JYVÄSKYLÄN YLIOPISTO
MATEMATIIKAN JA TILASTO-
TIETEEN LAITOS

MASTER'S THESIS IN STATISTICS AND DATA SCIENCE

Penalized Canonical Correlation Analysis for MEG Data

Juuso Koskinen

October 28, 2024

AMICA VERITAS
JYVÄSKYLÄN YLIOPISTO

Author Juuso Koskinen	
Title Penalized Canonical Correlation Analysis for MEG Data	
Degree Program Master's Degree Program in Statistics and Data Science	
Date 28.10.2024	**Page Count** 39 + 53 appendix pages

Abstract

Canonical correlation analysis is a statistical method used to examine linear relationships between two sets of variables measured on the same statistical units, by forming highly correlated linear combinations of the variables in each set. This method cannot be used in the context of high-dimensional data, where the number of variables in either variable set exceeds the sample size. In this setting, sparse canonical correlation analysis (SCCA) can be utilized to perform regularized canonical correlation for high-dimensional data, producing sparse solutions more feasible for interpretation.

In this thesis SCCA was used to explore the associations between temperamental traits and interoception. Temperamental traits decribe a person's dispositional responses to changes in their environment, while interoception refers to a person's sensitivity to stimuli originating from inside their own body, such as heart beat. Both of these attributes have a neurobiological basis, and some temperamental traits, especially ones related to anxiety have been found to be linked to interoceptive sensitivity. A data set consisting of magnetoencephalography (MEG) measurements of neuronal activity recorded during an interoception task and temperament questionnaire answers from 28 subjects was analyzed using SCCA with and without penalization in high dimensional setting, and after dimension reduction achieved by principal component analysis (PCA).

While a pattern of higher α-oscillation activity during an interoception task in the left parietal and right frontal lobe associated with lower scores on the Beck Anxiety Inventory and Fun seeking section of Behavioral Activation Scale, and higher α-activity in the left frontal lobe associated with higher scores on the same questionnaires was observed, no statistically significant canonical pairs were found based on permutation tests. SCCA was found to ease interpretation of the canonical coefficients of the questionnaire variables

via sparse coefficients, but overly sparse coefficients for MEG variables can hinder interpretation, as the spatial resolution of MEG is not enough to discern small areas of neuronal activation. For this reason larger areas of brain activation are preferred and canonical coefficients gained through PCA can be more useful for interpretation.

Keywords: interoception, temperamental trait, magnetoencephalography, canonical correlation analysis, lasso, penalized canonical correlation analysis

Tekijä Juuso Koskinen	
Otsikko Sakotetun kanonisen korrelaatioanalyysin sovellus MEG-datalle	
Tutkinto-ohjelma Tilastotieteen ja datatieteen maisteriohjelma	
Päivämäärä 28.10.2024	**Sivumäärä** 39 + 53 liitesivua

Tiivistelmä

Kanoninen korrelaatioanalyysi on tilastollinen menetelmä, jota voidaan hyödyntää kahden samoilta tilastoyksiköiltä mitattujen muuttujajoukkojen välisten lineaaristen yhteyksien tarkastelemiseen, muodostamalla korkeasti korreloituneita lineaarikombinaatioita kummankin joukon muuttujista. Tätä menetelmää ei voida käyttää korkeaulotteisen datan yhteydessä, jossa jommankumman muuttujajoukon muuttujien määrä ylittää datan otoskoon. Tässä tilanteessa harvaa kanonista korrelaatioanalyysiä voidaan hyödyntää sakotetun kanonisen korrelaatioanalyysin toteuttamiseksi korkeaulotteiselle datalle, tuottaen harvoja, tulkinnallisesti käyttökelpoisempia tuloksia.

Tässä tutkielmassa harvaa kanonista korrelaatioanalyysiä käytettiin temperamenttipiirteiden ja interoseption välisten yhteyksien tutkimiseen. Temperamenttipiirteet kuvaavat henkilön taipumuksellisia reaktioita ympäristön muutoksiin, kun taas interoseptio kuvaa henkilön herkkyyttä reagoida oman kehon sisältä peräisin oleviin ärsykkeisiin, kuten sydämen sykkeeseen. Molemmilla näistä ominaisuuksista on neurobiologinen pohja, ja joidenkin, etenkin ahdistukseen liittyvien temperamenttipiirteiden on huomattu olevan yhteydessä interoseptiiviseen herkkyyteen. 28 koehenkilöltä magnetoenkefalografialla (MEG) interoseptiotehtävän aikana tehtyjen aivoaktiivisuuden mittauksista ja temperamenttikyselyvastauksista koostuvaa aineistoa analysoitiin harvalla kanonisella korrelaatioanalyysillä sekä hyödyntäen sakotusta, että ilman sakotusta korkeaulotteisen datan kontekstissa. Sama analyysi tehtiin myös hyödyntäen pääkomponenttianalyysiä aineiston ulottuvuuksien vähentämiseen.

Vaikka menetelmällä olikin havaittavissa yhteys interoseptiotehtävän aikana korkeamman, vasemman päälakilohkon ja oikean otsalohkon α-aktiivisuuden, sekä Beck Anxiety Inventory -mittarin ja Behavioral Activation Scale -mittarin Fun seeking -osion matalampien tulosten välillä, ja korkeamman, vasemman otsalohkon α-aktiivisuuden ja samojen mittareiden korkeampien tulosten välillä, tilastollisesti merkitseviä kanonisia pareja ei permutaatiotestien perusteella löydetty. Harvan kanonisen korrelaatioanalyysin havaittiin helpottavan harvojen kanonisten kertoimien myötä kyselymuuttujien kertoimien tulkintaa, mutta liian harvat kanonisten parien kertoimet MEG-muuttujille voivat vaikeuttaa tulkintoja, koska MEG:n spatiaalinen tarkkuus ei riitä erottamaan pienien aivoaluiden neuronaalista aktivaatiota. Tästä syystä suurempien aivoalueiden aktivaatiot on tulkinnan kannalta suotavampia, ja pääkomponenttianalyysin avulla saatavat tasaisemmat kanoniset kertoimet voivat olla tulkinnallisesti hyödyllisempiä.

Avainsanat: interoseptio, temperamenttipiirre, magnetoenkefalografia, kanoninen korrelaatioanalyysi, lasso, sakotettu kanoninen korrelaatioanalyysi

Contents

1 Introduction **1**

2 Methods **3**
2.1 Canonical Correlation Analysis 4
2.2 Least absolute shrinkage and selection operator 6
2.3 Sparse Canonical Correlation Analysis 9

3 Data **14**
3.1 Temperament questionnaires 15
3.2 MEG measurements 18

4 Analysis **20**

5 Results **24**
5.1 SCCA without penalization 24
5.2 SCCA with penalization 26
5.3 SCCA with 12 principal components 29
5.4 SCCA with 11 principal components 33

6 Conclusion **34**

References **37**

Appendix A **39**

Appendix C **68**

Appendix C **83**

1 Introduction

The aim of this thesis is to explore the association between neural oscillation measured by magnetoencephalography during an interoception task and certain temperament traits via penalized canonical correlation analysis. This introductory section focuses on explaining the key terms related to the topic and on establishing the motivation for the thesis.

Rothbart and Bates (2007) define temperament as biologically based individual differences in self-regulation and reactivity in the context of expressing emotions, attention and activity. That is to say that temperament describes a person's responsiveness to changes in their environment and their ability to modulate those responses. Temperament is observable from early childhood and its development is influenced by experience and maturation. Haslam (2007) gives the example of noticeable differences in behavioral styles of infants, such as frequency of crying, sleep regularity and reactions to strange faces, as variations of temperament. Due to this early emergence, temperament has a neural basis and neuroimaging techniques may be used to shed light on individuals temperamental differences on a neurobiological level. Rothbart and Bates (2007) note that temperament forms the activational, attentional and affective core of personality, and when personality develops with maturation and temperamental processes become more moderated by social cognition, temperament still remains an underlying factor in tendency to react to one's environment.

Temperament consists of several temperament traits, classifications of which differ for different phases of human development. For example Haslam (2007) mentions activity level, negative emotionality and task persistence as childhood temperament factors, which are associated with adult personality traits of extraversion, neuroticism and conscientiousness, respectively. In the context of this thesis, some examples of adult temperament traits would include responsiveness to bodily sensations, and tendency for behavioral inhibition and activation. Temperament is usually studied via questionnaires, measuring different temperament traits.

Interoception refers to sensitivity for sensing visceral stimuli i.e., stimuli that originates from within one's own body (Garfinkel and Critchley, 2013). Its counterpart is exteroception, which refers to sensitivity for stimuli originating from outside the body. Like temperament, interoception has a biological basis and has been linked to temperamental traits such as anxiety, negative affect, emotional intensity, introversion and behavioral inhibition (Lyyra and Parviainen, 2018). Interoception is most commonly studied by heartbeat detection tasks, in which subjects either count their own heartbeats within a certain time span or report whether or not an external stimulus is in sync with their heartbeat.

Magnetoencephalography (MEG) is a technique for investigating neuronal activity in the brain via measuring weak magnetic fields generated by electric currents flowing in large amounts of synchronously firing neurons (Hämäläinen et al., 1993). MEG is a noninvasive method, as the magnetic fields are measured outside the body by magnetometers placed close to the head. MEG can be used to map neuronal oscillation, the synchronous firing of neurons at different frequency bands, on the brain as a whole, which is useful for observing the differences and similarities of neuronal activation between individuals during the same experimental task. In the context of this thesis, the most relevant neuronal oscillation is the so called α-oscillation. As stated by Palva and Palva (2007) α-oscillations ($\sim$10Hz) are most commonly linked to internal tasks such as mental calculations or mental imagery, and inhibition of cortical areas, which are not relevant to the current task.

The MEG scanner used in the collection of the data provided for this thesis, has 102 sensor units divided evenly in a helmet shape around the head of a subject, recording measurement from all parts of the cortex (CIBR, 2024). Each of the 102 sensor units houses three sensors, one magnetometer and two planar gradiometers perpendicular to each other. Different sensor types are sensitive to different orientations of the magnetic fields generated by neuronal activity, so the different sensors complement each other and increase the accuracy of measurements at each sensor location. The magnetometer is most sensitive to magnetic fields at the edges of a sensor unit, while the planar gradiometers are most sensitive to magnetic fields of different orientations directly beneath a sensor unit. An array of 102 sensor units results in measurements of 306 separate MEG channels divided in groups of three channels around the cortex.

Now, to recount the research problem mentioned in the beginning, the aim is to study the association between two biologically based attributes, interoception and temperament traits. To phrase this problem as a research question: Do people with different temperamental traits differ in neuronal processing of interoceptive stimuli? The data to be used for the purposes of this thesis consists of two distinct sets of variables measured on the same subjects during an MEG study.

The first set is comprised of the results of three temperament questionnaires, the Body Vigilance Scale (BVS) (Schmidt, Lerew, and Trakowski, 1997), the Behavioral Inhibition Scale (BIS) and the Behavioral Activation Scale (BAS) (Carver and White, 1994), as well as two clinical questionnaires, the Beck Anxiety Inventory (BAI) (Beck et al., 1988) and the Beck Depression Inventory (BDI) (Beck et al., 1961). BVS measures conscious attention to internal sensations, such as palpitations or sweating (Olatunji et al., 2007), BIS measures disposition to inhibit behavior to avoid punishment or negative outcomes, and BAS measures disposition to actively seek out reward and nonpunishment (Carver and White, 1994). BVS and BIS both result in one total value for the questionnaires and BAS results in three sum values for different segments of the scale. BAI and BDI are used to measure the severity of the symptoms of clinical anxiety and depression, both resulting in a sum value with higher values indicating higher symptoms (Beck et al., 1988; Beck et al., 1961).

The second set of variables are derived from MEG measurements taken during interoception and exteroception tasks. During an interoception task, subjects tried to discern whether or not an auditory stimulus was played synchronously or unsynchronously with their own heartbeat. During an exteroception task, the same auditory stimulus was played as in the interoception task, but this time the objective was to detect a slightly differing tone sometimes inserted into the auditory stimuli. The exteroception task serves as a baseline for the interoception task, differing only by the target to which attention is fixed, heartbeat during interoception and sound during exteroception.

The MEG recordings from both tasks were converted into power spectrums and the differences in power for the α-frequency band (7-12Hz) were calculated between the two tasks for each MEG channel, resulting in 306 α-power contrasts for each subject. As the MEG channels are divided evenly around the head during recording, each variable corresponds to one of 102 spatial location on the cerebral cortex, meaning that the variables describe which parts of the brain differ in α-power for the two tasks and by how much.

So, as a whole, the data set to be used consist of a set of 306 α-power variables derived from MEG recordings, and a set of seven temperament questionnaire variables measured with five different questionnaires, the BVS, BIS and BAS scales and the BDI and BAI inventories.

As a description of the structure of this thesis, section 2 establishes the methods to be used for the analysis, section 3 describes the data set mentioned here in more detail, section 4 presents the analyses carried out to address the research problem, section 5 describes the results of each analyses and section 6 offers discussion and conclusions based on the results.

2 Methods

This section describes the statistical methods used for the analysis of the interoception data, starting with the classical version of canonical correlation analysis (CCA), then covering variable selection done via least absolute shrinkage and selection operator (lasso) and finally introducing the sparse canonical correlation analysis, which generalizes the classical CCA into high-dimensional settings.

2.1 Canonical Correlation Analysis

Canonical correlation analysis is used to examine the linear relationship between two sets of variables measured on the same statistical units (Johnson and Wichern, 2013). The method was first introduced by Hotelling (1936). In the context of this thesis the two variable sets are the temperament questionnaire variables and the MEG measurement variables. Canonical correlation analysis finds a linear combination of the first set and a linear combination of the second set of variables, which have the largest correlation. Then a second pair of linear combinations is found with the largest correlation, that is uncorrelated with the first pair. This process is continued with each new pair having the largest possible correlation with the restraint of being uncorrelated with the preceding pairs. The pairs of linear combinations are referred to as canonical variates and their correlations are referred to as canonical correlations (Johnson and Wichern, 2013).

Represent random samples of size n of the two sets of variables by matrices $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_n)^\top$ and $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)^\top$, where $Y_1, \ldots, Y_n$ are p-vectors and $X_1, \ldots, X_n$ are q-vectors. $\boldsymbol{Y}$ is an $n \times p$ data matrix and $\boldsymbol{X}$ is an $n \times q$ data matrix. Then the sample covariance matrices are $\mathrm{cov}(\boldsymbol{Y}) = \hat{\boldsymbol{\Sigma}}_{yy}$, a $p \times p$ matrix, $\mathrm{cov}(\boldsymbol{X}) = \hat{\boldsymbol{\Sigma}}_{xx}$, a $q \times q$ matrix and $\mathrm{cov}(\boldsymbol{Y}, \boldsymbol{X}) = \hat{\boldsymbol{\Sigma}}_{yx} = \hat{\boldsymbol{\Sigma}}_{xy}^\top$, a $p \times q$ matrix, where $\hat{\boldsymbol{\Sigma}}_{yx} = \frac{1}{n-1} \sum_{i=1}^{n} (Y_i - \bar{Y})(X_i - \bar{X})^\top$. The joint sample covariance matrix of $\boldsymbol{Y}$ and $\boldsymbol{X}$ can be expressed as the $(p+q) \times (p+q)$ partitioned matrix

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_{yy} & \hat{\boldsymbol{\Sigma}}_{yx} \\ \hat{\boldsymbol{\Sigma}}_{xy} & \hat{\boldsymbol{\Sigma}}_{xx} \end{bmatrix},$$

with exact dimensions of the partitions of

$$\left[\begin{array}{c|c} (p \times p) & (p \times q) \\ \hline (q \times p) & (q \times q) \end{array} \right].$$

As stated by Johnson and Wichern (2013), the covariance matrix $\hat{\boldsymbol{\Sigma}}_{yx}$ captures the association between the two sets of variables, but with large amount of variables p and q it becomes difficult to interpret these pq elements collectively. The goal of canonical correlation analysis is to represent this variable set relationship with just a few highly correlated pairs of variable linear combinations.

Denote these linear combinations as $\hat{U} = \hat{a}^\top Y$ and $\hat{V} = \hat{b}^\top X$ for some coefficient vector pair $\hat{a} = (\hat{a}_1, \ldots, \hat{a}_p)^\top$ and $\hat{b} = (\hat{b}_1, \ldots, \hat{b}_q)^\top$. Then $\mathrm{var}(\hat{U}) = \hat{a}^\top \hat{\boldsymbol{\Sigma}}_{yy} \hat{a}$, $\mathrm{var}(\hat{V}) = \hat{b}^\top \hat{\boldsymbol{\Sigma}}_{xx} \hat{b}$ and $\mathrm{cov}(\hat{U}, \hat{V}) = \hat{a}^\top \hat{\boldsymbol{\Sigma}}_{yx} \hat{b}$. The correlation of the linear combinations is then

$$\mathrm{cor}(\hat{U}, \hat{V}) = \frac{\hat{a}^\top \hat{\boldsymbol{\Sigma}}_{yx} \hat{b}}{\sqrt{\hat{a}^\top \hat{\boldsymbol{\Sigma}}_{yy} \hat{a}} \sqrt{\hat{b}^\top \hat{\boldsymbol{\Sigma}}_{xx} \hat{b}}} \tag{2.1}$$

and the coefficient vectors $\hat{a}$ and $\hat{b}$ should be chosen so that this correlation is maximized.

Then define the first canonical variate pair $\hat{U}_1$, $\hat{V}_1$ as the pair that maximizes correlation (2.1) and having unit variances. The second canonical variate pair $\hat{U}_2$, $\hat{V}_2$ maximizes correlation (2.1) while having unit variances and being uncorrelated with the first canonical variate pair. Finally, the kth canonical variate pair $\hat{U}_k$, $\hat{V}_k$, maximizes correlation (2.1) under the constraints of having unit variances and being uncorrelated with all preceding pairs. The correlation between the kth canonical variate pair is referred to as the kth canonical correlation (Johnson and Wichern, 2013). The total number of canonical variate pairs will be $\min(p, q)$ (Rencher and Christensen, 2012).

If $p \leq q$ and $\hat{\boldsymbol{\Sigma}}$ has full rank, the first canonical correlation

$$\max_{\hat{a}, \hat{b}} \mathrm{cor}(\hat{U}, \hat{V}) = \hat{\rho}_1$$

is obtained by the first canonical variate pair $\hat{U}_1 = \hat{a}^\top Y = \hat{e}_1^\top \hat{\boldsymbol{\Sigma}}_{yy}^{-1/2} Y$ and $\hat{V}_1 = \hat{b}^\top X = \hat{f}_1^\top \hat{\boldsymbol{\Sigma}}_{xx}^{-1/2} X$, with $\hat{e}_1$ being calculated as the first eigenvector of

$$\hat{\boldsymbol{\Sigma}}_{yy}^{-1/2} \hat{\boldsymbol{\Sigma}}_{yx} \hat{\boldsymbol{\Sigma}}_{xx}^{-1} \hat{\boldsymbol{\Sigma}}_{xy} \hat{\boldsymbol{\Sigma}}_{yy}^{-1/2}$$

and $\hat{f}_1$ being calculated as the first eigenvector of

$$\hat{\boldsymbol{\Sigma}}_{xx}^{-1/2} \hat{\boldsymbol{\Sigma}}_{xy} \hat{\boldsymbol{\Sigma}}_{yy}^{-1} \hat{\boldsymbol{\Sigma}}_{yx} \hat{\boldsymbol{\Sigma}}_{xx}^{-1/2}.$$

The kth canonical variates $\hat{U}_k$, $\hat{V}_k$ maximize $\mathrm{cor}(\hat{U}_k, \hat{V}_k) = \hat{\rho}_k$, for $k = 2, \ldots, p$ while being uncorrelated with all preceding canonical variate pairs.

These $\hat{\rho}_1 \geq \hat{\rho}_2 \geq \ldots \geq \hat{\rho}_p$ are the canonical correlations and $\hat{\rho}_1^2 \geq \hat{\rho}_2^2 \geq \ldots \geq \hat{\rho}_p^2$ are the eigenvalues of

$$\hat{\boldsymbol{\Sigma}}_{yy}^{-1/2} \hat{\boldsymbol{\Sigma}}_{yx} \hat{\boldsymbol{\Sigma}}_{xx}^{-1} \hat{\boldsymbol{\Sigma}}_{xy} \hat{\boldsymbol{\Sigma}}_{yy}^{-1/2}$$

with $\hat{e}_1, \ldots, \hat{e}_p$ being the corresponding eigenvectors. $\hat{\rho}_1^2, \ldots, \hat{\rho}_p^2$ are also the p largest eigenvalues of

$$\hat{\mathbf{\Sigma}}_{xx}^{-1/2}\hat{\mathbf{\Sigma}}_{xy}\hat{\mathbf{\Sigma}}_{yy}^{-1}\hat{\mathbf{\Sigma}}_{yx}\hat{\mathbf{\Sigma}}_{xx}^{-1/2}$$

with $\hat{f}_1, \ldots, \hat{f}_p$ being the first p eigenvectors (Johnson and Wichern, 2013).

For all canonical variates $\hat{U}_k, \hat{V}_k$, $\mathrm{var}(\hat{U}_k) = \mathrm{var}(\hat{V}_k) = 1$ and they are uncorrelated with each other canonical variate, excluding their corresponding pair.

Johnson and Wichern (2013) and Rencher and Christensen (2012) note that if the variables are standardized, the eigenvalues (and the canonical correlations) remain unchanged, but the eigenvectors differ. This is the case if correlation matrices are used instead of the covariance matrices.

One way to frame the classical CCA is to think of it as a form of multiple regression with several predictors on both sides of a standard linear model (Tabachnick and Fidell, 2014). With just a few predictors p and q, the interpretation of the coefficients as relationships between the variable sets is feasible, but as p and q grow larger, the task quickly becomes unfeasible. To remedy this issue, regularization methods may be used to shrink the coefficient estimates to zero for better interpretability. One such method is discussed in the next section.

2.2 Least absolute shrinkage and selection operator

Assume a data set of size n with responses s_i and p predictors $t = (t_{i1}, \ldots, t_{ip})^\top$, $i = 1, \ldots, n$. Observations are independent and the predictors are standardized to have zero mean and unit variance. In this section S and $T_1, \ldots, T_p$ denote the response and predictor variables of a linear model. For the standard linear model

$$S = \beta_0 + \beta_1 T_1 + \ldots + \beta_p T_p + \epsilon,$$

with ϵ being homoscedastic random error terms with mean zero and independent of the predictors, describing the relationship between the response S and predictor variables $T_1, \ldots, T_p$, the least squares fit estimates for the coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are usually calculated using the values which minimize the Residual Sum of Squares (RSS),

$$RSS = \sum_{i=1}^{n}(s_i - \beta_0 - \sum_{j=1}^{p}\beta_j t_{ij})^2$$

(James et al., 2021). If the relationship between the response S and the predictor variables T is linear and the sample size n is much larger than the amount of predictors p the least squares estimates will have low variance and low bias. But if p is close to n, the least squares fit can have a lot of variability and be prone to overfitting and if p is larger than n there is no unique least squares solution. James et al. (2021) note that in such cases shrinking the coefficients towards zero can reduce variance with a small bias increase trade-off, increasing prediction accuracy. Shrinking some coefficients to zero also has the benefit of greatly increasing model interpretability, as less impactful predictors are essentially removed from the model.

One method to perform this coefficient shrinkage is the least absolute shrinkage and selection operator (lasso). First proposed by Tibshirani (1996), the lasso coefficients minimize the quantity

$$\sum_{i=1}^{n}\left(s_i - \beta_0 - \sum_{j=1}^{p}\beta_j t_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|, \tag{2.2}$$

where $\lambda \geq 0$ is a tuning parameter and $\sum|\beta_j| = ||\beta_j||_1$ is the so called l_1 penalty (James et al., 2021). This quantity differs from the RSS by the added shrinkage penalty, $\lambda\sum_{j=1}^{p}|\beta_j|$. This term is small when the coefficients are close to zero, so it effectively shrinks the coefficient estimates. When the tuning parameter $\lambda = 0$, lasso produces the least squares estimates and when λ grows sufficiently large, all of the coefficients are forced to exactly zero due to the l_1 penalty.

The lasso estimates $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^\top$ can also be defined as

$$\hat{\beta} = \arg\min\left\{\sum_{i=1}^{n}\left(s_i - \beta_0 - \sum_{j=1}^{p}\beta_j t_{ij}\right)^2\right\} \text{ subject to } \sum_{j=1}^{p}|\beta_j| \leq \kappa \tag{2.3}$$

(Tibshirani, 1996). For every λ there exists κ so that (2.2) and (2.3) give the same coefficient estimates. The intercept β_0 is not affected by the penalization and can be omitted when the data is standardized to have zero mean.

When fitting a linear model using the lasso, all available predictor variables p are included in the model. This is computationally a clear improvement over subset selection where a separate model for each combination of the desired amount of predictors has to be fitted (James et al., 2021). Subset selection also has the drawback of being prone to large changes in the chosen model due to only small changes in the data (Tibshirani, 1996).

The proper value for the tuning parameter λ is usually chosen via k-fold cross-validation. The data set is randomly divided into k evenly sized groups and one of the groups is used as a validation set, while the others form a unified training set. For a single candidate λ-value, a linear model is fit using that λ on the training set, and the mean squared error (MSE) is calculated by using the fitted model to predict the responses of the validation set and calculating the mean of the squared errors. Then the MSE is calculated for each of the other $k-1$ groups in a similar way and the cross-validation error estimate is received as the mean of all k MSEs. If the amount of observations in a given group is m, the mean squared error of a single group j is

$$MSE_j = \frac{1}{m}\sum_{i=1}^{m}(s_i - \hat{s}_i)^2,$$

and the k-fold cross-validation error is

$$CV_{(k)} = \frac{1}{k}\sum_{j=1}^{k} MSE_j.$$

Then a large grid of possible λ-values can be selected as candidates and the cross-validation error is calculated for models with each of these λ-values. The λ-value with the smallest cross-validation error is selected for the final model and this model is fitted with all available observations (James et al., 2021).

To illustrate the use of the lasso, a data set of $n = 1000$ vectors $t_i = (t_{i1,}, \ldots, t_{ip})^\top$ with $p = 25$ were generated from standard normal distribution. Then a response variable s was created as a function of the first four variables so that $s_i = \beta_0 + \beta_1 t_{i1} + \beta_2 t_{i2} + \beta_3 t_{i3} + \beta_4 t_{i4} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 5)$, $\beta_0 = 3$ and $(\beta_1, \beta_2, \beta_3, \beta_4)^\top = (2, 4, 6, 8)^\top$, so the response variable was unassociated with the variables $t_5, \ldots, t_{25}$. Now fitting a multiple linear regression model with s as a response and all 25 predictors yields least squares estimates for the coefficients, which are quite close to the real coefficient values of the first four predictors, and small but non-zero estimates for the coefficients $\beta_5, \ldots, \beta_{25}$.

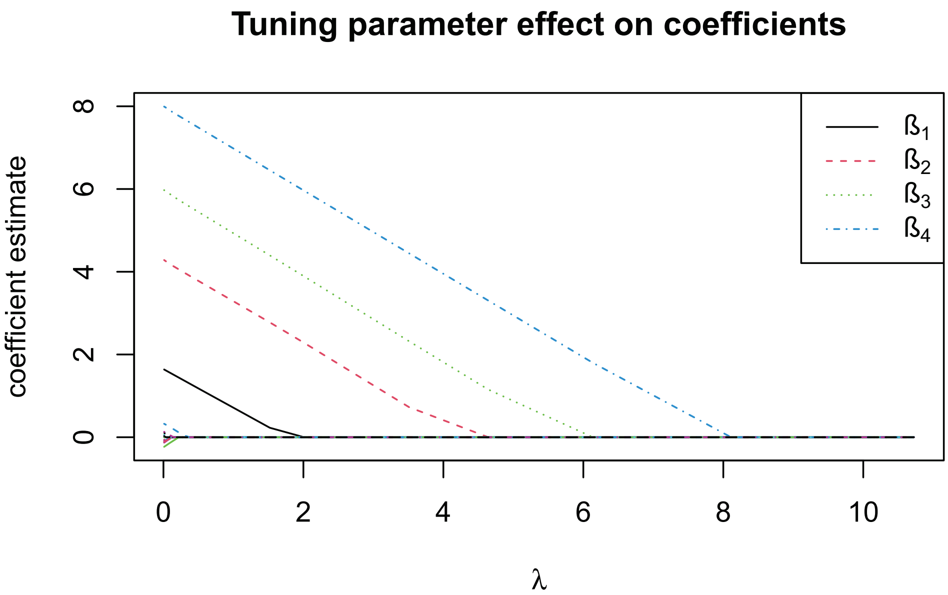


Figure 2.1: Effect of the tuning parameter λ on the coefficient estimates.

