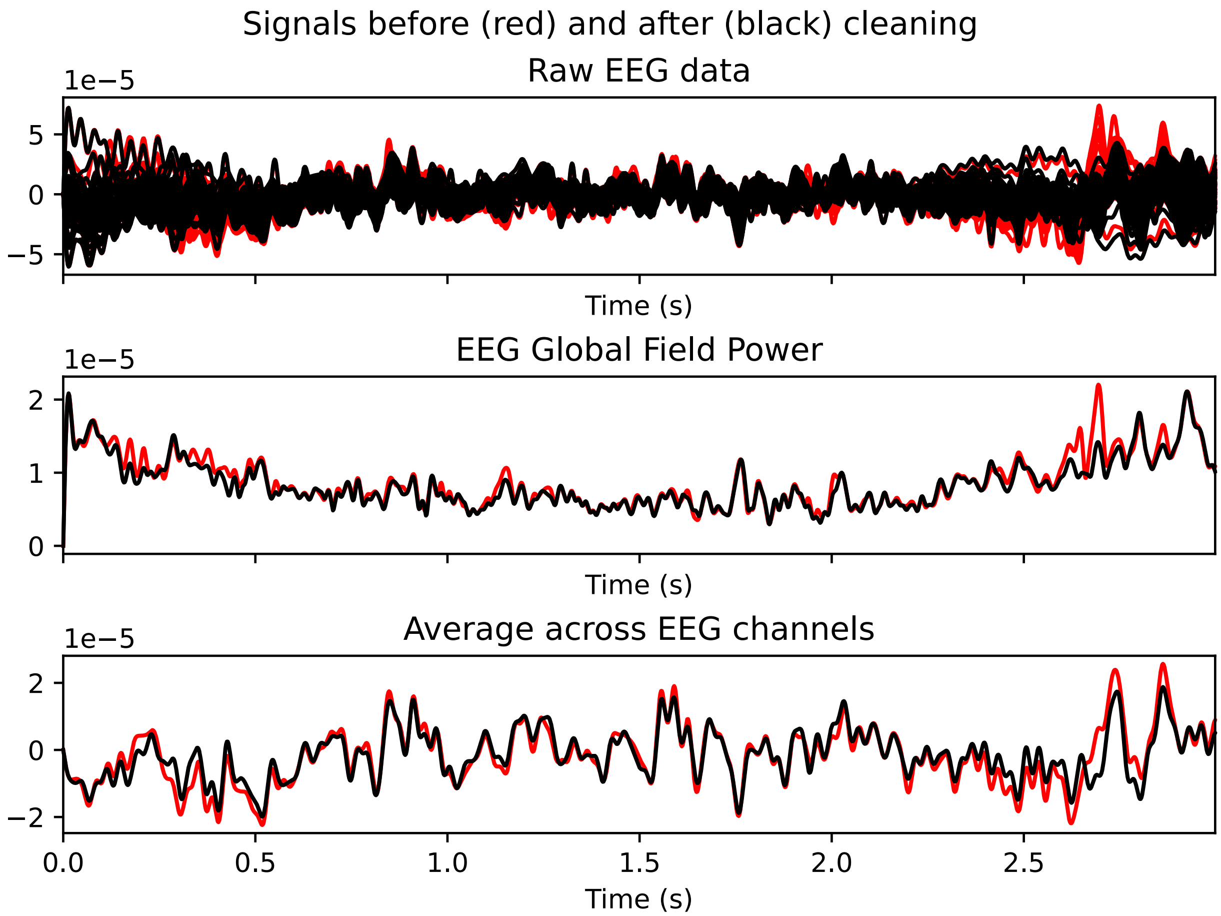


Fig 6: a) Raw EEGdata plot b) EEG Global Field Power plot c) Average across EEG channels plot

The plots show the clean data in black and unclean data in red which highlights the non-overlapped portions in red which are the noise and artifacts that are filtered. The Raw EEG plot shows the raw EEG signals. The EEG Global Field Power plot shows the overall electrical activity of the brain. The Average across EEG channels plot shows the average signal across all EEG channels. These graphs collectively signify the effectiveness of the cleaning process in improving EEG data quality.

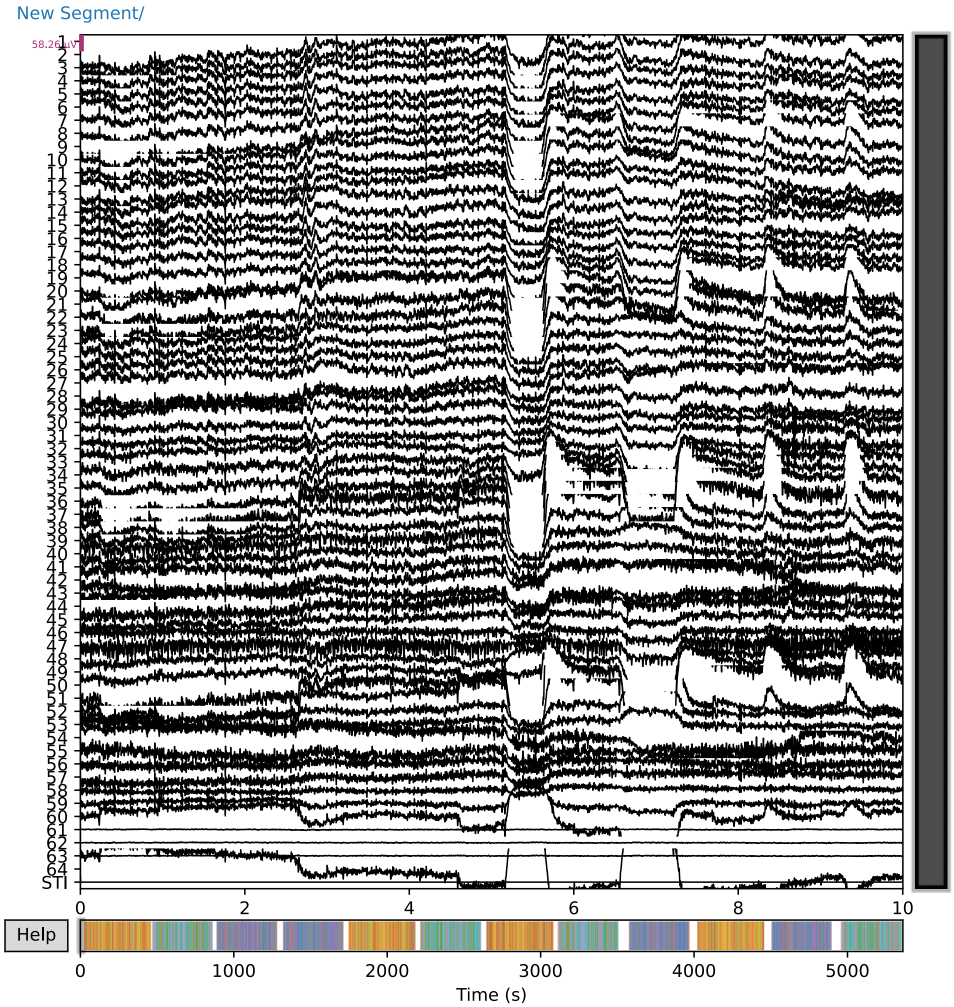


Fig.7: Unfiltered data for subject 15.

This plot shows raw EEG data for subject 15 before any preprocessing. The data includes various artifacts, like muscle movements and eye blinks, identifiable as irregular, large-amplitude fluctuations in the EEG signals.

