# Towards Scalable, Flexible, and Interpretable Self-Supervised Learning for Multiview Biomedical Data

by

**James Chapman**

**January 2022**

**PhD Thesis**

**i4health CDT**

**University College London**

# Declaration

I, James Chapman, confirm that the work presented in my thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

# Abstract

Biomedical data are essential for advancing our knowledge and practice of medicine and healthcare. However, biomedical data are also challenging to analyze due to their complexity, heterogeneity, high-dimensionality, and scarcity of labels. To overcome these challenges, self-supervised learning (SSL) has emerged as a promising paradigm for learning from unlabeled data by leveraging inherent structures or patterns in the data. SSL methods can exploit different forms of supervision signals derived from the data itself, such as contrastive learning, reconstruction, prediction, or clustering. SSL methods can also benefit from deep neural networks that can learn expressive and flexible representations from complex and high-dimensional data.

In this thesis, we focus on a specific type of SSL problem, namely multiview SSL, where data are represented by multiple distinct feature groups or modalities that describe the same phenomenon or entity. Each feature group or modality is referred to as a view, and different views may provide complementary or redundant information. Multiview SSL aims to learn useful representations from multiview data by exploiting the inherent structures or patterns across views. Multiview SSL has a wide range of applications in biomedical domains, such as integrating multiple types of genomic data for disease diagnosis or prognosis, generating natural language descriptions from brain images, and understanding human behaviors during social interactions based on multimodal signals.

In this thesis, we propose novel approaches to multiview SSL that are scalable, flexible, and interpretable. We address the following research questions:How can we reformulate classical subspace learning methods as unconstrained optimization problems that can be solved by gradient descent? How can we extend classical subspace learning methods to nonlinear functions using deep neural networks? How can we incorporate different forms of regularization or prior knowledge into subspace learning methods to improve their quality or robustness?

To answer these questions, we develop novel methods for multiview subspace learning that leverage mathematical optimization techniques, deep neural networks, regularization techniques. We evaluate our methods on various real-world biomedical datasets and demonstrate their effectiveness and advantages over existing methods.

# Impact Statement

This thesis contributes to the advancement of machine learning and biomedical data analysis by developing novel methods for multiview self-supervised learning that are scalable, flexible, and interpretable. The proposed methods can help researchers and practitioners to analyze complex and high-dimensional biomedical data more efficiently and effectively, and to discover new insights and opportunities for improving health outcomes. The proposed methods can also be applied to other domains where multiview data are available or desirable, such as natural language processing, computer vision, multimedia analysis, and social network analysis. This thesis also provides a valuable reference for future research on multiview self-supervised learning and related topics.

# List of Publications

## First Author Peer Reviewed Conference Proceedings

Chapman, James, Lennie Wells, and Ana Lawry Aguila (2023). *Efficient Algorithms for the CCA Family: Unconstrained Objectives with Unbiased Gradients*. arXiv: 2310.01012 [cs.LG].

## First Author Peer Reviewed Conference workshop and Abstract

Chapman, James and Lennie Wells (2023). "CCA with Shared Weights for Self-Supervised Learning". In: *NeurIPS 2023 Workshop: Self-Supervised Learning - Theory and Practice*. URL: https://openreview.net/forum?id=7rYseRZ7Z3.

James Chapman Janaina Mourao-Miranda, John Shawe-Taylor (n.d.). *A Framework for Regularised Canonical Correlation Analysis by Alternating Least Squares*.

## First Author Pre-Print

Chapman, James, Ana Lawry Aguila, and Lennie Wells (2022). "A Generalized EigenGame with Extensions to Multiview Representation Learning". In: *arXiv preprint arXiv:2211.11323*.

## Co-Authored Peer Reviewed Journal

Mihalik, Agoston et al. (2022). “Canonical correlation analysis and partial least squares for identifying brain-behaviour associations: a tutorial and a comparative study”. In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*.

## Co-Authored Peer Reviewed Conference Proceedings

Lawry Aguila, Ana, James Chapman, and Andre Altmann (2023). “Multi-modal Variational Autoencoders for Normative Modelling Across Multiple Imaging Modalities”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer Nature Switzerland Cham, pp. 425–434.

Lawry Aguila, Ana, James Chapman, Mohammed Janahi, et al. (2022). “Conditional VAEs for Confound Removal and Normative Modelling of Neurodegenerative Diseases”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer Nature Switzerland Cham, pp. 430–440.

# LIST OF FIGURES

# LIST OF TABLES

# Acronyms

**ADNI** Alzheimer's Disease Neuroimaging Initiative. 8, 15, 42–44, 56, 57, 61, 62, 87, 95, 96, 98, 100

**CCA** Canonical Correlation Analysis. 14, 15, 21, 29–34, 37–41, 43, 46, 74–82, 85–87, 89–91, 100, 101, 103

**DCCA** Deep Canonical Correlation Analysis. 30, 40

**FRALS** Flexible Regularised Alternating Least Squares. 54–56

**GFA** Group Factor Analysis. 15, 74, 77, 78, 80, 81, 86, 89–91

**HCP** Human Connectome Project. 8, 15, 42–44, 56–58, 61, 87, 95, 96, 98, 100

**KCCA** Kernel Canonical Correlation Analysis. 39, 40

**MCCA** Multiset Canonical Correlation Analysis. 31

**MRI** Magnetic Resonance Imaging. 52

**PCA** Principal Component Analysis. 25–27, 38, 39

**PLS** Partial Least Squares. 27–31, 38, 39, 75, 87, 89, 91, 101

**RCCA** Regularized Canonical Correlation Analysis. 87, 89, 101

**sPLS** Sparse Partial Least Squares. 87, 89, 91

# Glossary

**latent variables** Latent variables are variables that are not observed. They are also called hidden variables. Latent variables are used to model the relationship between the observed variables.. 25

**loadings** The loadings of a latent variable or representation are the correlations between the latent variable or representation and the observed variables.. 8, 25, 75, 77–86, 91, 97, 99–102

**representations** Representations are the latent variables in a multiview dataset. They can be either the same or different types of data. The key assumption of multiview learning is that the views are related to each other in some sense. For example, in a dataset of images and text, the images and text are related because they describe the same object. We could also have a dataset containing images of the same object taken from different angles, in which case the images are still related because they describe the same object.. 7, 8, 25, 60, 62, 68, 72, 96

**views** Views are the observed variables in a multiview dataset. They can be either the same or different types of data. The key assumption of multiview learning is that the views are related to each other in some sense. For example, in a dataset of images and text, the images and text are related because they describe the same object. We could also have a dataset containing images of the same object taken from different angles, in which case the images are still related because they describe the same object.. 21–24, 27, 29, 37

**weights** The weights of a latent variable or representation are the coefficients of the linear combination of the observed variables that make up the factor.. 7, 8, 10, 25, 44, 46, 47, 52, 59–62, 66–68, 70, 72, 75, 85, 89, 91, 99–103

# CONTENTS

# Chapter I

# Introduction

In the middle of my PhD journey, in June 2021, I self-referred to the Community Living Well service in London, UK, for help with my mental health. I was assigned a therapist, who I met with weekly for 12 weeks. During our sessions, we discussed my mental health and the challenges I was facing. I was also asked to complete a questionnaire at the beginning and end of each session, which asked me to rate my mood and answer questions about my mental health. Each time I did this, I questioned how well these subjective numbers truly represented my feelings.

A keen sportsperson, I also wear a Garmin watch that tracks my heart rate, my sleep, and my activity levels. I use this data to monitor my health and fitness, and I have found it to be a useful tool in my training. Using a physical 'stress level' metric based on Heart Rate Variability (HRV), I can see how alcohol affects my sleep[1], how well I have slept, and I know I am about to get sick before I feel it.

As a type 1 diabetic, I rely on a continuous glucose monitor for my blood sugar levels. This device measures my blood sugar every five minutes, and I can see the results on my phone. I can also see how my blood sugar changes over time, and I can use this information to adjust my insulin doses and improve my control.

This thesis stems from a simple idea: What if we could combine different kinds of health data to get a clearer, more complete picture of a person's health?

[1] badly

# 1 Thesis Structure and Contributions

This thesis presents innovative methodologies for scaling multiview data fusion to massive datasets, aiming to transform the way biomedical data is analyzed and understood. By leveraging advancements in self-supervised learning and multiview learning, the research herein explores the integration of diverse data sources, similar to how my mental health, physical activity, and diabetes data each provide unique insights into my well-being.

The overarching aim is to develop methodological improvements that are practical and user-friendly. We strive to create tools and methods that are theoretically robust yet intuitive and straightforward to use in real-life scenarios. The goal is to empower practitioners in biomedical research and other fields to fully leverage their data, without requiring deep technical expertise in data analysis algorithms.

This thesis offers three primary contributions:

- Developing a regularization method for CCA using structured priors, including the Elastic Net, to improve interpretability.
- Proposing the use of loadings over weights in CCA for better interpretability and relevance to biomedical data generation processes.
- Creating a new gradient descent-based formulation for CCA and generalized eigenvalue problems, suitable for large datasets.

## 1.1 Chapter Summaries

**Chapter II** reviews multiview and self-supervised learning techniques, focusing on their application in biomedical data.

**Chapter III** introduces a method to regularize CCA using structured priors, demonstrated with Human Connectome Project and Alzheimer's Disease Neuroimaging Initiative data.

**Chapter IV** examines the relationship between loadings and weights in CCA, using simulated data to show the advantages of loadings for interpretability.

**Chapter V** presents a new loss function for generalized eigenvalue problems, applicable to CCA and SSL methods, and demonstrates its effectiveness with various benchmarks.

**Chapter ??** introduces CCA-Zoo, a Python package implementing the methodologies of this thesis, and discusses its role in the Python ecosystem and biomedical research.

**Chapter 7** discusses the implications, challenges, and future directions for the research presented in this thesis.

I hope that this thesis and the work it represents will help to bridge the gap between the potential of biomedical data and the capabilities of current analytical methods.

# Chapter II

# Background: Multiview Machine Learning: Concepts, Methods, and Limitations

> Principal Component Analysis is a dimensionally invalid method that gives people a delusion that they are doing something useful with their data. If you change the units that one of the variables is measured in, it will change all the “principal components”! It’s for that reason that I made no mention of PCA in my book. I am not a slavish conformist, regurgitating whatever other people think should be taught. I think before I teach.
>
> Professor David MacKay

Contents

# 1 Introduction to Machine Learning and Multiview Learning

In this chapter, we gather the necessary background knowledge needed to motivate and understand the contributions of this thesis.

Machine learning enables models to automatically learn patterns and make decisions from data. Machine learning comprises three primary paradigms: supervised, self-supervised (in the past called unsupervised), and reinforcement learning, each distinct in its approach to learning from data. This thesis focuses on *multiview self-supervised machine learning*, which aims to develop robust representations by uncovering associations between various data types within datasets. These data types, known as views may include distinct sources of information such as MRI images, genomic data, and clinical records in the context of patient data analysis.

## 1.1 Multiview Machine Learning

Multiview machine learning encompasses a variety of techniques aimed at learning from data that have multiple sources or modalities, also known as views. These techniques can be broadly classified into supervised and self-supervised (or sometimes, equivalently, unsupervised) multiview learning, with some algorithms straddling the boundary between the two.

### 1.1.1 Supervised Multiview Learning

In supervised multiview learning, one view serves as the input while the other view is treated as the target label. The algorithm learns to predict the target view based on the input view, leveraging the information from both to enhance the predictive performance (Zong, Mac Aodha, and T. Hospedales, 2023).

### 1.1.2 Self-Supervised Multiview Learning

Self-Supervised Learning (SSL) is a paradigm where the training signal is derived from the data itself, rather than relying on external labels (Balestriero et al., 2023). The cornerstone of SSL is the concept of a 'pretext task,' a learning task created from the data that trains the model to capture useful features or representations. In the context of multiview machine learning, self-supervised learning often operates under the assumption that different views are generated from a common source. A natural pretext task, in this case, is to predict or estimate this source from the given views. In the prediction setting, we might mask the source and train the model to predict it from the remaining views, closer to supervised learning. In the estimation setting, we never directly observe the source, but we train the model to estimate it from the views. In this case, the model is forced to learn the underlying structure of the data without any direct supervision. This not only enables the model to learn associations between views but also allows it to derive robust and informative representations for subsequent tasks like classification or regression. This is particularly true if the source is unavailable at test time when an application must rely on the views alone to make predictions. In the case of latent variables, where the assumed source is unobserved, the model learns to estimate latent variables from the views which are unavailable even at training time. These are usually much lower-dimensional than the original views, and therefore provide a more compact and informative representation of the data. The models we deal with in this thesis are generally of this type.

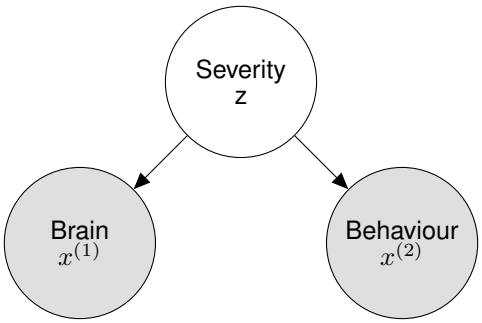



**Figure II.1:** ***Latent Variable Model of Mental Health:*** From this perspective the neuroimaging modality and behavioural data are both considered to have been generated with distributions conditioned on the severity of a mental health condition

## 1.2 Conditional Independence, Causality, and Multiview Learning

Consider the graphical model depicted in Figure II.1. It comprises two distinct observed views: a brain modality and a behavioral modality. The graphical represents the assumption that the brain and behaviour are conditionally independent given the severity of an unobserved 'latent' mental health condition.

In multiview machine learning, the relationship between conditional independence and causality is nuanced but crucial. When examining dependencies between events, such as those observed between brain activity and behavior, several scenarios emerge:

- direct causation (brain influencing behavior or vice versa or even both)
- both being influenced by a common, possibly unobserved, cause
- no direct causal link between them

Importantly, if a common cause does exist, conditioning on it renders brain and behavior independent; this 'screens off' their dependence, revealing key insights for our models (Reichenbach, 1956). However, it is essential to recognize that the presence of a common latent variable, inferred from these views, does not automatically imply causality in the observed data.

### 1.2.1 Complementary and Redundant Information

The nature of the information provided by different views (such as neuroimaging and behavioral data) is important for understanding multiview learning models. A particularly useful distinction is between *complementary* and *redundant* information (Nguyen and D. Wang, 2020). When views contain complementary information, they provide different perspectives on the same subject or sample. For example, we can understand different aspects of a mental health condition by examining both neuroimaging and behavioral data. On the other hand, when views contain redundant information about the latent variables, they provide the same information from different perspectives. For example, a disease diagnosis might be encoded in both neuroimaging and blood test results. This does not make the views useless, however, because they can be used to denoise each other, enhancing the clarity and reliability of the data. We can be more confident that a diagnosis is correct if it is supported by both neuroimaging and blood test results. A particularly famous example of this principle is the 'Wisdom of Crowds' effect, where the average of multiple noisy estimates is more accurate than any individual estimate (Galton, 1907). This process exploits the overlap in information to correct or reduce noise and errors, a principle fundamental to many denoising techniques in machine learning.

In this thesis we will work with Canonical Correlation Analysis, a multiview learning method which assumes that the views contain complementary information about latent variables. The next section builds a formal understanding of the principles behind Canonical Correlation Analysis and its variants.

# 2 Learning Representations: Definitions and Notation

Suppose we have a sequence of vector-valued random variables $X^{(i)} \in \mathbb{R}^{D_i}$ for $i \in \{1, \ldots, I\}$ We want to learn meaningful $K$-dimensional representations

$$Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)}). \tag{II.1}$$

For convenience, define $D = \sum_{i=1}^{I} D_i$ and $\theta = \left(\theta^{(i)}\right)_{i=1}^{I}$. Without loss of generality take $D_1 \geq D_2 \geq \cdots \geq D_I$. We will consistently use the subscripts $i, j \in [I]$ for views; $d \in [D_i]$ for dimensions of input variables; and $l, k \in [K]$ for dimensions of representations - i.e. to subscript dimensions of $Z^{(i)}, f^{(i)}$. Later on we will introduce total number of samples $N$.

In this report, when the functions $f$ are linear, we will typically refer to $u_k$ as *weights*, $Z_k = X_k u_k$ as *representations* or *latent variables*, depending on the context. We will sometimes consider a matrix $U = (u_1, \ldots, u_K) \in \mathbb{R}^{D \times K}$ of weights, and a matrix $Z = (Z_1, \ldots, Z_K) \in \mathbb{R}^{N \times K}$ of representations. We will refer to the Pearson correlation between features and their respective latent variable $\mathrm{Corr}(X_j^{(i)}, Z_k)$ as the *loadings* of $X_j^{(i)}$ on $Z_k$ (Rosipal and Krämer, 2005), noting that the same concept has also been referred to as *structure correlations* (Meredith, 1964).

## 2.1 Background: GEPs in linear algebra

A Generalized Eigenvalue Problem (GEP) is defined by two symmetric matrices $A, B \in \mathbb{R}^{D \times D}$ (Stewart and J.-G. Sun, 1990)[1]. They are usually characterized by the set of solutions to the equation:

$$Au = \lambda B u \tag{II.2}$$

with $\lambda \in \mathbb{R}, u \in \mathbb{R}^D$, called (generalized) eigenvalue and (generalized) eigenvector respectively. We shall only consider the case where $B$ is positive definite to avoid degeneracy. Then the GEP becomes equivalent to an eigen-decomposition of the symmetric matrix $B^{-1/2} A B^{-1/2}$. This is key to the proof of our new characterization. In addition, one can find a basis of eigenvectors spanning $\mathbb{R}^D$. We define a *top-$K$ subspace* to be one spanned by some set of eigenvectors $u_1, \ldots, u_K$ with the top-$K$ associated eigenvalues $\lambda_1 \geq \cdots \geq \lambda_K$. We say a matrix $U \in \mathbb{R}^{D \times K}$ *defines* a top-$K$ subspace if its columns span one.

**Uniqueness** In GEPs, the eigenvectors $u$ are not in general unique, but the canonical correlations $1 \geq \rho_1 \geq \rho_2 \geq \cdots \geq 0$ are unique (**mills1988calculation**).

## 2.2 Principal Components Analysis

Principal Components Analysis (Hotelling, 1933) (PCA) is a classical method in unsupervised machine learning for representation learning. It is widely used for dimensionality reduction and feature extraction. The primary goal of PCA is to transform the original high-dimensional data into a new coordinate system defined by orthogonal axes, capturing the most relevant aspects of the data.

[1] more generally, $A, B$ can be Hermitian, but we are only interested in the real case

In PCA, the representations are constrained to be linear transformations of the form:

$$Z_k = Xu_k, \tag{II.3}$$

where $u_k$ are the orthonormal basis vectors such that:

$$u_k^\top u_k = 1, \quad u_k^\top u_l = \delta_{kl} \text{ for } k \neq l. \tag{II.4}$$

The primary goal of PCA is to maximize the variance of the representations $Z_k$.

### 2.2.1 Optimization and Solution

Mathematically, for the first principal component, this can be formulated as:

$$u_{\text{opt}} = \underset{u}{\text{argmax}} \left(u^\top \Sigma u\right) \tag{II.5}$$

subject to:

$$u^\top u = 1$$

Where $\Sigma = \mathbb{E}[X^\top X]$ is the covariance matrix of the data.

The Lagrangian for this problem is:

$$f(u, \lambda) = u^\top \Sigma u + \lambda(1 - u^\top u), \tag{II.6}$$

where $\lambda$ is the Lagrange multiplier. Differentiating the Lagrangian yields the first-order conditions:

$$\Sigma u = \lambda u, \tag{II.7}$$

$$u^\top u = 1. \tag{II.8}$$

**Eigenvalue Problem** This transforms the problem into an eigenvalue equation for the covariance matrix $\Sigma$, which can be efficiently solved using standard libraries such as scikit-learn(Pedregosa et al., 2011).

The first principal component therefore corresponds to the eigenvector associated with the largest eigenvalue $\lambda$. Subsequent components are the remaining eigenvectors ordered by their corresponding eigenvalues.

### 2.2.2 Limitations

However, when applying PCA to datasets such as high-dimensional neuroimaging and behavioral data, PCA's main limitation arises: it only accounts for variance within a single dataset, so it cannot take advantage of the redundancy in multiview data.

## 2.3 Partial Least Squares

Partial Least Squares (PLS)(**wold1975path**) aims to maximize the shared covariance between two paired sets of data, referred to as views. PLS can be seen as a generalization of PCA, where PCA becomes a special case when the two views are identical.

### 2.3.1 Optimization and Solution

The optimization problem for PLS can be formulated as:

$$u_{\text{opt}}^{(1)} = \underset{u^{(1)}}{\operatorname{argmax}} \{ u^{(1)\,T} \Sigma_{12} u^{(2)} \} \tag{II.9}$$

subject to:

$$u^{(1)\,T} u^{(1)} = 1$$
$$u^{(2)\,T} u^{(2)} = 1$$

where $X^{(1)} \in \mathbb{R}^{n \times p_1}$ and $X^{(2)} \in \mathbb{R}^{n \times p_2}$, meaning we have two views with the same number of samples but potentially different number of features.

The Lagrangian for this optimization problem can be formulated as:

$$f(u^{(1)}, \lambda) = u^{(1)\,T} \Sigma_{12} u^{(2)} + \lambda_1 (1 - u^{(1)\,T} u^{(1)}) + \lambda_2 (1 - u^{(2)\,T} u^{(2)}) \tag{II.10}$$

Upon deriving the first order conditions, we get:

$$\Sigma_{21} u^{(1)} = \lambda_2 u^{(2)} \tag{II.11}$$

$$\Sigma_{12} u^{(2)} = \lambda_1 u^{(1)} \tag{II.12}$$

$$u^{(1)\,T} u^{(1)} = 1 \tag{II.13}$$

$$u^{(2)\,T} u^{(2)} = 1 \tag{II.14}$$

By substituting the constraint conditions into these equations, we find that $\lambda_1 = \lambda_2 = \lambda$ by symmetry. Further simplification yields:

$$\Sigma_{21}\Sigma_{12} u^{(2)} = \lambda^2 u^{(2)} \tag{II.15}$$

$$\Sigma_{12}\Sigma_{21} u^{(1)} = \lambda^2 u^{(1)} \tag{II.16}$$

**Eigenvalue Problem** Once again, we see that solving these equations will yield the $u^{(1)}$ and $u^{(2)}$ vectors as eigenvectors, this time of $\Sigma_{12}\Sigma_{21}$ and $\Sigma_{21}\Sigma_{12}$, respectively (Höskuldsson, 1988).

**Generalized Eigenvalue Problem** We can also represent the system of equations in matrix form as follows:

$$\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} = \lambda I \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} \tag{II.17}$$

Which is of the form $Av = \lambda Bv$. PLS is therefore also defined by the solution to a single generalized eigenvalue problem.

Given the notions of uniqueness in GEPs, the weights $u$ are not in general unique but we can write the vector of generalized eigenvalues, here representing covariances, as

$$\mathrm{PLS}_K(X^{(1)}, X^{(2)}) := (\rho_k)_{k=1}^K \tag{II.18}$$

### 2.3.2 Limitations

The problem with applying PLS to neuroimaging and behavioural modalities is that PLS is not scale invariant and is therefore biased towards the largest principal components in the data (Helmer et al., 2020). This is particularly problematic when

there is a low signal to noise ratio since PLS may find directions in either dataset which correspond to the largest directions of noise in the other. Additionally, PLS assumes that the structures contributing to variance in both datasets are linearly related, which may not be the case in complex biological systems like the brain or in intricate behavioral patterns (Rosipal and Krämer, 2005). The linearity assumption can sometimes be overly restrictive, failing to capture more complicated, nonlinear relationships between the data modalities. Another issue is the lack of sparsity in the PLS solution. Traditional PLS methods do not provide sparse weight vectors, which makes the interpretation of results challenging in high-dimensional settings such as neuroimaging where only a subset of features might be relevant. There are sparse variants of PLS available, but these typically introduce additional complexity and may require fine-tuning of regularization parameters (Chun and Keleş, 2010; D. M. Witten, R. Tibshirani, and Hastie, 2009). Furthermore, PLS can be sensitive to outliers, which are not uncommon in neuroimaging data due to motion artifacts or other sources of noise. Since the method aims to maximize covariance, extreme values in one dataset can disproportionately affect the resulting latent variables (Wold, 1973).

## 2.4 Canonical Correlation Analysis

In Canonical Correlation Analysis (CCA), we aim to find the directions that maximize correlation, as opposed to maximizing covariance between two views of a dataset. This nuance renders CCA invariant to feature scale.

### 2.4.1 Optimization and Solution

The optimization problem for CCA can be expressed as:

$$u_{\text{opt}} = \underset{u}{\operatorname{argmax}}\{u^{(1)\,T} X^{(1)\,T} X^{(2)} u^{(2)}\} \tag{II.19}$$

subject to:

$$u^{(1)\,T}\Sigma_{11}u^{(1)} = 1$$

$$u^{(2)\,T}\Sigma_{22}u^{(2)} = 1$$

Although non-convex, numerous methods exist for solving the CCA problem, including eigendecomposition and generalized eigendecomposition solvers(Uurtio et al., 2017) and block coordinate descent via alternating least squares regressions

(Golub and Zha, 1995; L. Sun, Ji, and Ye, 2008).

The first-order conditions derived in the same manner as the PLS case are:

$$\Sigma_{21}u^{(1)} = \lambda^{(2)}\Sigma_{22}u^{(2)} \tag{II.20}$$

$$\Sigma_{12}u^{(2)} = \lambda^{(1)}\Sigma_{11}u^{(1)} \tag{II.21}$$

$$u^{(1)\,T}\Sigma_{11}u^{(1)} = 1 \tag{II.22}$$

$$u^{(2)\,T}\Sigma_{22}u^{(2)} = 1 \tag{II.23}$$

**Eigenvalue Problems** Substituting the second two conditions into the first two, we get $\lambda^{(1)} = \lambda^{(2)} = \lambda$. Then, recognizing $X_i^\top X_i$ as the covariance matrix $\Sigma_{ii}$ and $X_i^\top X_j$ as the cross-covariance matrix $\Sigma_{ij}$, we obtain another pair of eigenvalue problems:

$$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}u^{(1)} = \lambda^2 u^{(1)}$$
$$\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}u^{(2)} = \lambda^2 u^{(2)}$$

An alternative form of the CCA problem can be developed by reparameterizing $u^{(i*)} = \Sigma_{ii}^{-\frac{1}{2}}u^{(i)}$. The optimization problem then becomes:

$$u_{\text{opt}} = \underset{u}{\operatorname{argmax}}\{u^{(1)\,T}\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}}u^{(2)}\} \tag{II.24}$$

subject to:

$$u^{(1)\,T}u^{(1)} = 1$$
$$u^{(2)\,T}u^{(2)} = 1$$

This reparameterized form will later underpin Deep Canonical Correlation Analysis (DCCA).

This form also shows that PLS and CCA can be made equivalent by whitening the data matrices before constructing the covariance matrix. When the number of features exceeds the number of samples ($p > n$), CCA becomes degenerate because the within-view covariance matrices cannot be inverted—contrasting with PLS, which is always computable.

**Generalized Eigenvalue Problem** We can also represent the system of equations in equation II.20 as a matrix equation:

$$\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} \tag{II.25}$$

Which is once again of the form $Av = \lambda Bv$. CCA, like PLS, is therefore also defined by the solution to a single generalized eigenvalue problem.

Given the notions of uniqueness in GEPs, the weights $u$ are not in general unique but we can write the vector of generalized eigenvalues, here representing covariances, as

$$\mathrm{CCA}_K(X^{(1)}, X^{(2)}) := (\rho_k)_{k=1}^K \tag{II.26}$$

## 2.5 Multiview CCA

Multiview CCA or MCCA is a straightforward extension of CCA to the case of 3-or more datasets. The goal is to find a set of directions $u^{(i)}$ such that the pairwise correlations between the views are maximized.