Fitting the same model using the lasso yields the same coefficient estimates when $\lambda = 0$, but even a slight increase in λ quickly drops the coefficient estimates $\hat{\beta}_5, \ldots, \hat{\beta}_{25}$ to exactly zero. This effect of the tuning parameter can be seen in Figure 2.1. When λ is increased enough, even the estimates of the real predictors $\beta_1, \ldots, \beta_4$ shrink to zero, with the coefficient of least impact on y reaching zero first. 10-fold cross-validation results in an optimal λ-value of 0.5, which shrinks all $\beta_5, \ldots, \beta_{25}$ to zero, with only a small effect on the real coefficients.

2.3 Sparse Canonical Correlation Analysis

The classical canonical correlation analysis described in the beginning of this section does not lend itself effortlessly to the analysis of high-dimensional data sets, where the amount of variables, p and q, in the two variable sets are much larger than the sample size n. In this case the sample covariance matrices $\hat{\boldsymbol{\Sigma}}_{yy}$ and $\hat{\boldsymbol{\Sigma}}_{xx}$ are not invertible, and classical CCA no longer produces unique canonical variates. In addition, interpretation of possibly hundreds of coefficients in each canonical pair could prove inordinately difficult and impractical, while sparse estimates with a small number of coefficients differing from zero would be ideal for interpretation. Many sparse CCA methods have been developed for producing these sparse canonical pair estimates in the high dimensional context, such as the Penalized Matrix Decomposition (PMD) (Witten, Tibshirani, and Hastie, 2009) and the Convex program with group-Lasso Refinement (CoLaR) (Gao, Ma, and Zhou, 2017).

The Penalized Matrix Decomposition (PMD) (Witten, Tibshirani, and Hastie, 2009) is a regularized version of the singular value decomposition, and can be employed to estimate canonical variates in high-dimensional settings where p, q or both exceed the sample size n. PMD substitutes the sample covariance matrices $\hat{\boldsymbol{\Sigma}}_{yy}$ and $\hat{\boldsymbol{\Sigma}}_{xx}$ with identity matrices to negate their non-invertibility and implements penalization to produce sparse canonical variates. This identity matrix substitution imposes an assumption on covariance matrices $\boldsymbol{\Sigma}_{yy}$ and $\boldsymbol{\Sigma}_{xx}$, and thus PMD might not produce consistent estimates in situations where these differ greatly from identity matrices (Mai and Zhang, 2019).

Convex program with group-Lasso Refinement (CoLaR) (Gao, Ma, and Zhou, 2017) is a computationally feasible procedure for estimating sparse canonical coefficient vectors in a high-dimensional setting, while achieving minimax estimation risk, and without making any assumptions on the covariance matrices. However, CoLaR imposes conditions on the sample size of the data set it is to be applied to, and as pointed out by Mai and Zhang (2019), does not produce nested solutions resulting in ambiguity of the results.

For the purposes of this thesis, only the SCCA method proposed by Mai and Zhang (2019) will be discussed here in more detail. This method differs from other prominent SCCA methods by reformulating high-dimensional canonical correlation analysis as an iterative penalized least squares problem, directly generalizing CCA to a high-dimensional setting, producing nested solutions. Notably, SCCA does not impose any assumptions on the covariance matrices $\hat{\boldsymbol{\Sigma}}_{yy}$ and $\hat{\boldsymbol{\Sigma}}_{xx}$, and can be applied to small data sets, which is beneficial given the small sample size of the data to be used in this thesis.

Assume independent and identically distributed data consisting of n measurements of two sets of variables: $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_n)^\top$, a set of p variables and $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)^\top$, a set of q variables, with both $\boldsymbol{Y}$ and $\boldsymbol{X}$ being centered, and the sample covariance matrices being defined as $\hat{\boldsymbol{\Sigma}}_{yy} = \frac{1}{n}\boldsymbol{Y}^\top\boldsymbol{Y}$, $\hat{\boldsymbol{\Sigma}}_{xx} = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$ and $\hat{\boldsymbol{\Sigma}}_{yx} = \frac{1}{n}\boldsymbol{Y}^\top\boldsymbol{X}$. Then, when $n > \max(p, q)$ the coefficients of the kth canonical variate pair, $\hat{a}_k^{CCA}$ and $\hat{b}_k^{CCA}$ for the classical canonical correlation can be defined as:

$$(\hat{a}_k^{CCA}, \hat{b}_k^{CCA}) = \arg\max_{a_k, b_k} a_k^\top \hat{\boldsymbol{\Sigma}}_{yx} b_k,$$

so that $a_k^\top \hat{\boldsymbol{\Sigma}}_{yy} a_k = 1$, $b_k^\top \hat{\boldsymbol{\Sigma}}_{xx} b_k = 1$, $a_k^\top \hat{\boldsymbol{\Sigma}}_{yy} \hat{a}_l = 0$ and $b_k^\top \hat{\boldsymbol{\Sigma}}_{xx} \hat{b}_l = 0$ for any

$l < k$, so $\hat{a}_l$, $\hat{b}_l$ are any of the previous $k-1$ canonical variate pairs.

Then, according to Mai and Zhang (2019), this classical CCA solution for the kth canonical pair can be reformulated as a constrained quadratic optimization problem as:

$$
\begin{aligned}
(\hat{a}_k', \hat{b}_k') = \arg\min_{a_k, b_k} \Bigg\{ & \frac{1}{2n} \sum_{i=1}^{n} (Y_i^\top a_k - X_i^\top b_k)^2 \\
& + a_k^\top \Big(\sum_{l<k} (\hat{a}_l')^\top \hat{\boldsymbol{\Sigma}}_{yx} \hat{b}_l' \hat{\boldsymbol{\Sigma}}_{yy} \hat{a}_l' \cdot (\hat{b}_l')^\top \hat{\boldsymbol{\Sigma}}_{xx} \Big) b_k \Bigg\},
\end{aligned}
$$

so that $a_k^\top \hat{\boldsymbol{\Sigma}}_{yy} a_k = 1$ and $b_k^\top \hat{\boldsymbol{\Sigma}}_{xx} b_k = 1$. Then $\hat{a}_k' = \hat{a}_k^{CCA}$ and $\hat{b}_k' = \hat{b}_k^{CCA}$.

Mai and Zhang (2019) note that this solution consist of two terms. $\frac{1}{2n} \sum_{i=1}^{n} (Y_i^\top a_k - X_i^\top b_k)^2$ is a measure of linear dependancy between the kth canonical pair, with a small value resulting in strong correlation between the canonical variates. The second term is used to account for the variability of the previous $k-1$ canonical pairs, and the constraints $a_k^\top \hat{\boldsymbol{\Sigma}}_{yy} \hat{a}_l = 0$ and $b_k^\top \hat{\boldsymbol{\Sigma}}_{xx} \hat{b}_l = 0$ of the classical CCA solution are removed by this term.

When $p, q > n$, the coefficient vectors a_k and b_k are assumed sparse, with most of the values being zero. The sparse canonical correlation analysis solution proposed by Mai and Zhang (2019) is then

$$
\begin{aligned}
(\hat{a}_k, \hat{b}_k) = \arg\min_{a_k, b_k} \Bigg\{ & \frac{1}{2n} \sum_{i=1}^{n} (Y_i^\top a_k - X_i^\top b_k)^2 \\
& + a_k^\top \Big(\sum_{l<k} \hat{a}_k^\top \hat{\boldsymbol{\Sigma}}_{yx} \hat{b}_k \hat{\boldsymbol{\Sigma}}_{yy} \hat{a}_l' \cdot \hat{b}_l^\top \hat{\boldsymbol{\Sigma}}_{xx} \Big) b_k \\
& + \lambda_{a_k} ||a_k||_1 + \lambda_{b_k} ||b_k||_1 \Bigg\},
\end{aligned} \tag{2.4}
$$

so that $a_k^\top \hat{\boldsymbol{\Sigma}}_{yy} a_k = 1$ and $b_k^\top \hat{\boldsymbol{\Sigma}}_{xx} b_k = 1$ and λ_{a_k}, $\lambda_{b_k} \geq 0$ are tuning parameters. The solution incorporates the penalty terms for the absolute values of the coefficient similar to the lasso, but the coefficients of both canonical variates are penalized separately with their own tuning parameter.

This formulation uses the l_1 penalty to impose the sparsity structure on the canonical pairs, but other penalty function can be applied to either or both coefficient vectors. When the tuning parameters are set to zero, the classical canonical correlation solution is produced.

Mai and Zhang (2019) also propose and implement an algorithm for finding the solution to (2.4). For any $k \in \ ^{+}$, they define $\hat{\boldsymbol{A}}_k = (\hat{a}_1, \ldots, \hat{a}_k)$, $\hat{\boldsymbol{B}}_k = (\hat{b}_1, \ldots, \hat{b}_k)$ as matrices of the k canonical variates,

$\boldsymbol{R}_k = \text{diag}(a_1^\top \hat{\boldsymbol{\Sigma}}_{yx} b_1, \ldots, a_k^\top \hat{\boldsymbol{\Sigma}}_{yx} b_k)$ as a diagonal matrix of the canonical correlations and $\boldsymbol{\Omega}_1 = \boldsymbol{I}_n$, $\boldsymbol{\Omega}_k = \boldsymbol{I}_n - \boldsymbol{Y}\boldsymbol{A}_{k-1}\boldsymbol{R}_{k-1}\boldsymbol{B}_{k-1}\boldsymbol{X}^\top/n$, where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. The algorithm is motivated by the result that when the previous $k-1$ canonical pairs are known and if a_k is fixed, then the solution for $\hat{b}_k$ in (2.4) is

$$\hat{b}_k = \{(\check{b}_k)^\top \hat{\boldsymbol{\Sigma}}_{xx} \check{b}_k\}^{-1/2} \cdot \check{b}_k,$$

where

$$\check{b}_k = \arg\min_{b_k} \Big\{ \frac{1}{2n} ||\boldsymbol{\Omega}_k^\top \boldsymbol{Y} a_k - \boldsymbol{X} b_k||_2^2 + \lambda_{b_k} ||b_k||_1 \Big\},$$

and if b_k is fixed, the solution for $\hat{a}_k$ in (2.4) is

$$\hat{a}_k = \{(\check{a}_k)^\top \hat{\boldsymbol{\Sigma}}_{yy} \check{a}_k\}^{-1/2} \cdot \check{a}_k,$$

where

$$\check{a}_k = \arg\min_{a_k} \Big\{ \frac{1}{2n} ||\boldsymbol{\Omega}_k^\top \boldsymbol{X} b_k - \boldsymbol{Y} a_k||_2^2 + \lambda_{a_k} ||a_k||_1 \Big\}.$$

So finding $\hat{a}_k$ and $\hat{b}_k$ boil down to solving $\check{a}_k$ and $\check{b}_k$, when one canonical variate is fixed, through l_1 penalized least squares problems. Then the algorithm proposed by Mai and Zhang (2019) is:

1. Compute $\boldsymbol{\Omega}_k$ using the previous $k-1$ canonical pairs $\boldsymbol{A}_{k-1}$, $\boldsymbol{B}_{k-1}$.
2. Choose initialization values $\{\hat{a}_k^{(0)}, \hat{b}_k^{(0)}\}$.
3. For $m = 1, 2, \ldots$ repeat steps 4. and 5. until convergence.
4. Set $\tilde{\boldsymbol{Y}}_k^{(m)} = \boldsymbol{\Omega}_k^\top \boldsymbol{Y} \hat{a}_k^{(m)}$, compute
$$\check{b}_k^{(m)} = \arg\min_{b_k} \Big\{ \frac{1}{2n} ||\tilde{\boldsymbol{Y}}_k^{(m)} - \boldsymbol{X} b_k||_2^2 + \lambda_{b_k} ||b_k||_1 \Big\},$$
and then set
$$\hat{b}_k^{(m)} = \{(\check{b}_k^{(m)})^\top \hat{\boldsymbol{\Sigma}}_{xx} \check{b}_k^{(m)}\}^{-1/2} \cdot \check{b}_k^{(m)}.$$
5. Set $\tilde{\boldsymbol{X}}_k^{(m)} = \boldsymbol{\Omega}_k^\top \boldsymbol{X} \hat{b}_k^{(m)}$, compute
$$\check{a}_k^{(m)} = \arg\min_{a_k} \Big\{ \frac{1}{2n} ||\tilde{\boldsymbol{X}}_k^{(m)} - \boldsymbol{Y} a_k||_2^2 + \lambda_{a_k} ||a_k||_1 \Big\},$$
and then set
$$\hat{a}_k^{(m)} = \{(\check{a}_k^{(m)})^\top \hat{\boldsymbol{\Sigma}}_{yy} \check{a}_k^{(m)}\}^{-1/2} \cdot \check{a}_k^{(m)}.$$
6. Output $(\hat{a}_k, \hat{b}_k)$.

R implementation of the iterative penalized least squares algorithm for solving $(\hat{a}_k, \hat{b}_k)$ can be found in the supplementary material of Mai and Zhang (2019).

The values of $\{\hat{a}_k^{(0)}, \hat{b}_k^{(0)}\}$ for initialization in the second step of the algorithm are recommended by Mai and Zhang (2019) to be chosen by one of two methods. When the canonical pairs are not very sparse the initial values can be calculated via singular value decomposition as the first pair of singular vectors of $\hat{\mathbf{\Sigma}}_{yx} - \sum_{l=1}^{k-1} \hat{\rho}_l \hat{a}_l \hat{b}_l^\top$, where $\hat{\rho}_l = \hat{a}_l^\top \hat{\mathbf{\Sigma}}_{yx} \hat{b}_l$, $l < k$. If the pairs are excessively sparse, an alternate initialization method is recommended. While knowing the preceding $k-1$ canonical pairs $(\hat{a}_j, \hat{b}_j)_{j=1}^{k-1}$, define

$$\hat{\mathbf{\Sigma}}_{yx}^{(k-1)} = \hat{\mathbf{\Sigma}}_{yx} - \hat{\mathbf{\Sigma}}_{yy}\left(\sum_{j=1}^{k-1} \hat{\rho}_j \hat{a}_j \hat{b}_j^\top\right)\hat{\mathbf{\Sigma}}_{xx} = (\hat{\sigma}_{yx,lm}^{(k-1)}).$$

Then define γ as the $\sqrt{n}$'th largest entry in $|\hat{\sigma}_{yx,lm}^{(k-1)}|$, $l = 1, \ldots, q$; $m = 1, \ldots, p$ and identify the sets:

$$D_y^{(k)} = \{l : \exists\ m \text{ so that } |\hat{\sigma}_{yx,lm}^{(k-1)}| \geq \gamma \text{ or } \exists\ j \text{ so that } \hat{b}_{jl} \neq 0\},$$

$$D_x^{(k)} = \{m : \exists\ l \text{ so that } |\hat{\sigma}_{yx,lm}^{(k-1)}| > \gamma \text{ or } \exists\ j \text{ so that } \hat{a}_{jl} \neq 0\}.$$

Then the initialization values can be calculated as the first pair of singular vectors of $\{\hat{\mathbf{\Sigma}}_{yx}^{(k-1)}\}_{D_y^{(k)}, D_x^{(k)}}$, meaning a submatrix of $\hat{\mathbf{\Sigma}}_{yx}^{(k-1)}$ for which maximum value of each row and column are greater than γ.

The tuning parameters λ_a and λ_b can be chosen via k-fold cross-validation, R implementation of which for the first canonical pair can be found in the supplementary material of Mai and Zhang (2019). A candidate set of possible λ-values is chosen for both λ_a and λ_b and the data set is randomly divided into k subgroups as evenly sized as possible. For each possible pairing of λ-values from different candidate sets, a performance measure value ρ_i, $i = 1, \ldots, k$, is calculated for each of the k subgroups, by using one of the groups as a validation set and combining the others into a training set. SCCA is performed on the training set using the λ-value pairing for which ρ_i is to be obtained, to receive a coefficient pair $\hat{a}_i$ and $\hat{b}_i$, and the ρ_i-value is then calculated as $\rho_i = |\text{cor}(\boldsymbol{X}_i \hat{b}_i, \boldsymbol{Y}_i \hat{a}_i)|$, where $\boldsymbol{X}_i$ and $\boldsymbol{Y}_i$ are the corresponding validation sets. The final performance measure ρ for a pair of λ-values is the sum of the ρ_i-values from each fold, $\rho = \sum_{i=1}^k \rho_i$. After the ρ-value has been calculated for each possible λ pairing, the pair resulting in the highest ρ is used in the final SCCA estimation of the canonical pairs.

Mai and Zhang (2019) further note that if prior information on the sparsity structure exists and other penalties than the l_1 penalty seem appropriate, other penalty functions such as group, fused or adaptive lasso may be used instead. Group lasso is best used in situations where some of the variables in the data are known to function together as a group. In this situation the group lasso enables simultaneous handling of the grouped variables, by shrinking the coefficients of the group all together to zero or keeping the whole group in the model, so the variable selection is done on group level instead of individual variable level (Yuan and Lin, 2005). The fused lasso incorporates a penalty for the differences of successive coefficients so if variables with similar effect are placed next to each other in the data, fused lasso may be used to include or exclude these segments of variables together (Tibshirani et al., 2004). Adaptive lasso, where different coefficients are assigned data-dependent adaptive weights, may be used if the l_1 penalty proves unstable in cross-validation (Zou, 2006).

3 Data

This section describes the data in more detail, summarizes the questionnaires used in collecting the temperament variables and outlines the data preprocessing steps taken before carrying out the analysis.

The data set used in this thesis was provided by Suvi Karjalainen and was collected in 2020 and 2021 by recruiting participants from the Central Finland area via university mailing lists, social media and posters. Questionnaire data were collected from 31 participants, but the MEG measurements could not be collected from three participants due to the Covid-19 pandemic, resulting in the final participant count of 28. Of these participants, 20 were female and 8 male, with a mean age of 23.6 years and ages ranging from 19 to 30 years.

The original data set consisted of both MEG measurements and questionnaire answers from 28 subjects. The MEG data comprised 306 power spectrums between 1 and 40 Hz range for each of the two task conditions, interoception and exteroception, for each subject. In other words, the MEG measurements resulted in 56 distinct 306×40 data matrices, one for each task condition, and two for each subject. For these matrices, each observation describes the neuronal activity at 40 different frequency bands for a single MEG channel.

The questionnaire data consisted of sum values of the answers to five different questionnaires, the Body Vigilance Scale (BVS), the Behavioral Inhibition Scale (BIS), the Behavioral Activation Scale (BAS), the Beck Anxiety Inventory (BAI) and the Beck Depression Inventory (BDI). Each subject has one sum value for each questionnaire, with the exception of BAS, for which each subject has three different sum values, representing the different subsections of the questionnaire. All in all, each subject has seven different sum values for the five applied questionnaires. The following section describes these in more detail.

3.1 Temperament questionnaires

The Body Vigilance Scale measures sensitivity to internal bodily sensations via a four item self-report questionnaire (Schmidt, Lerew, and Trakowski, 1997). The questions are answered based on one's personal observations of the past week. The first three items assess tendency to pay attention to, notice changes in and spending time on observing one's bodily sensations. The fourth item asks to estimate the attention paid to each of 15 different physical symptoms associated with panic attacks according to DSM-IV, such as heart palpitations, dizziness, sweating and shortness of breath. The first three items each give a score from 0 to 10 and the fourth is evaluated as a mean value of 15 answers given on a scale from 0 to 10. The final result of the questionnaire is the sum of all four items, so the possible values range from 0 to 40.

The BIS/BAS Scales form one self-report questionnaire to assess the sensitivity of two motivational systems, the behavioral inhibition system (BIS) and the behavioral activation system (BAS) (Carver and White, 1994). Activation of the BIS promotes inhibition of behavior that could lead to negative outcomes such as punishment or non-reward, while activation of the BAS promotes behavior that could lead to reward, achieving of goals or escape from punishment. As designed by Carver and White (1994), the BIS/BAS Scale questionnaire consist of 4-point Likert-scale items and has four subscales, one BIS-related and three BAS-related scales, as three types of behavioral activation can be separated, drive, fun seeking and reward responsiveness. A higher score in a scale relates to higher sensitivity to the activation of the respective motivational system. As an example, a high BIS score would relate to a higher tendency to inhibit behavior which could lead to negative outcomes, a high BAS Drive score to tendency to persistently pursue desired goals, a high BAS Fun Seeking score to tendency to seek out potentially rewarding events, and a high BAS Reward Responsiveness score to tendency for high positive response to anticipation and occurrence of new rewards. The questionnaire results in four sum values of the 4-point Likert-scale items, one for each subscale.

The Beck Anxiety Inventory (BAI) is a clinical self-report inventory used to measure the severity of symptoms of anxiety. (Beck et al., 1988). BAI consist of 21 items each corresponding to a single common symptom, and each item is answered on a Likert-scale ranging from 0 to 3, with higher values indicating higher symptom severity. The respondent answers how much each symptom has bothered them during the past week, with 0 indicating "not at all" and 3 indicating "severely". All answers are summed up to a final value, with possible values ranging from 0 to 63.

The Beck Depression Inventory (BDI) is used equivalently to the BAI for assessing the severity of clinical depression symptoms via a 21-item self-report inventory (Beck et al., 1961), with each item corresponding to a common depression symptom or attitude. Like the BAI, each item is answered on a 0 to 3 Likert-scale, probing the severity of the symptoms experienced during the past week. The answers are summed to a final score, which can be used to estimate the severity of clinical depression, with scores lower than 10 indicating none or minimal depression, 10–18 indicating mild, 19–29 indicating moderate, and 30–63 indicating severe depression (Beck, Steer, and Carbin, 1988).

While the BAI and the BDI are not specifically temperament questionnaires, the questionnaire answers were considered to be useful to include in the analysis, since especially the behavioral inhibition system relates to the experience of anxiety (Carver and White, 1994) and the attendance to bodily sensations related to the BVS are central in experiencing anxiety (Olatunji et al., 2007). Also the answers can be used to determine that no participants had moderate or severe symptoms of clinical depression or anxiety.

The means and standard deviations of the sums of the questionnaire results included in the data can be seen in Table 1. All BDI and BAI results were in the minimal-to-mild symptoms range.

Table 1: Means and standard deviations of the sums of the questionnaire answers

Questionnaire	Mean	SD
BVS	17.79	7.67
BIS	23.96	5.70
BAS Drive	14.50	2.87
BAS Fun Seeking	14.96	3.05
BAS Reward Responsiveness	19.61	2.86
BAI	4.71	3.49
BDI	5.46	3.53

With the different questionnaires having such widely differing answer scales, a rank transformation was carried out for the questionnaire variables, with the lowest values being replaced with the lowest and highest values with the highest rank. Tied values were replaced by the mean of the corresponding tied ranks. Tied questionnaire variable values were much more prominent in the BDI, BAI and the three BAS variables, for which the answers of 28 subjects resulted in only 10 to 12 unique sum values. For the BDI and BAI variables, this can be explained by all of the subjects scoring in the minimal to mild symptom range of 0 to 18, resulting in many equal scores. For the BAS variables, each of the scores are the results of sums of a fairly small number of 4-point Likert-scale items, so for 28 subjects many similar scores are likely.

Sample correlations of the questionnaire variables can be seen in Figure 3.1. Notably, the two clinical inventories, BDI and BAI are positively correlated with each other and the BIS and BVS questionnaires, while the BAS scales seem positively correlated with each other, but weakly or negatively correlated with all other questionnaire variables. The relatively high positive correlations between BAI, BIS and BVS seem natural since both body vigilance and the behavioral inhibition system are closely associated with anxiety sensitivity (Carver and White, 1994; Schmidt, Lerew, and Trakowski, 1997). Similarly, the positive correlations between the BAS scales seems reasonable, since they all measure different types of behavioral activation.

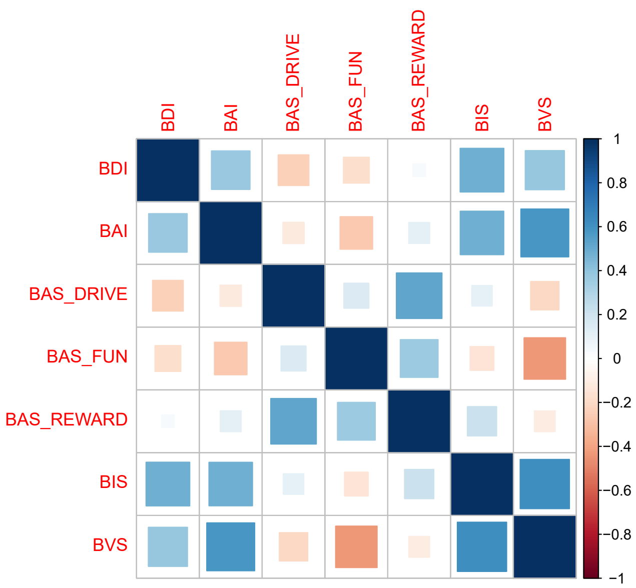


Figure 3.1: Correlation matrix of the questionnaire variables after rank transformation. Blue indicates positive, and red negative correlation.

3.2 MEG measurements

Since the aim of the analysis was to focus on the α-frequency band of the MEG recordings, the mean value of frequencies 7–12 Hz was calculated for each of the 1–40 Hz power spectrums, resulting in 56 data matrices of size 306×1 containing only the α-activity measurements during the MEG recording of the interoception and exteroception tasks. Then the difference in α-power between each task for each subject was calculated by subtracting the exteroception measurements from the interoception measurements, and the results were combined into one 28×306 data matrix, where each row corresponds to the α-power difference or contrast in each of the 306 channels between the two tasks for a single subject. As the channels are distributed evenly in groups of three around the head during an MEG recording, one of the 102 corresponding spatial location on the cerebral cortex can be discerned from the number of a channel.

As an example, the α-power contrast values of the 306 MEG channel variables for a single subject are plotted in Figures 3.2 and 3.3. As the channel variables are grouped in the data in groups of three as they are in the MEG sensor units, i.e. magnetometer, and two perpendicular planar gradiometers, Figure 3.2 shows every third channel variable of the data set, the magnetometer channels, and Figure 3.3 shows the alternating planar gradiometers. For example, channel 1 from Figure 3.2 and channels 1 and 2 from Figure 3.3 all together form the three sensors of the first sensor unit. From the two figures can be discerned that the magnetometer variables gain considerably smaller values than the planar gradiometer variables. This is due to magnetometers measuring magnetic fields in different units than planar gradiometers. Similar shapes of the plots in Figures 3.2 and 3.3 show how channels of the same sensory unit gain similar values, despite the difference in scales. If plotted in the same figure, magnetometer alpha-powers would not be distinguishable from zero due to the scale difference.

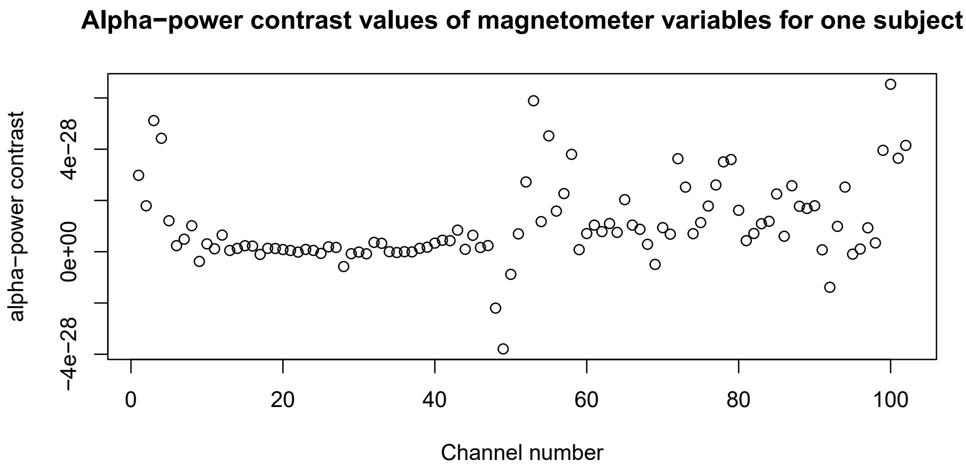


Figure 3.2: Scatter plot of 102 magnetometer channel variables for one subject.

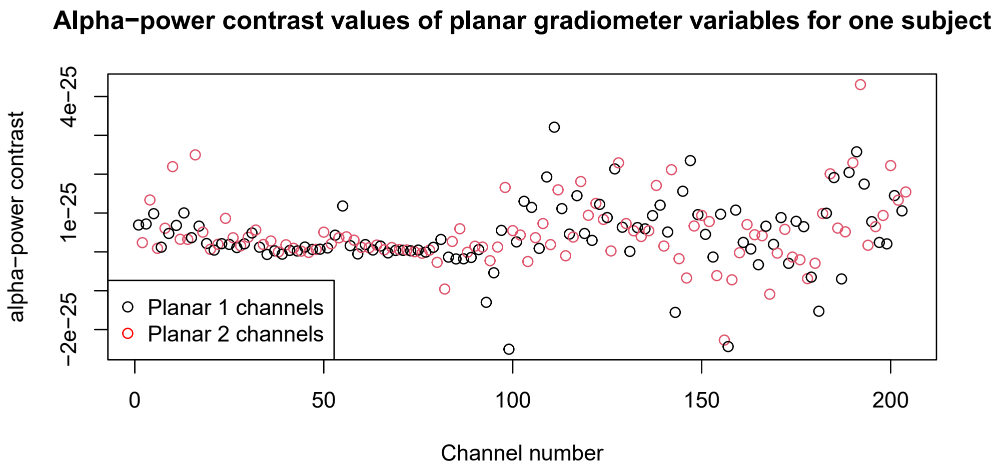


Figure 3.3: Scatter plot of 204 planar gradiometer channel variables for one subject.