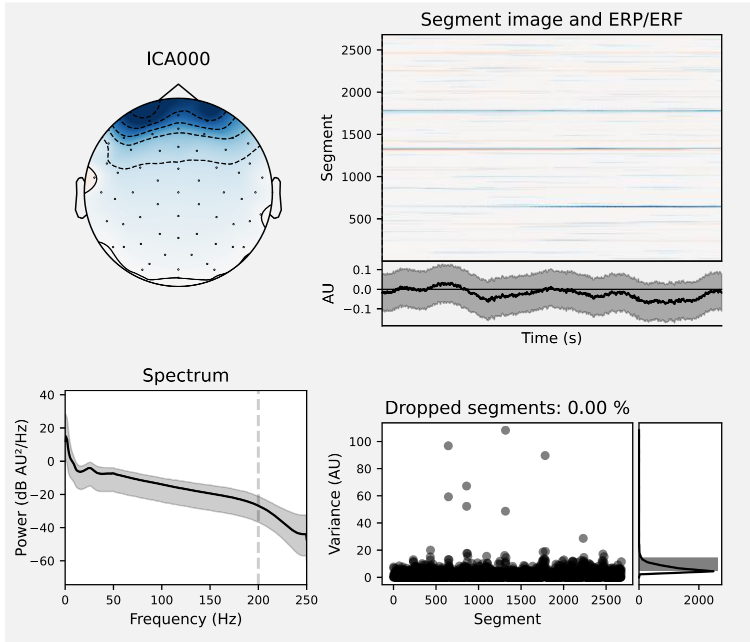


Fig. 8: ICA for Artifact Detection and Removal before filtering

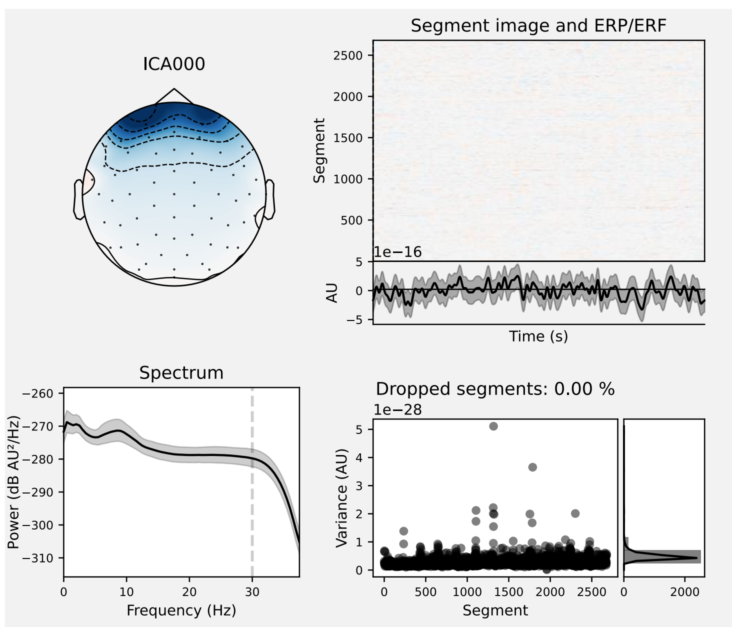


Fig.9: ICA plot for Artifact Detection and Removal after filtering

The graph before filtering shows more pronounced activity in the ERP/ERF plots, indicating more significant artifacts which are minimal or rarely found in the graph after filtering. The spectrum plot before filtering has broader values across frequencies while the plot after filtering shows a more concentrated range. The dropped segments graph before filtering shows higher variance in some segments, suggesting significant artifacts while the dropped segments graph after filtering with lower variance represents cleaner data with fewer artifacts.

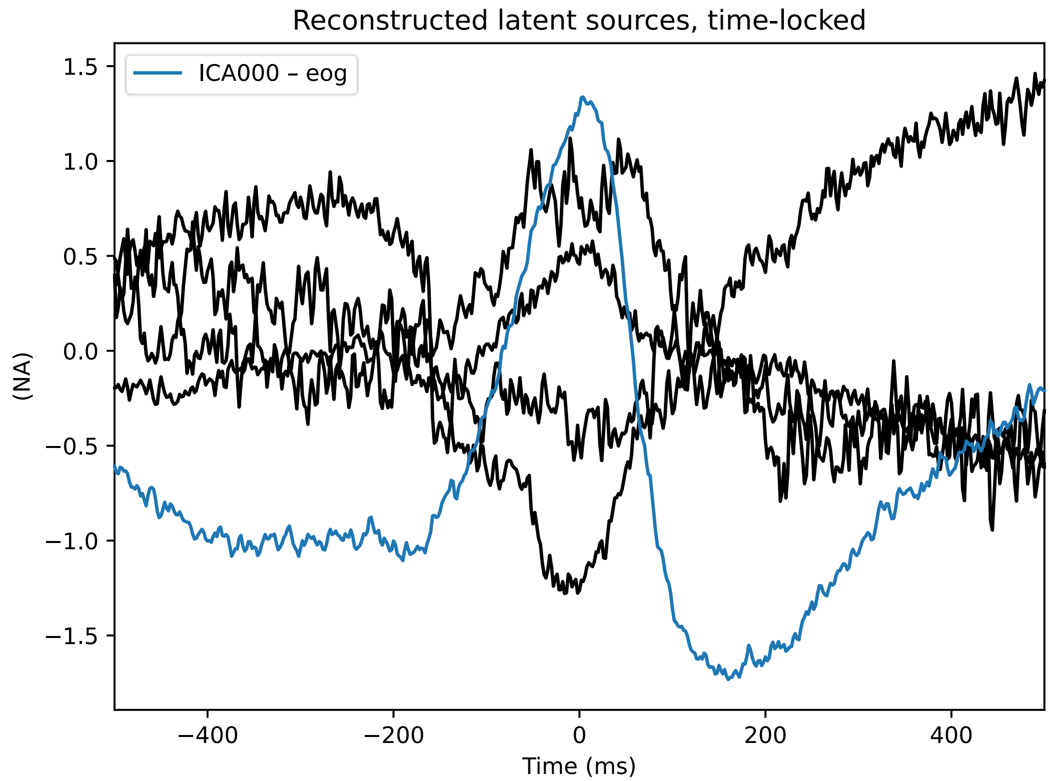


Fig. 10: Reconstructed Latent Sources, Time-Locked to EOG Events

This plot presents reconstructed latent sources time-locked to EOG events, specifically eye blinks. The blue line represents the EOG channel, and the black lines represent the independent components. This figure demonstrates how ICA isolated and removed eye blink artifacts from the EEG data.