### 2.5.1 Optimization and Solution

The optimization problem for MCCA can be stated as:

$$u_{\text{opt}} = \underset{u}{\operatorname{argmax}} \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} u^{(i)\,T} \Sigma_{ij} u^{(j)} \tag{II.27}$$

subject to:

$$\sum_{i=1}^{m} u^{(i)\,T} \Sigma_{ii} u^{(i)} = 1$$

**Generalized Eigenvalue Problem** The generalized eigenvalue problem (GEP) for MCCA can be written in matrix form as follows:

$$\begin{pmatrix} 0 & \Sigma_{12} & \cdots & \Sigma_{1m} \\ \Sigma_{21} & 0 & \cdots & \Sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m1} & \Sigma_{m2} & \cdots & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(m)} \end{pmatrix} . = \lambda \begin{pmatrix} \Sigma_{11} & 0 & \cdots & 0 \\ 0 & \Sigma_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{mm} \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(m)} \end{pmatrix} . \tag{II.28}$$

This GEP formulation of MCCA can be presented in a unified framework generalizing CCA and ridge-regularized extensions. Indeed, we now take $A, B_\alpha \in \mathbb{R}^{D \times D}$ to be block matrices $A = (A^{(ij)})_{i,j=1}^I, B_\alpha = (B_\alpha^{(ij)})_{i,j=1}^I$ where the diagonal blocks of $A$ are zero, the off-diagonal blocks of $B_\alpha$ are zero, and the remaining blocks are defined by:

$$A^{(ij)} = \mathrm{Cov}(X^{(i)}, X^{(j)}) \text{ for } i \neq j, \quad B_\alpha^{(ii)} = \alpha_i I_{D^{(i)}} + (1 - \alpha_i)\mathrm{Var}(X^{(i)}) \tag{II.29}$$

Where $\alpha \in [0, 1]^I$ is a vector of ridge penalty parameters: taking $\alpha_i = 0 \, \forall i$ recovers CCA and $\alpha = 1 \, \forall i$ recovers PLS. We may omit the subscript $\alpha$ when $\alpha = 0$ and we recover the 'pure CCA' setting; in this case, following Equation (II.26) we can define $\mathrm{MCCA}_K(X^{(1)}, \ldots, X^{(I)})$ to be the vector of the top-$K$ generalized eigenvalues.

## 2.6 Linear Discriminant Analysis LDA

Linear Discriminant Analysis (LDA) can be viewed as a special case of Canonical Correlation Analysis (CCA) where $X^{(2)}$ is a one-hot encoded matrix representing the class labels. This allows us to draw a connection between the unsupervised learning framework of CCA and the supervised framework of LDA, thus expanding the understanding of both algorithms.

**Intuition:** In LDA, the aim is to find a lower-dimensional subspace where the classes are maximally separated. This objective can be viewed through the lens of CCA, where the optimal directions $u^{(1)}$ and $u^{(2)}$ in the original and one-hot encoded spaces aim to maximize correlation. In the LDA context, $u^{(1)}$ would maximize the separation between classes.

### 2.6.1 Optimization and Solution

Mathematically, LDA is reduced to solving a generalized eigenvalue problem involving the between-class scatter matrix $S_B$ and the within-class scatter matrix $S_W$:

$$\hat{S_B} = \sum_{i=1}^{c} n_i(\mu_i - \mu)(\mu_i - \mu)\top$$

$$\hat{S_W} = \sum_{i=1}^{c} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)\top$$

**Connection to CCA:** When $X^{(2)}$ is the one-hot encoded matrix of class labels, the CCA problem effectively tries to maximize the correlation between the feature vectors and their corresponding labels. This turns out to be equivalent to maximizing the between-class variance in LDA while minimizing the within-class variance. Thus, LDA can be thought of as a constrained form of CCA, tailored to classification tasks.

This perspective unifies the two algorithms and shows that the core objective—finding meaningful relationships or directions in the data—is shared between both CCA and LDA.

## 2.7 Sample Covariance and Population Covariance

In the previous sections, the methods were described in terms of population covariance matrices such as $\Sigma_{11} = \mathbb{E}[X^{(1)\,T} X^{(1)}]$, $\Sigma_{22} = \mathbb{E}[X^{(2)\,T} X^{(2)}]$, and $\Sigma_{12} = \mathbb{E}[X^{(1)\,T} X^{(2)}]$. These population covariances assume an underlying probability distribution from which the data are drawn.

**Sample Covariance:** In practical settings, we often do not have access to the entire population but only to a sample. Hence, we can utilize the Sample Average Approximation to estimate these covariances:

$$\hat{\Sigma}^{(12)} = \frac{1}{b-1} \bar{\mathbf{X}^{(1)}} \bar{\mathbf{X}^{(2)}}\top$$

Here, $b$ denotes the size of the minibatch, and $\mathbf{X}^{(1)} \in \mathbb{R}^{p \times b}$ and $\mathbf{X}^{(2)} \in \mathbb{R}^{q \times b}$ are the data matrices for the samples from $X^{(1)}$ and $X^{(2)}$, respectively. The bar over $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ signifies that these are centered versions of the matrices, i.e., the mean has been subtracted from each column.

**Practical Implications:** Using sample covariance matrices introduces some estimation error but allows us to apply the methods in real-world scenarios where population-level data are unattainable. Additionally, the use of minibatches provides a computationally efficient way to estimate these covariances in large-scale problems, at the cost of some additional statistical noise.

**Connection to Previous Methods:** The use of sample covariance matrices

is directly applicable to algorithms like CCA and LDA. When replacing the population covariances $\Sigma^{(ij)}$ with sample estimates, the optimization problems remain structurally similar but are solved using the sample data.

This dual perspective—considering both population and sample covariance matrices—enables a more robust and flexible approach to the methods discussed, bridging the gap between theoretical analysis and practical application. It will be particularly useful in the context of chapter IV where we will use population variables as ground truth while estimating the models using sample data.

# 3 Practical Frameworks for Multiview Learning

At this point, we have introduced the theoretical foundations of multiview learning, including CCA and its variants. However, it is not yet clear how we should apply these methods to real-world datasets.

## 3.1 Machine Learning and Statistical Inference

Canonical Correlation Analysis (CCA) has been studied from both machine learning and statistical inference perspectives. In this section, we will explore the differences between these two approaches and their implications for multiview learning.

### 3.1.1 Statistical Inference Evaluation Framework

Statistical inference approaches provide a contrasting perspective to machine learning methods, focusing on understanding and quantifying the underlying data structure:

**Parameter Estimation** In statistical inference, parameter estimation involves estimating model parameters and their uncertainties. This process is fundamental to understanding the data and the model's fit.

**Hypothesis Testing** Hypothesis testing assesses the statistical significance of the relationships found by the model. It tests whether the observed data patterns are likely to have occurred under the null hypothesis.

**Confidence Intervals** Confidence intervals provide ranges within which the true parameter values are likely to fall, considering uncertainty. They are essential for understanding the reliability of parameter estimates.

**Permutation Testing** Permutation testing is a non-parametric method that evaluates the significance of models. It compares model performance on the original data with performance on randomly shuffled data, helping to ascertain the results' robustness.

### 3.1.2 Machine Learning Evaluation Framework

**Training, Validation, and Test Sets** In machine learning, data is typically partitioned into training, validation, and test sets, each serving a specific purpose in the model development process:

- Training Set: Used for fitting the model.
- Validation Set: Assists in model parameter tuning.
- Test Set: Evaluates the model's generalization capability.

**Cross-Validation** A fundamental technique in machine learning, cross-validation involves dividing the training dataset into smaller subsets for training and validation. This approach provides insights into the model's performance across different data segments.

**Holdout Method** The holdout method involves using a separate dataset, not involved in training or validation, for final model assessment. This ensures an unbiased performance evaluation.

**Out of Sample Correlation** Specific to canonical correlation analysis, this involves measuring the correlation between latent variables in new datasets, assessing the model's ability to uncover relationships in unseen data.

**Downstream Tasks** Evaluating model performance on downstream tasks like classification or prediction can offer practical insights into the utility of the learned representations.

## 3.2 Components and Subspaces in CCA: A Subspace Perspective

### 3.2.1 Context: Eigenvalue Problems in CCA

While our focus so far has primarily been on the top-1 eigenvector-eigenvalue pair, it's important to note that the methodology also extends to the top-k subspace problem. This broader approach involves identifying the top-k eigenvectors and their corresponding eigenvalues.

### 3.2.2 Addressing the Top-k Problem

Transitioning from a focus on the top-1 component to exploring the top-k subspace introduces additional complexities. One common method to solve the top-k problem is to identify the top-1 component and then apply a deflation process to find subsequent orthogonal components. Deflation involves removing the top-1 component from the data and then repeating the process to find the next top-1 component. This process is repeated until the desired number of components is found. For instance, Hotelling's Deflation (Hotelling, 1933) involves removing the top-1 component from the data, while Projection Deflation (Mackey, 2008) involves projecting the data onto the orthogonal complement of the top-1 component. Different deflation methods enforce different forms of orthogonality, which can impact the resulting components and their interpretation, particularly when the first component is not a true eigenvector.

### 3.2.3 Non-Uniqueness of Components

Furthermore, non-uniqueness is a significant challenge in CCA, particularly when eigenvectors have repeated eigenvalues. Imagine a scenario where the top-1 eigenvalue is repeated $k$ times. In this case, there are $k$ possible eigenvectors that can be associated with the top-1 eigenvalue. While this is unlikely to occur in practice, the eigenvalues can in practice be very close to each other, leading to numerical instability and non-uniqueness in the components. Particularly true in cross-validation settings, this non-uniqueness can lead to instability in the components, complicating their interpretation and comparison. For example, the top-1 component in one analysis might be the second component in another analysis, making it difficult to compare the results.

This non-uniqueness also has a grounding in the probabilistic perspectives on PCA and CCA, where the latent variables are considered unique only up to a

rotation. This perspective further reinforces the subspace approach, emphasizing the identification of a subspace rather than specific directions within it.

**Thesis Approach: Concentrating on the Top-1 Component** In this thesis, we focus on the top-1 component in CCA to align with and facilitate comparison with typical componentwise studies in brain-behavior research. This choice is driven by the complexity associated with the top-k problem and the variety of methods available to address it. Under the assumption of a significant eigengap[2], the first component can be considered equivalent to the top-1 subspace. This equivalence allows for a clear and interpretable analysis, making the top-1 subspace a straightforward and reliable choice for studying multivariate data. It's important to note that while we focus on the top-1 component, the later sections of the thesis introduce a method for simultaneously solving the complete subspace, addressing broader subspace analyses.

# 4 Multiview Learning in Neuroimaging

There have been a number of applications of CCA and related methods to multiview problems in neuroimaging. Using resting state fMRI data, modes of correlation have been found that relate to differences in sex and age relating to drug and alcohol abuse, depression and self harm (Mihalik, Ferreira, Rosa, et al., 2019). A similar mode relating to 'positive-negative' wellbeing has been found across studies (Stephen M Smith et al., 2015)suggesting that mental wellbeing has a relationship (though not necessarily causally) with functional connectivity between networks in the brain. Later in this dissertation we will replicate and build on the findings from this paper by using regularised and non-linear CCA methods.

CCA has also been used as a preprocessing step in order to identify groups of subjects in the latent variable space. In particular, CCA and clustering have been used to identify depression using fMRI data (Dinga et al., 2019; Drysdale et al., 2017). CCA has also been used in the manner we described to denoise two views of a dataset such as separate measures of neuroimaging data (Zhuang, Yang, and Cordes, 2020) to remove artefacts. Deep CCA has recently been used to extract features for the diagnosis of schizophrenia(Qi and Tejedor, 2016).

[2]An 'eigengap' refers to the difference in magnitude between consecutive eigenvalues in an eigenvalue problem. A significant eigengap between the first and second eigenvalues suggests that the first eigenvalue (and its corresponding eigenvector) is distinctly more significant than the next, lending credence to its uniqueness and importance.

# 5 Open challenges in Multiview Learning and CCA

This thesis has been motivated by a number of open challenges in multiview learning and canonical correlation analysis. Chapter III and IV will address the first challenge, which is the regularisation of CCA in high dimensional settings and the interpretation of the resulting components. Chapter **??** and **??** will address the second challenge, the efficient application of CCA to big data. Finally **??** will also address the third challenge, extending CCA to Deep Self-Supervised Learning.

## 5.1 Interpretability and Regularization

TODO: Add a paragraph on interpretability and regularization

## 5.2 Efficient Algorithms for High-Dimensional Data

The challenges of high-dimensional data often manifest when solving Generalized Eigenvalue Problems (GEPs) for Canonical Correlation Analysis (CCA). The computational burden of solving these problems becomes daunting as the number of features grows. To combat this issue, various efficient algorithms have been developed to reduce the complexity. In this section, we will explore some of these strategies.

### 5.2.1 Challenges in Solving Generalized Eigenvalue Problems

The GEP is often represented as $Ax = \lambda Bx$, where $A$ and $B$ are matrices. To generalize the dimensions of these matrices, let's denote them as $m \times m$. This dimension $m$ can vary based on the specific method in use. For instance, in Principal Component Analysis (PCA), represented as PCA, $m$ would be equal to $p$ since $A$ and $B$ are $p \times p$ matrices. In methods like Partial Least Squares (PLS) and Canonical Correlation Analysis (CCA), represented as PLS and CCA respectively, $m$ would be $p + q$, as $A$ and $B$ in these cases are $(p + q) \times (p + q)$.

To solve the GEP, one common technique is to transform it into a standard eigenvalue problem $B^{-\frac{1}{2}}AB^{-\frac{1}{2}}y = \lambda y$, followed by eigendecomposition. However, this approach has computational complexity $O(m^3)$ and may suffer from numerical instability.

| Method | $A$ | $B$ | $x$ | Dimensions |
|---|---|---|---|---|
| PCA | $\Sigma_{11}$ | $I$ | $u^{(1)}$ | $p \times p$ |
| LDA | $S_B$ | $S_W$ | $u^{(1)}$ | $p \times p$ |
| CCA | $\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$ | $\begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix}$ | $\begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix}$ | $(p+q) \times (p+q)$ |
| PLS | $\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix}$ | $I$ | $\begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix}$ | $(p+q) \times (p+q)$ |

**Table 5.1:** Definitions and dimensions of $A$and $B$for different subspace learning methods.

### 5.2.2 PCA-CCA

One way to reduce the complexity of solving GEPs is to use the PCA-CCA method, which first applies PCA to the data and then solves the GEP in the reduced space. An important advantage of using PCA-CCA is computational efficiency, especially for high-dimensional data. The overall complexity of PCA-CCA involves two main steps. First, applying PCA has a complexity of $O(\max(p^3, q^3))$, dominated by the larger of the two matrices. Second, solving the generalized eigenvalue problem in the reduced space with $K$ components in each view leads to a complexity of $O((2K)^3)$. Thus, the overall complexity of PCA-CCA is $O(\max(p^3, q^3) + (2K)^3)$, which is significantly lower than the complexity of solving the GEP directly. Since CCA, ridge CCA, and PLS can all be solved in the principal component space, PCA-CCA can be used to compute solutions efficiently *even if we keep all the principal components*. Most obviously, this is the case when the number of samples $n$ is smaller than either of the number of features $p$ or $q$, i.e. $n < p$ or $n < q$. In this case the maximum number of principal components is $K = n$, and the complexity of PCA-CCA is thus $O(n^3)$, which is significantly lower than the complexity of solving the GEP directly. This approach has been employed to great effect in neuroimaging but suprisingly is not used even in the scikit-learn implementation of CCA (Pedregosa et al., 2011). Nonetheless, for the large sample sizes (desirable for machine learning frameworks as well as statistical power), the complexity of even PCA-CCA can render the problems nearly intractable.

Kernel CCA (KCCA) also offers computational efficiency for high-dimensional data ($p > n$) as its complexity scales with the number of samples $n$, not the number of features $p$. It casts the CCA optimisation as a dual problem:

$$\alpha_{\text{opt}} = \underset{\alpha_{\text{opt}}}{\operatorname{argmax}} \{\alpha^{(1)} K^{(1)\,T} K^{(2)} \alpha^{(2)}\} \tag{II.30}$$

subject to:

$$\alpha^{(1)} K^{(1)\,T} K^{(1)} \alpha^{(1)} = 1$$

$$\alpha^{(2)} K^{(2)\,T} K^{(2)} \alpha^{(2)} = 1$$

Where $\alpha^{(1)} =$

The kernel function in KCCA can be computed iteratively on pairwise comparisons of samples, allowing for memory efficiency by not requiring the entire dataset to be loaded into RAM. This iterative approach uses slow hard drive memory access instead of RAM, making KCCA RAM memory-efficient but slower. However, a significant drawback of KCCA is the need for access to all training data at test time, which raises concerns about efficiency and scalability.

## 5.3 Non-linear CCA and Joint Embedding Self-Supervised Learning

As is standard in Kernel methods, KCCA can be extended to non-linear CCA by using a non-linear kernel function $k$. This allows for the discovery of non-linear relationships between the data modalities. However, the non-linear kernel function $k$ is typically fixed a priori, which can be problematic since the optimal kernel function may be unknown.

### 5.3.1 Deep CCA

Deep CCA (DCCA) is a non-linear extension of CCA that uses deep neural networks to find non-linear relationships between variables.

### 5.3.2 Joint Embedding Self-Supervised Learning

Joint Embedding Self-Supervised Learning is a method for learning representations of data that are useful for downstream tasks.

In many self-supervised learning methods, the data is augmented to produce multiple views of the same data. For example, in the case of images, the image can be cropped, rotated, or flipped to produce multiple views of the same image. The goal is to learn representations of the data that are invariant to these augmentations

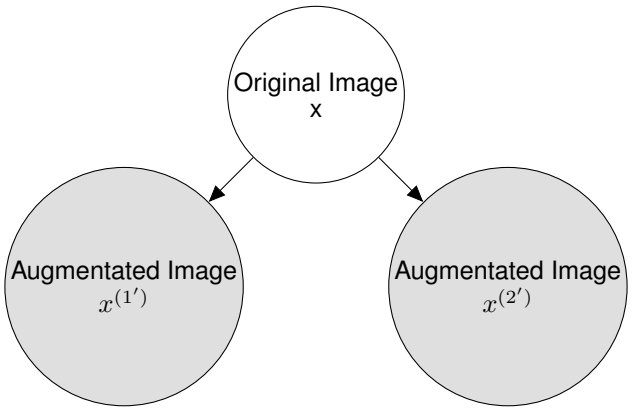



**Figure II.2:** ***Joint Embedding Data Generation Process:*** The original image $x$ is augmented twice to produce two augmentated images $x^{(1')}$ and $x^{(2')}$.

and thus capture the underlying structure of the data (e.g. features which are generally useful). From the background we have given on CCA, it is clear that CCA is a natural fit for this problem, precisely because all of the information is redundant rather than complementary; the augmentations are all views of the same data and they are chosen randomly.

# Chapter III

# Regularization of CCA Models

Contents

## Preface

This chapter expands on my work previously showcased at the OHBM conference and draws connections to a tutorial paper I co-authored, where I contributed a number of simulations (Mihalik, Chapman, et al., 2022).

This chapter explores the role of regularization in improving the performance and interpretation of CCA using simulated and brain-behaviour data. We develop a framework for regularized CCA which allows us to incorporate any regularized least squares solver to efficiently implement a wide range of regularization functions using any scikit-learn compatible solver, but in particular allows us to efficiently implement the elastic net penalty with controllable L2 and L1 penalties so that we can control the bias towards the largest principal components while still encouraging sparsity in the weights in contrast to most previous work on sparse Brain-Behavior analysis which has used a PLS objective with lasso constraints (SPLS), which inherits a bias towards the largest principal components from PLS.

## 1 Introduction

Large datasets neuroimaging datasets including the Human Connectome Project (HCP) and Alzheimer's Disease Neuroimaging Initiative (ADNI) datasets as well as the UK Biobank and Adolescent Brain Cognitive Development (ABCD) datasets [1] appear to offer unprecedented opportunities for understanding the relationship between brain structure and function and behavior (Stephen M. Smith and Thomas E. Nichols, 2018; Bzdok and B.T. Thomas Yeo, 2017; H.-T. Wang et al., 2020). Despite the impressive scale of these modern neuroimaging studies, the number of subjects is still often orders of magnitude smaller than the number of features in the data; the ADNI dataset we use in this chapter, for example, has 592 subjects and 168,130 structural MRI voxels. In this context, CCA models are prone to overfitting, leading to spurious correlations and poor generalization (Helmer et al., 2020; Mihalik, Ferreira, Moutoussis, et al., 2020). But given the reproducibility crisis in neuroscience (Button et al., 2013), it is important to ensure that our models generalize well to new data.

Regularization, having been extensively studied and well-understood in the contexts of Linear Regression and Inverse Problems (Engl, Hanke, and Neubauer,

[1] Not covered in this chapter

1996), introduces a deliberate bias to guide models towards more generalizable solutions, can be a powerful tool for addressing these problems. Furthermore, regularization can help us improve the interpretability of the results by imposing structure on the solutions, perhaps most obviously by encouraging sparsity(Bzdok, Thomas E Nichols, and Stephen M Smith, 2019). However, due to the complexity of the CCA problem, regularization is not as straightforward as in Linear Regression. The most popular approaches to 'sparse CCA' have, in practice, been based on Partial Least Squares (PLS), which simplifies the optimization problem but, as we shall see, causes the model to inherit a bias towards the largest principal components from PLS.

With this perspective in mind, we propose a flexible regularized alternating least squares (FRALS) framework for CCA which allows us to incorporate any regularized least squares solver to efficiently implement a wide range of regularization functions, but in particular allows us to efficiently implement the elastic net penalty with controllable L2 and L1 penalties so that we can control the bias towards the largest principal components while still encouraging sparsity in the weights. This is in contrast to much of the previous work on sparse Brain-Behavior analysis which has used a PLS objective with lasso constraints (SPLS), which inherits a bias towards the largest principal components from PLS.

We apply FRALS with Elastic Net regularization to the HCP and ADNI datasets. We show that it outperforms other CCA models in terms of out-of-sample canonical correlation. We also show that the identified mode of variation is distinct from previous work which identified latent variables with weights related to cognitive tests and negatively related to cigarette, tobacco or alcohol(Stephen M Smith et al., 2015). FRALS has stronger correlations with the Line Orientation test, which measures visuospatial abilities, and the parietal lobe, which is known to be involved in visuospatial processing.

# 2 Background: Regularization for High-Dimensional and Structured Data

In this section, we review a number of regularization techniques that have been applied to CCA and related methods.

## 2.1 The Bias-Variance Tradeoff

A key principle in machine learning is the bias-variance tradeoff. This concept posits that a tradeoff exists between the bias and variance of a model: high-bias models typically exhibit low variance, and vice versa. High-bias models are generally simpler and more stable, but they might oversimplify the problem, leading to underfitting. Conversely, low-bias, complex models are sensitive to data changes and prone to overfitting. As the number of features increases, there are more parameters to estimate, and models tend to become more complex, leading to higher variance and lower bias. This relationship highlights the importance of balancing model complexity to avoid overfitting, particularly in high-dimensional scenarios with a low signal-to-noise ratio (McIntosh, 2021)[2]. Regularization can be understood as a method for reducing the variance of a model by introducing a bias towards simpler models. This means regularization can improve the generalizability of models in high-dimensional settings.

**Implicit and Explicit Regularization** We can implement regularization in two different ways. *Explicit* regularization is achieved by adding a penalty term to the objective function. Weights the objective function against a term that penalizes complexity.

*Implicit* regularization is achieved by changing the optimization algorithm.

## 2.2 Shrinkage Regularization

Shrinkage regularization is a form of regularization that penalizes the magnitude of the model parameters. This technique is particularly effective in enhancing the performance of linear models in situations characterized by high dimensionality, multicollinearity, or low signal-to-noise ratios.

In high-dimensional situations where the number of features exceeds the number of observations in either view, Like Linear Regression, Canonical Correlation Analysis is non-identifiable, meaning there is no unique solution. This is because we can find perfectly correlated latent variables using a linear combination of the features, but there are many different linear combinations that will achieve this. Some of these linear combinations will generalize better than others, but there is no way to distinguish between them using the training data alone.

[2] It's worth noting that the number of model parameters, often used as a proxy for complexity, does not always directly correlate with model behavior, as illustrated by the 'double descent' phenomenon.

Even in low-dimensional situations, if features exhibit multicollinearity, they can also be non-identifiable or, at best, estimates of the parameters are unstable. Mathematically, this is because in both cases the covariance matrix of the features is not full rank and therefore is not invertible (non-identifiable) or ill-conditioned (matrix inversion is unstable). To capture this intuition, if two features are perfectly correlated, the model is not identifiable (has no unique solution) because we can arbitrarily swap the weights between the two features without changing the latent variables (CCA) or the predictions (regression). In practice, features are rarely perfectly correlated, but even when features are highly correlated, the model can be unstable (Mihalik, Ferreira, Moutoussis, et al., 2020), and small changes in the data can lead to large changes in the model parameters. Once again, some of these linear combinations will generalize better than others, but we might expect a model to generalize better if it spreads the weights across the correlated features rather than concentrating them on a single feature.

Finally, even in low-dimensional settings with little multicollinearity, the model parameters can sensitive to noise in the data, and once again small changes in the data can lead to large changes in the model parameters. For example, parameters associated with noisy features might 'cancel out' in the training set, but not in the test set, leading to poor generalization.

The premise of shrinkage regularization in all these cases is that the latent variables or predictions are too sensitive to small changes in the data because the model parameters are too large. Shrinkage regularization works by shrinking the model parameters towards zero, so that small changes in the data do not lead to large changes in the model estimates.

**PLS as Shrinkage Regularization** PLS can be interpreted as a form of shrinkage regularization applied to CCA. We can explain this by considering an analogy between CCA and *Linear Regression*[3].

In Linear Regression, the ridge regression solution is given by:

$$\hat{\beta}_{\text{ridge}} = ((1-c)\Sigma_{X,X} + cI)^{-1}\Sigma_{X,y} \tag{III.1}$$

Where $c$ is the regularization parameter between 0 and 1[4]. The ridge penalty acts in three important ways:

[3] indeed Linear Regression is a special case of CCA where $X^{(2)}$ has one feature

[4] It is more common to see $(\Sigma_{X,X} + cI)^{-1}\Sigma_{X,y}$ but these are equivalent up to a scalar factor and this form helps us later on

- It shrinks the weights towards zero.
- It shrinks the weights of correlated features towards each other.
- It biases the solution to high covariance directions rather than high correlation directions.

As $c$ becomes large, $\lim_{c\to\infty}(\Sigma_{X,X}+cI)^{-1}=(cI)^{-1}$ , so that $\hat{\beta}_{\text{ridge}}=\frac{\Sigma_{X,y}}{c}$, which is precisely the covariance of the features of $X$ with $Y$ scaled by $c$ (and shrunk towards zero for $c\geq 1$). Notice that the ridge regression solution is no longer sensitive to the correlation of features in $X$. Additionally, notice that for sufficiently large $c$, $(\Sigma_{X,X}+cI)$ is invertible even if $\Sigma_{X,X}$ is not invertible, so that ridge regression is always identifiable even when the number of features exceeds the number of observations.

Now consider the CCA problem. Firstly, recall that PLS and CCA are equivalent up to a scaling when the covariance matrices are identity matrices, a similar relationship to the relationship between Linear and Ridge Regression. Consider the well-known form of CCA given in equation III.2(Mihalik, Chapman, et al., 2022) (formed by reparameterizing $u^{(i)}=(\Sigma_{ii})^{-\frac{1}{2}}u^{(i)}$):

$$u_{\text{opt}}=\underset{u}{\operatorname{argmax}}\{u^{(1)\,T}(\Sigma_{11}+cI)^{-\frac{1}{2}}\Sigma_{12}(\Sigma_{22}+cI)^{-\frac{1}{2}}u^{(2)}\} \tag{III.2}$$

subject to:

$$u^{(1)\,T}u^{(1)}=1, u^{(2)\,T}u^{(2)}=1$$

As we increase $c$, $\lim_{c\to\infty}(\Sigma_{ii}+cI)^{-\frac{1}{2}}=(cI)^{-1}$ so that the objective approaches:

$$u_{\text{opt}}=\underset{u}{\operatorname{argmax}}\{u^{(1)\,T}(cI)^{-1}\Sigma_{12}(cI)^{-1}u^{(2)}\} \tag{III.3}$$

subject to:

$$u^{(1)\,T}u^{(1)}=1, u^{(2)\,T}u^{(1)}=1$$

Which is precisely the PLS objective and constraints with an arbitrary scaling of the covariance matrix $\Sigma_{12}$ by $\frac{1}{c^2}$. For this reason, we can consider PLS as an explicit shrinkage method for CCA, equivalent to adding a maximal ridge regularization term. The downside of using PLS as a regularised CCA is precisely its very

high bias. By strongly guiding the model towards high covariance solutions, it strongly biases the solution towards only the largest principal components. But what if the correlation between the views is not concentrated in the largest principal components? Although one would rarely resort to maximally regularized ridge regression except in extremely low sample sizes or high-dimensional data, it has become almost standard practice to use PLS in neuroimaging and genetics (Cruciani et al., 2022; Krishnan et al., 2011). One of the core contributions of this chapter will be to demonstrate that PLS is usually a poor choice for regularization even in these very high-dimensional settings and that more nuanced regularization methods can offer significant improvements in performance and interpretability. PLS is evidently not a nuanced tool for regularization because it offers no control over the degree of regularization applied.