4 Analysis

This section describes the analyses carried out to investigate the associations between the questionnaire and MEG measurement variables, with the aim of discovering combinations of psychological traits associated with distinct patterns of α-activity during the interoception task, and comparing the practicability of the analysis methods for this research problem. Four separate analyses were performed using SCCA described in Section 2.3. First, SCCA was used without any regularization, effectively performing equivalent analysis to classical CCA on a high dimensional data set, for which the CCA described in Section 2.1 would be unfeasible due to the larger amount of MEG variables compared to the sample size. Second, SCCA with regularization was carried out to induce sparse canonical coefficients to reveal the most relevant variables for the canonical pairs and to produce more interpretable results. Third, principal component analysis was used to reduce the dimensions of the MEG variable set, and SCCA was performed substituting the MEG variables with a set of 12 principal component score variables, accounting for 95% of the variance in the MEG variable set. For the fourth analysis, the third principal component was omitted from the data due to it possibly accounting for a heart beat artifact, i.e. noise irrelevant to the α-activity, and similar analysis to the previous one was carried out. The results of these analyses are reported in Section 5.

Each of the four analyses involved estimation of seven canonical pairs, number of which is determined by the number of variables in the smaller variable set, in this case the questionnaire variables. Functions provided in the supplementary material of Mai and Zhang (2019) were used to determine the initial values for the SCCA algorithm, performing cross-validations for choosing the tuning parameters of the first canonical pair and calculating the estimates of the canonical pairs. Cross-validation function intended for the first canonical pair from the material was further modified for use with canonical pairs beyond the first one. This involved altering the function to estimate ρ-values based on a given pair number, instead of the first pair. The altered version of the supplementary code from Mai and Zhang (2019) used for the analysis can be found in Appendix D.

Since the sample size of 28 subjects is fairly low, splitting the data into training and validation sets for the purpose of choosing the optimal tuning parameter values seems impractical. Instead, k-fold cross-validation, as described in Section 2.3, was used to choose the tuning parameters for each canonical pair. 5-fold cross-validation was chosen for this purpose, leaving 5 to 6 observations for each validation set, achieving a decent balance between the bias and variance for the cross-validation, and being computationally manageable when performed seven times for three separate analyses. The first analysis did not require cross-validation, since the tuning parameters were all set to zero.

The initialization values for the SCCA algorithm in each analysis were calculated using functions of the supplementary material of Mai and Zhang (2019), choosing the first initialization method described in Section 2.3. This choice was made motivated by the result that the second initialization method, assuming that the canonical pairs are excessively sparse, assigned all canonical weight on a single variable in the smaller, questionnaire set, and performed worse in the terms of the cross-validation performance measure ρ.

Permutation tests similar to the ones performed by Witten, Tibshirani, and Hastie (2009) and Mai and Zhang (2019) were used to test the validity of the first canonical pair. For each analysis, the questionnaire variable set was permuted N times, and SCCA was applied again to calculate the first canonical pairs using the permuted variable sets and the MEG variables corresponding to the analysis. Then, canonical correlations $\hat{c}_i$, $i = 1, \ldots, N$ of each of the N new pairs were calculated and a p-value was computed as $\frac{1}{N}\sum_{i=1}^{N} = \mathbf{I}(\hat{c}_i > \hat{c})$, where $\hat{c}$ is the canonical correlation of the first pair calculated using the original data, and $\mathbf{I}$ denotes an indicator function for which

$$\mathbf{I} = \begin{cases} 1 & \text{when } \hat{c}_i > \hat{c}, \\ 0 & \text{when } \hat{c}_i \leq \hat{c}. \end{cases}$$

$N = 500$ was chosen as a compromise between adequate accuracy and reasonable computation time.

In the first analysis, SCCA was used without any regularization, meaning tuning parameters λ_a and λ_b were both set to be zero, allowing the equivalent of the classical CCA to be performed even with the number of 306 MEG variables greatly exceeding the number of 28 observations. Since the scales of the MEG measurements differ based on the sensor types, data standardization was used to diminish the effect of scale differences, of giving too much emphasis on the measurements taken by the planar gradiometers. So in practice, the sample correlation matrices instead of the sample covariance matrices were used in calculating the canonical pairs with SCCA, which was also the case in the other three analyses. Since no regularization was used, the resulting canonical pairs give non-zero weights to all variables, which makes the interpretation of the canonical weights exceedingly difficult. Additionally, with such a high number of MEG variables the SCCA is able to find seven canonical pairs with extremely high canonical correlations, making it difficult to discern which pairs would be most relevant for further interpretation.

The second analysis incorporated regularization by applying SCCA after choosing separate tuning parameters for the two variable sets via 5-fold cross-validation. The l_1 penalty was used for both variable sets in all three analyses which used regularization, as this penalty produces sparse canonical weights more desirable for interpretation. Two groups of 25 candidate penalty values were set, each ranging from 0.001 to 0.049 with an increment of $0,002$ between each value, which formed a 25×25 grid of possible λ_a and λ_b combinations. The combination resulting in the largest ρ-value in the 5-fold cross-validation was chosen for the SCCA. This search for optimal tuning parameters was repeated for each canonical pair separately. However, the tuning parameter of the MEG variable set for the seventh canonical pair was lowered to avoid excessive penalization resulting in zero weights for all variables, since the cross-validation favored a relatively high λ_a-value.

The resulting canonical weights are considerably more sparse for each variable set, making the questionnaire weights especially better for interpretation. However, the MEG coefficients corresponding to just a few individual sensor locations are challenging to interpret from a theoretical α-activity standpoint, considering the relatively low spatial resolution of MEG, while larger areas of activation might be more desirable. Also, all of the canonical correlations are still extremely high, making all pairs close to equal in interpretational importance.

For the third analysis, the number of variables in the MEG set were reduced via principal component analysis, by substituting the MEG variables with the scores of the first 12 principal components. Figure 4.1 illustrates the non-zero eigenvalues of the principal components calculated from the standardized MEG data, and Figure 4.2 shows the variance explained by the same principal components, quickly diminishing after the sixth component. The number of components to use was determined by including enough principal components to account for 95% of the variance in the data, which was achieved with 12 components. The analysis was performed similarly to the second analysis, but now with 7 and 12 variables in the two sets, resulting in a noticeable drop-off in the canonical correlations of the estimated pairs. The canonical weights of the principal component score variables also produce larger groups of MEG sensor locations gaining similar weights, when the principal component rotation matrix is used to calculate the canonical weights for the original 306 variables. These larger areas of activation are more desirable for interpretation as they are more in line with the spatial resolution of MEG.

Lastly, the previous analysis was performed again without the scores of the third principal component, as it was deemed possible to represent a heart artifact, which would represent noise unrelated to α-activity of interest. The remaining 11 principal components accounted for roughly 83% of the variance in the MEG data. The tuning parameter associated with the principal component variable set of the second canonical pair was lowered to avoid all score variables gaining zero weights, since cross-validation resulted in a relatively high tuning parameter value being chosen.

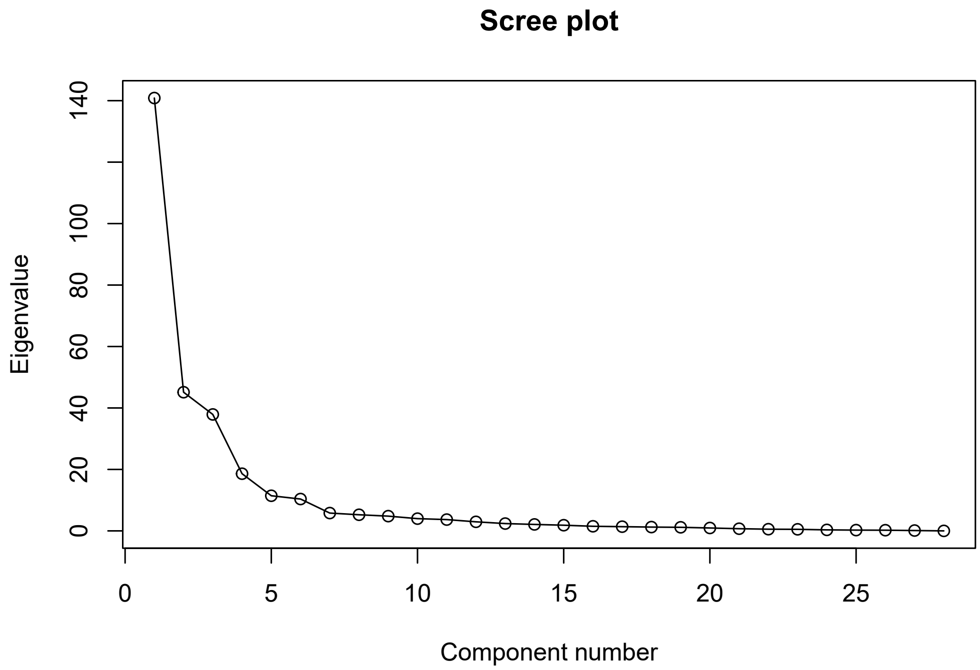


Figure 4.1: Scree plot, eigenvalues the of the 28 non-zero principal components.

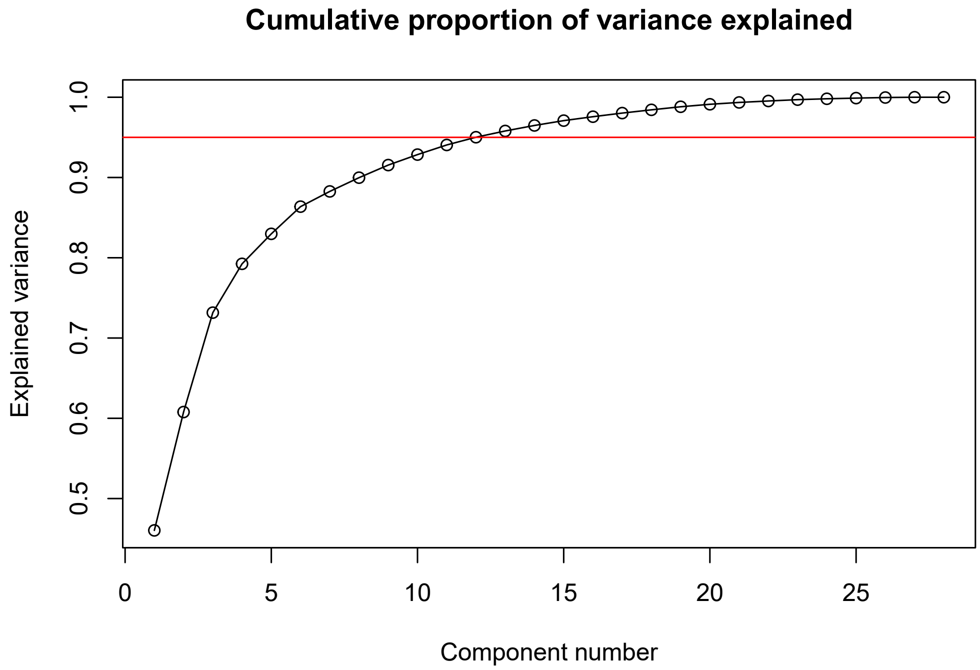


Figure 4.2: Plot of cumulative proportion of variance explained by each of the 28 non-zero principal component. 95% cut-off point at 12 principal components visualized with a horizontal line.

5 Results

This section presents the results of the four analyses described in Section 4. Only the MEG figures relevant to interpretation and illustrative examples of the results are included, while all other figures are included in Appendix A. The analysis results are discussed in the same order as the analyses in the previous section.

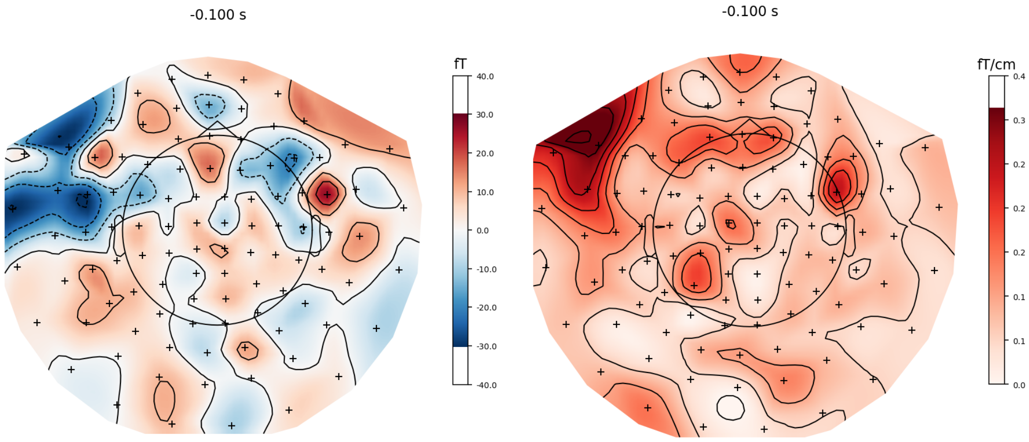
5.1 SCCA without penalization

When no penalization is used, all canonical coefficients of each canonical pair gain non-zero values as can be seen in Table 2 for the questionnaire variables. Plots of the coefficient values gained by the MEG variables for the first canonical pair are presented in Figure 5.1. Each of the plots shows a 102 sensor unit array that surrounds the head during an MEG imaging, and a top-down illustration of a head to help approximate the areas the sensor units correspond to. The sensor array points are used to create a topographic map of the sensors i.e. the MEG variables that gain coefficient values, with blue signifying negative, red positive and white zero coefficient values. While there are three types of sensors, four images are plotted for each canonical pair, the magnetometers, the two types of planar gradiometers individually, and a plot showing the root mean squares (RMS) of the two planar gradiometers in each sensor location. This last type of plot shows a better overall view of the coefficients gained by the planar sensors, but due to RMS, all coefficients are positive valued. Figure 5.1 and other MEG coefficient plots were created using the MNE-Python package (Gramfort et al., 2013). The time stamps at the top and units of measure should be ignored.

The coefficients of the magnetometer variables are shown in Figure 5.1a, while the root mean square values of planar gradiometer variables are shown in Figure 5.1b. From Figure 5.1 can be seen that all MEG variables gain non-zero coefficients, as no sensor locations appear completely white. Also, the signs of the coefficients vary frequently from sensor to sensor, making interpretations of α-activity in the corresponding brain areas difficult, as MEG's spatial resolution would not be enough to detect such frequent changes in magnetic fields. Larger joint areas of similar coefficient values would be more desirable for interpretation. Additionally, Table 3 shows that the canonical correlations of all canonical pairs are extremely close to one, making prioritization for interpretation between canonical pairs highly difficult.

Table 2: Canonical coefficients of the questionnaire variables for all seven canonical pairs from SCCA without penalization.

	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7
BDI	-0.449	0.377	-0.546	0.038	0.874	-0.321	-0.132
BAI	-0.564	0.194	-0.042	-0.989	-0.406	0.221	0.382
BAS Drive	0.134	0.373	0.520	-0.422	0.845	-0.255	0.567
BAS Fun	-0.226	0.404	-0.191	0.425	-0.203	-0.231	0.977
BAS Reward	0.075	0.321	0.084	0.173	-0.622	-0.468	-1.026
BIS	-0.101	0.053	0.459	0.552	-0.157	1.301	-0.149
BVS	-0.169	-0.460	0.462	0.529	0.044	-1.287	0.335



(a) Magnetometer channels (b) Planar gradiometer channels

Figure 5.1: Pair 1 canonical coefficients of magnetometer channels and root mean squares of coefficients of planar gradiometer channels plotted according to sensor locations from SCCA without penalization.

Table 3: Canonical correlations of all canonical pairs from SCCA without penalization.

Pair number	1	2	3	4	5	6	7
Canonical correlation	0.999	0.999	0.999	0.999	0.999	0.999	0.999

5.2 SCCA with penalization

Figure 5.2 shows a perspective plot of the cross-validation performance measure ρ in the 25×25 grid of candidate tuning values between 0 and 0.049, for the first canonical pair. The cross-validation can be seen to be quite unstable with many possible ρ-values gaining similar large values. This instability can cause largely varying tuning parameter values being chosen for different cross-validations. As an example, cross-validation of the first canonical pair resulted in values $\lambda_a = 0.013$ and $\lambda_b = 0.039$ prompting very sparse coefficients for the first canonical pair. Table 4 lists the canonical coefficients for the questionnaire variables for each canonical pair, and it can be seen that the sparsity between pairs varies highly, with the third pair being reduced to almost one variable and pairs 2 and 5 barely retaining all non-zero coefficients.

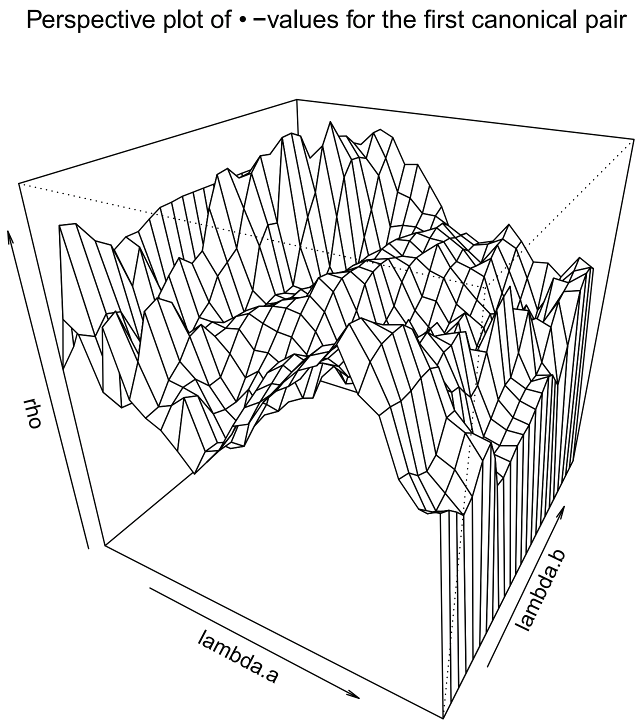
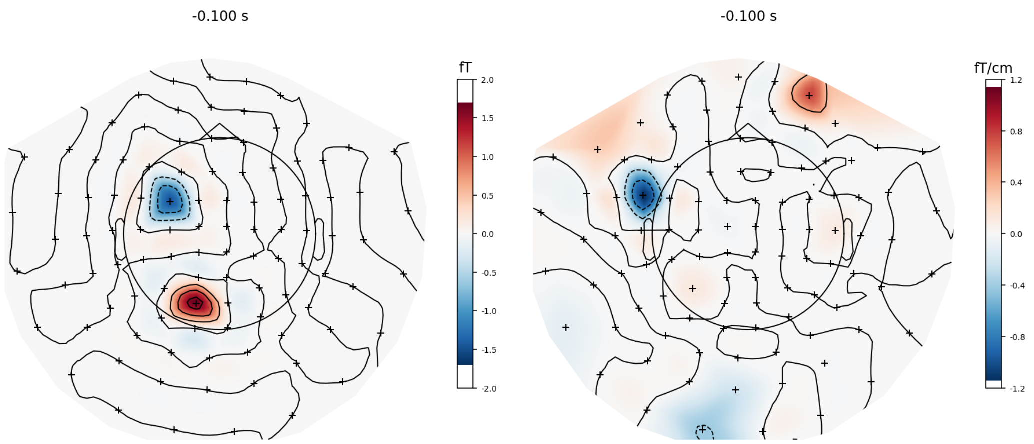


Figure 5.2: ρ-values plotted in the 25×25 grid of candidate tuning parameter values.

Table 4: Canonical coefficients of the questionnaire variables for all seven canonical pairs from SCCA with penalization.

	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7
BDI	0	-0.541	0	0	0.794	0.592	-0.540
BAI	-0.979	0.096	0	-0.625	-0.330	0.244	0.095
BAS Drive	0	-0.304	1.009	0	0.018	0.629	0.112
BAS Fun	-0.479	-0.194	0	0.207	0.596	0	0.857
BAS Reward	-0.099	-0.061	0	0	0.031	-1.070	-0.719
BIS	0	1.354	0	0.606	-0.047	0.126	0
BVS	0	-1.212	0.050	0.814	-0.256	-0.142	0.267

Similarly, Figure 5.3 illustrates that the MEG variable coefficients are highly sparse after penalization, showing high coefficient values for only a few sensor areas for both magnetometer and planar gradiometer variables. From an interpretational standpoint, the first canonical pair seems much simpler to interpret when compared to coefficients of the first canonical pair from the first analysis. As an example, a possible interpretation would be that higher α-activity during the interoception task in the left parietal and right frontal lobe (positive areas of Figure 5.3) would be associated with lower scores on the Beck Anxiety Inventory and Fun seeking section of the BAS Scale, while higher α-activity in the left frontal lobe (negative areas of Figure 5.3) during the interoception task would be associated with higher scores on the same questionnaires. The second set of planar channels were ignored in this example since coefficients shown in Figure 5.3b seem larger based on Figure 5.4, showing the RMS of planar channel coeffiecients. This interpretation is done without expert knowledge of the field and should be treated only as an illustrative interpretation of the method used.



(a) Magnetometer channels

(b) Planar 1 channels

Figure 5.3: Pair 1 canonical coefficients of magnetometer channels and coefficients of planar 1 gradiometer channels plotted according to sensor locations from penalized SCCA. Areas approximately in the left parietal and the right frontal lobe gain positive coefficients and an area in the left frontal lobe gains negative coefficients.

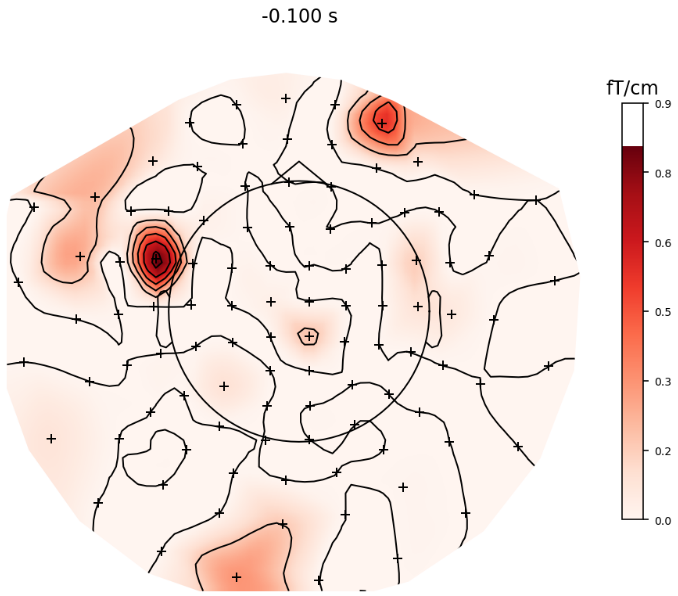


Figure 5.4: Root mean squares of canonical coefficients of planar gradiometer channels for the first canonical pair.

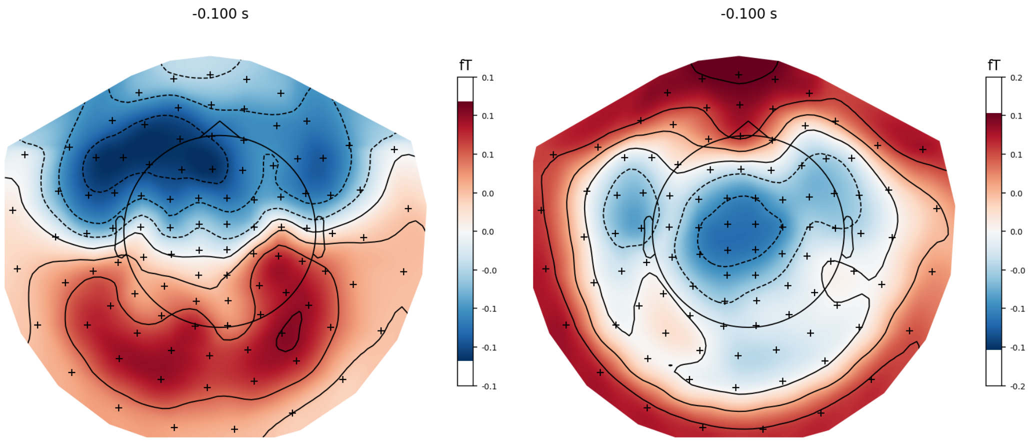
As shown in Table 5 all canonical pairs still have very high canonical correlations, and not much difference in these values makes choosing pairs for interpretation difficult. To remedy this issue, Principal Component Analysis was used to decrease the number of variables in the MEG variable set, as described in Section 4.

Table 5: Canonical correlations of all canonical pairs from SCCA with penalization.

Pair number	1	2	3	4	5	6	7
Canonical correlation	0.992	0.992	0.979	0.960	0.961	0.966	0.943

5.3 SCCA with 12 principal components

To give some interpretation to the first few principal components (PCs), the first PC gives all sensor variables similar weights, acting as a mean value component, while the other PCs give different weights to sensors in different brain areas. Figure 5.5 illustrates the principal component loading of the second and the third PC. From Figure 5.5a can be seen that the second PC gives opposing weights to sensor variables in the frontal and posterior parts of the head. Figure 5.5a shows how the third PC gives opposing weights to sensors situated at the top and the peripheral areas of the head. As further examples but not illustrated here, the fourth PC assigns opposing weights to the right side of the head as opposed to the top and peripheral areas, and the fifth PC gives opposite weights to the left side of the head as opposed to all other areas. The first 12 PCs included in this analysis explained 95% of the variance in the data.



(a) Second principal component loadings (b) Third principal component loadings

Figure 5.5: Second and third principal component loadings plotted according to sensor locations.

With a much lower number of 12 PC score variables replacing the 306 MEG variables, the canonical correlations as seen in Table 6 show a clear drop-off in value for the larger pair numbers. This indicates a higher importance for the first few canonical pairs in terms of interpretation. However, tuning parameters chosen by cross-validation achieve less sparsity on the coefficients of each variable set as can be seen from Tables 7 and 8, showing all coefficients of the canonical pairs for each variable set. Despite lack of sparsity, similarities to the previous analysis questionnaire coefficients can be seen in Tables 7 and 4. For the first canonical pair, BAI and BAS Fun questionnaires still gain large coefficients with same signs, but the signs differ between the analyses. However, this sign change does not change the interpretation of the pair entirely, since the signs of the coefficients seen in topographic maps in Figures 5.3b and 5.6b also change. The magnetometer channel coefficients seen in Figure 5.6a now show somewhat similar area activations as well, while the left frontal lobe still gains high coefficient values similar to coefficients seen in Figure 5.3. Also in the second canonical pair, BIS and BVS gain high opposing weights in both analyses.

Table 6: Canonical correlations of all canonical pairs from SCCA using 12 principal components.

Pair number	1	2	3	4	5	6	7
Canonical correlation	0.930	0.844	0.783	0.684	0.508	0.513	0.234

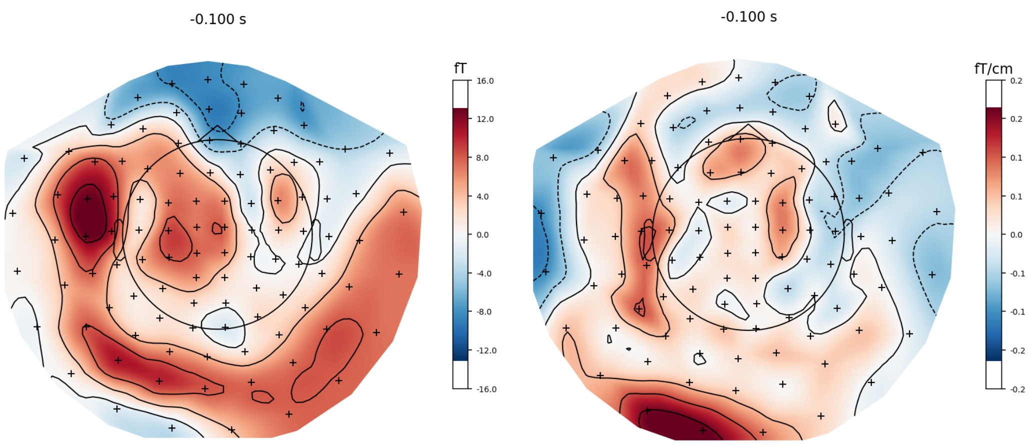
Table 7: Canonical coefficients of the questionnaire variables for all seven canonical pairs from SCCA using 12 principal components.