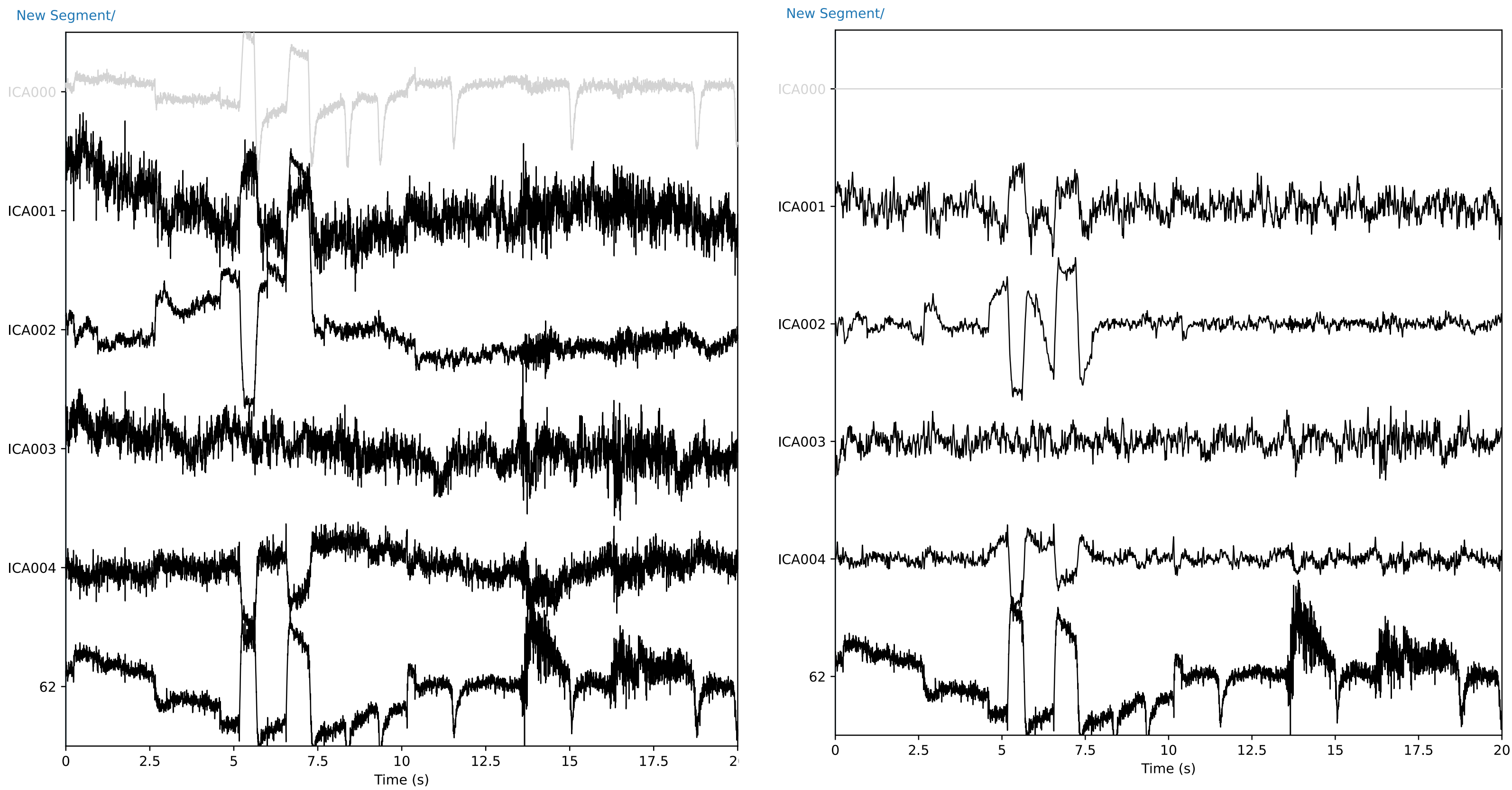


Fig.11: Time Series of Independent Components and EOG Channel before filtering (left) and after filtering (right)

The graph before filtering has more pronounced artifacts in both the EOG channel and ICA components, but the graph after filtering shows reduced artifacts. It also shows cleaner signals with less noise and artifacts compared to the unfiltered one.

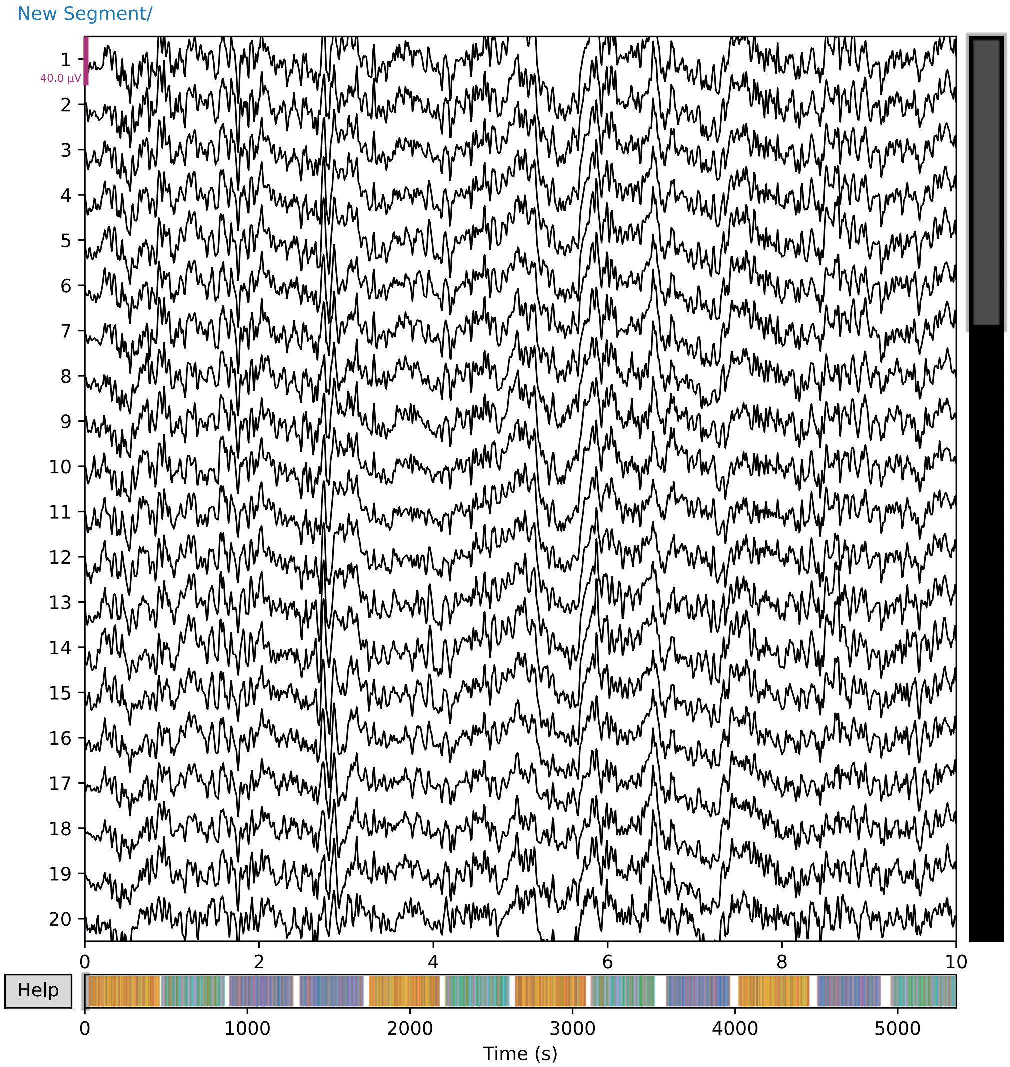


Fig.12: Filtered data for subject 15

The graph shows the EEG data that is obtained after it is cleaned using ICA and a bandpass filter.

3.2.3 Event Data Processing

Event data were mapped to their corresponding stimulus information (non_target/ category/ name/ modality). This involved defining mappings for targets, items, and modalities. A function was created to relabel event IDs based on these mappings. For example, event IDs were split into target, category, item, and modality components, and then combined into a comprehensive label.