**Ridge Regularization** For this reason, Vinod (1976) proposed the *Canonical Ridge* or *Ridge CCA*, which combined the PLS and CCA constraints in a single constrained optimization:

$$u_{\text{opt}}^{(1)} = \underset{u^{(1)}}{\text{argmax}}\{u^{(1)\,T}\hat{\Sigma_{12}}u^{(2)}\} \tag{III.4}$$

subject to:

$$(1-c_1)u^{(1)\,T}\hat{\Sigma_{11}}u^{(1)} + c_1 u^{(1)\,T}u^{(1)} = 1$$
$$(1-c_2)u^{(2)\,T}\hat{\Sigma_{22}}u^{(2)} + c_2 u^{(2)\,T}u^{(2)} = 1$$

Where $c_1$ and $\tau_2$ are the ridge regularization parameters for the first and second views respectively. By tuning these parameters, we can control the degree of regularization applied to each view independently. If we set $c_1$ and $c_2$ to zero, we recover the standard CCA objective while if we set $c_1$ and $c_2$ to one, we recover the PLS objective. This allows us to interpolate between the two extremes, allowing us to control the level of shrinkage and therefore the level of bias towards the largest principal components

**PCA-CCA** PCA be can be used as an implicit regularization method for CCA.

Most obviously, by using only the first $k$ principal components of each view as the input to CCA, we can reduce the dimensionality of the data and therefore reduce the number of parameters in the model. Moreover, by working with the principal components, we remove the correlation between the features, which can improve

the conditioning of the problem.

**A Visual Comparison of Shrinkage Techniques** The distinct effects of Ridge and PCA on the eigenvalues of the effective covariance matrices can be clearly visualized with a simple visualisation. We plot the eigenvalues of covariance matrices as perceived by models with different regularization techniques [5]. As shown in Figure III.1, Ridge regularization reduces the magnitude of the largest eigenvalues in the effective covariance matrix towards 1, and increases the magnitude of the smallest eigenvalues towards 1. On the other hand, PCA-CCA, leaves the largest eigenvalues unchanged, and ignores the smallest eigenvalues (we could have represented this by setting them to infinity).

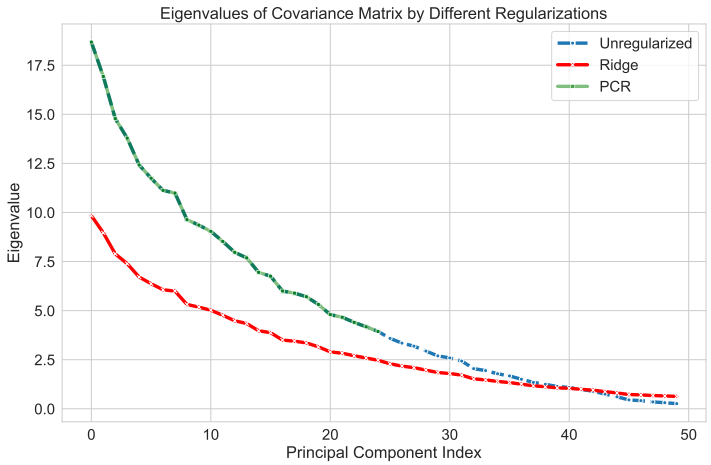



**Figure III.1:** Comparison of the effect of OLS, Ridge, and PCA regularization on the eigenvalues of the covariance matrix.

When these effective covariance matrices are inverted to form the CCA objective, these effects are reversed. Ridge regularization increases the magnitude of the weights associated with the largest eigenvalues and decreases the magnitude of the smallest eigenvalues. PCA maintains the weights associated with the largest eigenvalues and sets the weights associated with the smallest eigenvalues to zero. The visualization underscores the intrinsic nature of each regularization method:

- **Unregularized**: Presents the unaltered spectrum, making it susceptible to noise but preserving potential subtle patterns.

[5] e.g. the eigenvalues of $(1 - c_i)\hat{\Sigma_{ii}} + c_i I$ for ridge and $\hat{\Sigma_{ii}}$ truncated to include only the largest $k$ principal components for PCA

- **Ridge**: Warps the spectrum, shrinking the largest eigenvalues and expanding the smallest eigenvalues, potentially missing subtle patterns but offering a cleaner representation of stronger associations.
- **PCA**: Truncates the spectrum, ignoring the smallest eigenvalues and preserving the largest eigenvalues, potentially missing subtle patterns but offering a cleaner representation of stronger associations.

However, while these shrinkage techniques can improve the performance of CCA, they do not obviously improve the interpretability of the results. Weights are shrunk towards zero, but they are not set to zero. This means that the model still uses all the features, and the results are not sparse.

## 2.3 Sparse Regularization

Sparse regularization is a powerful tool for improving the performance and interpretability of linear models. Sparse regularization encourages the model to use only a subset of the features, which can both help to avoid overfitting and improve the interpretability of the model. Sparse regularization works on the premise that only a subset of the features are relevant to the model. Sparsity is typically achieved by adding either an L1 penalty or constraint[6]. The L1 penalty is defined as:

$$\|u\|_1 = \sum_i |u_i| \tag{III.5}$$

Intuitively, this is the sum of the absolute values of the elements of the vector. Now, with a foundational understanding of sparse regularization, we review a number of approaches to adding sparsity to the CCA problem.

**Sparse PLS: Penalized Matrix Decomposition** Penalized Matrix Decomposition (PMD) (D. M. Witten, R. Tibshirani, and Hastie, 2009) provides an approximate solution to the sparse CCA problem by altering the constraints of the classical CCA formulation. Specifically, PMD replaces the constraints $u^{(i)\,T}\hat{\Sigma_{ii}}u^{(i)} = 1$ with the PLS constraints $u^{(i)\,T}u^{(i)} = 1$ and additionally imposes $\|u^{(i)\,T}\|_1 \leq \tau$. The optimization problem for PMD is then given by:

[6]The L0 norm of the weight vector is the number of non-zero elements in the vector and is arguably a closer match to the goal, but the L0 norm is (a) not a proper norm in the mathematical sense and (b) not convex and so is difficult to optimize.

$$u^{opt} = \underset{u}{\operatorname{argmax}}\{u^{(1)\,T}\hat{\Sigma_{12}}u^{(2)}\} \tag{III.6}$$

subject to:

$u^{(1)\,T}u^{(1)} = 1, u^{(2)\,T}u^{(2)} = 1$

$\|u^{(1)}\|_1 \leq \tau_1, \|u^{(2)}\|_1 \leq \tau_2$

This Sparse PLS (SPLS) approximation has been highly influential as a form of Sparse CCA because it is extremely computationally efficient method [7]. There are a number of other sparse CCA methods that employ the PLS approximation(Parkhomenko, Tritchler, and Beyene, 2009; Waaijenborg, Witt Hamer, and Zwinderman, 2008; Lindenbaum et al., 2021). However, while the PLS approximation is efficient, it means these methods inherit a bias towards the largest principal components from PLS.

To address these problems and truly tackle the sparse CCA optimization, another class of approaches have adopted a penalized least squares approach.

**Sparse CCA: Least Squares Approaches** It is well known that the CCA problem can be formulated as a constrained least squares problem with the intuition that for $X^{(1)}u^{(1)} = 1$ and $X^{(2)}u^{(2)} = 1$, correlation is maximized when the squared distance between $X^{(1)}u^{(1)}$ and $X^{(2)}u^{(2)}$ is minimized. (Golub and Zha, 1995) proved the convergence of a simple algorithm which alternates between solving the least squares problem for $u^{(1)}$ and $u^{(2)}$ while keeping the other fixed.

With this intuition, Wilms and Croux, 2015 and Mai and Zhang, 2019 separately proposed iterative penalized least squares methods for sparse CCA.

$$u^{opt} = \underset{u}{\operatorname{argmin}} \left\{ \|X^{(1)}u^{(1)} - X^{(2)}u^{(2)}\|_2^2 + P(u) \right\} \tag{III.7}$$

subject to:

$u^{(1)\,T}\hat{\Sigma_{11}}u^{(1)} = 1$

$u^{(2)\,T}\hat{\Sigma_{22}}u^{(2)} = 1$

Where $P(u)$ is a penalty function. The penalty term can be any function that penalizes the norm of the vector $u$. (Mai and Zhang, 2019) proved that solving

[7] it can be solved by a variant of the power method; iteratively multiplying $u^{(1)}$ by $\hat{\Sigma_{12}}$ and soft-thresholding

the subproblems where one of $u^{(i)}$ is fixed is easy for one-homogenous $P$ where $P((\mu+1)\theta) = (\mu+1)P(\theta)$ which notably includes the lasso penalty. This means a sparse CCA based on alternating lasso regressions can be solved relatively efficiently using existing solvers. However, the one homogenous penalty in practice limits the flexibility of the method. For example, the elastic net penalty is not one-homogenous and therefore cannot be used with this method. Chi et al. (2013) and Mullins et al., 2021 added ridge penalties to the subproblems to improve the conditioning of the problem in a way that could be considered a form of elastic net regularization but the subproblems no longer correctly optimize the global objective[8].

**Sparse CCA: Proximal Gradient Descent and ADMM** Kanatsoulis et al. (2018) proposed solving equation III.7 for more general classes of $P$ using the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). Fu et al., 2017 propose a regularized CCA based on an alternative classical CCA formulation, sometimes called the MAXVAR formulation, which views the problem as a constrained least squares with an auxiliary representation $T$(Carroll, 1968; Kettenring, 1971).

$$\underset{U,T}{\operatorname{argmin}} \left\{ \sum_i \|X^{(i)}U^{(i)} - T\|_F^2 \right\} \tag{III.8}$$

$$\text{subject to: } T^\top T = I \tag{III.9}$$

$$\tag{III.10}$$

In this formulation, $U^{(i)}$ represents the weights for the $i^{\text{th}}$ view, and $T$ denotes the latent variable matrix. The premise is that when $T$ closely mirrors $X^{(i)}U^{(i)}$ across all $i$, the scores correlate. Notably, this method is adaptable to multiple views. The authors employed proximal gradient descent for regularization, specifically suited for penalties like the lasso. While these methods are flexible, they don't have the plug-and-play nature of the penalized least squares methods. Not just a matter of convenience, this means that these methods are not compatible with existing solvers for regularized least squares problems like for example total variation regularisation solvers in nilearn, which are often highly optimized for specific problems and modalities.

**Structured Regularization** As highly structured data, linear models using both Structural and Functional MRI data have been shown to benefit from structured

[8]when rescaling the penalized solutions back to unit variance

regularization methods but notably these methods have not been applied to CCA. Total variation regularization, which biases spatially neighboring weights to be similar, has been shown to improve the performance of PCA (De Pierrefeu et al., 2017) and regression (Michel et al., 2011; Dohmatob et al., 2014; Baldassarre, Mourao-Miranda, and Pontil, 2012). Similarly, Laplacian (or *GraphNet*) regularization, which induces a similar spatial bias with additional smoothness, has been shown to improve the performance of CCA on functional MRI data (Grosenick et al., 2013).

Having discussed the benefits of both shrinkage (e.g., PCA-CCA, Ridge CCA, PLS), sparsity (SPLS, Sparse CCA), and structure (Total Variation, Laplacian) in handling high-dimensional, noisy, and structured data, a natural progression is to integrate these advantages. Specifically, the challenge lies in creating a framework that allows for users to match the regularization method to their data and research question, enhancing the interpretability and performance of Brain-Behaviour association models. The solution? A method that employs readily available regularized regression solvers, allowing for flexible and tunable regularization in CCA. This leads us to propose the Flexible Regularized Alternating Least Squares (FRALS).

# 3 Methods - Flexible Regularized Alternating Least Squares (FRALS)

The primary goal of our Flexible Regularized Alternating Least Squares framework is to provide a versatile and user-friendly interface for Canonical Correlation Analysis (CCA). This is achieved by designing the framework to be compatible with any scikit-learn compatible regularized least squares solver. This compatibility is pivotal as it allows researchers and practitioners to leverage the extensive range of solvers available in scikit-learn, a popular machine learning library in Python.

This approach marks a significant departure from traditional methodologies in CCA, which often focused on developing or utilizing specific solvers tailored for particular types of data or computational constraints. By contrast, FRALS democratizes access to advanced CCA techniques, allowing users to select solvers that best fit their specific data characteristics, computational needs, or familiarity. Such flexibility is particularly advantageous in interdisciplinary fields like neuroimaging, where diverse datasets and varying levels of technical expertise are common.

For example, users dealing with high-dimensional, sparse neuroimaging data could opt for solvers optimized for such datasets, while those needing parallel computation for large data sets might choose solvers with GPU acceleration capabilities. In principle, FRALS can even be used with Neural Network-based solvers, which are becoming increasingly popular in machine learning[9]. This adaptability enhances FRALS' accessibility and future-proofs the framework against evolving computational technologies and data analysis needs.

In the FRALS framework, we consider the formulation for a single latent variable $t$ with regularization $\lambda_i P_i$ on the weights $u^{(i)}$:

$$\underset{u}{\operatorname{argmin}} \left\{ \sum_i \|X^{(i)}u^{(i)} - t\|_2^2 + \lambda_i P_i(u^{(i)}) \right\} \qquad \text{(III.11)}$$
$$\text{subject to: } t^\top t = 1$$

This problem can be decomposed into three subproblems. The first subproblem for the auxiliary variable $t$:

[9] Though for reasons that will later become clear, we do not reccommend this!

$$\underset{t}{\operatorname{argmin}} \left\{ \sum_i \|X^{(i)}u^{(i)} - t\|_2^2 \right\} \quad \text{subject to: } t^\top t = 1 \tag{III.12}$$

is a standard least squares problem and can be solved in closed form by averaging $X^{(i)}u^{(i)}$ and normalizing i.e. $t = \frac{\sum_i X^{(i)}u^{(i)}}{\|\sum_i X^{(i)}u^{(i)}\|_2}$. As shown earlier this makes $t$ an estimate of the latent variables of a generative CCA model.

The subproblems for the weights $u^{(i)}$:

$$\underset{u^{(i)}}{\operatorname{argmin}} \left\{ \|X^{(i)}u^{(i)} - t\|_2^2 + \lambda_i P_i(u^{(i)}) \right\} \tag{III.13}$$

are regularized least squares problems that can be solved using any suitable regularized least squares solver[10].

In this chapter, we illustrate the power of the FRALS framework by implementing the well-tested Elastic Net solver from the `scikit-learn` package (Pedregosa et al., 2011), where $P_i = \alpha_i \times \text{l1\_ratio}\|u^{(i)}\|_1 + \alpha_i \times (1 - \text{l1\_ratio})\|u^{(i)}\|_2^2$, allowing for independent tuning of shrinkage and sparsity of the weights in both views.

In summary, the FRALS framework is a flexible and user-friendly interface for CCA that allows users to combine scikit-learn compatible regularized least squares solvers to solve regularized CCA problems.

# 4 Experiments

In this section, we outline the methodologies employed in our study of FRALS and related techniques.

## 4.1 The predictive framework for CCA

To evaluate the performance of CCA models, we employ a standard predictive framework. We split the data into training and test sets using a 80:20 split, and use the training set to fit the model. We then use the test set to evaluate the model's performance. Where relevant, pre-processing is performed on the training set and

[10] We could also in principle replace $X^{(i)}u^{(i)}$ with $f(X^{(i)})$ for any function $f$ including kernels, neural networks, or random forests

the same pre-processing is applied to the test set. This is important to avoid data leakage, where information from the test set is used to fit the model.

### 4.1.1 Model Comparisons

In the experiments in this section, we are interested in illustrating the effects of tunable shrinkage and sparsity on the performance and interpretability of CCA models, enabled by the FRALS framework. To this end, we compare the performance of Elastic Net FRALS with other CCA variants, including PCA, PLS, Ridge CCA, Sparse PLS, and Elastic Net CCA.

**Table 4.1:** Employed CCA Variants

| Model | Abbreviation | Hyperparameters | Hyperparameter Range |
|---|---|---|---|
| Principal Component Analysis | PCA | - | - |
| Regularized CCA | RCCA | $c_1, c_2$ | 0-1 (log scaled) |
| FRALS - Elastic | Elastic | $\alpha_1, \alpha_2, \text{l1}_1, \text{l1}_2$ | (1e-5,1e-1), (0-1) |
| Partial Least Squares | PLS | - | - |
| Sparse PLS | SPLS | $\tau_1, \tau_2$ | 0-1[11] (log scaled) |

### 4.1.2 Model Selection

For the models that require hyperparameter tuning, we use a grid search to find the best hyperparameters. Specifically, we use 5-fold cross-validation to evaluate the performance of a model with a given set of hyperparameters on 5 different splits of the training data with non-overlapping validation sets. We optimise for the hyperparameters that give the best average out of sample correlation.

## 4.2 Datasets

For this chapter, we chose the HCP and the ADNI datasets to facilitate comparison with two influential brain-behaviour studies (Stephen M Smith et al., 2015; João M Monteiro et al., 2016) as well as the tutorial paper that this chapter is loosely related to (Mihalik, Chapman, et al., 2022).

### 4.2.1 The Human Connectome Project (HCP)

The HCP offers publicly available resting-state functional MRI (rs-fMRI) and non-imaging measures like demographics, psychometrics, and other behavioral measures. Specifically, we sourced data from 1003 subjects out of the 1200-subject data

**Table 4.2:** HCP Data Parameters

| Parameter | Value |
|---|---|
| Number of samples ($n$) | 1003 |
| Number of features in View 1 ($p$) | 19900 |
| Number of features in View 2 ($q$) | 145 |

**Table 4.3:** ADNI Data Parameters

| Parameter | Value |
|---|---|
| Number of samples ($n$) | 592 |
| Number of features in View 1 ($p$) | 168130 |
| Number of features in View 2 ($q$) | 31 |

release of the HCP. This dataset is constructed using brain connectivity features of the thoroughly processed rs-fMRI data. This processing results in 19,900 brain connectivity variables for every subject. Additionally, there are 145 non-imaging measures employed. Notably, nine confounding variables were regressed out from both data modalities. Each variable was standardized for zero mean and unit variance. More details can be found in Stephen M Smith et al., 2015; Mihalik, Chapman, et al., 2022. We summarize the parameters of the HCP data in table 4.2.

### 4.2.2 The Alzheimer's Disease Neuroimaging Initiative (ADNI)

The ADNI database is found at `adni.loni.usc.edu`. Launched in 2003, ADNI's main objective is to assess the combination of serial MRI, PET (Positron emission tomography), biological markers, and clinical and neuropsychological assessment in tracking the progression of Mild Cognitive Impairment (MCI) and early Alzheimer's disease. For our experiments, we used a subset of 592 unique subjects from the ADNI. The MRI scans underwent a series of processing stages, yielding a grey matter probability map The Mini-Mental State Examination (MMSE) scores were employed to investigate the association with the grey matter maps. Composed of a series of brief tasks, the MMSE evaluates various cognitive domains including memory, attention, language, and visuospatial skills. The MMSE is a widely used test for assessing cognitive impairment. The MMSE scores range from 0 to 30, with lower scores indicating more severe cognitive impairment. We summarize the parameters of the ADNI data in table 4.3.

# 5 Results

## 5.1 HCP Results

Next, we consider the results of applying the various CCA variants to the HCP data. Since the HCP data is high-dimensional, we drop CCA from the analysis since it would produce random results.

### 5.1.1 Out of Sample Correlation

Both Ridge CCA and Elastic Net outperformed PLS and SPLS in terms of holdout correlation captured (Figure III.7). This suggests that tunable L2 regularization is important, even for very high-dimensional data, and that resorting to PLS is suboptimal. On the other hand, while the additional sparsity improved SPLS over PLS (consistent with previous work João M Monteiro et al., 2016), it did not improve the performance of the Elastic Net model over Ridge CCA.

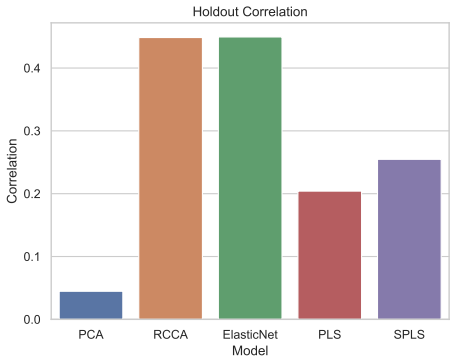



**Figure III.2: HCP:** Out-of-sample canonical correlations for each model.

Nonetheless, the Elastic Net model did produce sparser weights than the Ridge CCA model (Figure **??**) with the Elastic Net model using 241 and 96 non-zero weights for the brain and behaviour views respectively. This is compared to 300 and 145 non-zero weights for the brain and behaviour views respectively for the Ridge CCA model. The SPLS model used even fewer variables with 118 and 56 non-zero weights for the brain and behaviour views respectively. Given that the Elastic Net model can produce the same performance as the Ridge CCA model with fewer variables, we might then be inclined to prefer the Elastic Net model.

**Table 5.1: HCP:** Number of non-zero weights for each model.

| Model | Brain Weights | Behaviour Weights |
|---|---|---|
| PCA | 300 | 145 |
| RCCA | 300 | 145 |
| Elastic Net | 241 | 96 |
| PLS | 300 | 145 |
| SPLS | 118 | 56 |

### 5.1.2 Behaviour Weights

FigureIII.3 plots the top 8 positive and negative non-imaging weights for each model. This is to illustrate some of the effects we have observed in the previous section. PCA finds a mode of variation in the behavioural data that is positively correlated with psychiatric and life function tests and negatively correlated with a number of emotion and personality tests. The RCCA and Elastic Net models find a mode of variation in the behavioural data that is negatively correlated with the Line Orientation test and to a lesser extent smoking and positively correlated with a number of other cognitive tests. The PLS model finds a mode of variation in the behavioural data that is somewhat similar to the 'positive-negative' mode in Stephen M Smith et al. (2015) with a positive correlation with agreeableness, vocabulary tests, and feelings about ones' life and a strong negative correlation with smoking, rule-breaking, and antisocial personality traits. The SPLS mode is similar but selects out the rule-breaking and antisocial personality traits in favour of the vocabulary tests and smoking. This appears consistent with the additional preprocessing steps in Stephen M Smith et al. (2015), which included a top-100 PCA projection of both the brain and behaviour data.

### 5.1.3 Brain Connectivity Weights

In this section, we use two different methods to visualize the brain connectivity weights. The first method is to use chord diagrams to visualize the top 8 positive and negative brain weights for each model. This approach is inspired by the chord diagrams used in Stephen M Smith et al., 2015. The second method is to use surface maps to visualize the brain connectivity weights. This approach has been used by both Ferreira et al., 2022 and Stephen M Smith et al., 2015.

**Chord Diagrams** We grouped the nodes of the connectivity matrix of our data into 7 parcels according to the Yeo 7 network parcellation BT Thomas Yeo et al., 2011.

These are then arranged around the circumference of the chord diagram using the Nichord package(Bogdan et al., 2023). The plots then show the 8 strongest positive and negative weights for each model as 'chords'. The chord diagrams in Figure III.4 show the top 8 positive and negative brain weights for each model.

### 5.1.4 Model Similarity

In this section, we compare the models in terms of their similarity. We can measure the pairwise similarity between two models by comparing their weights and their representations. We can compare the weights by computing the correlation between the weights of the two models and we can compare the representations by computing the correlation between the representations of the two models.

In Figure IV.13, we plot the correlation between the brain and behaviour representations for each model. We can see clearly that both PCA, PLS, and SPLS are all highly correlated in terms of their brain representations, revealing the bias of PLS towards the largest principal components. On the other hand, in the behaviour space, the models are less correlated, with the exception of PLS and SPLS which are highly correlated with one another. There is however still substantial correlation between the PCA and PLS models. The very low correlation between the Ridge CCA and Elastic Net models with the PCA model is evidence that there are stronger correlations outside of the first principal components.

In Figure III.6, we similarly plot the correlation between the brain and behaviour weights for each model. The story is similar, albeit with marginally lower correlations between the PLS and PCA-based models. Finally, in the weights space, the Ridge CCA and ElasticNet models are even less correlated with the PCA model.

**Table 5.2: ADNI:** Number of non-zero weights for each model.

| Model | Brain Weights | Behaviour Weights |
|---|---|---|
| PCA | 168130 | 31 |
| RCCA | 168130 | 31 |
| Elastic Net | 59617 | 17 |
| PLS | 168130 | 31 |
| SPLS | 74995 | 10 |

## 5.2 ADNI Results

We now turn to the ADNI data.

### 5.2.1 Out of Sample Correlation

In this experiment, the Elastic Net model outperformed all other models in terms of out-of-sample correlation (Figure III.7). The RCCA model also outperformed the PLS and SPLS models while SPLS outperformed PLS. Suprisingly, PCA performed almost as well as PLS. This suggests that there is value in both tunable shrinkage and sparsity in this dataset. It also reveals that the correlated signal between the brain structure and behavioural data is relatively much stronger than in the HCP data.

### 5.2.2 Sparsity of Weights

Table 5.2 once again shows the number of non-zero weights for each model. We can see that tuned SPLS and Elastic Net once again identify sparse weights. In this case, the difference in performance is more convincing and suggests that this sparsity is less spuriously induced than for the HCP data. This is supported by the fact that Elastic Net and SPLS models find a similar level of sparsity in the brain weights. On the other hand SPLS finds a much sparser set of behavioural weights.

### 5.2.3 Behaviour Weights

As for the HCP data, Figure III.8 plots the top 8 positive and negative non-imaging weights for each model. Some of the identified behavioural weights including a number of orientation tests are similar across all of the models, including even PCA. This is indicative of the strong shared signal between the behavioural data and the brain structure data. SPLS and Elastic Net both hone in on the orientation and recall tests in the weight space. The RCCA and Elastic Net models are suprisingly

different in the weight space, with the RCCA weights on a couple of attention and calculation tests in addition to the ubiquitous orientation and recall tests.

#### 5.2.4 Brain Structure Weights

We plot the weights as a mosaic plot with 3 slices in each direction in Figure **??**. Previous work using SPLS with the ADNI dataset identified the same striking pattern of weights with the model strikingly selecting the hippocampal weightsJoão M Monteiro et al., 2016. The Elastic Net has a less visually appealing selection of weights, with a honeycomb pattern near the edges of the brain and likewise for RCCA. It is noticeable that PCA, PLS and SPLS both weights in the same direction whereas RCCA and Elastic Net weight different regions with opposite signs.

#### 5.2.5 Model Similarity

In this section, we once again compare the models in terms of their similarity. In Figure III.10, we can see that all of the models are highly correlated in terms of their behaviour representations. The brain representations are less correlated, but once again PCA, PLS, and SPLS are highly correlated with one another and less correlated with the Ridge CCA and Elastic Net models.

Suprisingly, in Figure III.11, we can see that the weights in both views are less correlated. This is particularly true for the brain weights where PCA exhibits a very low correlation with Ridge CCA and Elastic Net.

### 5.3 Timings

Finally, we consider the timings of the different models. This is an important metric because one of the main reasons for the popularity of SPLS is its speed and therefore convenience. Figure III.12 shows an estimate of the time taken to fit each model for each complete training dataset over 10 runs. We can see clearly that the Elastic Net is much slower than the other models when using the high dimensional ADNI data. Despite also being an iterative algorithm, the SPLS model is much faster than the Elastic Net and only slightly slower than the PLS and RCCA models which call optimised solvers in C. Since PLS and RCCA both use PCA preprocessing for efficiency, it is unsuprising that PCA is the fastest model.

# 6 Discussion and Limitations

In this section, we discuss the implications of our findings as well as the limitations of our study and the proposed FRALS method, some of which we address in later chapters of this thesis.

## 6.1 Discussion

**Ridge CCA is typically much better than PLS across datasets:** Our results show that Ridge CCA is typically much better than PLS across datasets. Much like regularised regression, it is unusual to need to use maximal ridge regularization even in high dimensions. This means that while PLS might be more stable for a given dataset, it is not necessarily more stable across random samples from the same population.

**FRALS is a useful tool for implementing Elastic Net CCA:** Our results show that FRALS is a useful tool for implementing Elastic Net CCA.

## 6.2 FRALS Limitations

While FRALS offers promising performance in terms of out-of-sample correlation, it does come with significant drawbacks, the most noteworthy being its computational inefficiency. Below, we outline the primary factors contributing to the slow speed of FRALS and provide some insights into the computational bottlenecks.