	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7
BDI	0.116	0.220	1.045	-0.360	0.361	-0.186	0
BAI	0.850	-0.583	-0.114	0	0	-0.641	0
BAS Drive	-0.433	0	0.492	0.471	0.127	-0.705	-0.672
BAS Fun	0.503	0.233	0.455	0.462	-0.053	0.511	-0.437
BAS Reward	0.035	0	-0.697	-0.634	0.792	0.468	0
BIS	0.298	0.772	-0.621	0.687	0	-0.252	0.522
BVS	-0.546	-0.700	0.398	0.508	0.329	0.963	0

The coefficient values for the original MEG variables to produce Figure 5.6 were calculated by multiplying the rotation matrix of the principal component analysis by the SCCA coefficient matrix, values of which are shown in Table 8. The resulting topographic maps, as seen in Figure 5.6, of the sensor variable coefficients show much larger areas of similarly signed weights, preferable for topographic map interpretation of brain areas. These maps are more in line with the spatial resolution of MEG.

Table 8: Canonical coefficients of the principal component score variables from SCCA using 12 principal components.

	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7
PC1	-0.311	0.097	-0.363	0.586	-0.186	0.296	-0.138
PC2	0.106	0.398	-0.406	0.030	0.448	-0.074	0.407
PC3	-0.480	0.265	-0.059	-0.417	0.010	-0.184	-0.212
PC4	-0.061	-0.030	-0.237	-0.009	-0.125	-0.104	-0.364
PC5	0.362	0.058	-0.070	-0.255	-0.674	-0.213	0.409
PC6	0	-0.423	-0.597	-0.005	-0.088	-0.335	0
PC7	-0.095	-0.305	-0.238	-0.152	-0.090	-0.069	0
PC8	-0.027	-0.634	0.136	0.042	0.391	-0.107	0
PC9	0.213	-0.004	-0.442	-0.441	0.283	0.336	0
PC10	0.684	0.050	-0.013	0.231	0.110	0.016	-0.476
PC11	-0.061	-0.223	-0.090	0.120	-0.184	0.479	0.444
PC12	0	0.178	-0.059	0.360	0	-0.588	0.217



(a) Magnetometer channels

(b) Planar1 channels

Figure 5.6: Pair 1 canonical coefficients of magnetometer channels and planar1 gradiometer channels plotted according to sensor locations from SCCA using 12 principal components.

5.4 SCCA with 11 principal components

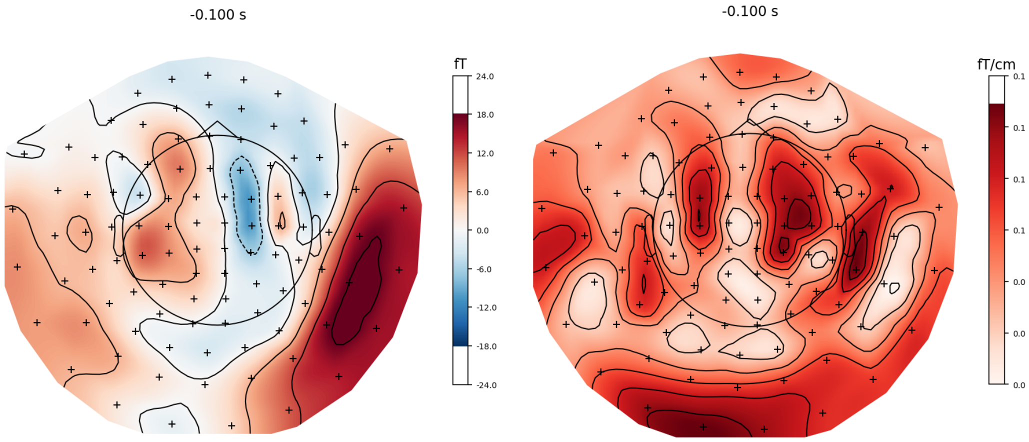
Based on expert opinion of Tiina Parviainen from the Center for Interdisciplinary Brain Research (CIBR), the third principal component was identified as a possible heart artifact, and was removed from the data for a final analysis with 11 PCs, explaining 83% of the variance in the MEG data. Table 9 shows that the canonical correlations diminish even more when the number of variables in the PC score set is further dropped by one. The canonical coefficient values for the questionnaire variables now differ considerably from previous analyses, with BDI and BVS gaining largest weights in addition to BAS Fun for the first pair, and BAI and BAS Drive for the second pair. The first pair planar gradiometer variables still retain high weights for the left occipital lobe, however, as shown by Figures 5.6b and 5.7b as a red area in the lower left of both figures. The magnetometer variables also show slight similarities in weights in Figures 5.6a and 5.7a

Table 9: Canonical correlations of all canonical pairs from SCCA using 11 principal components.

Pair number	1	2	3	4	5	6	7
Canonical correlation	0.875	0.817	0.739	0.545	0.554	0.409	-0.398

Table 10: Canonical coefficients of the questionnaire variables from SCCA using 11 principal components.

	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7
BDI	0.518	-0.177	-0.953	0.218	0.035	0.400	0.459
BAI	0	0.987	0	0.267	-0.478	0.200	0
BAS Drive	-0.130	-0.599	0	0.493	-0.256	0.843	-0.231
BAS Fun	0.600	-0.083	0	0.223	-0.546	-0.470	0
BAS Reward	0	0.472	0.188	0	1.209	0	-0.714
BIS	0.478	-0.210	0.953	0	0	0	0
BVS	-0.646	-0.483	-0.247	0.758	0	-0.641	0.042



(a) Magnetometer channels (b) Planar gradiometer channels

Figure 5.7: Pair 1 canonical coefficients of magnetometer channels and root mean squares of coefficients of planar gradiometer channels plotted according to sensor locations from SCCA using 11 principal components.

As a final note to this section, the p-values of the permutation tests performed on the first canonical pair of each analysis can be found in Table 11. None of the first canonical pairs are significant, since all of the p-values are much greater than 0.05.

Table 11: p-values of the permutation tests performed on the first canonical pairs of each of the four analyses

Analysis	CCA	SCCA	12 PC SCCA	11 PC SCCA
p-value	0.26	0.76	0.31	0.49

6 Conclusion

None of the first canonical pairs of the four previous analyses were significant based on the permutation tests described in Section 4, so interpretations of the results should be made with caution. Also, patterns most expected based on the correlation matrix of the questionnaire variables in Figure 3.1 were not found, that is, opposing weights to the three BAS variables and any of the other questionnaire variables, and similarly signed weights to the four strongly correlated questionnaires, BDI, BAI, BIS and BVS.

The most consistent results were the similarly signed high weights for BAI and BAS Fun seeking in the first canonical pair and oppositely signed high weights of BIS and BVS for the second pair in analyses using SCCA with penalization and SCCA with 12 PCs. The sensor variables in these pairs also gained somewhat similar weights, even though the latter analysis had weaker sparsity in the coefficients. As to the research question, the lower BAI and BAS Fun scores associated with higher α-activity in the left parietal and right frontal lobe and and higher BAI and BAS Fun scores associated with higherα-activity in the left frontal lobe during the interoception task would be the main observed pattern, but without significant canonical pairs, the result seems speculative.

Overall, sparse canonical correlation eases the interpretation of canonical pairs in high dimensional setting, where several coefficients gain similarly high values, and can help reveal variables of most interest. With MEG data, however, overly sparse coefficients corresponding to certain brain areas can be questionable to interpret, when the spatial resolution of MEG should not be able to identify large changes in magnetic field of neighboring areas. For this purpose, the larger areas of activation present in topographic maps produced from PCA score variables are more beneficial. PCA is also useful when dimension reduction achieves smaller canonical correlations of the canonical pairs by limiting the number of variables available for SCCA. This can help discern canonical pairs which are most important for interpretation. Still dimension reduction feels counter-intuitive for SCCA, method created specifically for high dimensional data. Maybe the problem of several pairs of high correlation can be circumvented by focusing only on the first canonical pair.

To further the analyses carried out, a larger sample size might help with stabilizing the cross-validations to achieve more consistent tuning parameter values. This might be problematic with MEG, when collecting data takes considerable time and effort. With a large enough sample, dedicated test and validation sets could be used to validate the choice of tuning parameters, as was done by Mai and Zhang (2019). Also increasing folds in the k-fold cross-validation could remedy the instability, with the cost of computation time. Performing fairly unstable cross-validation for each canonical pair leads to inconsistent choices of tuning parameters between different analyses. If easier interpretation is the main goal of the analysis, the tuning parameter values could also be set manually for each pair to produce sufficiently sparse coefficients, with the cost of canonical correlations.

Also using different penalization for the MEG variables might improve the analyses, since the l_1 penalty does not take into consideration the grouping of the sensors in the sensor units. Especially the group lasso penalty could be ideal for this data, to include or drop the sensor variables as groups defined by the sensor units.

References

Beck, A. T., N. Epstein, G. Brown, and R. A Steer (1988). “An Inventory for Measuring Clinical Anxiety: Psychometric Properties”. *Journal of Consulting and Clinical Psychology* 56.6, pp. 893–897. DOI: https://doi.org/10.1037/0022-006X.56.6.893.

Beck, A. T., C. H. Ward, M. Mendelson, J. Mock, and J. Erbaugh (1961). “An Inventory for Measuring Depression”. *Archives of General Psychiatry* 4.6, pp. 561–571. DOI: https://doi.org/10.1001/archpsyc.1961.01710120031004.

Beck, Aaron T., Robert A. Steer, and Margery G. Carbin (1988). “Psychometric Properties of the Beck Depression Inventory: Twenty-five Years of Evaluation”. *Clinical Psychology Review* 8.1, pp. 77–100. DOI: https://doi.org/10.1016/0272-7358(88)90050-5.

Carver, C. S. and T. L. White (1994). “Behavioral Inhibition, Behavioral Activation, and Affective Responses to Impending Reward and Punishment: the BIS/BAS Scales”. *Journal of Personality and Social Psychology* 67.2, p. 319. DOI: https://doi.org/10.1037/0022-3514.67.2.319.

CIBR (2024). *Meg principles*. https://cibr.jyu.fi/en/intranet/meg-skills/introduction/copy2_of_introduction. Accessed: 9-10-2024.

Gao, C., Z. Ma, and H. H. Zhou (2017). “Sparse CCA: Adaptive Estimation and Computational Barriers”. *The Annals of Statistics* 45.5, pp. 2074–2101. DOI: https://doi.org/10.1214/16-AOS1519.

Garfinkel, S. N. and H. D. Critchley (2013). “Interoception, Emotion and Brain: New Insights Link Internal Physiology to Social Behaviour. Commentary on: “Anterior Insular Cortex Mediates Bodily Sensibility and Social Anxiety” by Terasawa et al. (2012)”. *Social Cognitive and Affective Neuroscience* 8.3, pp. 231–234. DOI: https://doi.org/10.1093/scan/nss140.

Gramfort, Alexandre, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen (2013). “MEG and EEG Data Analysis with MNE-Python”. *Frontiers in Neuroscience* 7.267, pp. 1–13. DOI: 10.3389/fnins.2013.00267.

Hämäläinen, M., R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa (Apr. 1993). “Magnetoencephalography—Theory, Instrumentation, and Applications to Noninvasive Studies of the Working Human Brain”. *Reviews of Modern Physics* 65 (2), pp. 413–497. DOI: https://doi.org/10.1103/RevModPhys.65.413.

Haslam, N. (2007). *Introduction to Personality and Intelligence*. SAGE Publications Ltd. Chap. 7. DOI: https://doi.org/10.4135/9781446279144.

Hotelling, H. (1936). “Relations Between Two Sets of Variates”. *Biometrika* 28.3/4, pp. 321–377. DOI: https://doi.org/10.2307/2333955.

James, G., D. Witte, T. Hastie, and R. Tibshirani (2021). *An Introduction to Statistical Learning : With Applications in R*. Second. Springer. Chap. 6. URL: https://www.statlearning.com/.

Johnson, R. A. and D. Wichern (2013). *Applied Multivariate Statistical Analysis*. Sixth. Pearson.

Lyyra, P. and T. Parviainen (2018). “Behavioral Inhibition Underlies the Link Between Interoceptive Sensitivity and Anxiety-related Temperamental Traits”. *Frontiers in Psychology* 9.

Mai, Q. and X. Zhang (Feb. 2019). “An Iterative Penalized Least Squares Approach to Sparse Canonical Correlation Analysis”. *Biometrics* 75.3, pp. 734–744. DOI: https://doi.org/10.1111/biom.13043.

Olatunji, B. O., B. J. Deacon, J. S. Abramowitz, and D. P. Valentiner (2007). “Body Vigilance in Nonclinical and Anxiety Disorder Samples: Structure, Correlates, and Prediction of Health Concerns”. *Behavior Therapy* 38.4, pp. 392–401. DOI: https://doi.org/10.1016/j.beth.2006.09.002.

Palva, S. and J. M. Palva (2007). “New Vistas for Alpha-frequency Band Oscillations”. *Trends in Neurosciences* 30.4, pp. 150–158. DOI: https://doi.org/10.1016/j.tins.2007.02.001.

Rencher, A. C. and W. F. Christensen (2012). *Methods of Multivariate Analysis*. John Wiley and Sons, Incorporated.

Rothbart, M. K. and J. E. Bates (2007). *Handbook of Child Psychology*. Sixth. Vol. 3. John Wiley and Sons, Incorporated. Chap. 3.

Schmidt, N. B., D. R. Lerew, and J. H. Trakowski (1997). “Body Vigilance in Panic Disorder: Evaluating Attention to Bodily Perturbations”. *Journal of Consulting and Clinical Psychology* 65 (2), pp. 214–220. DOI: https://doi.org/10.1037//0022-006x.65.2.214.

Tabachnick, B. G. and L. S. Fidell (2014). *Using Multivariate Statistics*. Sixth. Harlow, England: Pearson.

Tibshirani, R. (Jan. 1996). “Regression Shrinkage and Selection Via the Lasso”. *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288. DOI: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

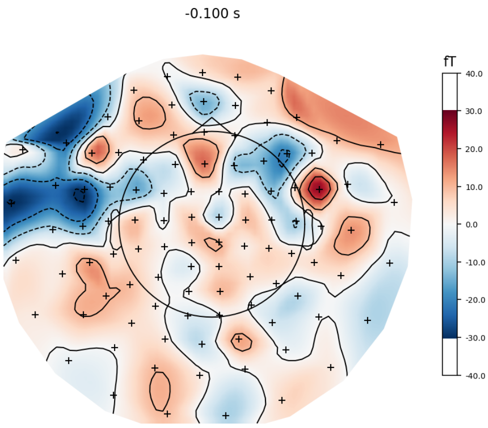
Tibshirani, R., M. Saunders, S. Rosset, J. Zhu, and K. Knight (Dec. 2004). “Sparsity and Smoothness Via the Fused Lasso”. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.1, pp. 91–108. DOI: https://doi.org/10.1111/j.1467-9868.2005.00490.x.

Witten, D. M., R. Tibshirani, and T. Hastie (2009). “A Penalized Matrix Decomposition, with Applications to Sparse Principal Components and

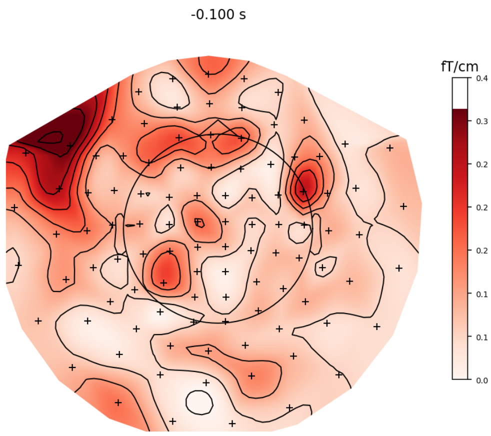
Canonical Correlation Analysis". *Biostatistics* 10.3, pp. 515–534. DOI: https://doi.org/10.1093/biostatistics/kxp008.
Yuan, M. and Y. Lin (Dec. 2005). "Model Selection and Estimation in Regression with Grouped Variables". *Journal of the Royal Statistical Society Series B: Statistical Methodology* 68.1, pp. 49–67. DOI: https://doi.org/10.1111/j.1467-9868.2005.00532.x.
Zou, H. (2006). "The Adaptive Lasso and Its Oracle Properties". *Journal of the American Statistical Association* 101.476, pp. 1418–1429. DOI: https://doi.org/10.1198/016214506000000735.

Appendix A

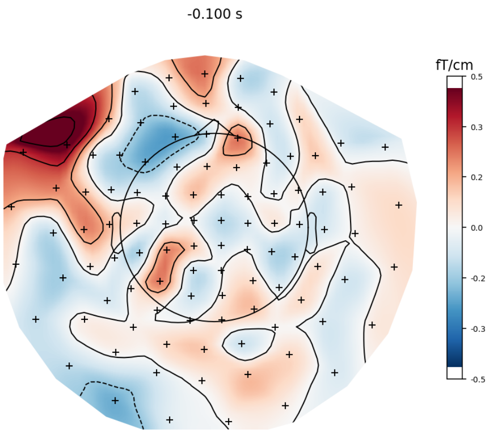
The canonical coefficient plots for the MEG variables for each canonical pair of each analysis are included in this appendix section. Each canonical pair has plots in groups of 4, canonical coefficients of the magnetometer channels (top left), the root mean squares of the planar gradiometer channels (top right) the planar gradiometer 1 (bottom left) and the planar gradiometer 2 channels (bottom right) plotted separately. Each of the four analyses has 28 plots, four plots for each of the seven canonical pairs.