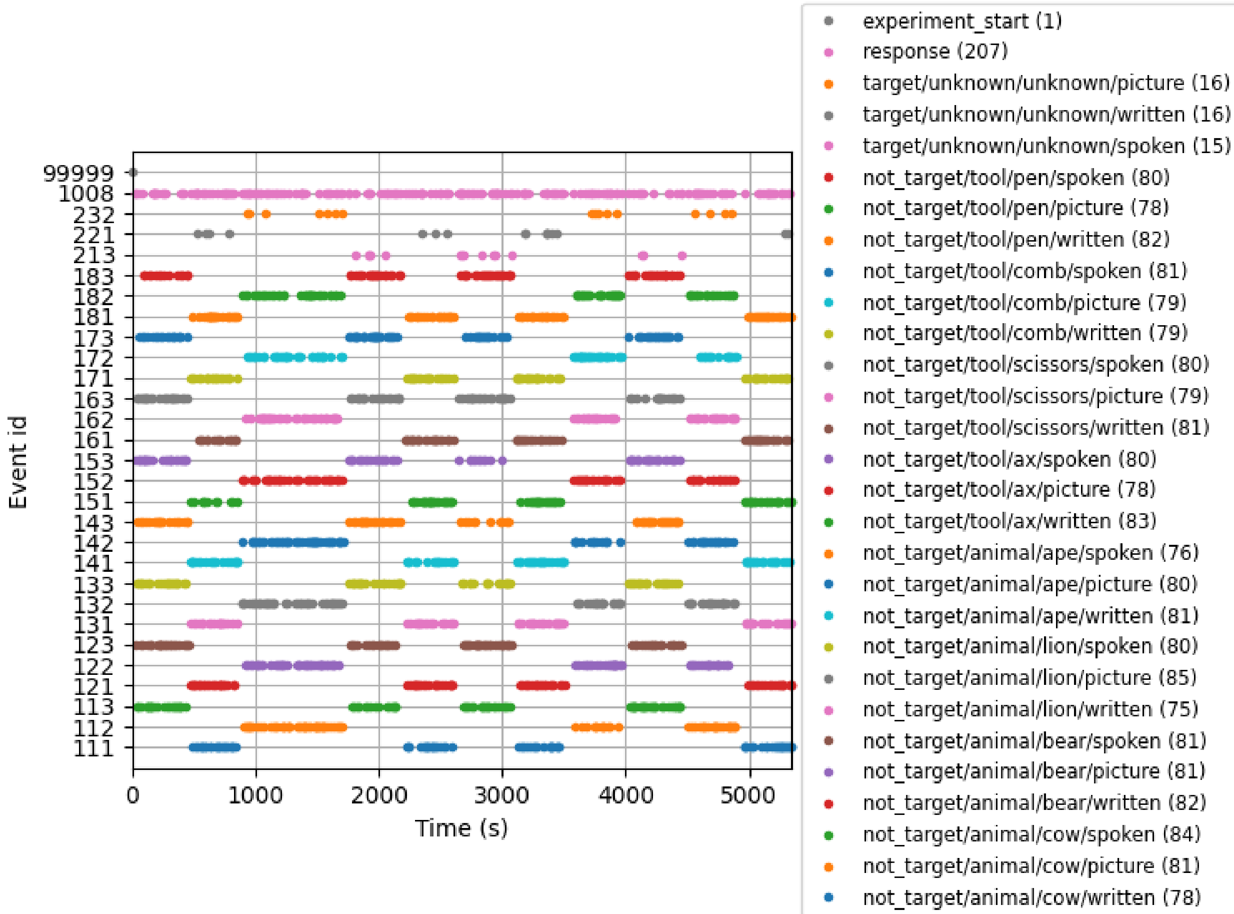
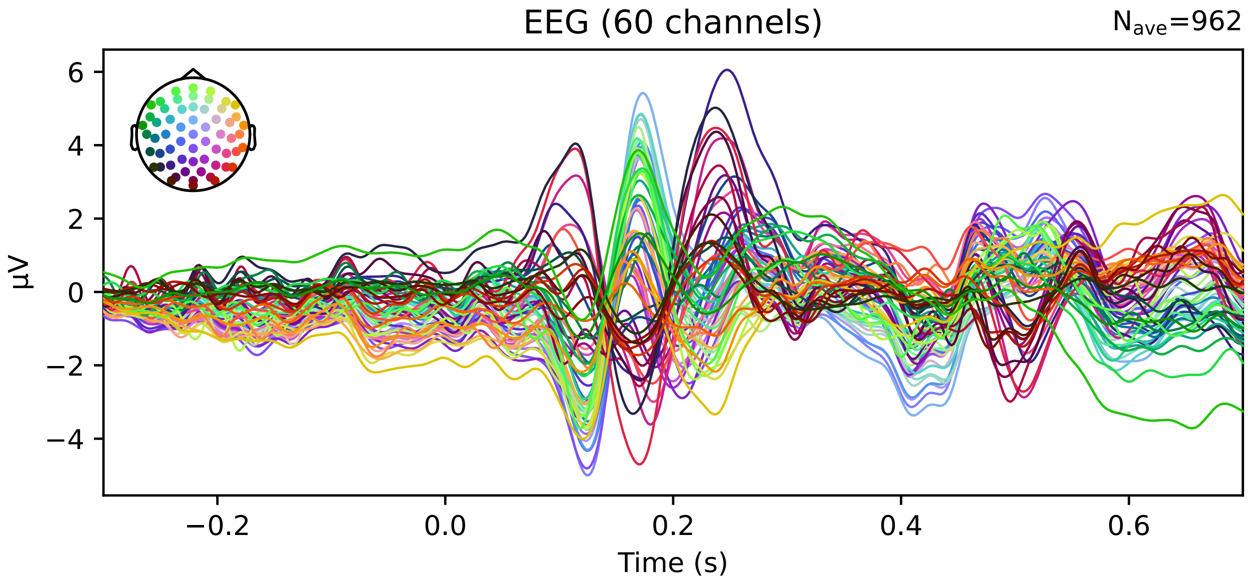


Fig.13: Event Timeline Visualization of EEG Experiment Data

The graph shows all the events with event IDs and the occurrences at different points of time for subject 15. The legend shows the list of all the labeled data (after event processing) along with the number of their occurrence in braces.

3.2.4 Data Epoching

The data was epoched using the mne.Epochs function, with bad epochs being rejected. Epochs were created around each event, with a time window from -0.3 to 0.7 seconds relative to the event. A rejection threshold was set to exclude epochs with voltage variations greater than 150 microvolts, which could indicate artifacts from eye movements, muscle activity, or other noise sources.



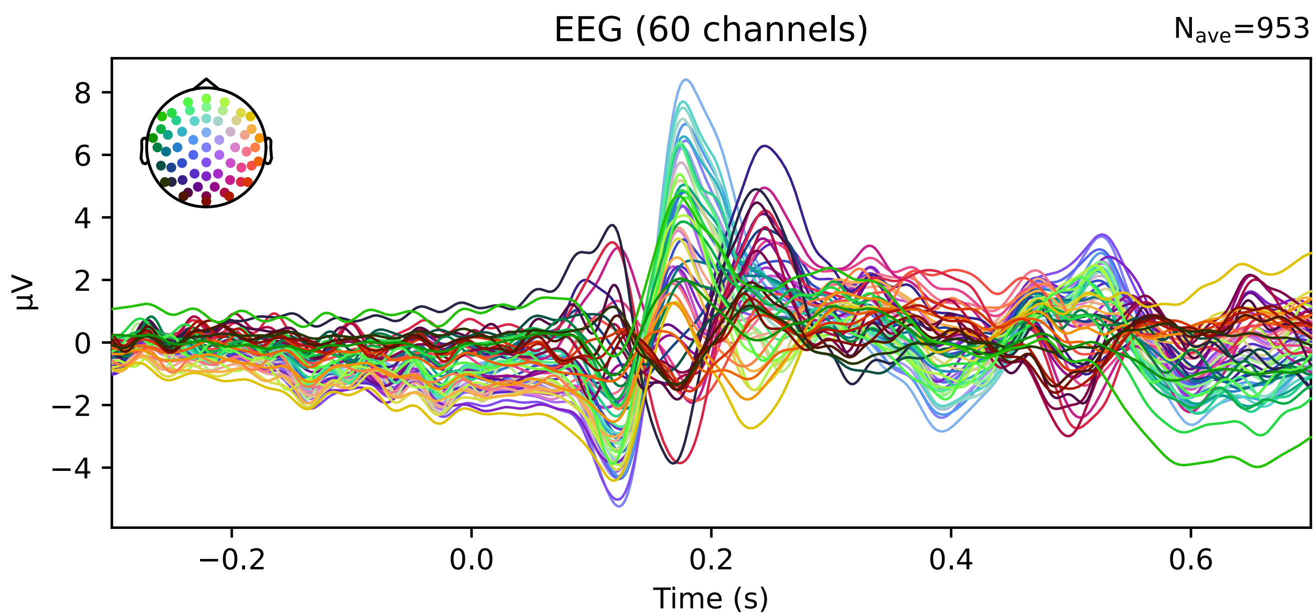


Fig.14: Average EEG Responses to Animal and Tool Stimuli

The EEG response to various trials that correspond to Animal and Tool Stimuli respectively.

3.3 Methods

After preprocessing, the data is stored in pickle files for convenient reuse. The processed EEG data and their various stimuli are extracted from the files. The dataset was found to be imbalanced on stimuli and was balanced so that there are equal number of trials for every stimulus type.

Data Splitting:

We devised two different methods of splitting the data for classification, one being Overlapping and the other named Disjoint.

- The Overlapping method has training data that contains trials from the same stimulus that overlap with the test set. Stratified K-Fold cross-validation with 4 folds was used. This ensures that each fold is representative of the entire dataset and maintains the class distribution within each fold.
- The Disjoint method has training data that does not contain trials from the same stimulus that are in the test set. The data were sorted in a specific order making sure that every fold has a balanced split (1 animal category stimulus trials and All 1 tool category stimulus trials). This ensures that the classifier is evaluated on completely unseen exemplars, providing a clearer assessment of its generalization capabilities and the potential impact of the repeated exemplar confound.

```
Overlapping
Stratified K-Fold split

Ape, Bear, Cow
Lion, Ax, Comb
Pen, Scissors
```

Fig.15 Overlapping Methodology Split

```
Disjoint Method:

Fold 1    Fold 2    Fold 3    Fold 4

Ape       Bear      Cow       Lion
Ax        Comb      Pen       Scissors
```

Fig.16: Disjoint Methodology split.