**Changing Regression Targets** Adding to the computational burden is the fact that the regression targets, i.e., the projections of the other view, are not static but change dynamically throughout the algorithm's run. Each update to the least squares solution consequently alters the global objective, leading to a constantly shifting landscape that the algorithm needs to navigate. This also leads to a significant amount of redundant computation, as the algorithm needs to recompute the least squares solution for each view at each iteration.

**Computational Time** The primary bottleneck in FRALS is the computation of the least squares solution. For each iteration of the algorithm, we need to compute the least squares solution for each view. This is a computationally expensive operation. It is the primary factor contributing to the slow speed of FRALS (depending on

the experiment around 10 times slower than Ridge CCA). In the next chapter, we address this bottleneck by introducing a novel algorithm based on gradient descent.

# 7 Conclusion

We have also shown that FRALS can be a useful tool for brain-behaviour studies, but that it is computationally expensive.

In the next chapter,

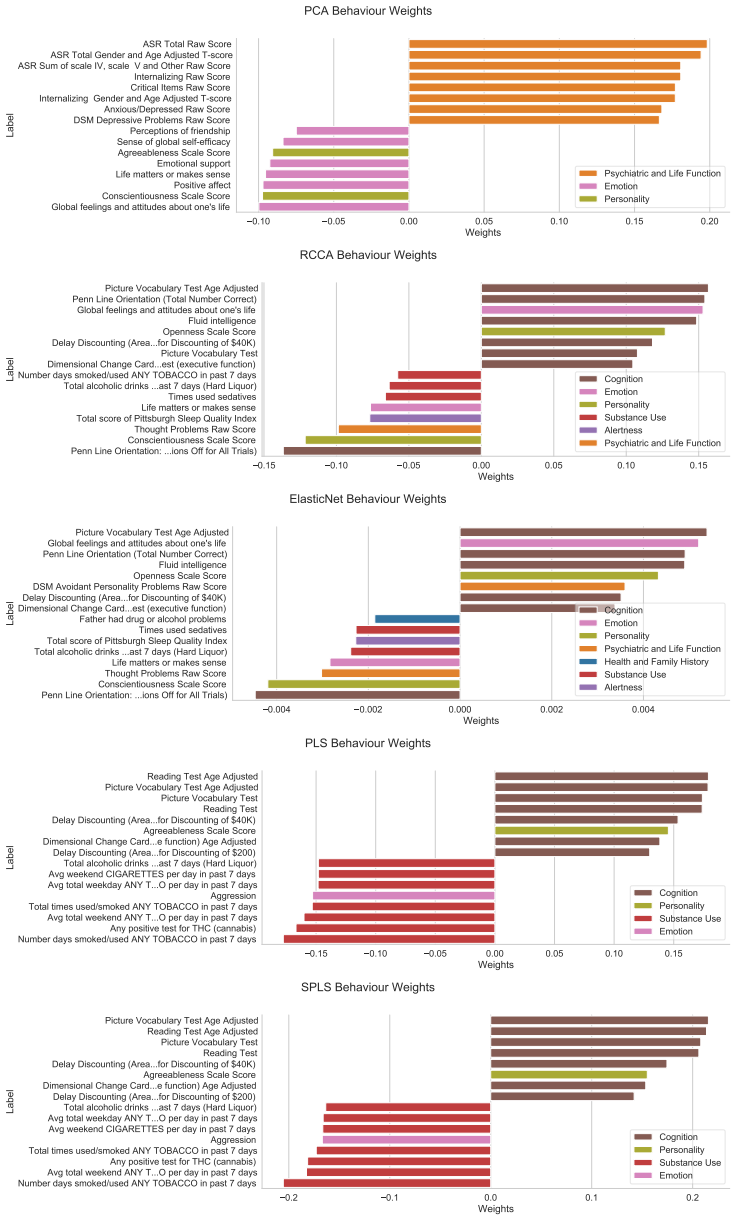



**Figure III.3: HCP:** Top 8 positive and negative non-imaging weights for each model

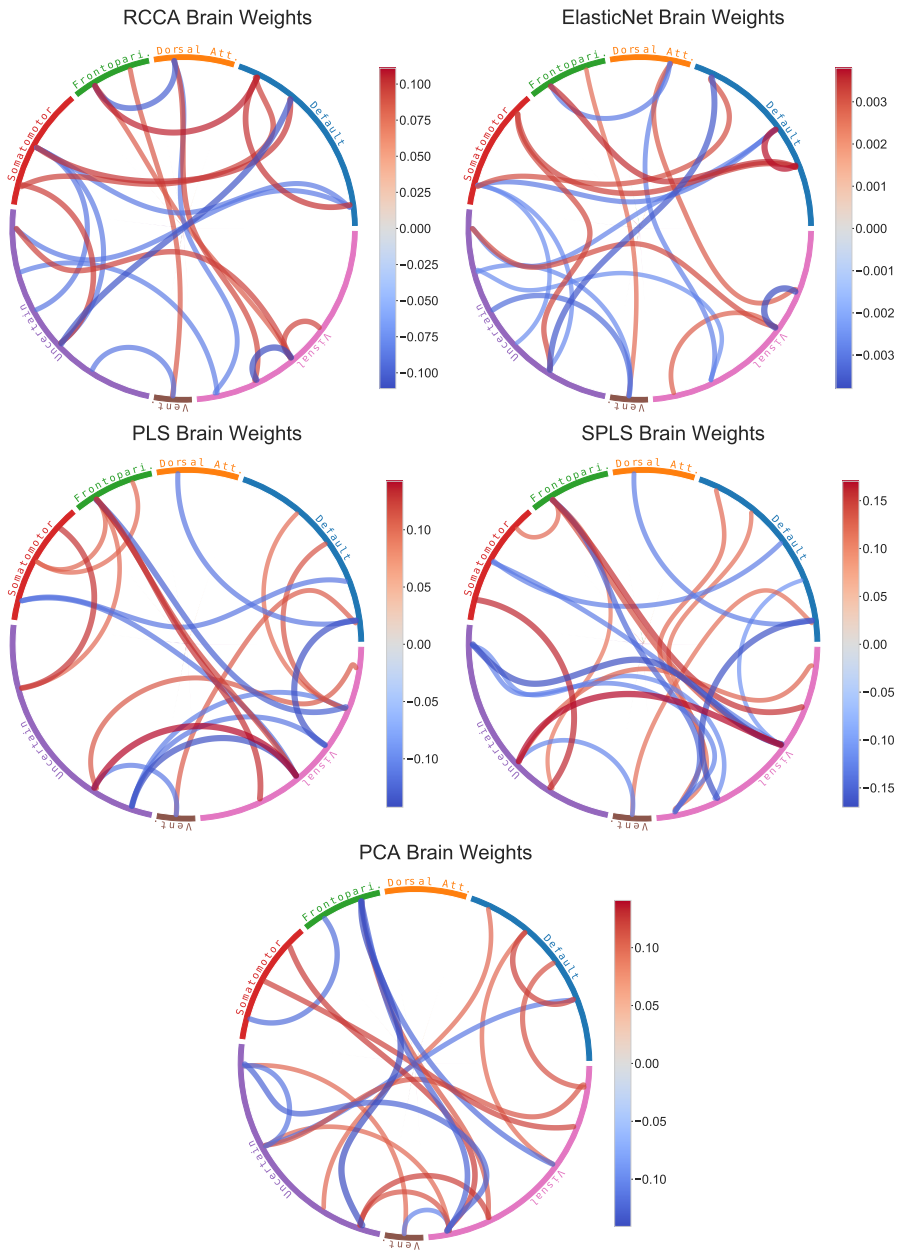



**Figure III.4: HCP:** Chord diagrams of the top 8 positive and negative brain weights for each model.

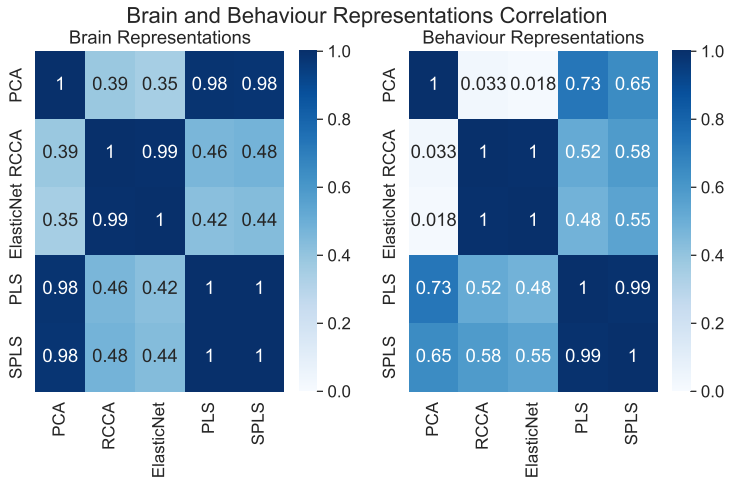



**Figure III.5: HCP:** Correlation between the brain and behaviour representations for each model.

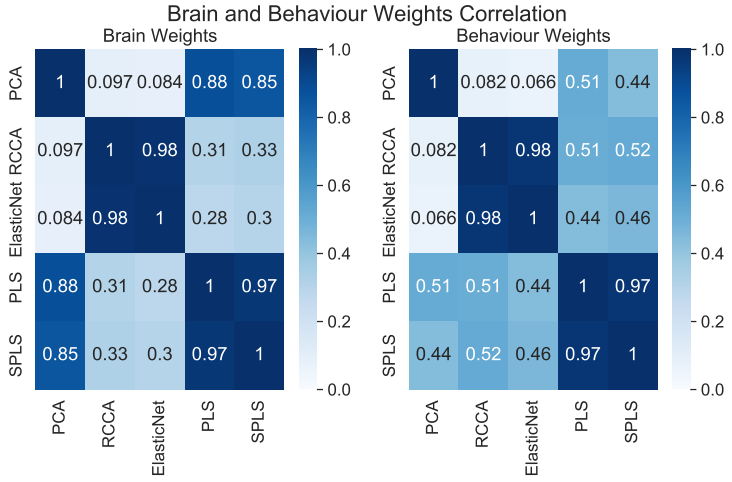



**Figure III.6: HCP:** Correlation between the brain and behaviour weights for each model.

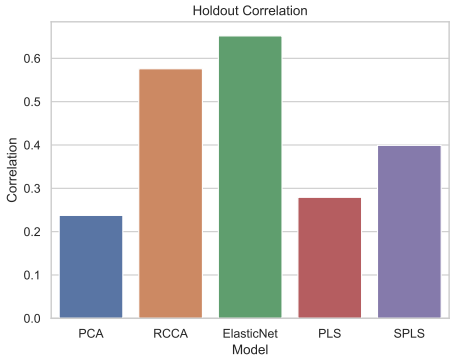



**Figure III.7: ADNI:** Out-of-sample canonical correlations for each model.

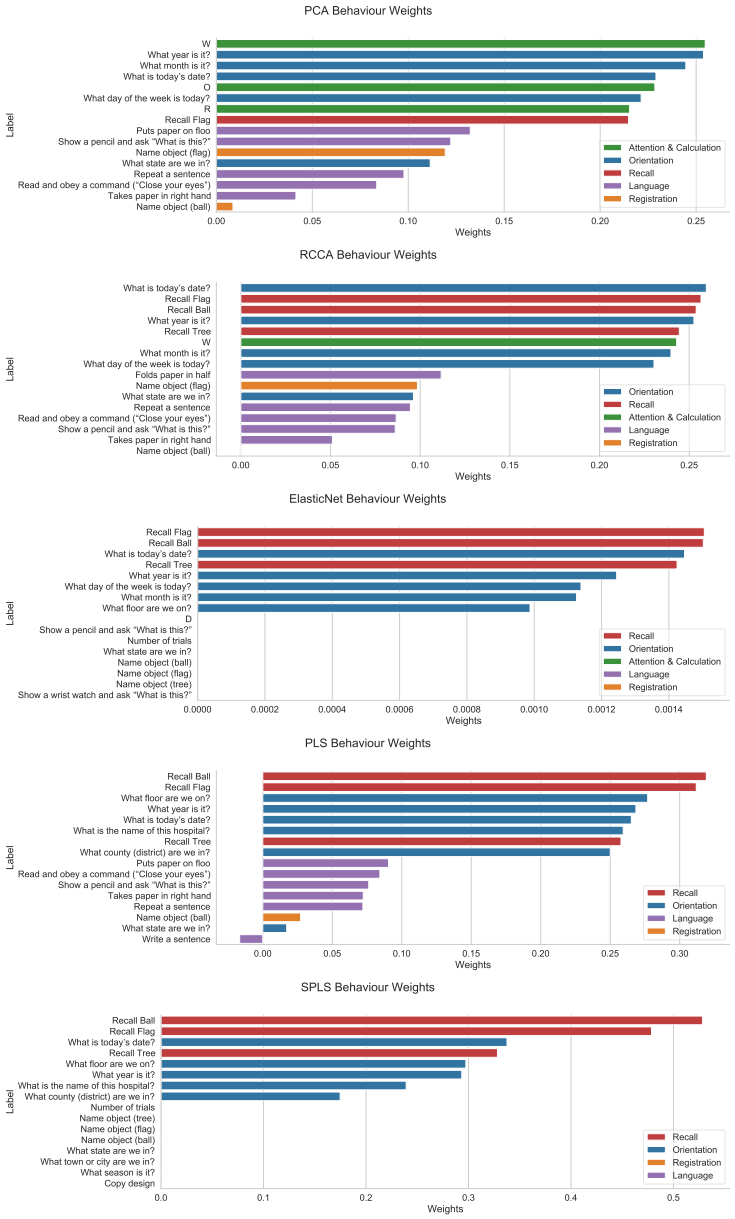



**Figure III.8: ADNI:** Bar plots of the behaviour weights for each model.

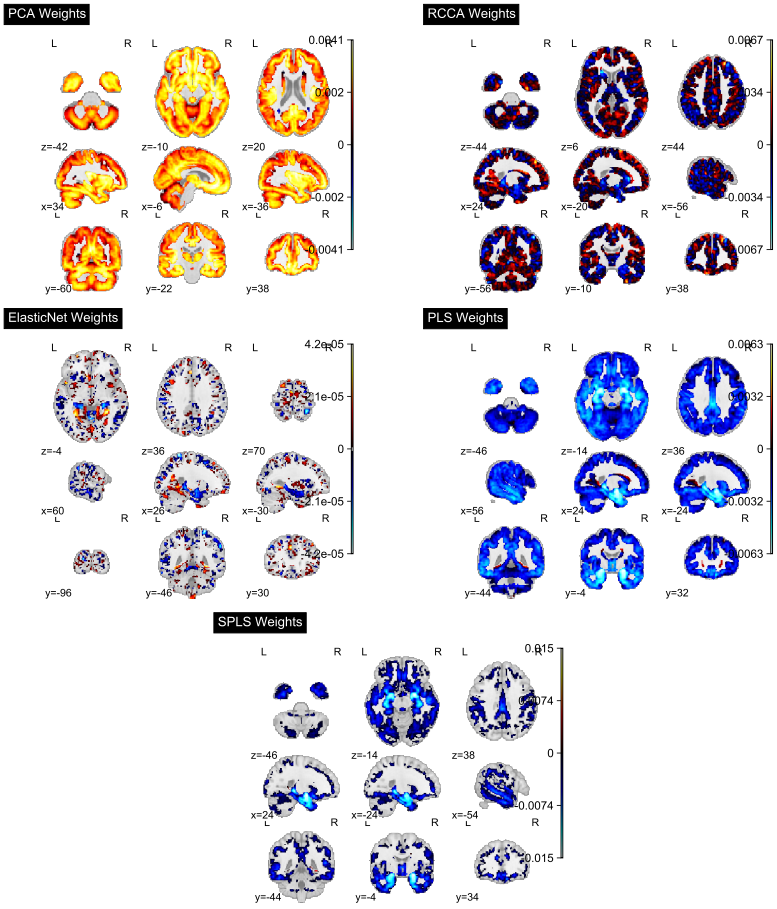



**Figure III.9: ADNI:** Statistical maps of brain structure weights for each model.

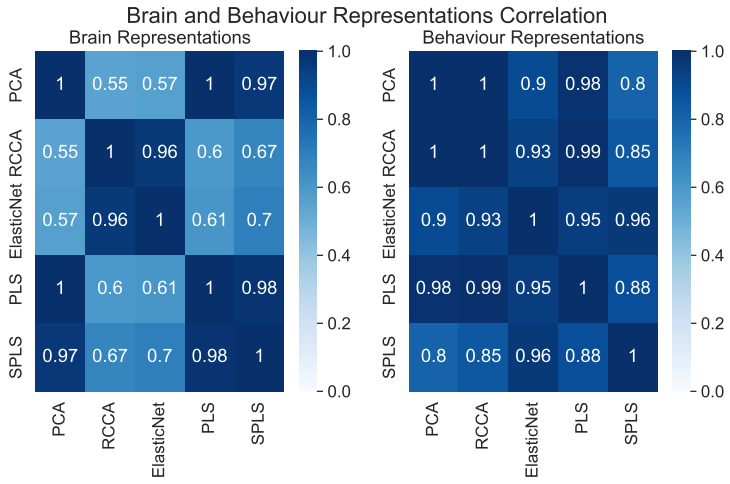



**Figure III.10: ADNI:** Correlation between the brain and behaviour representations for each model.

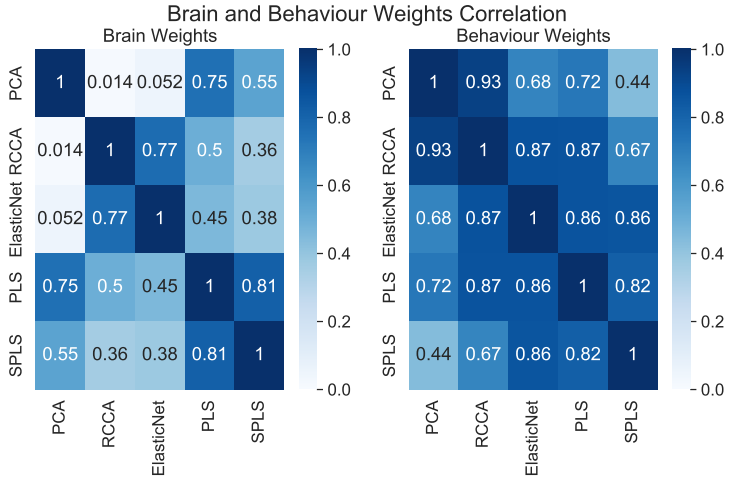



**Figure III.11: ADNI:** Correlation between the brain and behaviour weights for each model.

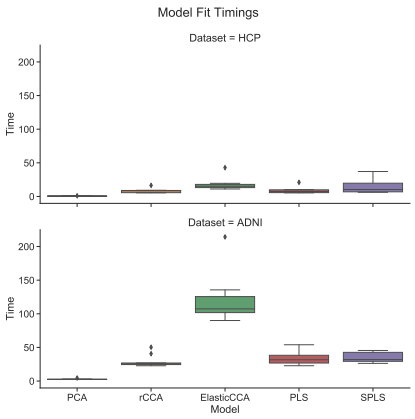



**Figure III.12:** Time taken to fit each model.

# Chapter IV

# On Using Loadings to Interpet CCA Models

Contents

# Preface

This chapter is based on observations made across a number of projects. While the work itself is unpublished, it has influenced , most notably by the use of loadings rather than weights for model interpretation in **<empty citation>** where I contributed parts of the appendix and encouraged the use of loadings for model interpretation. The simulated data generation methods were also used to generate simulated data in Mihalik, Chapman, et al. (2022). This chapter makes the theoretical case for using loadings rather than weights for model interpretation in CCA and PLS models. It also presents a number of experiments demonstrating the relationship between loadings and weights in CCA models.

# 1 Introduction

In practical applications, (CCA) has two aspects: estimating latent variables associated with different views, and exploring the expression of these latent variables in each view, ideally providing insight into biomedical mechanisms. This dual focus is similar to the distinction in machine learning between discriminative and generative approaches.

Discriminative approaches in Canonical Correlation Analysis (CCA) prioritize estimating latent variables from observed data, typically employing 'weights' as parameters. Conversely, generative approaches in CCA seek to understand the data generation process. This perspective is illustrated by the 'loadings' in our model, which represent the parameters transforming the latent variable into observed data.

In this context, the generative or probabilistic CCA is parameterized by the loadings, which play a crucial role in understanding how the latent variable (severity) influences the observed data (brain and behaviour). On the other hand, the discriminative or machine learning approach to CCA is characterized by weights, focusing on how observations can be used to estimate the latent variables.

The ideal CCA model aims to accurately estimate latent variables while maintaining the interpretability of the model, particularly through the loadings. This allows for a deeper understanding of the relationship between different views (brain

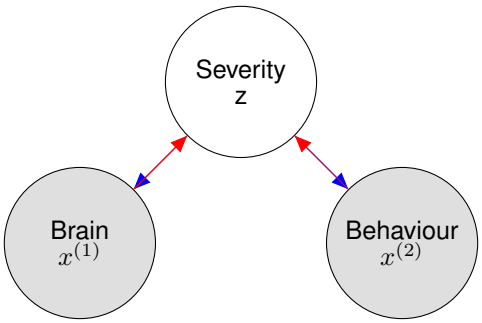



**Figure IV.1:** ***Forward and Backward Multiview Models:*** The generative approach to CCA focuses on the forward model from latent variables to observed data, while the machine learning approach focuses on the backward model from observed data to latent variables.

and behaviour) through the data generation process. The significance of loadings for interpretability, particularly in the forward model, as opposed to weights in the backward model, was highlighted in the context of linear models like SVM and Lasso (Haufe et al., 2014). Our work in this chapter contributes to this discourse by demonstrating a similar interpretative advantage of loadings over weights in CCA models.

In this chapter, we reexamine the relationship between these discriminative and generative approaches to CCA approaches using simulated data. We categorize the main simulated data generation methods for CCA into two groups: explicit latent variable models and implicit latent variable models; in particular by reformulating the joint covariance matrix perspective from Suo et al. (2017) and M. Chen et al. (2013) as an implicit latent variable model. With this context, we reconsider the role of regularization in CCA models and give recommendations for the use of loadings over weights for model interpretation.

# 2 Background: Unifying Generative Perspectives on CCA

## 2.1 Probabilistic CCA and GFA (Explicit Latent Variable Models)

Let's reconsider the graphical model depicted in Figure IV.1. Now we further assume that the brain is generated via a linear model with added noise, while the behavioural modality similarly arises from a linear model with noise. Once again they are conditionally independent, given the latent variable.

The distributions of the two views are given by:

$$z \sim \mathcal{N}(0, I) \tag{IV.1}$$

$$x^{(i)} \sim \mathcal{N}(W^{(i)}z + \mu^{(i)}, \Psi^{(i)}) \tag{IV.2}$$

Where $z$ represents the latent variable (disease severity), $x^{(i)}$ represents the $i^{\text{th}}$ view, $W^{(i)}$ represents the model loadings, $\mu^{(i)}$ represents the mean, and $\Psi^{(i)}$ represents the noise covariance matrix for the $i^{\text{th}}$ view. Notice that if it were not for the view-specific noise, the two views would be perfectly correlated subject to a linear transformation.

Bach and Jordan (2005) showed that the maximum likelihood solution for this model is equivalent to the solution of the CCA problem in the sense that the loadings are the same as the CCA weights multiplied by the covariance:

$$\hat{W}^{(i)} = \Sigma_{ii}\hat{U}^{(i)}R \tag{IV.3}$$

Where $R$ is an arbitrary rotation matrix and $\hat{U}^{(i)}$ is the matrix of CCA weights for the $i$th view. This implies that for invertible covariance matrices, we can access the 'true' CCA weights associated with the top-k subspace by multiplying the loadings by the inverse of the covariance matrix:

$$\hat{U}^{(i)}R = \Sigma_{ii}^{-1}\hat{W}^{(i)} \tag{IV.4}$$

In practice we do not have access to the covariance matrices $\Sigma_{ii}$, so we must estimate them from the data using the sample covariance matrices $\hat{\Sigma}_{ii}$.

**Notice that for Identity covariance matrices, the CCA weights are the same as the loadings.** Otherwise, there is a linear transformation between the two. For singular covariance matrices, the CCA weights are not uniquely defined.

Moreover, the mean of the posterior distribution of the latent variables is proportional to the mean of the CCA scores (Klami, Virtanen, and Kaski, 2013). Group Factor Analysis (GFA) is a closely related model that assumes diagonal covariance in $\Psi^{(i)}$:

$$z \sim \mathcal{N}(0, I) \tag{IV.5}$$

$$x^{(i)} \sim \mathcal{N}(W^{(i)}z, \sigma^{(i)}I) \tag{IV.6}$$

An interesting feature of the GFA model is that as the noise level approaches zero, the marginal distribution of the views is the same as the probabilistic PCA model for each view (Tipping and Bishop, 1999). This suggests that for small noise levels, we should in fact be able to recover much of the mutual information between the views by using PCA on each view separately. For this reason, we will use and recommend PCA as a baseline in our later experiments. Because the diagonal covariance assumption makes inference computationally cheaper, this line of work has been able to extend to incorporate sparsity on the loadings (Virtanen, Klami, and Kaski, 2011) as well as missing data (Ferreira et al., 2022).

By marginalizing out the latent variables of the generative CCA and GFA models, we can write down the joint distribution of the two views:

$$\begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu^{(1)} \\ \mu^{(2)} \end{bmatrix}, \begin{bmatrix} W^{(1)}W^{(1)\,T} + \Psi_1 & W^{(1)}W^{(2)\,T} \\ W^{(2)}W^{(1)\,T} & W^{(2)}W^{(2)\,T} + \Psi_2 \end{bmatrix} \right) \tag{IV.7}$$

Importantly, this shows us that the true covariance in each view is a function of the loadings and the noise covariance matrix. Specifically, the covariance matrix of the $i^{\text{th}}$ view is given by:

$$\Sigma_{ii} = W^{(i)}W^{(i)\,T} + \Psi_i \tag{IV.8}$$

While these generative models are provide a clear interpretation of the data generation process and possible biological processes, their application in practice is limited compared to classical CCA. This is primarily due to their computational

intensity and the need for a careful selection of priors. Moreover, while these models can generate data with sparse loadings, generating data with sparse weights is challenging due to the dependence of CCA weights on the covariance matrices of the views.

## 2.2 A Joint Covariance Matrix Perspective (Implicit Latent Variable Model)

In contrast to the explicit latent variable models discussed earlier, the joint covariance matrix perspective offers an implicit approach to understanding the data generation process. This method focuses on the covariance matrices of the views, rather than directly modeling latent variables. A key advantage of this perspective, particularly noted in the sparse CCA literature, is its ability to generate data with sparse weights and or known canonical correlations. This is achieved by constructing the joint covariance matrix of the two views as follows:

$$\begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \tag{IV.9}$$

Where $\Sigma_{11}$ and $\Sigma_{22}$ are the within-view covariance matrices and $\Sigma_{12}$ and $\Sigma_{21}$ are the between-view covariance matrices.

This has the advantage of allowing us to control the within-view covariance and therefore test the methods under specific conditions. The process was first described by Chen (M. Chen et al., 2013) and further explained by (Suo et al., 2017) and has been the basis behind findings in **Matkovi**; Helmer et al. (2020).

We can control the true signal by setting the active variables and correlations in the between-view covariance matrices $\Sigma_{12}$ and $\Sigma_{21}$. Specifically we construct the between-view covariance matrices as follows:

$$\Sigma_{12} = \sum_{k=1}^{K} \rho_k \Sigma_{11} u_k^{(1)} u_k^{(2)\,T} \Sigma_{22} \tag{IV.10}$$

Where $\rho_k$ is the $k^{\text{th}}$ canonical correlation and $u_k^{(i)}$ is the $k^{\text{th}}$ column of the matrix of weights $U^{(i)}$.

We can still access the true loadings of the implied latent variable model by using the relationship in IV.3 and multiplying the weights $u^{(i)}$ by the within-view covariance

matrix $\Sigma_{ii}$.

## 2.3 Summary of Data Generation Methods

**Comparison of Joint Covariance Matrices** To understand the distinct approaches of each data generation method, we present a comparison of their covariance structures. This comparison highlights the differences in how these methods model the relationship within and between views.