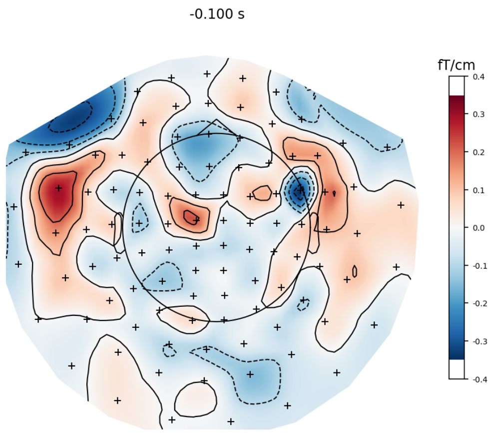
(a) Magnetometers



(b) Gradiometer RMS

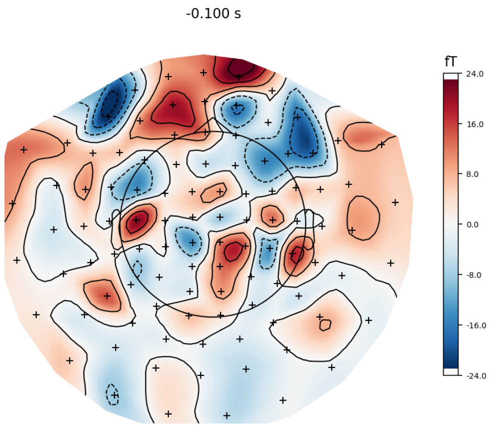


(c) Planar1

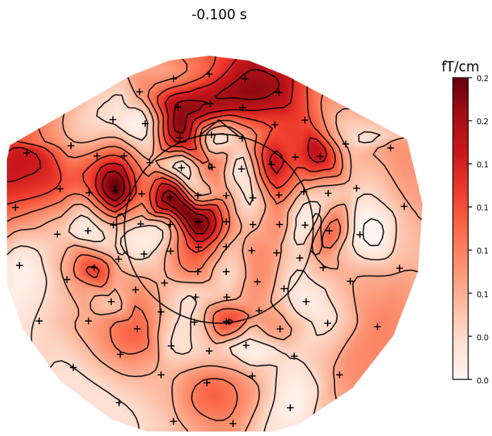


(d) Planar2

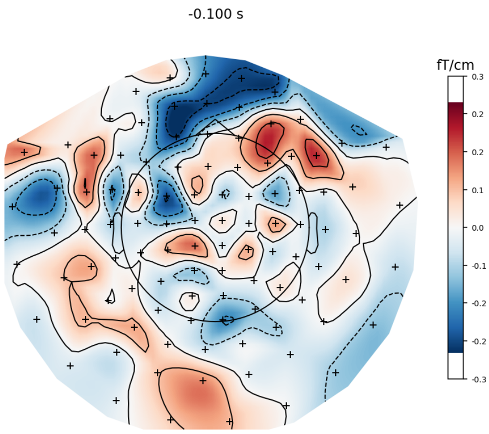
Figure A1: CCA Canonical pair 1



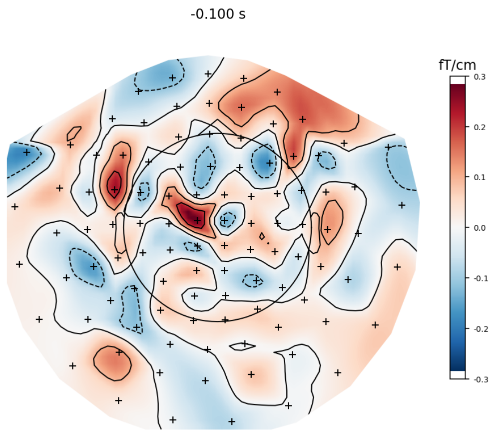
(a) Magnetometers



(b) Gradiometer RMS

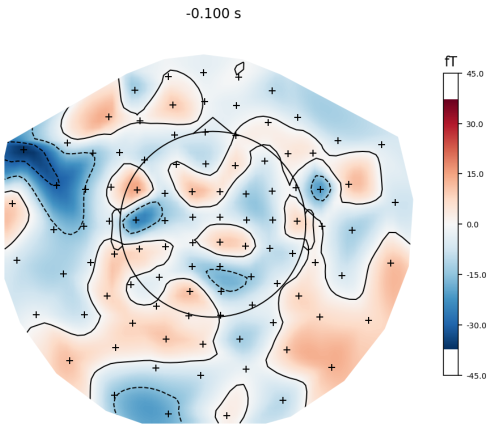


(c) Planar1

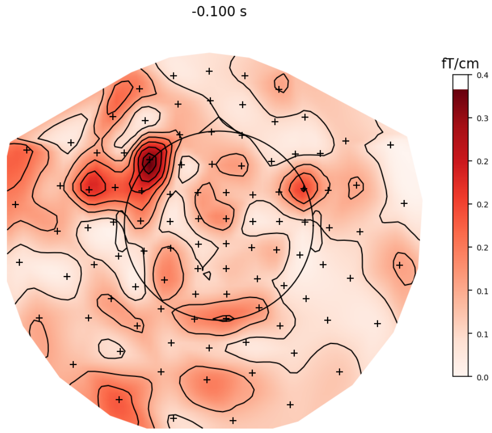


(d) Planar2

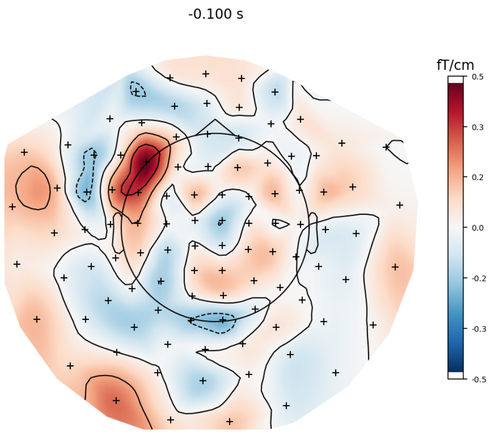
Figure A2: CCA Canonical pair 2



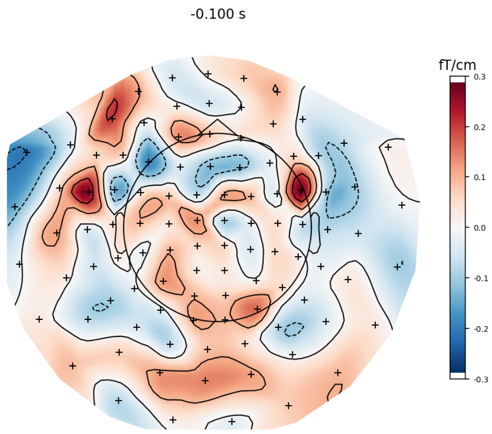
(a) Magnetometers



(b) Gradiometer RMS

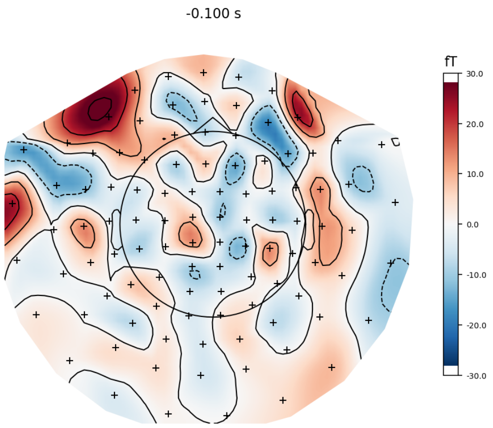


(c) Planar1

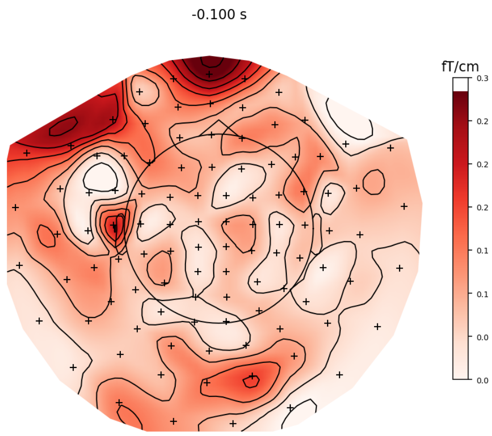


(d) Planar2

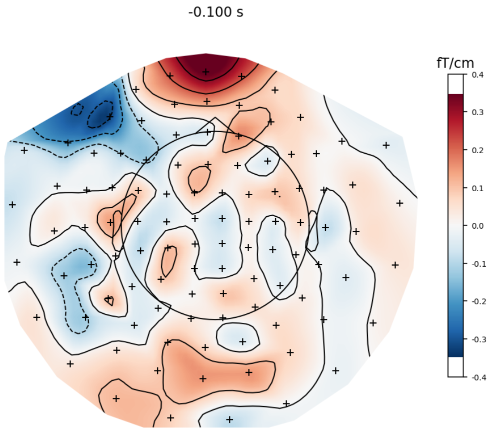
Figure A3: CCA Canonical pair 3



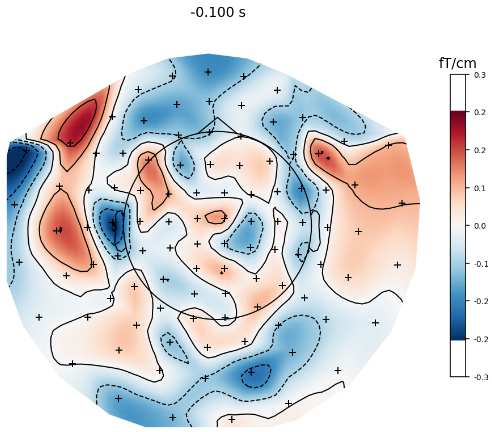
(a) Magnetometers



(b) Gradiometer RMS

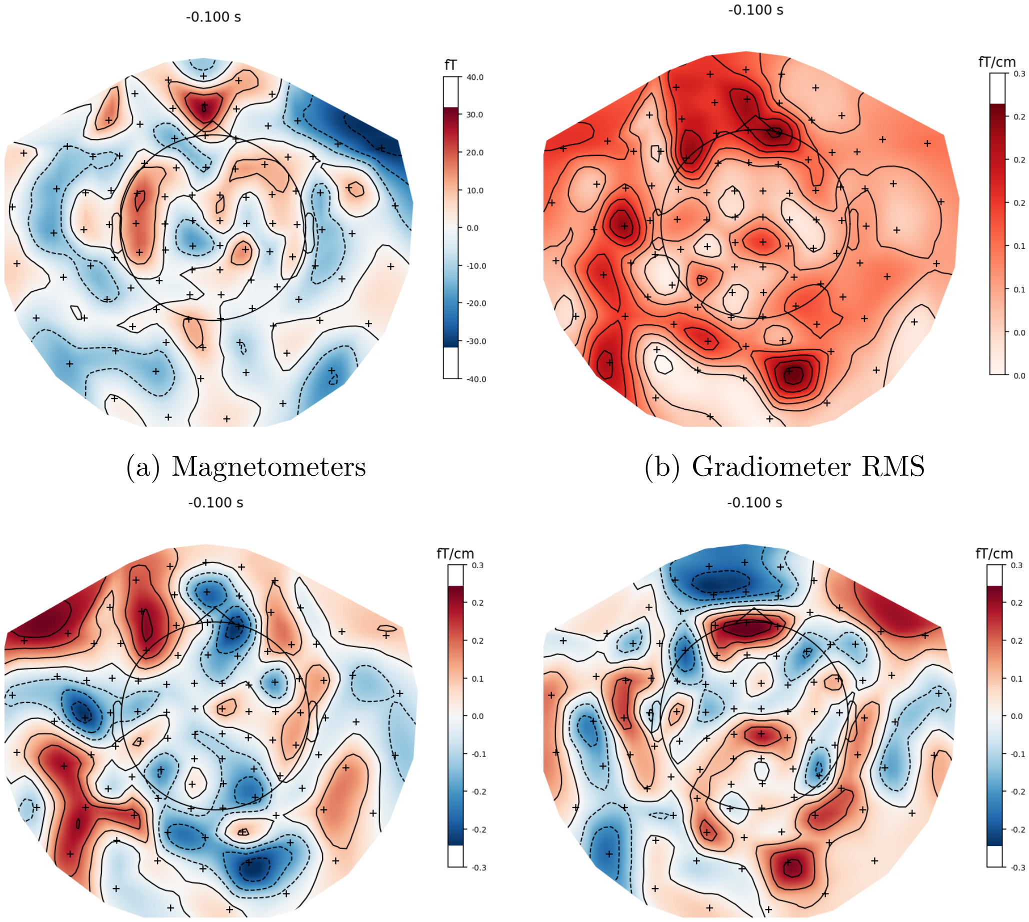


(c) Planar1



(d) Planar2

Figure A4: CCA Canonical pair 4



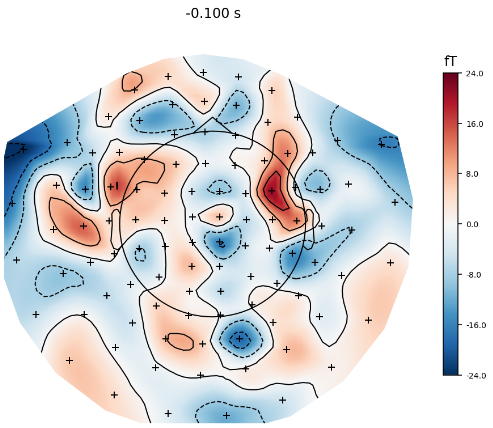
(a) Magnetometers

(b) Gradiometer RMS

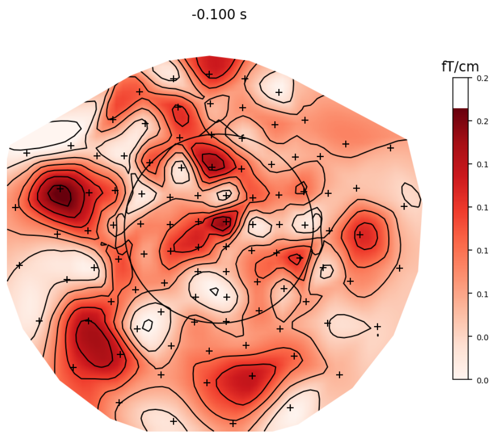
(c) Planar1

(d) Planar2

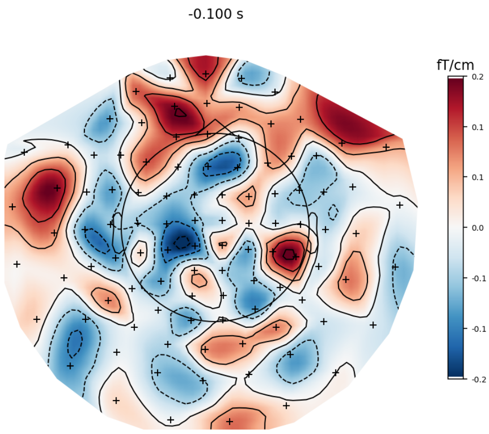
Figure A5: CCA Canonical pair 5



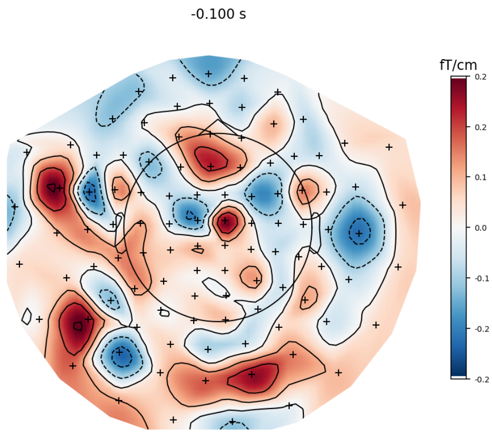
(a) Magnetometers



(b) Gradiometer RMS

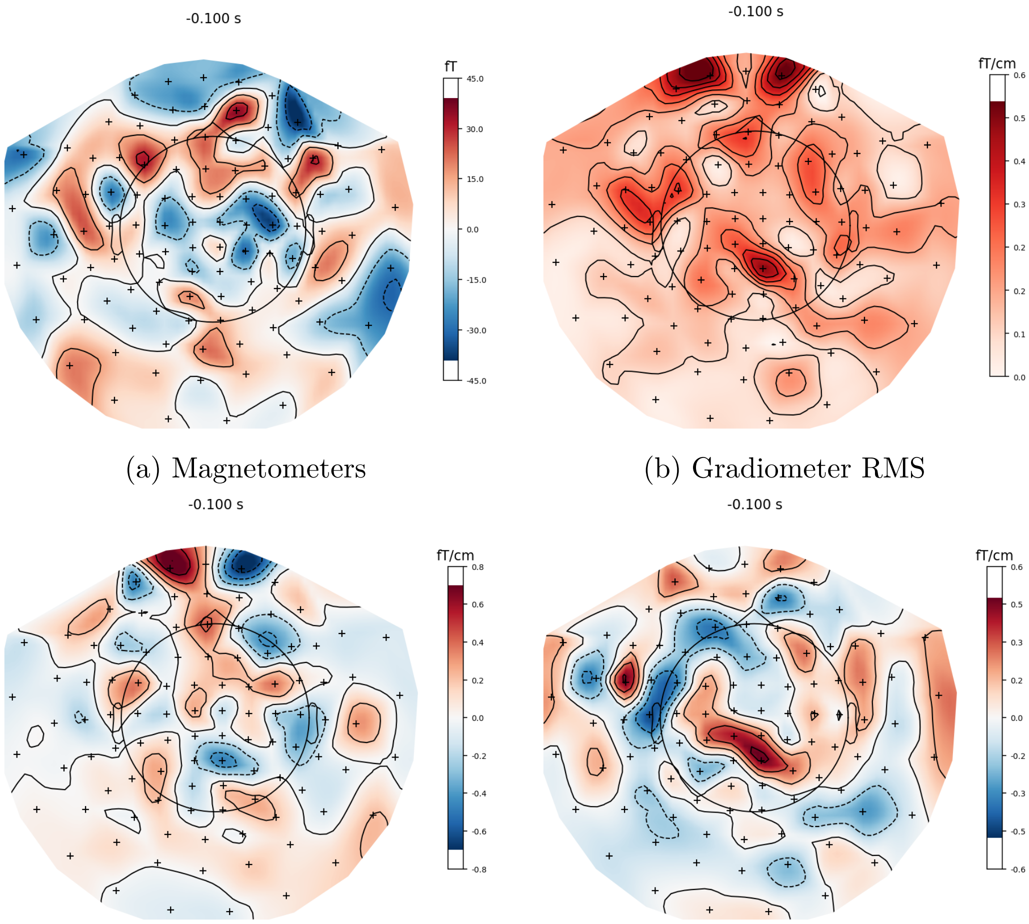


(c) Planar1



(d) Planar2

Figure A6: CCA Canonical pair 6



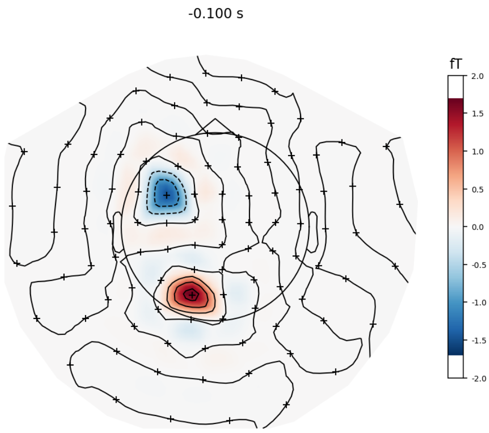
(a) Magnetometers

(b) Gradiometer RMS

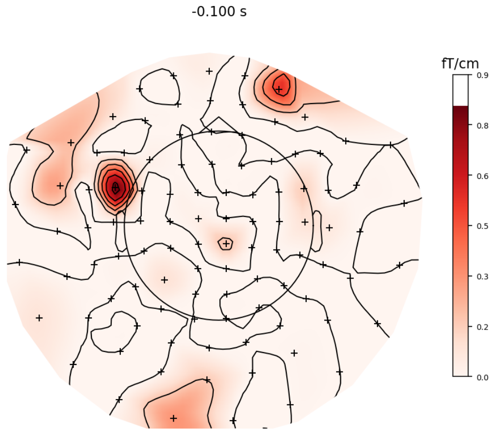
(c) Planar1

(d) Planar2

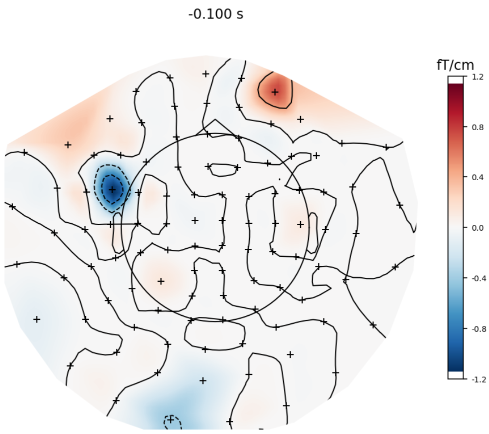
Figure A7: CCA Canonical pair 7



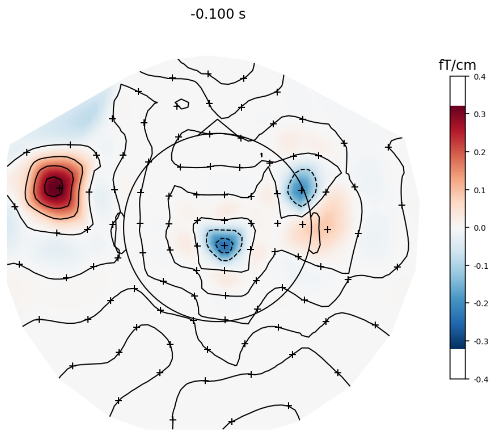
(a) Magnetometers



(b) Gradiometer RMS

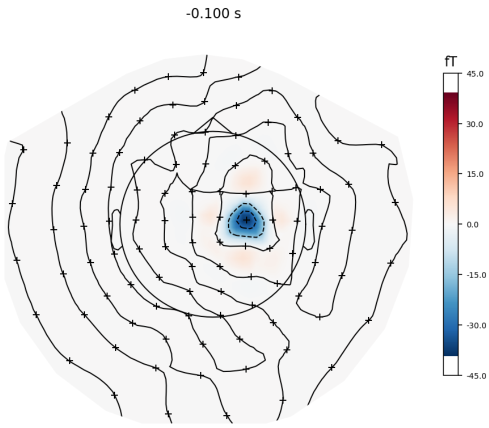


(c) Planar1

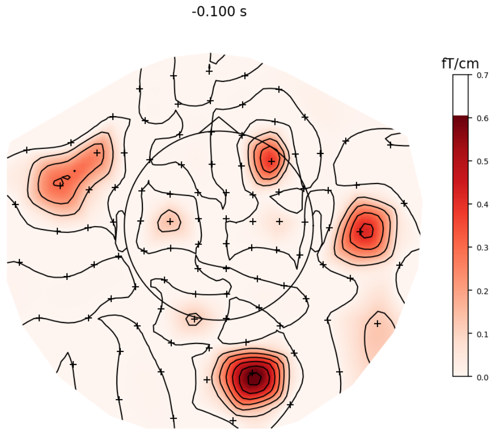


(d) Planar2

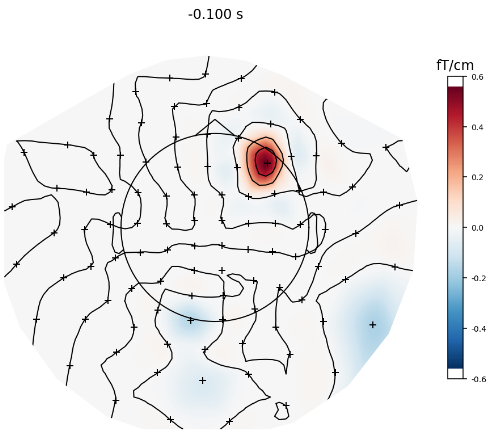
Figure A8: SCCA Canonical pair 1



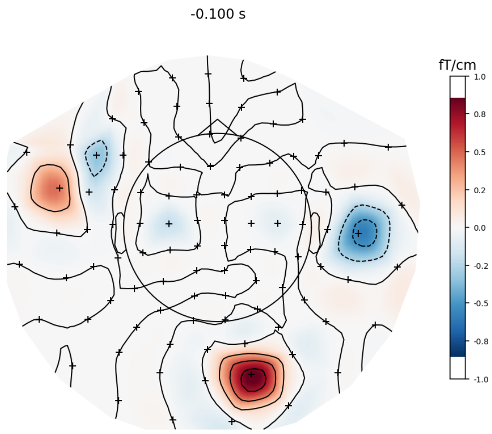
(a) Magnetometers



(b) Gradiometer RMS

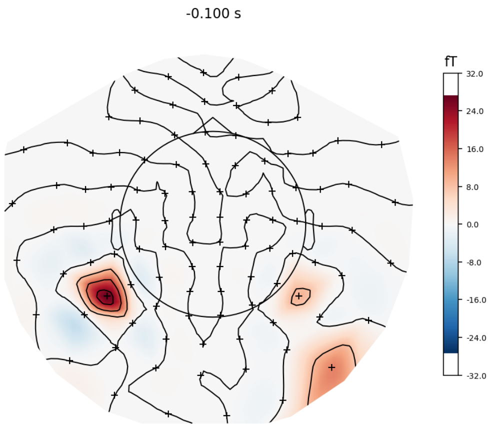


(c) Planar1

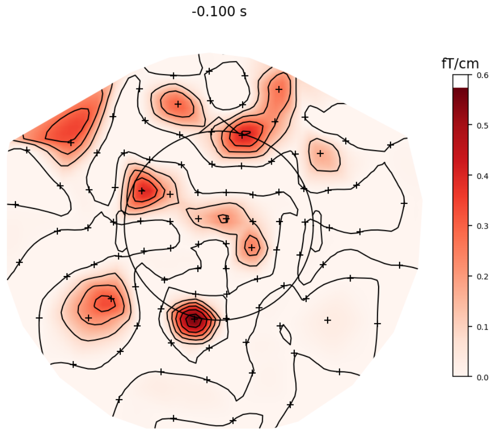


(d) Planar2

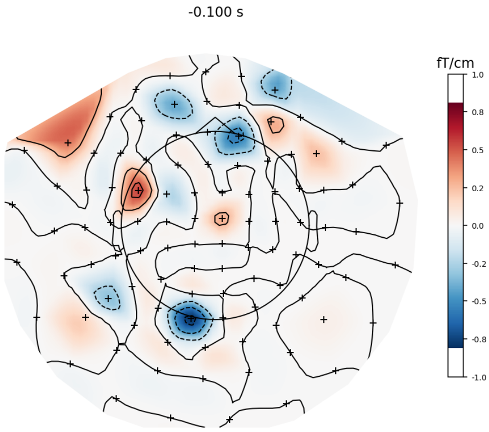
Figure A9: SCCA Canonical pair 2



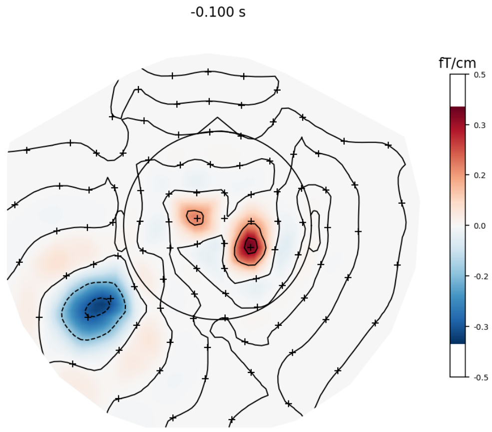
(a) Magnetometers



(b) Gradiometer RMS

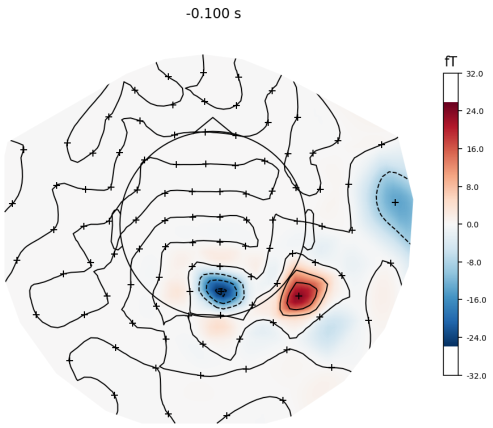


(c) Planar1

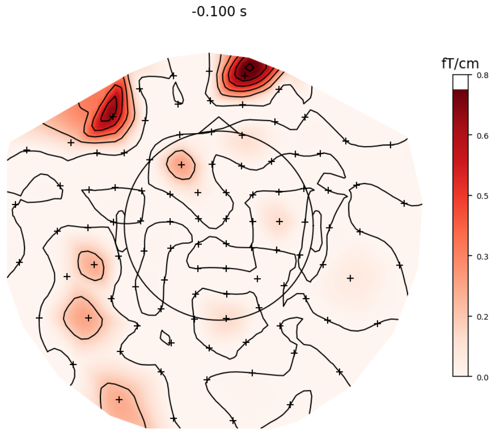


(d) Planar2

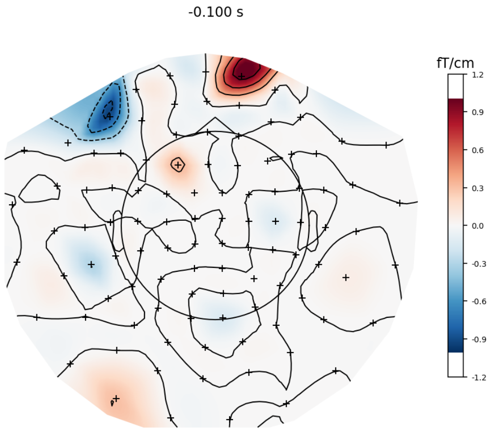
Figure A10: SCCA Canonical pair 3



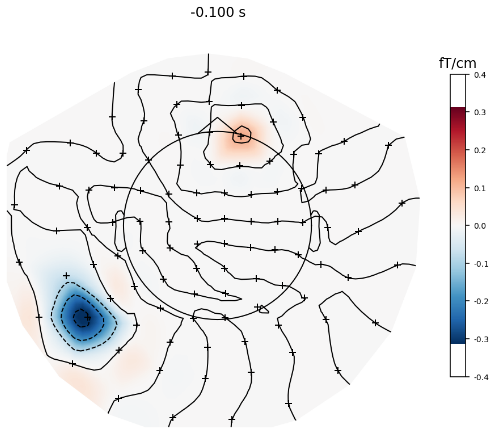
(a) Magnetometers



(b) Gradiometer RMS

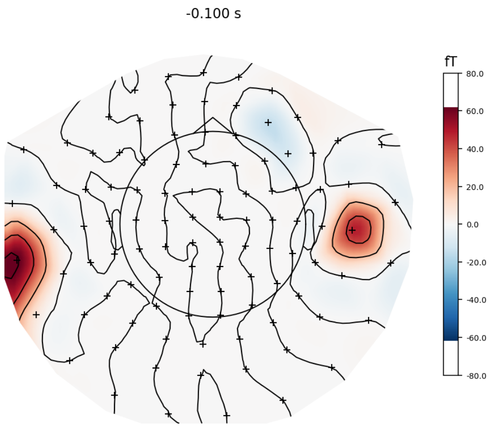


(c) Planar1

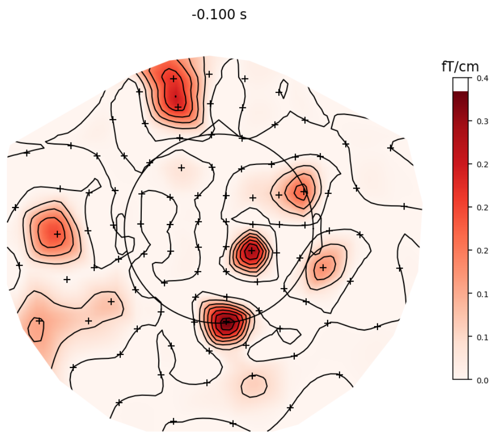


(d) Planar2

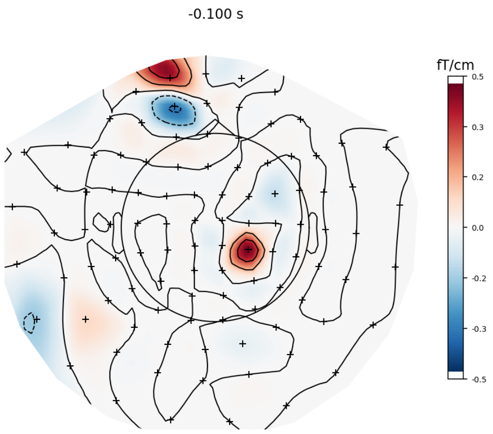
Figure A11: SCCA Canonical pair 4



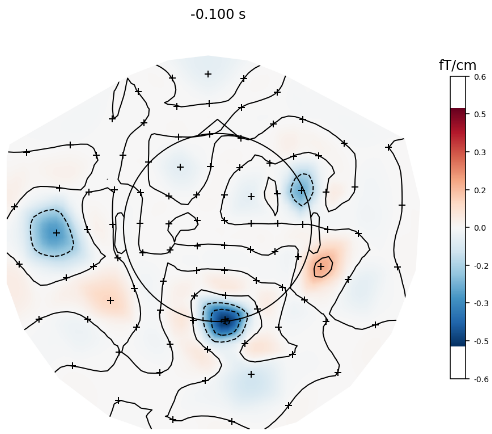
(a) Magnetometers



(b) Gradiometer RMS

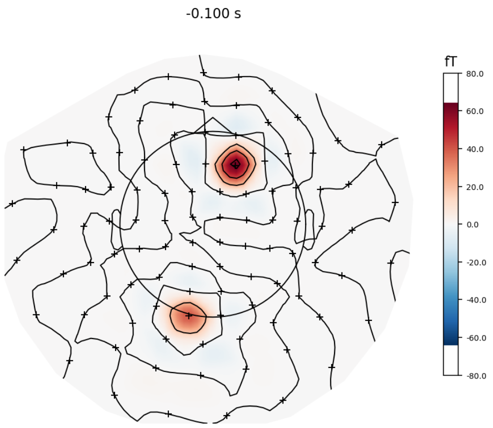


(c) Planar1

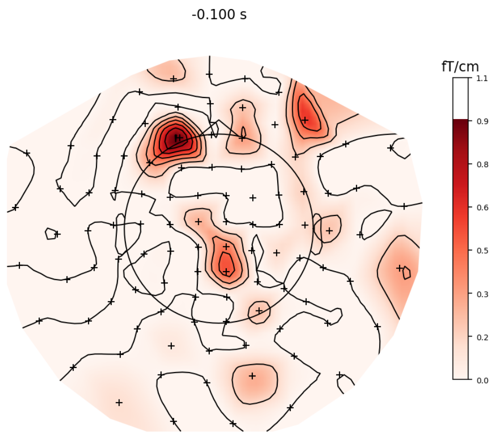


(d) Planar2

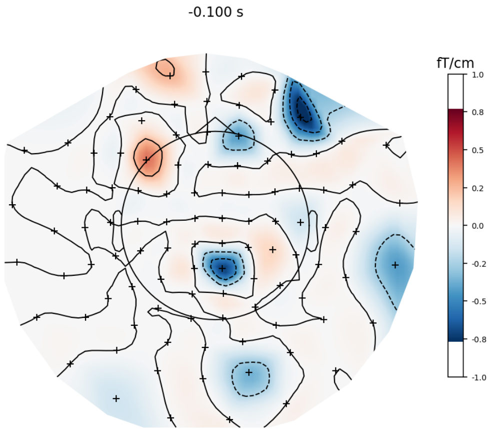
Figure A12: SCCA Canonical pair 5



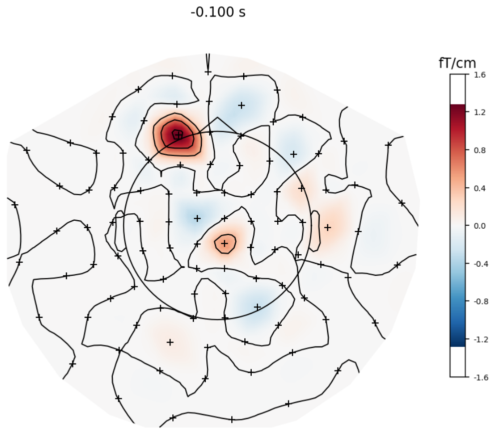
(a) Magnetometers



(b) Gradiometer RMS

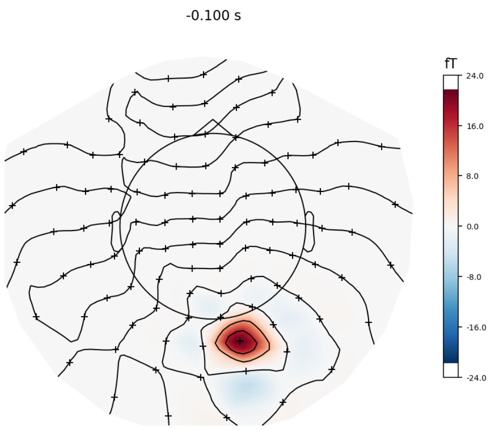


(c) Planar1

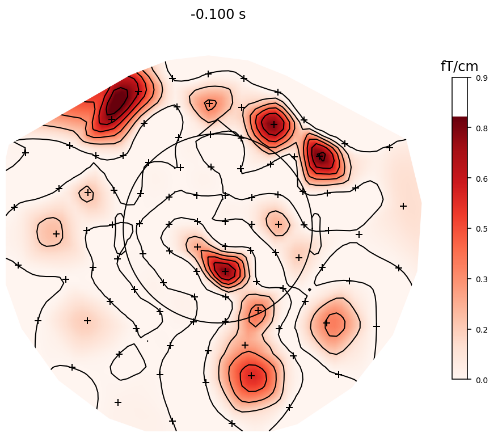


(d) Planar2

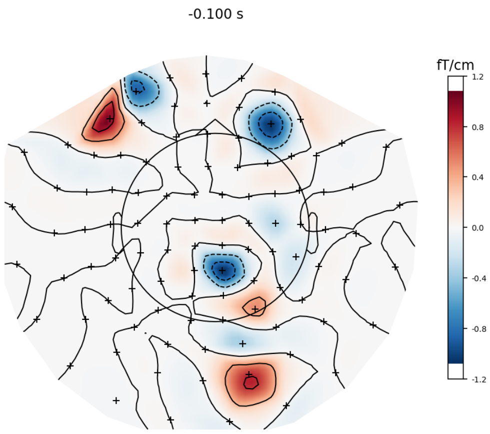
Figure A13: SCCA Canonical pair 6



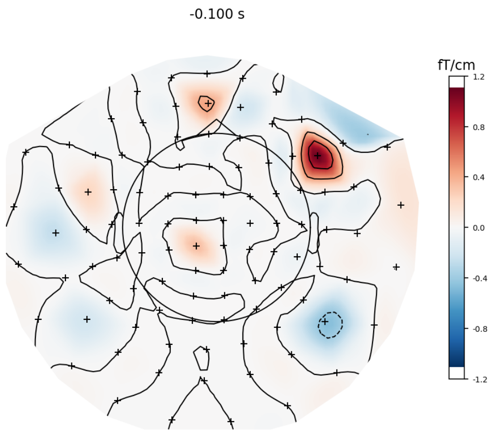
(a) Magnetometers



(b) Gradiometer RMS

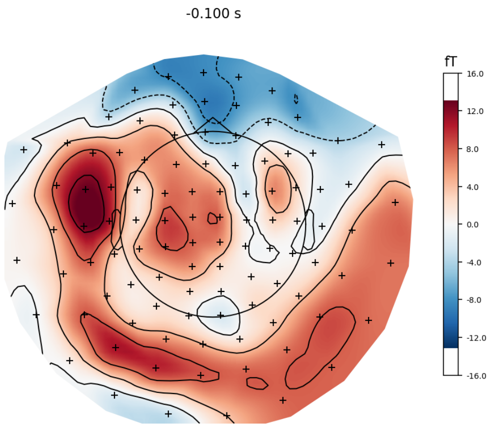


(c) Planar1

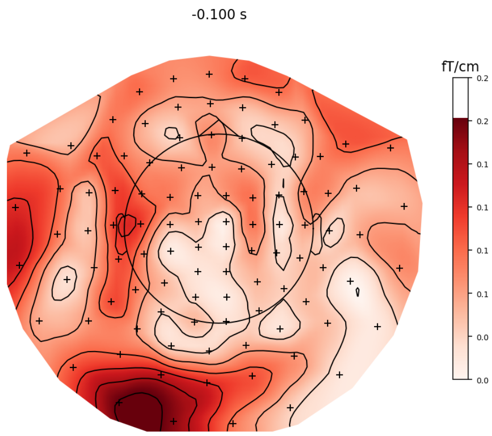


(d) Planar2

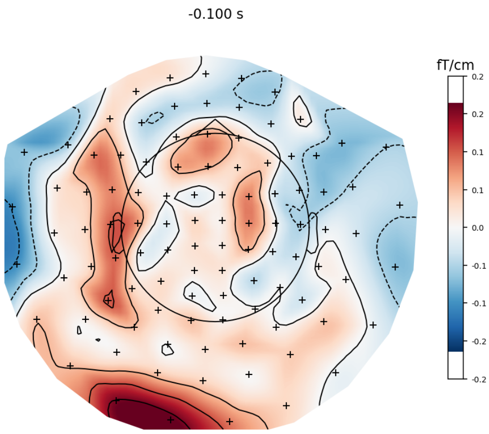
Figure A14: SCCA Canonical pair 7



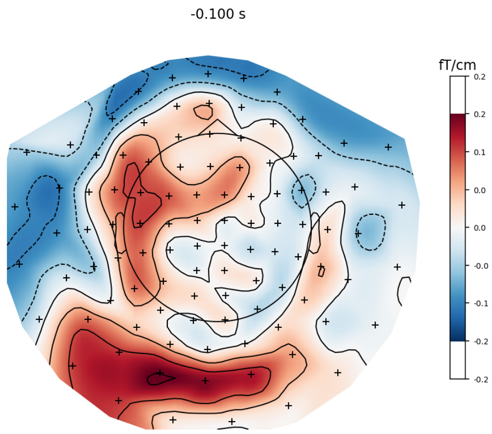
(a) Magnetometers



(b) Gradiometer RMS

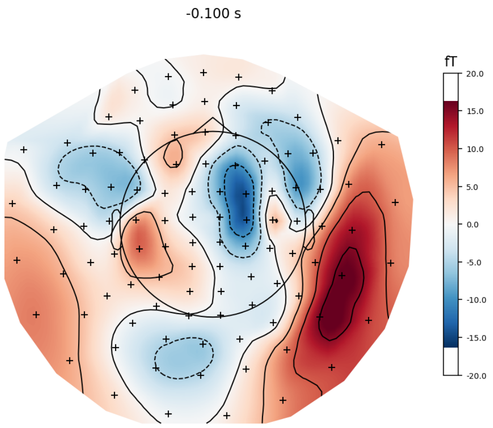


(c) Planar1

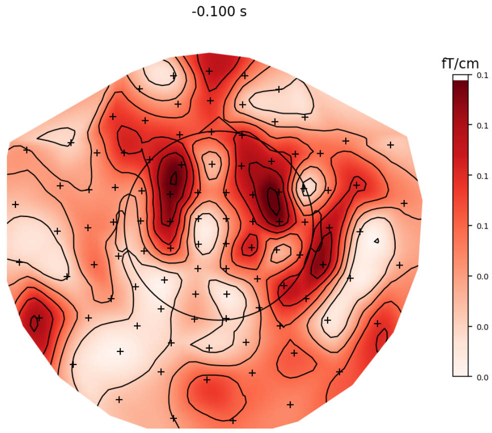


(d) Planar2

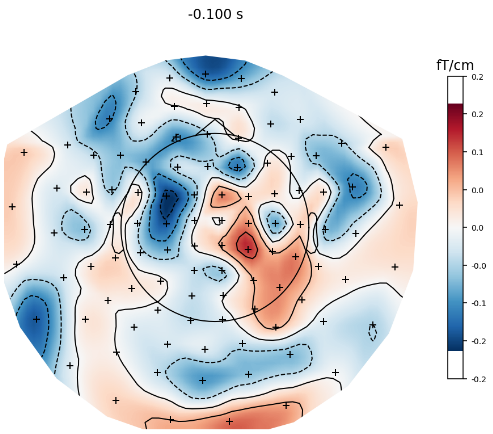
Figure A15: PCA12 Canonical pair 1



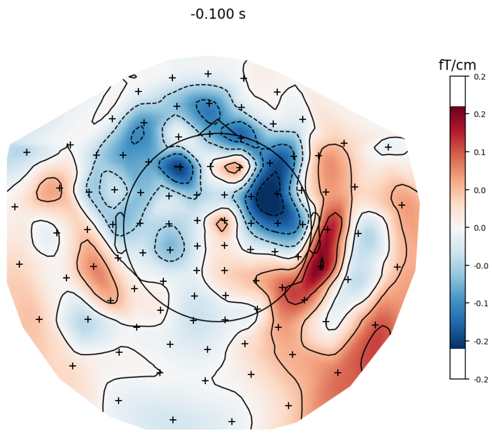
(a) Magnetometers



(b) Gradiometer RMS

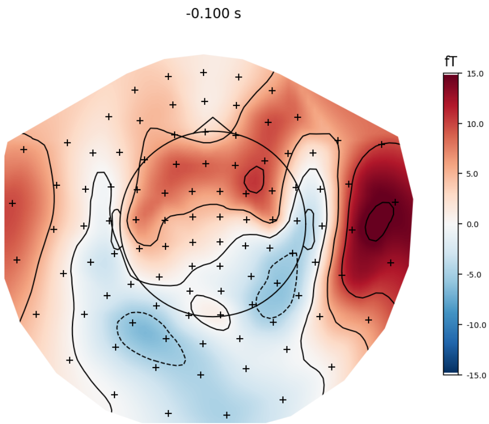


(c) Planar1

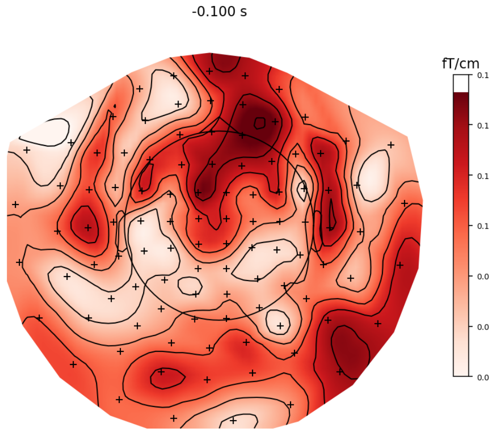


(d) Planar2

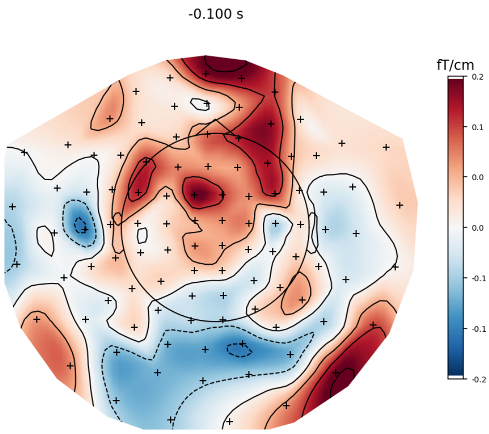
Figure A16: PCA12 Canonical pair 2



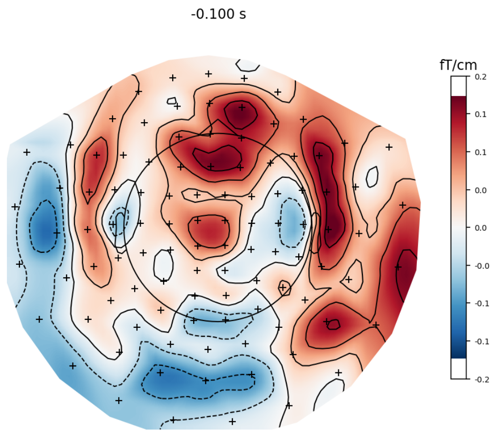
(a) Magnetometers



(b) Gradiometer RMS

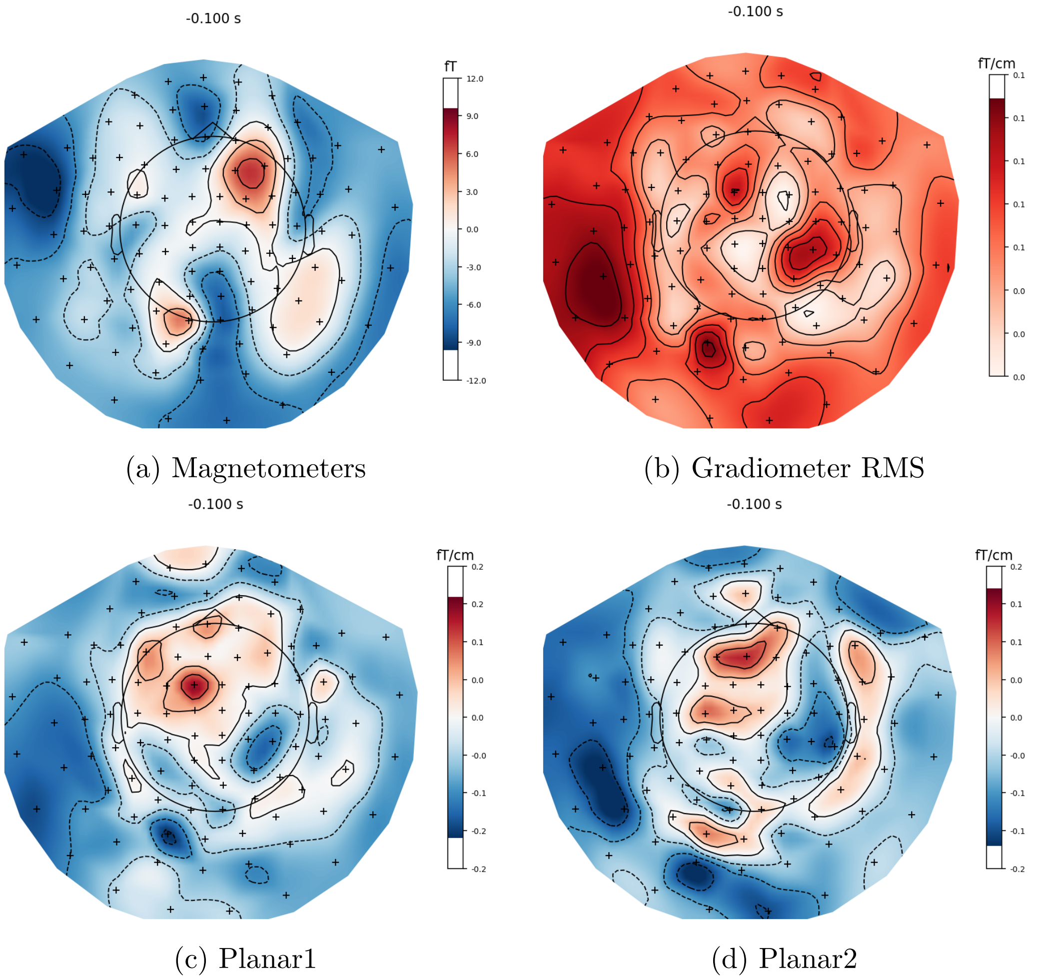


(c) Planar1



(d) Planar2

Figure A17: PCA12 Canonical pair 3



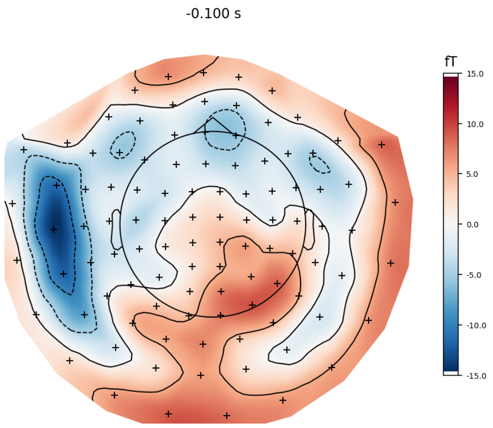
(a) Magnetometers

(b) Gradiometer RMS

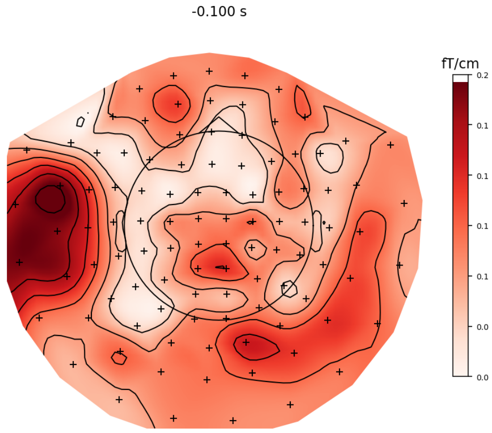
(c) Planar1

(d) Planar2

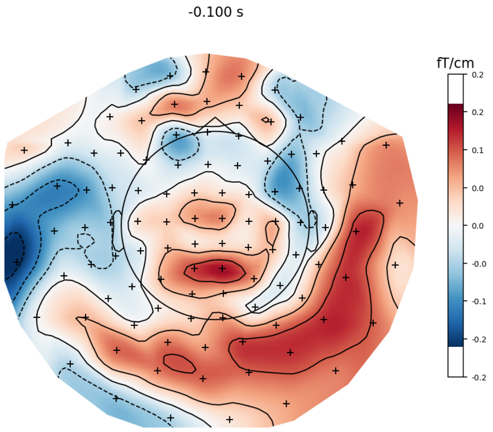
Figure A18: PCA12 Canonical pair 4



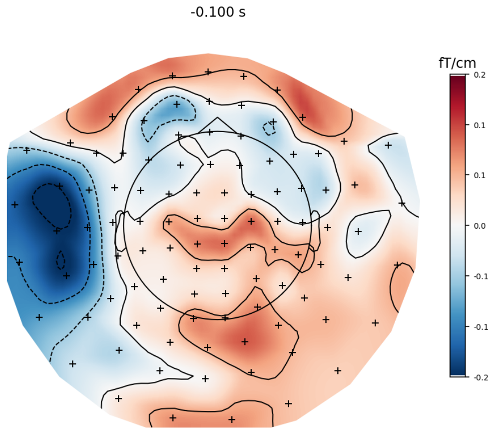
(a) Magnetometers



(b) Gradiometer RMS

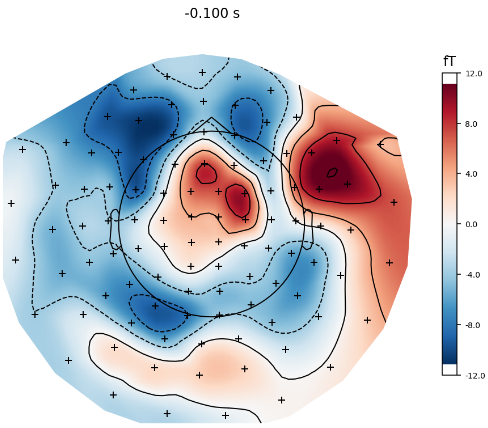


(c) Planar1

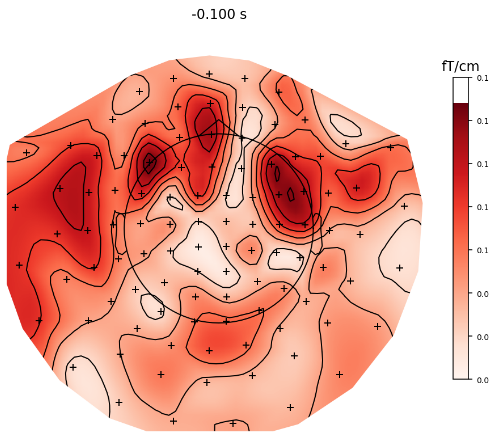


(d) Planar2

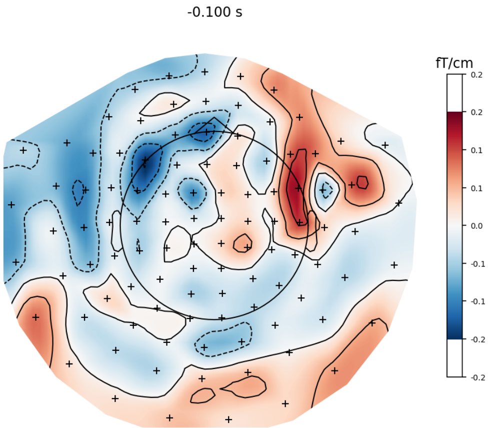
Figure A19: PCA12 Canonical pair 5



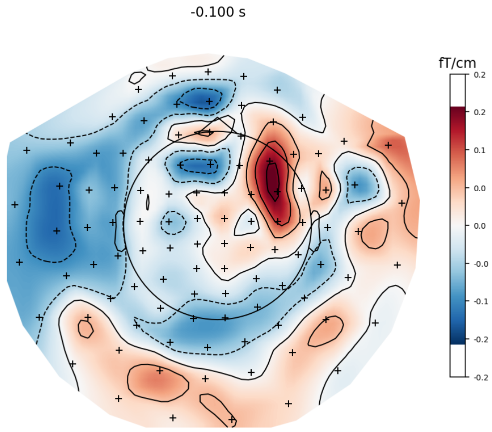
(a) Magnetometers



(b) Gradiometer RMS

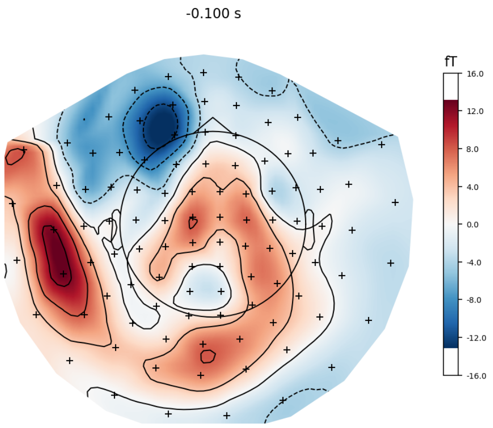


(c) Planar1

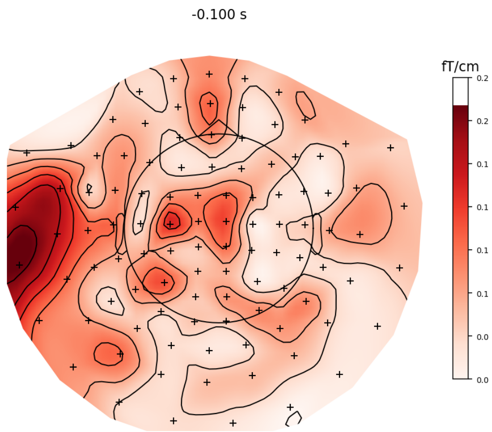


(d) Planar2

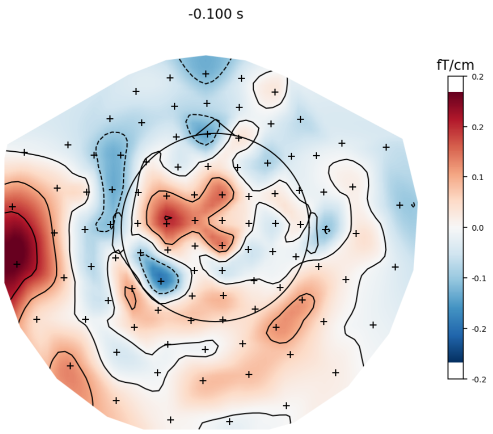
Figure A20: PCA12 Canonical pair 6



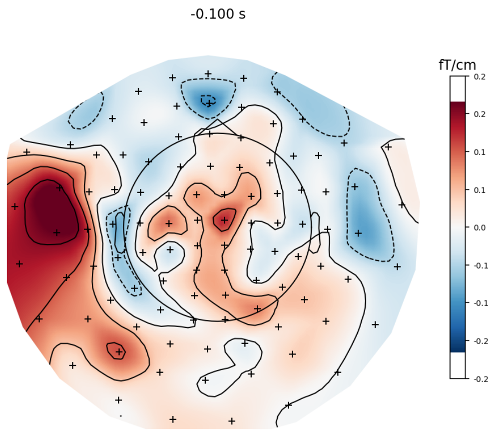
(a) Magnetometers



(b) Gradiometer RMS

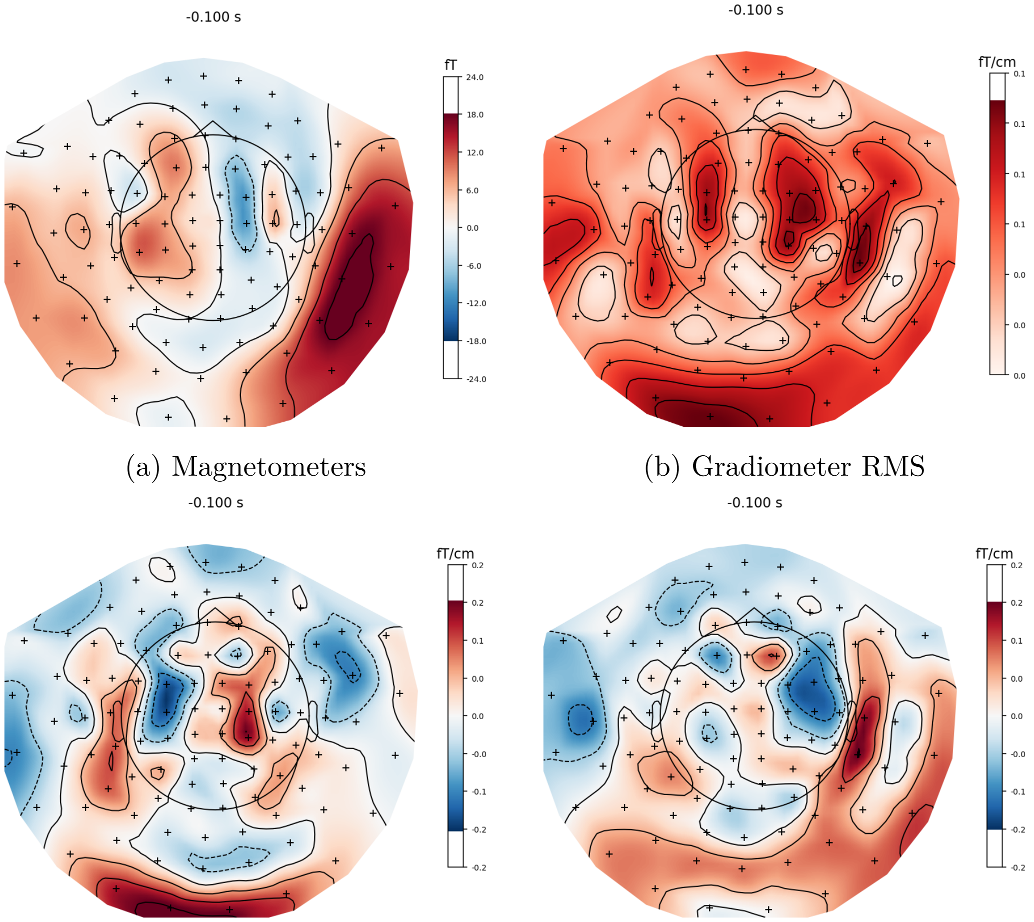


(c) Planar1



(d) Planar2

Figure A21: PCA12 Canonical pair 7



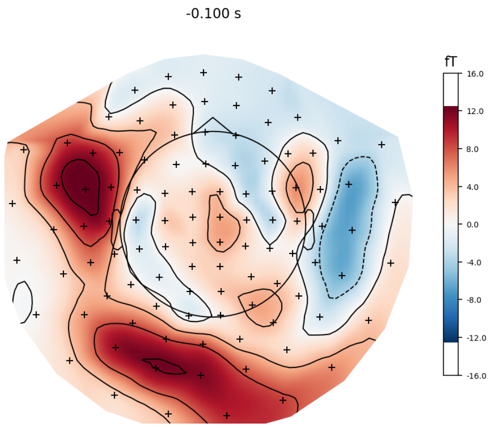
(a) Magnetometers

(b) Gradiometer RMS

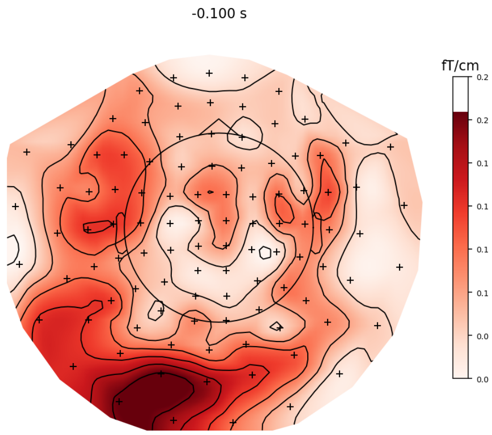
(c) Planar1

(d) Planar2

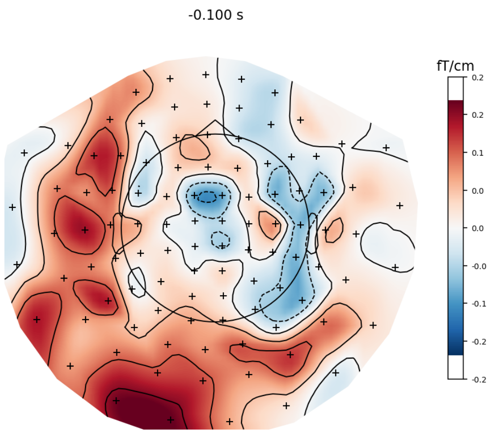
Figure A22: PCA11 Canonical pair 1



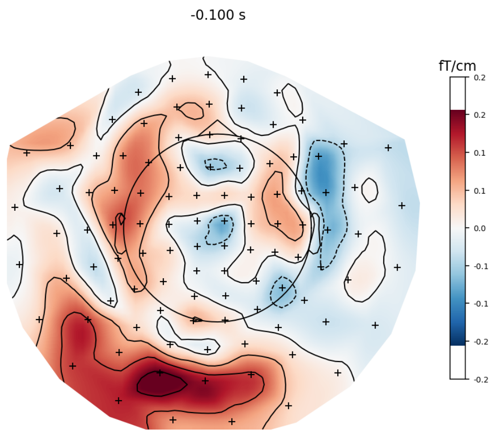
(a) Magnetometers



(b) Gradiometer RMS

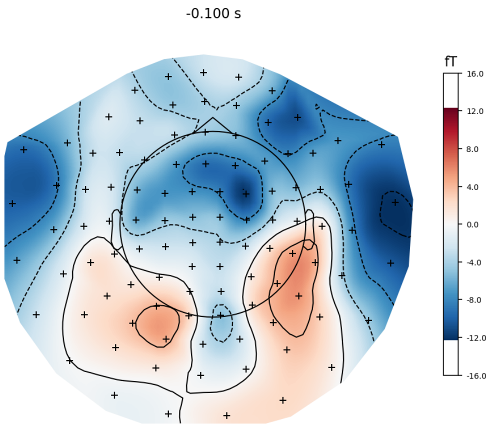


(c) Planar1

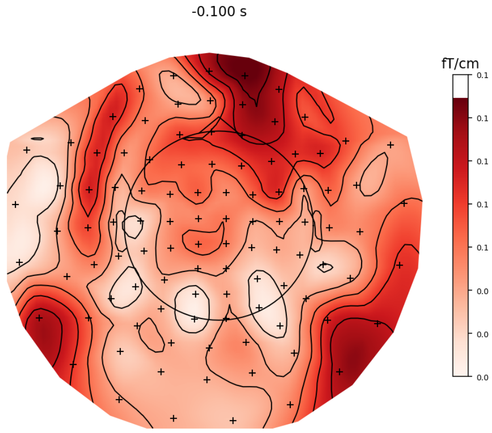


(d) Planar2

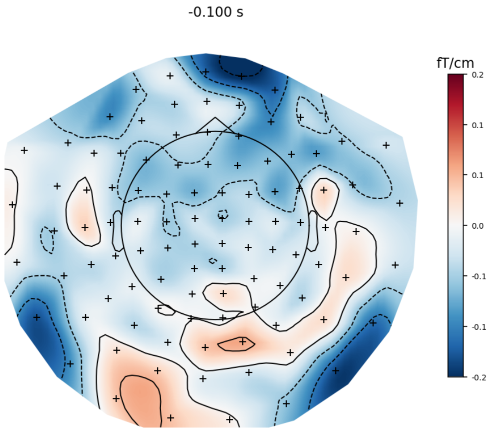
Figure A23: PCA11 Canonical pair 2



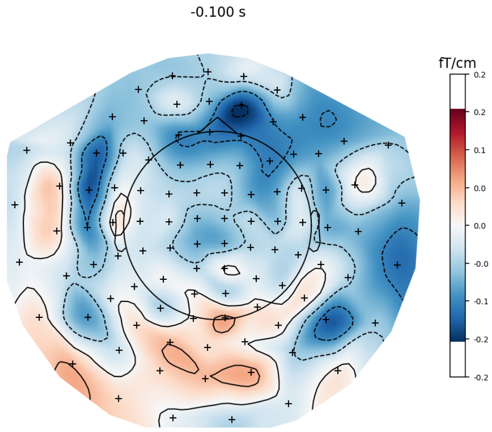
(a) Magnetometers



(b) Gradiometer RMS

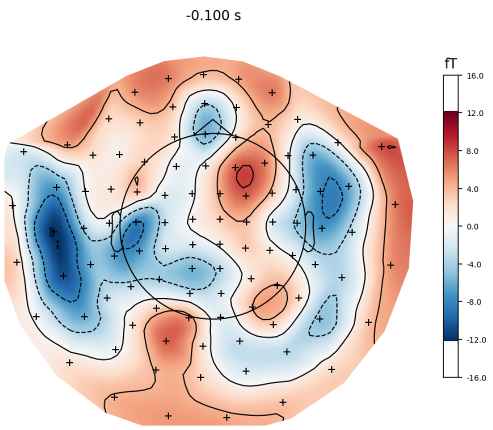


(c) Planar1

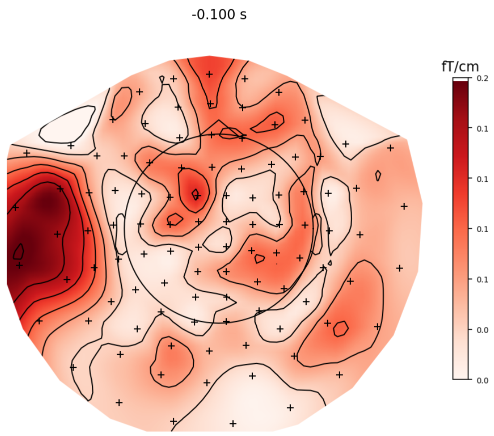


(d) Planar2

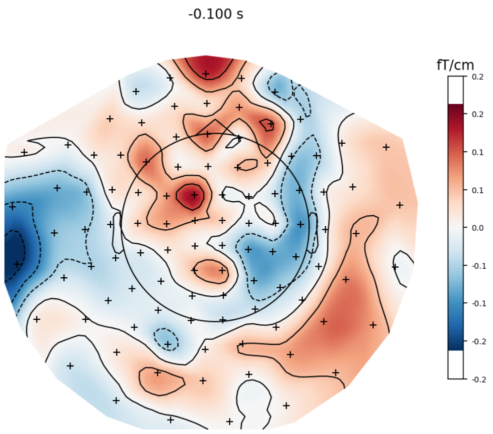
Figure A24: PCA11 Canonical pair 3



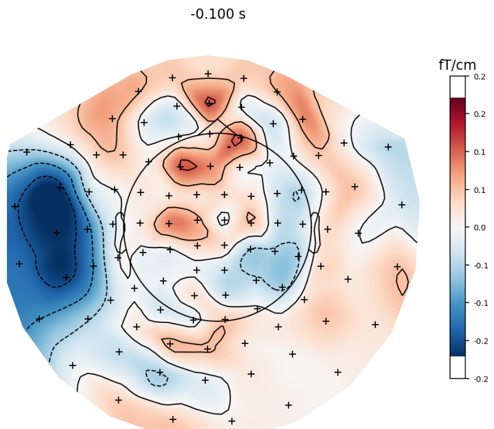
(a) Magnetometers



(b) Gradiometer RMS

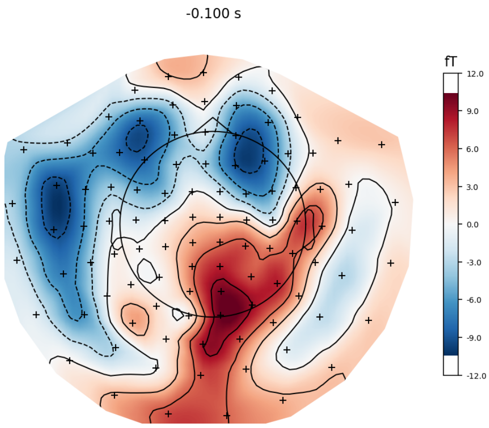


(c) Planar1

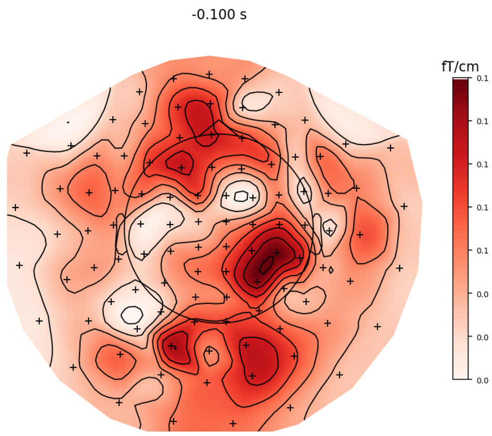


(d) Planar2

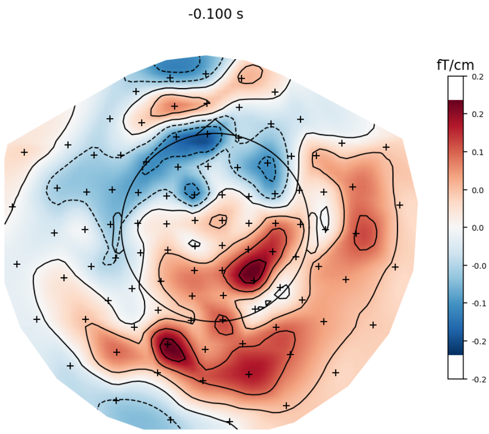
Figure A25: PCA11 Canonical pair 4



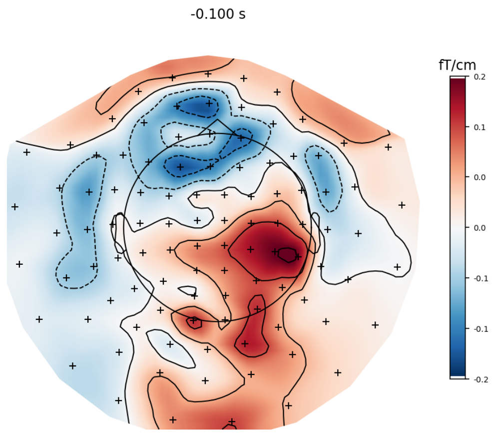
(a) Magnetometers



(b) Gradiometer RMS

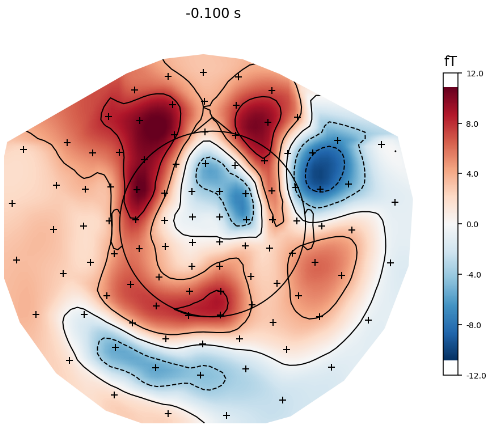


(c) Planar1

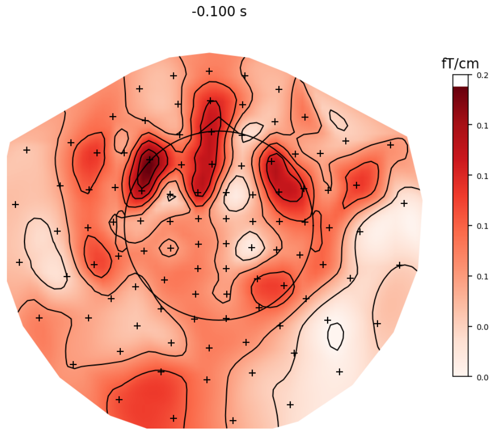


(d) Planar2

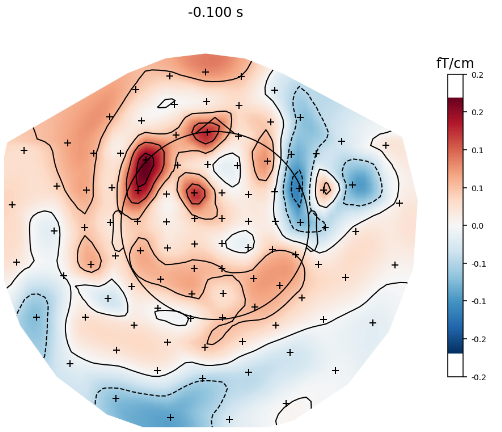
Figure A26: PCA11 Canonical pair 5



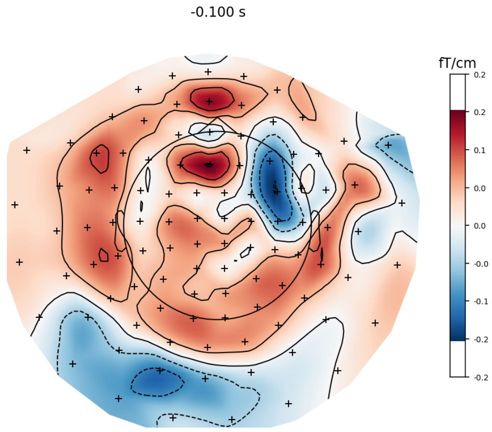
(a) Magnetometers



(b) Gradiometer RMS

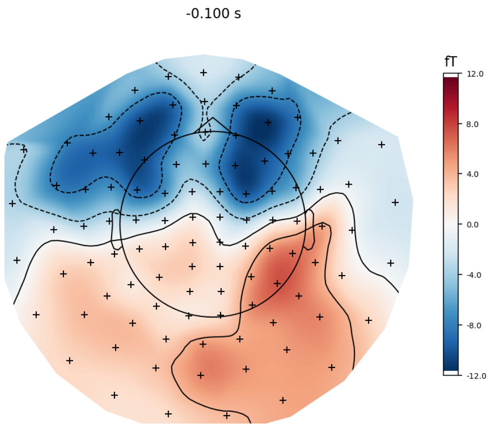


(c) Planar1

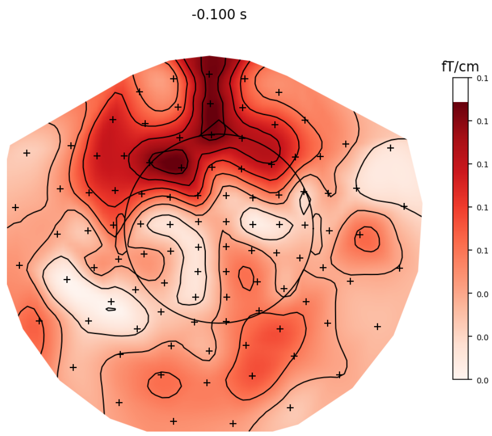


(d) Planar2

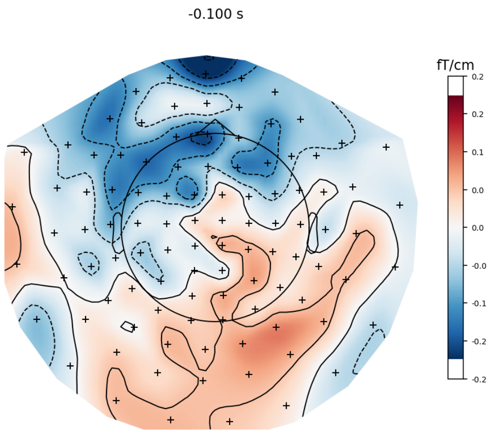
Figure A27: PCA11 Canonical pair 6



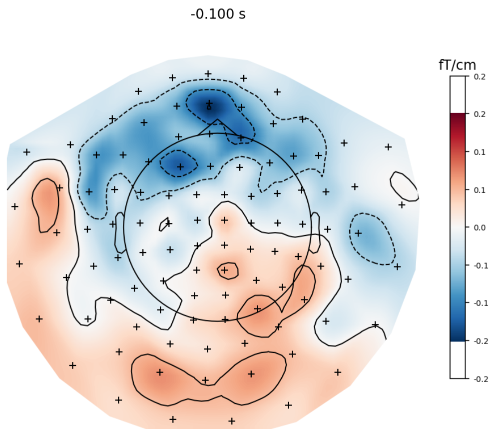
(a) Magnetometers



(b) Gradiometer RMS



(c) Planar1



(d) Planar2

Figure A28: PCA11 Canonical pair 7

Appendix B

The R code used to perform the four SCCA analyses is included in this appendix section.

```r
# Data preparation. As the data is not publicly available, reading data into R is commented out
# Reading MEG data
# alphadif <- read.csv("MEG_data.csv", header = FALSE)
# alphadif <- as.matrix(alphadif)
# Reading questionnaire data
# library(readxl)
# quest <- read_excel("Questionnaire_data.xlsx")
#quest <- quest[-c(2, 11, 14), ] # Removing NA rows
#quest <- quest[, 2:9] # Removing subject ID column
#quest <- quest[, -7] # Dropping sum of BISBAS, keeping BDI, BAI, BIS, BAS and BVS
#rquest <- as.matrix(sapply(quest[, 1:7], rank)) # Rank transformation, ties averaged for the remaining 7 variables

# Analysis 1: CCA i.e. SCCA without penalization
# Loading SCCA implementation code and required libraries
library(glmnet)
source("SCCA_functions_Koskinen.R") # Assumes SCCA_functions_Koskinen.R is in the working directory
# Sample covariances
sigma.X.hat <- cov(rquest) # Questionnaires = X
sigma.Y.hat <- cov(alphadif) # MEG data = Y
sigma.YX.hat <- cov(alphadif, rquest)
# Initial values for CCA
initsvd1 <- init0(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", npairs = 1, n = 28)
# Calculating first canonical pair, Unpenalized CCA, both lambdas = 0
CCAresult1 <- SCCA(rquest, alphadif, lambda.alpha = 0, lambda.beta = 0, alpha.init = initsvd1$alpha.init, beta.init = initsvd1$beta.init, niter = 200, standardize = TRUE)
# CAlculating the other six canonical pairs similarly
# Pair 2
initsvd2 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 2, npairs0 = 1, alpha.current = CCAresult1$alpha, beta.current = CCAresult1$beta)
```

```
CCAresult2 <- SCCA(rquest, alphadif, lambda.alpha = 0, lambda.beta = 0, alpha.current = CCAresult1$alpha, beta.current = CCAresult1$beta, alpha.init = initsvd2$alpha.init, beta.init = initsvd2$beta.init, niter = 200, npairs = 2)
# Pair 3
initsvd3 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 3, npairs0 = 2, alpha.current = CCAresult2$alpha, beta.current = CCAresult2$beta)
CCAresult3 <- SCCA(rquest, alphadif, lambda.alpha = 0, lambda.beta = 0, alpha.current = CCAresult2$alpha, beta.current = CCAresult2$beta, alpha.init = initsvd3$alpha.init, beta.init = initsvd3$beta.init, niter = 200, npairs = 3)
# Pair 4
initsvd4 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 4, npairs0 = 3, alpha.current = CCAresult3$alpha, beta.current = CCAresult3$beta)
CCAresult4 <- SCCA(rquest, alphadif, lambda.alpha = 0, lambda.beta = 0, alpha.current = CCAresult3$alpha, beta.current = CCAresult3$beta, alpha.init = initsvd4$alpha.init, beta.init = initsvd4$beta.init, niter = 200, npairs = 4)
# Pair 5
initsvd5 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 5, npairs0 = 4, alpha.current = CCAresult4$alpha, beta.current = CCAresult4$beta)
CCAresult5 <- SCCA(rquest, alphadif, lambda.alpha = 0, lambda.beta = 0, alpha.current = CCAresult4$alpha, beta.current = CCAresult4$beta, alpha.init = initsvd5$alpha.init, beta.init = initsvd5$beta.init, niter = 200, npairs = 5)
# Pair 6
initsvd6 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 6, npairs0 = 5, alpha.current = CCAresult5$alpha, beta.current = CCAresult5$beta)
CCAresult6 <- SCCA(rquest, alphadif, lambda.alpha = 0, lambda.beta = 0, alpha.current = CCAresult5$alpha, beta.current = CCAresult5$beta, alpha.init = initsvd6$alpha.init, beta.init = initsvd6$beta.init, niter = 200, npairs = 6)
# Pair 7
initsvd7 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28,
```

```
    npairs = 7, npairs0 = 6, alpha.current = CCAresult6$alpha, beta.
    current = CCAresult6$beta)
CCAresult7 <- SCCA(rquest, alphadif, lambda.alpha = 0, lambda.beta
    = 0, alpha.current = CCAresult6$alpha, beta.current = CCAresult6
    $beta, alpha.init = initsvd7$alpha.init, beta.init = initsvd7$
    beta.init, niter = 200, npairs = 7)
# Calculating canonical correlations of each pair
cc <- rep(0, 7)
for (i in 1:7) {
  cc[i] <- cor(scale(rquest) %*% CCAresult7$beta[, i], scale(
      alphadif) %*% CCAresult7$alpha[, i])
}
# Beta (questionnaire) coefficients for all 7 pairs
round(CCAresult7$beta, 3)
# Significance testing function of the first canonical pair, CCA
    version
signiCCA <- function(N, x, y, ccor) {
  counter <- 0
  for (i in 1:N) { # Creating permutations of the data
    samx <- sample(nrow(x), 28, replace = FALSE)
    X1 <- x[samx, ]
    Y1 <- y

    # Calculating first pair with permuted data, increasing counter
        if resulting correlation larger than that of original data
    CCAres <- SCCA(X1, Y1, lambda.alpha = 0, lambda.beta = 0, init.
        method = "svd", niter = 200, standardize = TRUE)