We used four models to observe the impact of the confound in each of these classifier models. The EEG data was fed into the models to evaluate their performance under both data splitting methods for each subject and each modality in that subject.

- Random Forest: An ensemble learning method that constructs multiple Decision tree classifier models during training and outputs the best (mode) or mean prediction of the individual tree classifiers. It is an ideal simple model for classification tasks.
- EEGNet: A compact convolutional neural network architecture designed specifically for EEG based brain-computer interfaces. It involves depth wise and separable convolutions to reduce computational cost while retaining reliable performance.
- DeepConvNet: A deep CNN architecture tailored for decoding brain signals. It uses multiple convolutional layers to capture spatial and temporal features from EEG data.
- Bayesian Logistic regression: This classifier is used in the base paper Simanova et al. (2010). This model employs a probabilistic approach to logistic regression, incorporating prior distributions to manage uncertainty in model parameters.

For each combination of modality and subject, the preprocessed data was balanced (same number of trials in each stimulus), binarized (output converted into Animal and Tool), label-encoded, reshaped, and standardized. This data was then used to train and test in each method for the classifier. The accuracies of each subject, modality under both overlapping and disjoint methodologies were recorded. Raw Differences in accuracies between methodologies, as well as Proportional differences, were calculated for each modality and subject. This was repeated for all four of the classifiers. These results were analyzed to understand the performance variations across different classifiers and methodologies. Some statistical tests were performed like the t-test with Bonferroni correction and t-test with Holm-Bonferroni correction to verify the significance of the results. Then a Mixed Linear Regression Model was fit to determine the impact of leakage on accuracies across different modalities.

Chapter 4: Results and Analysis

4.1 Results

The study used two data splitting methods to evaluate the effect of repeated exemplar leakage on EEG based classification tasks done with the two data-splitting methods. The results contain Average Overlapping Accuracy, Average Disjoint Accuracy, Raw Difference (Overlapping-Disjoint), Proportional Difference (Raw Difference/Overlapping Accuracy)

Classifier	Modality	Average Overlapping Accuracy	Average Disjoint Accuracy	Raw Difference	Proportional Difference
Random Forest	Picture	0.7034	0.6678	0.0355	0.0532
	Spoken	0.5978	0.5446	0.0532	0.0977
	Written	0.5784	0.5573	0.0211	0.0379
EEGNet	Picture	0.8658	0.7197	0.1461	0.2030
	Spoken	0.7102	0.5215	0.1887	0.3619
	Written	0.6108	0.5028	0.1081	0.2149
DeepConvNet	Picture	0.8618	0.6305	0.2313	0.3668
	Spoken	0.7040	0.5123	0.1917	0.3741
	Written	0.6425	0.5124	0.1302	0.2540
Bayesian Logistic Regression	Picture	0.7357	0.7164	0.0193	0.0269
	Spoken	0.5762	0.5322	0.0439	0.0826
	Written	0.5531	0.5216	0.0315	0.0604

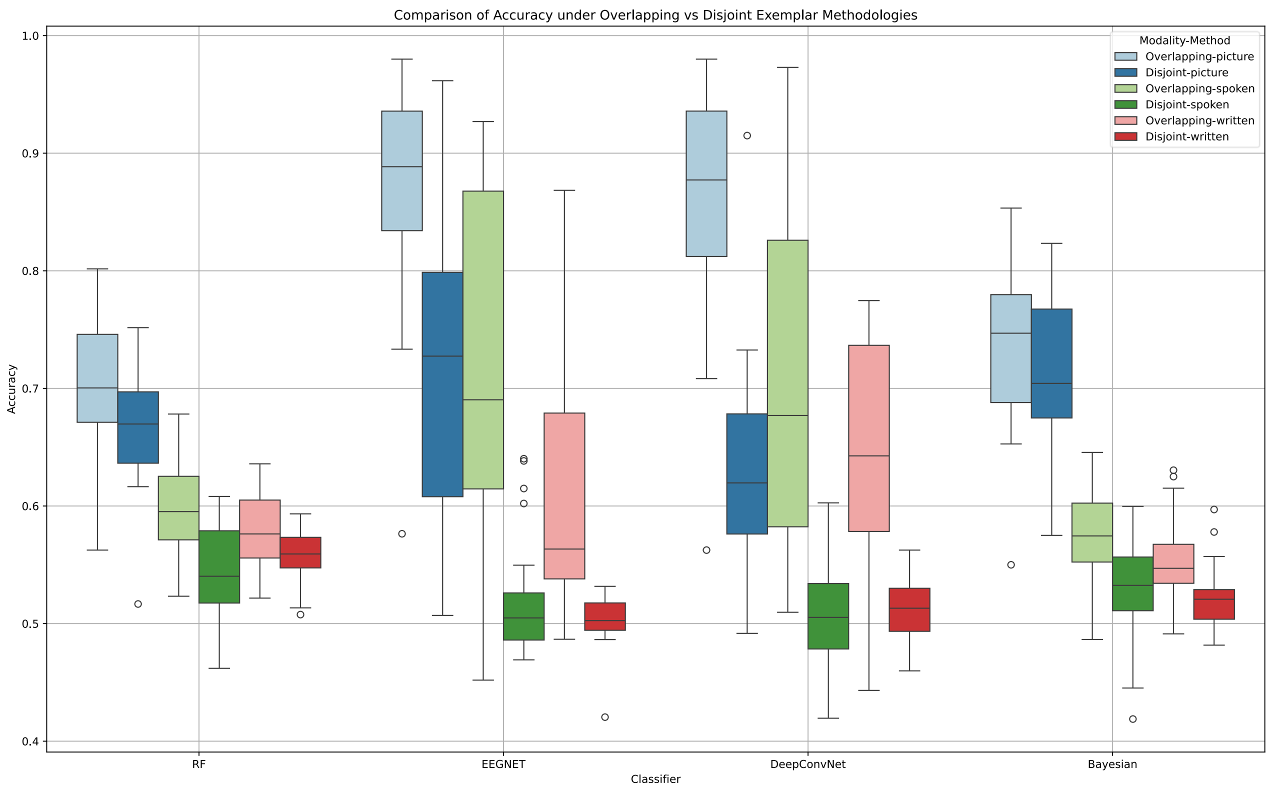


Fig 17: Boxplot for Comparison of Accuracies

The box plot shows the accuracies of various classifiers across different modalities comparing Overlapping and Disjoint methodologies. The boxplot is colored in paired palettes so that it is easy to compare the performance undern the two methodologies for each modality.

The results and the box plot indicate the substantial impact of the repeated exemplar leakage on the accuracy of EEG-based classification models. There is a noticeable difference in accuracy between the two methodologies (Overlapping and Disjoint). The raw difference in the accuracies varies from 0.0193 to 0.2313 (2% to 23%). The results obtained were complemented by the box plot, and they signified that there was considerable difference between the methods with and without leakage.