**Table 2.1:** Covariance Structures in Data Generation Methods

| | **Method** | **Within-view Covariance $\boldsymbol{\Sigma_{ii}}$** | **Between-view Covariance $\boldsymbol{\Sigma_{12}}$** |
|---|---|---|---|
| Explicit | Probabilistic CCA | $W^{(i)}W^{(i)\top} + \Psi_i$ | $W^{(1)}W^{(2)\top}$ |
| | GFA | $W^{(1)}W^{(1)\top} + \sigma^{(1)}I$ | $W^{(1)}W^{(2)\top}$ |
| Implicit | Joint Covariance | $\Sigma_{ii}$ | $\sum_{k=1}^{K} \rho_k \Sigma_{11} u_k^{(1)} u_k^{(2)\top} \Sigma_{22}$ |
| | Joint Covariance (Identity) | $I$ | $\sum_{k=1}^{K} \rho_k u_k^{(1)} u_k^{(2)\top}$ |

**Comparison of True Weights and Loadings** We summarize the relationship between the weights and loadings in each data generation method, distinguishing between population and sample cases. This distinction is crucial, especially in scenarios where the population covariance matrix $\Sigma$ is identity, but the sample covariance matrix $\hat{\Sigma}$ is only an approximation.

**Table 2.2:** Relationship Between Weights and Loadings in Population and Sample Cases

| | Method | Case | Weights | Loadings |
|---|---|---|---|---|
| Explicit | Probabilistic CCA | Population | $(W^{(i)}W^{(i)\top} + \Psi_i)^{-1}W^{(i)}$ | $W^{(i)}$ |
| | | Sample | $\hat{\Sigma_{ii}}^{-1}W^{(i)}$ | $W^{(i)}$ |
| | GFA | Population | $(W^{(i)}W^{(i)\top} + I)^{-1}W^{(i)}$ | $W^{(i)}$ |
| | | Sample | $\hat{\Sigma_{ii}}^{-1}W^{(i)}$ | $W^{(i)}$ |
| Implicit | Joint Covariance (Non-Identity) | Population | $U^{(i)}$ | $\Sigma_{ii}U^{(i)}$ |
| | | Sample | $U^{(i)}$ | $\hat{\Sigma_{ii}}\hat{U^{(i)}}$ |
| | Joint Covariance (Identity) | Population | $U^{(i)}$ | $U^{(i)}$ |
| | | Sample | $U^{(i)}$ | $\hat{\Sigma_{ii}}\hat{U^{(i)}}$ |

# 3 A Mathematical Argument for the use of Loadings not Weights for Interpretability

In this section, we make a mathematical argument for the use of loadings over weights for the interpretation of CCA models. In particular, we show that the loadings are invariant to columnwise transformations of the data matrix, while the weights are not.

CCA can be solved in the principal component space. Consider the singular value decomposition (SVD) of the data matrices:

$$X^{(i)} = U^{(i)}\Sigma^{(i)}V^{i\top} \tag{IV.11}$$

Here, $U^{(i)}$ and $V^{(i)}$ are the left and right singular vectors of $X^{(i)}$ respectively, and $\Sigma^{(i)}$ is a diagonal matrix of singular values. The intuition behind this decomposition is that we are representing the data matrix in terms of its fundamental components: the directions of maximum variance (captured by $V^{(i)}$), the scale of these directions (captured by $\Sigma^{(i)}$), and the projections of the data onto these directions (captured by $U^{(i)}$). $U^{(i)}$ are the principal components of $X^{(i)}$.

Substituting Equation IV.11 into the CCA objective function, we have:

$$\max_{U^{(1)},u^{(2)}} \mathrm{Corr}(X^{(1)}U^{(1)}, X^{(2)}u^{(2)}) = \max_{U^{(1)},u^{(2)}} \mathrm{Corr}(U^{(1)}\Sigma^{(1)}V^{1\top}U^{(1)}, U^{(2)}\Sigma^{(2)}V^{2\top}u^{(2)}) \tag{IV.12}$$

Reparameterizing the weights as $v^{(i)} = \Sigma^{(i)} V^{i\top} u^{(i)}$, we obtain:

$$\max_{v^{(1)}, v^{(2)}} \mathrm{Corr}(U^{(1)} v^{(1)}, U^{(2)} v^{(2)}) \tag{IV.13}$$

This reparameterization simplifies the optimization problem in two ways. Firstly, if the data matrices are low rank (which is guarenteed if the number of samples is less than the number of features), then the matrix of principal components $U^{(i)}$ is lower dimensional than the data matrix $X^{(i)}$, reducing the number of parameters in the optimization problem. Secondly, the reparameterization ensures that $v^{(1)\,T} U^{(1)\,T} U^{(1)} v^{(1)} = v^{(1)\,T} v^{(1)}$, making the constraints independent of the data. We can therefore solve the CCA problem by solving the simpler PLS problem in the principal component space. , which is computationally more feasible but also gives us a convenient way to understand how the weights and loadings of CCA models change under different transformations of the data.

**Definition:** *Loadings* are defined using the reparameterized weights as follows:

$$w_j^{(i)} = \mathrm{Corr}(X_j^{(i)}, U^{(i)} v^{(i)}) = \frac{\mathrm{Cov}(X_j^{(i)}, U^{(i)} v^{(i)})}{\sqrt{\mathrm{Var}(X_j^{(i)})} \sqrt{\mathrm{Var}(U^{(i)} v^{(i)})}} \tag{IV.14}$$

By convention, and without loss of generality, we standardize the latent variables to have unit variance so that:

$$w_j^{(i)} = \frac{\mathrm{Cov}(X_j^{(i)}, U^{(i)} v^{(i)})}{\sqrt{\mathrm{Var}(X_j^{(i)})}} \tag{IV.15}$$

Intuitively, loadings measure how much each original feature contributes to the latent variables, providing insight into the structure of the data.

### 3.0.1 Invariance to Scale

First, we show that the loadings are invariant to column-wise scaling of the data matrix whereas the weights are not.

**Lemma 3.1.** *Scaling the columns of the data matrix does not affect the left singular vectors $U^{(i)}$.*

*Proof.* Scale the columns of the data with a matrix $C$:

$$C = \begin{pmatrix} c_{11} & 0 & \cdots & 0 \\ 0 & c_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{nn} \end{pmatrix} \tag{IV.16}$$

where $c_{ii}$ represents the scaling factor for the $i$-th column of the data matrix. For columns that are not scaled, $c_{ii} = 1$. This means that the corresponding column remains unchanged.

Since $C$ is diagonal it can be represented by a diagonal matrix $S = C$ and an orthogonal matrix (the identity matrix) $R$. The transformed dataset is therefore $X^{(1')} = X^{(1)}C$:

$$X^{(1')} = U^{(i)}\Sigma^{(i)}V^{i\top}C = U^{(i)}(\Sigma^{(i)}S^{(i)})({}^{(i)}V^{i\top}I) = \tag{IV.17}$$

which makes clear that the left and right singular vectors $U^{(i)}$ and $V^{(i)}$ right remain unchanged. Therefore, the modified equation can be represented as:

$$X^{(1')} = U^{i}\Sigma^{(i')}(V^{(i)})^{T} \tag{IV.18}$$

where $\Sigma^{(i')} = \Sigma^{(i)}S^{(i)}$. □

**Intuition** Scaling the data is like changing the units of measurement. It stretches or compresses the data but does not change the relationships between the samples.

Noting that the CCA optimisation problem remains the same as in Equation IV.13, we can now show that the weights are not invariant to scaling of the data matrix but the loadings are.

**Weights change** From the earlier reparameterization, and given that $v^{(i')} = v^{(i)}$, the weights post-scaling are:

$$u^{(i')} = V^{(i)}(\Sigma^{(i')})^{-1}v^{(i)} = V^{(i)}(\Sigma^{(i')})^{-1}v^{(i)} = {C^{(i)}}^{-1}u^{(i)} \tag{IV.19}$$

which are the original weights scaled by the inverse of the scaling matrix $C$. This means that the weights are not invariant to scaling of the data matrix. Furthermore

it means we can set the weights to arbitrary values by scaling the data matrix. While we can build pipelines with standardized data, there is no a priori reason to do so.

**Loadings are invariant** Since loadings are correlations between the original features and the latent variables, they are invariant to scaling of the data. This follows from the definition of correlation and the unchanged latent variables:

$$w_j^{(i)} = \text{Corr}(X_j^{(i')}, U^{(i)}v^{(i)}) = \text{Corr}(c_{jj}X_j^{(i)}, U^{(i)}v^{(i)}) = \text{Corr}(X_j^{(i)}, U^{(i)}v^{(i)}) \tag{IV.20}$$

**Intuition** The loadings remain the same because scaling the data does not change the relative contributions of each feature to the latent variables.

### 3.0.2 Invariance to Repeated Linear Combinations of Columns

We can also prove a more general result that the loadings are invariant to repeated linear combinations of columns of the data matrix. This is not as contrived as it sounds, since we often need to decide which features to include or exclude in a model, and when we work with highly correlated variables like survey questions, we may choose to use summary scores instead of individual questions.

**Lemma 3.2.** *Adding linear combinations of columns to the data matrix does not affect the left singular vectors* $U^{(i)}$.

*Proof.* Now, consider adding columns that are linear combinations of existing columns in $X^{(i)}$ to form $X^{(i'')}$. We can represent this using a transformation matrix $A$ such that $X^{(i'')} = X^{(i)}A$:

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 & a_{11} & a_{12} & \cdots & a_{1m} \\ 0 & 1 & \cdots & 0 & a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \tag{IV.21}$$

where $a_{ij}$ represents the weight of the $j$-th column in the $i$-th linear combination. The key is that we can still represent the transformed dataset as a product of the original left singular vectors $U^{(i)}$ and a new diagonal matrix $\Sigma^{(i'')}$ and right singular vectors $V^{(i'')}$.

$$X^{(i'')} = U^{(i)}(\Sigma^{(i)}V^{i\top}A) = U^{(i)}(\Sigma^{(i')}V^{(i')\,T}) \tag{IV.22}$$

Where we know that the transformation is rank preserving because the first $n$ columns of $A$ are the identity matrix. The left singular vectors $U^{(i'')}$ therefore remain the same as $U^{(i)}$. □

**Weights are Underdetermined** The weights $u^{(i'')}$ are underdetermined in the transformed space due to the added linear dependencies in the columns. The specific weights will depend on the SVD computation approach.

$$u^{(i'')} = V^{(i)}(\Sigma^{(i'')})^{-1}v^{(i)} \tag{IV.23}$$

**Intuition** In the extreme case, if we have two identical columns in the data matrix, then we can use any weights we like for these columns provided that their sum is the same.

**Loadings Remain Invariant** The loadings, as before, remain unchanged because the original columns are unchanged and the latent variables are unchanged.

### 3.1 Summary

We have shown that the loadings are invariant to columnwise transformations of the data matrix, while the weights are not. This is a key advantage of loadings over weights for the interpretation of CCA models. We might even be inclined to state this even more strongly: **the loadings are the true parameters of the model, while the weights are not**.

## 4 Experiments

Having given a mathematical argument for the use of loadings over weights for the interpretation of CCA models, we now present a number of experiments demonstrating the relationship between loadings and weights in CCA models.

In this chapter, we use simulated data to illustrate properties of CCA models and then revisit the results of chapter **??** to demonstrate the relationship between weights and loadings in real-world datasets.

## 4.1 Simulated Data

We used 6 simulated datasets, generated using the methods described in section **??**. Simulated data was characterized by distinct properties, including sparse weights and/or loadings. In our experiments, both low-dimensional (10 features per view) and high-dimensional (100 features per view) scenarios were considered. We utilized 50 training and 50 test samples for each of 10 independent random draws from the data generation process, as detailed in table 4.1.

**Implicit Latent Variable Models and Sparse Weights:** In line with the Joint Covariance method described in section 2.2, we generated data under two scenarios:

- Using identity covariance matrices, aligning with the 'Implicit' latent variable model (Joint Covariance (Identity)) where true weights are equivalent to true loadings.
- Employing non-identity covariance matrices, consistent with the 'Implicit' latent variable model (Joint Covariance (Non-Identity)) where true weights differ from true loadings, usually resulting in non-sparse weights.

The true loadings are defined as the product of the true weights and the true population within-view covariance matrix. For each model, we estimated model loadings using the pseudo-inverse[1] of the sample covariance matrix.

**Explicit Latent Variable Models and Sparse Loadings:** We generated data with sparse loadings using the Probabilistic CCA and GFA models, as outlined in section 2.1. The signal-to-noise ratio was calibrated to mirror the correlations observed in the Joint Covariance method, with the sum of the signal's eigenvalues being twice that of the noise. The true weights were defined as the product of the true loadings and the inverse of the true population within-view covariance matrix. For each model, we once again estimated model loadings using the pseudo-inverse of the sample covariance matrix. Note that because we multiply model weights by the sample covariance matrix to estimate the loadings, the estimated loadings are sometimes not sparse even when both the model weights are sparse and the true covariance matrix is identity.

[1] Defined as $A^+ = (A^\top A)^{-1} A^\top$, it inverts the closest matrix to $A$ in a least squares sense

**Table 4.1:** Simulated Data Parameters

| Parameter | Value |
|---|---|
| Number of samples ($n$) | 50 train, 50 test |
| Number of features in View 1 ($p$) | 10 (low-dimensional), 100 (high-dimensional) |
| Number of features in View 2 ($q$) | 10 (low-dimensional), 100 (high-dimensional) |
| True Latent dimensions | 1 |
| True Feature Density in View 1 | 0.5 |
| True Feature Density in View 2 | 0.5 |

# 5 Simulated Data Results

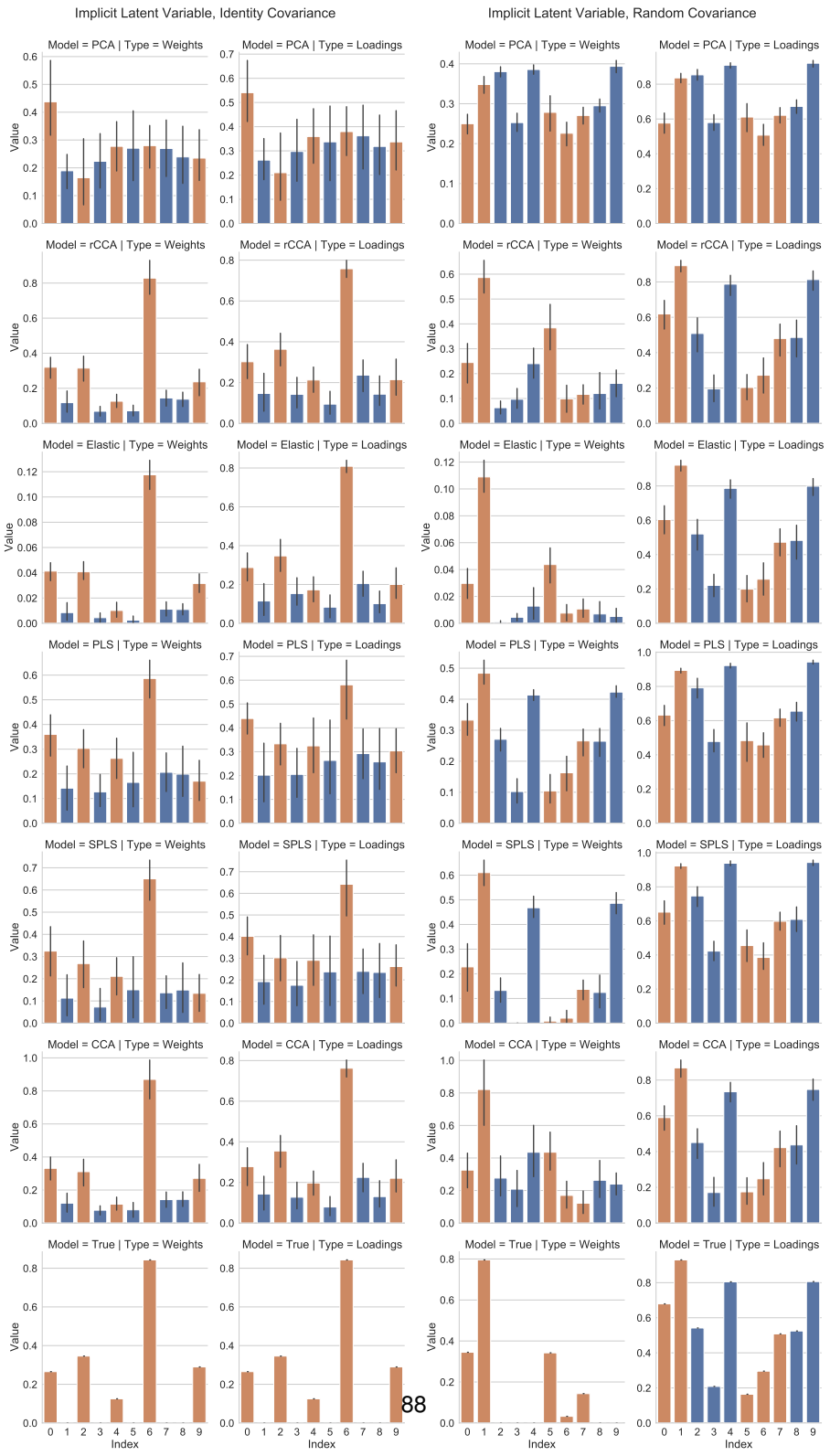
In this section, we present the results of our experiments. We begin with the results of the low and high-dimensional simulated data experiments, followed by the results of the HCP and ADNI data.

## 5.1 Low-Dimensional Data

### 5.1.1 Implicit Latent Variables (Sparse Weights):

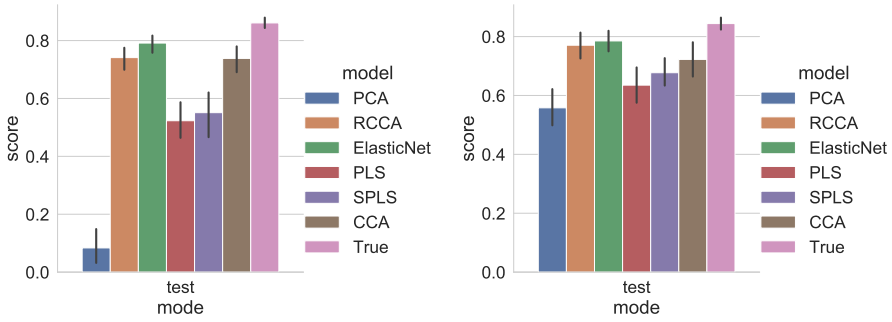
**Recovery of Weights and Loadings** Elastic regularization sets true zero weights close to zero and accurately retrieves true weights in both the identity (Figure **??**a) and non-identity (Figure **??**b) scenarios. Unregularized CCA and Ridge CCA are comparable to Elastic Net regularization but slightly worse in both scenarios (Figure IV.3) Figure **??**a highlights the disparity between PLS and RCCA compared to CCA. In particular, it is clear that the PLS and sPLS models have been skewed by the principal components. Furthermore, the sPLS does not result in appropriate shrinkage and identifies a number of false negatives. This is perhaps suprising because all three problems are equivalent in the population setting (due to identical view covariances). The models diverge in a sample setting because of non-identical sample covariance matrices, underscoring the distinction between population and sample settings and the interpretation complexities in the latter. PCA performs well in the random covariance scenario, but poorly in the identity covariance scenario. This suggests that the random covariance scenario results in a higher signal-to-noise ratio for these parameters.

**Recovery of Latent Variables**




**(a)** Identity Covariance Matrices **(b)** Random Covariance Matrices

**Figure IV.2:** Weights and Loadings for Implicit Latent Variable Data Generation. Blue signifies true zero weights and loadings, while Orange indicates estimated true non-zero weights and loadings.




**(a)** Identity Covariance Matrices **(b)** Random Covariance Matrices

**Figure IV.3:** Test Scores for Implicit Latent Variable Data Generation.

### 5.1.2 Explicit Latent Variables (Sparse Weights):

**Recovery of Weights and Loadings** A striking observation, though theoretically consistent, from Figure IV.4a is that PCA almost perfectly recovers the true weights and loadings for the GFA model with identity noise covariance. Admittedly, we have chosen a reasonably high signal-to-noise ratio for this experiment, but this nonetheless demonstrates that PCA can be a useful baseline for multiview data under an isotropic latent variable noise model. Indeed, in the identity noise covariance scenario, all the models perform similarly (Figure IV.5a) with the exception of CCA which appears to be unstable in this setting (Figure IV.4a). In the random noise covariance scenario, PCA performs poorly, and CCA now performs well (Figure IV.5b). Figure IV.4b indicates that with anisotropic noise covariance, PCA no longer captures the true loadings. There is no strong evidence that sPLS or Elastic Net regularization outperform PLS or RCCA, respectively, in this setting. This is unsurprising because the true weights are not sparse, and so the additional sparsity constraints do not help. However, this does illustrate that priors on weights do not translate well to priors on loadings.

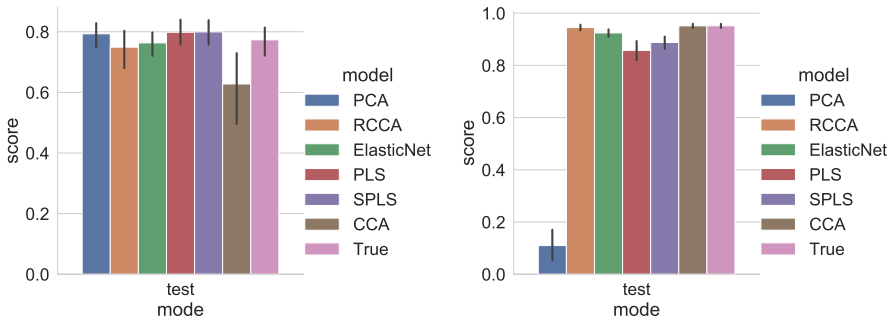
**Recovery of Latent Variables**

### 5.1.3 Measuring the Identitiness of the Covariance Matrices

The theory we developed in section **??** suggests that the identitiness of the covariance matrices is crucial for understanding how imposing sparsity on the weights imposes a prior belief in sparsity on the more biologically interesting loadings. We can measure the identitiness of the covariance matrices by looking at the eigenval-


**(a)** GFA (Identity Noise) **(b)** Probabilistic CCA (Random Noise)

**Figure IV.4:** Weights and Loadings for Explicit Latent Variable Data Generation Models. Blue signifies true zero weights and loadings, while Orange indicates estimated true non-zero weights and loadings.




**(a)** GFA **(b)** Probabilistic CCA

**Figure IV.5:** Test Scores for Explicit Latent Variable Data Generation Models.

ues of the covariance matrices. If the eigenvalues of the sample covariance matrix are all close to 1, then the sample covariance matrix is close to identity. Departures from 1 indicate that the sample covariance matrix is not close to identity and imply multicollinearity in the data.

In the simulated data, we can see that the data generation models with identity noise covariance matrices, have eigenvalues closer to one than (Figure IV.6). On the other hand, these plots (shown for 10 random samples) show that all of the sample covariance matrices depart from the ideal case, *even when the true covariance matrices are identity*.

## 5.2 Repeated Columns

In this section, we explore the stability of weights and loadings in CCA models when faced with repeated columns in the data. Our theoretical analysis suggests that while weights may vary and are arbitrary with repeated columns, loadings remain consistent.

Empirical results from our experiments validate this theoretical proposition. We observed that in PLS and sPLS models, which are significantly regularized (L2 regularization), the weights for repeated features tend to be uniform. However, in the case of CCA the weights applied to repeated features are almost completely arbitrary (see the scale). For Elastic and Ridge CCA, the weights applied to repeated features, though not identical, display greater stability due to shrinkage effects. In contrast, loadings for repeated features remained consistent across all tested models, reinforcing our theoretical stance on their preferential use for model interpretation in scenarios with repeated columns or near-repeated columns.

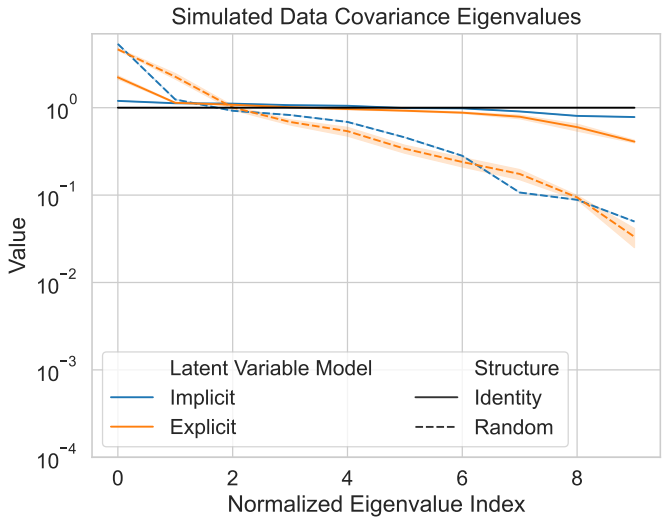
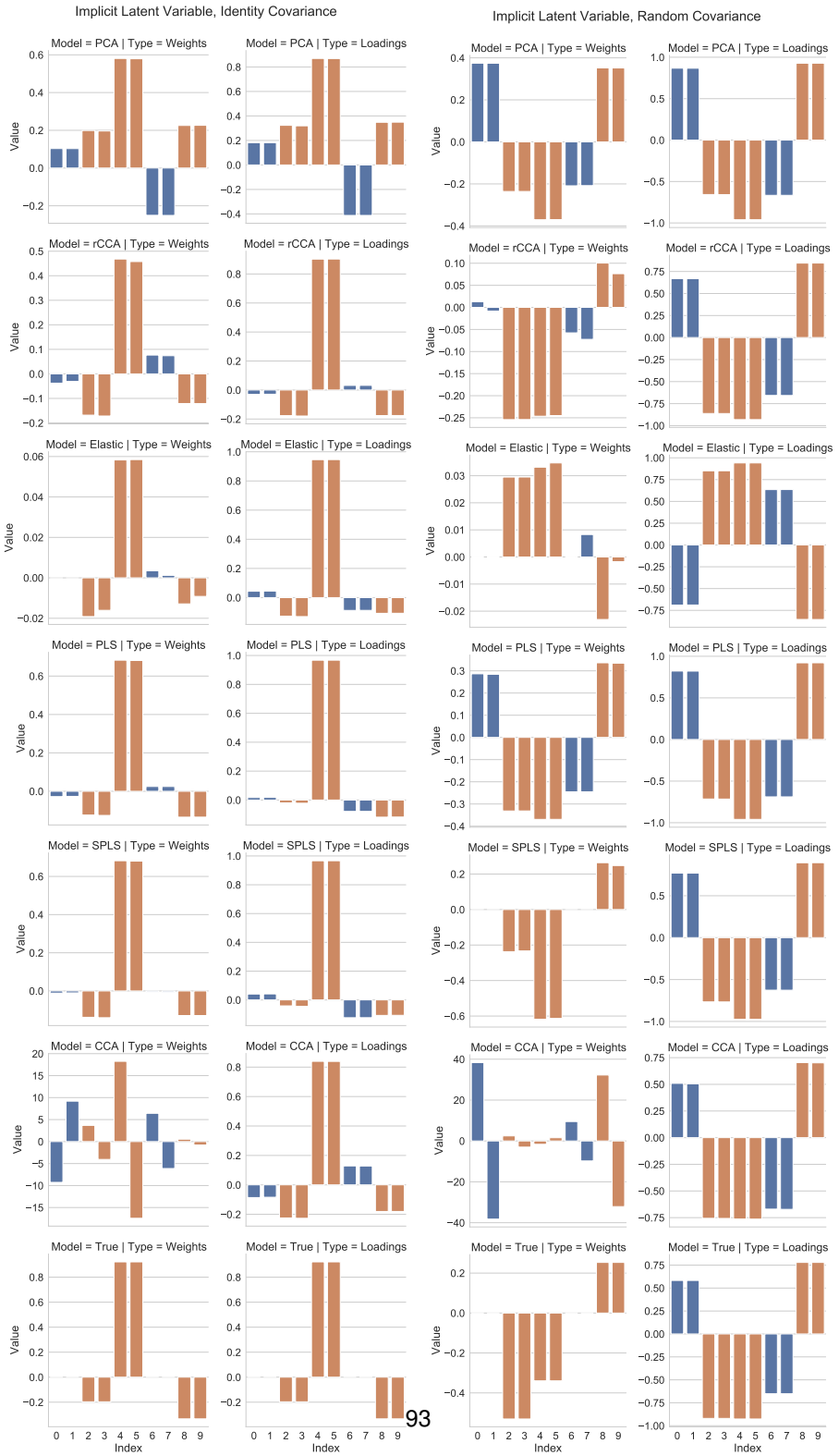


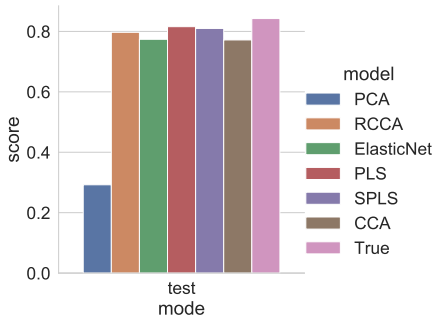

**Figure IV.6:** Eigenvalues of the covariance matrices for the simulated datasets.

## 5.3 Brain-Behaviour Simulations

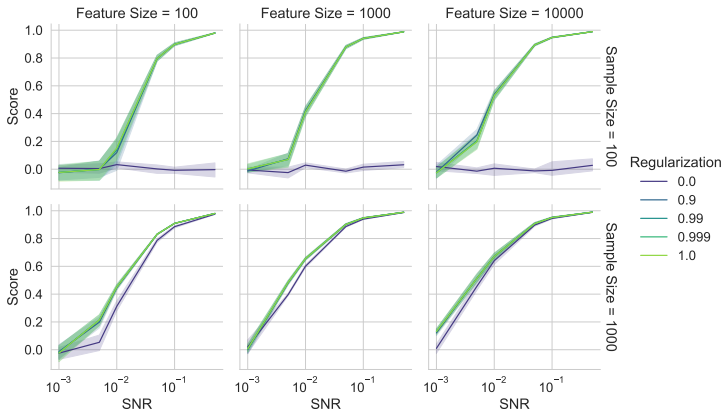



**(a)** Identity Covariance Matrices **(b)** Random Covariance Matrices

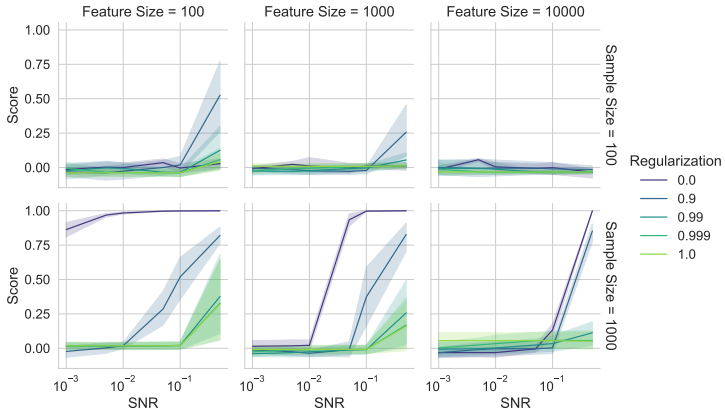
**Figure IV.7:** Weights and Loadings for Implicit Latent Variable Data Generation.