    c <- cor(scale(X1) %*% CCAres$beta, scale(Y1) %*% CCAres$alpha)

    if (c > ccor) {
      counter <- counter + 1
    }
  }
  counter / N # p-value, number of times canonical correlation was
      larger with permuted data / number of permutations
}
# Calculating the p-value
cc1 <- cor(scale(rquest) %*% CCAresult7$beta[, 1], scale(alphadif)
    %*% CCAresult7$alpha[, 1])
pval <- signiCCA(500, rquest, alphadif, cc1)

# Analysis 2: SCCA, lambda values chosen via 5-fold cross
    validation
```

```
# Two lambda grids, candidate values for tuning parameter values to be used
lambda1 <- seq(from = 0.001, to = 0.05, by = 0.002) # lambda grid for MEG-variables (Y)
lambda2 <- seq(from = 0.001, to = 0.05, by = 0.002) # lambda grid for Questionnaire variables (X)
# Calculating the first canonical pair
# Calculating initial values
initsvd1 <- init0(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 1)
set.seed(13102024) # Setting seed for reproducibility
# Using cross-validation function for the first canonical pair to choose tuning parameter values
tuningsvd1 <- cv.SCCA(x = rquest, y = alphadif, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd1$alpha.init, beta.init = initsvd1$beta.init, nfolds = 5, niter = 10)
# Using the chosen tuning parameter values to calculate the first canonical pair
SCCAresult1 <- SCCA(rquest, alphadif, lambda.alpha = tuningsvd1$bestlambda.alpha, lambda.beta = tuningsvd1$bestlambda.beta, alpha.init = initsvd1$alpha.init, beta.init = initsvd1$beta.init, niter = 200, npairs = 1)
# Perspective plot of the rho values in the 5-fold cross-validation.
persp(lambda1, lambda2, tuningsvd1$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the first canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Canonical correlation of first pair of the second analysis
cc2 <- cor(scale(rquest) %*% SCCAresult1$beta, scale(alphadif) %*% SCCAresult1$alpha)
# Significance testing funcion of first canonical pair, p-values as in the paper, SCCA version
signi <- function(N, x, y, lambda.a, lambda.b, ccor) {
  counter <- 0
  for (i in 1:N) {
    samx <- sample(nrow(x), 28, replace = FALSE)
    X1 <- x[samx, ]
    Y1 <- y

    SCCAres <- SCCA(X1, Y1, lambda.alpha = lambda.a, lambda.beta = lambda.b, init.method = "svd", niter = 200, standardize =
```

```
TRUE)
    c <- cor(scale(X1) %*% SCCAres$beta, scale(Y1) %*% SCCAres$alpha)
    if (c > ccor) {
      counter <- counter + 1
    }
  }
  counter / N
}
# Calculating the p-value
pval <- signi(500, rquest, alphadif, tuningsvd1$bestlambda.alpha, tuningsvd1$bestlambda.beta, cc2)
# Other six canonical pairs calculated like the first pair, but using modified cv-function
# Pair 2
initsvd2 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 2, npairs0 = 1, alpha.current = SCCAresult1$alpha, beta.current = SCCAresult1$beta)
set.seed(13102024)
# Using the cross validation function modified for pairs beyond the first one
tuningsvd2 <- cv.SCCA2(x = rquest, y = alphadif, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd2$alpha.init, beta.init = initsvd2$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult1$alpha, beta.cur = SCCAresult1$beta, npair = 2)
SCCAresult2 <- SCCA(rquest, alphadif, lambda.alpha = tuningsvd2$bestlambda.alpha, lambda.beta = tuningsvd2$bestlambda.beta, alpha.current = SCCAresult1$alpha, beta.current = SCCAresult1$beta, alpha.init = initsvd2$alpha.init, beta.init = initsvd2$beta.init, niter = 200, npairs = 2)
persp(lambda1, lambda2, tuningsvd2$rho, theta = 20, phi = 30, main = expression(paste("Perspective␣plot␣of␣", rho, "␣values␣for␣the␣second␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 3
initsvd3 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 3, npairs0 = 2, alpha.current = SCCAresult2$alpha, beta.current = SCCAresult2$beta)
set.seed(13102024)
tuningsvd3 <- cv.SCCA2(x = rquest, y = alphadif, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd3$alpha.init
```

```
, beta.init = initsvd3$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult2$alpha, beta.cur = SCCAresult2$beta, npair = 3)
SCCAresult3 <- SCCA(rquest, alphadif, lambda.alpha = tuningsvd3$bestlambda.alpha, lambda.beta = tuningsvd3$bestlambda.beta, alpha.current = SCCAresult2$alpha, beta.current = SCCAresult2$beta, alpha.init = initsvd3$alpha.init, beta.init = initsvd3$beta.init, niter = 200, npairs = 3)
persp(lambda1, lambda2, tuningsvd3$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the third canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 4
initsvd4 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 4, npairs0 = 3, alpha.current = SCCAresult3$alpha, beta.current = SCCAresult3$beta)
set.seed(13102024)
tuningsvd4 <- cv.SCCA2(x = rquest, y = alphadif, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd4$alpha.init, beta.init = initsvd4$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult3$alpha, beta.cur = SCCAresult3$beta, npair = 4)
SCCAresult4 <- SCCA(rquest, alphadif, lambda.alpha = tuningsvd4$bestlambda.alpha, lambda.beta = tuningsvd4$bestlambda.beta, alpha.current = SCCAresult3$alpha, beta.current = SCCAresult3$beta, alpha.init = initsvd4$alpha.init, beta.init = initsvd4$beta.init, niter = 200, npairs = 4)
persp(lambda1, lambda2, tuningsvd4$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the fourth canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 5
initsvd5 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 5, npairs0 = 4, alpha.current = SCCAresult4$alpha, beta.current = SCCAresult4$beta)
set.seed(13102024)
tuningsvd5 <- cv.SCCA2(x = rquest, y = alphadif, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd5$alpha.init, beta.init = initsvd5$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult4$alpha, beta.cur = SCCAresult4$beta, npair = 5)
SCCAresult5 <- SCCA(rquest, alphadif, lambda.alpha = tuningsvd5$bestlambda.alpha, lambda.beta = tuningsvd5$bestlambda.beta, alpha.current = SCCAresult4$alpha, beta.current = SCCAresult4$
```

```
    beta, alpha.init = initsvd5$alpha.init, beta.init = initsvd5$
    beta.init, niter = 200, npairs = 5)
persp(lambda1, lambda2, tuningsvd5$rho, theta = 20, phi = 30, main
    = expression(paste("Perspective␣plot␣of␣", rho, "␣values␣for␣the
    ␣fifth␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda.
    beta")
# Pair 6
initsvd6 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.
    X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28,
    npairs = 6, npairs0 = 5, alpha.current = SCCAresult5$alpha, beta
    .current = SCCAresult5$beta)
set.seed(13102024)
tuningsvd6 <- cv.SCCA2(x = rquest, y = alphadif, lambda.alpha =
    lambda1, lambda.beta = lambda2, alpha.init = initsvd6$alpha.init
    , beta.init = initsvd6$beta.init, nfolds = 5, niter = 10, alpha.
    cur = SCCAresult5$alpha, beta.cur = SCCAresult5$beta, npair = 6)
SCCAresult6 <- SCCA(rquest, alphadif, lambda.alpha = tuningsvd6$
    bestlambda.alpha, lambda.beta = tuningsvd6$bestlambda.beta,
    alpha.current = SCCAresult5$alpha, beta.current = SCCAresult5$
    beta, alpha.init = initsvd6$alpha.init, beta.init = initsvd6$
    beta.init, niter = 200, npairs = 6)
persp(lambda1, lambda2, tuningsvd6$rho, theta = 20, phi = 30, main
    = expression(paste("Perspective␣plot␣of␣", rho, "␣values␣for␣the
    ␣sixth␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda.
    beta")
# Pair 7
initsvd7 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.
    X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28,
    npairs = 7, npairs0 = 6, alpha.current = SCCAresult6$alpha, beta
    .current = SCCAresult6$beta)
set.seed(13102024)
tuningsvd7 <- cv.SCCA2(x = rquest, y = alphadif, lambda.alpha =
    lambda1, lambda.beta = lambda2, alpha.init = initsvd7$alpha.init
    , beta.init = initsvd7$beta.init, nfolds = 5, niter = 10, alpha.
    cur = SCCAresult6$alpha, beta.cur = SCCAresult6$beta, npair = 7)
# lambda.alpha manually lowered to avoid all coefficients shrinking
     to zero
SCCAresult7 <- SCCA(rquest, alphadif, lambda.alpha = 0.019, lambda.
    beta = tuningsvd7$bestlambda.beta, alpha.current = SCCAresult6$
    alpha, beta.current = SCCAresult6$beta, alpha.init = initsvd7$
    alpha.init, beta.init = initsvd7$beta.init, niter = 200, npairs
    = 7)
```

```
persp(lambda1, lambda2, tuningsvd7$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the seventh canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Canonical correlation of each canonical pair
cc <- rep(0, 7)
for (i in 1:7) {
  cc[i] <- cor(scale(rquest) %*% SCCAresult7$beta[, i], scale(alphadif) %*% SCCAresult7$alpha[, i])
}
# Beta (questionnaire) coefficients for all 7 pairs
round(SCCAresult7$beta, 3)

# Analysis 3:SCCA using 12 first principal components of MEG data, 95% of variance explained
# Principal component analysis, data standardized
pca_alphadif <- prcomp(alphadif, center = T, scale. = T)
summary(pca_alphadif)
# Screeplot
eigen <- pca_alphadif$sdev^2
plot(eigen / sum(eigen), type = "o", main = "Screeplot", xlab = "Principal component", ylab = "Percentage of variance explained")
# Plotting cumulative proportion of variance explained by each non-zero principal component
plot(1:28, cumsum(eigen / sum(eigen)), type = "o", xlab = "Component number", ylab = "Explained variance", main = "Cumulative proportion of variance explained")
# Scores of the first 12 PCs used for SCCA
Y <- pca_alphadif$x[, 1:12]
U <- pca_alphadif$rotation[, 1:12] # PCA rotation matrix of the first 12 PCs (loadings)
# Sample covariances
sigma.X.hat <- cov(rquest) # Questionnaires = X
sigma.Y.hat <- cov(Y) # MEG data, 12 PCs = Y
sigma.YX.hat <- cov(Y, rquest)
# Calculating first canonical pair similarly to previous analysis, 12 PCs
initsvd1 <- init0(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 1)
set.seed(13102024)
tuningsvd1 <- cv.SCCA(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd1$alpha.init, beta.
```

```
    init = initsvd1$beta.init, nfolds = 5, niter = 10)
SCCAresult1 <- SCCA(rquest, Y, lambda.alpha = tuningsvd1$bestlambda
    .alpha, lambda.beta = tuningsvd1$bestlambda.beta, alpha.init =
    initsvd1$alpha.init, beta.init = initsvd1$beta.init, niter =
    200, npairs = 1)
persp(lambda1, lambda2, tuningsvd1$rho, theta = 20, phi = 30, main
    = expression(paste("Perspective␣plot␣of␣", rho, "␣values␣for␣the
    ␣first␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda.
    beta")
# Canonical correlation of first pair of the third analysis
cc3 <- cor(scale(rquest) %*% SCCAresult1$beta, scale(Y) %*%
    SCCAresult1$alpha)
cc3
# Significance testing for the first canonical pair, SCCA version
pval <- signi(500, rquest, Y, tuningsvd1$bestlambda.alpha,
    tuningsvd1$bestlambda.beta, cc3)
pval
# Other six canonical pairs calculated like the first pair, but
    using modified cv-function
# Pair 2
initsvd2 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.
    X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28,
    npairs = 2, npairs0 = 1, alpha.current = SCCAresult1$alpha, beta
    .current = SCCAresult1$beta)
set.seed(13102024)
tuningsvd2 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1,
    lambda.beta = lambda2, alpha.init = initsvd2$alpha.init, beta.
    init = initsvd2$beta.init, nfolds = 5, niter = 10, alpha.cur =
    SCCAresult1$alpha, beta.cur = SCCAresult1$beta, npair = 2)
SCCAresult2 <- SCCA(rquest, Y, lambda.alpha = tuningsvd2$bestlambda
    .alpha, lambda.beta = tuningsvd2$bestlambda.beta, alpha.current
    = SCCAresult1$alpha, beta.current = SCCAresult1$beta, alpha.init
     = initsvd2$alpha.init, beta.init = initsvd2$beta.init, niter =
    200, npairs = 2)
persp(lambda1, lambda2, tuningsvd2$rho, theta = 20, phi = 30, main