4.2 Analysis

For analyzing the performance of the models for every modality, paired t-test was done with Overlapping accuracies and Disjoint accuracies for every subject. The test was initially done with Bonferroni correction which was a conventional correction that might ignore the significance of a slightly significant model, then was done with a Holm-Bonferroni correction. The results are as follows:

Classifier	Modality	t-statistic	p-value	Significance (Bonferroni)	adjusted p-value	Significant (Holm-Bonferroni)
Random Forest	Picture	4.858222	8.382754e-05	Yes	2.768533e-09	Yes
Random Forest	Spoken	6.035737	5.451467e-06	Yes	4.223870e-06	Yes
Random Forest	Written	2.992204	6.944417e-03	No	6.394263e-06	Yes
EEGNet	Picture	6.682687	1.296035e-06	Yes	7.988256e-06	Yes
EEGNet	Spoken	7.250522	3.839882e-07	Yes	1.036828e-05	Yes
EEGNet	Written	4.711175	1.188634e-04	Yes	3.816027e-05	Yes
DeepConvNet	Picture	11.267003	2.307111e-10	Yes	5.029652e-04	Yes
DeepConvNet	Spoken	6.857282	8.875840e-07	Yes	5.943168e-04	Yes
DeepConvNet	Written	7.010041	6.394263e-07	Yes	1.932013e-03	Yes
Bayesian	Picture	2.898831	8.587420e-03	No	1.470408e-03	Yes
Bayesian	Spoken	4.124042	4.830033e-04	Yes	1.388883e-02	Yes
Bayesian	Written	4.117912	4.901358e-04	Yes	8.587420e-03	Yes

After the paired t-test, a mixed Linear Regression model was fitted on the accuracies to find the effect of leakage with modality as a fixed effect and classifier and subject as random effects. The results are given below:

```
                  Mixed Linear Model Regression Results
==========================================================================
Model:                 MixedLM     Dependent Variable:     disjoint_accuracy
No. Observations:      264         Method:                 REML
No. Groups:            4           Scale:                  29.2014
Min. group size:       66          Log-Likelihood:         -832.2744
Max. group size:       66          Converged:              Yes
Mean group size:       66.0
--------------------------------------------------------------------------
                              Coef.  Std.Err.    z    P>|z| [0.025 0.975]
--------------------------------------------------------------------------
overlapping_accuracy           0.611    0.027  22.887 0.000  0.559  0.664
overlapping_accuracy_spoken   -0.420    0.037 -11.419 0.000 -0.492 -0.348
overlapping_accuracy_written  -0.403    0.053  -7.641 0.000 -0.507 -0.300
Group Var                     11.600    1.597
==========================================================================
```

Fig. 18: Mixed Linear Model Regression Results

The results can be interpreted as

Acc_dis= (0.611*Acc_over) - (0.42*Acc_over_Mod_Spk) - (0.403*Acc_over_Mod_Writ) + (mu + psy)

where,

Acc_over is the accuracy with the leakage- Chance accuracy

Acc_dis is the accuracy without the leakage- Chance accuracy

Acc_over_Mod_Spk is a categorical variable that signifies if the trial is of a spoken modality.

Acc_over_Mod_Writ is a categorical variable that signifies if the trial is of a written modality.

mu is the random effect for the Classifier

psy is the random effect for the Subject

** Chance Accuracy =50%   Everything is in percentage.

The equation can be explained as follows

- If the trial is of spoken modality, a 1% increase in the overlapping accuracy (Acc_over), increases the disjoint accuracy (Acc_dis) by 0.611%, holding all other factors constant.
- If the trial is of a spoken modality, the disjoint accuracy (Acc_dis) increases by 0.191% (0.611-0.42%) for every 1% increase in overlapping accuracy (Acc_over), holding all other factors constant.
- If the trial is of a written modality, the disjoint accuracy (Acc_dis) increases by 0.208% (0.611-0.403%) for every 1% increase in overlapping accuracy (Acc_over), holding all other factors constant.

Chapter 5: Conclusions and Future work

5.1 Conclusions

The findings from this study demonstrate that repeated exemplar leakage significantly affects the accuracy of EEG-based classification models across various modalities and classifiers. The statistical analyses confirmed that removing this confound leads to a noticeable drop in accuracy, underscoring the importance of addressing it to achieve reliable and valid results. Specifically, the leakage affects the classifiers' ability to generalize, with EEGNet and DeepConvNet showing the most substantial drops in accuracy for picture and spoken modalities. Random Forest and Bayesian Logistic Regression classifiers showed smaller reductions, indicating they are either less sensitive to the confound or better at generalizing. Mitigating the repeated exemplar confound is essential for developing robust EEG-based classifiers that accurately reflect the underlying neural processes. Addressing this issue will not only enhance the reliability of future EEG studies but also contribute to the advancement of cognitive neuroscience by providing more accurate insights into brain function.

5.2 Future Work

Mitigation Strategies for Repeated Exemplar Confound:

To prevent models from overfitting to recurring patterns, strategies are needed to address the repeated exemplar confound. In order to decrease the model's dependence on particular exemplars, our future research should concentrate on sophisticated data augmentation approaches that produce a variety of realistic training data variants. More generalized models may be produced with the use of methods like adversarial training, augmentation with noise, and transformations. Furthermore, incorporating these methods with current training procedures and assessing their effectiveness in other modalities will be essential for creating reliable EEG-based classifiers that can handle a variety of datasets.

Transfer Learning and Domain Adaptation:

This is a promising approach to enhance the generalization of EEG based classifiers across different datasets. We may fine-tune pre-trained models to specific tasks and reduce the requirement for extensive labeled data by using these models on big, diversified datasets. In order to resolve disparities brought about by the repeated exemplar confound, domain adaptation approaches can assist in aligning the feature distributions across the source and target domains. Subsequent studies investigate the optimization of these methods for EEG data, with the possibility of integrating domain-specific modifications to enhance performance and robustness in diverse experimental and real-world contexts.

References

1. Simanova I, van Gerven M, Oostenveld R, Hagoort P (2010) Identifying Object Categories from Event-Related EEG: Toward Decoding of Conceptual Representations. PLoS ONE 5(12): e14465. https://doi.org/10.1371/journal.pone.0014465
2. Kilgallen, J. A., Pearlmutter, B. A., & Siskind, J. M. Repeated Exemplar Leakage in EEG Category Decoding. https://www.cs.nuim.ie/research/csworkshop/2024/abstracts/JackEatonKilgallen.pdf
3. Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K. M., Malave, V. L., Mason, R. A., & Just, M. A. (2008). Predicting Human Brain Activity Associated with the Meanings of Nouns. Science, 320(5880), 1191–1195. https://doi.org/10.1126/science.1152876
4. Huth, A. G., Nishimoto, S., Vu, A. T., & Gallant, J. L. (2012). A Continuous Semantic Space Describes the Representation of Thousands of Object and Action Categories across the Human Brain. Neuron, 76(6), 1210–1224. https://doi.org/10.1016/j.neuron.2012.10.014
5. Sudre, G., Pomerleau, D., Palatucci, M., Wehbe, L., Fyshe, A., Salmelin, R., & Mitchell, T. (2012). Tracking neural coding of perceptual and semantic features of concrete nouns. NeuroImage, 62(1), 451–463. https://doi.org/10.1016/j.neuroimage.2012.04.048
6. Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, *15*(5), 056013. https://doi.org/10.1088/1741-2552/aace8c
7. Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, *38*(11), 5391–5420. https://doi.org/10.1002/hbm.23730
8. Fahrenfort, J. J., Snijders, T. M., Heinen, K., van Gaal, S., Scholte, H. S., & Lamme, V. A. (2012). Neuronal integration in visual cortex elevates face category tuning to conscious face perception. Proceedings of the National Academy of Sciences, 109(52), 21504-21509. https://doi.org/10.1073/pnas.120741411

9. Cichy, R. M., Pantazis, D., & Oliva, A. (2014). Resolving human object recognition in space and time. Nature Neuroscience, 17(3), 455-462. doi:10.1038/nn.3635

10. Isik, L., Meyers, E. M., Leibo, J. Z., & Poggio, T. (2014). The dynamics of invariant object recognition in the human visual system. Journal of Neurophysiology, 111(1), 91-102. doi:10.1152/jn.00394.2013

11. Grootswagers, T., Wardle, S. G., & Carlson, T. A. (2016). Decoding dynamic brain patterns from evoked responses: A tutorial on multivariate pattern analysis applied to time series neuroimaging data. Journal of Cognitive Neuroscience, 29(4), 677-697. doi:10.1162/jocn_a_01068

12. Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science, 293(5539), 2425-2430 doi:10.1126/science.1063736

13. Cox, D. D., & Savoy, R. (2003). fMRI Brain Reading: detecting and classifying distributed patterns of fMRI activity in human visual cortex. NeuroImage, 19(2), 261-270. https://doi.org/10.1016/S1053-8119(03)00049-1

14. Polyn, S. M., Natu, V. S., Cohen, J. D., & Norman, K. A. (2005). Category-specific cortical activity precedes retrieval during memory search. Science, 310(5756), 1963-1966. DOI: 10.1126/science.1117645

15. Reddy, L., Tsuchiya, N., & Serre, T. (2010). Reading the mind's eye: decoding category information during mental imagery. Neuroimage, 50(2), 818-825. https://doi.org/10.1016/j.neuroimage.2009.11.084

Appendix

GitHub Repository link for Code and Data (Readme file):

https://github.com/Sarunachalesh/Project

Google Drive link for Unformatted files:

https://drive.google.com/drive/folders/1WERYgDuWKuYe9GVRxFb8SVaj23ZIwsh0?usp=sharing

Google Drive link for Preprocessed files:

https://drive.google.com/drive/folders/1HKl8fgbyxnQEJkv_4s9Aza7DEjMGIfg-?usp=sharing

The output files (files that contain the metrics) are part of the Zip file.