**(a)** Identity Covariance Matrices




**(a)** Identity Covariance Matrices




**(a)** Random Covariance Matrices

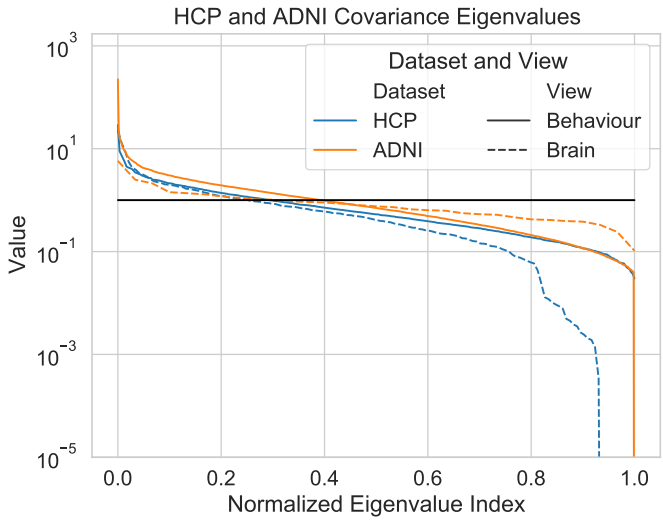



**Figure IV.11:** Eigenvalues of the covariance matrices for the HCP and ADNI datasets.

# 6 Revisiting Brain-Behaviour Results

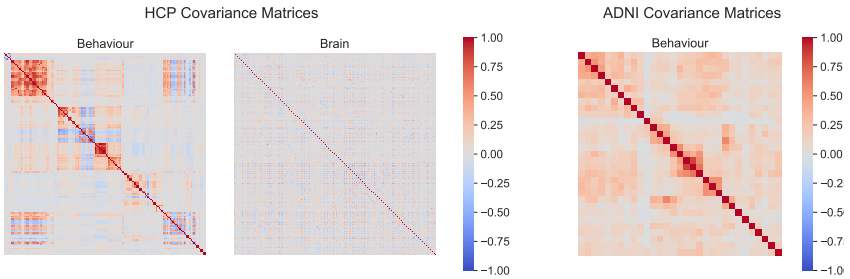
## 6.1 Identitiness of Covariance Matrices

In this section, we consider the identitiness of the covariance matrices for the HCP and ADNI datasets. Figure IV.11 shows the eigenvalues of the covariance matrices for the HCP and ADNI datasets while Figure IV.12 shows the covariance matrices themselves (with the ADNI brain covariance matrix left out due to its size). From Figure IV.11, we can see that the eigenvalues of the covariance matrices for the ADNI data are much closer to the ideal for identity covariance than for the HCP data.

From Figure IV.12, we can see the block structure of the covariance matrices.

## 6.2 Loading Similarity

‘




**(a)** HCP **(b)** ADNI

**Figure IV.12:** Covariance matrices for the HCP and ADNI datasets.

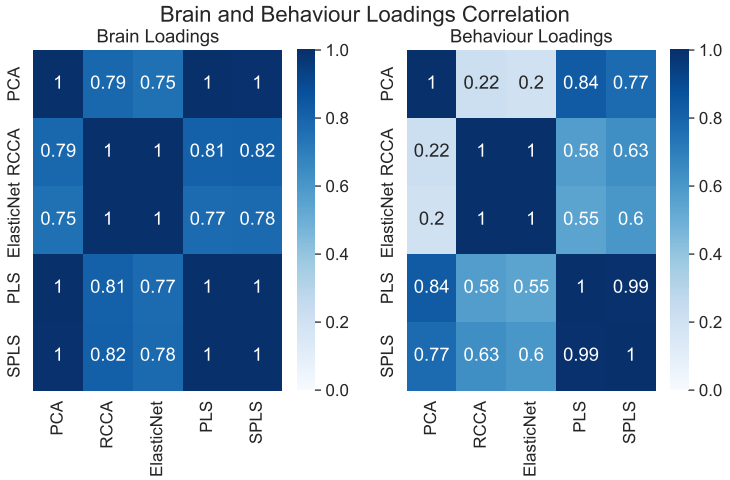



**Figure IV.13: HCP:** Correlation between the brain and behaviour representations for each model.

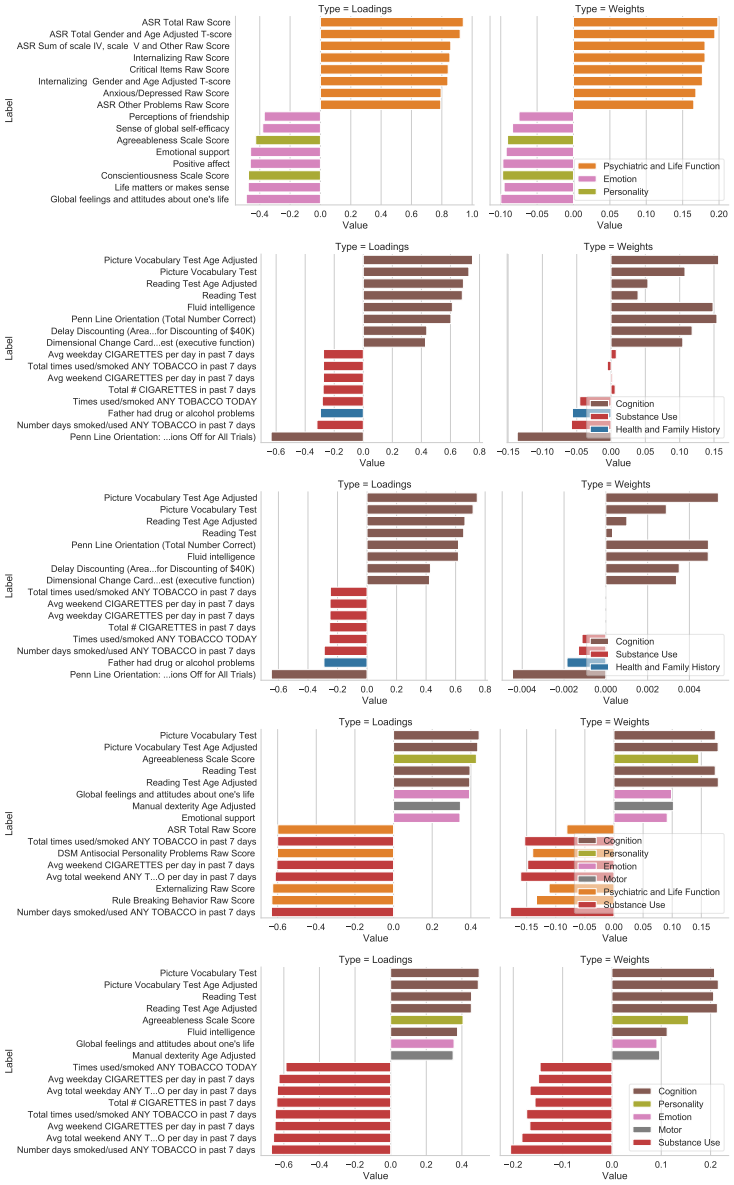



**Figure IV.14:** Top 8 positive and negative non-imaging loadings for each model

## 6.3 Comparing Behaviour Weights and Loadings

### 6.3.1 Human Connectome Project (HCP) Data

### 6.3.2 Alzheimer's Disease Neuroimaging Initiative (ADNI) Data

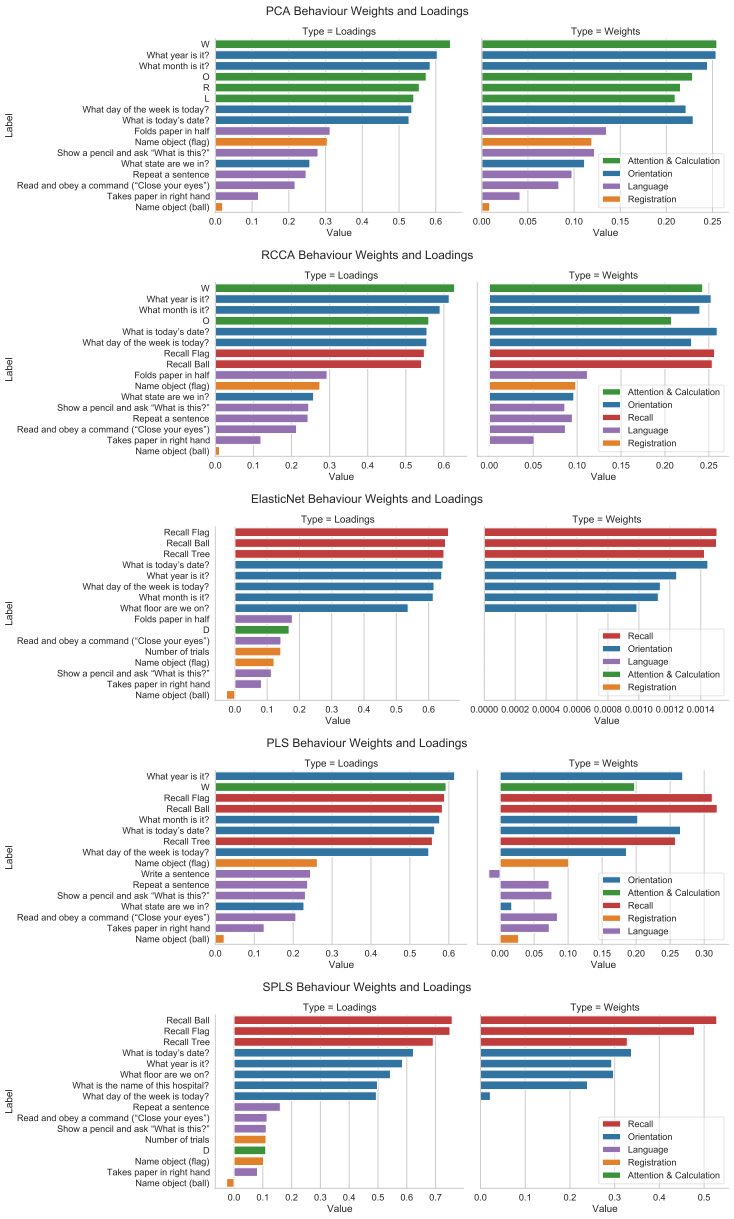



**Figure IV.15:** Bar plots of the behaviour weights and loadings for each model.

# 7 Discussion and Limitations

In this section, we discuss the implications of our findings as well as the limitations of our study.

## 7.1 Discussion

**Interpreting the Forward and Backward Models of CCA:** Consistent with Haufe et al., 2014, we have shown that assumptions imposed on the weights of a model do not in general transfer to the loadings. This is because the weights and loadings are only equivalent when the covariance matrices are identity. We have gone a step further than Haufe et al., 2014 by showing that the identitiness of the covariance matrices is crucial for understanding how imposing sparsity on the weights imposes a prior belief in sparsity on the more biologically interesting loadings.

**Sparsity on the weights does not imply sparsity on the loadings:** On the other hand, our results raise whether sparsity on the weights makes sense in the first place. For latent variable models of brain-behaviour associations, we have argued that the loadings are the more biologically relevant quantity. Since in practice, identity covariance is rarely a good assumption, the weights and loadings are not equivalent. This means that sparsity on the weights does not imply sparsity on the loadings, and so we should not expect sparsity on the weights to lead to more interpretable loadings. A practical step this implies is *to ensure that the data covariances are at least close to identity before applying sparse CCA methods*.

**Identitiness of real covariance matrices:** Between the HCP and ADNI datasets, only the ADNI data had eigenvalue spectra that were reasonably close to those of an identity matrix. This suggests that the only dataset and view where we can expect sparsity on the weights to imply sparsity on the loadings is the ADNI data. This is consistent with the results we have seen in the previous section, that sparsity improves performance in the ADNI data but not in the HCP data. Furthermore, the loadings and weights are much more similar to each other in the ADNI data than in the HCP data, supporting the idea that the ADNI weights are themselves somewhat interpretable as estimates of the biologically relevant loadings. Finally, in the well understood Alzheimer's disease data, we know that the identified weights (and loadings) are consistent with the known biology of the disease.

**Sample versus Population Setting:** The results from the simulated data illustrate the disparities that can arise between population and sample settings. Although PLS, RCCA, and CCA are equivalent under isotropic noise in a population framework, experiments showed that their performance can vary substantially in a sample setting. In particular, this manifested as PLS underperforming RCCA and CCA under isotropic noise *even though this is exactly the scenario where covariance identity holds and covariance thus equals correlation*. This is because the sample covariance matrix is not the same as the population covariance matrix and PLS is sensitive to even small differences in the principal components of the sample covariance matrix. Furthermore, in limited sample sizes, our estimations of the covariance matrices are not accurate, and so the identitiness of the covariance matrices is not guaranteed. This generally resulted in poor estimation of loadings from model weights *even when the weights themselves were estimated almost perfectly*. Therefore, it's crucial for researchers to recognize these nuances and adopt appropriate measures when extrapolating results, especially in brain-behaviour studies where typically only one sample is available and often limited in size.

**Ridge CCA is typically much better than PLS across datasets:** Our results show that Ridge CCA is typically much better than PLS across datasets. Much like regularised regression, it is unusual to need to use maximal ridge regularization even in high dimensions. Our results in simulated data even cast doubt on the touted stability of PLS over CCA with respect to population CCA problem. This means that while PLS might be more stable for a given dataset, it is not necessarily more stable across random samples from the same population.

**Can We Construct a Regularization Functional that Imposes Sparsity on the Loadings?** Finally, given our observations, a natural question to ask is whether we can construct a regularization functional that imposes sparsity on the loadings (instead of the weights). The answer is yes, but it is not straightforward and in the small sample setting, it is not clear that it is a good idea. The principle would be much the same as the Lasso, but we would need to use the sample covariance matrix to define the norm:

$$P(W) = \|W\|_1 \tag{IV.24}$$

$$P(L) = \|\hat{\Sigma}U\|_1 \tag{IV.25}$$

Which imposes an L1 penalty on the loadings via an L1 penalty on the weights multiplied by the sample covariance matrix. We could in principle apply the soft-thresholding operator to the estimated loadings. However we would need to be careful to ensure that the sample covariance matrix is invertible in order to get back to the weights. This is of course not guaranteed in the small sample setting.

# 8 Conclusion

In this chapter, we have explored the performance of several CCA variants on simulated and real data. We have shown that the choice of CCA variant can have a significant impact on the results. We have also shown that the identitiness of the covariance matrices is crucial for understanding how imposing sparsity on the weights imposes a prior belief in sparsity on the more biologically interesting loadings. We described a way to check the identitiness of the covariance matrices by looking at the eigenvalues of the covariance matrices and comparing to the ideal case. In the next chapter, we address the scalability of CCA and its variants by introducing a novel algorithm based on gradient descent.

# Chapter V

# Efficient Algorithms for the CCA Family: Unconstrained Losses with Unbiased Gradients

It seems easier to train a bi-directional LSTM with attention than to compute the SVD of a large matrix.**gemp**

Chris Ré

Contents

# Preface

The content of this chapter is based on a series of papers (Chapman, Aguila, and Wells, 2022; Chapman, Wells, and Aguila, 2023) as well as a NeurIPS workshop paper (**chapman2023neurips**). I am grateful to my co-authors Lennie Wells and Ana Lawry Aguila for their contributions to this work. In particular, Lennie's mathematical expertise improved the theoretical grounding of the idea greatly and Ana's access to the UK Biobank dataset enabled the application of our methods to a real-world biomedical dataset. In this thesis I include much of the work from these papers, but I exclude many of Lennie's extensive proofs where I can make no claim to have contributed beyond proofreading. It was a joy to bring this work to fruition and I am proud of the results we achieved.

# 1 Introduction

# 2 Background: A unified approach to the CCA family

So far in this thesis, we have considered only linear functions.

$$Z_k^{(i)} = \langle u_k^{(i)}, X^{(i)} \rangle. \tag{V.1}$$

**Stochastic PLS and CCA:** To the best of our knowledge, the state-of-the-art in Stochastic PLS and CCA are the subspace Generalized Hebbian Algorithm (**SGHA**)

of Z. Chen et al., 2019 and $\gamma$**-EigenGame** from I. M. Gemp et al., 2020; I. Gemp, McWilliams, et al., 2021. Specifically, SGHA utilizes a Lagrange multiplier heuristic along with saddle-point analysis, albeit with limited convergence guarantees. EigenGame focuses on top-k subspace learning but introduces an adaptive whitening matrix in the stochastic setting with an additional hyperparameter. Both methods set the benchmarks we aim to compare against in the subsequent experimental section. Like our method, both can tackle other symmetric Generalized Eigenvalue Problems in principle.

**DCCA and Deep Multiview CCA:** The deep canonical correlation analysis (DCCA) landscape comprises three principal approaches with inherent limitations. The first, known as the full-batch approach, uses analytic gradient derivations based on the full sample covariance matrix (Andrew et al., 2013). The second involves applying the full batch objective to large mini-batches, an approach referred to as **DCCA-STOL** (W. Wang, Arora, Livescu, and Bilmes, 2015). However, this approach gives biased gradients and therefore requires batch sizes much larger than the representation size in practice. This is the approach taken by both **DMCCA** (**somandepalli2019multimodal**) and **DGCCA** (**benton2017deep**) . The final set of approaches use an adaptive whitening matrix (W. Wang, Arora, Livescu, and Srebro, 2015; Chang, Xiang, and T. M. Hospedales, 2018) to mitigate the bias of the Deep CCA objective. However, the authors of **DCCA-NOI** highlight that the associated time constant complicates analysis and requires extensive tuning. These limitations make existing DCCA methods less practical and resource-efficient.

**Self-Supervised Learning: Barlow Twins** and **VICReg** have come to be known as part of the canonical correlation family of algorithms (Balestriero et al., 2023). Barlow Twins employs a redundancy reduction objective to make the representations of two augmented views both similar and decorrelated (Zbontar et al., 2021). Similarly, VICReg uses variance-invariance-covariance regularization, which draws upon canonical correlation principles, to achieve robust performance in diverse tasks (Bardes, Ponce, and LeCun, 2021). These methods serve as vital baselines for our experiments, owing to their foundational use of canonical correlation ideas.

**Deep CCA:** was originally introduced in Andrew et al., 2013; this was extended to an GEP-based formulation of **Deep Multi-view CCA** (DMCCA) in **somandepalli2019multimodal**. This can be defined using our $\mathrm{MCCA}$ notation as maximizing

$$\|\mathrm{MCCA}_K\left(Z^{(1)},...Z^{(I)}\right)\|_2 \tag{V.2}$$

over parameters $\theta$ of neural networks defining the representations $Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)})$ for $i \in [I]$.

# 3 Methods: Novel Objectives and Algorithms

## 3.1 Unconstrained objective for GEPs

First, we present proposition 3.1, a formulation of the top-$K$ subspace of GEP problems, which follows by applying the Eckhart–Young–Minsky inequality (Stewart and J.-G. Sun, 1990) to the eigen-decomposition of $B^{-1/2}AB^{-1/2}$. However, making this rigorous requires some technical care which we defer to the proof in supplement **??**.

**Proposition 3.1** (Eckhart–Young inspired objective for GEPs)**.** *The top-$K$ subspace of the GEP $(A, B)$ can be characterized by minimizing the following objective over $U \in \mathbb{R}^{D\times K}$:*

$$\mathcal{L}_{\textit{EY-GEP}}(U) := \operatorname{trace}\left(-2\,U^T AU + \left(U^T BU\right)\left(U^T BU\right)\right) \tag{V.3}$$

*Moreover, the minimum value is precisely $-\sum_{k=1}^{K}\lambda_k^2$, where $(\lambda_k)$ are the generalized eigenvalues.*

This objective also has appealing geometrical properties. It is closely related to a wide class of unconstrained objectives for PCA and matrix completion which have no spurious local optima (Ge, Jin, and Zheng, 2017), i.e. all local optima are in fact global optima. This implies that certain local search algorithms, such as stochastic gradient descent, should indeed converge to a global optimum.

**Proposition 3.2.** *[No spurious local minima] The objective $\mathcal{L}_{\textit{EY-GEP}}$ has no spurious local minima. That is, any matrix $\bar{U}$ that is a local minimum of $\mathcal{L}_{\textit{EY-GEP}}$ must in fact be a global minimum.*

It is also possible to make this argument quantitative by proving a version of the strict saddle property from **ge2015escaping**; Ge, Jin, and Zheng, 2017; we state an informal version here and give full details in **??**.

**Corollary 3.1** (Informal: Polynomial-time Optimization)**.** *Under certain conditions on the eigenvalues and generalized eigenvalues of $(A, B)$, one can make quantitative the claim that: any $U_K \in \mathbb{R}^{D\times K}$ is either close to a global optimum, has a large gradient $\nabla\mathcal{L}_{\textit{EY-GEP}}$, or has Hessian $\nabla^2\mathcal{L}_{\textit{EY-GEP}}$ with a large negative eigenvalue.*

*Therefore, for appropriate step-size sequences, certain local search algorithms, such as sufficiently noisy SGD, will converge in polynomial time with high probability.*

## 3.2 Corresponding Objectives for the CCA family

For the case of linear CCA we have $U^TAU = \sum_{i\neq j}\mathrm{Cov}(Z^{(i)}, Z^{(j)}), U^TBU = \sum_i \mathrm{Var}(Z^{(i)})$. To help us extend this to the general case of nonlinear transformations, Equation (II.1), we define the analogous matrices of total between-view covariance and total within-view variance

$$C(\theta) = \sum_{i\neq j}\mathrm{Cov}(Z^{(i)}, Z^{(j)}), \quad V(\theta) = \sum_i \mathrm{Var}(Z^{(i)}) \tag{V.4}$$

In the case of linear transformations, Equation (V.1), it makes sense to add a ridge penalty so we can define

$$V_\alpha(\theta) = \sum_i \alpha_i U^{(i)\,T}U^{(i)} + (1-\alpha_i)\mathrm{Var}(Z^{(i)}) \tag{V.5}$$

This immediately leads to following unconstrained objective for the CCA-family of problems.

**Definition 3.1** (Family of EY Objectives)**.** *Learn representations $Z^{(i)} = f^{(i)}(X^{(i)};\theta^{(i)})$ minimizing*

$$\mathcal{L}_{EY}(\theta) = -2\,\mathrm{trace}\,C(\theta) + \|V_\alpha(\theta)\|_F^2 \tag{V.6}$$

**Unbiased estimates:** since empirical covariance matrices are unbiased, we can construct unbiased estimates to $C, V$ from a batch of transformed variables $\mathbf{Z}$.

$$\hat{C}(\theta)[\mathbf{Z}] = \sum_{i\neq j}\widehat{\mathrm{Cov}}(\mathbf{Z}^{(i)}, \mathbf{Z}^{(j)}), \quad \hat{V}(\theta)[\mathbf{Z}] = \sum_i \widehat{\mathrm{Var}}(\mathbf{Z}^{(i)}) \tag{V.7}$$

In the linear case we can construct $\hat{V}_\alpha(\theta)[\mathbf{Z}]$ analogously by plugging sample covariances into Equation (V.5). Then if $\mathbf{Z}, \mathbf{Z}'$ are two independent batches of transformed variables, the batch loss

$$\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}'] := -2\,\mathrm{trace}\,\hat{C}[\mathbf{Z}] + \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}']\rangle_F \tag{V.8}$$

gives an unbiased estimate of $\mathcal{L}_{\mathsf{EY}}(\theta)$.This loss is a differentiable function of $\mathbf{Z}, \mathbf{Z}'$ and so also of $\theta$.

**Simple algorithms:** We first define a very general algorithm using these estimates in Algorithm 1. In the next section we apply this algorithm to multi-view stochastic CCA and PLS, and Deep CCA.

**Algorithm 1:** GEP-EY: General algorithm for learning correlated representations

**Input:** data stream of mini-batches $(\mathbf{X}(b))_{b=1}^{\infty}$ where each consists of $M$ samples from the original dataset. Learning rate $(\eta_t)_t$. Number of time steps $T$. Class of functions $f(\cdot;\theta)$ whose outputs are differentiable with respect to $\theta$.
**Initialize:** $\hat{\theta}$ with suitably random entries
**for** $t = 1$ **to** $T$ **do**
  Obtain two independent mini-batches $\mathbf{X}(b), \mathbf{X}(b')$ by sampling $b, b'$ independently
  Compute batches of transformed variables $\mathbf{Z}(b) = f(\mathbf{X}(b);\theta), \mathbf{Z}(b') = f(\mathbf{X}(b');\theta)$
  Estimate loss $\hat{\mathcal{L}}_{\mathsf{EY}}(\theta)$ using Equation (V.8)
  Obtain gradients by back-propagation and step with your favourite optimizer.
**end for**

## 3.3 'EigenGame' Approach for Ordered Subspaces

We now present a second formulation of the top-$K$ subspace of a GEP, which is more closely related to the 'EigenGame' approach of I. M. Gemp et al., 2020; I. Gemp, McWilliams, et al., 2021.

Our first proposed method solves the general form of the generalized eigenvalue problem in equation (**??**) for the top-k eigenvalues and their associated eigenvectors in parallel. We are thus interested in both the top-k subspace problem and the top-k eigenvectors themselves. Our method extends the Generalized Hebbian Algorithm to GEPs, and we thus refer to it as GHA-GEP.

In the full-batch version of our algorithm when $A$ is known to be positive semidefinite, each eigenvector estimate has updates with the form

$$\begin{aligned}\Delta_i^{\text{GHA-GEP-PSD}} &= \overbrace{A\hat{w}_i}^{\text{Reward}} - \overbrace{\sum_{j\leq i} B\hat{w}_j\left(\hat{w}_j^\top A\hat{w}_i\right)}^{\text{Penalty}} &&= \overbrace{A\hat{w}_i}^{\text{Reward}} - \overbrace{B\hat{w}_i\left(\hat{w}_i^\top A\hat{w}_i\right)}^{\text{Variance Penalty}} - \overbrace{\sum_{j<i} B\hat{w}_j\left(\hat{w}_j^\top A\hat{w}_i\right)}^{\text{Orthogonality Penalty}} \\ &= A\hat{w}_i - \sum_{j\leq i} B\hat{w}_j\Gamma_{ij} &&= A\hat{w}_i - B\hat{w}_i\Gamma_{ii} - \sum_{j<i} B\hat{w}_j\Gamma_{ij}\end{aligned} \tag{V.9}$$

where $\hat{w}_j$ is our estimator to the eigenvector associated with the $j^{\text{th}}$ largest eigenvalue and in the stochastic setting, we can replace $A$ and $B$ with their unbiased estimates $\hat{A}$ and $\hat{B}$. We will use the notation $\Gamma_{ij} = \left(\hat{w}_j^\top A \hat{w}_i\right)$ to facilitate comparison with previous work in Appendix **??**. $\Gamma_{ij}$ has a natural interpretation as Lagrange multiplier for the constraint $w_i^\top B w_j = 0$; indeed, Z. Chen et al. (2019) prove that $\left(\hat{w}_j^\top A \hat{w}_i\right)$ is the optimal value of the corresponding Lagrange multiplier for their GEP formulation; we summarise this derivation in Appendix **??** for ease of reference. We also label the terms as rewards and penalties to facilitate discussion with respect to the EigenGame framework in Appendix **??** and recent work in self-supervised learning in Appendix **??**.

However, when $A$ has negative eigenvalues, the iteration defined in (V.9) can 'blow-up' from certain initial values. We therefore propose the following modification:

$$\Delta_i^{\text{GHA-GEP}} = A\hat{w}_i - B\hat{w}_i \max(\Gamma_{ii}, 0) - \sum_{j<i} B\hat{w}_j \Gamma_{ij} \tag{V.10}$$

Note that this reduces to (V.9) when $A$ is positive semi-definite. The following proposition, proved in Appendix **??**, justifies this choice of update.

*Proposition* 3.1 (Unique stable stationary point). Given exact parents and assuming the top-k generalized eigenvalues of $A$ and $B$ are distinct and positive, the only stable stationary point of the iteration defined by (V.10) is the eigenvector $w_i$ (up to sign).

## 3.4 Defining Utilities and Pseudo-Utilities with Lagrangian Functions

Now observe that the updates (V.9) can be written as the gradients of a Lagrangian pseudo-utility function:

$$\mathcal{PU}_i^{\text{GHA-GEP-PSD}}(w_i | w_{j<i}, \Gamma) = \tfrac{1}{2}\hat{w}_i^\top A \hat{w}_i + \tfrac{1}{2}\Gamma_{ii}(1 - \hat{w}_i^\top B \hat{w}_i) - \sum_{j<i} \Gamma_{ij} \hat{w}_j^\top B \hat{w}_i. \tag{V.11}$$

We show how this result is closely related to the pseudo-utility functions in Z. Chen et al. (2019) and suggests an alternative pseudo-utility function for the work in I. Gemp, McWilliams, et al. (2021) in Appendix **??** which, unlike the original work, does not require stop gradient operators.

If we plug in the relevant $w_i$ and $w_j$ terms into $\Gamma$, we obtain the following utility

function:

$$\begin{aligned}\mathcal{U}_i^{\delta\text{-PSD}}(w_i; w_{j<i}) &= \tfrac{1}{2}\hat{w}_i^\top A\hat{w}_i + \tfrac{1}{2}\hat{w}_i^\top A\hat{w}_i \left(1 - \hat{w}_i^\top B\hat{w}_i\right) - \sum_{j<i} \hat{w}_i^\top A\hat{w}_j\hat{w}_j^\top B\hat{w}_i \\ &= (\hat{w}_i^\top A\hat{w}_i) - \tfrac{1}{2}(\hat{w}_i^\top A\hat{w}_i)(\hat{w}_i^\top B\hat{w}_i) - \sum_{j<i}(\hat{w}_i^\top A\hat{w}_j)(\hat{w}_j^\top B\hat{w}_i)\end{aligned} \tag{V.12}$$

Again, we apply a modification to prevent blow-up when $A$ has negative eigenvalues, giving utility

$$\mathcal{U}_i^{\delta}(w_i; w_{j<i}) = (\hat{w}_i^\top A\hat{w}_i) - \tfrac{1}{2}\max((\hat{w}_i^\top A\hat{w}_i), 0)\,(\hat{w}_i^\top B\hat{w}_i) - \sum_{j<i}(\hat{w}_i^\top A\hat{w}_j)(\hat{w}_j^\top B\hat{w}_i) \tag{V.13}$$

A remarkable fact is that this utility function actually defines a solution to the GEP problem! We prove the following consistency result in Appendix **??**.

*Proposition* 3.2 (Unique utility maximiser). Assuming the top-$i$ generalized eigenvalues of the GEP (**??**) are positive and distinct. Then the unique maximizer of the utility in (V.13) for exact parents is precisely the $i^{th}$ eigenvector (up to sign).

This utility function allows us to formalise $\Delta$-EigenGame, whose solution corresponds to the top-k solution of equation (**??**).

**Definition 3.2.** *Let $\Delta$-EigenGame be the game with players $i \in \{1, ..., k\}$, strategy space $\hat{w}_i \in \mathbb{R}^d$, where d is the dimensionality of A and B, and utilities $\mathcal{U}_i^\delta$ defined in equation (V.13).*

An immediate corollary of Proposition 3.2 is:

**Corollary 3.2.** *The top-$k$ generalized eigenvectors form the unique, strict Nash equilibrium of $\Delta$-EigenGame.*

Furthermore, the penalty terms in the utility function (V.11) have a natural interpretation as a projection deflation as shown in appendix **??**.

Next note that it is easy to compute the derivative

$$\begin{aligned}\Delta_i^\delta &= \frac{\partial \mathcal{U}_i^\delta(w_i; w_{j<i})}{\partial w_i} \\ &= 2A\hat{w}_i - \{A\hat{w}_i(\hat{w}_i^\top B\hat{w}_i) + (\hat{w}_i^\top A\hat{w}_i)B\hat{w}_i\} - \sum_{j<i}\{A\hat{w}_j(\hat{w}_j^\top B\hat{w}_i) + (\hat{w}_j^\top A\hat{w}_i)B\hat{w}_j\} \\ &= \Delta_i^{\text{GHA-GEP}} + \{A\hat{w}_i - \sum_{j\leq i} A\hat{w}_j(\hat{w}_j^\top B\hat{w}_i)\}\end{aligned} \tag{V.14}$$

We can use these gradients as updates step for an alternative algorithm for the GEP which we call $\delta$-EigenGame (where, consistent with previous work, we use upper case for the game and lower case for its associated algorithm). We can now discuss stochastic versions of the algorithms introduced above, the setting where our methods excel.

## 3.5 Stochastic/Data-streaming versions

This paper is motivated by cases where the algorithm only has access to unbiased sample estimates of $A$ and $B$. These estimates, denoted $\hat{A}$ and $\hat{B}$, are therefore random variables. A nice property of both our proposed GHA-GEP and $\delta$-EigenGame is that $A$ and $B$ appear as multiplications in both of their updates (as opposed to as divisors). This means that we can simply substitute them for our unbiased estimates at each iteration. For the GHA-GEP algorithm this gives us updates based on stochastic unbiased estimates of the gradient

## 3.6 Applications to (multi-view) stochastic CCA and PLS, and Deep CCA

**Lemma 3.1** (Objective recovers GEP formulation of linear (multi-view) CCA)**.** *When the $f^{(i)}$ are linear, as in Equation* (V.1)*, the population loss from Equation* (V.6) *recovers MCCA as defined in* **??***.*

*Proof.* By construction, for linear MCCA we have $C = U^T A U$, $V_\alpha = U^T B_\alpha U$, where $(A, B_\alpha)$ define the GEP for MCCA introduced in Equation (II.29). So $\mathcal{L}_{\text{EY}}(U) = \mathcal{L}_{\text{EY-GEP}}(U)$ and by Proposition 3.1 the optimal set of weights define a top-$K$ subspace of the GEP, and so is a MCCA solution. □

Moreover, by following through the chain of back-propagation, we obtain gradient estimates in $\mathcal{O}(MKD)$ time. Indeed, we can obtain gradients for the transformed variables in $\mathcal{O}(MK^2)$ time so the dominant cost is then updating $U$; we flesh this out with full details in **??**.

**Lemma 3.2.** *[Objective recovers Deep Multi-view CCA] Assume that there is a final linear layer in each neural network $f^{(i)}$. Then at any local optimum, $\hat{\theta}$, of the population problem, we have*

$$\mathcal{L}_{EY}(\hat{\theta}) = -\|\mathrm{MCCA}_K(\hat{Z})\|_2^2$$

*where* $\hat{Z} = f_{\hat{\theta}}(X)$. *Therefore,* $\hat{\theta}$ *is also a local optimum of objectives from* **somandepalli2019multimodal***; Andrew et al., 2013 as defined in Equation* (V.2).

*Proof sketch: see* **??** *for full details.* Consider treating the penultimate-layer representations as fixed, and optimising over the weights in the final layer. This is precisely equivalent to optimising the Eckhart-Young loss for linear CCA where the input variables are the penultimate-layer representations. So by Proposition 3.2, a local optimum is also a global optimum, and by Proposition 3.1 the optimal value is the negative sum of squared generalised eigenvalues. □

## 3.7 Application to SSL

We can directly apply Algorithm 1 to SSL. If we wish to have the same neural network transforming each view, we can simply tie the weights $\theta^{(1)} = \theta^{(2)}$. When the paired data are generated from applying independent, identically distributed (i.i.d.) augmentations to the same original datum, it is intuitive that tying the weights is a sensible procedure, and perhaps acts as a regulariser. We make certain notions of this intuition precise for CCA and Deep CCA in **??**.

To provide context for this proposal, we also explored in detail how VICReg and Barlow twins are related to CCA. For now we focus on VICReg, whose loss can be written as

$$\mathcal{L}_{\mathsf{VR}}(Z^{(1)}, Z^{(2)}) = \gamma \mathbb{E}\|Z^{(1)} - Z^{(2)}\|^2 + \sum_{i \in \{1,2\}} \left[ \alpha \sum_{k=1}^{K} \left( 1 \text{ - } \sqrt{\mathrm{Var}(Z_i^{(i)})} \right)_+ + \beta \sum_{\substack{k,l=1 \\ k \neq l}}^{K} \mathrm{Cov}(Z_k^{(i)}, Z_l^{(i)})^2 \right]$$

where $\alpha, \beta, \gamma > 0$ are tuning parameters and, as in the framework of Section 2, the $Z^{(1)}, Z^{(2)}$ are $K$-dimensional representations, parameterised by neural networks in Equation (II.1). Our main conclusions regarding optima of the population loss are:

- Consider the linear setting with untied weights. Then global optimisers of the VICReg loss define CCA subspaces, but may not be of full rank.
- Consider the linear setting with tied weights and additionally assume that the data are generated by i.i.d. augmentations. Then the same conclusion holds.
- In either of these settings, the optimal VICReg loss is a component-wise decreasing function of $\mathrm{CCA}_K(X^{(1)}, X^{(2)})$ the vector of population canonical correlations.
- VICReg can therefore be interpreted as a formulation of Deep CCA, but one that will not in general recover full rank representations.

We give full mathematical details and further discussion in **??**. The analysis for Barlow twins is more difficult, but we present a combination of mathematical and empirical arguments which suggest all the same conclusions hold, again see **??**.

# 4 Experiments

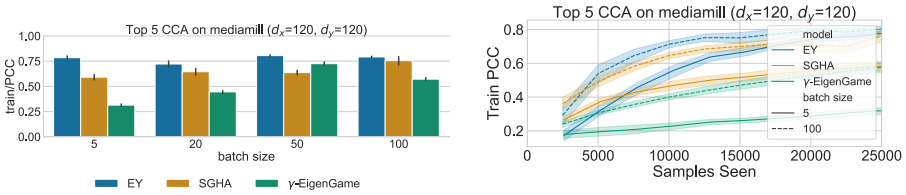
## 4.1 Stochastic CCA

First, we compare our proposed method, CCA-EY, to the baselines of $\gamma$-EigenGame and SGHA. Our experimental setup is almost identical to that of Z. Meng, Chakraborty, and Singh, 2021; I. Gemp, C. Chen, and McWilliams, 2022; unlike I. Gemp, C. Chen, and McWilliams, 2022 we do not simplify the problem by first performing PCA on the data before applying the CCA methods, which explains the decrease in performance of $\gamma$-EigenGame compared to their original work. All models are trained for a single epoch with varying mini-batch sizes ranging from 5 to 100. We use Proportion of Correlation Captured (PCC) as our evaluation metric, defined as PCC $= (\sum_{i=k}^{K} \rho_k)/(\sum_{k=1}^{K} \rho_k^*)$ where $\rho_k$ are the full batch correlations of the learnt representations, and $\rho_k^*$ are the canonical correlations computed numerically from the full batch covariance matrices.

**Parameters:** For each method, we searched over a hyperparameter grid using Biewald (2020).

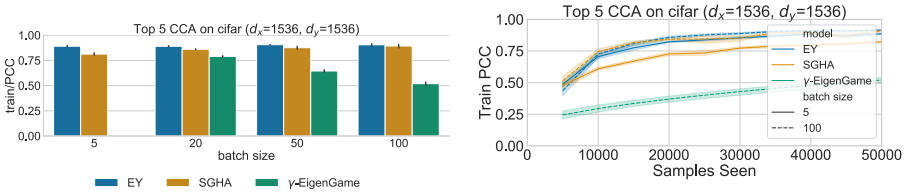
| Parameter | Values |
|---|---|
| minibatch size | 5,20,50,100 |
| components | 5 |
| epochs | 1 |
| seed | 1, 2, 3, 4, 5 |
| lr | 0.01, 0.001, 0.0001 |
| $\gamma^1$ | 0.01,0.1,1,10 |

**Observations:** Figure V.1 compares the algorithms on the MediaMill dataset. Figure V.1a shows that CCA-EY consistently outperforms both $\gamma$-EigenGame and SGHA in terms of PCC across all evaluated mini-batch sizes. Figure V.1b examines the learning curves for batch sizes 5 and 100 in more detail; CCA-EY appears to learn more slowly than SGHA at the start of the epoch, but clearly outperforms SGHA as the number of samples seen increases. $\gamma$-EigenGame significantly underperforms SGHA and CCA-EY, particularly for small batch sizes.




**(a)** **(b)**

**Figure V.1:** Stochastic CCA on MediaMill using the Proportion of Correlation Captured (PCC) metric: (a) Across varying mini-batch sizes, trained for a single epoch, and (b) Training progress over a single epoch for mini-batch sizes 5, 100. Shaded regions signify $\pm$ one standard deviation around the mean of 5 runs.




**(a)** **(b)**

**Figure V.2:** Stochastic CCA on CIFAR using the Proportion of Correlation Captured (PCC) metric: (a) Across varying mini-batch sizes, trained for a single epoch, and (b) Training progress over a single epoch for mini-batch sizes 5, 100. Shaded regions signify $\pm$ one standard deviation around the mean of 5 runs.

## 4.2 Deep CCA

Second, we compare DCCA-EY against the DCCA methods described in **??**. The experimental setup is identical to that of W. Wang, Arora, Livescu, and Srebro, 2015. We learn $K = 50$ dimensional representations, using mini-batch sizes ranging from 20 to 100 and train for 50 epochs. Because there is no longer a ground truth we have to use Total Correlation Captured (TCC), given by TCC $= \sum_{i=k}^{K} \rho_k$ where $\rho_k$ are now the empirical correlations between the representations on a validation set.

**Further details:** As in W. Wang, Arora, Livescu, and Srebro (2015), we used multilayer perceptrons with two hidden layers with size 800 and an output layer of 50 with ReLU activations. We train for 20 epochs.

**Parameters:** For each method, we searched over a hyperparameter grid using Biewald (2020).

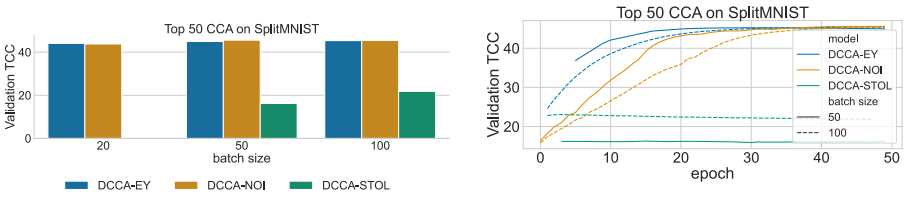
| Parameter | Values |
|---|---|
| minibatch size | 100, 50, 20 |
| lr | 1e-3, 1e-4, 1e-5 |
| $\rho^2$ | 0.6, 0.8, 0.9 |
| epochs | 50 |

**Observations:** Figure V.3 compares the methods on the splitMNIST dataset. DCCA-STOL captures significantly less correlation than the other methods, and breaks down when the mini-batch size is less than the dimension $K = 50$ due to low rank empirical covariances. DCCA-NOI performs similarly to DCCA-EY but requires careful tuning of an additional hyperparameter, and shows significantly slower speed to convergence (Figure V.3b).

Figure V.4 compares the methods on the XRMB dataset. DCCA-STOL captures significantly less correlation than the other methods, and breaks down when the mini-batch size is less than the dimension $K = 50$ due to low rank empirical covariances. DCCA-NOI performs similarly to DCCA-EY but requires careful tuning of an additional hyperparameter, and shows significantly slower speed to convergence (Figure V.3b).

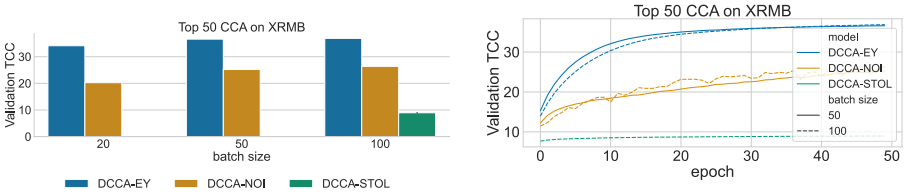
## 4.3 Deep Multiview CCA: Robustness Across Different Batch Sizes

Third, we compare DCCA-EY to the existing DMCCA and DGCCA methods on the mfeat dataset; this contains 2,000 handwritten numeral patterns across six distinct feature sets, including Fourier coefficients, profile correlations, Karhunen-Love



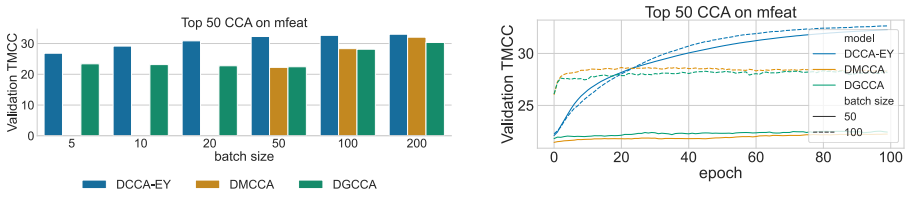

**(a)** **(b)**

**Figure V.3:** Deep CCA on SplitMNIST using the Validation TCC metric: (a) after training each model for 50 epochs with varying batch sizes; (b) learning progress over 50 epochs.




**(a)** **(b)**

**Figure V.4:** Deep CCA on XRMB using the Validation TCC metric: (a) after training each model for 50 epochs with varying batch sizes; (b) learning progress over 50 epochs.




**(a)** **(b)**

**Figure V.5:** Deep Multi-view CCA on mfeat using the Validation TMCC metric: (a) after training each model for 100 epochs with varying batch sizes; (b) learning progress over 100 epochs.

coefficients, pixel averages in $2 \times 3$ windows, Zernike moments, and morphological features. We again learn $K = 50$ dimensional representations, but now train for 100 epochs. We use a multiview extension of the TCC metric, which averages correlation across views; we call this Total Multiview Correlation Captured (TMCC), defined as $\text{TMCC} = \sum_{k=1}^{K} \frac{1}{I(I-1)} \sum_{i,j \leq I, i \neq j} \text{corr}(Z_k^{(i)}, Z_k^{(j)})$, using the notation of Section 2.

**Parameters:** For each method, we searched over a hyperparameter grid using Biewald (2020).

| Parameter | Values |
|---|---|
| minibatch size | 5,10,20,50,100,200 |
| components | 50 |
| epochs | 100 |
| lr | 0.01, 0.001, 0.0001, 0.00001 |

**Observations:** Figure V.5a shows that DCCA-EY consistently outperforms both DGCCA and DMCCA across various mini-batch sizes in capturing validation TMCC. Just like DCCA-NOI, DMCCA breaks down when the batch size is smaller than $K$. This is due to singular empirical covariances; DGCCA does not break down, but does significantly underperform with smaller batch sizes. This limits their practical applicability to large-scale data. Figure V.5b shows learning curves for batch sizes 50 and 100. DMCCA and DGCCA both quickly learn significant correlations but then plateau out; our method consistently improves, and significantly outperforms them by the end of training.

## 4.4 Stochastic PLS UK Biobank

Next, we demonstrate the scalability of our methods to extremely high-dimensional data by applying stochastic PLS to imaging genetics data from the UK Biobank

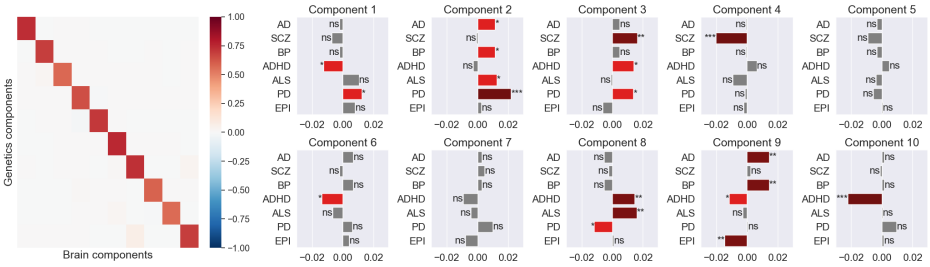
(Sudlow et al., 2015). PLS is typically used for imaging-genetics studies owing to the extremely high dimensionality of genetics data requiring lots of regularisation. PLS can reveal novel phenotypes of interest and uncover genetic mechanisms of disease and brain morphometry. Previous imaging genetics analyses using full-batch PLS were limited to much smaller datasets (**Lorenzi2018**; Taquet et al., 2021; Le Floch et al., 2012). The only other analysis on the UK Biobank at comparable scale partitions the data into clusters and bootstrapping local PLS solutions on these clusters (**lorenzi2017secure**; **altmann2023tackling**). We ran PLS-EY with mini-batch size 500 on brain imaging (82 regional volumes) and genetics (582,565 variants) data for 33,333 subjects. See supplement (Section **??**) for data pre-processing details. To our knowledge, this is the largest-scale PLS analysis of biomedical data to-date.

**Further details:** The UK BioBank data consisted of real-valued continuous brain volumes and ordinal, integer genetic variants. We used pre-processed (using FreeSurfer (Fischl, 2012)) grey-matter volumes for 66 cortical (Desikan-Killiany atlas) and 16 subcortical brain regions and 582,565 autosomal genetic variants. The affects of age, age squared, intracranial volume, sex, and the first 20 genetic principal components for population structure were removed from the brain features using linear regression to account for any confounding effects. Each brain ROI was normalized by removing the mean and dividing the standard deviation. We processed the genetics data using PLINK (Purcell et al., 2007) keeping genetic variants with a minor allele frequency of at least 1% and a maximum missingness rate of 2%. We used mean imputation to fill in missing values and centered each variant.

To generate measures of genetic disease risk, we calculated polygenic risk scores using PRSice (Euesden, Lewis, and O'Reilly, 2014). We calculated scores, with a p-value threshold of 0.05, using GWAS summary statistics for the following diseases; Alzheimer's (Lambert et al., 2013), Schizophrenia (Trubetskoy et al., 2022), Bipolar (Mullins et al., 2021), ADHD (Demontis et al., 2023), ALS (Rheenen et al., 2021), Parkinson's (Nalls et al., 2019), and Epilepsy (International League Against Epilepsy Consortium on Complex Epilepsies, 2018), using the referenced GWAS studies.

The GEP-EY PLS analysis was trained for 100 epochs using a learning rate of 0.0001 with a minibatch size of 500.

**Observations:** We see strong validation correlation between all 10 corresponding pairs of vectors in the PLS subspace and weak cross correlation, indicating that our model learnt a coherent and orthogonal subspace of covariation (Figure V.6a), a remarkable feat for such high-dimensional data. We found that the PLS brain




**(a)** **(b)**

**Figure V.6:** (a) Correlations between PLS components for UK Biobank. (b) Correlations between PLS brain components and genetic risk scores. AD=Alzheimer's disease, SCZ=Schizophrenia, BP=Bipolar, ADHD=Attention deficit hyperactivity disorder, ALS=Amyotrophic lateral sclerosis, PD=Parkinson's disease, EPI=Epilepsy. ns : $0.05 < p <= 1, * : 0.01 < p <= 0.05, ** : 0.001 < p <= 0.01, *** : 0.0001 < p <= 0.001$.

subspace was associated with genetic risk measures for several disorders (Figure V.6b), suggesting that the PLS subspace encodes relevant information for genetic disease risk, a significant finding for biomedical research.

## 4.5 Self-Supervised Learning with SSL-EY

Finally, we benchmark our self-supervised learning algorithm, SSL-EY, with Barlow Twins and VICReg on CIFAR-10 and CIFAR-100. Each dataset contains 60,000 labelled images, but these are over 10 classes for CIFAR-10 and 100 classes for CIFAR-100.

We follow a standard experimental design (Tong et al., 2023). Indeed, we use the sololearn library (Da Costa et al., 2022), which offers optimized setups particularly tailored for VICReg and Barlow Twins. All methods utilize a ResNet-18 encoder coupled with a bi-layer projector network. Training spans 1,000 epochs with batches of 256 images. For SSL-EY, we use the hyperparameters optimized for Barlow Twins, aiming not to outperform but to showcase the robustness of our method. We predict labels via a linear probe on the learnt representations and evaluate performance with Top-1 and Top-5 accuracies on the validation set. For more details, refer to the supplementary material **??**.

**Observations:** Table 5.1 shows that SSL-EY is competitive with Barlow Twins and VICReg. This is remarkable because we used out-of-the-box hyperparameters

| Method | CIFAR-10 Top-1 | CIFAR-10 Top-5 | CIFAR-100 Top-1 | CIFAR-100 Top-5 |
|---|---|---|---|---|
| Barlow Twins | **92.1** | 99.73 | **71.38** | **92.32** |
| VICReg | 91.68 | 99.66 | 68.56 | 90.76 |
| **SSL-EY** | 91.43 | **99.75** | 67.52 | 90.17 |

**Table 4.1:** Performance comparison of SSL methods on CIFAR-10 and CIFAR-100.

for SSL-EY but used hyperparameters for Barlow Twins and VICReg that had been heavily optimized in previous studies.

# 5 Further Experiments with CIFAR-10 and CIFAR-100

**Model Convergence:** The Learning curves in Figure V.7 indicate that the performance variation at 1,000 epochs in table 5.1 mainly results from optimization noise and speed of convergence is similar.

**Smaller Projector or None at All:** One key motivation for projectors is to prevent excessive collapse of meaningful information. Because SSL-EY learns does not suffer from collapse, we had a prior that it may be more robust to projector size, and perhaps even to removing the projector altogether. For this reason, in another set of experiments, we explored varying the projector's output dimensions from 2048 to 64 and removing the projector completely while holding the encoder output size constant. Figure V.8a demonstrates that SSL-EY maintains good performance even with a smaller projector, making the representations more efficient than Barlow Twins and VICReg (they contain the same amount of useful information for the classification task in much fewer dimensions). While Figure V.8a shows the strong performance of Barlow Twins and VICReg at larger projector sizes for this task, we would argue that our objective is more robust to this design choice, potentially offering a more reliable choice for practitioners employing SSL to unfamiliar datasets. At the bottom of Table 5.1, we further highlight the efficiency of SSL-EY by showing that our model performs similarly when we have no projector (just using the a 2048 dimensional representation), suggesting that SSL-EY is less reliant on this architecture[3]. In contrast, we show in appendix **??** that Barlow Twins and VICReg's performance drops substantially without the use of a projector.

**$\mathcal{L}_{\mathbf{EY}}$ is an informative metric:** Figure V.8b offers two key insights. First, it shows that the EY loss, which provides an unbiased estimate of the canonical correlations of the embeddings, is closely related to classification accuracy. This suggests that

[3]We note that W-MSE, a close relative of our work, also didn't use a projector despite its use being seemingly ubiquitous

maximizing canonical correlation is a promising pretext task for self-supervised learning. Second, the figure reveals that even a reduced-dimensionality projector output (64 dimensions) has not reached its full capacity by 1,000 epochs. Specifically, the sum of squared canonical correlations reaches 46, out of a maximum possible value of 64. This indicates that there is still room for further optimization, implying that SSL-EY's representations have not yet saturated their capacity for capturing meaningful information. Lastly, the evolution of the correlation, as measured by $\mathcal{L}_{\text{EY}}$, offers a novel way of monitoring model training even without the need for a separate validation task like classification, and could potentially eliminate the requirement for a validation set altogether. This is a particularly interesting direction given recent work on the stepwise eigenvalue behavior of the representations in SSL models **simon2023stepwise**.

| Method | CIFAR-10 Top-1 | CIFAR-10 Top-5 | CIFAR-100 Top-1 | CIFAR-100 Top-5 |
|---|---|---|---|---|
| Barlow Twins | **92.1** | 99.73 | **71.38** | **92.32** |
| VICReg | 91.68 | 99.66 | 68.56 | 90.76 |
| **SSL-EY** | 91.43 | **99.75** | 67.52 | 90.17 |
| SSL-EY No Proj. | 90.98 | 99.69 | 65.21 | 88.09 |

**Table 5.1:** Performance comparison of SSL methods on CIFAR-10 and CIFAR-100.

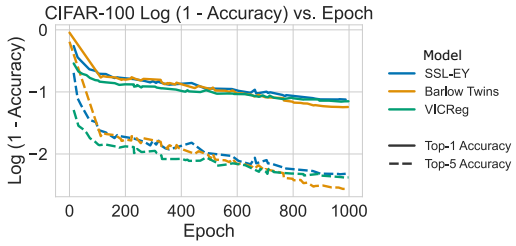
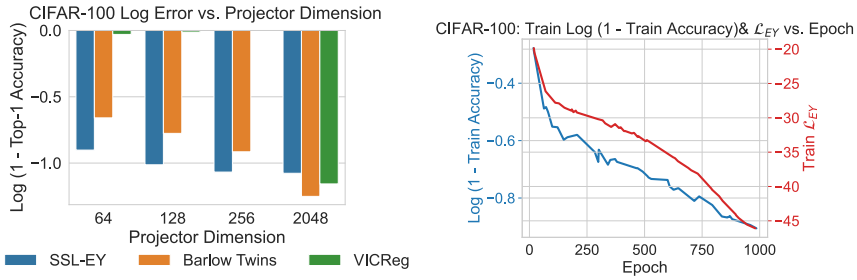



**Figure V.7:** Learning curves for SSL-EY, Barlow Twins, and VICReg on CIFAR-100, showing performance across 1,000 epochs.




**(a)** **(b)**

**Figure V.8:** (a) Performance of SSL-EY with reduced projector size compared to Barlow Twins and VICReg. (b) SSL-EY's learned embeddings indicate untapped representation capacity.

# 6 Conclusion

In this paper, we introduced a class of efficient, scalable algorithms for Canonical Correlation Analysis and Self-Supervised Learning, rooted in a novel unconstrained loss function. These algorithms are computationally lightweight, making them uniquely suited for large-scale problems where traditional methods struggle.

We have two distinct avenues for future research. Firstly, we aim to incorporate regularization techniques to improve both generalizability and interpretability, building upon existing sparse methods in CCA (D. M. Witten and R. J. Tibshirani, 2009). We also intend to investigate the utility of correlation as a metric for measuring the quality of learnt representations. This holds the potential to replace traditional validation methods like classification accuracy, especially in situations where validation labels are not available.

In summary, this paper sets a new benchmark for addressing large-scale CCA problems and opens new avenues in self-supervised learning, paving the way for more accessible and efficient solutions in various applications.

# Chapter VI

# CCA-Zoo: A collection of Regularized, Deep Learning-based, Kernel, and Probabilistic methods in a scikit-learn style framework

## 1 Introduction

The Python programming language has seen a surge in popularity in the machine learning community due to its versatility and extensive libraries. However, when it comes to the domain of multiview learning, there is a noticeable void in the Python ecosystem. Existing libraries, such as `scikit-learn`Pedregosa et al., 2011, offer basic implementations for Canonical Correlation Analysis (CCA) and Partial Least Squares (PLS), yet fall short of providing a comprehensive toolkit for multiview learning techniques. This is particularly striking given the widespread recognition that the availability of quality software implementations often acts as a catalyst for the adoption of novel methodologies in the statistical learning community.

One glaring example of this trend is Sparse PLS. Despite its known limitations, Sparse PLS has effectively become the go-to method for sparse CCA applications,

primarily due to its robust implementation in the R programming language. The discrepancy between the availability of multiview learning tools in R and Python has not only hindered the diversification of methodologies but also impeded the community from leveraging the more recent advances in the field.

# 2 Background

In recent years, the research community has shown a heightened interest in multiview learning, as evident from the proliferation of scholarly articles and the diversification of use-cases ranging from bioinformatics to natural language processing. Traditionally, this field has been dominated by contributions from statistical learning researchers who predominantly utilized R and MATLAB for their work. These platforms have been the birthplace of many state-of-the-art algorithms and methodologies, including but not limited to Sparse PLS.

However, this posed a challenge for Python-oriented researchers and practitioners, leaving them with two less-than-ideal options: either port existing R or MATLAB code into Python, often a non-trivial task requiring domain expertise, or resort to using the limited set of methods available in native Python libraries like `scikit-learn`. This fragmentation has, in effect, created barriers to entry and possibly slowed down the progress in applying multiview learning techniques in Python-based projects.

The `CCA-Zoo` package aims to bridge this divide by offering a broad range of multiview learning algorithms, creating a unified platform that fosters both academic research and practical applications in Python.

# 3 Methods

In this section, we describe the implementation of `CCA-Zoo` and the design decisions that were made during its development. We will also demonstrate how the package is optimised for use with high-dimensional biomedical data.

## 3.1 API

The `scikit-learn` API is familiar to many machine learning practitioners and researchers, and is the de facto standard for machine learning in Python. Users of machine learning libraries in Python expect a consistent API, and so we have designed `CCA-Zoo` to be compatible with the `scikit-learn` API. Moreover, by using the `scikit-learn` API, `CCA-Zoo` is compatible with the `scikit-learn` ecosystem.

The major barrier to using the `scikit-learn` API for multiview learning is that the `scikit-learn` API is designed for single-view data. In particular, the `scikit-learn` API for fitting a model is `model.fit(X, y)`, where `model` is an instance of a model, `X` is a matrix of features, and `y` is a vector of labels.

In multiview learning, we have multiple views of data `X`, and so we need to be able to fit a model with multiple views of data. For this reason, models in `CCA-Zoo` have a `fit` method that takes a list of views as input, i.e. `model.fit([X1, X2, X3], y)`.

## 3.2 Code Availability

The code for `CCA-Zoo` is available at.

`CCA-Zoo` has received 155 stars and 30 forks on GitHub, and has nearly 500 downloads per month on PyPI [1].

I am particularly proud of the diversity of projects CCA-Zoo has found use in outside of medical imaging given that the goal was to create a general purpose multiview learning library.

# 4 Benchmarking

### 4.0.1 Objective

The objective of the benchmarking experiments is to compare the performance of `CCA-Zoo` against `scikit-learn`, focusing on the efficiency of Canonical Correlation Analysis (CCA) and Partial Least Squares (PLS) methods. We conducted experiments on synthetic datasets with varying dimensions to evaluate their average execution time.

### 4.0.2 Experimental Setup

The datasets consisted of random matrices with a varying number of dimensions: $50$, $100$, $200$, $400$, and $800$. Each matrix had $100$ samples. We set the latent dimensions for both CCA and PLS to $10$. For each dimension, the experiment was repeated $10$ times to obtain reliable performance metrics.

**Libraries Used:**

[1] https://pypistats.org/packages/cca-zoo

- `CCA-Zoo` (version: X.X.X)
- `Scikit-learn` (version: X.X.X)

### 4.0.3 Results

**Canonical Correlation Analysis:** Figure VI.1 presents the comparison between `CCA-Zoo` and `scikit-learn` for Canonical Correlation Analysis. We observe that `CCA-Zoo` exhibits a competitive runtime profile when compared to `scikit-learn` across all dimensions.

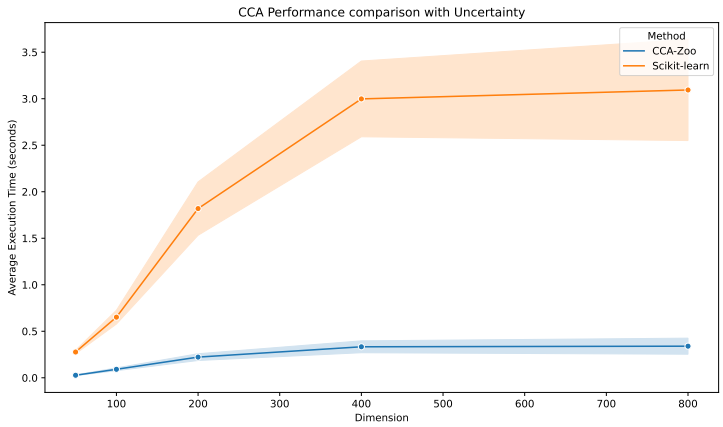



**Figure VI.1:** Performance comparison for CCA methods

**Partial Least Squares:** The comparison for Partial Least Squares is shown in Figure VI.2. Like the CCA experiment, `CCA-Zoo` maintains a robust performance profile that is competitive with `scikit-learn`.

### 4.0.4 Discussion

The results indicate that `CCA-Zoo` is an efficient Python package for both CCA and PLS methods, holding its own against the widely-used `scikit-learn` library. These experiments underscore the capability of `CCA-Zoo` to handle high-dimensional data efficiently, making it a suitable choice for applications in bioinformatics, natural language processing, and other high-dimensional data domains.

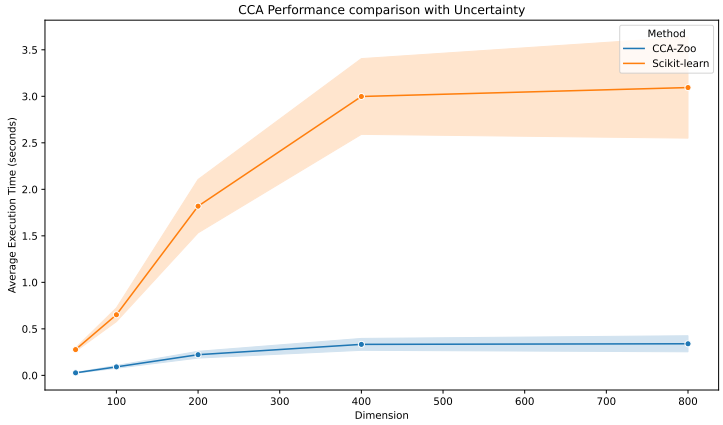



**Figure VI.2:** Performance comparison for PLS methods

## 4.1 Conclusion

`CCA-Zoo` has not only served as a tool for my research but aims to be a community resource that can accelerate research and application in multiview learning. Its design decisions, such as API compatibility and focus on both linear and deep models, reflect a comprehensive understanding of the challenges and opportunities in this field.

# References


Andrew, Galen et al. (2013). “Deep canonical correlation analysis”. In: *International conference on machine learning*. PMLR, pp. 1247–1255.

Bach, Francis R and Michael I Jordan (2005). “A probabilistic interpretation of canonical correlation analysis”. In: URL: https://statistics.berkeley.edu/sites/default/files/tech-reports/688.pdf.

Baldassarre, Luca, Janaina Mourao-Miranda, and Massimiliano Pontil (2012). “Structured sparsity models for brain decoding from fMRI data”. In: *2012 Second International Workshop on Pattern Recognition in NeuroImaging*. IEEE, pp. 5–8.

Balestriero, Randall et al. (2023). “A Cookbook of Self-Supervised Learning”. In: *arXiv preprint arXiv:2304.12210*.

Bardes, Adrien, Jean Ponce, and Yann LeCun (2021). “Vicreg: Variance-invariance-covariance regularization for self-supervised learning”. In: *arXiv preprint arXiv:2105.04906*.

Biewald, Lukas (2020). *Experiment Tracking with Weights and Biases*. Software available from wandb.com. URL: https://www.wandb.com/.

Bogdan, Paul C et al. (2023). “ConnSearch: A framework for functional connectivity analysis designed for interpretability and effectiveness at limited sample sizes”. In: *NeuroImage* 278, p. 120274.

Boyd, Stephen et al. (2011). “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine learning* 3.1, pp. 1–122.

Button, Katherine S et al. (2013). “Power failure: why small sample size undermines the reliability of neuroscience”. In: *Nature reviews neuroscience* 14.5, pp. 365–376.

Bzdok, Danilo, Thomas E Nichols, and Stephen M Smith (2019). “Towards algorithmic analytics for large-scale datasets”. In: *Nature Machine Intelligence* 1.7, pp. 296–306.

Bzdok, Danilo and B.T. Thomas Yeo (2017). “Inference in the age of big data: Future perspectives on neuroscience”. In: *NeuroImage* 155, pp. 549–564. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2017.04.061. URL: https://www.sciencedirect.com/science/article/pii/S1053811917303816.

Carroll, J Douglas (1968). “Generalization of canonical correlation analysis to three or more sets of variables”. In: *Proceedings of the 76th annual convention of the American Psychological Association*. Vol. 3. Washington, DC, pp. 227–228.

Chang, Xiaobin, Tao Xiang, and Timothy M Hospedales (2018). “Scalable and effective deep CCA via soft decorrelation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1488–1497.

Chapman, James, Ana Lawry Aguila, and Lennie Wells (2022). “A Generalized EigenGame with Extensions to Multiview Representation Learning”. In: *arXiv preprint arXiv:2211.11323*.

Chapman, James, Lennie Wells, and Ana Lawry Aguila (2023). *Efficient Algorithms for the CCA Family: Unconstrained Objectives with Unbiased Gradients*. arXiv: 2310.01012 [cs.LG].

Chen, Mengjie et al. (2013). “Sparse CCA via precision adjusted iterative thresholding”. In: *arXiv preprint arXiv:1311.6186*.

Chen, Zhehui et al. (2019). “On constrained nonconvex stochastic optimization: A case study for generalized eigenvalue decomposition”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 916–925.

Chi, Eric C. et al. (2013). “Imaging genetics via sparse canonical correlation analysis”. In: *2013 IEEE 10th International Symposium on Biomedical Imaging*, pp. 740–743. DOI: 10.1109/ISBI.2013.6556581.

Chun, Hyonho and Sündüz Keleş (2010). “Sparse partial least squares regression for simultaneous dimension reduction and variable selection”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72.1, pp. 3–25.

Cruciani, Federica et al. (2022). “What PLS can still do for Imaging Genetics in Alzheimer’s disease”. In: *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, pp. 1–4.

Da Costa, Victor Guilherme Turrisi et al. (2022). “solo-learn: A Library of Self-supervised Methods for Visual Representation Learning.” In: *J. Mach. Learn. Res.* 23.56, pp. 1–6.

De Pierrefeu, Amicie et al. (2017). “Structured sparse principal components analysis with the TV-elastic net penalty”. In: *IEEE transactions on medical imaging* 37.2, pp. 396–407.

Demontis, Ditte et al. (Feb. 2023). “Genome-wide analyses of ADHD identify 27 risk loci, refine the genetic architecture and implicate several cognitive domains”. en. In: *Nat. Genet.* 55.2, pp. 198–208.

Dinga, Richard et al. (2019). “Evaluating the evidence for biotypes of depression: Methodological replication and extension of”. In: *NeuroImage: Clinical* 22, p. 101796.

Dohmatob, Elvis Dopgima et al. (2014). “Benchmarking solvers for TV-L1 least-squares and logistic regression in brain imaging”. In: *2014 International Workshop on Pattern Recognition in Neuroimaging*. IEEE, pp. 1–4.

Drysdale, Andrew T et al. (2017). “Resting-state connectivity biomarkers define neurophysiological subtypes of depression”. In: *Nature medicine* 23.1, pp. 28–38.

Engl, Heinz Werner, Martin Hanke, and Andreas Neubauer (1996). *Regularization of inverse problems*. Vol. 375. Springer Science & Business Media.

Euesden, Jack, Cathryn M. Lewis, and Paul F. O’Reilly (Dec. 2014). “PRSice: Polygenic Risk Score software”. In: *Bioinformatics* 31.9, pp. 1466–1468. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu848. eprint: https://academic.oup.com/bioinformatics/article-pdf/31/9/1466/50306478/bioinformatics_31_9_1466.pdf. URL: https://doi.org/10.1093/bioinformatics/btu848.

Ferreira, Fabio S et al. (2022). “A hierarchical Bayesian model to find brain-behaviour associations in incomplete data sets”. In: *NeuroImage* 249, p. 118854.

Fischl, Bruce (Aug. 2012). “FreeSurfer”. en. In: *Neuroimage* 62.2, pp. 774–781.

Fu, Xiao et al. (2017). “Scalable and flexible Max-Var generalized canonical correlation analysis via alternating optimization”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5855–5859.

Galton, Francis (1907). “Vox populi”. In: *Nature* 75.1949, pp. 450–451.

Ge, Rong, Chi Jin, and Yi Zheng (July 2017). “No Spurious Local Minima in Nonconvex Low Rank Problems: A Unified Geometric Analysis”. en. In: *Proceedings of the 34th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, pp. 1233–1242. URL: https://proceedings.mlr.press/v70/ge17a.html (visited on 05/16/2023).

Gemp, Ian, Charlie Chen, and Brian McWilliams (2022). “The Generalized Eigenvalue Problem as a Nash Equilibrium”. In: *arXiv preprint arXiv:2206.04993*.

Gemp, Ian, Brian McWilliams, et al. (2021). *EigenGame Unloaded: When playing games is better than optimizing*. arXiv: 2102.04152 [stat.ML].

Gemp, Ian M. et al. (2020). “EigenGame: PCA as a Nash Equilibrium”. In: *CoRR* abs/2010.00554. arXiv: 2010.00554. URL: https://arxiv.org/abs/2010.00554.

Golub, Gene H and Hongyuan Zha (1995). “The canonical correlations of matrix pairs and their numerical computation”. In: *Linear algebra for signal processing*. Springer, pp. 27–49. DOI: 10.1007/978-1-4612-4228-4_3.

Grosenick, Logan et al. (2013). “Interpretable whole-brain prediction analysis with GraphNet”. In: *NeuroImage* 72, pp. 304–321.

Haufe, Stefan et al. (2014). “On the interpretation of weight vectors of linear models in multivariate neuroimaging”. In: *Neuroimage* 87, pp. 96–110.

Helmer, Markus et al. (2020). “On stability of Canonical Correlation Analysis and Partial Least Squares with application to brain-behavior associations”. In: *bioRxiv*.

Höskuldsson, Agnar (1988). “PLS regression methods”. In: *Journal of chemometrics* 2.3, pp. 211–228.

Hotelling, Harold (1933). “Analysis of a complex of statistical variables into principal components.” In: *Journal of educational psychology* 24.6, p. 417.

International League Against Epilepsy Consortium on Complex Epilepsies (Dec. 2018). “Genome-wide mega-analysis identifies 16 loci and highlights diverse biological mechanisms in the common epilepsies”. en. In: *Nat. Commun.* 9.1, p. 5269.

Kanatsoulis, Charilaos I et al. (2018). “Structured SUMCOR multiview canonical correlation analysis for large-scale data”. In: *IEEE Transactions on Signal Processing* 67.2, pp. 306–319.

Kettenring, Jon R (1971). “Canonical analysis of several sets of variables”. In: *Biometrika* 58.3, pp. 433–451.

Klami, Arto, Seppo Virtanen, and Samuel Kaski (2013). “Bayesian Canonical correlation analysis.” In: *Journal of Machine Learning Research* 14.4.

Krishnan, Anjali et al. (2011). “Partial Least Squares (PLS) methods for neuroimaging: a tutorial and review”. In: *Neuroimage* 56.2, pp. 455–475.

Lambert, J C et al. (Dec. 2013). “Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for Alzheimer’s disease”. en. In: *Nat. Genet.* 45.12, pp. 1452–1458.

Le Floch, Édith et al. (2012). “Significant correlation between a set of genetic polymorphisms and a functional brain network revealed by feature selection and sparse Partial Least Squares”. In: *NeuroImage* 63.1, pp. 11–24. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2012.06.061. URL: https://www.sciencedirect.com/science/article/pii/S1053811912006775.

Lindenbaum, Ofir et al. (2021). “L0-sparse canonical correlation analysis”. In: *International Conference on Learning Representations*.

Mackey, Lester (2008). “Deflation methods for sparse PCA”. In: *Advances in neural information processing systems* 21.

Mai, Qing and Xin Zhang (2019). “An iterative penalized least squares approach to sparse canonical correlation analysis”. In: *Biometrics* 75.3, pp. 734–744. DOI: 10.1111/biom.13043.

McIntosh, Anthony R (2021). “Comparison of Canonical Correlation and Partial Least Squares analyses of simulated and empirical data”. In: *arXiv preprint arXiv:2107.06867*.

Meng, Zihang, Rudrasis Chakraborty, and Vikas Singh (2021). “An Online Riemannian PCA for Stochastic Canonical Correlation Analysis”. In: *Advances in Neural Information Processing Systems* 34, pp. 14056–14068.

Meredith, William (1964). “Canonical correlations with fallible data”. In: *Psychometrika* 29.1, pp. 55–65.

Michel, Vincent et al. (2011). “Total variation regularization for fMRI-based prediction of behavior”. In: *IEEE transactions on medical imaging* 30.7, pp. 1328–1340.

Mihalik, Agoston, James Chapman, et al. (2022). “Canonical correlation analysis and partial least squares for identifying brain-behaviour associations: a tutorial and a comparative study”. In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*.

Mihalik, Agoston, Fabio S Ferreira, Michael Moutoussis, et al. (2020). “Multiple holdouts with stability: Improving the generalizability of machine learning analyses of brain–behavior relationships”. In: *Biological psychiatry* 87.4, pp. 368–376.

Mihalik, Agoston, Fabio S Ferreira, Maria J Rosa, et al. (2019). “Brain-behaviour modes of covariation in healthy and clinically depressed young people”. In: *Scientific reports* 9.1, pp. 1–11.

Monteiro, João M et al. (2016). “A multiple hold-out framework for Sparse Partial Least Squares”. In: *Journal of neuroscience methods* 271, pp. 182–194.

Mullins, Niamh et al. (June 2021). “Genome-wide association study of more than 40,000 bipolar disorder cases provides new insights into the underlying biology”. en. In: *Nat. Genet.* 53.6, pp. 817–829.

Nalls, Mike A et al. (Dec. 2019). “Identification of novel risk loci, causal insights, and heritable risk for Parkinson’s disease: a meta-analysis of genome-wide association studies”. en. In: *Lancet Neurol.* 18.12, pp. 1091–1102.

Nguyen, Nam D and Daifeng Wang (2020). “Multiview learning for understanding functional multiomics”. In: *PLoS computational biology* 16.4, e1007677.

Parkhomenko, Elena, David Tritchler, and Joseph Beyene (2009). “Sparse canonical correlation analysis with application to genomic data integration”. In: *Statistical applications in genetics and molecular biology* 8.1, pp. 1–34. DOI: 10.2202/1544-6115.1406.

Pedregosa, Fabian et al. (2011). “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12, pp. 2825–2830.

Purcell, Shaun et al. (Sept. 2007). “PLINK: a tool set for whole-genome association and population-based linkage analyses”. en. In: *Am. J. Hum. Genet.* 81.3, pp. 559–575.

Qi, Jun and Javier Tejedor (2016). “Deep multi-view representation learning for multi-modal features of the schizophrenia and schizo-affective disorder”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 952–956.

Reichenbach, Hans (1956). *The direction of time*. Vol. 65. Univ of California Press.

Rheenen, Wouter van et al. (Dec. 2021). “Common and rare variant association analyses in amyotrophic lateral sclerosis identify 15 risk loci with distinct genetic architectures and neuron-specific biology”. en. In: *Nat. Genet.* 53.12, pp. 1636–1648.

Rosipal, Roman and Nicole Krämer (2005). “Overview and recent advances in partial least squares”. In: *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*. Springer, pp. 34–51.

Smith, Stephen M et al. (2015). “A positive-negative mode of population covariation links brain connectivity, demographics and behavior”. In: *Nature neuroscience* 18.11, p. 1565.

Smith, Stephen M. and Thomas E. Nichols (2018). “Statistical Challenges in “Big Data” Human Neuroimaging”. In: *Neuron* 97.2, pp. 263–268. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2017.12.018. URL: https://www.sciencedirect.com/science/article/pii/S0896627317311418.

Stewart, G. W. and Ji-Guang Sun (July 1990). *Matrix Perturbation Theory*. en. Google-Books-ID: bIYEogEACAAJ. ACADEMIC PressINC. ISBN: 978-1-4933-0199-7.

Sudlow, Cathie et al. (2015). “UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age”. In: *PLoS medicine* 12.3, e1001779.

Sun, Liang, Shuiwang Ji, and Jieping Ye (2008). “A least squares formulation for canonical correlation analysis”. In: *Proceedings of the 25th international conference on Machine learning*, pp. 1024–1031.

Suo, Xiaotong et al. (2017). “Sparse canonical correlation analysis”. In: *arXiv preprint arXiv:1705.10865*.

Taquet, Maxime et al. (June 2021). “A structural brain network of genetic vulnerability to psychiatric illness”. en. In: *Mol. Psychiatry* 26.6, pp. 2089–2100.

Tipping, Michael E and Christopher M Bishop (1999). “Probabilistic principal component analysis”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 61.3, pp. 611–622.

Tong, Shengbang et al. (2023). “EMP-SSL: Towards Self-Supervised Learning in One Training Epoch”. In: *arXiv preprint arXiv:2304.03977*.

Trubetskoy, Vassily et al. (Apr. 2022). “Mapping genomic loci implicates genes and synaptic biology in schizophrenia”. en. In: *Nature* 604.7906, pp. 502–508.

Uurtio, Viivi et al. (2017). “A tutorial on canonical correlation methods”. In: *ACM Computing Surveys (CSUR)* 50.6, pp. 1–33.

Vinod, Hrishikesh D (1976). “Canonical ridge and econometrics of joint production”. In: *Journal of econometrics* 4.2, pp. 147–166.

Virtanen, Seppo, Arto Klami, and Samuel Kaski (2011). “Bayesian CCA via group sparsity”. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 457–464.

Waaijenborg, Sandra, Philip C Verselewel de Witt Hamer, and Aeilko H Zwinderman (2008). “Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis”. In: *Statistical applications in genetics and molecular biology* 7.1.

Wang, Hao-Ting et al. (2020). “Finding the needle in a high-dimensional haystack: Canonical correlation analysis for neuroscientists”. In: *NeuroImage* 216, p. 116745.

Wang, Weiran, Raman Arora, Karen Livescu, and Jeff A Bilmes (2015). “Unsupervised learning of acoustic features via deep canonical correlation analysis”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4590–4594.

Wang, Weiran, Raman Arora, Karen Livescu, and Nathan Srebro (2015). “Stochastic optimization for deep CCA via nonlinear orthogonal iterations”. In: *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, pp. 688–695.

Wilms, Ines and Christophe Croux (2015). “Sparse canonical correlation analysis from a predictive point of view”. In: *Biometrical Journal* 57.5, pp. 834–851.

Witten, Daniela et al. (2013). “Package ‘pma’”. In: *Genetics and Molecular Biology* 8.1, p. 28.

Witten, Daniela M, Robert Tibshirani, and Trevor Hastie (2009). “A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis”. In: *Biostatistics* 10.3, pp. 515–534.

Witten, Daniela M and Robert J Tibshirani (2009). “Extensions of sparse canonical correlation analysis with applications to genomic data”. In: *Statistical applications in genetics and molecular biology* 8.1.

Wold, Herman (1973). “Nonlinear Iterative Partial Least Squares (NIPALS) Modelling: Some Current Developments”. In: *Multivariate Analysis–III*. Ed. by PARUCHURI R. KRISHNAIAH. Academic Press, pp. 383–407. ISBN: 978-0-12-426653-7. DOI: https://doi.org/10.1016/B978-0-12-426653-7.50032-6. URL: https://www.sciencedirect.com/science/article/pii/B9780124266537500326.

Yeo, BT Thomas et al. (2011). “The organization of the human cerebral cortex estimated by intrinsic functional connectivity”. In: *Journal of neurophysiology*.

Zbontar, Jure et al. (2021). “Barlow twins: Self-supervised learning via redundancy reduction”. In: *arXiv preprint arXiv:2103.03230*.

Zhuang, Xiaowei, Zhengshi Yang, and Dietmar Cordes (2020). “A technical review of canonical correlation analysis for neuroscience applications”. In: *Human Brain Mapping* 41.13, pp. 3807–3833.

Zong, Yongshuo, Oisin Mac Aodha, and Timothy Hospedales (2023). “Self-Supervised Multimodal Learning: A Survey”. In: *arXiv preprint arXiv:2304.01008*.