    = expression(paste("Perspective␣plot␣of␣", rho, "␣values␣for␣the
    ␣second␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda
    .beta")
# Pair 3
initsvd3 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.
    X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28,
    npairs = 3, npairs0 = 2, alpha.current = SCCAresult2$alpha, beta
    .current = SCCAresult2$beta)
```

```r
set.seed(13102024)
tuningsvd3 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd3$alpha.init, beta.init = initsvd3$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult2$alpha, beta.cur = SCCAresult2$beta, npair = 3)
SCCAresult3 <- SCCA(rquest, Y, lambda.alpha = tuningsvd3$bestlambda.alpha, lambda.beta = tuningsvd3$bestlambda.beta, alpha.current = SCCAresult2$alpha, beta.current = SCCAresult2$beta, alpha.init = initsvd3$alpha.init, beta.init = initsvd3$beta.init, niter = 200, npairs = 3)
persp(lambda1, lambda2, tuningsvd3$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the third canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 4
initsvd4 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 4, npairs0 = 3, alpha.current = SCCAresult3$alpha, beta.current = SCCAresult3$beta)
set.seed(13102024)
tuningsvd4 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd4$alpha.init, beta.init = initsvd4$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult3$alpha, beta.cur = SCCAresult3$beta, npair = 4)
SCCAresult4 <- SCCA(rquest, Y, lambda.alpha = tuningsvd4$bestlambda.alpha, lambda.beta = tuningsvd4$bestlambda.beta, alpha.current = SCCAresult3$alpha, beta.current = SCCAresult3$beta, alpha.init = initsvd4$alpha.init, beta.init = initsvd4$beta.init, niter = 200, npairs = 4)
persp(lambda1, lambda2, tuningsvd4$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the fourth canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 5
initsvd5 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 5, npairs0 = 4, alpha.current = SCCAresult4$alpha, beta.current = SCCAresult4$beta)
set.seed(13102024)
tuningsvd5 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd5$alpha.init, beta.init = initsvd5$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult4$alpha, beta.cur = SCCAresult4$beta, npair = 5)
```

```r
SCCAresult5 <- SCCA(rquest, Y, lambda.alpha = tuningsvd5$bestlambda.alpha, lambda.beta = tuningsvd5$bestlambda.beta, alpha.current = SCCAresult4$alpha, beta.current = SCCAresult4$beta, alpha.init = initsvd5$alpha.init, beta.init = initsvd5$beta.init, niter = 200, npairs = 5)
persp(lambda1, lambda2, tuningsvd5$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the fifth canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 6
initsvd6 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 6, npairs0 = 5, alpha.current = SCCAresult5$alpha, beta.current = SCCAresult5$beta)
set.seed(13102024)
tuningsvd6 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd6$alpha.init, beta.init = initsvd6$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult5$alpha, beta.cur = SCCAresult5$beta, npair = 6)
SCCAresult6 <- SCCA(rquest, Y, lambda.alpha = tuningsvd6$bestlambda.alpha, lambda.beta = tuningsvd6$bestlambda.beta, alpha.current = SCCAresult5$alpha, beta.current = SCCAresult5$beta, alpha.init = initsvd6$alpha.init, beta.init = initsvd6$beta.init, niter = 200, npairs = 6)
persp(lambda1, lambda2, tuningsvd6$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the sixth canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 7
initsvd7 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 7, npairs0 = 6, alpha.current = SCCAresult6$alpha, beta.current = SCCAresult6$beta)
set.seed(13102024)
tuningsvd7 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd7$alpha.init, beta.init = initsvd7$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult6$alpha, beta.cur = SCCAresult6$beta, npair = 7)
SCCAresult7 <- SCCA(rquest, Y, lambda.alpha = tuningsvd7$bestlambda.alpha, lambda.beta = tuningsvd7$bestlambda.beta, alpha.current = SCCAresult6$alpha, beta.current = SCCAresult6$beta, alpha.init = initsvd7$alpha.init, beta.init = initsvd7$beta.init, niter = 200, npairs = 7)
```

```r
persp(lambda1, lambda2, tuningsvd7$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the seventh canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Canonical correlation of each canonical pair
cc <- rep(0, 7)
for (i in 1:7) {
  cc[i] <- cor(scale(rquest) %*% SCCAresult7$beta[, i], scale(Y) %*% SCCAresult7$alpha[, i])
}
# Alpha (MEG PCs) coefficients for all 7 pairs
round(SCCAresult7$alpha, 3)
# Beta (questionnaire) coefficients for all 7 pairs
round(SCCAresult7$beta, 3)
# Calculating canonical coefficients for the original 306 MEG variables via PCA rotation matrix
OrigWeights <- U %*% SCCAresult7$alpha

# Analysis 4: SCCA using 11 principal components, 3rd PC excluded, 83% of variance explained
Y <- Y[, -3] # Dropping third PCA, possible heart artifact
U <- U[, -3]
# Sample covariances
sigma.X.hat <- cov(rquest) # Questionnaires = X
sigma.Y.hat <- cov(Y) # MEG data, 11 PCs = Y
sigma.YX.hat <- cov(Y, rquest)
# First canonical pair, 11 PCs
initsvd1 <- init0(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 1)
set.seed(13102024)
tuningsvd1 <- cv.SCCA(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd1$alpha.init, beta.init = initsvd1$beta.init, nfolds = 5, niter = 10)
SCCAresult1 <- SCCA(rquest, Y, lambda.alpha = tuningsvd1$bestlambda.alpha, lambda.beta = tuningsvd1$bestlambda.beta, alpha.init = initsvd1$alpha.init, beta.init = initsvd1$beta.init, niter = 200, npairs = 1)
persp(lambda1, lambda2, tuningsvd1$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the first canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Canonical correlation of first pair, fourth analysis
```

```r
cc4 <- cor(scale(rquest) %*% SCCAresult1$beta, scale(Y) %*% SCCAresult1$alpha)
# Significance testing for the first canonical pair
pval <- signi(500, rquest, Y, tuningsvd1$bestlambda.alpha, tuningsvd1$bestlambda.beta, cc4)
# Other six canonical pairs calculated like the first pair, but using modified cv-function
# Pair 2
initsvd2 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 2, npairs0 = 1, alpha.current = SCCAresult1$alpha, beta.current = SCCAresult1$beta)
set.seed(13102024)
tuningsvd2 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd2$alpha.init, beta.init = initsvd2$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult1$alpha, beta.cur = SCCAresult1$beta, npair = 2)
# lambda.alpha lowered to avoid zero coefficient vector
SCCAresult2 <- SCCA(rquest, Y, lambda.alpha = 0.020, lambda.beta = tuningsvd2$bestlambda.beta, alpha.current = SCCAresult1$alpha, beta.current = SCCAresult1$beta, alpha.init = initsvd2$alpha.init, beta.init = initsvd2$beta.init, niter = 200, npairs = 2)
persp(lambda1, lambda2, tuningsvd2$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the second canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 3
initsvd3 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 3, npairs0 = 2, alpha.current = SCCAresult2$alpha, beta.current = SCCAresult2$beta)
set.seed(13102024)
tuningsvd3 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd3$alpha.init, beta.init = initsvd3$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult2$alpha, beta.cur = SCCAresult2$beta, npair = 3)
SCCAresult3 <- SCCA(rquest, Y, lambda.alpha = tuningsvd3$bestlambda.alpha, lambda.beta = tuningsvd3$bestlambda.beta, alpha.current = SCCAresult2$alpha, beta.current = SCCAresult2$beta, alpha.init = initsvd3$alpha.init, beta.init = initsvd3$beta.init, niter = 200, npairs = 3)
persp(lambda1, lambda2, tuningsvd3$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the
```

```
␣third␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 4
initsvd4 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 4, npairs0 = 3, alpha.current = SCCAresult3$alpha, beta.current = SCCAresult3$beta)
set.seed(13102024)
tuningsvd4 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd4$alpha.init, beta.init = initsvd4$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult3$alpha, beta.cur = SCCAresult3$beta, npair = 4)
SCCAresult4 <- SCCA(rquest, Y, lambda.alpha = tuningsvd4$bestlambda.alpha, lambda.beta = tuningsvd4$bestlambda.beta, alpha.current = SCCAresult3$alpha, beta.current = SCCAresult3$beta, alpha.init = initsvd4$alpha.init, beta.init = initsvd4$beta.init, niter = 200, npairs = 4)
persp(lambda1, lambda2, tuningsvd4$rho, theta = 20, phi = 30, main = expression(paste("Perspective␣plot␣of␣", rho, "␣values␣for␣the␣fourth␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 5
initsvd5 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 5, npairs0 = 4, alpha.current = SCCAresult4$alpha, beta.current = SCCAresult4$beta)
set.seed(13102024)
tuningsvd5 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd5$alpha.init, beta.init = initsvd5$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult4$alpha, beta.cur = SCCAresult4$beta, npair = 5)
SCCAresult5 <- SCCA(rquest, Y, lambda.alpha = tuningsvd5$bestlambda.alpha, lambda.beta = tuningsvd5$bestlambda.beta, alpha.current = SCCAresult4$alpha, beta.current = SCCAresult4$beta, alpha.init = initsvd5$alpha.init, beta.init = initsvd5$beta.init, niter = 200, npairs = 5)
persp(lambda1, lambda2, tuningsvd5$rho, theta = 20, phi = 30, main = expression(paste("Perspective␣plot␣of␣", rho, "␣values␣for␣the␣fifth␣canonical␣pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 6
initsvd6 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28,
```

```
npairs = 6, npairs0 = 5, alpha.current = SCCAresult5$alpha, beta.current = SCCAresult5$beta)
set.seed(13102024)
tuningsvd6 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd6$alpha.init, beta.init = initsvd6$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult5$alpha, beta.cur = SCCAresult5$beta, npair = 6)
SCCAresult6 <- SCCA(rquest, Y, lambda.alpha = tuningsvd6$bestlambda.alpha, lambda.beta = tuningsvd6$bestlambda.beta, alpha.current = SCCAresult5$alpha, beta.current = SCCAresult5$beta, alpha.init = initsvd6$alpha.init, beta.init = initsvd6$beta.init, niter = 200, npairs = 6)
persp(lambda1, lambda2, tuningsvd6$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the sixth canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Pair 7
initsvd7 <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = "svd", n = 28, npairs = 7, npairs0 = 6, alpha.current = SCCAresult6$alpha, beta.current = SCCAresult6$beta)
set.seed(13102024)
tuningsvd7 <- cv.SCCA2(x = rquest, y = Y, lambda.alpha = lambda1, lambda.beta = lambda2, alpha.init = initsvd7$alpha.init, beta.init = initsvd7$beta.init, nfolds = 5, niter = 10, alpha.cur = SCCAresult6$alpha, beta.cur = SCCAresult6$beta, npair = 7)
SCCAresult7 <- SCCA(rquest, Y, lambda.alpha = tuningsvd7$bestlambda.alpha, lambda.beta = tuningsvd7$bestlambda.beta, alpha.current = SCCAresult6$alpha, beta.current = SCCAresult6$beta, alpha.init = initsvd7$alpha.init, beta.init = initsvd7$beta.init, niter = 200, npairs = 7)
persp(lambda1, lambda2, tuningsvd7$rho, theta = 20, phi = 30, main = expression(paste("Perspective plot of ", rho, " values for the seventh canonical pair")), xlab = "lambda.alpha", ylab = "lambda.beta")
# Canonical correlations of each canonical pair
cc <- rep(0, 7)
for (i in 1:7) {
  cc[i] <- cor(scale(rquest) %*% SCCAresult7$beta[, i], scale(Y) %*% SCCAresult7$alpha[, i])
}
# Alpha (MEG PCs) coefficients for all 7 pairs
round(SCCAresult7$alpha, 3)
```

```r
# Beta (questionnaire) coefficients for all 7 pairs
round(SCCAresult7$beta, 3)
# Calculating canonical coefficients for the original 306 MEG
    variables via PCA rotation matrix
OrigWeights <- U %*% SCCAresult7$alpha
```

Appendix C

The modified version of the R code from the supplementary material of Mai and Zhang (2019), used in this thesis for the four SCCA analyses, is included in this appendix section.

```r
library(glmnet)
#### SCCA is the main functions that produces estimates of the
    canonical pairs.
SCCA <- function(x, y, alpha.init = NULL, beta.init = NULL, lambda.
    alpha, lambda.beta, niter = 100, npairs = 1, init.method = c("
    sparse", "uniform", "svd", "random"), alpha.current = NULL, beta
    .current = NULL, standardize = TRUE, eps = 1e-4) {
  p <- ncol(x)
  q <- ncol(y)
  n <- nrow(x)

  x <- scale(x, center = T, scale = standardize)
  y <- scale(y, center = T, scale = standardize)

  sigma.YX.hat <- cov(y, x)
  sigma.X.hat <- cov(x)
  sigma.Y.hat <- cov(y)

  alpha <- matrix(0, q, npairs)
  beta <- matrix(0, p, npairs)
  rho <- matrix(0, npairs, npairs)

  if (length(init.method) > 1) {
    init.method <- init.method[1]
  }

  if (missing(alpha.current)) {
    npairs0 <- 0
  } else {
    npairs0 <- ncol(alpha.current)
```

```

    alpha[, 1:npairs0] <- alpha.current
    beta[, 1:npairs0] <- beta.current
  }

  if (missing(alpha.init)) {
    if (missing(alpha.current)) {
      obj.init <- init0(sigma.YX.hat, sigma.X.hat, sigma.Y.hat, init.method = init.method, npairs, n = n)

      alpha.init <- obj.init$alpha.init
      beta.init <- obj.init$beta.init
    } else {
      alpha.current <- as.matrix(alpha.current)
      beta.current <- as.matrix(beta.current)

      obj.init <- init1(sigma.YX.hat = sigma.YX.hat, sigma.X.hat = sigma.X.hat, sigma.Y.hat = sigma.Y.hat, init.method = init.method, npairs = npairs, npairs0 = npairs0, alpha.current = alpha.current, beta.current = beta.current, n = n, eps = eps)
      alpha.init <- obj.init$alpha.init
      beta.init <- obj.init$beta.init

      alpha[, 1:npairs0] <- alpha.current
      beta[, 1:npairs0] <- beta.current
    }
  }

  n.iter.converge <- rep(0, npairs - npairs0)

  for (ipairs in (npairs0 + 1):npairs) {
    alpha.init <- as.matrix(alpha.init)
    beta.init <- as.matrix(beta.init)

    omega <- find.Omega(sigma.YX.hat, ipairs, alpha = alpha[, 1:(ipairs - 1)], beta = beta[, 1:(ipairs - 1)], y = y, x = x)

    x.tmp <- omega %*% x
    y.tmp <- t(omega) %*% y

    lambda.alpha0 <- lambda.alpha[ipairs - npairs0]
    lambda.beta0 <- lambda.beta[ipairs - npairs0]
```

```

    alpha0 <- alpha.init
    beta0 <- beta.init

    obj <- SCCA.solution(x = x, y = y, x.Omega = x.tmp, y.Omega = y.tmp, alpha0, beta0, lambda.alpha0, lambda.beta0, niter = niter, eps = eps)

    alpha[, ipairs] <- obj$alpha
    beta[, ipairs] <- obj$beta
    n.iter.converge[ipairs - npairs0] <- obj$niter

    if ((ipairs < npairs) & (init.method == "sparse")) {
      obj.init <- init1(sigma.YX.hat, sigma.X.hat, sigma.Y.hat, init.method = init.method, npairs, npairs0 = ipairs, alpha.current = alpha[, 1:ipairs], beta.current = beta[, 1:ipairs])
      alpha.init <- obj.init$alpha.init
      beta.init <- obj.init$beta.init
    }
  }

  list(alpha = alpha, beta = beta, alpha.init = alpha.init, beta.init = beta.init, n.iter.converge = n.iter.converge)
}

#### The function init0 finds the initial value when no canonical pairs have been obtained. If init.method="sparse",
#### only one pair of initial value will be returned. For other options of init.method, the number of pairs of initial
#### values can be specified with the argument npairs.
init0 <- function(sigma.YX.hat, sigma.X.hat, sigma.Y.hat, init.method, npairs, n, d = NULL) {
  p <- ncol(sigma.X.hat)
  q <- ncol(sigma.Y.hat)

  if (init.method == "svd") {
    obj.svd <- svd(sigma.YX.hat, nu = npairs, nv = npairs)
    alpha.init <- obj.svd$u[, 1:npairs, drop = F]
    beta.init <- obj.svd$v[, 1:npairs, drop = F]
  }
  if (init.method == "uniform") {
    alpha.init <- matrix(1, q, npairs)
```

```
    beta.init <- matrix(1, p, npairs)
  }
  if (init.method == "random") {
    alpha.init <- matrix(rnorm(q * npairs), q, npairs)
    beta.init <- matrix(rnorm(p * npairs), p, npairs)
  }
  if (init.method == "sparse") {
    alpha.init <- matrix(0, q, npairs)
    beta.init <- matrix(0, p, npairs)
    if (missing(d)) d <- sqrt(n)
    thresh <- sort(abs(sigma.YX.hat), decreasing = T)[d]
    row.max <- apply(abs(sigma.YX.hat), 1, max)
    col.max <- apply(abs(sigma.YX.hat), 2, max)
    obj.svd <- svd(sigma.YX.hat[row.max > thresh, col.max > thresh])

    alpha1.init <- rep(0, q)
    beta1.init <- rep(0, p)
    alpha1.init[row.max > thresh] <- obj.svd$u[, 1]
    beta1.init[col.max > thresh] <- obj.svd$v[, 1]

    alpha.init[, 1] <- alpha1.init
    beta.init[, 1] <- beta1.init
  }
  alpha.scale <- diag(t(alpha.init) %*% sigma.Y.hat %*% alpha.init)[1:npairs, drop = F]
  alpha.init <- sweep(alpha.init[, 1:npairs, drop = F], 2, sqrt(alpha.scale), "/")
  beta.scale <- diag(t(beta.init) %*% sigma.X.hat %*% beta.init)[1:npairs, drop = F]
  beta.init <- sweep(beta.init[, 1:npairs, drop = F], 2, sqrt(beta.scale), "/")
  list(alpha.init = alpha.init, beta.init = beta.init)
}

#### The function init1 finds the initial value when npairs0 canonical pairs have been obtained. If init.method="sparse",
#### only one pair of initial value will be returned. For other options of init.method, init1 returns npairs - npairs0 pairs
#### of initial values.
init1 <- function(sigma.YX.hat, sigma.X.hat, sigma.Y.hat, init.method, npairs, npairs0, alpha.current, beta.current, n = n, eps = 1e-4, d = NULL) {
  p <- ncol(sigma.X.hat)
```

```
q <- ncol(sigma.Y.hat)

alpha.init <- matrix(0, q, 1)
beta.init <- matrix(0, p, 1)
alpha.current <- as.matrix(alpha.current)
beta.current <- as.matrix(beta.current)

npairs0 <- ncol(alpha.current)

if (init.method == "svd") {
  obj.svd <- svd(sigma.YX.hat)
  alpha.init <- obj.svd$u[, npairs0 + 1, drop = F]
  beta.init <- obj.svd$v[, npairs0 + 1, drop = F]
}
if (init.method == "uniform") {
  alpha.init <- matrix(1, q, 1)
  beta.init <- matrix(1, p, 1)
}
if (init.method == "random") {
  alpha.init <- matrix(rnorm(q * npairs), q, 1)
  beta.init <- matrix(rnorm(p * npairs), p, 1)
}
if (init.method == "sparse") {
  id.nz.alpha <- which(apply(abs(alpha.current), 1, sum) > eps)
  id.nz.beta <- which(apply(abs(beta.current), 1, sum) > eps)

  rho.tmp <- t(alpha.current) %*% sigma.YX.hat %*% beta.current

  sigma.YX.tmp <- sigma.YX.hat - sigma.Y.hat %*% alpha.current %*%
      rho.tmp %*% t(beta.current) %*% sigma.X.hat

  if (missing(d)) d <- sqrt(n)

  thresh <- sort(abs(sigma.YX.tmp), decreasing = T)[d]
  row.max <- apply(abs(sigma.YX.tmp), 1, max)
  col.max <- apply(abs(sigma.YX.tmp), 2, max)

  id.row <- unique(c(id.nz.alpha, which(row.max > thresh)))
  id.row <- sort(id.row, decreasing = FALSE)
  id.col <- unique(c(id.nz.beta, which(col.max > thresh)))
  id.col <- sort(id.col, decreasing = FALSE)

  sigma.tmp <- sigma.YX.tmp[id.row, id.col]
```

```
    obj.svd <- svd(sigma.tmp)

    alpha.init[id.row] <- obj.svd$u[, 1]
    beta.init[id.col] <- obj.svd$v[, 1]
  }
  alpha.scale <- drop(t(alpha.init) %*% sigma.Y.hat %*% alpha.init)
  alpha.init <- alpha.init / sqrt(alpha.scale)
  beta.scale <- drop(t(beta.init) %*% sigma.X.hat %*% beta.init)
  beta.init <- beta.init / sqrt(beta.scale)

  list(alpha.init = alpha.init, beta.init = beta.init)
}

#### The function find.Omega is used by SCCA.solution.
find.Omega <- function(sigma.YX.hat, npairs, alpha = NULL, beta =
    NULL, y = NULL, x = NULL) {
  n <- nrow(y)
  if (npairs > 1) {
    rho <- t(alpha) %*% sigma.YX.hat %*% beta
    omega <- diag(rep(1, n)) - y %*% alpha %*% rho %*% t(beta) %*% t
        (x) / n
  } else {
    omega <- diag(rep(1, n))
  }
  omega
}

#### The function SCCA.solution is used by SCCA.
SCCA.solution <- function(x, y, x.Omega, y.Omega, alpha0, beta0,
    lambda.alpha, lambda.beta, niter = 100, glmnet.alg = NULL, eps =
     1e-4) {
  n <- nrow(x)
  p <- ncol(x)
  q <- ncol(y)

  for (i in 1:niter) {
    x0 <- x.Omega %*% beta0

    m <- glmnet(y, x0, standardize = FALSE, intercept = FALSE,
        lambda = lambda.alpha)

    alpha1 <- coef(m, s = lambda.alpha)[-1]

```

```
    if (sum(abs(alpha1)) < eps) {
      alpha0 <- rep(0, q)
      break
    }
    id.nz <- which(alpha1 != 0)
    alpha1.scale <- y[, id.nz, drop = F] %*% alpha1[id.nz, drop = F]

    alpha1 <- alpha1 / drop(sqrt(t(alpha1.scale) %*% alpha1.scale / (n - 1)))

    y0 <- y.Omega %*% alpha1

    m <- glmnet(x, y0, standardize = FALSE, intercept = FALSE, lambda = lambda.beta)

    beta1 <- coef(m, s = lambda.beta)[-1]

    if (sum(abs(beta1)) < eps) {
      beta0 <- rep(0, p)
      break
    }
    id.nz <- which(beta1 != 0)
    beta1.scale <- x[, id.nz, drop = F] %*% beta1[id.nz, drop = F]

    beta1 <- beta1 / drop(sqrt(t(beta1.scale) %*% beta1.scale / (n - 1)))

    if (sum(abs(alpha1 - alpha0)) < eps & sum(abs(beta1 - beta0) < eps)) break
    alpha0 <- alpha1
    beta0 <- beta1
  }

  list(alpha = alpha0, beta = beta0, niter = i)
}

# The function cv.SCCA is the cross validation function for the first canonical pair.
cv.SCCA <- function(x, y, lambda.alpha, lambda.beta, nfolds = 5, alpha.init, beta.init, eps = 1e-3, niter = 10, standardize = TRUE) {
  n <- nrow(x)
  id.folds <- cut(seq(1:n), breaks = nfolds, labels = 1:nfolds)
```

```
  id.folds <- sample(id.folds, n, replace = FALSE)
  id.folds <- as.numeric(id.folds)
  rho <- matrix(0, length(lambda.alpha), length(lambda.beta))

  for (i.lambda in 1:length(lambda.alpha)) {
    for (j.lambda in 1:length(lambda.beta)) {
      for (i in 1:nfolds) {
        obj <- SCCA(x[id.folds != i, ], y[id.folds != i, ], alpha.init = alpha.init, beta.init = beta.init, lambda.alpha = lambda.alpha[i.lambda], lambda.beta = lambda.beta[j.lambda], eps = eps, niter = niter, standardize = standardize)

        rho[i.lambda, j.lambda] <- rho[i.lambda, j.lambda] + abs(cor(x[id.folds == i, ] %*% obj$beta, y[id.folds == i, ] %*% obj$alpha))
      }
    }
  }
  rho[is.na(rho)] <- 0
  id.alpha.max <- max(which(apply(rho, 1, max) == max(rho)))
  id.beta.max <- max(which(apply(rho, 2, max) == max(rho)))
  rho[is.na(rho)] <- 0

  list(rho = rho, bestlambda.alpha = lambda.alpha[id.alpha.max], bestlambda.beta = lambda.beta[id.beta.max])
}

# Creating a cross validation function for canonical pairs beyond the first based on the original function.
cv.SCCA2 <- function(x, y, lambda.alpha, lambda.beta, nfolds = 5, alpha.init, beta.init, eps = 1e-3, niter = 10, standardize = TRUE, alpha.cur, beta.cur, npair) {
  n <- nrow(x)
  id.folds <- cut(seq(1:n), breaks = nfolds, labels = 1:nfolds)
  id.folds <- sample(id.folds, n, replace = FALSE)
  id.folds <- as.numeric(id.folds)
  rho <- matrix(0, length(lambda.alpha), length(lambda.beta))

  for (i.lambda in 1:length(lambda.alpha)) {
    for (j.lambda in 1:length(lambda.beta)) {
      for (i in 1:nfolds) {
```

```r
        obj <- SCCA(x[id.folds != i, ], y[id.folds != i, ], alpha.init = alpha.init, beta.init = beta.init, lambda.alpha = lambda.alpha[i.lambda], lambda.beta = lambda.beta[j.lambda], alpha.current = alpha.cur, beta.current = beta.cur, eps = eps, niter = niter, standardize = standardize, npairs = npair)

        rho[i.lambda, j.lambda] <- rho[i.lambda, j.lambda] + abs(cor(x[id.folds == i, ] %*% obj$beta[, npair], y[id.folds == i, ] %*% obj$alpha[, npair]))
      }
    }
  }
  rho[is.na(rho)] <- 0
  id.alpha.max <- max(which(apply(rho, 1, max) == max(rho)))
  id.beta.max <- max(which(apply(rho, 2, max) == max(rho)))
  rho[is.na(rho)] <- 0

  list(rho = rho, bestlambda.alpha = lambda.alpha[id.alpha.max], bestlambda.beta = lambda.beta[id.beta.max])
}

# The function cv.SCCA.equal is the cross validation function for the first canonical pair, assuming that alpha and beta uses the same tuning parameter.
cv.SCCA.equal <- function(x, y, lambda, nfolds = 5, alpha.init, beta.init, eps = 1e-3, niter = 20) {
  n <- nrow(x)
  id.folds <- cut(seq(1:n), breaks = nfolds, labels = 1:nfolds)
  id.folds <- sample(id.folds, n, replace = FALSE)
  id.folds <- as.numeric(id.folds)
  rho <- matrix(0, length(lambda), nfolds)
  for (i.lambda in 1:length(lambda)) {
    for (i in 1:nfolds) {
      obj <- SCCA(x[id.folds != i, ], y[id.folds != i, ], alpha.init = alpha.init, beta.init = beta.init, lambda.alpha = lambda[i.lambda], lambda.beta = lambda[i.lambda], eps = eps, niter = niter)

      rho[i.lambda, i] <- abs(cor(x[id.folds == i, ] %*% obj$beta, y[id.folds == i, ] %*% obj$alpha))
    }
  }
```

```
  rho <- apply(rho, 1, mean)
  list(rho = rho, lambda = lambda, bestlambda = lambda[which.max(rho
     )])
}
```