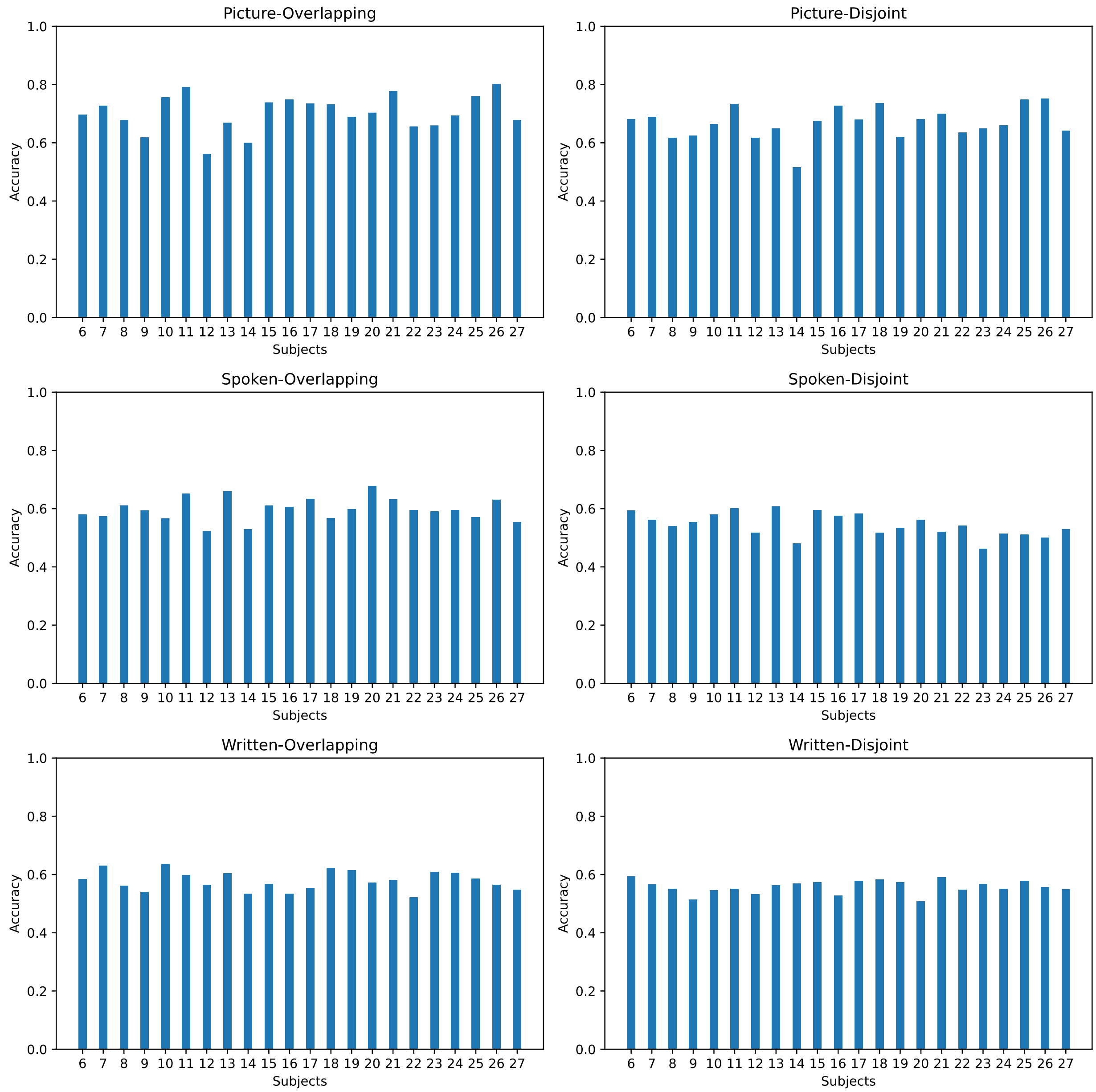
Random Forest

Code:

Disjoint methodology code:

```
# Disjoint method function
def RF_disjoint_cross_validation(X, y, i, modality):
    avg_acc = 0
    accuracies = []
    results = []

    # KFold cross-validation splits
    kf = KFold(n_splits=4, shuffle=False)
    fold = 1
    for train_index, test_index in kf.split(X):
        X_train, X_test = X[train_index], X[test_index]
        y_train, y_test = y[train_index], y[test_index]

        # Reshaping and Standardizing
        X_train = X_train.reshape(X_train.shape[0], -1)
        X_test = X_test.reshape(X_test.shape[0], -1)

        sc = StandardScaler()
        X_train = sc.fit_transform(X_train)
        X_test = sc.transform(X_test)

        # Train and evaluate the RandomForestClassifier
        rf = RandomForestClassifier(n_estimators=100, random_state=45)
        rf.fit(X_train, y_train)

        y_pred = rf.predict(X_test)
        cm = confusion_matrix(y_test, y_pred)
        cr = classification_report(y_test, y_pred, output_dict=True)
        accuracy = accuracy_score(y_test, y_pred)
        avg_acc += accuracy
        accuracies.append(accuracy)
        fold += 1
```

Performance measure:



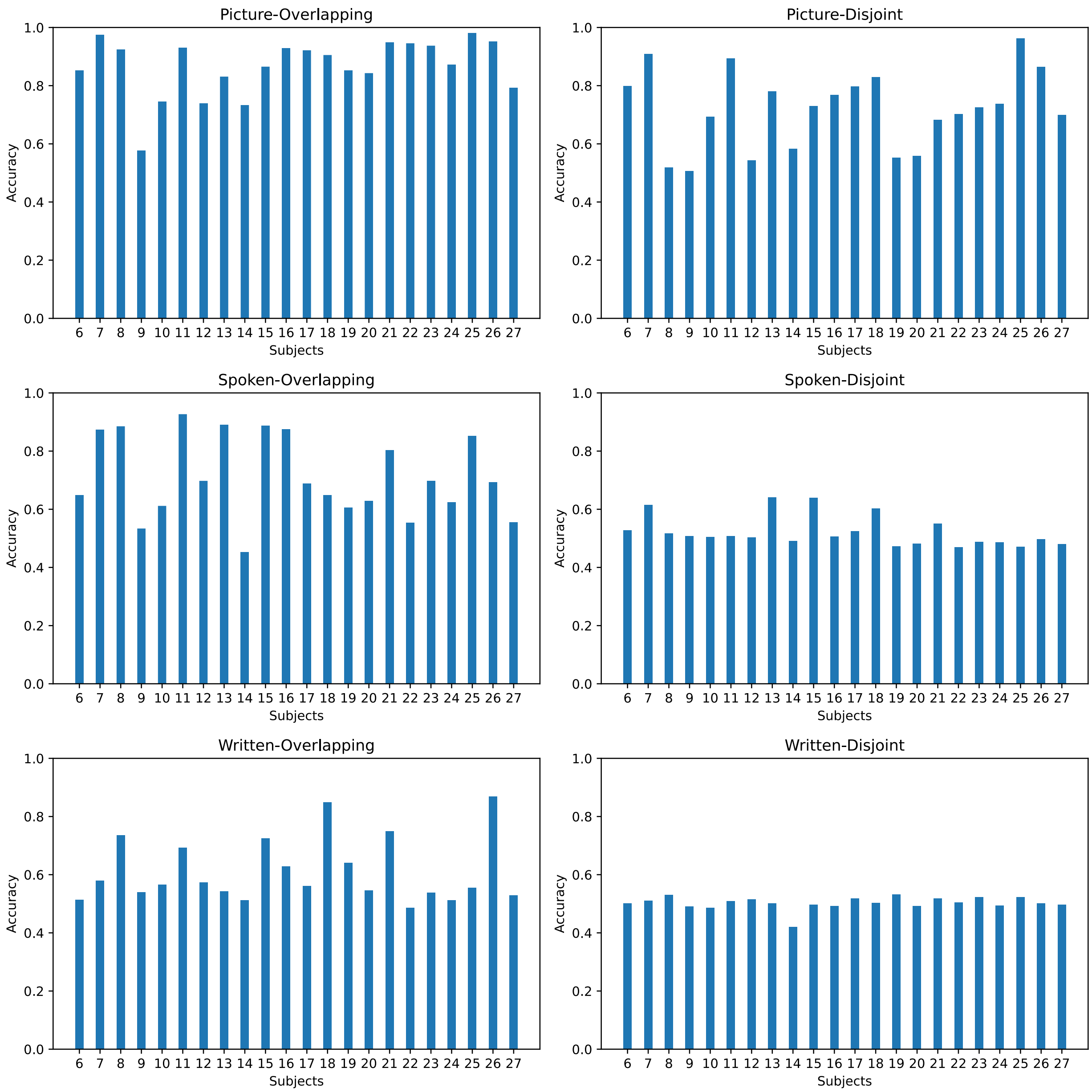
EEGNET

Code:

Model definition:

```
def build_eegnet_model(nb_classes, Chans=64, Samples=128, dropoutRate=0.5):
    input_shape = (Chans, Samples, 1)
    model = Sequential()
    model.add(Conv2D(16, (1, 64), padding='same', input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(DepthwiseConv2D((Chans, 1), use_bias=False, depth_multiplier=2,
depthwise_constraint=max_norm(1.)))
    model.add(BatchNormalization())
    model.add(Activation('elu'))
    model.add(AveragePooling2D((1, 4)))
    model.add(Dropout(dropoutRate))
    model.add(SeparableConv2D(16, (1, 16), use_bias=False, padding='same'))
    model.add(BatchNormalization())
    model.add(Activation('elu'))
    model.add(AveragePooling2D((1, 8)))
    model.add(Dropout(dropoutRate))
    model.add(Flatten(name='flatten'))
    model.add(Dense(nb_classes, name='dense', kernel_constraint=max_norm(0.25)))
    model.add(Activation('softmax', name='softmax'))
    return model
```

Performance measure:



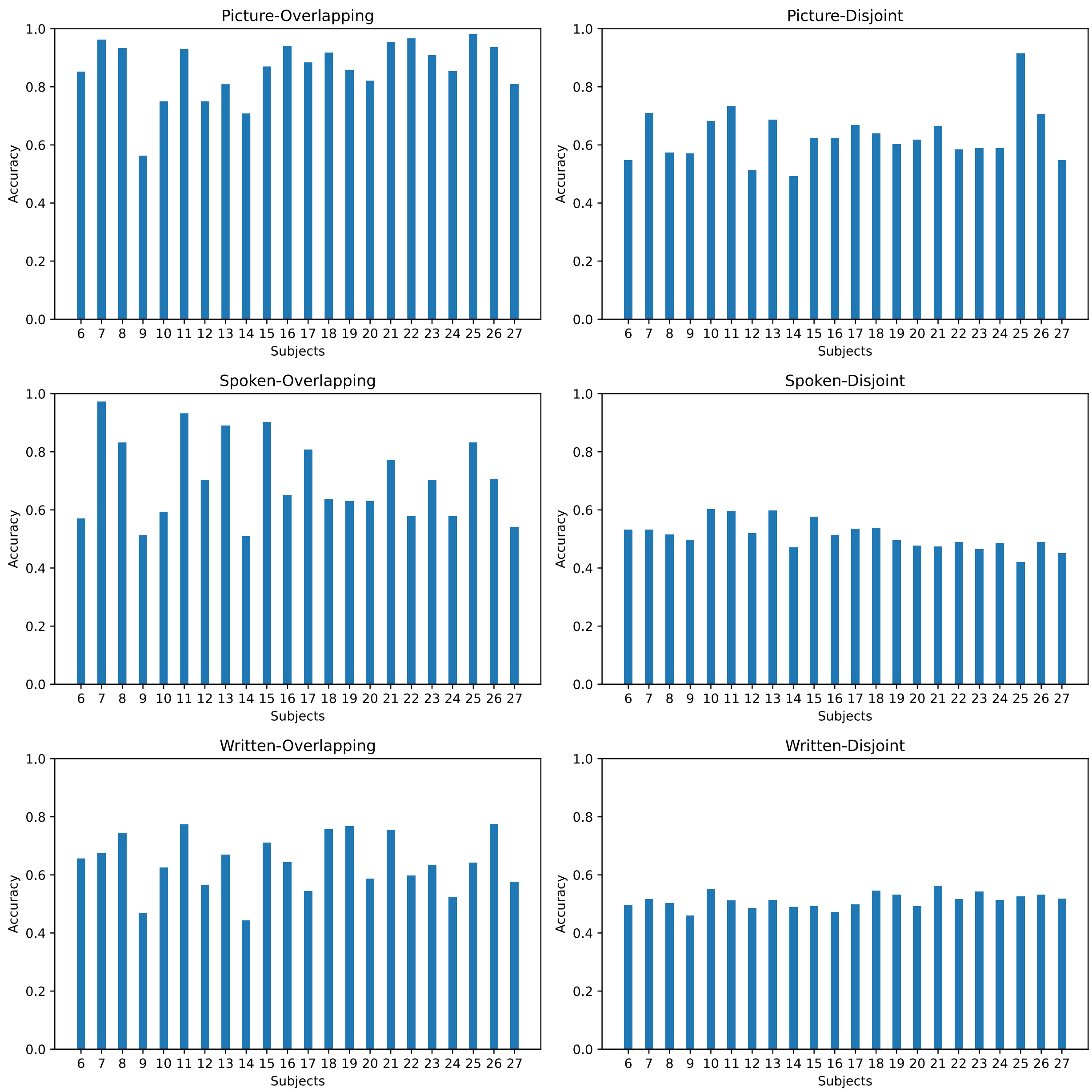
DeepConvNet

Code:

Model definition:

```python
def build_deepconvnet_model(nb_classes, Chans=64, Samples=128, dropoutRate=0.5):
    input_shape = (Chans, Samples, 1)
    model = Sequential()
    model.add(Conv2D(25, (1, 5), input_shape=input_shape, padding='same'))
    model.add(Conv2D(25, (Chans, 1), padding='valid'))
    model.add(BatchNormalization())
    model.add(Activation('elu'))
    model.add(MaxPooling2D(pool_size=(1, 2)))
    model.add(Dropout(dropoutRate))

    model.add(Conv2D(50, (1, 5), padding='same'))
    model.add(BatchNormalization())
    model.add(Activation('elu'))
    model.add(MaxPooling2D(pool_size=(1, 2)))
    model.add(Dropout(dropoutRate))

    model.add(Conv2D(100, (1, 5), padding='same'))
    model.add(BatchNormalization())
    model.add(Activation('elu'))
    model.add(MaxPooling2D(pool_size=(1, 2)))
    model.add(Dropout(dropoutRate))

    model.add(Conv2D(200, (1, 5), padding='same'))
    model.add(BatchNormalization())
    model.add(Activation('elu'))
    model.add(MaxPooling2D(pool_size=(1, 2)))
    model.add(Dropout(dropoutRate))

    model.add(Flatten())
    model.add(Dense(nb_classes, activation='softmax'))
    return model
```

Performance measure:



Bayesian Logistic Regression

Code:

Model definition:

```
def build_bayesian_logistic_regression_model(input_shape):
    inputs = tf.keras.Input(shape=(input_shape,))
    dense = tfp.layers.DenseFlipout(1, activation='sigmoid',
kernel_divergence_fn=lambda q, p, _: tfp.distributions.kl_divergence(q, p) /
X.shape[0])(inputs)
    model = tf.keras.Model(inputs=inputs, outputs=dense)

    model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),
                  loss=tf.keras.losses.BinaryCrossentropy(),
                  metrics=['accuracy'])
    return model
```

Performance measure: